

Simulation of Zhang Suen Algorithm using Feed-Forward Neural Networks

Ritika Luthra
Research Scholar
Chandigarh University

Gulshan Goyal
Associate Professor
Chandigarh University

ABSTRACT

Image Skeletonization plays a very crucial role in image processing as it has been used in various applications such as pattern recognition, fingerprint analysis, Signature verification etc. Image skeletonization process generates unit pixel width skeletons. Present paper considers feed forward neural network approach for simulation of Zhang-Suen algorithm. Network parameters are chosen based on experimentation. Values of MSE, PSNR, and Execution time are calculated for Gurumukhi characters. Performance graphs have been plotted.

Keywords

Feed-forward, MSE, Neural Networks, OCR, PSNR, Skeletonization, Zhang and Suen

1. INTRODUCTION

Skeletonization is the process of extracting skeletons by deleting unwanted pixels from an image. It is morphological operation that deletes black foreground pixels iteratively layer by layer until one pixel width skeleton is obtained. It is a procedure of reducing an object to its minimum size so that it can be used in various image processing applications. Skeletonization is usually applied on binary images which consist of black (foreground) and white (background) pixels. It takes input to be a binary image, and produces another binary image as output as shown in fig1.

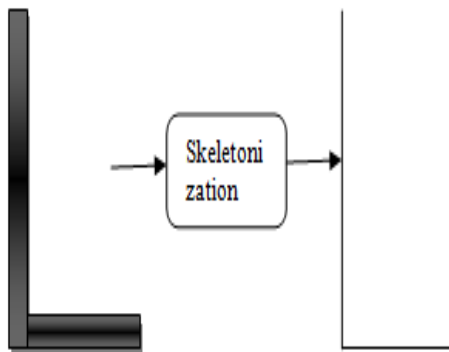


Fig 1: General concept of Skeletonization

For a skeletonization algorithm to be effective, it should reduce the images into thin like objects and should retain the topological and geometric properties as well. However, a good skeletonization algorithm must have the following features:

1. The resulting skeletons should maintain connectivity.
2. The resulting skeletons should be of unit pixel width.
3. No excessive deletion of pixels should takes place.

4. It should be efficient in terms of execution time.
5. It should ideally compress the data.

1.1 Need of Skeletonization

Skeletonization is a significant step in many image processing applications such as Pattern Recognition [2], Optical character recognition [1], and fingerprint classification [3] etc. Therefore it is an active area of research.

Skeletonization is a significant step in many image processing applications such as Pattern recognition [2], Optical character recognition [1], and fingerprint classification [3] etc. Therefore, it is an active area of research.

So, there is always a need for good skeletonization algorithms in reference to following parameters:

1. The amount of data required to be processed is reduced as it takes less time [3].
2. To reduce processing time.
3. Extraction of important characteristics such as critical ending-points, joining-points, etc. can be helpful in many applications [3].
4. By reducing an object to only a skeleton, unimportant features and image noise can be filtered out.
5. Skeletonization is commonly used for the higher degree analysis and recognition for applications such as diagram understanding, OCR, feature detection, and fingerprint analysis.

1.2 Applications of Skeletonization

Skeletonization has been used for variety of image processing applications like:

1. Optical character recognition (OCR) [1,4]
2. Pattern recognition[2]
3. Fingerprint classification[3]
4. Biometric authentication[4]
5. Signature verification[4]
6. Medical Imaging [3]

2. PROBLEM BACKGROUND

Skeletonization is a way to produce skeletons with very limited amount of information. But in this process sometimes excessive erosion takes place. Generally, a good and an effective algorithm must meet the following requirements: sensitivity to noise, preservation of geometrical and topological properties, no excessive erosion, efficient in terms of execution time, should maintain connectivity,

Reconstructability etc. Till now, there exist a huge number of skeletonization algorithms by different authors. Due to the complicated nature of skeletonization, it is very difficult to understand that how the approaches are related to each other in terms of the processing time and quality of the skeletonized images. Each of the algorithms has their own advantages and disadvantages.

