Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Fuzzy Neural Hybrid Method for Sinhala Character Recognition

[1]M. A. P. Chamikara*, [2]S. R. Kodituwakku, [3]A. A. C. A. Jayathilake, [4]K. R. Wijeweera
[1] Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya, Sri Lanka
[2] Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya, Sri Lanka
[3] Department of Mathematics, Faculty of Science, University of Peradeniya, Sri Lanka
[4] Postgraduate Institute of Science, Faculty of Science, University of Peradeniya, Sri Lanka

***Abstract-** Identifying patterns in Sinhala characters is a very difficult task because the similarities between the Sinhala characters lie in a very mild extent. Printed documents and hand written character recognition play a major role in many areas like security, defence, business and so forth. However, identifying hand written Sinhala characters is difficult because of their bizarre arrangements. Due to this problem most of the proposed character recognition systems in the literature record low level of accuracy. This paper presents a hybrid method for Sinhala character recognition. The method is based on fuzzy logic and neural network concepts. The algorithm has been tested for several different hand written letters and the results showed an overall average accuracy of 81%.*

***Keywords:** Artificial Neural Networks, Character Recognition, Fuzzy Logic, Image Processing, Sinhala language*

I. INTRODUCTION

Basically, there are different versions of characters such as handwritten characters, machine printed characters, scanned characters and so on. Handwritten character recognition can be explained as the mechanism that enables the computers to translate human writings into text [1]. Printed character recognition can also be defined as the translation of machine printed characters into text. Different approaches are necessary in dealing with different versions [1] .

Technical challenges that have to be encountered in character recognition can be classified into three classes of problems. First problem is that there are a large number of symbols to be handled. Second one is the deformation that can lead the same character to appear in different shapes. After applying geometric transformations like translation, rotation, scaling, stretching and so forth, we can derive different shapes of the same character. Third problem is the image defects that happen due to printing, optics, scanning, quantization, binarization and so on[1].

The most difficult problem to address from the above is the hand written character recognition. The reason for this is the handwriting consists of elongated strokes, whereas the machine printed characters consist of regularly spaced blobs in more clear manner [1].

English character recognition is relatively simple because the shapes of the characters in English language are simple and contains only 26 letters. Moreover, there are no modifier symbols attached to those characters to make vocal sounds. The vocal sounds are inherited from the vowels which are already available within the same 26 characters. Therefore, recognition task is not that difficult. When it comes to Sinhala character recognition, things become much more difficult. Sinhala alphabet is made of 18 vowels, 41 consonants and 17 modifier symbols. A vowel can appear only as the first character of a word. A consonant is modified using one or more modifier symbols to produce vocal sounds. Total number of modifications from the entire alphabet including the basic characters is nearly 400 [2]. Dealing with such a large number of variations is not a simple task.

Though characters in Sinhala language are different to each other and can be distinguished, there are some different characters those resemble each other. Sinhala characters are generally round in shape and differ from English characters as depicted in figure 1.

Fig.1. Three different Sinhala characters with similar figures

Several approaches have been proposed to recognize Sinhala handwritten characters and they can be found in the literature. The conventional and the most widely used technique is the use of artificial neural networks [2] . One of the major advantages in using the neural networks is their ability to respond to variations[3]. This is more significant when applied to hand written character recognition. Rajapkse et. al. [4] have developed a neural network based approach in which the neural networks are used to recognize off-line Sinhala characters. The foundation of this method is the popular back propagation which is a pattern classification neural network. When the handwriting of an individual was trained to the system it has recorded the accuracy of 88%. The accuracy of this method decreases to 75%, when choosing handwritten characters of a set of individuals.

Some other methods have been developed to recognize Sinhala characters using a fuzzy approach. Kodituwakku S.R. et. al [5] presented a fuzzy logic based techniques to segment characters. This method works mainly for smoothly written characters and for characters which do not tend to conflict with another character. The system's accuracy is also limited to 65%. Grouping of characters into subsets before applying fuzzy logic could be used to improve the accuracy [5] . Another segmentation based procedure has been developed by Kodituwakku S.R. et. al [12] which is based on character segmentation. They have developed this algorithm for the English character recognition. The test results have shown an accuracy of 75% for both uppercase and lowercase characters.

Use of hidden Markov models (HMM) is another approach to address the character recognition problem. Hewavitharana S. et. al. [6] have introduced a statistical method to recognize offline Sinhala handwritten characters by using the hidden Markov models This method uses the discrete form of the hidden Markov models. In this method, first the characters are classified into one of three character groups based on structural properties of the text line and then recognize them. The classification accuracy of this method is 64.3%. Because of the feature extraction process the two-dimensional spatial information of character images are reduced into a single dimension array of values. Therefore, some information is lost during the recognition process. This leads to under-represent character classes and hence contributed to the low accuracy. The HMM classifier is then used for final recognition. The HMM have been used widely in speech recognition and recent trend is to use them in character recognition [7, 8, 9] .

Incorporating the ideas in evolutionary computation to solve the character recognition problem is also an interesting approach. Jayasekara B. et. al. .[10] have proposed a genetic algorithm based alphabet training method. This method consists of two phases: alphabet training using genetic algorithm and script recognition using correlation based mapping. In the first phase, each character is trained using a separate genetic algorithm. The alphabet was trained using a genetic algorithm after pre-processing the character set. The horizontal and vertical distances are used as the features to represent the characters and are trained using both horizontal and vertical distances to corresponding pixels of each character sample. The language scripts are segmented using both horizontal and vertical projections in the recognition step. After that, the segmented characters are grouped using three layered structure of the line and core character is mapped with the trained alphabet based on correlation. This method shows 94% recognition rate when the characters have unique distinct shaped and 75% recognition rate for confusing shapes.

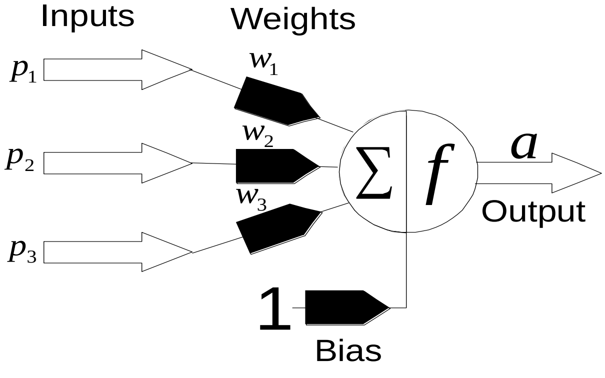
Several types of decision methods, including statistical methods, neural networks, structural matching (on trees, chains, etc.) and stochastic processing (Markov chains, etc.) have been used along with different types of features. Many recent approached mix several of these techniques together in order to obtain improved reliability, despite wide variation exist in handwriting [6].

In this research work we attempted to combine both neural network and fuzzy logic techniques for developing a hybrid method to identify Sinhala characters. Experimental results showed that this combination works better than the existing methods in the sense of recognition of more Sinhala characters. Neural networks are used to categorize the characters into groups and further processing is done with the help of fuzzy logic.

