



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

A Novel Pattern Classifier for Character Recognition Based On Spiking Neural Network

Soni Chaturvedi, A.A.Khurshid, Prachi R.Bajaj

Research Scholar, GHRCE, Nagpur, India

Dept of Electronics Engg, Nagpur University, RCOEM, India

Dept of E&C Engg, Nagpur University, P.I.E.T, India

Abstract- In this paper we focus on how spiking neural networks can be used to identify a class of an input image by LIF model. We address the problem of identifying English alphabet character image By Artificial Neural Network which are mathematical computational models inspired by biological Neurons. An English alphabet character images ("A to z") are first analyzed with Feed Forward Neural Network which belongs to Artificial Neural Network (ANN). Classes K1&K2 are considered for Capital letter and small letter respectively. The whole Recognition process is divided into four steps-Preprocessing, Classification, Post-Processing, and finally the comparison of Feed Forward Neural Network and Leaky-Integrate Firing Neuron Model is presented. In Proposed work, we have used Neural Network tool box and Image Processing Tool box in MATLAB software.

Key Words- Feed Forward Neural Network, Leaky-Integrate Firing Neuron Model, Pattern Recognition, SNN, ANN.

I. Introduction

Pattern Recognition is assignment of a physical object or a label to one pre-specified categories. Characters are the basic building blocks of any language. Pattern Recognition deals with categorizing an image according to its class to which it belongs. A pattern can be an x-ray image, an alphabet, finger prints, satellite images, microscopic images of blood cells etc. The measurement or properties used to classify objects is called as FEATURES, and the type or categories into which they are classified are called as CLASSES. In this paper we have considered four different images of an English character. Two of them are capital letters ("A", and "B"), other two are small case letter ("a" and "b"). Here we are trying to identify the class of an input image i.e. whether it is a capital or small letter. Classification process is first implemented with Artificial Neural Network by Feed Forward Neural Network and then with Leaky Integrate Firing Neuron Model.

II. Pre-processing

Pre-processing basically is an image processing. Image processing in turn is divided into two parts, i.e., Edge Detection and Feature –Extraction..Pre-processing is done to obtain the gray (monochrome) form of an image. In this paper Edge Detection is implemented by using Gaussian Filter. The algorithm used for edge detection is canny algorithm. Feature Extraction is done by using Gabor filter.

A. Edge Detection

A digital image can be considered as a picture, a photograph, graph etc. Image can be captured naturally (Human eye) or Artificially (By sensors or digital camera). Digital image is a Numeric representation having binary (0s and 1s) pixel values. Edge of an image can be considered as a boundary between certain set of pixels which vary in their intensity. Edge Detection is a process of identifying points in a digital image at which digital image changes. An edge can be considered as a line or curve or segment at which the change in pixel value occurs. There are various methods for edge detection. Here canny edge detection is used for detecting edges in an image due to its excellent stability against noisy data.

B. Canny Algorithm

Following steps describe the implementation of canny Edge Detection

a. Smoothing : It is the very first step used to remove the noise from a noisy image. It is mostly used for blur images. Images may suffer from noise due to Heat, Smoke, Dust, etc which makes them Blur. Hence Smoothing is necessary and can be considered as filtering the image, in order to eliminate noise and other unwanted harmonics contained in it. In this paper we have used Gaussian filter for image smoothing.

Gaussian Filter used here acts as an optimal smoothing filter and can enhance the input image for Pre-Processing. It reduces the high frequency components and thus acts as a low pass filter.

The input image is first convolve with a Gaussian Function $G(x, y)$ given by following eq. (1)

$$G(x, y) = 1/2\pi\sigma^2 e^{-(x^2+y^2/\sigma^2)} \dots \text{eq. (1)}$$

This Gaussian function when applied to two dimensional images produces concentric circles with a Gaussian distribution.

