International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 03, March 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

HANDWRITTEN NUMERAL PATTERN RECOGNITION TECHNIQUES: REVIEW PAPER

Ritika Dewan[1], Arun Kumar[2]

[1]ECE, [2]CSE, BMIET/DCRUST, (India)

ABSTRACT

Primary goal of handwritten numeral pattern recognition techniques is to efficiently recognize the offline handwritten digital text with higher accuracy than that of the previous work done in this field. Among various frameworks in which pattern recognition has been traditionally formulated, statistical techniques intensively used in practice. While recognizing the textual patterns the system requires very careful attention towards definition of pattern classes, pattern representation, sensing environment, image acquisition, preprocessing, feature extraction, classification, cluster analysis, classifier design and learning, selection of samples and training the data. Aim of many persons in this field remains almost same that is to recognize characters with better accuracy but my main goal is to recognize multiple digits simultaneously at single time with batter accuracy.

Keywords: Pattern Recognition, Image acquisition, Preprocessing, Feature Extraction, Training, Back-Propagation algorithm

I. INTRODUCTION

Pattern recognition techniques are used to process data and also for decision making. By the time , many children able to recognize letters , digits and text. Variety of characters which are handwritten or machine printable which can be easily recognizable by youngsters. As best pattern recognizers are humans. We want this ability in machines to do the same work as that of humans by using artificial intelligence. This needs various steps that is image acquisition, preprocessing, feature extraction, classification, training and many more.

II. PATTERN RECOGNITION TECHNIQUES

2.1 Introduction

Various statistical techniques pattern is represented in terms of d features or measurements and is viewed as a point in a d-dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from each class are provided, the objective is to establish the decision boundaries in the feature space which separate patterns belonging to different classes. In statistical decision approach, the decision boundaries determined by probability distributions of the patterns belonging to each class, which must either be specific or learned [1].

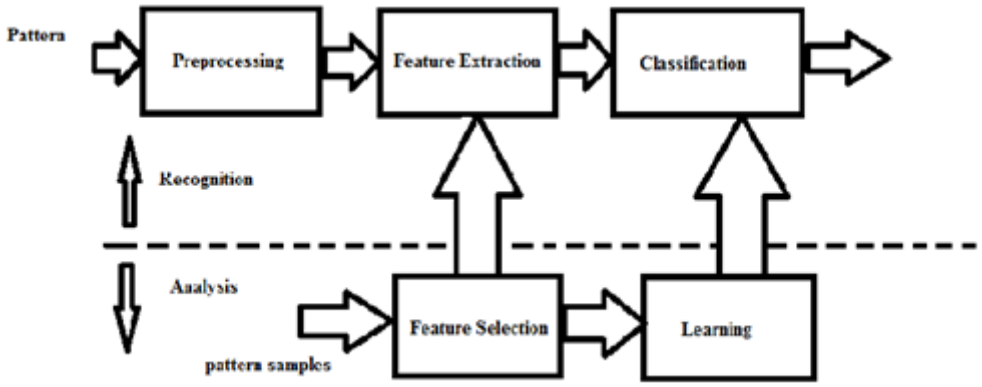

Fig. 1 Block Diagram of Statistical pattern Recognition System

2.2 Image Acquisition

Digital image can be acquired either through scanning or using digital camera. Scanning is the process of converting document into electronic form, which is suitable for subsequent processing on a digital computer. The choice of spatial resolution for scanning is determined by two factors that is contents of documents and purpose of subsequent operation. Too high resolution may reveal texture of paper. Sample is given below learn from [3].

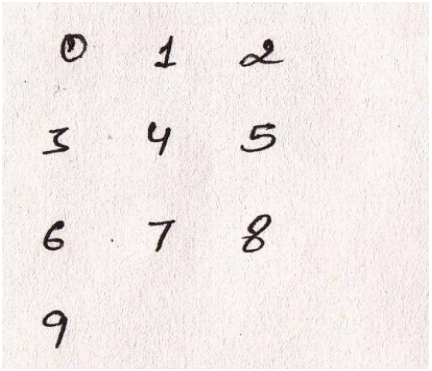

Fig. 2 Sample text image

2.3 Preprocessing

Selection of an adequate data representation is the key point in pattern recognition. If a low level representation is used in that case a very large training set for neural network is required. So a high level data into high level data. Operations like noise removal, thresholding, linking broken digits, thinning, rotation pruning and cropping come under this stage. Most of these based on mathematical morphological techniques. During image processing, both input and output are images, and in some cases inputs are images and outputs are attributes extracted from those images.

2.3.1 Segmentation

This operation is performed to isolate the desired characters. For cursive handwriting it became difficult to isolate the objects, therefore appropriate segmentation method must be used depending upon the data set present and application requirement. There are various types of segmentation methods present in image processing like

threshold techniques, edge-based methods, region-based techniques and connectivity-preserving relaxation methods. Segmentation of a digit is shown below learn from [3].