II. MATERIALS AND METHODOLOGY

A. Artificial Neural Networks.

An artificial neural network is a collection of artificial neurons which simulate the learning method of biological neurons which use mathematical models or computational models in information processing. Such a network is composed of an interconnected group of artificial neurons. Therefore, this kind of artificial neural networks will work as an adaptive system that change their structure based on external or internal information, that have been used to train the neural network. Figure 2 shows a sample neural network.



$$a = f\left(p_1 w_1 + p_2 w_2 + p_3 w_3 + b\right) = f\left(\sum p_i w_i + b\right)$$

Fig.2. An artificial neuron

The weights in this network are analogous to the synaptic strengths. Sum of all the weighted inputs will be taken and this value decides the activation level of the neuron. Let us denote the net input from the neuron as NET. Then NET signal is further tuned using an activation function F. Therefore, output signal of the neuron is, OUT=F (NET), where figure 2 denotes the OUT as 'a' [4].

In implementing intelligent systems, neural networks has been identified as a better alternative to conventional artificial intelligent techniques. The origin of artificial neural network concept is believed to occur in 1940s. The first mathematical model that imitates the biological neuron was published in 1943 by McCulloch and Pitts. The new concept was named as "a neural network" because of its similarity to a network of interconnected neuron like nodes [4].

B. Learning Mechanism of Neural Networks.

The behaviour of the artificial neural network is changed with the environment. This feature enables the neural network to learn the things from the environment. They self-adjust to produce consistent responses. There are a large number of learning algorithms to be used in neural networks [4].

Once we have trained the neural network, the response is not sensitive to the small changes in its input. This significant characteristic helps to see through noise and distortion. In the real world situations, noise and distortion is a normal phenomenon. Therefore, the neural network is a vital tool that can be used in real world applications more usefully. Furthermore, the artificial neural networks have the ability to abstract the essential set of inputs from a collection of inputs. Artificial neural networks can be trained for a collection of distorted or deformed versions of a given letter. After the training phase has been done adequately, such a distorted character will result the neural network to produce a perfectly formed letter [4].

The back propagation algorithm is the widely used algorithm for training multilayer artificial neural networks. This algorithm provides a systematic method for training multilayer artificial neural networks. Figure 3 shows the basic model of a feed forward back propagation neural network.

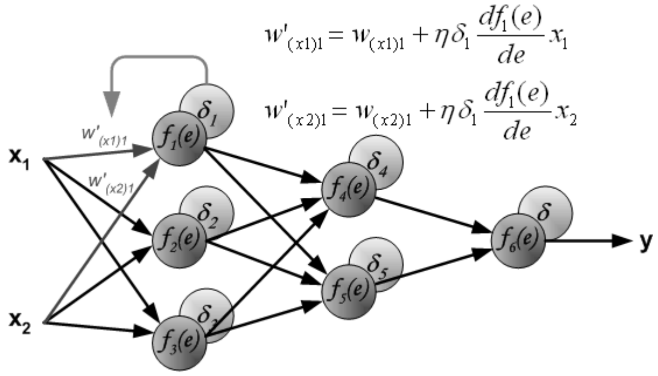


Fig. 3 Feed forward back propagation neural

C. Fuzzy Logic.

The term "fuzzy logic" refers to logic of approximation. Boolean logic assumes that every fact is either entirely true or false. Fuzzy logic allows for varying degrees of truth. Computers can apply this logic to represent vague and imprecise ideas, such as "hot", "tall". Fuzzy set theory offers a variable notion of membership [11].

A fuzzy relation acts as an elastic constraint on the values that may be assigned to a variable. In the following formula, a person of age 25 could still belong to the set of young people, but only to a degree of less than one, say 0.9.

$$\mu_{young}(x) = \begin{cases} 1 & age(x) \le 20 \\ 1 - \dfrac{age(x) - 20}{10} & 20 < age(x) \le 30 \\ 0 & age(x) > 30 \end{cases}$$

Now the set of young contains peop early decreasing degree of membership. Elements of a fuzzy set are mapped a function-theoretic form. This function maps elements of a fuzzy set A to a If an element in the universe, say x, is a member of fuzzy set A, then this mapping is given in the following form [11].

$$\mu_A(x) \in [0,1]$$

D. Fuzzy Expert System

Fuzzy expert system is a collection of membership functions and rules that are used to reason about data. Usually, the rules in a fuzzy expert system have the following form [11]:

"if x is low and y is high then z is medium"

E. Fuzzy Inference System

A fuzzy inference system (FIS) essentially defines a nonlinear mapping of the input data vector into a scalar output, using fuzzy rules [11] .The mapping process involves

- input/output membership functions
- fuzzy logic operators
- fuzzy if–then rules
- aggregation of output sets
- defuzzification.

Figure 4 depicts the architecture of a fuzzy inference system.

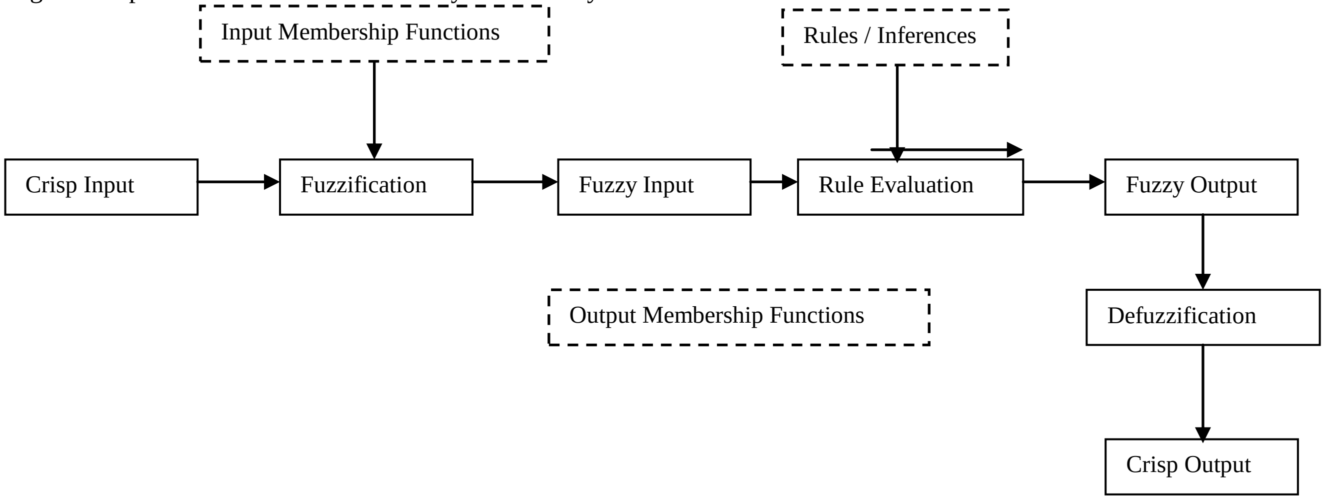


Fig. 4. Architecture of a fuzzy nce system

F. Sinhala Language and Characters.

The Sinhala alphabet contains 18 vowels, 41 consonants and 17 modifier symbols as mentioned above. A vowel can appear only as a first character of a word. A consonant can be modified by adding one or more modifier symbols to generate vocal sounds. Sinhala characters are entirely different from those based on Latin script and other languages for which recognition systems have been implemented [4] . So, it is needed to develop a different model to identify Sinhala characters. Most of the traditional approaches used for characters evolved from Latin script, cannot be used to recognize Sinhala characters without ambiguity. That is why most of the Sinhala character recognition systems have a poor performance.