Zhang and Suen [14] is one of the existing algorithms that have been used in this paper because it has been used as a comparison for skeletonization methods for so many years as it is very fast and simple to implement. But, the algorithm has some disadvantages. Skeletons obtained are not unit pixel width and there is loss of patterns sometimes. So there is need of an algorithm which overcomes these disadvantages.

Present paper proposes feed forward neural network approach for simulation of Zhang and Suen algorithm. This algorithm produces skeletonized images in two sub-iterations.

Sub-iteration 1:

- (1) Number of non zero pixels should be between two and six. $2 <= B(P1) <= 6$
- (2) Number of 0-1 transitions should be equals to one. $A(P1) = 1$
- (3) At least one of P2 and P4 and P6 is white. $P2 \times P4 \times P6 = 0$
- (4) At least one of P4 and P6 and P8 is white. $P4 \times P6 \times P8 = 0$ [14]

Sub-iteration 2:

- (1)) Number of non zero pixels should be between two and six. $2 <= B(P1) <= 6$
- (2) Number of 0-1 transitions should be equals to one. $A(P1) = 1$
- (3) At least one of P2 and P4 and P8 is white. $P2 \times P4 \times P8 = 0$
- (4) At least one of P2 and P6 and P8 is white. $P2 \times P6 \times P8 = 0$

After checking all the conditions, the pixel is deleted otherwise not [14].

3. INTRODUCTION TO NEURAL NETWORKS

Neural networks [13] functions in a same way as our human brain does. It contains a very large number of highly interconnected elements known as neurons. These all neurons work together and process to solve some special problems. Neural networks always learn by examples. The neural networks examples must be selected with much care otherwise useful and effective time is wasted in training the neural network or even worse the network might not be functioning properly. A simple neuron is shown below in Fig. 2.

There are different uses of neural networks [13]:

(a) Classification [13]: Classification problem in neural networks is defined as when we have to classify data or objects according to some given criteria. Example: Pattern recognition [2] problems. Type of networks that can be used in doing classification is feed-forward neural networks.

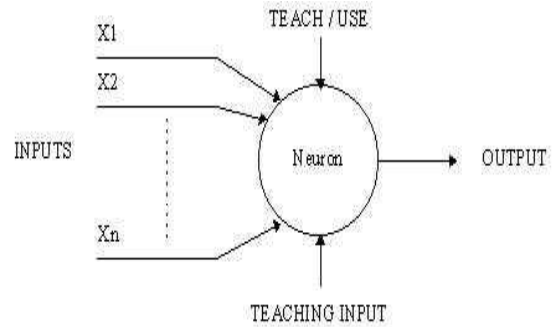


Fig 2: A simple neuron [5]

(b) Prediction: if a neural network is given some inputs, it can be trained to produce outputs from the same. One can use the neural network to predict future values if one have some sort of network that fits properly in a known sequence of values. Type of networks that can be used to predict values are feed-forward neural networks.

(c) Clustering: In some situations, we have some data that is so complex that there is no way to classify them into different categories. Neural networks have extraordinary capabilities to identify some important features from these particular types of data and help them to classify the data into different categories. This technique is helpful in data mining. Types of networks used for clustering can be Adaptive Resonance Theory (ART) [9], Self-organizing maps (SOM) [9].

(d) Association: A neural network can be trained to learn a large number of patterns or objects, when a misshape object is given to the neural network as an input, then it associates it with the nearest one stored in its memory and returns the desired output of that particular object. Type of neural networks that can be used in association can be Hopfield network.

4. APPLICATIONS OF NEURAL NETWORKS

Neural networks are an active area of research. It can be used in wide variety of applications such as:

1. Character Recognition [8]
2. In medicine [3]
3. Signature verification [4]
4. Forecasting [4]
5. Skeletonization [7, 8]

5. ROLES OF NEURAL NETWORKS IN SKELETONIZATION

- a) Neural networks has extraordinary abilities to extract the meaning from any sort of complicated data that can be used to derive some objects/patterns that humans and other computer based techniques are not able to detect.
- b) A trained neural network can be called as an "expert" for analyzing and processing any kind of information. [6]
- c) Thinning problem requires two major tasks to be implemented: (a) removing the unwanted thick pixels (b) stopping the thinning process when the skeletons are very close to unit pixel width [6].
- d) The first task can be done easily. The main problem arises in the second part, because the stopping decision

has to be done automatically. This can be achieved using a real time cellular neural network by training the neural network.

