HYBRID SCHEMES OF HOMOGENEOUS AND HETEROGENEOUS CLASSIFIERS FOR CURSIVE WORD RECOGNITION

JIN HO KIM[1,2], KYE KYUNG KIM[1] AND CHING Y. SUEN[1]

[1]*CENPARMI,Concordia University, Montreal, Quebec, Canada*
E-mail: {kjinho, kkkim, suen}@cenparmi.concordia.ca
[2]*Department of Electronic Engineering, Kyungil University,Kyungsan, Kyungpook, Korea*
E-mail: kjinho@bear.kyungil.ac.kr

Sophisticated hybrid schemes of the homogeneous and heterogeneous classifiers for cursive word recognition are presented. Two homogeneous MLPs (multi-layer perceptrons) are combined into a new single powerful classifier at the architectural level, and HMM (hidden Markov model) is added to the new classifier as a heterogeneous one at the output level. This is based on the idea that classifiers with more different methodologies and different features can better complement each other. The presented scheme achieves a recognition rate of 92.7% for English legal words of a CENPARMI database, a performance which is better than several previous hybrid schemes reported in the literature.

1 Introduction

The recognition of cursive words has been studied for several decades, and during this period many classifiers have been developed[1-7]. However none of them can achieve satisfactory performance when dealing with completely unconstrained cursive words. Recently several approaches called "combination of multiple classifiers" have shown quite promising results in handwriting recognition[4,5,6].

In general, homogeneous classifiers are considered such as those having the same classification methodology but different feature vectors, i.e. multiple modular NNs (neural networks), and heterogeneous classifiers are considered such that each classifier uses a different classification methodology, i.e. NN and HMM. The homogeneous classifiers such as NNs can be implemented in three different ways: (1) a large scale single NN, (2) a *hybrid* of the multiple modular NNs at the output level[4], so called "loosely coupled", and (3) a *fusion* of multiple NNs at architectural level[4,5], so called "tightly coupled". The heterogeneous classifiers such as NN and HMM can also be implemented in two different ways: (1) a *hybrid* at the output level[4], and (2) a *fusion* at architectural level[6]. All the above efforts are directed to overcome the limitations of a single classifier.

In this paper, sophisticated hybrid schemes of the homogeneous and heterogeneous classifiers for cursive word recognition are presented. At first, two homogeneous MLPs are implemented and combined into a new single MLP classifier at the architectural level so called "fusion of two classifiers". Next, HMM is implemented and combined with the new MLP as a heterogeneous one. In the

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 433-442

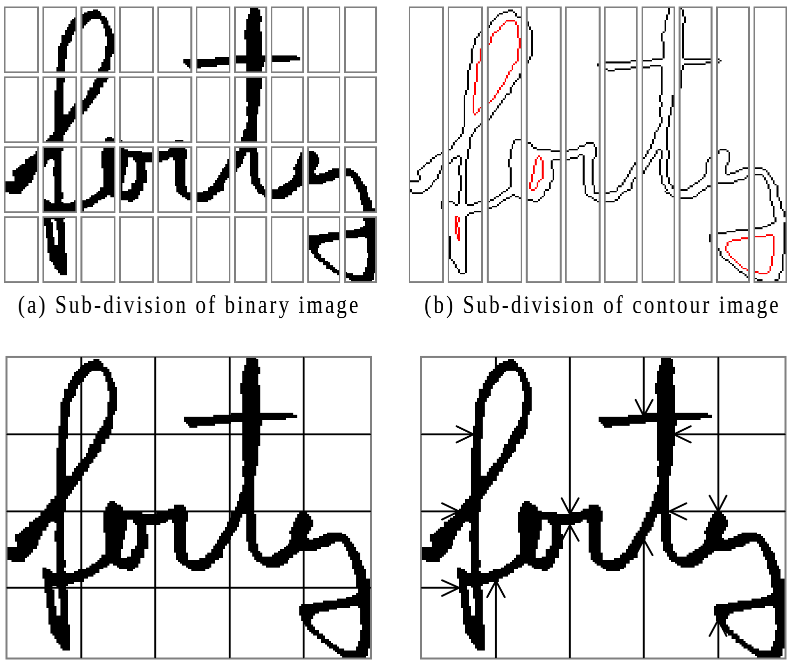
combining step, we introduced a new probability measure for the hybrid classifiers as well as conventional combining schemes. As a result of combination a new parameter for evaluating the ratio of improvement of the performance is also defined. To verify the feasibility of the presented methodology, experiments were conducted using the English legal word database of CENPARMI.