The widely used language in Sri Lanka is Sinhala. It is one of the oldest writing systems available in South and East Asia. The earliest Sinhala writing can be found in third century BC. The origin of Sinhala language has occurred from Brahmi script. The alphabet of Sinhala language is one of the largest alphabets available in the world. It contains 58 letters. According to some scholars Sinhala language consists of 16 vowels, 2 semi-consonants, 40 consonants and 13 consonant modifiers also known as strokes of character modifiers. These graphical signs always used in conjunction with consonants. Consonant modifiers can be located at different locations around a character. Therefore, Sinhala language characters are very complex compared to English. The basic shape of Sinhala characters are curve shaped [4] .

Some of the characters in Sinhala language show a very close similarity as depicted in figure 5. This feature makes the recognition task very difficult.

Fig.5. Sinhala characters with a very close similarity

Figure 5 shows the characters are "TA", "MA", "WA", "CHA", "EA" from left to right. When observing the third and fourth characters, the only difference between those two characters is a small line segment. Similarly the fourth and fifth characters only differ by a small curve. First and second characters differ only from the centre of the character.

Fig.6. Sinhala characters with a very close similarity

In figure 6 "NA", "THA", "KA" characters are depicted from left to right. Here the first and second characters only differ by a slight difference at the front of the characters, and the second and third characters only differ at the bottom of two characters.

Fig.7. Sinhala characters with a very close similarity

"JA", "PA" characters are shown in figure 7 from the left to right. The two characters differ only by a stroke.

Fig.8. Sinhala characters with a very close

In figure 8, "GA", "HA" characters are depicted from the left to right. Only a slight difference at the top left corners of the characters can be observed.

Fig.9. Sinhala characters with a very close similarity

In figure 9, the "DA", "O", "MBA" characters are shown from the left to right. Observe the first and second characters, only difference is the middle shape of the characters. The second and third characters only differ at the bottom of the characters.

Because of the above pointed similarities among characters, the recognition of the Sinhala characters becomes very difficult. What we have discussed so far is just the difficulties occur when trying to distinguish isolated characters. In Sinhala language, modifier symbols are used to generate vocal sounds.

Fig.10. Adding vocal sounds to Sinhala characters

In Figure 10, left character is the original one "SA", by adding a modifier symbol we can make it "S". Observe the slightness of the difference before and after adding the modifier symbol. Therefore, the appearance of modifier symbols makes the recognition task much more difficult.

As we have mentioned earlier, handwritten character recognition is the most difficult task in character recognition. On top of all those difficulties mentioned above, the same character has been written differently by different persons as shown in Figure 11. This will add more complexity to the recognition process.

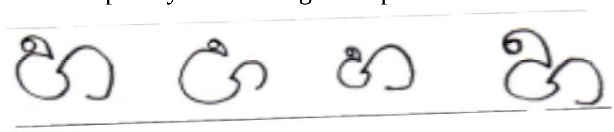


Fig.11. Same character in different

The position of the character and the relative sizes of parts of the character can be changed. Therefore, a huge extent of pre-processing is necessary in such situations to make all the letters similar to each other so that the recognition process generates a high accuracy.

G. Architecture of the System

Architecture of the proposed system is depicted in figure 12. All the programming of the introduced system was carried out using MATLAB 7.12.0.

H. Image Pre-processing steps

i) RGB to Grey:
Scanned image is converted from RGB to grey for further processing.

ii) Noise Removal:
The grey image was then filtered with an averaging filter to blur the noise. Then the resulting image was filtered using a median filter to filter the noise and obtained a less blurred and noise removed image.

iii) Image binarization:
Noise removed image was then subjected to the next level of dimensionality reduction through binarization. Image dilation was performed on the resulted image to reduce the pixel discontinuities of the characters. Then the individual characters were cropped in such a way that the area of interest is equal to the character sizes.

iv) Image skeletonization:
The cropped characters were then subjected to skeletonization. The spur generated during the skeletonization process was then removed by using an image morphological operation to remove spur as depicted in figure 13.

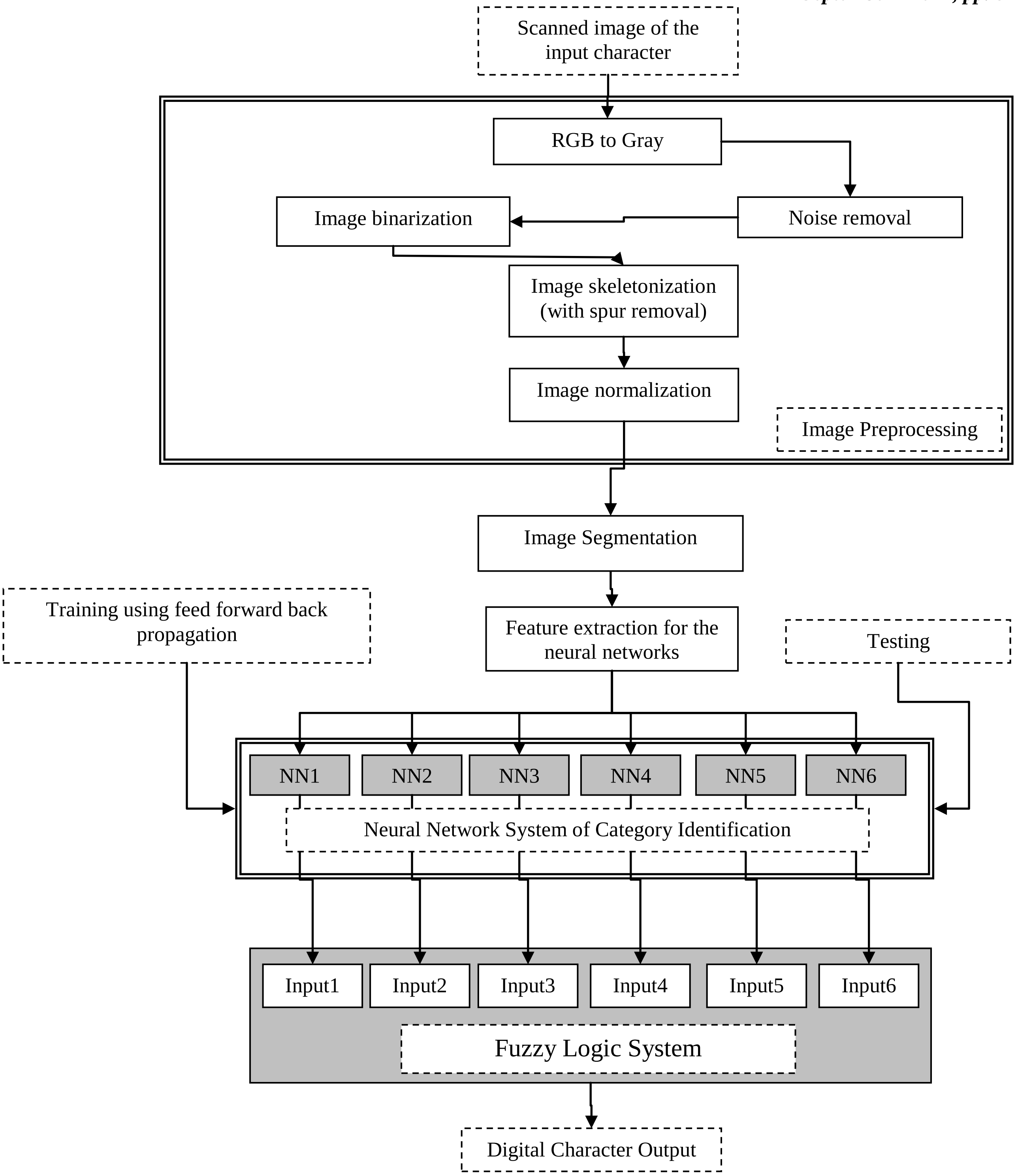


Fig.12. Architecture of the system

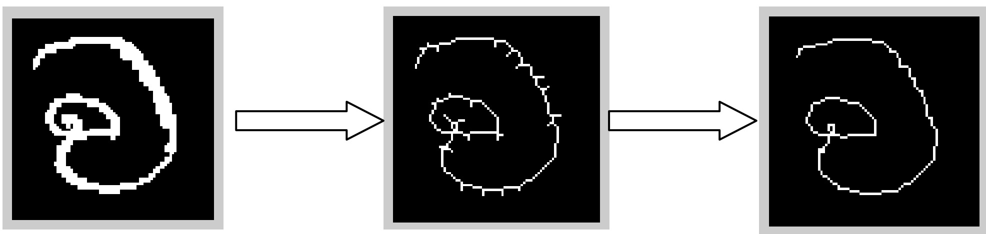


Fig.13. Skeletonization and Spur removal of the hand written Sinhala character "VA"