- e) Most of the conventional thinning approaches suffer from noise sensitivity and rotation dependency. With the use of neural networks we can perform thinning invariant under arbitrary rotations. [6]

Neural network takes a different approach as compare to conventional skeletonization. Conventional skeletonization is based on algorithmic approach i.e. computer follows a specific set of conditions/instructions so as to delete the unwanted pixels. The proper steps need to be given to delete the necessary pixels. This thing bound the problem solving capabilities in conventional skeletonization approach. But a neural network function in the same way as human brain does. Neural networks always learn by examples. They reduce the number of instructions to be executed. Hence they take less execution time and faster than the conventional thinning approaches.

6. FEED-FORWARD NEURAL NETWORKS

Feed forward networks are the most popular types of neural networks used. A feed forward network is composed of large number of processing units called as neurons that are organized in many layers. In feed forward neural networks every layer has a connection to its next layer but there are no connections to the backward layer. For example: “Input layer has a connection with hidden layer”. Each connection has some weight assigned to it and even some bias value can be there. The units in a neural network are probably called as nodes. A simple feed-forward network is as shown below in the fig. 3.

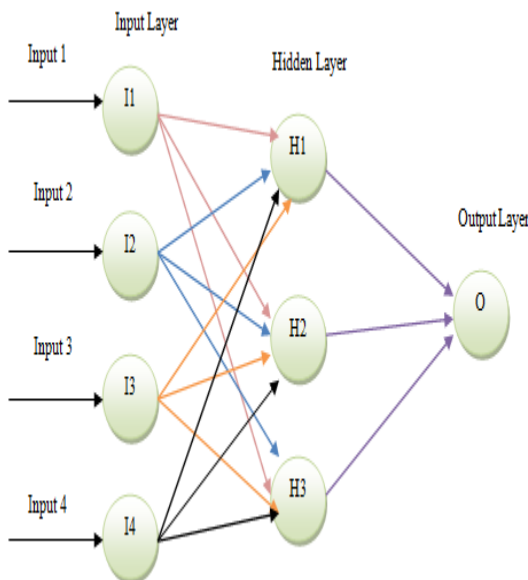


Fig 3: A simple feed-forward neural network

Feed forward neural networks can be trained by using some set of patterns called as the training set for which the desirable outputs are already known. This type of learning is known as supervised learning.

There are some conditions necessary for operating a network successfully. Some of them are described below:

(a) The number of hidden nodes: If there are less hidden nodes used, network will not be able to learn any given pattern. So the number of hidden nodes in feed forward networks is chosen to be one.

(b) The number of training iterations/epochs: If there is less iteration used to train the network, the network will not be able to process the information. So number of training iterations should be chosen carefully.

There are numerous other types of neural networks available. In this paper, results of skeletonization are produced by using the feed forward networks and newpr function for pattern recognition.

7. PROPOSED FRAMEWORK

Present paper uses Zhang and Suen algorithm to produce the skeletons and uses input and output images to train the network using feed-forward approach. Proposed algorithm is described below in the fig 4. Proposed Methodology is as shown in the fig 5.

<p>Input Data: Input and Target Image Output Data: Skeletonized image</p> <p>PROCEDURE</p> <p>Step 1: Obtain a set of input images and its corresponding output images</p> <p>Step 2: Determine Neural network Architecture</p> <ul style="list-style-type: none"> a) Number of layers b) Number of input, hidden and output layers <p>Step 3: Determine neural network parameters</p> <ul style="list-style-type: none"> a) Number of input nodes b) Number of output nodes c) Number of hidden nodes d) Learning Rate and momentum e) Activation Function f) Maximum validation checks g) Weights and Biases <p>Step 4: Start training using feed forward approach and propagate through the layers from the input to hidden and then from hidden to output layer.</p> <p>Step 5: Simulate the network.</p> <p>Step 6: if skeletonized Display skeletonized image else Train neural network again.</p>

Fig 4: Proposed Algorithm

8. IMPLEMENTATION

(a) Dataset used: Dataset is composed of 41 different Gurumukhi characters of 65*95 sized images.

