Fusion of n-Tuple Based Classifiers for High Performance Handwritten Character Recognition

Konstantinos Sirlantzis[1], Sanaul Hoque[1],
Michael C. Fairhurst[1], and Ahmad Fuad Rezaur Rahman[2]

[1] Department of Electronics, University of Kent
Canterbury, Kent, United Kingdom
{ks30,msh4,mcf}@ukc.ac.uk
[2] BCL Technologies Inc.,
990 Linden Drive, Suite #203, Santa Clara, CA 95050, USA
fuad@bcltechnologies.com

Abstract. In this paper we propose a novel system for handwritten character recognition which exploits the representational power of n-tuple based classifiers while addressing successfully the issues of extensive memory size requirements usually associated with them. To achieve this we develop a scheme based on the ideas of multiple classifier fusion in which the constituent classifiers are simplified versions of the highly successful scanning n-tuple classifier. In order to explore the behaviour and statistical properties of our architecture we perform a series of cross-validation experiments drawn from the field of handwritten character recognition. The paper concludes with a number of comparisons with results on the same data set achieved by a diverse set of classifiers. Our findings clearly demonstrate the significant gains that can be obtained, simultaneously in performance and memory space reduction, by the proposed system.

1 Introduction

Handwritten character recognition is still one of the most challenging problems in pattern classification. Over the years a great number of algorithms have been developed to achieve improved performance. Some of the simplest yet most successful among them are the so-called 'n-tuple' based classifiers. Unfortunately, there is usually a trade-off between high performance and either increased computational load or increased memory requirements. The Scanning n-tuple classifier (SNT) [8] is a typical example of the case where superior recognition rates are attained at the expense of significant storage requirements, especially in applications with samples of realistic size. On the other hand, in recent years there has been a significant shift of interest from the development of powerful but demanding individual classifiers to the development of strategies to fuse the outcomes of a number of relatively simpler classifiers [7]. Such structures have the

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 770–779, 2002.

inherent ability to exploit the diverse recognition mechanisms of the participating classifiers resulting, in most cases, in a scheme demonstrably more successful than the best of its constituent members [12]. In this paper we propose a novel system for handwritten character recognition which exploits the representational power of n-tuple based classifiers while addressing successfully the issues of extensive storage requirements. This is achieved by developing a scheme based on the ideas of multiple classifier fusion in which the constituent classifiers are simplified versions of the highly efficient Scanning n-tuple classifier. Although the individual performances of these classifiers can be comparatively inferior, our findings demonstrate the significant gains that can be obtained, simultaneously in performance and storage space reduction, by the proposed system. In the next sections we initially give a brief description of the components of the system we propose followed by the derivation of the multiple classifier combination rule we adopted which also provides a justification for this choice. To demonstrate the statistical properties of our scheme, we present results obtained over a series of cross-validation experiments drawn from the field of handwritten character recognition. The paper concludes with a number of comparisons with results on the same data set achieved by a diverse set of classifiers reported in the literature, and discussion of our findings.

2 The Proposed Scheme

We start the presentation of the components of our system with the Frequency Weighted Scheme (FWS) which is an n-tuple based classifier reported to demonstrate reasonable levels of performance while requiring comparatively low memory size. We then continue with a description of the original Scanning n-tuple scheme (SNT). Subsequently, the *Bit-Plane Decomposition* technique is described, which is the method we employed to simplify the original feature space used by the SNT and thus reduce the size of the storage space needed by the classifiers constructed on it. Figure 1 gives a schematic representation of the system illustrating the information flow in the parallel combination architecture through the components of the scheme we call 'Layered Sampling of the Bit-Plane Decomposition'.

2.1 Frequency Weighted Scheme (FWS)

In a conventinal n-tuple classifier, the n-tuples are formed by selecting multiple sets of n distinct locations from the pattern space. Each n-tuple thus sees an n-bit feature derived from the pattern. For classification, a pattern is assigned to that class for which the number of matching features found in the training set is maximum. The training process, therefore, requires remembering the occurrences of different features as seen by individual n-tuples. This is usually achieved by setting a 'flag' bit in a large memory array [3].

