

2013 AASRI Conference on Intelligent Systems and Control

Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique

Amit Choudhary^{a,*}, Rahul Rishi^b, Savita Ahlawat^c

^aMaharaja Surajmal Institute, New Delhi, India

^bUIET, Maharshi Dayanand University, Rohtak, India

^cMaharaja Surajmal Institute of Technology, New Delhi, India

Abstract

The choice of pattern classifier and the technique used to extract the features are the main factors to judge the recognition accuracy and the capability of an Optical Character Recognition (OCR) system. The main focus of this work is to extract features obtained by binarization technique for recognition of handwritten characters of English language. The recognition of handwritten character images have been done by using multi-layered feed forward artificial neural network as a classifier. Some preprocessing techniques such as thinning, foreground and background noise removal, cropping and size normalization etc. are also employed to preprocess the character images before their classification. Very promising results are achieved when binarization features and the multilayer feed forward neural network classifier is used to recognize the off-line cursive handwritten characters.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](#).
Selection and/or peer review under responsibility of American Applied Science Research Institute

Keywords: OCR; Binarization; Feature Extraction; Character Recognition; Backpropagation Algorithm; Neural Network.

1. Introduction

The significance of piece of paper cannot be overlooked towards improving people's memory. It is used

* Corresponding author. Tel.: +91-991-133-5069.
E-mail address: amit.choudhary69@gmail.com.

for both private (letters, notes, addresses, reminders, lists, diaries etc.) and official correspondence (bank cheques, tax forms, admission forms etc.). The paper is important in our daily life because it is cheap, reliable, easily available, flexible in filling, secure for future references and is easy to keep. A huge amount of important historical data is also written on papers. So, there is a great demand to digitize all these paper documents so that the people all over the world can access these important sources of knowledge. For this purpose, the image of handwritten text is preprocessed and segmented into individual characters and are recognized by a neural network classifier.

The process of reading handwritten text from the static surfaces is termed as off-line cursive handwriting recognition. Simulating the behaviour of the human brain into a machine (for the task of reading handwritten or printed text) opened innovative prospects to improve man-machine interface. For the last four decades, the classification of cursive and unconstrained handwritten characters has been a major issue in this field of research.

2. Related work

The off-line character recognition is an active area of research these days. As compared to machine printed character recognition, the work done by the researchers in the area of handwritten character recognition is very limited as mentioned by Apurva A. Desai [1]. In 2002, Kundu & Chen [2] used HMM to recognize 100 postal words and reported 88.2 % recognition accuracy. In 2007, Tomoyuki et al. [3] used 1646 city names of European countries in the recognition experiment and the accuracy of 80.2% is achieved. In 2006, Gatos et al. [4] used K-NN classifier to recognize 3799 words from IAM database and reported 81% accuracy.

3. Handwritten Character Database Preparation

The handwritten character images are captured with the help of a digital camera. The character images can also be scanned by using a scanner. This process is known as Image Acquisition [5]. All the handwritten character images are converted to a uniform image format such as .bmp or .jpg so as to make all the images ready for the next processing step. Pure white background or some colored (noisy) background may be used to write/print these handwritten character images. These samples may be written with different pens of various colored ink. Character image samples contributed by 10 different people (age 15-50 years) are collected where each contributor writes 5 samples of the complete English alphabet (a-z). In this way 1300 ($10 \times 5 \times 26 = 1300$) character image samples are collected for the proposed experiment.

4. Preprocessing

Preprocessing is done to remove the variability that is present in off-line handwritten characters.

4.1. Grayscale conversion

In this phase of preprocessing, the input image of handwritten character in .bmp format from the local database as shown in Fig 1(a) is converted to grayscale format by using “rgb2gray” function of MATLAB and the resultant handwritten character image is shown in Fig 1(b).

4.2. Binarization

Binarization is an important image processing step in which the pixel values are separated into two groups; white as background and black as foreground. Only two colors, white and black, can be present in a binary image. The goal of binarization is to minimize the unwanted information present in the image while protecting the useful information. It must preserve the maximum useful information and details present in the image, and on the other hand, it must eliminate the background noise associated with the image in an efficient way.

It is assumed that the intensity of the text is less than that of background i.e. the input image has black foreground pixels and white background pixels. The colors can be inverted if the input image has text intensity more than that of background. Also, the background intensity remains almost uniform throughout the whole image and does not change drastically anywhere in the input image. Hence, in the proposed binarization technique, global gray scale intensity thresholding is employed and the resulting handwritten character image is shown in Fig 1(c), which is free from any background noise. The character image sample after foreground noise removal is shown in Fig 1(d). Resized image after cropping is shown in Fig 1(f).