b. Estimation of Gradient

Now the second step in canny edge detection is to determine the Gradient of an image (G_x , G_y) along with its magnitude and direction given by following equations. Let $f(x, y)$ be the original image with spatial co-ordinates. Then Gradient $G [f(x, y)]$ is obtained as

$$G_x = \delta f(x, y) / \delta x = f(x+1, y) - f(x, y)$$

$$G_y = \delta f(x, y) / \delta y = f(x, y+1) - f(x, y)$$

Magnitude of the Gradient $|G|$ is given as

$$|G| = |G_x| + |G_y| = [G_x^2 + G_y^2]^{1/2}$$

Direction of the gradient is given by $\theta = \tan^{-1}(G_y/G_x)$

c. Non-Maxima suppression and Thresholding

Non maxima suppression is done to trace along an edge direction and suppress any pixel value which is not considered as an edge. After this two thresholds are set (T_1 - High and T_2 -low) for detection of strong and weak edges respectively. All the pixels having values below the pre decided threshold are considered as "0" (ZERO) and above the threshold are considered as "1" (ones).

An original RGB image is first smoothed for eliminating noise, then it is converted to YCbCr to obtain Y as gray level image as shown in the following fig.1 The Canny algorithm is then applied to this monochrome image to obtain finally, an Edge Detected out-put of an original image. An example of an English Alphabet "A" is shown in the following fig.

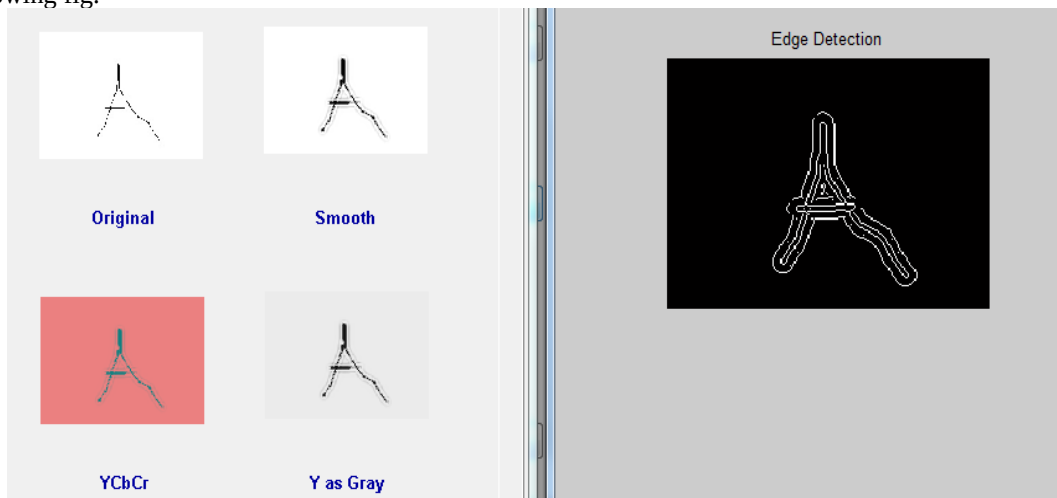


Fig 2.2 (a)

Fig2.2 (b)

C. Feature Extraction

Feature Extraction is a process of extracting some area of interest from an image. Any parameter that can be computed from the information in an image can be considered as a "Feature". Feature is used to denote a required part of total information which can be further used for solving, classification, or computation. It can refer as a neighborhood operation applied to an image. A feature can be shape-defined in terms of curves, Boundaries between different regions of an image. The input data fed to the Neural Network is given in terms of a Feature Vector. Feature vector is an N-dimensional vector having numerical values that represents certain object. Values in a feature vector represent certain pixel values of desired objects in an image. These values are combined with Weights to obtain dot products in Neural Networks. The vector space associated with these Vectors is called as Feature Space. In order to reduce dimensionality of feature space, dimensionality reduction techniques can be applied. There are various methods available for extraction of features. Some of them are Wavelet transformation, Least-Square Methods, Dutch Filter, Gabor filter etc.

In our paper, we have used Gabor Filter for feature extraction because of its similar characteristics with certain brain (neural) cells in a human visual cortex system.