v) *Image Normalization:*
The skeletonized characters were then taken into a predefined resolution of 96x96.

I. Image Segmentation

Each image was segmented in to six segments. The segmentation process is depicted in figure 14 for the hand written Sinhala character “VA”. The six segments are not mutually exclusive. As shown in figure 14 the segments can have intersections with at most of two other segments.
The segments have been identified in such a way that, each character is divided to six subsections with a significant characteristic. Basically the four outer most segments starting from segment 1 to 4 have been identified to extract the external characteristics of a particular character while the inner segments, segment 5 and 6, are there to extract the internal characteristics of the letter.

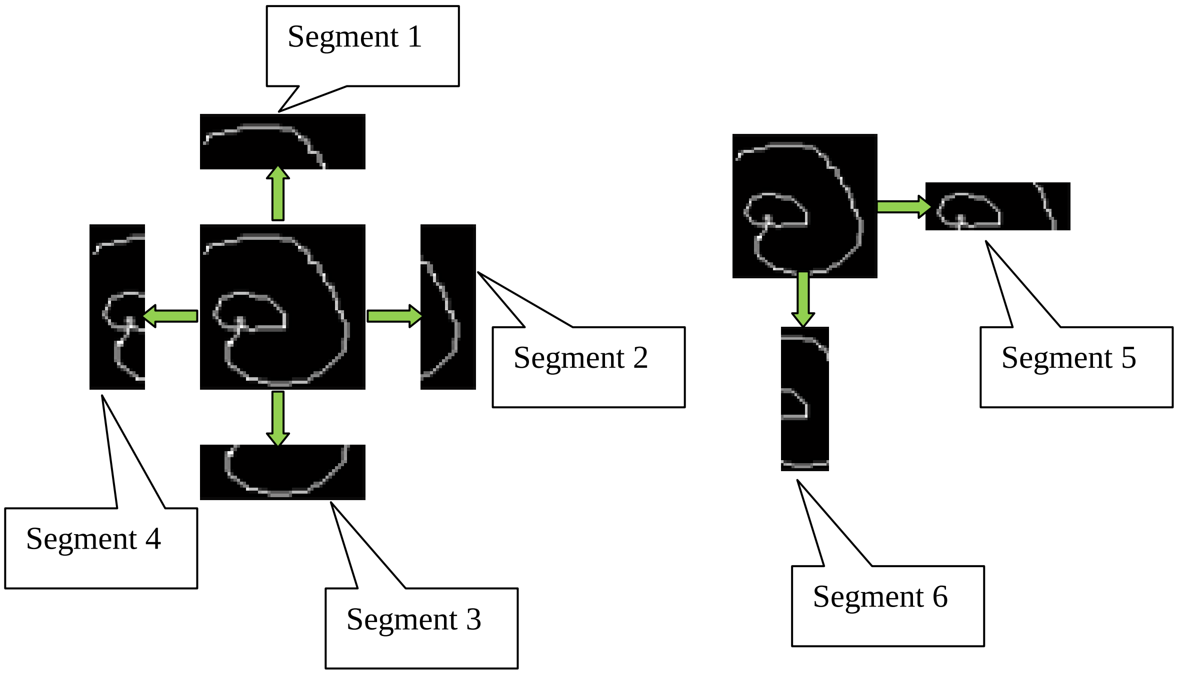


Fig.14. Six segments extracted for the hand written Sinhala character “VA”

J. Feature extraction for the neural networks

The character recognition system is composed of six neurons to separately recognize the features of the six segments. Figure 15 shows how the summations of pixel values are taken from one segment in order to feed a neural network.

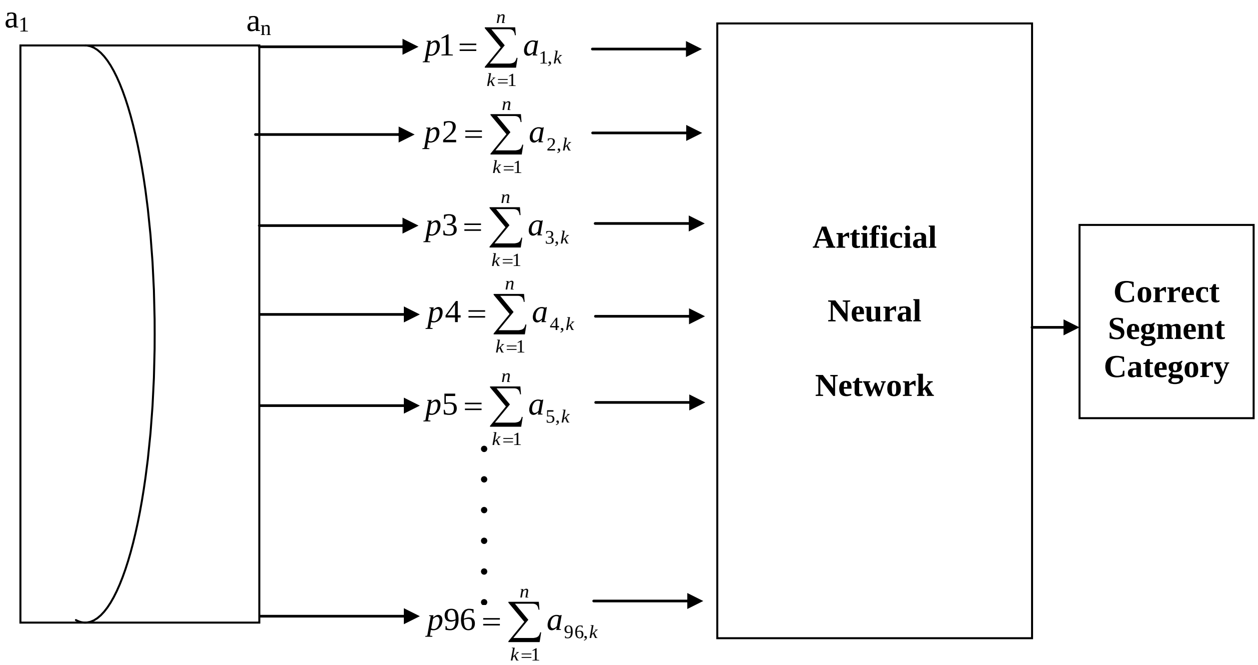


Fig.15. Taking inputs from one segment to the neural network

Vertical segments have 96 rows while horizontal segments have 96 columns. Therefore, in order to feed the inputs to the neural networks, the horizontal segments are rotated and made vertical. Now each segment will provide 96 inputs to the neural network as depicted in figure 16.

