Proceedings of the Fourth IEEE International Conference on Electronics, Circuits and Systems,
Cairo, Egypt, December 15-18 1997, Volume 2, pp 850-853.

A HYBRID NEURAL NETWORK/RULE-BASED TECHNIQUE FOR ON-LINE GESTURE AND HAND-WRITTEN CHARACTER RECOGNITION

*M.P.Craven, K.M. Curtis, B.R.Hayes-Gill and C.D. Thursfield**
University of Nottingham, Department of Electrical and Electronic Engineering,
University Park, Nottingham, NG7 2RD, UK.
Tel: +44 115 9515151 x2060 Fax: +44 115 9515616 E-mail: mpc@eee.nott.ac.uk
*Access to Communication and Technology, Regional Rehabilitation Centre,
91 Oak Tree Lane, Selly Oak, Birmingham B29 6JA, UK.

ABSTRACT

A technique is presented which combines rule-based and neural network pattern recognition methods in an integrated system in order to perform learning and recognition of hand-written characters and gestures in real-time. The *GesRec* system is introduced which provides a framework for data acquisition, training, recognition, and gesture-to-speech transcription in a Windows environment. A recognition accuracy of 92.5% was obtained for the hybrid system, compared to 89.6% for the neural network only and 82.7% for rules only. Training and recognition times are given for an able-bodied and a disabled user.

I. INTRODUCTION

As part of an investigation into multi-input communication aids for people with combined motor and speech disability, the Rehabilitation Engineering group at the University of Nottingham is designing software to capture 3D gesture information and to provide a framework for recognition using a wide variety of algorithms. Gesture information is provided in real-time by a Polhemus 3SPACE FASTRAK six-degrees-of-freedom (6DOF) electro-magnetic tracking system[1]. One method of processing this information, which consists of time samples of three spatial co-ordinates and three orientation angles, is to project it on to a plane of spatial co-ordinates so that recognition can be carried out in 2D in analogy with on-line isolated character recognition. Alternatively, 2D samples can be obtained directly from a graphics tablet or from a mouse.

There are many proposed methods for on-line hand-written character recognition which use a wide variety of pattern recognition techniques[2]. In addition, neural networks have been proposed as good candidates for character recognition[3,4]. Studies have also been carried out for gesture recognition, comparing techniques such as dynamic programming, Hidden Markov Models and recurrent neural networks[5].

Recently, hybrid systems for pattern recognition which combine neural networks with rule-based methods have been found to be very useful in the areas of speech and natural language processing, and other applications[6,7]. For classification problems the neural network is able to give statistical information about the classification and is easy to train, but it is often not clear how the neural network has arrived at its answer. On the other hand, the operation of the rule-base is traceable, but the set of rules chosen may be more difficult to train and may not generalise as well as a neural network. Hybrid systems combine the advantages of the two methods and can be used as an alternative to fuzzy logic.

II. METHOD

A generalised *D*-dimensional gesture or character can be described as a sequence of *m* co-ordinates,

$$G = g_1 g_2 \cdots g_m \tag{1}$$

where g = g(X(0), ..,X(D)).

Projecting to 2D this becomes:

$$G' = g'_1 g'_2 \cdots g'_m \tag{2}$$

where g’= g’(X,Y).

The timings of the samples are not required here as we are using the shape of the gesture rather than its dynamics. A set *C* of *N* classes is defined so that the gestures may be mapped to different actions. In the case of character recognition, these classes typically correspond to letters of the alphabet or numerals. Classes can also correspond to a limited vocabulary of words or phrases, or alternatively to command strings to be used for environmental control.

