VHDL Modelling of Fuzzy Wavelet Based Handwritten Recognition System

W.L. PANG,[*] M.B.I. REAZ,[*] M.I. IBRAHIMY,[**] L.C. LOW,[*] F. MOHD-YASIN,[*] R.A. RAHIM [**]
[*]Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia
[**]Kulliyyah of Engineering, International Islamic University, 53100 Selangor, Malaysia

Abstract: - The handwritten character recognition system using fuzzy wavelet is modelled using Very High Speed Hardware Description Language (VHDL). The fuzzy wavelet handwritten recognition is a system based on human perception. The time-frequency localization capabilities of wavelet transform in feature extraction enhances the accuracy of recognition at pixel level. The characteristic features of handwritten character in image format are extracted by taking Haar wavelet transform. The design was implemented using VHDL. The VHDL model is simulated and synthesized using Altera's Quartus II. The system operates up to 13.27 Mega Hertz (MHz) with equivalent longest propagation delay of 75.336ns.

Key-Words: - Fuzzy Wavelet, Handwritten Recognition, VHDL

1 Introduction

During the last forty years, human handwriting processing has most often been investigated and developed in many countries all around the world. At the dawn of the 3rd millennium, we guess that human handwriting processing will be more considered as a perceptual and interpretation task closely with research into human language [1]. Therefore, we would observe that, human handwriting processing is a very interesting field that could lead us to an easier interaction with digital devices such as computers, PDA and else.

The fundamental idea behind wavelets [2, 3, 4, 5] is to analyze according to scale. Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. Wavelet algorithms process data at different scales or resolutions. The wavelet analysis procedure is to adopt a wavelet prototype function, called an "analyzing wavelet" or "mother wavelet." The original signal or function can be represented in terms of a wavelet expansion; data operations can be performed using just the corresponding wavelet coefficients. Haar wavelet is the oldest among other mother wavelets. The Haar wavelet algorithms published are applied to time series where the number of samples is a power of two (e.g., 2, 4, 8, 16, 32, 64...). The Haar wavelet uses a rectangular window to sample the time series. The first pass over the time series uses a window width of two. The window width is doubled at each step until the window encompasses the entire time series. Each pass over the time series generates a new time series and a set of coefficients. The new time series is the average of the previous time series over the sampling window. The coefficients represent the average change in the sample window.

During the sixties, Professor Lofti Zadeh proposed a mathematical way of looking at the intrinsic vagueness of the natural human language, the "fuzzy logic". The fuzzy logic [6, 7, 8, 9] is used to make the computers think like humans. A fuzzy controller works in a similar way to a conventional control system where it accepts input, perform inference and generate outputs. However, it is more intelligent, stable and robust. The core of a fuzzy controller is a linguistic description prescribing appropriate action for a given state. Fuzzy linguistic descriptions involve associations of fuzzy variables and procedures for interference.

2 Design Overview
2.1 Input Processing

A user wrote a character on paper and being scanned to computer as an image file with .jpeg or .bmp extension. By using image processing software, the character image will be further processed. The character image will be standardized to 4 x 4 pixels. At first, by finding the most right, most left, most upper, lower point of image character, the input character image will be cropped into 1 pixel more than the point found. Next, the character image resizes to 4 x 4 pixels. At this stage, the image would

be passed to Matlab to convert it to matrix representation format. The input image is gray scale image. It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored.

To represent the brightness of pixel, class uint8 is used. Each pixel assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. Each cell value of matrix would be the data input to wavelet feature extraction system. The graphical view of conversion of character image to matrix format is shown in Fig. 1. It is seen that each character image will transform to a matrix that consists of 16 data within. Each cell value of matrix would be the data input to wavelet feature extraction system. These 16 data will be passed to handwritten character recognition system for further processing.
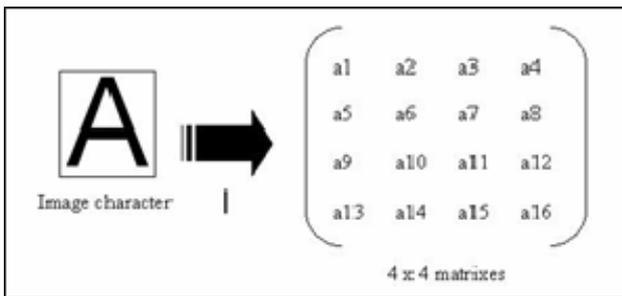

Fig. 1: Conversion of Character Image to Matrix Format

2.2 Handwritten Character Recognition System

The character recognition system is consists of two main subsystem, namely wavelet feature extraction system and fuzzy logic controller system. The first system is used to extract the characteristic feature from image using Haar wavelet transform at different layers while the next system is used for classification and recognition purpose. The overview of overall system is shown in Fig. 2.