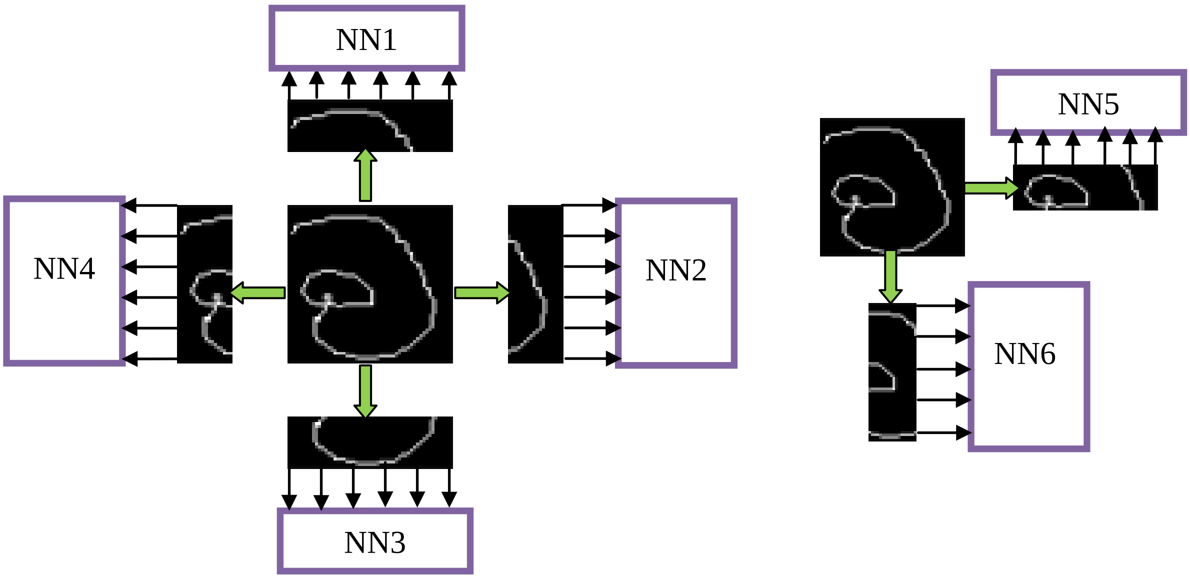


Fig.16. Taking inputs from the segments to the corresponding neural networks

The training process of the neural networks is carried out for the training set of 200 instances in which 10 different characters were tested with 20 instances per each character. Before the training process the output of each segment based neural network was identified through a categorization based on the corresponding character shapes. Figure 17 depicts the output identification for NN3 which is based on segment 1. The categories were identified using the plots generated for each segment as shown in figure 23.

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3	C4 OUTPUT=4
			

Fig.17. Character categorization based on segment 1

Figure 18 shows the identification of the output of NN2 based on the feature categorization of characters based on segment 2.

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3
		

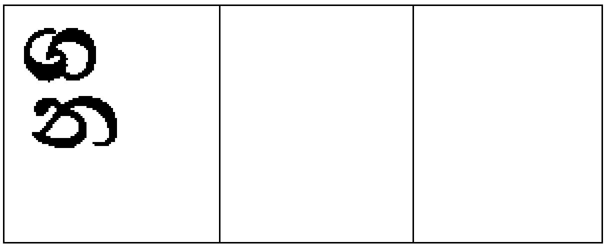


Fig.18. Character categorization based on segment 2

As depicted in figure 19, the shapes of characters have resulted more categories based on segment 3.

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3	C4 OUTPUT=4	C5 OUTPUT=5	C6 OUTPUT=6

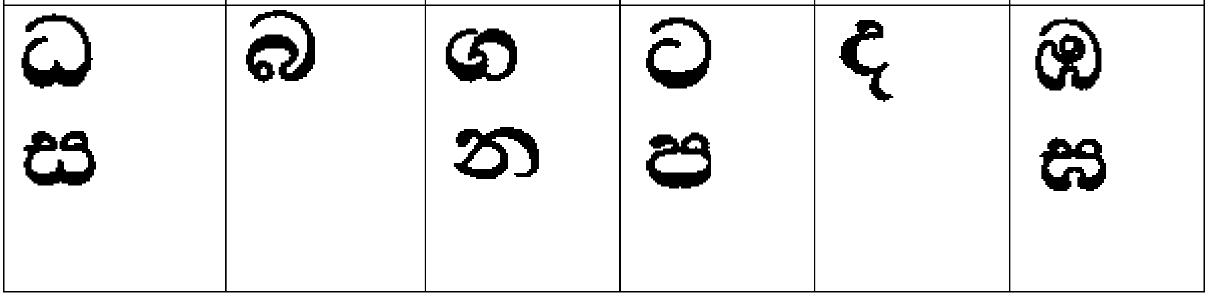


Fig.19. Character categorization based on segment 3

Figure 20 depicts the set of categories resulted for segment 4. Likewise the characters are categorized in to different groups under segment 5 and 6. They are shown in figure 21 and 22 respectively.

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3

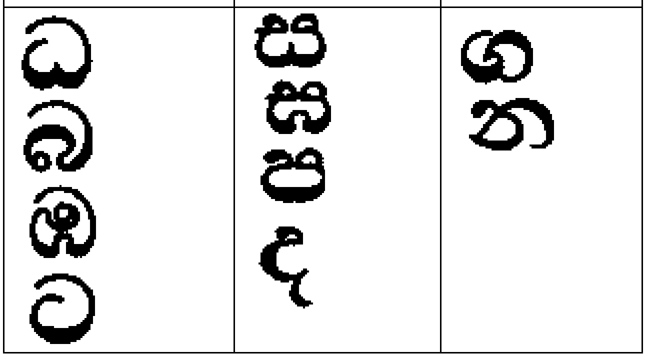


Fig.20. Character categorization based on segment 4

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3	C4 OUTPUT=4	C5 OUTPUT=5	C6 OUTPUT=6

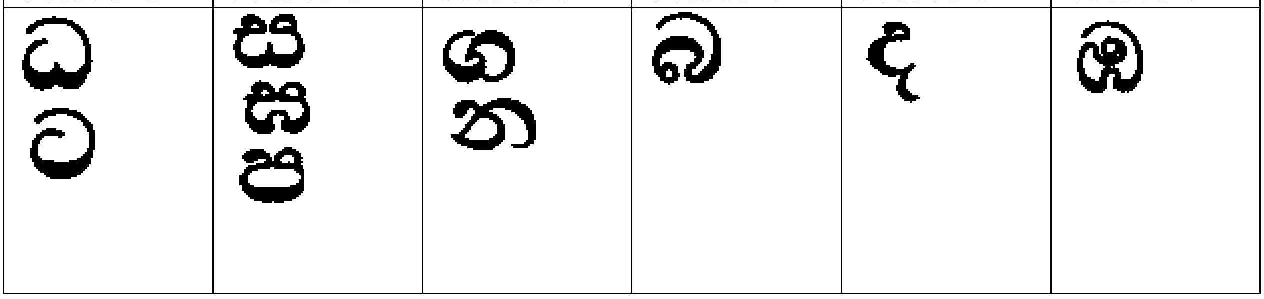
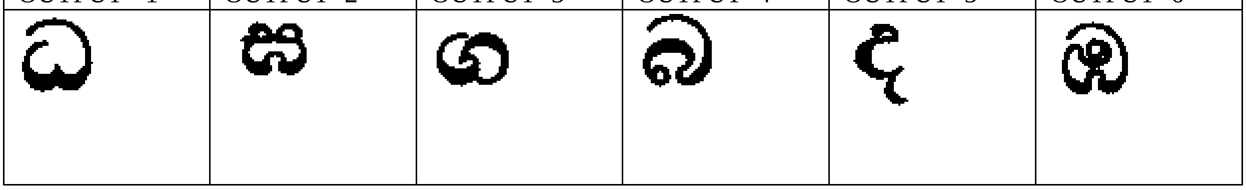