The FWS is the simplest enhancement of the basic n-tuple classification system. In the basic scheme, both the common and rare feature occurrences are

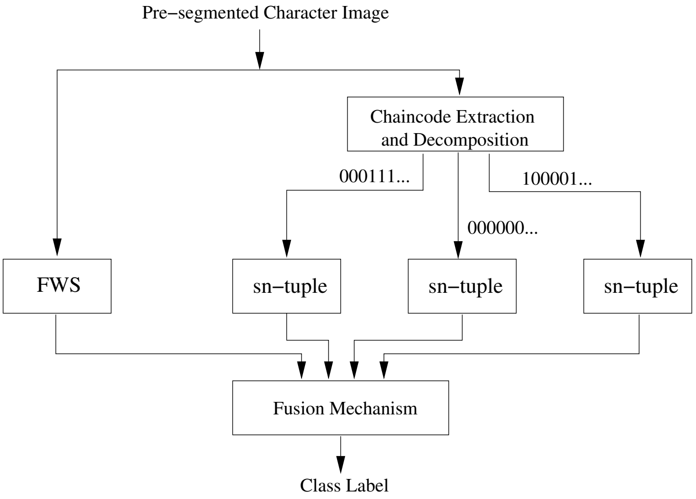

Fig. 1. Schematic of a classification based on Layered Sampling of the Bit-Plane Decomposition

accorded the same discriminatory weight. Thus, the presence of even one rogue pattern in the training set of a class can reduce the discriminatory power of the n-tuple network significantly. As a remedy, in the FWS, instead of setting the flag to record the occurrence of a certain feature in the training set, the relative frequencies are recorded. The frequency counts need to be normalized when different classes have different numbers of training images. The sum of these frequencies corresponding to a particular test image determine its class label.

2.2 Scanning n-Tuple Classifier (SNT)

The Scanning n-tuple (or simply sn-tuple) classifier [8] has been introduced as a statistical-cum-syntactic method for high performance character recognition applications. This is also a variant of the n-tuple classifier except that instead of using the two dimensional raw images directly, the operation is conducted on a one dimensional gray scale representation of the bitmap image. Another difference between the n-tuple and the sn-tuple is that, whereas each n-tuple samples a set of fixed points in the input space, each sn-tuple defines a set of relative offsets between its input points. Each sn-tuple is then scanned over the entire input space.

The one dimensional representation of the binary pattern image is obtained by tracing the contour edges of the image and representing the path by Freeman chain codes [4]. The sn-tuple algorithm is designed to model only one chain code string per pattern and a difficulty arises for images consisting of more than one contour. This is dealt with by mapping a set of strings to a single string

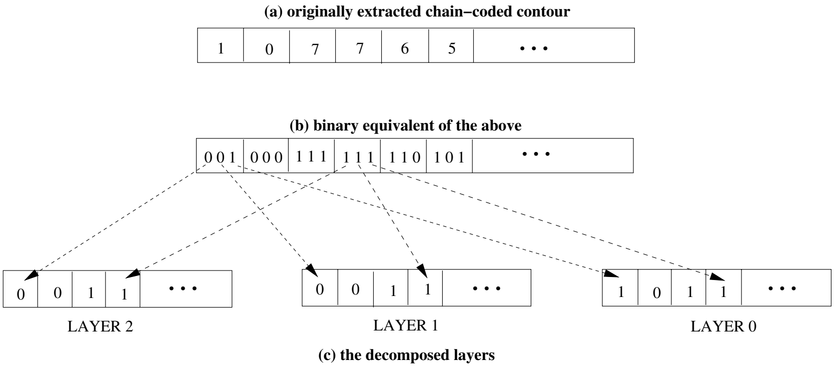
(a) originally extracted chain–coded contour
1
0
7
7
6
5
• • •
(b) binary equivalent of the above
001
000
111
111
110
101
• • •
0
0
1
1
• • •
LAYER 2
0
0
1
1
• • •
LAYER 1
1
0
1
1
• • •
LAYER 0
(c) the decomposed layers

Fig. 2. The Proposed Bit Plane Decomposition Technique

by discarding the positional information (i.e., the start coordinates) and then concatenating the strings together. Besides, the length of a chain coded string is dependent on the character class as well as writing style, degree of slant, image quality etc. Since image classes with short chain codes may be adversely affected, all chains are expanded to a predefined fixed length before training and testing.

2.3 Bit Plane Decomposition

The size of the memory space required by a typical n-tuple based scheme is σ^n units per tuple per class, where σ is the number of distinct values a pixel may have levels and n is the size of the tuples. It can readily be seen that this can become excessively large even with a fairly small number of gray levels. Bit-Plane Decomposition was initially introduced by Schwarz [11] as a means of data compression. However, it can also be used to handle the memory space problem faced by n-tuple based systems [5]. The basic idea is to decompose an image into a collection of binary images ($\sigma = 2$).