(b) Framework of the proposed skeletonization algorithm is discussed in section 7.

(c) Network Training Parameters:
 By experimentation on various images, these network training parameters have been taken.

- i. Number of neurons used in the hidden layer=20
- ii. Learning function=trainidx(adaptive learning function) and logsig (logarithmic sigmoid)
- iii. Cost/Performance function= SSE(Sum Squared Error)
- iv. Learning rate=0.75

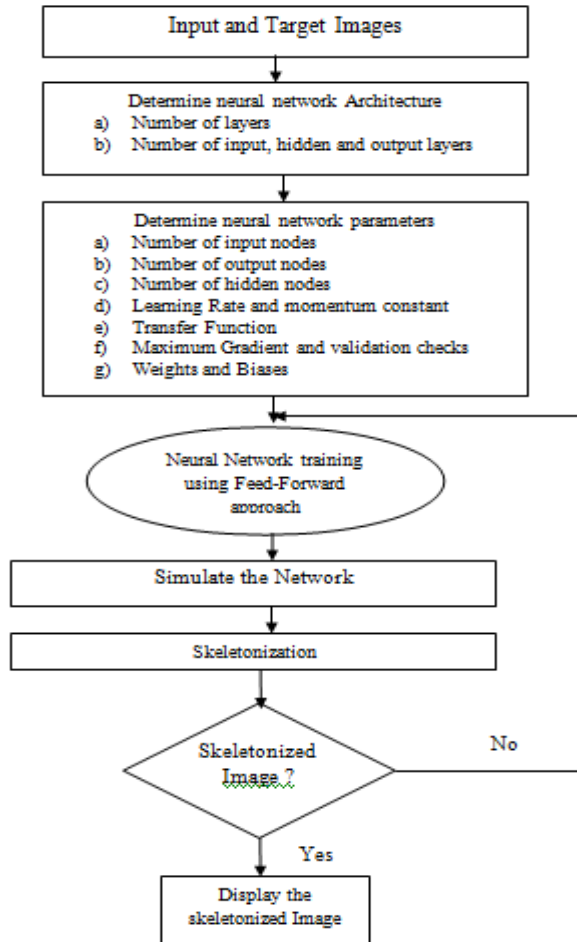


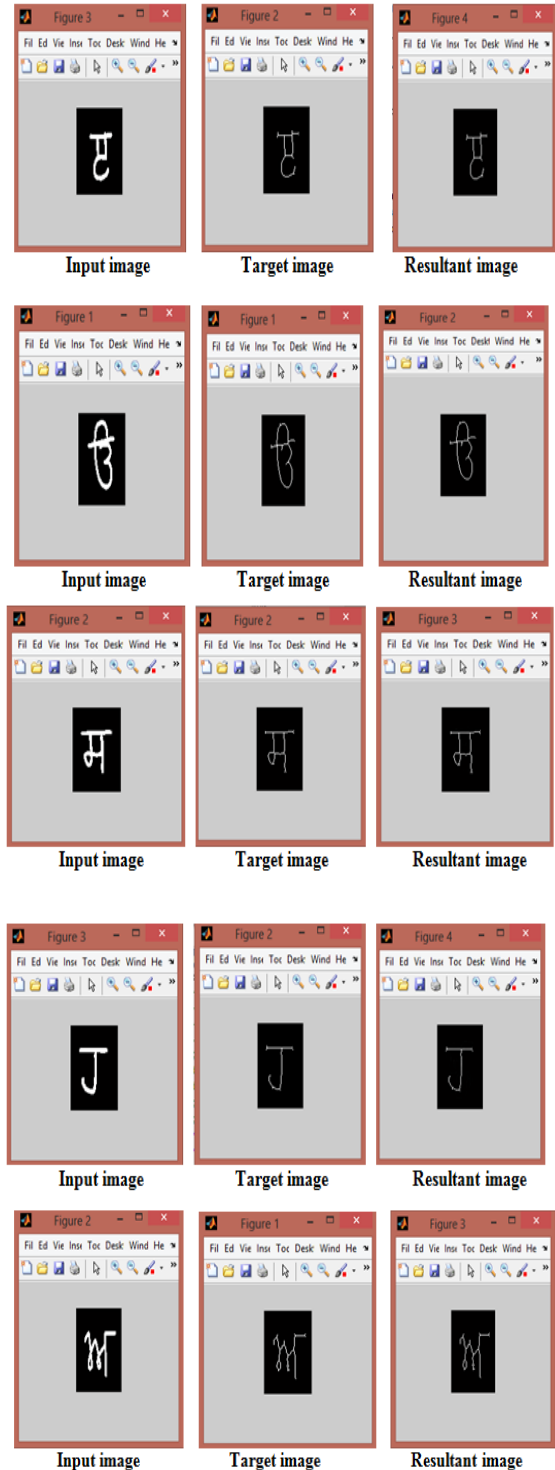
Fig 5: Proposed Methodology

- v. Type of neural network used=newff (feed forward networks)
- vi. Neural network function=newpr(for pattern recognition)
- vii. Number of epochs=200
- viii. Momentum constant=0.95
- ix. Performance goal=0.10
- x. Input image=95x65 size image
- xi. Target image=95x65 skeletonized image
- xii. Type of learning used=supervised learning

9. RESULTS AND DISCUSSIONS

Neural network is trained by using 41 different images of size 65x95 each as shown in figure. Neural network is trained by using some of the network training parameters mentioned above. Training process of the network is shown below in

figure 6. After training the neural network we get the desired outputs. Epochs and time constraints put an upper bound on the training iterations. Goal constraint stops the training when the performance drops below it. If the performance falls below the minimum gradient value then training stops as shown in figure 4. Maximum validation checks (max_fail) function parameter is a strictly positive integer. Zero validation checks indicate that the neural network learns at every iteration as shown in fig 7.