1. Gabor Filter

Gabor filter can be viewed as a sinusoidal signal modulated by a Gaussian envelope. By applying Gabor filters to a character image, dominant orientation matrix can be obtained and used to form feature vector for recognition. It is

a kind of frequency filter which has been applied to texture analysis, moving object tracking, finger print recognition, and character recognition. Gabor filter performs special frequency analysis on an image. Gabor filter can be defined by its impulse response given by $h(x,y) = g(x',y') \cdot s(x,y)$

Where $g(x, y)$ = Gaussian distribution defined as

$$g(x,y) = 1 / [2\pi \sigma_x \sigma_y]^{1/2} e^{-[(x/\sigma_x)^2 + (y/\sigma_y)^2]^{1/2}} \quad \text{where}$$

$$(x',y') = (x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta), \text{ and } \theta = \text{orientation of the gabor filter, } \sigma_x, \sigma_y = \text{standard deviation}$$

We apply this kind of a Gabor filter with impulse response $h(x,y)$, given by the above equation. Then we convolve the input image $U(x,y)$ with $h(x,y)$ to obtain an output image $V(x,y)$. We have considered two frequencies (f_1, f_2) and six different orientation (θ_1 to θ_6) to obtain filter bank of 12 filters. There are Four Stages of filters each having 12 filters. Output of these filters is considered as feature vectors.

III. Classification By Feed Forward Neural Networks (FFNN)

A. Feed Forward Neural Network

Neural network is an adjustable model of outputs as functions of inputs. It consists of several layers:

Input layer, which receives the input data, Hidden layer, which consists of processing nodes called *neurons* and Output layer, which consists of one or several neurons, whose outputs are the network outputs. All nodes of adjacent layers are interconnected. These connections are called *synapses*. Every synapse has an assigned scaling coefficient, by which the data propagated through the synapse is multiplied. These scaling coefficients are called weights (w).

In this paper we adapt the successful technique of newff MATLAB Command to create the Feed Forward Network. Before training a feed forward network, the weights and Biases must be initialized. The standard network creating Function which we have used (i.e. “newff”) will automatically initialize the weights, but if we need to re-initialize the weights then it can be done with initializing commands. (“init”)

Once the Network is created, it is trained with Levenberg-Marquardt Back propagation Algorithm as this algorithm is the fastest speed method for all moderate sized Feed forward networks.

Let the Feature Vector be represented as X given by,

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

Where $x_1, x_2, \dots, x_n$ represents the input elements of a vector,

n = dimension of a vector,

Let W = Weight Matrix,

Th = Predetermined Threshold, Y = output of a feed forward network, then relation between X , W and Y is

$$Y = \sum_{i=1}^n X_i W_i$$

This neural output “ Y ” is then comparatively processed with preset Threshold to classify the input character image according to its class. Class “K1” is considered for capital letters and “K2” is considered for small case letters. Categorization of class of the input image is done by analyzing of output “ Y ” with respect to Threshold value “ T ”. Mathematically this condition is stated as following equation

$$Y = \begin{cases} 1 (K1) & \text{if } xw > T \\ 0 (K2) & \text{if } xw < T \end{cases}$$

Following figure shows the result obtained with the FFNN method for alphabet capital “E” as an example of character as well as class recognition.

Following table represents the analysis of four input images (A, B, a, b), with their classification according to its class and time elapsed in computation and Fitness Function in terms of Accuracy.

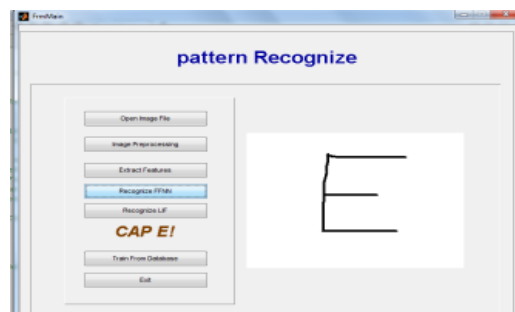


Fig 3.1.1(a)- Recognition of “E” by (FFNN)

English Alphabet	Prest (Th)	FFN O/P (P)	Class	Toc (Sec) ± 1%	Fitness Function (P / T)	Fitness Function in (%)
‘A’	0.12	0.11	Cap (K1)	0.696	0.9166	91.66 %
‘a’	0.3	0.29	Small (K2)	0.195	0.97033	97.03 %
‘B’	0.5	0.48	Cap (K1)	0.059	0.9688	96.8 %
‘b’	0.8	0.69	Small (K2)	0.134	0.8683	86.83 %

Table 1.1 Analysis with Feed Forward Neural Network

It should be noted that Standard deviation and firing rates are not applicable to Feed Forward Networks; hence they are not included in measurement parameters.

