A Novel Hybrid Classifier for Recognition of Handwritten Numerals

Ping Zhang, Lihui Chen, Alex C Kot

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798

Abstract

A hybrid neural network and tree classification system for handwritten numeral recognition is proposed. The recognition system consists of coarse and fine classification based on a variety of stable and reliable global features and local features. For the coarse classifier: a four-layer feed forward neural networks with back propagation learning algorithm is employed to distinguish six subsets {0}, {6}, {8}, {1,7}, {4,9}, {2,3,5} based on the similarity of character's geometrical features. Three character classes {0},{6} and {8} are directly recognized from ANN. For each of the last three subsets, a decision tree classifier is built for a fine classification as follows: Firstly, the specific feature-class relationship is heuristically and empirically created between the feature primitives and corresponding semantic class. Then, an iterative growing and pruning algorithm is used to form a tree classifier. Experiments demonstrated that the proposed hybrid recognition system is robust and flexible, which can achieve a high recognition rate.

Keywords: Handwritten numeral recognition, Feature extraction, Tree decision classifier, Neural networks.

1 Introduction

Handwritten character recognition, with extensive variety of writing styles, has been an active research field for long time due to its potential commercial perspective. Many methodologies have been proposed and various character recognition systems have been commercialized in recently years. However, there still exists room in pursuit of a higher recognition rate and faster processing time for real world applications such as the recognition of severely, omnifont machine-printed and unconstrained handwritten characters.

Two of the commonly used classifiers for character recognition are Artificial Neural Networks(ANN) classifier and Decision Tree (DT) classifier. An ANN classifier, due to its useful properties such as the highly parallel mechanism, excellent fault tolerance, adaptation, self-learning, has been increasingly developed and successfully used in character recognition [1,2,3,4,5]. However, the decision making processing of a neural network is difficult to understand. On the contrary, a decision tree classifier, because of its simplicity and computational efficiency, has long been investigated. A large variety of methods have been proposed for the design of a classification tree [6,7,8,9,10]. Recently, some researchers have successfully combined ANN with DT to automatically design decision trees for various applications [11,12,13]. However, given a set of suitable features, how to create an optimal decision tree for solving the multiple recognition problems is still a challenging research topic.

A pattern classifier uses a series of tests or decision functions to determine the identity of an unknown pattern or object. The evaluation of a classifier is planned in such a way that the successive outcome reduces uncertainty about the unknown patterns being considered for the classification. A more challenging approach is to configure a classification system by using a set of suitable features. The paper is organized as follows: In Section 2, a variety of the global features and local features are briefly introduced. In section 3, A decision tree growing and pruning algorithm is reviewed. In Section 4, a hybrid recognition system is proposed. The relationship between the features and corresponding classes is discussed. In section 5, the comparisons of the recognition rate between ANN classifier and the proposed hybrid classifier are conducted. Finally, the conclusions are given in section 6.

2 Feature Extraction

In the character recognition system, two kinds of features, namely global features and local features are used here. The detailed discussion on those features can be found in [14]. A brief introduction to those features is given here to provide the completeness.

2.1 Global Feature

- **Middle line feature**

This feature consists of a set of middle points between two neighboring strokes, in which the middle line can be established from horizontal direction or vertical direction, respectively. In this paper, only vertical middle features are used. If the beginning/end point of

0-7803-6583-6/00/$10.00 © 2000 IEEE

the middle line is a cross point of two adjacent strokes, we define the point status is *closed(1)*; otherwise it is *opened(0)*. The position of beginning/end point and the open/close status of the point can be encoded as the middle line features.

Some of the middle lines extracted from the character image using the method above are shown in Fig.1 which are indicated by the thin lines.

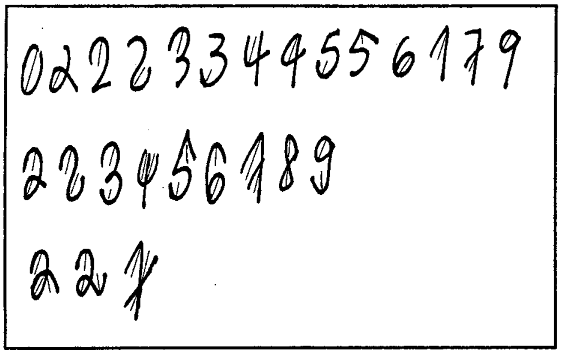

Fig.1 Middle Lines extracted

- **Concave feature**

Concave feature describes the concavity in character's outer profiles from the top/bottom/left/right direction point of view. The concave features of character "2" and "8" are shown in Fig.2 indicated by an arrow. The position and the direction of a concave can be encoded as the concave feature.