Fig.21. Character categorization based on segment 5

C1 OUTPUT= 1	C2 OUTPUT=2	C3 OUTPUT=3	C4 OUTPUT=4	C5 OUTPUT=5	C6 OUTPUT=6



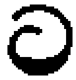
C7 OUTPUT=7	C8 OUTPUT=8	C9 OUTPUT=9	C10 OUTPUT=10		
					

Fig.22. Character categorization based on segment 6

The differences of the segments under each segment type was plotted and assured to have correct categorisation as depicted in figure 23. Figure 23 shows the feature plot of all the ten characters under segment 3. As figure 23 shows, 6 different categories can be observed using the plot.

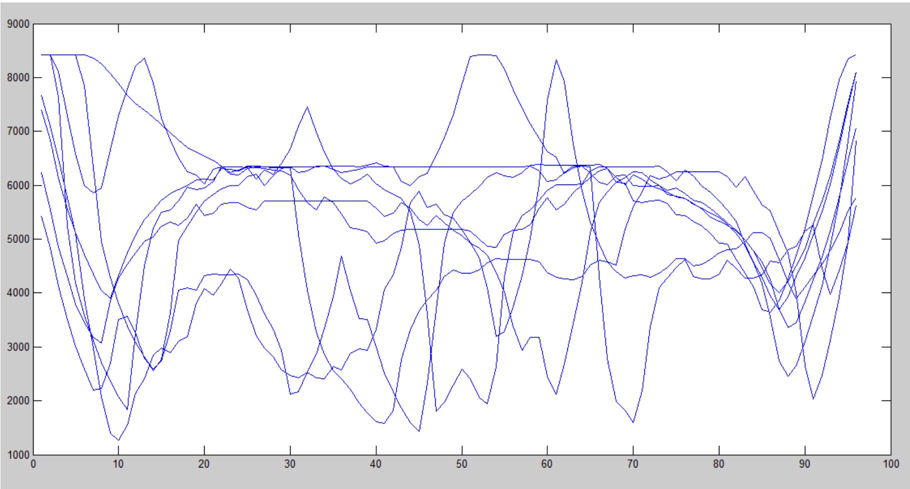


Fig.23 Feature plot of inputs to neural network (NN3) of segment 3.

K. Neural Network System of Category Identification

All the 6 neural networks which correspond to the 6 segments were first created with the same architecture in which there were 3 layers with the first layer of 5 tan-sigmoid neurons, second layer of 10 log-sigmoid neurons, and a third layer of 1 linear neuron which is depicted in figure 24. When the training process started, the architectures of the neural networks were changed to increase the performance of the network.

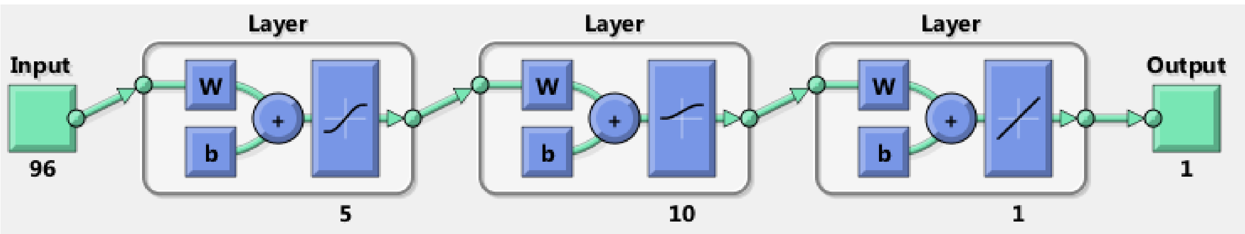


Fig.24. Architecture of the common network created before the training process

All the neural networks were trained with an error goal of 0.05. The changed neural network which corresponds to segment one, NN1, is composed of four layers in which the first layer is composed of 10 tan-sigmoid neurons, second layer is composed of 10 log-sigmoid neurons, third layer is composed of 5 tan –sigmoid neurons and the output layer is composed of 1 linear neuron as depicted in figure 25.

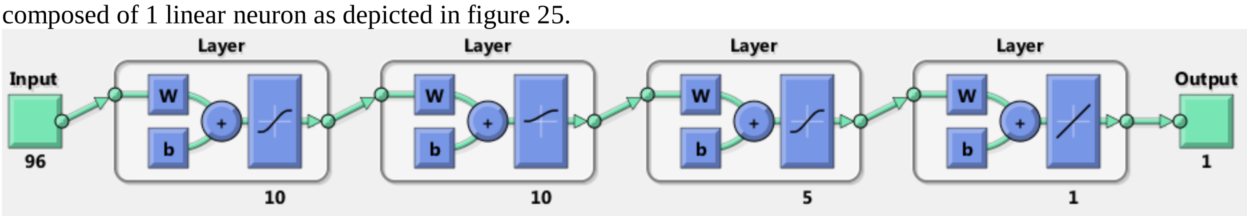


Fig.25. Architecture of the common network created before the training process

Likewise during the training process the architectures of NN3, NN4, NN5 and NN6 were changed to obtain a high accuracy as shown in figures 26, 27,28 and 29 respectively.

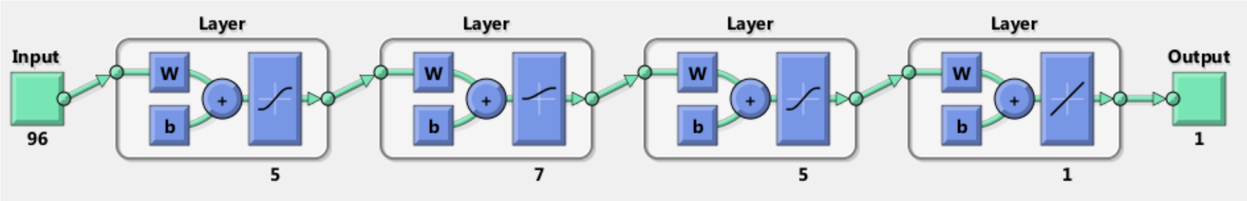


Fig.26. Architecture of NN3 which was subjected to modification during the training process

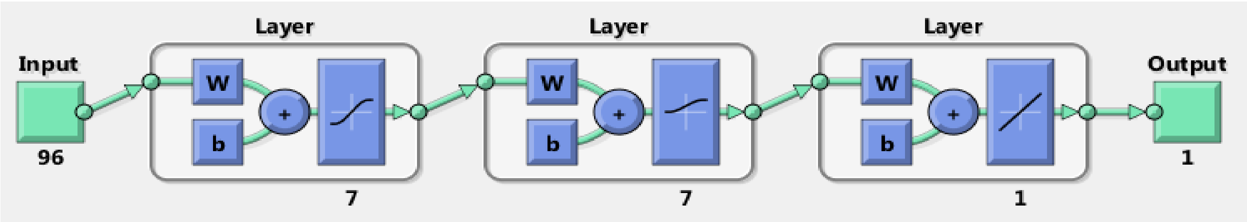


Fig.27. Architecture of NN4 which was subjected to modification during the training process