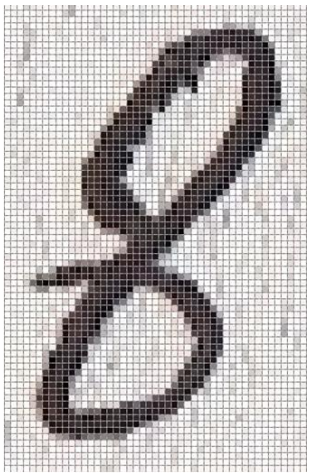

Fig. 3 Segmented image

2.3.2 Thresholding

Thresholding is most commonly used preprocessing techniques applied to separate the foreground information of an image from its background. One obvious way to do this task is to select a threshold value T that separate these modes. Before, selecting this value a detailed analysis is performed on the intensity histograms of character images. We have used a threshold value of 0.7 in our system. Binary image obtained after applying thresholding operation learn from [3].

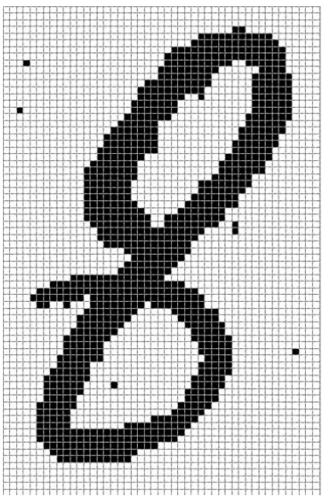

Fig. 4 Image after Preprocessing.

2.3.3 Noise Removal

Digital capture of images nmay introduce noise from scanning devices and transmission media. The noise can be removed by reconstructing the image surface from the surface of inscribed blocks found by gray scale skeleton transformation. Various filters mean filter, median filter can also be used to remove noise. Noise free image obtained after passing through median filter learn from [3].

International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 03, March 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

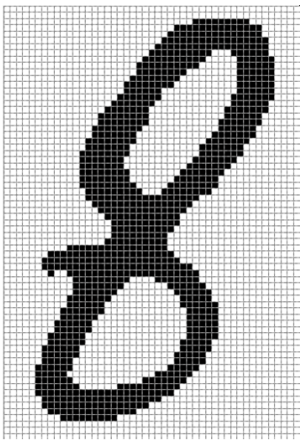

Fig. 5 Image after Noise removal.

2.3.4 Slant Correction

Numerals written in different orientations requires rotation prior to recognition. The image is rotated after determining skew angle. As in unconstrained handwriting different numerals will possess different slant. They shall need different amount of alignment in clockwise or anti clockwise direction.obtained after performing 15 degree anticlockwise operation below learn from [3].

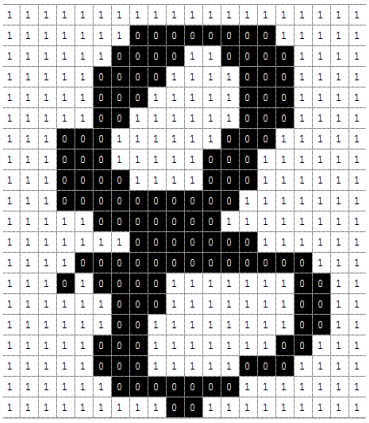

Fig. 6 Slant corrected image

2.3.5 Thinning

This operation is performed to remove the meaningless line-width variations. This simplifies structural analysis, reduce storage size and transmission time and reduce significantly the complexity of character recognition time. Basic idea of thinning is to repeatedly delete objects boundaries pixels so as to reduce line width to one pixel. Figure is obtained after performing thinning operation learn from [3].

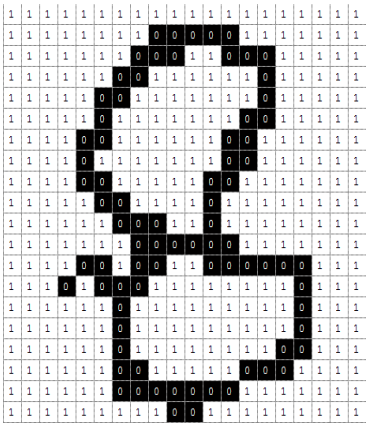

Fig. 7 Image after Thinning.

2.3.6 Pruning

The thinned images may contains spur due to non-uniformities in the strokes composing the digits. This spurs reversely effects structural analysis and removed by pruning operation. Morphological techniques are used to suppress the parasitic branch by successively eliminating its end-point.