In the proposed hybrid method, the raw co-ordinate 2D projection is first quantised on to a 10-by-10 grid, which is in turn divided into 9 zones, as shown in Figure 1. This is

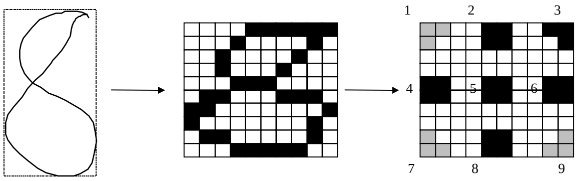


Figure 1 Transformation of a 2D gesture or character to a 10-by-10 grid, then to 9 zones

achieved by a size invariant transformation of the bounding rectangle of the drawn gesture, which is mapped on to the grid. In order to make sure that grid squares are not missed if samples are widely spaced, a pre-processing step is employed to linearly interpolate between samples. Quantisation is then is carried out by detecting which blocks in the grid are crossed by the interpolated gesture, resulting in a set B of 100 binary elements, which have a value of 1 if the grid block is crossed, 0 otherwise. The height and width of the rectangle and the start and end co-ordinates of the gesture are also stored. A second level of quantisation is then carried out to detect which of the nine zones are occupied by the gesture.

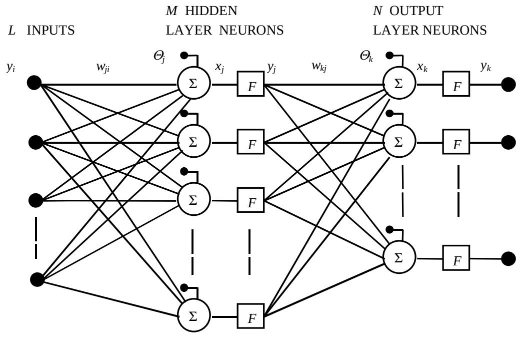


Figure 2 *L-M-N* multi-layer perceptron neural network architecture

The binary data from the 10-by-10 grid is used as input to a *L-M-N* Multi-Layer Perceptron (MLP) neural network classifier with sigmoidal outputs using the logistic function $F(x)=1/(1+e^{-x})$ as shown in Figure 2, where L=100, M is the number of hidden neurons forming a set of feature detectors for the classification problem, and N is the number of classes as above. The network is trained using the backpropagation algorithm[8] so that the output gives a ‘1’ for the selected class corresponding to the input map, and ‘0’ for the other classes. However, the outputs are continuous in the range [0,1], so in practice the class with the largest output activation is chosen and assigned an integer score $s_c = s$, whereas the other classes are given a score $s_c = 0$. The value of the largest output is tested against an acceptance threshold so that no class is selected if all outputs are below this value. This is achieved because, if the input pattern is ambiguous, the neural network output activation is split between the most likely classes e.g. two outputs of 0.4 rather than one of 0.8. In this case a value of 0.6 would provide a good acceptance threshold. This demonstrates one of the advantages of using the hybrid system, as it is more difficult to define an acceptance criteria for logical rules.

Further to this, a set R is constructed consisting of rules corresponding to r extracted binary features b_r, where b_r=+1 if the feature is present and b_r=-1 if the feature is not present. The rules consist of 3 main categories; height-to-width ratio (HWR) of the drawn character (e.g. HWR>1.5), start-to-end positions of the drawn character e.g. from left to right, and zone rules e.g. drawn character crosses Zone 1. In our current implementation there are 4 HWR rules, 4 start-to-end rules and 9 zone rules i.e. a 17-dimensional feature vector.

A training set T consisting of several examples of each class is constructed, such that $T_c(B, R)_i$ is the ith example of class c, comprising the neural network inputs and the rule-based features. The binary rule-based features are found for each example and these are then combined to give rule weightings w_r. The weights are specified using a 3-state logic; w_r=+1 meaning the feature r must be present, w_r=-1 the feature r must not be present, and w_r= 0 the feature r may or may not be present. This 3-state logic makes the rules trainable. The weights are assigned values as follows: w_r=+1 if the feature is found in over 60% of the examples, w_r=-1 if the feature is found in less than 40% of the examples, otherwise w_r=0. By combining the features in this way, only one feature vector is required for each class, which simplifies and speeds up recognition.