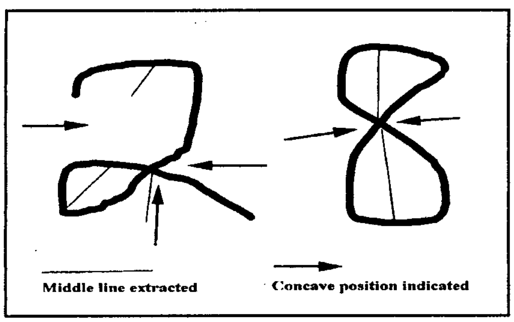

Fig.2 Concave Feature

- **Width feature**

A normalized character is sliced into 4 sub-regions equally along the vertical direction. The maximum width of each sub-region is calculated and encoded to form the feature.

- **Point feature**

End points, Branch points, and Cross points features defined in [15] are applied in this system. These features are easily extractable, and can be encoded as the Point Features.

2.2 Local Feature

A floating feature detector (FFD) introduced in [14] is used to detect some tiny segments in a character image. Those features are used as local features. FFD can move along the edge of a character to detect any horizontal-like and vertical-like segments.

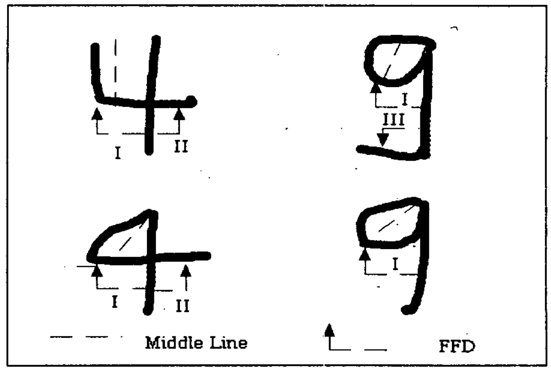

Fig.3 Floating detector

For distinguishing two writing style characters "4" and "9" indicated in Fig.3. FFD(I) can detect a left-profile segment in both character"4" and "9", However, FFD(II) can only extract a right-profile segment in character "4". Combined with the middle line open/close status, a decision tree can be constructed which will be elaborated in Section 4.

3 Binary Decision Tree Classifier

There are three major parts in the design of a DT classifier.

1) The choice of an appropriate tree structure,

2) The choice of feature subsets to be used at each internal node,

3) The choice of the decision rule or strategy to be used at each internal node.

We will address the issues related to build up an appropriate tree for classification in this section. For a binary DT classifier, a set of decision rules is used to assign an unknown sample to a pattern class. A hierarchical structure is noted as a tree (T) consisting of several levels: Level 0 contains a node, called root node. Level 1 contains the node 2 and 3; level 2 contains the nodes 4,5,6,and 7 and so on. Specifically, level i contains 2^i nodes numbering from 2^i to $2^{(i+1)}-1$. Nodes with descending branches are the Non-Terminal Nodes (NTN). Nodes without descending branches are the Terminal Nodes (TN). Each NTN contains a decision rule. Each TN belongs to one of the recognized classes.

A tree can be grown by recursively finding splitting rules until all the terminal nodes have a pure or nearly pure class membership or cannot be split further. According to [11], let N be the number of training samples, N(t) be the number of training samples which land in node t, $N_j(t)$ be the number of training samples which land in node t and belong to class j, and M be the number of classes to be classified. Define:

$$P(t) = N(t)/N$$
$$P_L(t) = P(t_L)/ P(t)$$
$$P_R(t) = P(t_R)/P(t)$$
$$P(j/t) = N_j(t)/N(t) \qquad (1)$$