4. SNN As Pattern Classifier Leaky Integrate Firing Neural Network (LIF-Model) is a Spiking Neuron Model. Spiking neural network belong to third generation of Neural Networks which incorporate the concept a concept of time. These models increase the level of realism in wide range of areas from the field of neuroscience. In proposed method we are trying to identify the class as well as recognition of input English alphabet character image with this LIF model. An input image is first converted into proportional input signal with certain time period “T” (milliseconds). The spiking neuron is then stimulated with this input signal having “T” time period and fires when its membrane potential reaches a specific value. As a result, LIF model generates an action potential (spike) or a train of spikes. As LIF neuron cannot be directly stimulated with input pattern; it is stimulated with an input injection current ‘I’

As synaptic weights are connected to the input patterns, input injection current ‘I’ in turn can be calculated as $I = x.w.\theta$; Where, x = input pattern, W = synaptic weight, θ = gain factor

Synaptic weights of a neuron mode are adjusted such that input patterns belonging to same class produces same (almost) firing rates and input patterns belonging to different class generate different firing rate. The firing rate can be defined as the ratio of the number of spikes generated to the time period of input signal ‘T’.

It can be expressed as follows $fr = N / T$

Where N = No of spikes generated in an interval T ; T = Time period of input signal pulse in milliseconds. Once the firing rate is computed the Average Firing Rate (AFR) and Standard Deviation (STDV) can be computed. Mathematically, LIF model can be represented by various methods. In this paper, we use another representation of LIF model defined as $V = V - (V/(R*C)) + (I/C)$

Where dv/dt can be expressed as $dV/dt = -V/RC + I/C$

In this Paper, the parameters for the LIF Neuron are defined as, $C = 1nf$, $R = 40\text{ M}\Omega$, $I = 1n\text{-Amp}$, $V_{th} \leq 10\text{ v}$ (Threshold for generating spike), Neuron refractory period = 5ms, and $T = 200\text{ ms}$. (Time period of input current Pulse). The product RC is called as a time constant of the network represented by “ τ ” (called as “ τ_{ow} ”) $\tau = RC$ Following fig shows the result obtained with the LIF model for alphabet small case “e” as an example of character as well as class recognition

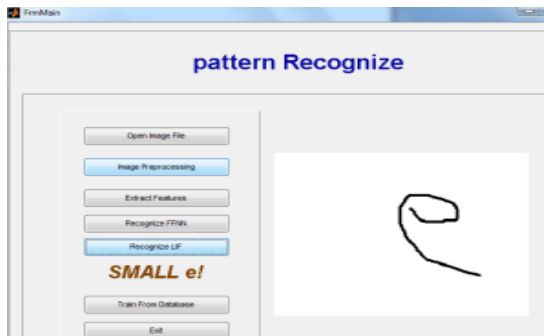


Fig 4.1.2- 'e'- out-put spikes

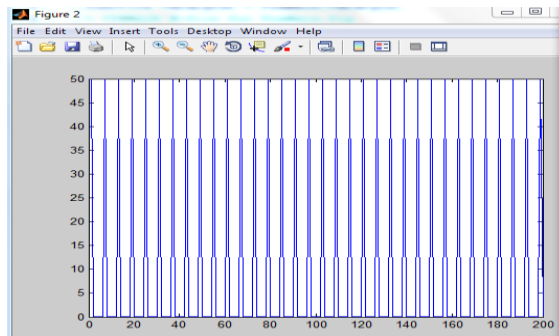


Fig- 4.1.3 Recognition of “e” by (LIF-Model)

The following table represents the analysis of input character image with fitness function, and STDV as Measurement parameters.

English Letter	Preset (Th)	LIF Model o/p (v)	Class	TOC (Sec) ($\pm 1\%$)	Fitness Function V/T	STDV
‘A’	0.12	0.11	Cap (K1)	0.105	0.9166	16.20
‘a’	0.3	0.2911	Small(K2)	0.0004	0.9703	16.90
‘B’	0.5	0.4844	Cap (K1)	0.0169	0.9688	20.37
‘b’	0.8	0.6947	Small(K2)	0.0037	0.8683	18.40

Table-1.2 Analysis With Leaky Integrate and Firing Neuron Model

IV. Invalid Input Condition

In Section 3 and 4, we have seen the Processing of Alphabets using FFNN and LIF model. However this network cannot identify any image other than English characters.