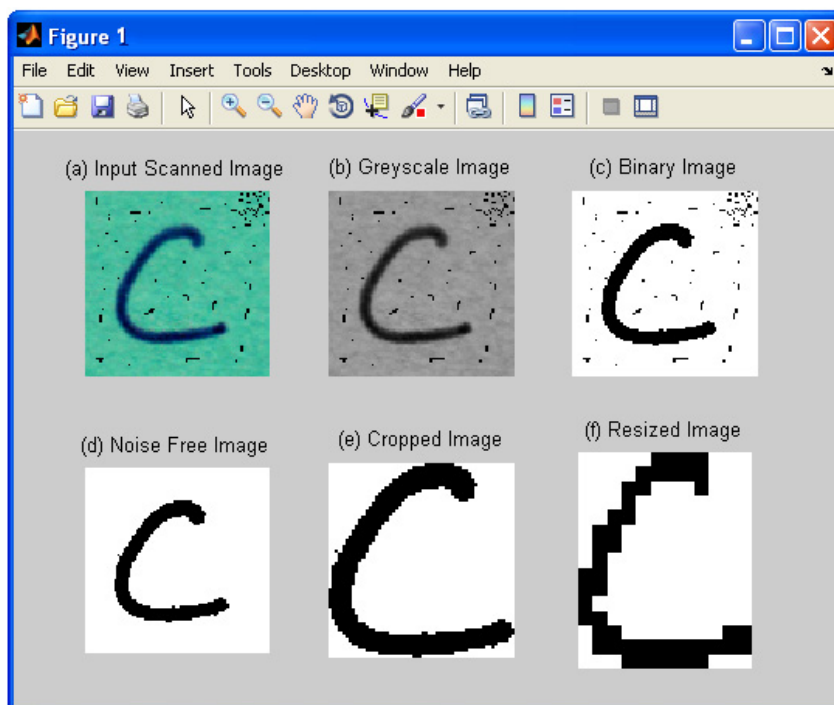


Fig. 1. (a) Input scanned handwritten character image; (b) Handwritten character image in grayscale format; (c) Character Image in binary format; (d) Character image after foreground noise removal; (e) Cropped character image; (f) Resized handwritten character image.

5. Feature Extraction and Training Sample Preparation

The binary image of character 'c' is shown in Fig 2(a). It is resized to 15×12 matrix as shown in Fig 2(b). A '0' indicates the presence of a white pixel and a '1' represents the presence of a black pixel as shown in the

binary matrix representation of character 'c' in Fig 2(c). This binary matrix of size 15×12 is then reshaped in a row first manner to a binary matrix of size 180×1 by using 'reshape' function of MATLAB and is shown in Fig 2(d). This column vector of size 180×1 is a feature vector of character 'c'.

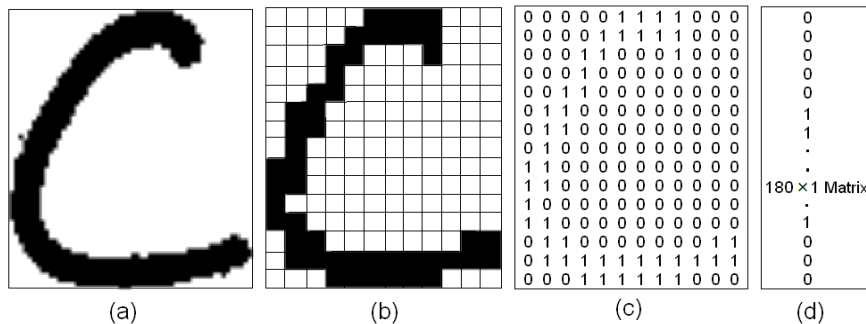


Fig.2. (a) Binary image of character 'c'; (b) Resized binary image of character 'c'; (c) Binary matrix representation & (d) Feature vector of character 'c'.

Similarly, the feature vectors of all the 26 characters (a-z) are created in the form of binary column matrix of size 180×1 each. All these 26 feature vectors are combined a binary matrix of size 180×26 as shown in Fig 3. This matrix is termed as a sample.

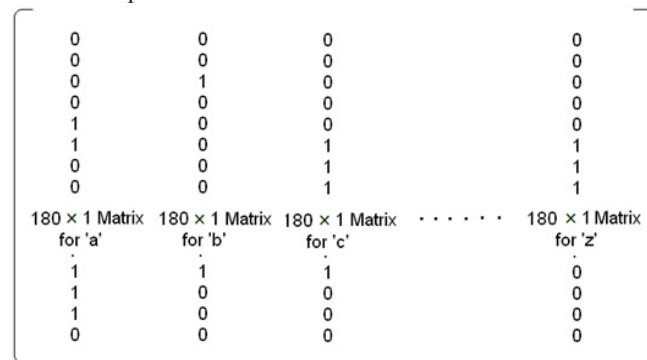


Fig.3. Matrix representation of input sample of size 180×26

In this matrix, first column represents feature vector of character 'a', second column represents feature vector of character 'b', third column represents feature vector of character 'c' and so on. In order to create samples, 1300 character images are collected from 10 contributors (age 15-50 years) where each writer contributed 5 samples of the complete English alphabet (a-z) ($10 \times 5 \times 26 = 1300$). Thus each sample consists of 26 English alphabets. All these samples are used to train the neural network classifier.