Where P(t) is the probability that a randomly selected training sample lands in node t, $P_L(t)$ ($P_R(t)$) is the conditional probability that training samples belong to left t_L (right t_R) branch given it lands in node t, and P(j/t) is the conditional probability that the training sample belongs to class j given it lands in node t. Define a tree splitting criterion based on a node impurity function such as the Gini criterion[6] by:

$$g(t) = \sum_i \sum_{j \neq i} P(i/t).P(j/t)$$

Next, define the change in node impurity $\Delta G(f,\theta,t)$ due to a split at node t with two parameters: the feature vector f and the threshold θ by:

$$\Delta G(f,\theta,t) = g(t) - g(t_L).P_L(t) - g(t_R).P_R(t) \qquad (2)$$

The best feature f* and the threshold θ* at node t can be obtained by maximizing the decrease in node impurity.

$$\Delta G(f^*,\theta^*,t) = \max_{f \in F,\theta} \{\Delta G(f,\theta,t)\} \qquad (3)$$

Where F is the feature set. In our recognition system, f is chosen from a global and local feature space F, whereas the recognized class $y \in \{1,2,3,\ldots 9,0\}$. A decision rule d(.) is a function that maps f into class y with d(f) representing the class label of a feature vector f. The misclassification rate of the decision classification tree is denoted by

$$R(T) = P(d(f) \neq y) \qquad (4)$$

In practical applications, the misclassification rate is simply estimated by the ratio of samples misclassified to the total number of the testing samples.

$$R(T) = N_{error}/N \qquad (5)$$

Where N_{error} is the number of samples such that d(f) $\neq$y. There is a guideline on how to find a pruned tree from a tree T. Suppose T1 is a pruned subtree of tree T if T1 has same root node as T and has fewer either NTN or NT. This is denoted by $T_1 < T$. For an optimal seeking pruned subtree T_1 from T, we construct many tree T' and satisfy:

$$R(T_1) = \min\{R(T'), \forall\, T' \leq T\} \qquad (6)$$

In this paper, an iterative growing and pruning algorithm[10] is adopted for the construction of a decision tree. The training algorithm is described as follows:

The training data is split into two independent sets, called the first and second training sets.

- A large tree is grown based on the first training set by splitting until all terminal nodes have a pure class membership, or have fewer than a specified number of samples, or cannot be split such that both descendents are nonempty.
- A pruned subtree is selected by minimizing the misclassification rate over the second training set.
- A tree is grown off of the terminal nodes of the selected pruned subtree based on the second training set by splitting until all terminal nodes have pure class memberships, or have fewer than a specified number of samples, or cannot be split such that both descendents are nonempty.
- A pruned subtree is selected by minimizing the misclassification rate over the first training set.

The procedure is then iterated, successively interchanging the roles of the first and second training sets. It will be shown that the process of selected pruned subtree converges. Then the tree is formed.

4 Recognition System

The block diagram of proposed system for handwritten numeral recognition is shown in Fig.4.

The recognition system consists of three main parts: namely feature extraction, coarse classification and fine classification. The methods described in Section 2 will be used in our system for feature extraction.

For coarse classification, a four-layer feed forward neural network with back-propagation algorithm is employed. Totally 98 bits of global features and 15 bits of local features are fed into the input layer. The output layer is composed of 6 nodes representing 6 character subsets {0},{6},{8},{1,7},{4,9}, {2,3,5}.

The network is a fully connected one with 20 units in each of the two hidden layers.

After being trained, the neural network can be used as a coarse classifier. Reducing the number of classes from 10 into 6 has made a great impact on speeding up learning and shortening training time. For the coarse classification, characters "0","6", and "8" can then be directly recognized from the trained ANN. The other three subsets need to be further classified through the fine classifier.

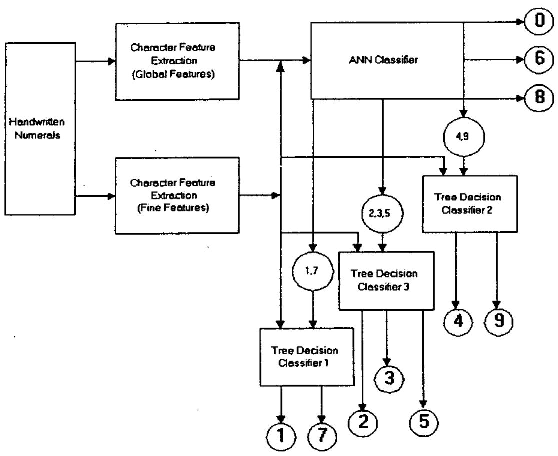

Fig.4 The system block diagram of the recognition system

For the classification of the remaining three subsets {1,7}, {4,9}, (2,3,5}. A special attention has been paid to the geometrical difference between (among) those characters in each subgroup in order to distinguish one from others. A heuristic and empirical method has been applied to build up the relationship between feature vectors and decision rule d(.).