The overall structure consists of the main entity of the recognition system, "CharacterRecog" and two components namely, "FeatureExtraction" and "MainFLC". The CharacterRecog entity declaration is shown in Fig. 3. The inputs of g1 to g16 are directly mapped to the inputs of FeatureExtraction. The outputs of featureExtraction are mapped to the inputs of MainFLC at input values of X0 to X_11. The detail of each component will be further explained in following section.
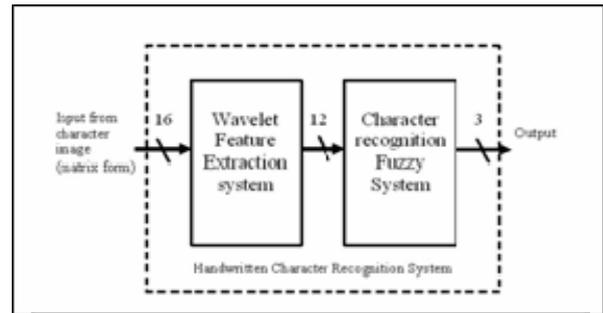

Fig. 2: Fuzzy-Wavelet Based Handwritten Character Recognition System

Fig. 3: The CharacterRecog Entity Declaration

2.3 Wavelet Module

After input processing progress, the character image will be converted to 4 x 4 matrix formats that consist of 16 data. These 16 data are the inputs to the wavelet feature extraction system. The system use the Haar wavelet function, $\psi_{jk} = \psi (2^j t - k)$ to extract the feature coefficients from the character image at different frequency layers. Since the character image file is 4 x 4 pixels only, 2 frequency layers of feature coefficients will be collected. Thus, inside the system, 2 wavelet functions are derived from the mother wavelet Haar to do the wavelet transform, where each wavelet function will extract the feature coefficients at highest frequency while the other one would extract the coefficients at second highest

frequency. The wavelet functions that used in this system is based on this general equation:

$$C_{j,k} = (a_{j+1, 2k} - a_{j+1, 2k+1}) / 2 \qquad \dots (1)$$

$$a_{j,k} = (a_{j+1, 2k} + a_{j+1, 2k+1}) / 2 \qquad \dots (2)$$

Where $C_{j,k}$ are the wavelet coefficients, $a_{0,k}$ are the input data, j represents the frequency layer, k represent arbitrary integer. The wavelet feature extraction system is shown in Fig. 4.

The input data will be decomposed to feature coefficients at the highest frequency layer in the Wavelet_1. This block takes in 16 data as the input. As reference to [1], the highest frequency layer is obtained when j = -1. From equation (1) and (2), the wavelet function at this block will be as below.

$$C_{-1,k} = (f(2k) - f(2k + 1)) / 2 \qquad \dots (3)$$

Where f(2k) represent the $a_{0,2k}$ (the input data at number 2k), f(2k – 1) represent the $a_{0,2k+1}$ (the input data at number 2k+1).
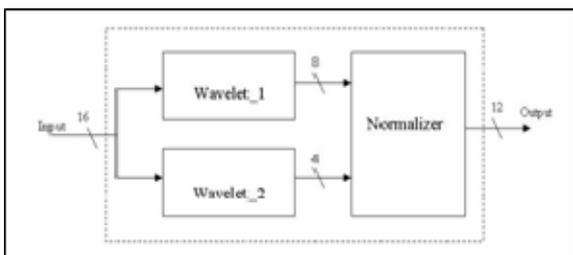

Fig. 4: Wavelet Feature Extraction System

The input data will be decomposed to feature coefficients at the second highest frequency layer at Wavelet_2. Same as Wavelet_1 block, it takes in 16 data inputs from character image matrix. As reference to [1], the second highest frequency layer is obtained when j = -2. From equation (1) and (2), the wavelet function at this block is shown as below.

$$C_{-2,k} = \tfrac{1}{4} (f(4k) + f(4k+1) - f(4k+2) - f(4k+3)) \dots (4)$$

Where f(k) represent the data values at number k.