2 Implementation of MLPs, homogeneous classifiers

One of the most frequently used neural networks in the pattern recognition field is MLP. The performance of MLP is largely dependent on the input features as well as its topology.

2.1 Feature extraction for MLPs

The *five basic features* of MLPs for cursive word recognition are derived from binary image and contour image as shown in Fig. 1. They are defined as follows:



(a) Sub-division of binary image (b) Sub-division of contour image

(c) Crossing feature of binary image (d) Distance feature of binary image

Figure 1. Various feature extraction schemes for MLP input.

1. Mesh feature: The number of black pixels in each of the sub-divided local regions of the binary image as shown in Fig. 1 (a).
2. Chain feature: The number of same directional pixels in each of the sub-divide local regions of the contour image as shown in Fig. 1 (b). In general, four or eight directions can be considered.
3. Crossing feature: The number of times the strokes are crossed by projection lines which are defined as equally spaced horizontal and vertical lines as shown in Fig. 1 (c).
4. Distance feature: The distance from minimum boundary rectangle of the word to the first black pixel of the word image as shown in Fig. 1 (d).
5. Gradient feature[8]: The number of pixels that have the same gradient angle in each of sub-divided local region of the binary image as shown in Fig. 1 (a). In general, 4, 8 or 12 partitions of 360^o can be considered.

2.2 Implementation of MLPs

After extracting the above features, we can implement MLPs in three different ways, i.e. (1) implementation of a large single MLP, (2) implementation of multiple MLPs, and combination at output level, and (3) implementation of multiple MLPs and fusion at the architectural level.

In Fig. 2, three examples of MLP implementation for two input feature sets are shown. In the hybrid scheme as shown in Fig. 2(b), two separate MLPs are implemented by using each input feature set, and the outputs of each MLP are combined according to various combination schemes. Meanwhile, in the fusion scheme as shown in Fig. 2(c), two separate MLPs are also implemented by using each input feature set. The fusion is made such that a new MLP is implemented by using the outputs of the neurons in two hidden layers as new input features.

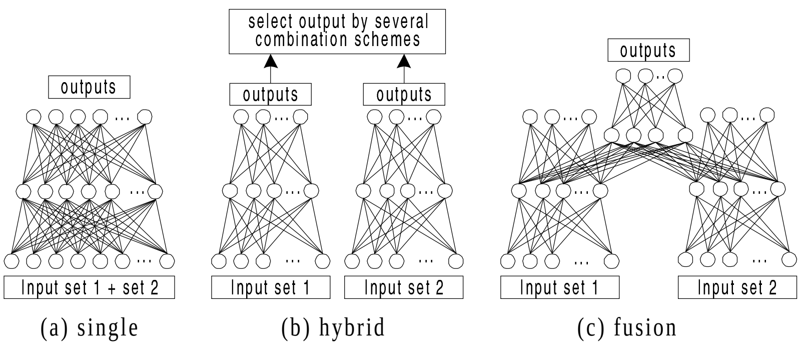


Figure 2. Three implementation schemes of MLPs for two input feature sets.

3 Implementation of HMM, heterogeneous classifier

Our strategy of HMM classifier design is that it should operate rather differently from MLP to increase the complementary capability of the hybrid classifier.

3.1. Pseudo temporal sequences

We design an explicit segmentation scheme for cursive words to obtain a pseudo temporal sequence, i.e. a feature vector for HMM. The stable and robust pseudo temporal sequences can be extracted from a series of graphemes of the segmented cursive word. Grapheme is a segmented part of a word, which looks like a character but sometimes not an actual character. The basic rules behind of the segmentation are that individual characters in a cursive word are connected by ligatures and most ligatures include a local down or up hill. However exceptional cases of the rules lead to over- or under-segmentations, so we have considered more criteria to obtain more precise segmentation results. Nevertheless due to the irregular shapes of cursive words, it is difficult to obtain perfect segmentation results. So even sophisticated segmentation algorithms can also produce over- or under-segmentations frequently. We will discuss the solutions of this problem further in the classifier implementation step.

3.2. Topology of HMM classifier

The segmentation process produces several cascaded graphemes that may include over- or under-segmentations of the cursive word. So we implement a segmentation based HMM classifier to represent these graphemes as shown in Fig. 3. This topology can solve the over- or under-segmentation problems. In Fig. 1, single state transition a_{ij} represents a single grapheme and three cascaded states represent a single character. During the state transitions, we can observe the output probability $b_{ij}(o_t)$ of the given observation symbol sequence $\mathbf{O}$. Each character has three states s_i and three different state transition paths to represent over-, exact- and under-segmentations.