6. Implementation

The size of the input layer depends on the size of the sample presented at the input and the size of the output layer is decided in accordance with the number of output classes in which each of the input patterns is to be classified. In the proposed experiment, the feature vector of each of the 26 character images is of size 180×1 . Hence, 180 neurons are used in the input layer and 26 neurons are used in the output layer of the

neural network classifier. For optimal results, 80 neurons are kept in the hidden layer by trial and error method. The ‘tansig’ activation function is used for hidden as well as output layer neurons. Neural network training process is shown in Fig 4 and the adaptive learning function ‘traingdx’ has been used. Mean Square Error (MSE) has been selected as a cost function in the training process shown in Fig 4.

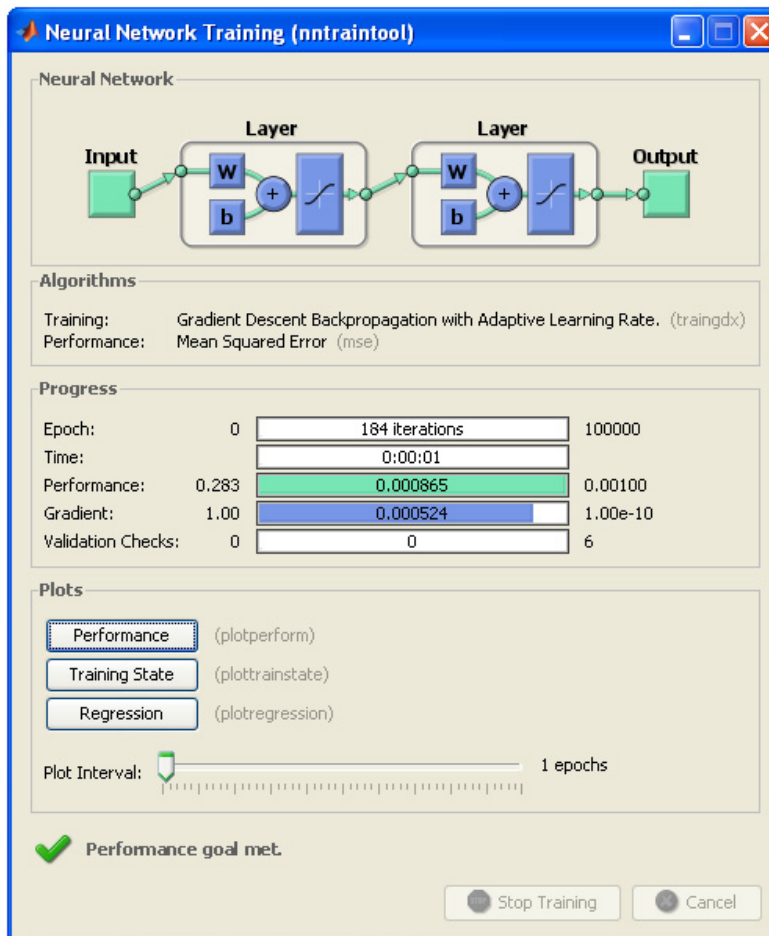


Fig.4. Training process of the network

In case of back-propagation neural networks, universally accepted cost function to measure the generalization performance is MSE. The lower value of the cost function indicates that the neural network is capable of mapping the input and output in a right manner. The acceptable threshold for the MSE (cost function value) has been selected as 0.001 and the training of the neural network will come to an end when the error becomes less than or equal to this threshold value. The performance value indicates the extent of training of the network. A low performance value (0.000865) indicates that the network has been trained properly.

In case of real-world applications, the performance of the neural network classifier also depends on the number of training iterations required to train the network. Too less number of training epochs result in a

poorly trained network due to under-fitting of the network. On the other hand, too many training epochs result in poor generalization due to over-fitting of the network. The network learning iterations must be selected in such a way that the network may converge properly with least generalization error. The maximum allowed epochs for the training process has been set to 100000 as shown in Fig 4. If the network could not converge within the maximum allowed epochs count, the training will stop.