The normalizer block takes in 12 inputs, namely all the 12 coefficients from the wavelet_1 and wavelet_2 blocks. From observation, it is seen that each coefficients of the output wavelet blocks having different ranges. Some of them are positive integer; some are negative integers or floating-point numbers. Due to this reason, it is very hard for further processing in the future blocks. Thus, this normalizer block functions as converting those numbers to a fix

range, that is 0 to 255. In this way, each value could be represented using 8 bits ($2^8 = 256$).

2.4 Character Recognize Fuzzy System

A Character Recognize Fuzzy system will be designed. It will be 12 inputs and 3 outputs. This system is targeted to recognize character A, B, C and D only. Depend on the result of 3 outputs, the system will determine that either characters A, B, C, and D is detected. In this model, there will be 13 different membership functions that comprise of X0, X1, X2, X3, X4, X5, X6, X7, X8, X9, X_10, X_11 and output.

At first, the sample wavelet coefficient sets collected in this project for character A, B, C, and D, which are listed as follow:

Character A = [0.5 2 2 2.5 2 0.5 -1.5 4.5 2.25 1.75 0.75 2]

Character B = [1 -1 -1.5 1.5 1 -0.5 -3 -4 0.5 -1 -1.25 1.5]

Character C = [-3 1 -0.5 3 3 0.5 4 -6 0 0.75 2.25 1.5]

Character D = [-1.5 -2 0.5 -2 2 -4 1 -1.5 -0.75 -1.25 -1.5 2.75]

The first coefficient of each character would be X0, the second coefficient is X1, and the third coefficient is X2... and the twelfth coefficient is X_11. From the samples coefficient set, X0 for characters A, B, C, D is [0.5 1 -3 -1.5]. These four values will be set to maximum of fuzzy value, 1. Due to comparison with other sample set, the acceptance range for each membership function is specified. The range is chosen from -4 to 2 and it has four fuzzy variables, LOWER, LOW, MEDIUM and HIGH. This membership function is shown in Fig. 5.
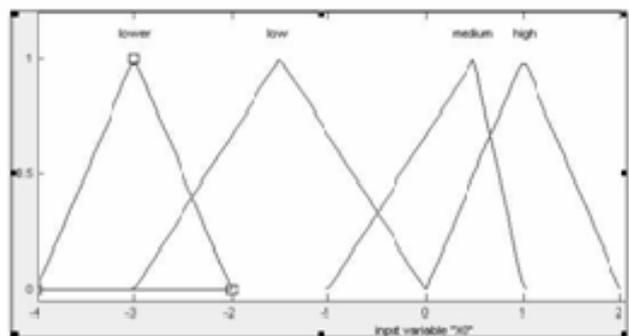

Fig. 5: Membership Function of X0

After designing the membership functions of inputs and output and if-then rule, the component blocks inside the character recognize fuzzy system would be discussed. The Fig. 6 shows the overview

of character recognize fuzzy system. From the Fig., the fuzzy system consists of 4 main modules: Fuzzifier, Inference, Rule Selector, and Deffuzifier. The system consists of twelve inputs X0 up to X_11 each corresponding to the values of coefficients extracted from the wavelet system respectively. The output of the system will give us 3 output values to be applied at each corresponding values of X0 to X_11.