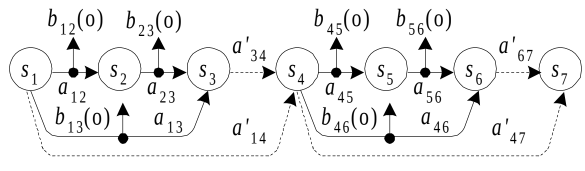


Figure 3. Segmentation based grapheme level hidden Markov model.

During the null transition a'_{ij}, the output symbol probability is not observed since a null transition represents "letter skip" due to under-segmentation. So on the way of state transition along the fixed single transition path, if the null transition is included, the observation time is not increased while the state is changed to the next one.

3.3. Training of HMM classifier

In the training step, the parameter re-estimation procedures are performed over the entire HMM λ_i of each word class *i* in such a way to maximize the posterior probability $P(\mathbf{O} \mid \lambda_i)$. However the calculation of posterior probability entails a very high computational complexity. So we can use efficient methods such as the well-known forward-backward procedures to calculate the probability. Detailed procedures of HMM parameter re-estimation are well presented in Rabiner's study[7]. Here, we attempt to describe only the calculation methods of the forward and backward variables for the presented HMM topology.

The forward variable $\alpha_t(i)$ denotes the probability of the partial observation sequence $O_1 O_2 ... O_t$ until time t, and the state s_i at time *t*, given hidden Markov model λ.

We can derive $\alpha_t(i)$ of the presented HMM classifier inductively as follows:

$$\alpha_t(j) = \sum_{i=1}^{N} [a_{ij} \prod_{p=1}^{P} \{b_{ij}^{p}(O_{t-1})\} \alpha_{t-1}(i) + a'_{ij}\ \alpha_t(i)] \tag{1}$$

where *p* denotes the number of observation symbol sequences. Similarly we can consider a backward variable $\beta_t(i)$ that is the probability of the partial observation sequence $O_{t+1} O_{t+2} O_{t+3} ... O_T$ from t+1 to the end, given the state s_i at time *t* and the HMM λ. We can calculate $\beta_t(i)$ in the same procedure of the forward variable as follows:

$$\beta_t(i) = \sum_{j=1}^{N} [a_{ij} \prod_{p=1}^{P} \{b_{ij}^{p}(O_t)\} \beta_{t+1}(j) + a'_{ij}\ \beta_t(i)] \tag{2}$$

In the calculation of the forward and backward variables, the state transitions are heading to the time incremental direction. However in the null transitions, there are no time increments. The forward and backward variables are calculated inductively according to the state transitions. We can calculate a posterior probability as well as re-estimated parameters according to the conventional ways[7] using the above forward and backward variables.

4 Combining homogeneous and heterogeneous classifiers

First we combined two homogeneous classifiers, MLPs, and the heterogeneous classifier, HMM, next. Before that, each output of HMM and MLP has been normalized to obtain equal level measurement values. Let $\Psi = \{1,2,3,...,V\}$ denote a set of word classes, O and F denote an observation symbol vector and a feature vector of the unknown pattern to be classified respectively.

A conditional probability $P(\mathbf{O} \mid \lambda_v)$ of each HMM λ_v given symbol vector O is normalized over all classes. Meanwhile each class output of MLP can be represented as a probability value by normalization of its output. Let $f(n_v, \mathbf{F}_i)$ denote an output of neuron n_v associated with class *v* given feature vector **F** of classifier *i*. A posterior probability $P(n_v \mid \mathbf{F}_i)$ of each neuron n_v given feature vector **F** can be obtained by normalization of output $f(n_v, \mathbf{F}_i)$ over all classes. Thereafter we can derive several combination rules based on two probabilities $P(\mathbf{O} \mid \lambda_v)$ and $P(n_v \mid \mathbf{F}_i)$ of each class *v* to calculate a new class probability $P(C_v \mid \mathbf{O}, \mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M)$ of the hybrid classifier *C*, where M denotes the number of homogeneous classifiers, MLPs.

Three combination schemes have been considered such as conventional voting, LCA[4] (linear confidence accumulation) introduced at CENPARMI and new multiplication methods.