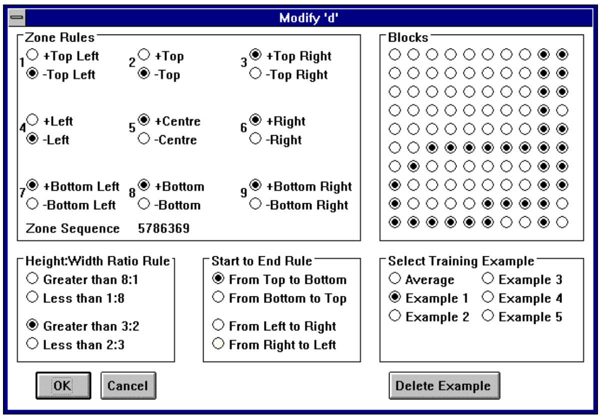
In order to carry out hybrid recognition, a total score S_c is calculated for each class by adding the products of the rule weightings and the binary features b_r extracted from the test example, which is then added to the neural network score s_c for that class,

$$S_c = s_c + \sum_r w_r b_r \qquad (3)$$

The class with the highest total score is picked as the one most closely representing the drawn character or gesture. Note that the highest scores are obtained for a class when features that should be present are present, and vice-versa, and when the neural network also selects that class. The score s for the class chosen by the neural network is determined heuristically to give a good balance with the scoring of the rule-base. The neural network acceptance threshold is also determined heuristically. Both parameters can be adjusted after training to fine tune the recognition.

III. EXPERIMENT

Software *GesRec* was designed to perform data acquisition from either the Polhemus tracker, the graphics tablet, or mouse in the form of a 2D array of co-ordinates, and prompt the user for several examples of each character or gesture in an automatic training session. To provide an output for the recognition, segmented characters or gestures were linked to text to be spoken by a text-to-speech synthesis system. The software was coded in the C programming language to implement the hybrid system in a Microsoft Windows environment, and was executed on an IBM-compatible PC with an Intel 486DX 66MHz processor and Creative Sound Blaster 16 sound card. The recognition software was linked via a Dynamic Link Library to the Creative TextAssist text-to-speech system.



Zone Rules
1 ○ +Top Left ● -Top Left
2 ○ +Top ● -Top
3 ● +Top Right ○ -Top Right
4 ○ +Left ● -Left
5 ● +Centre ○ -Centre
6 ● +Right ○ -Right
7 ● +Bottom Left ○ -Bottom Left
8 ● +Bottom ○ -Bottom
9 ● +Bottom Right ○ -Bottom Right
Zone Sequence 5786369

Blocks
○○○○○○○○●●
○○○○○○○○●●
○○○○○○○○●●
○○○○○○○○●○
○○○○○○○○●●
○○●●●●●●●●
○●○○○○○○●●
●○○○○○○○●●
●○○○○●●●●○
●●●●●●○○●○

Height:Width Ratio Rule
○ Greater than 8:1
○ Less than 1:8
● Greater than 3:2
○ Less than 2:3

Start to End Rule
● From Top to Bottom
○ From Bottom to Top
○ From Left to Right
○ From Right to Left

Select Training Example
○ Average ○ Example 3
● Example 1 ○ Example 4
○ Example 2 ○ Example 5

Figure 3 *GesRec* dialog box during training of the letter ‘d'

In order to test the software, a training set of the hand-written characters *a-z* plus 4 extra gestures (Space - , Backspace /, Delete], and Speak ^) was selected i.e. 30 classes. The software prompted each user for 5 examples of each character, giving a total training set of 150 examples.

The system was tested first of all by an able-bodied person. Characters were drawn in the air in a plane perpendicular to the floor, using the Polhemus tracker as the input source with a sensor attached to the index finger. The software performed as designed, enabling the data to be acquired and to be used to train the hybrid recognition system. Data acquisition, including on-line segmentation, took 9 minutes. During training, the software performed the transformation to the 10-by-10 grid and extracted the rule-based features. After data acquisition was completed, the 10-by-10 grid information was used as input to a 100-12-30 MLP neural network, which was trained with the backpropagation algorithm with initial weights in the range [-0.01,0.01] and a learning rate of η=0.9 until convergence was achieved. The neural network completed training successfully, converging in 28 epochs in 10 seconds, which is a short enough time to allow retraining as required.