Following figure shows the Processing of a non character image (Flower) which Results in an Invalid Character Recognition (Not-Match) When applied as an input to both the networks.

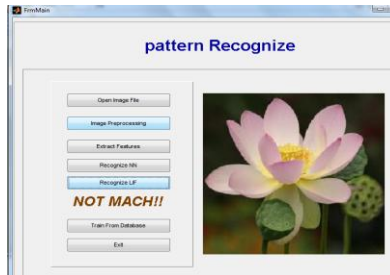


Fig-5.1.1 Invalid Input Condition (By FFNN and SNN)

V. Conclusion and Futurescope

In this paper we show that four English alphabets (two capitals and two small cases) are classified as well as recognized according to the class to which they belong.

The input images as well as pre-determined Threshold “T” is kept same for both the methods, for comparison of the results obtained from two methods.

First method was based on Artificial Neural Networks (Feed Forward Networks) and second method incorporates the concept of time and is based on spiking neural networks (LIF-Model).

It can be observed from above analysis from table 1.1 and 1.2 that time elapsed for in computation with LIF Spiking neural network is less than Feed Forward Networks. Hence Leaky Integrate and Firing Neuron Model are faster than first method. However the fitness function estimated in terms of Accuracy found out to be the same by both the methods.

Apart from this LIF model can be also compared with other networks such as oscillatory neural networks when they are applied to various nonlinear problems.

In future, certain experiments may be done to apply neural networks to various input vectors at the same time. Apart from Character Recognition, various other models of Neural Networks can be implemented for Recognition of Numbers and Special Characters, Symbols etc.

References

- [1] Roberto A. Vazquez, Aleistre Cachon, “Integrate and Fire Neurons and their Application in Pattern Recognition,” 2010, 7th International Conference on Electrical Engg, Computing science and Automatic Control (CCE 2010), September 8-10, 2010
- [2] C. Lin, L. Ko, and T. Shen, “Computational Intelligent Brain Computer Interaction and its Applications on Driving Cognition,” Computational Intelligence Magazine, IEEE, vol. 4, no. 4, pp. 32-46, 2009.
- [3] Paulo Roberto, Nunes de Souza, M’ara Regina and Labuto Fragoso da Silva, “Fourier Transform Graphical Analysis: an Approach for Digital Image Processing. Universidad Federal do Esp’irito Santo. 2009.
- [4] Price, K., Storn, R. M., Lampinen, J. A. “Differential evolution: a practical approach to global optimization”, Springer-verlag, 2005.
- [5] R. C. Gonzales, R. E. Woods and S. L. Eddins. *Digital Image Processing using MATLAB*. Prentice Hall, 2004. ISBN-13: 978-0130085191.
- [6] Thorpe, S. J., R. Guyonneau, et al. “SpikeNet: Real-time visual processing with one spike per neuron” *Neuro computing* 58(60):857-864, 2004.
- [7] Izhikevich, E. M. “Which model to use for cortical spikingneurons?”, *Neural Networks, IEEE Transactions on* 15(5):1063-1070, 2004.
- [8] Gerstner, W., Kistler, W. “Spiking Neuron Models”, Cambridge University Press, 2002.
- [9] R. Nayak, L.C. Jain and B.K.H. Ting. *Artificial Neural Networks in Biomedical Engineering: A Review*. In Proceedings Asia-Pacific Conference on Advance Computation. 2001. QUT Digital Repository.
- [10] Martin T. Hagan, Howard B. Demuth and Mark H. Beale. *Neural Network Design*. PWS Pub., 1996. ISBN: 9780534943325.
- [11] F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks*. New York: Springer-Verlag, 1997.
- [12] *Introduction to Artificial Neural system* -By Jacek.M. Zurada. ISBN-0314-93391-3, QA 76.87.Z.1992, West publishing company
- [13] *Pattern Recognition and image Analysis* -By Earl Gose and Steve Jost., Prentice Hall Publications ISBN-978-81-317-1537-6
- [14] *Pattern Classification* - By Richard O. Duda and Peter E. Hart, 2nd Edition, ISBN-81.