4.1 Maximum Sorting

Given the probabilities $P(\mathbf{O} \mid \lambda_v)$ and $P(n_v \mid \mathbf{F}_i)$ of HMM and MLPs respectively, the maximum among the probabilities can be derived as a new probability of the hybrid classifier as follows:

$$P(C_v \mid \mathbf{O}, \mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M) = \max[P(\mathbf{O} \mid \lambda_v), \max_i P(n_v \mid \mathbf{F}_i)] \quad (3)$$

Given word classes $\Psi = \{1,2,3,...,V\}$, the final decision is made such as

$$v^* = \arg\max_{v \in \Psi} P_H(C_v \mid \mathbf{O}, \mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M) \quad (4)$$

4.2 Linear Confidence Accumulation

According to the LCA method[4], a new probability of each class is derived with weighting coefficient, namely.

$$P(C_v \mid \mathbf{O}, \mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M) = w_0 \times P(\mathbf{O} \mid \lambda_v) + \sum_{i=1}^{M} w_i \times P(n_v \mid \mathbf{F}_i) \quad (5)$$

The weighting factors w_i can be defined according to the performance of each classifier. The final decision is made in the same way as shown in equation (4).

4.3 Weighted Multiplication

We defined a new probability of each class based on the multiplication of the probabilities with weighting factors such as

$$P(C_v | \mathbf{O}, \mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_M) = P(\mathbf{O} | \lambda_v)^{w_0} \times \prod_{i=1}^{M} P(n_v | \mathbf{F}_i)^{w_i} \qquad (6)$$

The weighting factors w_i can be defined in the same manner as the LCA method. The final decision of this method can also be obtained using equation (4).

5 Experimental results

To verify the presented approach, experiments were carried out on the legal word databases of CENPARMI. The database has been created from 2,500 handwritten checks written in English. The number of writers is estimated to be close to 800. We trained 5,223 words and tested 2,482 words for 32-class legal word recognition. Some example legal word images used in these experiments are shown in Fig. 4.

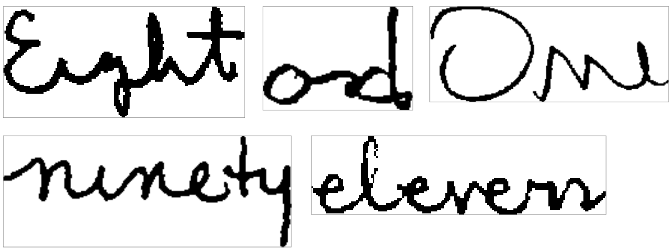


Figure 4. Segmented and tagged legal word samples used in experiments.

In the first experiment, homogeneous classifiers are implemented as shown in Fig. 2. For feature set 1, 120 features are extracted such as 48 mesh features (12 horizontal and four vertical divisions), 40 chain features (10 horizontal divisions and four directions per each division), 17 crossing features and 15 distance features. On the other hand, for feature set 2, 120 gradient features are extracted (10 horizontal and three vertical divisions, and 4 gradient bins per each division). Table 1 shows the simulation results of various implementation methods

Table 1. The performances of MLPs for legal word recognition

Classifier types		Recognition	Improvement
MLP A (120 - 60-32)		85.1%	-
MLP B (120 - 60-32)		83.5%	-
MLP C (240-120-32): Large scale		86.0%	-
Hybrid of MLP A and MLP B	(1) Max	86.2%	1.2%
	(2) LCA	86.5%	1.6%
	(3) Multiply	87.1%	2.3%
Fusion of MLP A and MLP B (120-80-32)		87.9%	3.3%

In Table 1, three concatenated numbers, for instance 120-60-32, denote the number of neurons in input, hidden and output layers of MLP respectively. The "Improvement" is the ratio of improvement of the performance as a result of combination and defined as

$$Improvement = \frac{P_C - \max_i P_i}{\max_i P_i} \times 100 \tag{7}$$

where P_C denotes the recognition rate of combined classifiers and P_i denotes the recognition rate of classifier i before combinations. Though the "weighted multiplication" method produced better results than the other two hybrid methods, the fusion of two classifiers shows the best performance among them with 87.9% recognition rate and 3.3% improvement rate where 3.3% is derived from $P_C = 87.9$ and $\max_i P_i = 85.1$.

In the next experiment, HMM classifier is implemented and combined with MLPs as a heterogeneous classifier. The performance is shown in Table 2.

Table 2. The performances of MLP and HMM combinations for legal word recognition

Classifier types		Recognition	Improvements
HMMs		82.0%	-
Hybrid HMM and MLP C by Multiply		90.1%	4.8%
Hybrid of MLP A, MLP B and HMM	(1) Max	88.3%	2.4%
	(2) LCA	89.6%	3.6%
	(3) Multiply	92.2%	5.9%
Hybrid an HMM and a fusion MLP by Multiply		92.7%	5.4%

In Table 2, although the performance of HMM classifier is lower than two modular classifier MLP A and MLP B, the combination results with HMM and MLPs are quite encouraging since the performance improvement rates are better than the previous simulation. These results lead us to conclude that the combination of heterogeneous classifiers gives better complementary ability than the combination of homogeneous classifiers. Especially, the hybrid combination of an HMM and a fusion classifier of two MLPs show 92.7% recognition rates and 5.4% performance improvement rate. The value 5.4% is derived from $P_C = 92.7$ and $\max_i P_i = 87.9$.