Figure 3 shows the *GesRec* ‘Modify’ dialog box which automatically records the rule weightings and neural network input data for a single training example, in this case for the letter *d*. It can be seen that the HWR>1.5 rule is satisfied, as are several of the zone rules. Since the letter was ended at a lower point than it was started, the ‘From Top to Bottom’ rule is satisfied. Also shown is the sequence of zones passed through as the character was drawn, however this information was not incorporated into the recognition algorithm presented in this paper. Figure 4 shows five examples of the same letter, demonstrating variation for the same user.

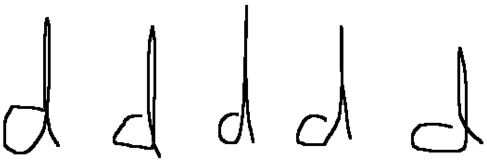


Figure 4 Five examples of letter ’d’, showing variation for a single user

VI. RESULTS

A good measure of how well the recognition system is able to separate the characters with rules alone is the total distance between rule vectors in the training set. Rather than show the full table, Figure 5 shows each character and its distance from the nearest and next-nearest character in the training set. It can be seen that most of the distances are large enough for the system to classify correctly using the rules alone. However a few distances are smaller, for example between s and g, h and k, x and z, which means some misclassifications are likely.

Character	Nearest Character	Distance to Nearest	Next Nearest Character	Distance to Next Nearest
a	c	4	]	4
b	k	3	h	5
c	a	5	e	6
d	y	3	j	4
e	c	4	m	4
f	d	7	y	7
g	s	1	j	4
h	k	1	b	4
i	l	7	h	8
j	g	4	s	5
k	h	2	b	3
l	b	5	k	5
m	e	6	u	6
n	h	6	u	6
o	a	4	c	6
p	y	7	b	9
q	l	5	x	7
r	v	6	x	7
s	g	0	]	6
t	a	7	e	7
u	m	6	n	6
v	r	6	c	8
w	m	6	h	8
x	z	2	r	6
y	d	4	j	6
z	x	3	e	6
-	^	10	l	11
/	x	9	y	9
]	g	4	s	5
^	n	6	r	10

Figure 5 Table comparing distances between characters in the training set

	Number of rejections	Number of mis-classifications	Total Number of Entries	% Correct
Rules only	Not applicable	26	150	82.7
Neural Network only	13	4	163	89.6
Hybrid	10	2	160	92.5

Figure 6 Table of results, comparing the recognition accuracy for rules only, neural network only, and the hybrid system

To test the recognition accuracy, the entire set of characters was entered again five times, and the number of

Proceedings of the Fourth IEEE International Conference on Electronics, Circuits and Systems,
Cairo, Egypt, December 15-18 1997, Volume 2, pp 850-853.

misclassifications was recorded. If the neural network gave a score below the acceptance threshold, the entry was rejected and the character re-entered until something was accepted. A neural network score of s =3 was found to give a good balance between the neural network and the 17 rules, and an acceptance threshold of 0.5 was found to be a good value to reject ambiguous entries. Figure 6 shows a comparison of recognition accuracy for the rules used alone, for the neural network used alone, and for the hybrid system. The percentage correct value is the total number of correct entries compared to the total number of entries. One of the criteria which can be used to judge the performance of a communication aid is the speed of user input. The total time to recognise a character including the time to enter the character and to perform recognition took between 1 and 2 seconds. Most of this time was taken up in drawing the character, including 0.2s for a time-out at the end of the character in order to perform segmentation. The recognition algorithm itself took less than 1ms to execute i.e. less than 0.1% of the total entry time., which compares favourably with other methods which have more computationally intensive algorithms, such as dynamic programming.