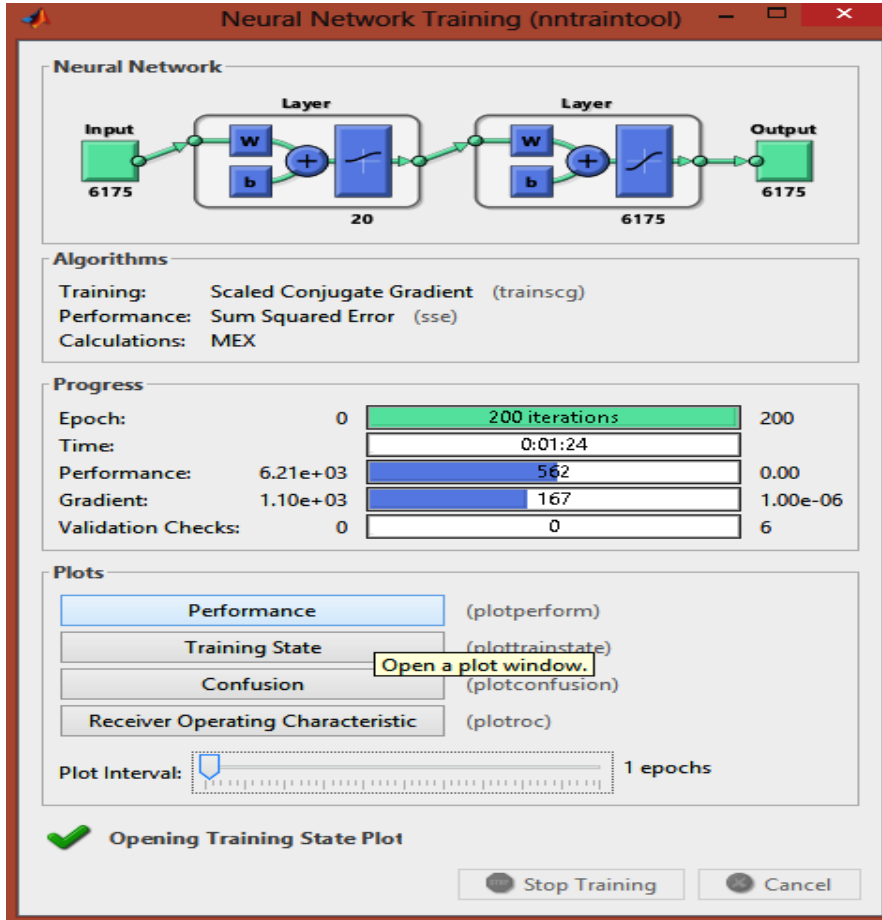


Fig 6: Training process of the network

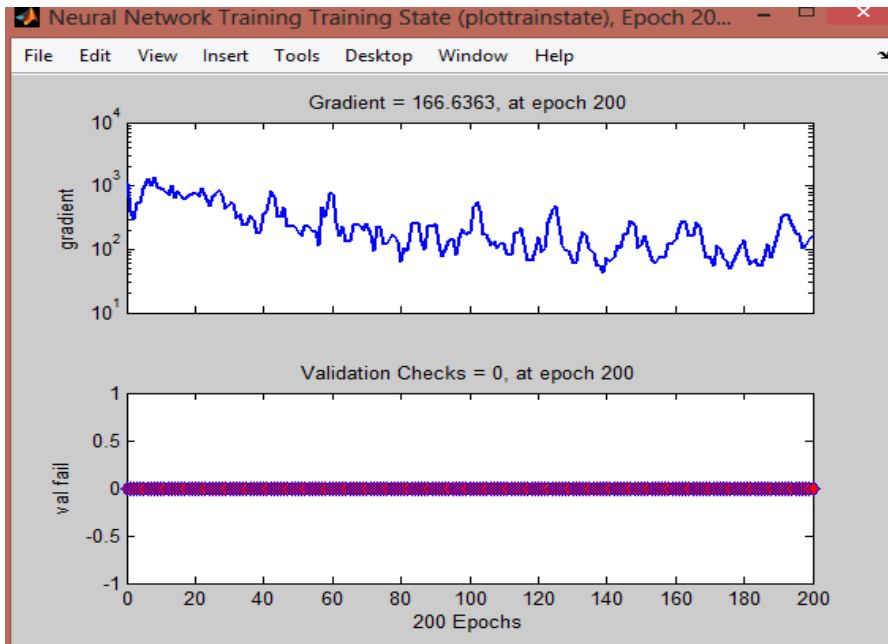


Fig 7: Neural network training state

Table 1 Performance Evaluation for skeletonization with feed forward neural networks and ZS algorithm

Image	PSNR		MSE		Execution Time	
	Zhang and Suen	Skeletonization with NN	Zhang and Suen	Skeletonization with NN	Zhang and Suen	Skeletonization with NN
ੳ	33.90	58.52	33.25	0.09	0.90	0.62
ਅ	32.85	59.45	33.42	0.07	0.91	0.65
ੲ	32.95	59.87	34.91	0.07	0.74	0.43
ਮ	32.94	59.45	32.61	0.07	0.81	0.36
ੲ	32.90	60.53	32.96	0.06	0.68	0.42
ੲ	32.98	59.26	32.98	0.08	0.81	0.36
ਖ	32.96	59.60	32.98	0.07	0.79	0.35
ਗ	33.07	59.73	32.83	0.07	0.76	0.68
ਘ	32.90	59.42	33.19	0.07	0.79	0.71
ਙ	32.90	58.85	32.95	0.09	0.74	0.77
Average	33.03	59.46	33.20	0.074	0.79	0.53

Table 1 above describes the performance evaluation parameters such as Peak signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Execution Time of both the Zhang and Suen algorithm and proposed approach for skeletonization using neural networks. The proposed algorithm is tested on the Gurmukhi characters and following values are obtained.