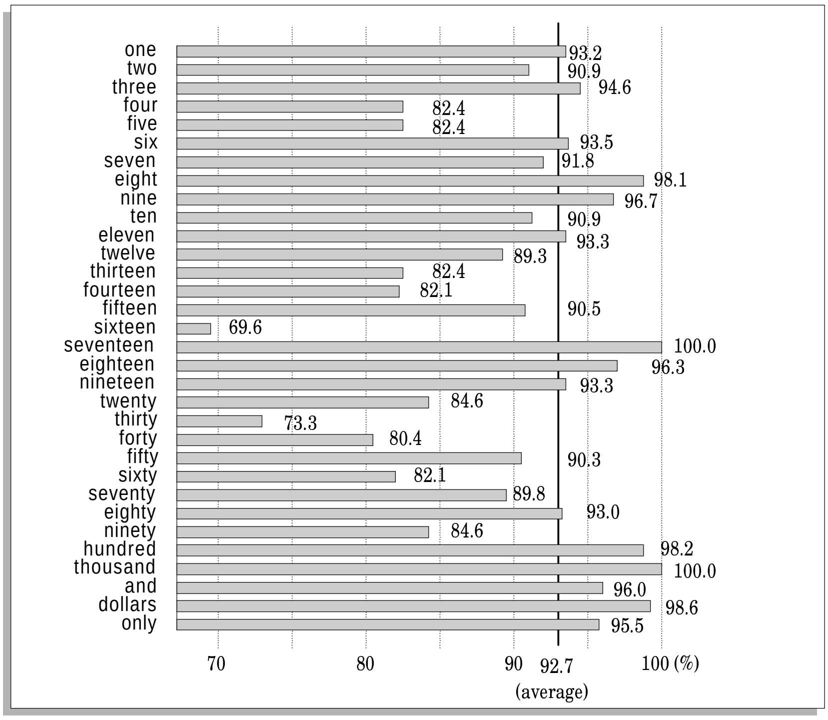


Figure 5. Recognition performance of each class.

Recognition ratio of each legal word class is shown in Fig. 5. The distinct words such as "hundreds", 'thousand", "dollars" and "only" have higher performance since these words did not confuse with others.

6 Conclusion

We have shown the efficiency of the presented combining schemes of HMM and MLP for cursive word recognition. To maximize the complementary capability, an HMM has been implemented as a heterogeneous classifier against two homogeneous classifiers of MLPs. The "weighted multiplication" combining method produced a higher performance than other methods, and a combination of heterogeneous classifiers revealed better complementary ability than a combination of homogeneous classifiers.

We estimate that the recognition rate of 92.7% for 32 cursive word classes with unconstrained database is probably the highest performance among those ever reported in the literature on English legal word recognition.

7 Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada, the FCAR program of the Ministry of Education of Quebec. The authors also like to thank Christine Nadal and Guiling Guo for preparing the CENPARMI database for the experiment.

References

1. A. Yacoubi, M. Gilloux, R. Sabourin, and C.Y. Suen, "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition," *IEEE Trans. on PAMI*, vol. 21, no. 8, pp. 752-760, 1999.
2. C.Y. Suen, L. Lam, D. Guillevic, N. Strathy et. al., "Bank Check Processing System," *International Journal of Imaging Systems and Technology*, vol. 7, pp. 392-403, 1996.
3. D. Guillevic and C.Y. Suen, "Cursive Script Recognition Applied to the Processing of Bank Cheques," *Proceedings of the International Conference on Document Analysis and Recognition*, Montreal, Canada, pp. 11-14, 1995.
4. Y.S. Huang and C.Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. on PAMI*, vol. 17, no. 1, pp. 90-94, 1995.
5. K.T. Lim and S.I. Chien, "Neural Network Based Feature Space Generation for Multiple Database of Handwritten Numerals," *Proceedings of the International Conference on Pattern Recognition*, Brisbane, Australia, pp. 375-377, 1998.
6. S.B. Cho, "Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals," *IEEE Trans. on Neural Networks*, vol. 8, no.1, pp. 43-53, 1997.
7. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceeding of the IEEE*, vol. 77, no. 2, pp. 257-285, 1989.
8. G. Srikantan, S.W. Lam, and S.N. Srihari, "Gradient-based Contour Encoding for Character Recognition," *Pattern Recognition*, vol. 29, no. 7, pp. 1147-1160, 1996.