For Bit-Plane Decomposition, the individual chain codes extracted from the image are represented in binary. Each is represented by a 3 bit binary code. The chain-coded string is decomposed, subsequently, into 3 layers where layer 'i' is composed of the ith bits of the binary code values. Thus, for example, Layer '0' is formed by collecting all the *least significant bits* of the binary coded string. The decomposition we used here can be seen in Figure 2. The sn-tuple classifiers constructed on each one of the 3 layers extracted from the original chain codes are indicated by SNTL0, SNTL1, and SNTL2 in our Tables (with SNTL0 indicating the one trained on the least significant bit layer).

2.4 Multiple Classifier Decision Fusion

In the present work the four individually trained classifiers used are arranged in a parallel structure to form a multi-classifier recognition system. The choice of this simple architecture to form the ensemble was preferred so that our experimental results could be more easily studied and interpreted.

Although a variety of fusion rules have been devised by researchers [7], the choice of the most appropriate scheme is usually dependent on the degree of diversity of the feature spaces on which the participant classifiers are trained, and the nature of their outputs.

To better understand this, let us consider that a pattern q is to be assigned to one of the m possible classes $\{\omega_1, \ldots, \omega_m\}$ and there are K independent classifiers, in each q is represented by a distinct feature vector $\mathbf{x}_i$, $i = 1, \ldots, K$, each drawn from a corresponding feature space χ_i, $i = 1, \ldots, K$. Let each class ω_k be modelled by the probability density function $P(\mathbf{x}_i|\omega_k)$. Following a Bayesian perspective, each classifier is considered to provide an estimate of the true class posterior probability $P(\omega_j|\mathbf{x}_i)$ given $\mathbf{x}_i$. The idea underlying multiple classifier fusion it to obtain a better estimator by combining the resulting individual estimates [7]. The pattern q should be assigned, consequently, to the class having the highest *posterior* probability. Assuming equal *a priori* probabilities for all the classes, the corresponding decision rule is:

$$\text{assign} \quad \theta \rightarrow \omega_j \quad \text{if}$$

$$P(\omega_j|\mathbf{x}_1, \ldots, \mathbf{x}_K) = \max_{k=1}^{m} P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_K),$$

where θ is the class label of the pattern under consideration q.

Following this line of reasoning in the case where the individual classifiers are sampling identical feature spaces (i.e. $\chi_1 = \ldots = \chi_K$), averaging the estimates will suppress the estimation error as well as the effects of individual classifier overtraining (bias), subsequently reducing the classification error [7]. This gives rise to the well-known 'sum' or 'mean' rule. Alternatively, if the product of the individual estimates is used in this case it will result in an amplification of the estimation noise. In contrast, the latter combination method will obtain maximal gains from independent pattern representations (i.e. if the classifiers sample independent feature spaces χ_i [1]).

From the preceding discussion it becomes clear that the averaging process is the most beneficial choice of fusion scheme here, since the sn-tuple based classifiers sample the Bit-Plane Decomposition of the same original chain codes. The corresponding 'mean' combination rule we used can be formally expressed as follows:

$$\text{assign} \quad \theta \rightarrow \omega_j \quad \text{if}$$

$$K^{-1} \sum_{i=1}^{K} P(\omega_j|\mathbf{x}_i) = \max_{k=1}^{m} \left[K^{-1} \sum_{i=1}^{K} P(\omega_k|\mathbf{x}_i) \right].$$

3 Experimental Results

To observe the behaviour of our system and its statistical properties we employed an inhouse database which consists of 34 classes of pre-segmented characters (numerals 0-9, and upper case letters A-Z, without differentiation between the pairs

Table 1. Mean Error Rates (%) of the components of the proposed system

	Digits (10 classes)	Digits & Letters (34 classes)
FWS	10.00	22.28
SNTL0	23.89	44.92
SNTL1	21.28	40.67
SNTL2	22.45	42.08
SNTL0 + SNTL1 + SNTL2	11.41	23.69

0/O and 1/I). The database corresponds to handwritten characters, every class has 300 samples (10200 characters in total), and the images are provided at a resolution of 16 × 24 pixels. Two recognition problems were constructed. The first included only the numerals (a 10-class task), while the second consisted of all the 34 classes (numerals and letters). For each problem randomly defined disjoint partitions in training and test sets are used to produce the cross-validation estimates of the statistics reported here. The training sets contain 150 samples per class (1500 characters in the digits case and 5100 in the alphanumeric case). The test set size is fixed at 75 samples per class.