10. CONCLUSIONS AND FUTURE SCOPE

Present paper is based on the comparison of existing technique Zhang and Suen algorithm and the new proposed technique. New proposed technique is based on simulation of Zhang and Suen algorithm by using feed forward neural networks approach so as to enhance in values of parameters. Experimental results prove that the proposed technique using feed forward neural networks is better in terms of PSNR, MSE in comparison to the existing technique. Results also show that by using neural networks the algorithm takes less execution time for performing skeletonization.

Future work will be related to the use of back-propagation neural networks and other neural networks too for skeletonization of Gurmukhi characters. Conditions for noise robustness and unit pixel width can be included in training patterns. The results can be tested for the other datasets as well.

11. REFERENCES

- [1] Abu-Ain W., Bataineh B., Abu-Ain T. and OmarK., "Skeletonization Algorithm for Binary Images", Fourth International Conference on Electrical Engineering and Informatics (ICEEI) Elsevier, Vol. 11, pp.704-709, 2013.
- [2] Padole G.V. and Pokle S. B., "New Iterative Algorithms for Thinning Binary Images" IEEE Third International Conference on Emerging Trends in Engineering and Technology, Vol. 7, pp. 166-171, 2010.
- [3] Lam L, Lee S.W. and Suen C.Y., "Thinning methodologies-A comprehensive survey", IEEE transactions on pattern analysis and machine intelligence, Vol. 14, No. 9, pp. 869-885, 1992.
- [4] Chatbri H. and Kameyama K., "Using Scale Space Filtering to Make Thinning Algorithms Robust Against Noise in Sketch Images", International Conference on Pattern Recognition letters Elsevier, Vol. 42, pp. 1-10, 2014.
- [5] Sharma V. et al. "A comprehensive study of artificial neural networks" International journal of advanced research in computer science and computer engineering Volume 2, Issue 10 pp. 278-284 2012
- [6] Matsumoto T., Chua L.O. and Yokohama T., "Image Thinning with a Cellular Neural Network", IEEE Journal



- on Circuits and Systems, Vol. 37, Issue 5, pp. 638-640, 1990.
- [7] Datta A. et al “Shape Extraction: A Comparative Study Between Neural Network-Based and Conventional Techniques” *International Journal of Neural Computing & Applications* Springer pp. 343-355 1998
- [8] Ahmed P., “A Neural Network based Dedicated Thinning Method”, *International Journal of Pattern Recognition Letters Elsevier* ,Vol. 16, Issue 6, pp. 585-590, 1995.
- [9] Altuwaijri M. and Bayoumi A., “A Thinning Algorithm for Arabic Characters Using ART2 Neural Network”, *IEEE Transactions on Circuits and Systems-II Analog and Signal Processing*, Vol. 45, No. 2, pp. 260-264, 1998.
- [10] Shang L. and Yi Z., “A Class of Binary Images using Two PCNNs”, *International Conference on Intelligent Computing*, Elsevier, Vol. 70, Issue4-6, pp. 1096-1101, 2007.
- [11] Xu D. et al “ A novel approach based on PCNNs template for fingerprint image thinning” *IEEE* pp.115-119 2009
- [12] Li Z., Wang R. and Zhang Z., “Modified Binary Image Thinning using Template based PCNN”, *International Conference on Information Technology and Software Engineering Springer*, Vol. 212, pp.731-740, 2013.
- [13] Zhang P.G. “ Neural Networks for Classification: A Survey” *IEEE transactions on systems, man, and cybernetics*, vol. 30, no. 4, pp. 451-462 November 2000
- [14] Zhang T.Y. and Suen C.Y., “A Fast Parallel Algorithm for Thinning Digital Patterns”, *Communications of the Association of Computer Machinery (ACM)*, Vol. 27, No. 3, pp. 236-239, 1984.