The communication aid was also tested in a clinical trial with a person who cannot speak properly due to a permanent tracheotomy, and cannot write because of a condition called myoclonus. The volunteer already communicated by spelling in the air with her finger, so she was an ideal person to test the system. In this trial the volunteer was able to train the hybrid system with the letters *a*,*b*,*c*,*d* and *e*, which took 16 minutes, including a break of 5 minutes. The increased time was partly due to the slower drawing time, which was between 2 and 4 seconds including a longer segmentation time-out of 1s, and partly because the volunteer needed to rest for several seconds between characters. No further characters were trained in that session because of the volunteer becoming tired. However, even with this partial training set, it was still possible to test the recognition accuracy, albeit for an easier task with 5 classes, rather than 30. As with the previous test, a neural network score s=3 and an acceptance threshold of 0.5 was used. Nine letters were entered, of which eight were classified correctly, and one ambiguous letter (which looked partly like *c* and partly like *e*) was rejected by the acceptance threshold. There were also two misclassifications due to unintentional movement, which on closer examination of the training data would have been rejected by a higher acceptance threshold of 0.6. The volunteer appeared to be highly motivated by the use of the software.

V. CONCLUSIONS

The results show that recognition accuracy can be improved using a hybrid rule-based and neural network system, rather than either of the methods used alone. With a carefully chosen acceptance threshold, ambiguous or spurious entries can be rejected. In the future, it will be beneficial to automatically determine the neural network score and the acceptance threshold. Furthermore, the acceptance threshold need not be the same for each character or gesture. Work is ongoing to examine both 2D and 3D algorithms for gesture recognition.

VI. ACKNOWLEDGEMENTS

This work was funded by grant A/P/0543 from the UK medical research charity Action Research for the project: “Improvement of assessment and the use of communication aids through the quantitative analysis of body movements of people with motor disabilities”. The authors also wish to thank the volunteers who have taken part in the testing of *GesRec*.

VII. REFERENCES

[1] Polhemus 3SPACE FASTRAK User’s Manual, Revision F, November 1993, Polhemus Incorporated, Colchester, Vermont, USA.
[2] TAPPERT C.C., SUEN C.Y. and WAKAHARA T. : “The State of the Art in On-Line Handwriting Recognition”, IEEE Trans. Pattern Analysis and Machine Intelligence”, vol. 12, no. 8, pp787-808, August 1990.
[3] HSIEH K-R and CHEN W-T : “A Neural Network Model which Combines Unsupervised and Supervised Learning”, IEEE. Trans. Neural Networks, vol. 4, no. 2, pp357-360, March 1993.
[4] MORNS I.P. and DLAY S.S. : “The dynamic supervised forward-propagation neural network for handwritten character recognition using fourier descriptors and incremental training”, Proc. Third IEEE International Conference on Electronics, Circuits and Systems (ICECS’96), Rodos, Greece, October 13-16 1996, vol. 2, pp1123-1126
[5] CAIRNS A.Y. “Towards the Automatic Recognition of Gesture”, PhD Thesis, University of Dundee, Scotland, November 1993.
[6] BURNISTON J.D. , CURTIS K.M. and CRAVEN M. : “A Hybrid Rule Based/ Rule Following Parallel Processing Architecture”, Proc. International Conference and Exhibition on Parallel Computing and Transputer Applications (PACTA) 1992, Barcelona, Spain, 21-24 September 1992, Part 1, pp729-735.
[7] GUBBINS P.R., CURTIS K.M. and BURNISTON J.D. : “A Hybrid Neural Network/Rule Based Architecture Used as a Text to Phoneme Transcriber”, Proc. International Symposium on Speech, Image Processing and Neural Networks (ISSIPNN’94), Hong Kong, pp113-116, April 1994.
[8] RUMELHART D.E., HINTON G.E. and WILLIAMS M.J. : “Learning Internal Representations by Backpropagation of Errors”, Nature 323, pp533-536, 1986.