2.4 Feature Extraction

For handwritten character recognition features are extracted from images and then classification is performed based on resultant feature map. The main task of this stage is to describe the relevant properties of the object to be recognized as precisely as possible with a fixed number, as small as possible, of feature variables. The extracted features should minimize the within-class pattern variability. The widely used feature extraction methods are Template Matching, Deformable templates, Unitary image transforms, Graph description, Projection histograms, Contour profiles, Zoning, Geometric moment invariants, Zernike moments, Spline curve approximation, FDourier descriptors, Gradient features and Gabor features [2]. Characteristic features are:

- Global features
- Statistical features

III. BACK PROPAGATION ALGORITHM

Back-propagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard back-propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term *back-propagation* refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. The simplest implementation of back-propagation learning updates the network

weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written as

$$x_{k+1} = x_k - \alpha_k \ g_k$$

where x_k is vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate. The various back-propagation algorithms are:

3.1 Batch Gradient Descent

The weights and biases are updated in the direction of the negative gradient of the performance function. The learning rate is multiplied with the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge. Back-propagation is used to calculate derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent:

dX= Learning Rate* d(Performance)/dX

3.2 Gradient Descent with Momentum

Gradient descent with momentum, allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum. With momentum a network can slide through such a minimum. Gradient descent with momentum depends on two training parameters: Learning rate and momentum constant. Momentum constant is set between 0 (no momentum) and values close to 1 (lots of momentum) [2]. A momentum constant of 1 results in a network that is completely insensitive to the local gradient and, therefore, does not learn properly). Back-propagation is used to calculate derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent with momentum.

dX_n = momentum constant*dX_{n-1}+Learning Rate*d(Performance)/dX

3.3 Gradient Descent with Adaptive Learning Rate

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface [4]. The performance of the steepest descent algorithm can be improved if we allow the learning rate to change during the training process. An adaptive learning rate attempts to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface. Back-propagation is used to calculate derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent:

dX = Learning Rate*d(Performance)/dX

3.4 Gradient Descent Momentum with Adaptive Learning Rate

It is invoked in the same way as Gradient Descent with Adaptive Learning Rate, except that it has the momentum coefficient as an additional training parameter. Back-propagation is used to calculate derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent with momentum

$$dX_n = \text{momemntum constant} * dX_{n-1} + \text{Learning Rate} * \text{momentum constant} * d(\text{Perf.})/dX$$

3.5 Resilient Back-propagation

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when I use steepest descent to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient back-propagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. Back-propagation is used to calculate derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to the following:

$$dX = \text{deltaX} * \text{sign}(\text{Gradient})$$

3.6 Conjugate Gradient Descent Algorithm

The basic back-propagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient), the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In most of the conjugate gradient algorithms, the step size is adjusted at each iteration. A search is made along the conjugate gradient direction [4] to determine the step size that minimizes the performance function along that line.

IV. PERFORMANCE ANALYSIS PARAMETERS

While designing neural network, these parameters must be decided. No. of neurons in hidden layers, Learning Rates, Momentum, Training Types, Epoch, Minimum Error

4.1 Mean Squared Normalized Error Performance Function (MSE)

Mean Squared Error is the average squared difference between outputs and targets.MSE is the second moment of the error, and thus incorporates both the variance with respect to target value. The MSE of an output value $\hat{y}$ with respect to the target value y is defined as

$$MSE(\hat{y}) = E[(\hat{y}-y)^2]$$

4.2 Percentage Recognition Accuracy

Percentage Recognition Accuracy indicates the fraction of samples which are correctly classifies. It can be estimated by analyzing confusion matrix [5]. A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. In unsupervised learning it is usually called a matching matrix. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

TABLE-I Comparison in terms of accuracy

NAME OF TECHNIQUE	ACCURACY
DISCRIMINANT ANALYSIS	90.50%
PRINCIPAL COMPONENT ANALYSIS	78.4%
BACK PROPAGATION NETWORK	99.7%
GENERAL REGRESSION NETWORK	91.70%

V. CONCLUSION

Performance of handwritten numeral recognition system depends upon respective efficiency of image preprocessing subsystem and network training subsystem. Activation function affects performance behavior of a learning system. An optimum performance is achieved with conjugate gradient descent Back propagation algorithm with 99.7% recognition accuracy and 8.26e-5MSE [6]. Performance is improved when activation function combination is 'Logsig'-'Tansig' and Tansig'-'Tansig'.

REFERENCES

[1] L. Devroye, L. Gyorfi, and G. Lugosi, A Probabilistic Theory of Pattern Recognition. Berlin: Springer - Verlag.

[2] J. Pradeep *et al*., "Diagonal Based Feature Extraction for Handwritten Character Recognition System Using Neural Network." IEEE 2011.

[3] Ritika , Handwritten numeral pattern recognition in neural network, Theses Report"BMIET 2016.

[4] A.L. Knoll, "Experiments with 'Characteristic loci' for recognition of handprinted characters," IEEE Trans. Computers, vol 18, pp. 366-372, 1969

[5] P. Mermelstein and M. Eden, "Experiments on computer recognition of connected handwritten words, "Inform Contr. , vol 7.pp. 255-270, 1964

[6] B. Gatos, I. Pratikakis and S.J. Perantonis, "Hybrid Off-Line Cursive Handwritten Word Recognition ", 18th International Conference on Pattern Recognition (ICPR2006),pp. 998-1001, Hong Kong, August 2006.