The fuzzifier is the starting block of the entire fuzzy system. It takes in twelve inputs namely, X0, X1, X2, X3, X4, X5, X6, X7, X8, X9, X_10 and X_11. Both these values have been assigned their value ranges in the membership functions of the fuzzy rule based system. These input values are crisp values from the real world and need to be converted to fuzzy values thus requiring the fuzzifier block. The universe of discourse of each input that contains the membership functions is divided into several sections based on the mathematical equations of the fuzzy variables.
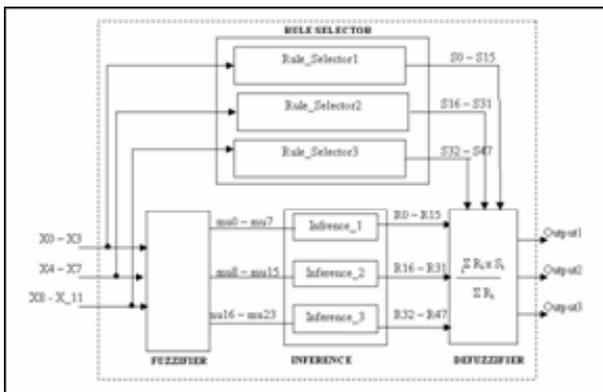

Fig. 6: Structure of Character Recognition Fuzzy System

The inference block accepts twenty-four inputs from the fuzzifier block. Thus, the inputs of the inference would be u[0], u[1], u[2], u[3], u[4], u[5], u[6], u[7], u[8], u[9], u[10], u[11], u[12], u[13], u[14], u[15], u[16], u[17], u[18], u[19], u[20], u[21], u[22], and u[23]. In the inference block, the inputs will be inferred using the max-min composition and Mamdani- min implication operator where forty-eight outputs R0 to R47 will be produced.

Within the rule selector block, each membership function of every input will be subdivided to several regions as to ensure that only two fuzzy variables exist in each region of comparison. The rule selector will examine and compare each region with others to determine which rule will be fired. Therefore, after analyzing the membership functions of all the inputs, it was decided that X0 be divided into 8 region, X1

be divided into 6 regions, X2 be divided into 7 regions, X3 be divided into 7 regions, X4 be divided into 4 regions, X5 be divided into 6 regions, X6 be divided into 9 regions, X7 be divided into 9 regions, X8 be divided into 9 regions, X9 be divided into 9 regions, X_10 be divided into 8 regions, and X_11 be divided into 5 regions.

Since each of inference block and rule selector block had been divided to 3 small block, namely Inference_1, Inference_2, Inference_3, Rule_Selector1, Rule_Selector2, and Rule_Selector3, thus the deffuzifier part would consists of 3 block also; Deffuzifier_1, Deffuzifier_2, and Deffuzifier_3. The Deffuzifier_1 will associate with Inference_1 and Rule_Selector1; Deffuzifier_2 will associate with Inference_2 and Rule_Selector2, and Deffuzifier_3 will associate with Inference_3 and Rule_Selector3. The function of the defuzzifier is to convert the fuzzy values in the system back into crisp values to be fed out to the real world. The defuzzification method chosen in this module is the Center of Average or the Centroid method. This method of defuzzification is expressed by the equation $(\Sigma R_i * S_i)/(\Sigma R_i)$.

3 Results

Twenty (20) character image file had been prepared for analysis, 5 images file contains character A, 5 image file contains character B, 5 image file contains character C, and the last 5 image file contains character D. For every 5 image per character, there is a sample set for each character included within. The 20 character images are as shown in Fig. 7.
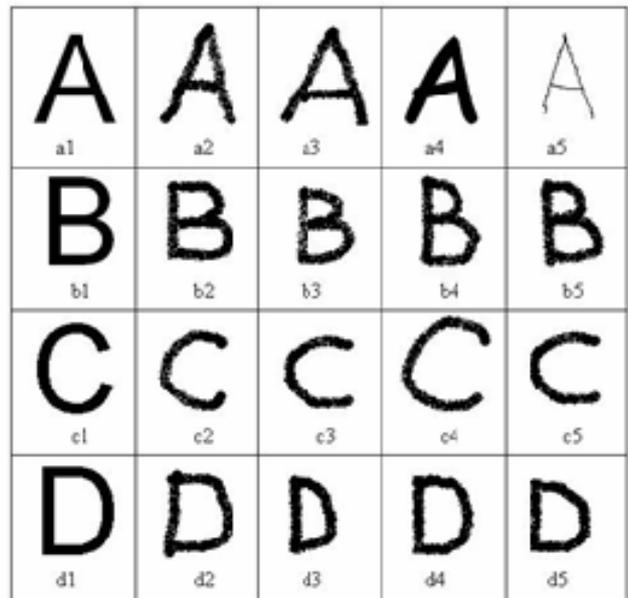

Fig. 7: 20 Character Images

When matrix values of each character image are inserted to this handwritten character system, which consists of wavelet system and fuzzy system, the result, are represented by 3 outputs, Output1, Output2, and output3. Each of them would give 8-bit unsigned digital signal. However, actually, these 8-bit signals represent the floating-point binary value.