We first examine the individual performances of the participants of the multiple classifier system. The upper part of Table 1 shows the recognition error rates of the FWS as well as the sn-tuple classifiers trained on the 3 layers of the Bit-Plane Decomposition for the two task domains. It becomes readily apparent that the performance of the FWS classifier is superior compared to the others. In fact, in the best case the 3 sn-tuple classifiers present more than double the error rate of FWS. Considering the fact that the original sn-tuple algorithm results in error rates of 4.59% and 12.42% for the 10 and 34-class problem respectively, we may safely conclude that significant discriminatory information has been lost by the decomposition. However, observing the significant improvement showed in the lower part of Table 1 representing the 'mean' rule combination of the classifiers trained on the decomposed layers, leads us to hypothesise that they encapsulate information significantly complementary to each other. It should be noted here that despite their relatively poor performance the are very efficient with respect to memory space (they require only memory sizes of the order of 2^n instead of 8^n of the original SNT, n being the tuple size used).

In Table 2 we present the classification error rates achieved by combining the FWS with one or more of the layer-based sn-tuple classifiers. It is easy to observe that additional gains are obtained by these multiple classifier systems, since the highest reduction in error rates achievable, in comparison to the best of the constituent members of the system (i.e. the FWS), are of the order of 75% for the 10-class problem and of the order of 65% for the 34-class task.

Figures 3 and 4 provide a comparative illustration of the achievable gains in recognition accuracy with respect to a diverse set of classification schemes tested on the same database. The figures plot mean values of the performance statistics obtained in our cross-validation experiments. In addition to identifying the best

Table 2. Mean Error Rates (%) of the proposed multiple classifier systems

	Digits (10 classes)	Digits & Letters (34 classes)
FWS + SNTL0	6.59	16.81
FWS + SNTL1	4.51	14.74
FWS + SNTL2	5.63	14.10
FWS + SNTL0 + SNTL1	3.23	11.51
FWS + SNTL0 + SNTL2	3.95	11.34
FWS + SNTL1 + SNTL2	2.40	10.23

performing of the proposed architectures [indicated as Best of Proposed (BoP)], error rates for five other classifiers are shown. The FWS and the conventional sn-tuple scheme (SNT) have been described previously. The remaining classifiers included in the Figures are briefly described below:

Moment-based Pattern Classifier(MPC): This is a statistical classifier which explores the possible cluster formation with respect to a distance measure. This particular implementation used Mahalanobis distance calculated on the nth order mathematical moments derived from the binary image [9].

Multilayer Perceptron (MLP): This is the well-known Artificial Neural Network architecture with 40 hidden units trained using Backpropagation Learning [10].

Moving Window Classifier (MWC): This is again an $n-$tuple based scheme which utilizes the idea of a window scanning the binarised image to provide partial classification indices which are finally combined to obtain the overall classification decision [2, 6]. A 21×13 pixel window with 12-tuples were used for this particular implementation.

Finally, it is worth noting that the schemes proposed in this paper perform favorably even in comparison to a multiple classifier system (denoted by GA in the Figures) optimised by a genetic algorithm, introduced in [12]. The corresponding error rates for the same database achieved by the genetically designed multi-classifier system were 3.40% for the 10-class, and 8.62% for the 34-class tasks.

4 Conclusions

In this paper we have introduced a novel system for high performance handwritten character recognition based on n-tuple classifiers. Our primary idea was to exploit the superior performance characteristics of the Scanning n-tuple (SNT) classifier, while at the same time reducing its excessive requirements for memory space. To this end we proposed the development of a scheme based on multiple classifier systems, which have been shown to achieve increased performance by fusing relatively weaker classifiers. The participants in this system are chosen, then, from a class of appropriately simplified versions of the original SNT algorithm, which have significantly reduced memory size requirements at the expense of considerably higher error rates. A series of cross-validation experiments on a

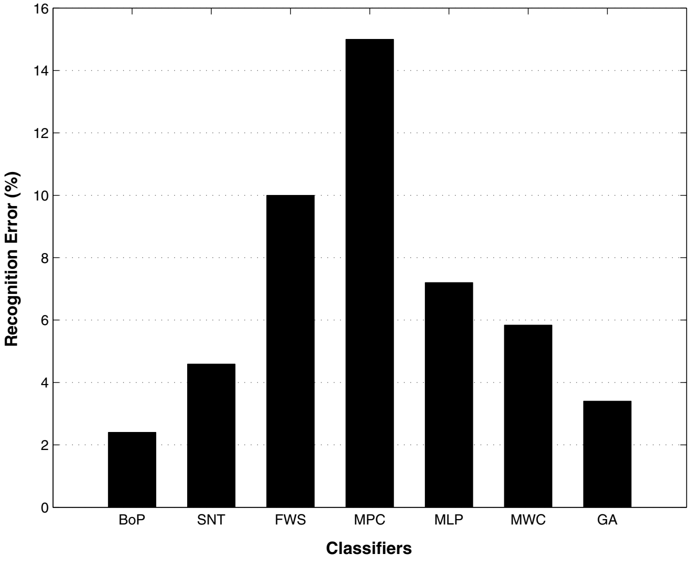

Fig. 3. Comparison of the Error Rates (%) between the Best of the Proposed Schemes (BoP) and Other Classifiers for the 10-class problem