7. Discussion of Results

Exactly 50 samples of each character image are prepared for the training process and are presented to the neural network classifier. Each character pattern presented at the input layer will put a '1' at only one neuron of the output layer in which there is the highest confidence. A '0' is put at all the remaining neurons. For every character pattern at the input, the output is a 26×1 column matrix, in which a '1' is present at a single place only and the remaining 25 entries are all '0' e.g. character 'a' results in (1, 0, 0 ..., 0) and character 'b' results in (0, 1, 0 ..., 0) & so on. In this way, all the individual characters at the input are represented by column vectors of size 26×1 each at the output. As there are 26 characters in a sample, the output of a sample presented at the input is a 26×26 matrix.

Output Class	Alphabets	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	Success (%)
	a	43	0	0	0	0	0	0	1	0	0	0	0	2	8	5	0	0	0	0	0	0	3	0	0	0	0	86
	b	0	49	0	2	0	0	0	0	0	0	3	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	98
	c	0	0	44	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88
	d	0	1	0	38	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	76
	e	2	0	4	0	44	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	88
	f	0	0	0	0	0	46	0	0	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	1	92
	g	0	0	0	0	0	0	39	0	0	10	0	0	0	0	0	1	4	0	0	0	0	0	0	0	13	0	78
	h	0	0	0	0	0	0	0	49	0	0	5	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	98
	i	0	0	0	0	0	0	0	0	37	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74
j	0	0	0	0	0	0	1	0	5	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58	
k	0	0	0	0	0	0	0	0	0	0	42	2	0	0	0	0	0	0	0	1	0	0	0	0	1	0	84	
l	0	0	0	1	0	1	0	0	3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	
m	0	0	0	0	0	0	0	0	0	0	0	0	41	2	0	0	0	1	1	0	0	0	0	0	0	0	82	
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	1	0	0	0	0	0	0	76	
o	5	0	1	0	1	0	0	0	0	0	0	0	0	0	0	43	0	0	0	1	0	0	0	0	0	0	86	
p	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	10	0	0	0	0	0	88	
q	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	1	0	92	
r	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	49	0	0	0	0	0	0	0	0	98	
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	92	
t	0	0	0	4	0	3	0	0	5	0	0	6	0	0	0	0	0	0	0	38	0	0	0	0	0	2	76	
u	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	48	5	0	0	0	0	96	
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	40	0	0	0	0	80	
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50	0	0	0	0	100	
x	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	50	0	1	0	100	
y	0	0	0	0	0	0	8	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	70	
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	92	
Overall Character Recognition Accuracy =																										85.62 %		
Target Class																												

Fig.5. Confusion matrix representing the performance of the neural network classifier

In the proposed handwritten character recognition experiment, the neural network has been trained by each of the 26 characters 50 times i.e. 1300 (50× 26=1300) character image samples from the database has been involved in the learning process. Recognition uncertainty between the various characters is presented in Fig 5. Character 'a' is recognized in an accurate manner 43 times out of 50. Out of 7 miss-classifications, character 'a' is classified as 'e', 2 times and 'o', 5 times respectively. The average overall recognition accuracy of 85.62% is quiet good for this handwritten character recognition experiment as shown in the form of a matrix representing the confusion among the various English alphabets as presented in Fig 5.

8. Conclusion and Future Scope

The use of binarization features along with the neural network classifier employing back-propagation algorithm delivers outstanding classification accuracy of 85.62 %. Training sample quality, feature extraction technique and the classifier are the main factors deciding the accuracy of the recognition system. All these techniques can be refined because a scope of improvement is always there. In future, a combination of binarization features with some other type of features such as Projection profile Features, can be investigated in the recognition experiment. Apart from MLP classifier, other classifiers such as RBF, HMM, SVM etc. can also be examined in future.

References

- [1] Desai, A. A., 2010. “Gujarati handwritten numeral optical character recognition through neural network”, *Pattern Recognition*, 43, pp. 2582—2589.
- [2] Kundu, Y. H., Chen, M., 2002. “Alternatives to variable duration HMM in handwriting recognition”, *IEEE Trans Pattern Anal Mach Intell*, 20(11), pp. 1275–1280.
- [3] Tomoyuki, H., Takuma, A. & Bunpei, I., 2007. “An analytic word recognition algorithm using a posteriori probability”, in *proceedings of the 9th international conference on document analysis and recognition*, 2, pp. 669–673.
- [4] Gatos, B., Pratikakis, I. & Perantonis, S. J., 2006. “Hybrid off-line cursive handwriting word recognition”, in *proceedings of 18th international conference on pattern recognition (ICPR’06)*, 2, pp. 998–1002.
- [5] Choudhary, A., Rishi, R., and Ahlawat, S., 2010. “Handwritten Numeral Recognition Using Modified BP ANN Structure”, *Communication in Computer and Information Sciences (CCIS-133)*, *Advanced Computing*, Springer-Verlag, pp. 56-65.