The project is successfully compiled under Quartus II 4.1 Web Edition. The maximum frequency achieved for this project is 13273866Hz. The compilation report is as shown below:

Flow Status	Successful - Sat Mar 12 16:53:00 2005
Quartus II Version	4.1 Build 181 06/29/2004 SJ Web Edition
Revision Name	CharacterRecog
Top-level Entity Name	CharacterRecog
Family	Stratix II
Device	EP2S15F484C3
Timing Models	Preliminary
Total ALUTs	7,183 / 12,480 (57 %)
Total registers	0
Total pins	152 / 343 (44 %)
Total memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	63 / 96 (65 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)

Simulation is done on all the 20 character images. The simulation result of character image a1 is used as example for further explanation. The character a1 in matrix format is as shown below:

$$a1 = \begin{pmatrix} 10 & 9 & 7 & 3 \\ 10 & 6 & 7 & 2 \\ 9 & 5 & 6 & 5 \\ 8 & 11 & 10 & 1 \end{pmatrix}$$

The simulation output of a1 is shown in Fig. 8. Each matrix cell data are the inputs to handwritten recognition system. These data will be mapped to the inputs of the whole system as input values g1, g2, g3, g4, g5, g6, g7, g8, g9, g10, g11 and g12. During the simulation, all the value are represented by hexadecimal number thus the value of 10 will converted to 0A, value of 9 convert to 09 and so on. These data will first be processed by Wavelet Feature Extraction System where Wavelet_1 and Wavelet_2 block take the inputs in to generate coefficients C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 and C12 with values 1, 4, 4, 5, 4, 1, -3, 9, 9, 7, 3 and 8. These values map to the inputs of normalizer and give BF(191), D4(212), D0(208), C8(200), 7F(127), DD(221), 48(72), DB(219), D2(210), CF(207), 86(134) and 66(102) as outputs at X0 to X_11. At

Fuzzifier stage, these values will be fuzzified to membership function values as FE, 00, 03, FC, FB, 00, 03, FC, 02, FD, 00, FF, 06, F9, F9, 00, FF, 00, FE, 00, FE, 00, FF and 00 to outputs MU01, MU02, MU11…MU81…MUa1, MUa2, MUb1 and MUb2.
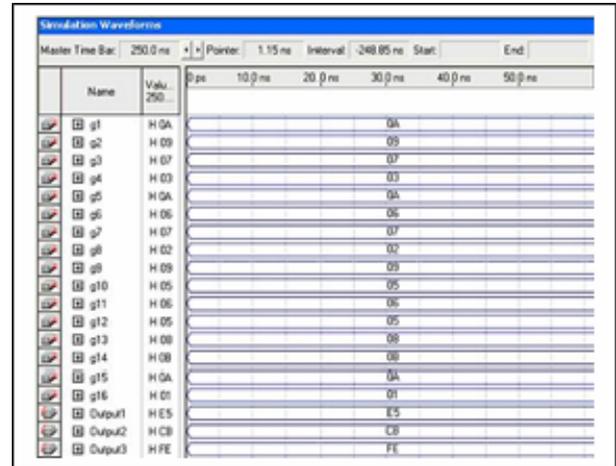

Fig. 8: Simulation Result of Character a1

After collecting all the outputs from each simulation, the results are tabulated as Fig. 9. There are only 1 character A is recognized out of 5, 3 characters B are recognized out of 5, 1 character C is recognized out of 5, and 1 character D is recognized out of 5.