For example, in order to distinguish 1 from 7 in {1,7} subset, two most stable and distinguishable features, which are the width feature and the local segment feature, are chosen to build up the relationship between the features and class.

Some decision rules d(.) can be deduced based on the above relationship such as:

If {(one or more segments in the left profile) or (the difference of width feature in different slices)}
The character belongs to "7"
Else
The character belongs to "1"

For subset {4,9}, the middle line features, the local segment features and the point features are used to describe the relationship between the features and the corresponding class. The overall features extracted by the proposed method are listed in Table I. Those features can form the feature vectors used to construct corresponding decision tree to distinguish between character "4" and "9".

Table I: the list of extracted features of character "4" and "9"

Char.	Open/Close status of the middle line		Segment detected by FFD		Point Features		
	Begin	End	Left	Right	End	Branch	Cross
4	0(1)	1	1	1	4(3)	1(2)	1
9	1	1	1	0	1	1	0

For subset {2,3,5}, some of global features such as the point features, the concave features, and the local segment features are chosen to deduct several decision rules. The similar principle used in distinguishing subset {4,9} can be applied to extract rules for the tree construction.

All the trees are generated by recursively partitioning the feature space in such a way that all of the terminal nodes belong to a specific class, then an iterative growing and pruning algorithm can be used to prune and grow the trees in pursuit of optimal design of the trees.

5 Experiments

10000 free handwritten numerals are collected written by 200 people with unconstrained writing style. 5000 out of those characters are used as training samples; the others are then input as testing samples. We have conducted two types of experiments.

For experiment one, a four-layer feed forward artificial neural network (ANN) was employed as the character recognizer. The global features and local features extracted are fed into the input layer. The network has a ten-output layer (10 units standing for characters from 0-9), and two hidden layers with 20 units each.

The network is trained by using the training samples, then a recognition performance is conducted by using testing samples. The overall recognition rates of an ANN recognizer are tabulated in Table II.

For experiment Two, the similar structure of an ANN used in experiment one is employed as the coarse character recognizer indicated in Fig.4. The difference is that the output layer only has 6 units to represent the six subsets {0},{6},{8},{1,7},{4,9},{2,3,5}.

The overall coarse recognition rate for the six subsets is shown in the second row of Table II. It is shown that the recognition rate has been improved. The remained characters will be further recognized by three tree classifications.

An iterative growing and pruning algorithm is employed to form the three decision tree classifiers. During the growing and pruning procedure, the training data were divided into two equal sub-training sets, and used to split and prune the tree in an iteration way. The Table III lists the average number of nodes in three decision trees.

The overall recognition rate of the hybrid classifier for testing data is summarized in the third row of Table II.

The Fig.5 shows the tendency of recognition rate versus the number of the training samples by using of ANN and the hybrid classification method respectively. For an ANN recognizer, when the number of the training samples increases, the convergence will become more and more difficult and the training will take much longer time. Sometime the whole system will confront collapse. From experiments, The conclusions can be drawn that (1): the proposed recognition system needs fewer training samples to achieve the same recognition rate, which is comparable with a pure ANN classifier. (2): when the training samples increase, although the difference of recognition rate of both classifiers tends to decrease, the hybrid classifier has a more stable convergence property and needs less computation.

Table II: The comparison of handwritten numeral recognition rate by ANN and proposed hybrid method

Classifier	Recognition rate(%)	Mis-Recognition rate(%)	Rejection rate(%)
ANN	97.50	1.20	1.30
Coarse Classification (six outputs)	98.60	0.50	0.90
Hybrid Classification	98.10	0.80	1. 10