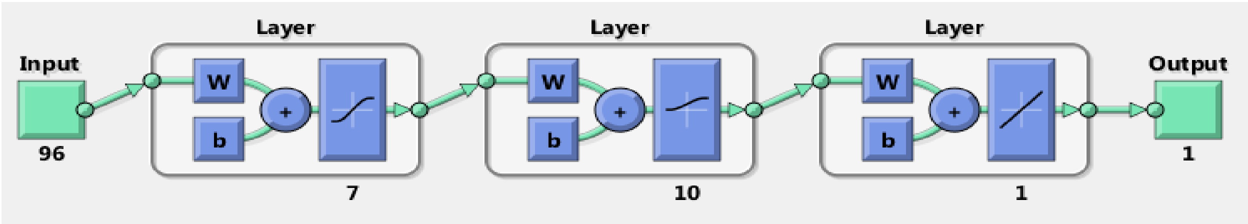


Fig.28. Architecture of NN5 which was subjected to modification during the training process

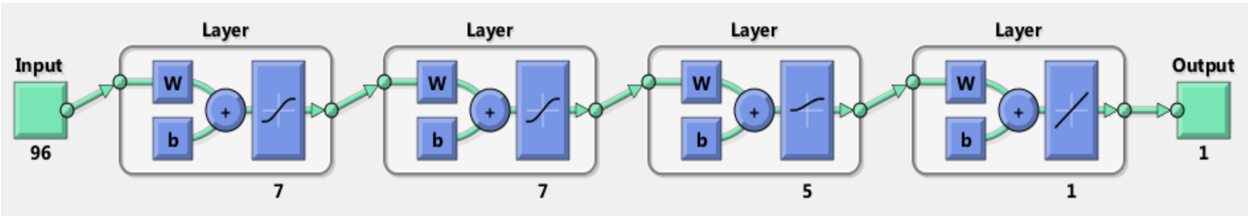


Fig.29. Architecture of NN6 which was subjected to modification during the training process

L. Fuzzy Logic System

The outputs taken from each neural network which correspond to different segments were then introduced as crisp inputs to 6 fuzzy input membership functions. The number of membership functions available in one fuzzy input was equal to the number of categories identified in each of its corresponding segment. Then the output generated from the neural network (NN1) which corresponds to segment one was fed into the corresponding fuzzy input which is composed of four membership functions. Likewise fuzzy input 2 is composed of three membership functions since only 3 categories are there under segment 2. Figure 30 shows the fuzzy inputs which correspond to each neural network. Then the fuzzy inference system was introduced with a set of rules generated by observing the segment categorizations.

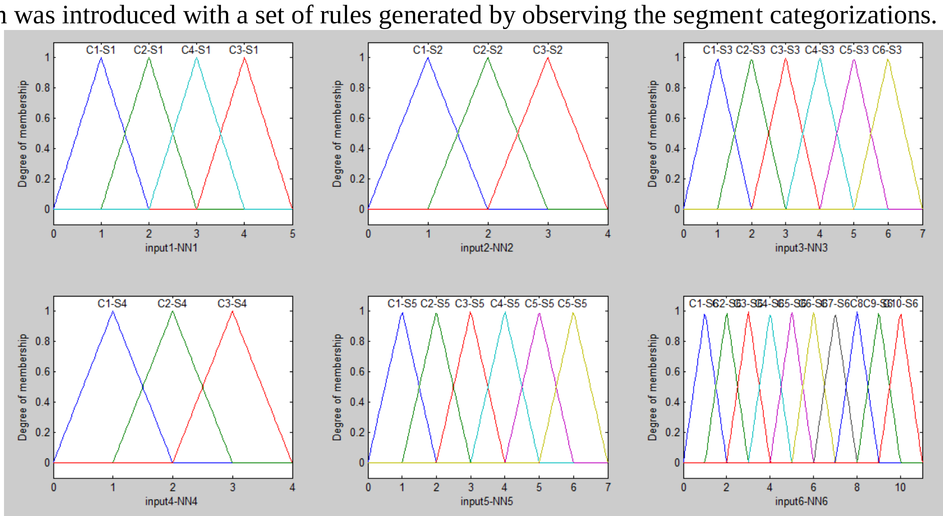


Fig.30. Six fuzzy inputs which correspond to the 6 neural networks

Figure 31 depicts few rules which were used in creating the fuzzy inference system.

If (input1-NN1 is C2-S1) and (input2-NN2 is C2-S2) and (input3-NN3 is C4-S3) and (input4-NN4 is C2-S4) and (input5-NN5 is C2-S5) and (input6-NN6 is C8-S6) then (output1 is PA) (1)
If (input1-NN1 is C1-S1) and (input2-NN2 is C1-S2) and (input3-NN3 is C4-S3) and (input4-NN4 is C1-S4) and (input5-NN5 is C1-S5) and (input6-NN6 is C9-S6) then (output1 is TA) (1)
If (input1-NN1 is C3-S1) and (input2-NN2 is C1-S2) and (input3-NN3 is C3-S3) and (input4-NN4 is C3-S4) and (input5-NN5 is C3-S5) and (input6-NN6 is C10-S6) then (output1 is NA) (1)
If (input1-NN1 is C3-S1) and (input2-NN2 is C1-S2) and (input3-NN3 is C3-S3) and (input4-NN4 is C3-S4) and (input5-NN5 is C3-S5) and (input6-NN6 is C3-S6) then (output1 is GA) (1)

Fig.31. Rules of the fuzzy inference system

The fuzzy output of the Fuzzy inference system is composed of 10 membership functions for the ten characters used in the testing process. These are illustrated in figure 32.

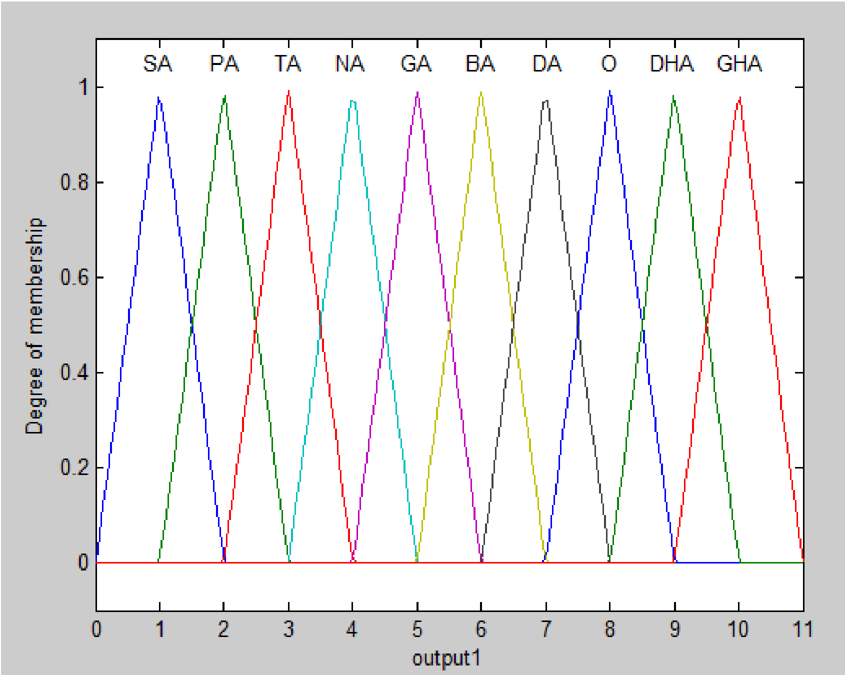


FIG.32. FUZZY OUTPUT OF THE FUZZY INFERENCE SYSTEM

III. RESULTS AND DISCUSSION

To test the FIS, 200 instances of the same test characters with 20 instances from each, were taken. Table 32 shows the average accuracies obtained in each situation.