Total recognition efficiency
$$= \frac{\text{number of successfully recognized} * 100}{\text{total test character}}$$
$$= 6 * 100 / 20 = 30 \%$$

Test Character	Output1	Output2	Output3	Output Mapped	Recognized Character
a1	89.45	79.29	99.22	A1, A2, A3	A
a2	0	0	0	-	-
a3	0	0	0	-	-
a4	0	0	0	-	-
a5	0	0	0	-	-
b1	39.45	24.61	14.45	B1, B2, B3	B
b2	0	0	0	-	-
b3	39.45	24.61	14.45	B1, B2, B3	B
b4	0	0	0	-	-
b5	39.45	24.61	14.45	B1, B2, B3	B
c1	29.29	69.53	59.38	C1, C2, C3	C
c2	0	0	0	-	-
c3	0	49.61	0	-, D2, -	-
c4	0	0	0	-	-
c5	0	0	0	-	-
d1	5.29	49.61	0.39	D1, D2, D3	D
d2	0	0	0	-	-
d3	0	0	0	-	-
d4	0	0	0	-	-
d5	0	0	0	-	-

Fig. 9: Simulation Results of 20 Character Images Matrix

4 Discussion

The recognition process of system did not go well as planned. Most of the handwritten character image fails to be recognized through this system. The main cause of this problem would be the designs of membership function for each wavelet coefficient are not effective. These triangle membership functions are generated based on several set of coefficients of character A, B, C, and D. Each membership function has its designed range so that it can accept variety values other than the exact value. However, if a value or coefficient is outside the range of the function, then it could not be processed. In others mean, if the other coefficients extracted from the handwriting character image are different from the sample coefficient more than expected, then the coefficient could not be recognized. In fact, from the simulation results, it is clearly seen that the handling of various coefficients from different handwritten characters is very weak.

5 Conclusion

A fuzzy-wavelet based offline handwritten character recognition system was successfully built. The model provided major insights into functionality and implementation of fuzzy-wavelet technique and it is proved that the fuzzy wavelet can be applied to handwritten character recognition. This model was constructed by wavelet feature extraction system and fuzzy system. The first system used the Haar wavelet to extract characteristic feature from character image. Feature extraction with wavelet transform is faster than other method. It is concluded that wavelet transform is a powerful technique for representing data at different scales and frequencies. The next system, fuzzy system used fuzzy logic in classification and recognition of character. It can also be concluded that fuzzy logic is indeed very important and useful for solving and implementing complex and imprecise models. Such models would not be able to be implemented effectively and efficiently by any other simple models such as PID controls.

References:
[1] I. Rybak, V. Gusakova, A. Golovan, N. Shevtsova and L. Podladchikova, Modeling of a Neural Network System for Active Visual Perception and Recognition, *Proceedings of ICPR*, vol. 2, pp. 371-373, 1994.

[2] S. Correia, J. M. Carvalho and R. Sabourin, On the Performance of Wavelets for Handwritten Numerals Recognition, *Proceeding of ICPR 2002*, Quebec - Canada, August 2002.

[3] R. Plamondon, *A Handwriting Model Based on Differential Geometry*, Computer Recognition and Human Production of Handwriting, R. Plamondon, C. Y. Suen, M. L. Simner (eds.), Word Scientific Publishing, pp. 179-192, 1989.

[4] R. Plamondon and G. Lorette, *Designing an Automatic Signature Verifier: Problem Definition and System Description*, Computer Processing of Handwriting, R. Plamondon and C. G. Leedham (eds.), World Scientific, pp. 3-20, 1990.

[5] J. T. Favata and S. N. Srihari, Recognition of Handwritten Words for Address Reading, *Proceedings of the 4th U.S. Postal Service Advanced Technology Conference*, Washington DC, pp. 191-205, 1990.

[6] W. G. Wee and K. S. Fu, A Formulation of Fuzzy Automata and Its Application as a Method of Learning System, *IEEE Tran. System Science and Cybernetics*, vol. SSC-5, July 1969.

[7] F. Kimura, M. Shridhar, Handwritten Numerical Recognition Based on Multiple Algorithms, *Patt. Recog.* 1991, vo1. 24, no. 10, pp. 969-983.

[8] Zhu Xiaoyan, Multiple Neural Networks Model and its Application in Pattern Recognition, *Proceedings of the IEEE International Conference on Neural Information Processing*, 1995, pp. 996-969, Beijing.

[9] Wang Song and Zhu Xiaoyan, Multiple Experts Recognition System Based on Neural Network, *Proceedings of the International Conference on Pattern Recognition*, 1996, pp.545-548, Vienna, Austria.