Table III. The number of nodes in three decision tree

Decision Tree	The Number of Nodes
{1,7}	15
{2,3,5}	63
{4,9}	31

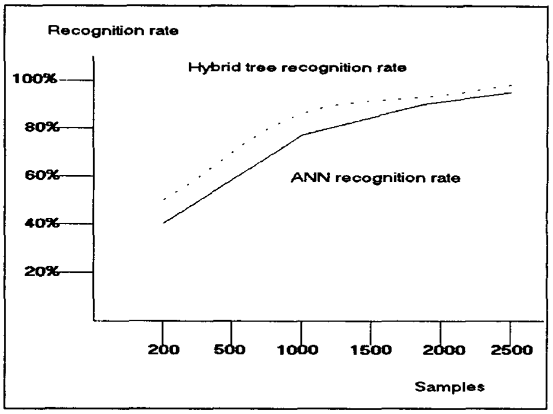

Fig.5 Recognition rate versus sample size for ANN, and proposed hybrid classifier

6 Conclusions

In this paper, the hybrid recognition system is proposed for handwritten numeral recognition. A set of global features and a floating feature detector are used in the system. The former depicts the global geometrical features of the characters, the later describes local features of the characters. The hybrid classifier proposed consists of two sub-classifiers, namely ANN coarse classifier and three decision tree classifiers. In the coarse classifier, the characters with large differences in geometric shapes are directly recognized, those characters with similar geometric structure need to be further classified. An iterative growing and pruning algorithm is adopted to build up three decision trees for recognizing those similar characters. Experiments demonstrated that our proposed system has improved the recognition rate, achieved a better convergence rate and needs less computation compared to that of a pure ANN recognizer.

Reference

[1] William E.Weideman, Michael T.Manry, Hung-Chun Yau, and Wei Gong, Comparisons of a neural network and a nearest-neighbor classifier via the numeric handprint recognition problem. IEEE Trans. Neural Networks, 1995,6(6), pp1524-1530.

[2] Bailing Zhang, Minyue Fu, Hong Yan, and Marwan A.Jabri, Handwritten digit recognition by adaptive-subspace self-organizing map, IEEE Trans. Neural Networks, 1999,10(4),pp 939-953.

[3] Paul.D.Gader, Mohamed Ali Khabou, Automatic feature generation for handwritten digit recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, 1996, 18(12), pp1256-1261.

[4] Sung-Bae Cho, Neural-network classifiers for recognizing totally unconstrained handwritten numerals, IEEE Trans. Neural Networks, 1997, 8(1), pp43-53.

[5] Adnan Amin, Humoudal-Sadoun and Stephen Fischer, Hand-printed Arabic character recognition system using an artificial network, Pattern Recognition,1996, 29(4), pp663-675.

[6] L.Breiman, J.H.Friedman, R.A.Olshen, and C.J.Stone, Classification and Regression Trees, Belmont, California, Wadsworth Int.1984.

[7] I.K.Sethi and G.P.R.Sarvarayudu, Hierarchical classifier design using mutual information. IEEE Trans. Pattern Analysis and Machine Intelligence. 1982, 4, pp441-445.

[8] J.R.Quinlan, Induction of decision trees, Machine Learning 1,1986,pp81-106.

[9] Q.R.Wang and C.Y.Suen, Large Tree Classifier with Heuristic Search and Global Training, IEEE Trans. Pattern Analysis and Machine Intelligence 1987,9(1),pp91-102.

[10] Saul B.Gelfand, C.S.Ravishankar, and Edward J.Delp, An iterative growing and pruning algorithm for classification tree design, IEEE Trans. Pattern Analysis and Machine Intelligence,1991, 13(2),pp163-174.

[11] Heng Guo, Saul B. Gelfand, Classification trees with neural network feature extraction, IEEE Trans. Neural Networks,1992, 3(6), pp923-933.

[12] Ishwar K.Sethi, Neural implementation of tree classifiers, IEEE Trans. Systems, Man, and Cybernetics, 1995,25(8),pp1243-1249.

[13] R.Krishna, G.Sivakumar, P.Bhattacharya, Extracting decision trees from trained neural networks. Pattern Recogition, 1999,32,pp1999-2009.

[14] Zhang Ping, Chen Lihui, Alex C Kot, A floating feature detector for handwritten numeral recognition, 15th International Conference on Pattern recognition , September 3-8,Barcelona 2000.

[15] Ø.D.Trier,Anil K.Jain and Torfinn Taxt, Feature extraction methods for character recognition-A survey, Pattern recognition,1996, 29(4), pp641-662.