Table 1. The accuracies generated for the different characters

Letter	ස	ප	ට	න	ග	බ	ද	ඹ	ධ	ඝ
Correct Instances										
Incorrect Instances	2	2	0	6	6	5	4	8	0	5
Accuracy	90%	90%	100%	70%	70%	75%	80%	60%	100%	75%

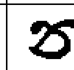
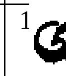
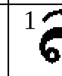
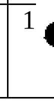
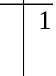
According to the results shown in table 1, the proposed method has produced an average accuracy of 81%. But the system has showed this accuracy only for the 10 characters which were used under the study. Therefore, the system's accuracy should yet to be proved for the whole Sinhala alphabet which is made of 18 vowels, 41 consonants and 17 modifier symbols. The hand written characters used in this process were clear and less skewed. Therefore, the system has produced a good accuracy close to 81%. Though the system produces an accuracy of 81%, it shows less accuracies in recognizing complex characters such as න (NA), ඹ (MBA). These are some problems generated due to overfitting of the neural network and the less amount of rule availability of the FIS. Therefore, the generalization of the neural network could be improved by using techniques like regularization and early stopping during the training process of the neural networks. The number of rules of the fuzzy rule base could be increased by finding the category intersections available in the character.

The neural networks were trained with an error goal of 0.05. Therefore, the accuracy which each of the neural networks has tried in achieving is around 95%. By decreasing the error goal down to a value less than 0.05, the performance of each of the neural networks could be improved.

The average time elapsed to recognize each character is provided in table 2. When the complexity of the character is high, the algorithm has taken more time to recognize the character. The time taken to recognize each character seems to be high. The process of segmentation might have caused the system to take more time in recognizing a particular character.

One problem available is the amount of processing needed when more different characters are introduced and it becomes high since the algorithm depends totally on the characteristics of a particular character. For example, if more characters are introduced to the training process, it will increase the number of categories available within a particular segment of the character. Therefore, the training complexity will be increased making the training process more complex. In such a situation, more changes will have to be done to the architectures of the six neural networks used in the system. Also, the increased amount of categories will result in increased amount of fuzzy rules with an increased number of antecedents in one rule. This may cause the FIS to have a less effective decision surface introducing more flat areas having no outcome at all.

Table 2. Average Time elapsed to recognize characters.

Character									
Time elapsed in seconds	[illegible] 	[illegible] 	1 [illegible] 		1 [illegible] 	1 [illegible] 	1 [illegible] 	[illegible] 	1 [illegible] 

IV. CONCLUSION

A system based on fuzzy logic and neural networks was proposed for identifying hand written Sinhala characters. The system provides a generalized method for this purpose with an average accuracy of 81%. However, the system's complexity increased when more different Sinhala characters were introduced to the FIS. The system seems to consume a decent amount of time in recognizing one character. The bizarreness available in hand written characters may decrease the accuracy of the proposed system. The segmentation procedure is the one that consumes 80% of the total time consumed. This could be reduced by introducing a different segmentation procedure or removing the segmentation to identify different characteristics of the Sinhala characters. So the proposed method is still open for improvements.

REFERENCES

[1] MUKHERJI P. AND REGE P.P. (2009) SHAPE FEATURE AND FUZZY LOGIC BASED OFFLINE DEVNAGARI HANDWRITTEN OPTICAL CHARACTER RECOGNITION. JOURNAL OF PATTERN RECOGNITION RESEARCH 4 (2009) 52-68.

[2] Premaratne H.L. and Bigun J. (2002) Recognition of Printed Sinhala Characters Using Linear Symmetry. The 5th Asian Conference on Computer Vision (2002), 23-25

[3] Aleksander, Igore & Morton, Helen (1991) : "An Introduction to Neural Computing", Chapman & Hal, ISBN 0 412 37780 2.

[4] Rajapakse R.K., Weerasinghe A.R and Seneviratne E.K.. (1996) A neural network based character recognition system for Sinhala script. The South East Asian Regional Computer Confederation, Conference and Cyber exhibition (SEARCC'96) held in Bangkok, Thailand, July 4-7th 1996.

[5] Kodituwakku S.R. and Nilanthi P.S. (2010) Investigation a fuzzy approach for handwritten Sinhala character recognition. International Journal of Engineering Science and Technology. Vol. S (11), 2010, 6031-6034.

[6] Hewavitharana S., Fernando H. C., Kodikara N. D. Offline Sinhala Handwriting Recognition using Hidden Markov Models.

[7] Rabiner L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Prox. IEEE, vol. 77, no. 2, 1987, pp. 257-285.

[8] Bunke H., Roth M., And Talamazzini E., Offline Cursive Handwritten Recognition using Hidden Markov Models, Pattern Recognition, vol. 28, no. 9, 1995,pp. 1399-1413.

[9] Kim H. J., Kim K. H., Kim S. K., And Lee J. K. Online Recognition of Handwritten Chinese Characters based on Hidden Markov Models, Pattern Recognition, vol. 30, no. 9, 1997, pp. 1489-1500.

[10] Jayasekara B., And Udawatta L. Non-Cursive Sinhala Handwritten Script Recognition: A Genetic Algorithm Based Alphabet Training Approach. Proceedings of the International Conference on Information and Automation, December 15-18, 2005, Colombo, Sri Lanka.

[11] S.Rajasekaran, G.A. Vijayalakshmi Pai,"Neural Networks,Fuzzy Logic, and Genetic Algorithms",Prentice-Hall,New Delhi,2003.

[12] Kodituwakku S. R. and Rambukwella A.I, An enhanced algorithm for offline connected handwritten character segmentation, International Journal of Latest trends in Computing (IJLTC), 2011, Vol.2 (1), pp: 60 - 66.

M. A. P. Chamikara is a Lecturer in Computer Science at the Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya, Sri Lanka. He received his BSc (Special) degree in Computer Science from the University of Peradeniya, Sri Lanka (2010). His research interests include Crime Analysis, GIS (Geographic Information Systems), Image Processing, Computer Vision and Artificial Intelligence.

S. R. Kodituwakku is a professor in Computer Science at the Department of Statistics and Computer Science, University of Peradeniya, Sri Lanka. His research interests include Database Systems, Distributed Computing, Role Based Access Control Systems, and Software Engineering.

A.A.C.A. Jayathilake is a Temporary Lecturer in Mathematics at the Department of Mathematic, Faculty of Science, University of Peradeniya, Sri Lanka. She received her BSc (Special) degree in Mathematics from the University of Peradeniya, Sri Lanka (2013). Her research interests include Combinatorics, Graph Theory, Fuzzy Mathematics, Computational Mathematics, Abstract Algebra and Group Theory.

K. R. Wijeweera is a postgraduate student at the Postgraduate Institute of Science, University of Peradeniya, Sri Lanka. His research interests include Computer Graphics, Image Processing, Computer Vision, and Artificial Intelligence.