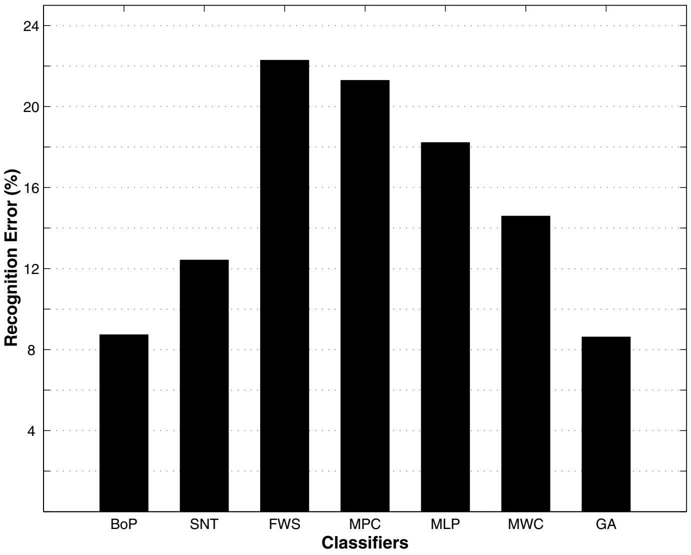

Fig. 4. Comparison of the Error Rates (%) between the Best of the Proposed Schemes (BoP) and Other Classifiers for the 34-class problem

10-class and a 34-class problems from the area of handwritten character recognition serve, finally, to demonstrate the statistical properties of our proposals. Our findings show that although the constituent members of the fusion scheme present poor performance, our system outperforms even established algorithms well-known for their efficiency in the task at hand.

Acknowledgement

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council.

References

[1] F. M. Alkoot and J. Kittler. Improving product by moderating k-nn classifiers. In J. Kittler and F. Roli, editors, *Second International Workshop on Multiple Classifier Systems*, pages 429–439. Springer, 2001. 774

[2] M. C. Fairhurst and M. S. Hoque. Moving window classifier: Approach to off-line image recognition. *Electronics Letters*, 36(7):628–630, March 2000. 776

[3] M. C. Fairhurst and T. J. Stonham. A classification system for alpha-numeric characters based on learning network techniques. *Digital Processes*, 2:321–329, 1976. 771

[4] H. Freeman. Computer processing of line-drawing images. *ACM Computing Surveys*, 6(1):57–98, March 1974. 772

[5] M. S. Hoque and M. C. Fairhurst. Face recognition using the moving window classifier. In *Proceedings of 11th British Machine Vision Conference (BMVC2000)*, volume 1, pages 312–321, Bristol, UK, September 2000. 773

[6] M. S. Hoque and M. C. Fairhurst. A moving window classifier for off-line character recognition. In *Proceedings of 7th International Workshop on Frontiers in Handwriting Recognition*, pages 595–600, Amsterdam, The Netherlands, September 2000. 776

[7] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. 770, 774

[8] S. Lucas and A. Amiri. Recognition of chain-coded handwritten character images with scanning n-tuple method. *Electronic Letters*, 31(24):2088–2089, November 1995. 770, 772

[9] A. F. R. Rahman and M. C. Fairhurst. Machine-printed character recognition revisited: Re-application of recent advances in handwritten character recognition research. *Special Issue on Document Image Processing and Multimedia Environments, Image & Vision Computing*, 16(12-13):819–842, 1998. 776

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation, in Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986. D. E. Rumelhart and J. L. McClelland(Eds.). 776

[11] J. W. Schwarz and R. C. Barker. Bit-plane encoding: A technique for source encoding. *IEEE Transaction on Aerospace and Electronic Systems*, 2(4):385–392, 1966. 773

[12] K. Sirlantzis, M. C. Fairhurst, and M. S. Hoque. *Genetic Algorithms for Multiple Classifier System Configuration: A Case Study in Character Recognition*, volume 2096 of *LNCS*, pages 99–108. Springer, 2001. 771, 776