Handwritten Character Recognition using K-NN Classification Algorithm

Siddhartha Roy[1], M.Saravanan[2]

[1]*UG Graduate, Computer Science Engineering, SRM University, Chennai, India*
[2]*Assisstant Professor, Computer Science Engineering, SRM University, Chennai, India*

ABSTRACT

Building an effective methodology to detect hand-written characters from images with less error rate is the great task. Our aim is to make such an algorithm that will be able to generate error free recognition of hand written text from the given input image which will be a hand written character, and will help in document digitizing. OCR has always been an intensive research topic for more than 4 decades, it is probably one of the most time consuming, as well as labor intensive work of inputting the data through keyboard .This paper discusses about mechanical or electronic conversion of scanned images, text which contain graphics ,image captured by camera ,scanned images and the recognition of images where characters may be broken or corrupted .The optical character recognition is the desktop based application developed using Python 3.0 and .We should gain > 97.82% accuracy when applied on different data sets, during pre-processing we will use different techniques to remove noise from the background of the image. We will convert the labelled data into grey scaled images and train the classifier and generalize it using the validation set to reduce any kind of validation error.
Finally, we will test the module using the test_data and see the outcome of the algorithm. We display the images as well using the matplot library provided by Python.

Keywords: *Grey scale image, K –NN Classification algorithm, OCR, machine learning, matplot, MNIST dataset, sklearn, nump.*

Handwritten Character Recognition using K-NN Classification Algorithm

I. INTRODUCTION

Hand written character recognition is the electronic conversion of optically processed characters. Character recognition can be offline or online, in online character recognition computer recognises the character when it is detected.[4]

This paper discuss about the implementation of offline character recognition in python technology. The offline characters are the single characters which can be printed or handwritten.

In general the input to the OCR system is the image which are taken from camera, text images, handwritten text, typed or printed these images may have joint characters, fragmented characters, images with graphics or geometry and a noise containing text images.[1]

In this method of OCR, there are three important steps which are Segmentation, Feature Extraction, and Classification. In this paper we discuss about how our algorithm deals with these problems.

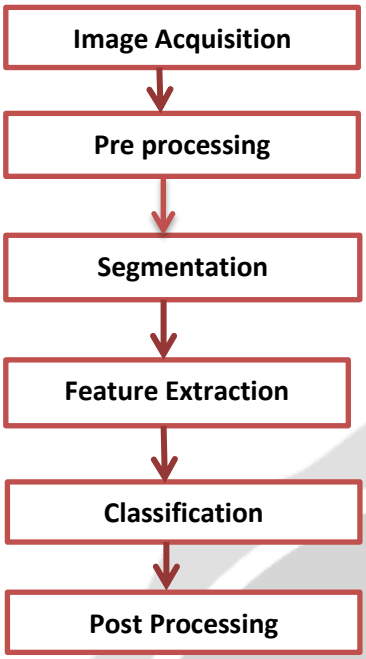

Fig. 1. Steps of Hand-Written character recognition

During segmentation we determine the elements of an image, it is important to determine the printed data and separate it out from the figures and graphics, isolated characters i.e. characters which will be recognized individually are segmented, this technique is easy to implement but trouble occurs when characters are overlaping each other. Noise containing images, graphics and geometry in the image are some other problems in the image moreover due to connected characters and merged characters in graphics, the recognition stage do not get input text for recognition [6].

The third important step of recognition is the classification, identifying each element character and assigning it to the correct character class. There are different approaches by which classification can be done.First is the decision-theoretic method, this method is used when the characters can be numerically represented in a feature vector. Second method is
structural approach this method is used when there is
relationship between characteristics of characters for example if we know that the character consists of one vertical line and one horizontal line, so we can predict if it's "L" or "T". In this paper we have implemented the classification through neural network using back propagation algorithm.[10]

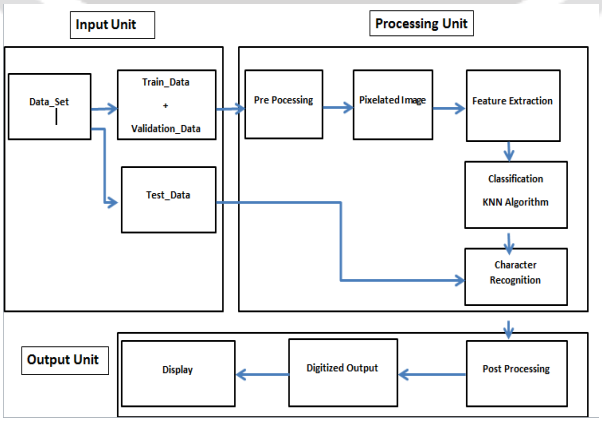

Fig2: OCR Architecture Diagram

The rate of recognition of characters directly depends on the quality of image i.e. the image resolution. The scanned images are more complicated due to many possible variations in background and fonts. In this paper we will discuss the robust algorithms which work on different kind of images with different font size, colour and text.

The software is completely implemented in Python. First image is loaded in the initial module, from where it reaches pixel extraction in its original form. The dimensions of the image can vary in size. In pre-processing of image we applied different techniques to remove unwanted graphics, noise, and unwanted text .Image can be cropped, resized, rotated left or right, zoom in ,zoom out, and can also adjust resolution of
image, if unwanted text, graphics will be removed from image extra noise in the sample image will be reduced greatly, which will increase the accuracy of recognition of text, all the module has been implemented in Python.

II. MODULES

Module 1

The first module deals with the extraction of data from the MNIST data set and the pre- processing of it. MNIST data set contains pictures of hand-written letters from different sources. The classification algorithm cannot work with the raw data which was extracted from the site. Thus, we have to make few changes to the data sets. Thus, we change the raw data into a single one dimentional array. We also reduce the noise from the images. Finally we convert them into data sets with which the classification algorithm can work on.

Module 2

The second module deals with the classification of the data sets which is the training data_set being provided to it. Along with the validation_set which will be used to generalize the classification algorithm and make it ready to deal with real-world problems. Thus, after that the test_data, which works as the real-world problem is fed to the system and what the classification algorithm does is to classify the data which is given. The test_data is also a type of train_data, in other words, the format of the data is the same as the train_data. Finally, the classification is done and the post processing takes place in which the data is converted back into the digital format and is displayed as an output.

III. DATABASE AND PRE-POCESSING

a) Pre-processing

The proposed method uses MNIST (Modified NIST) [6] database which includes a training set of 50,000 images and a test set of 10,000 images. The training and test sets are subset of NIST digit base. The MNIST digit database contained fixed size images and digit image (foreground pixels) is center alignment with respect to the background pixels. The MNIST digit database is good database for applying learning techniques and patterns recognition methods because of this database need less time for noise removal in preprocessing. Originally, The MNIST database was constructed from NIST's Special Databases (SD) 3 and 7 which contain grayscale images of handwritten digits. The SD-3 was collected from employees of Census Bureau and SD-7 images were collected local high-school students. The MNIST training set is composed of 30000 images from SD- 3 and 30000 images from SD-7. Totally, 60000 images are taken from 750 writers. However, the two databases, were written by totally different sources of writers and show different styles. The sample binary images were normalized into 8x8 gray-scale images with aspect ratio preserved and the normalized image is located in a 28 × 28 plane. The normalized image data are available at the homepage of LeCun [6]. Some images are shown in Fig. 3.

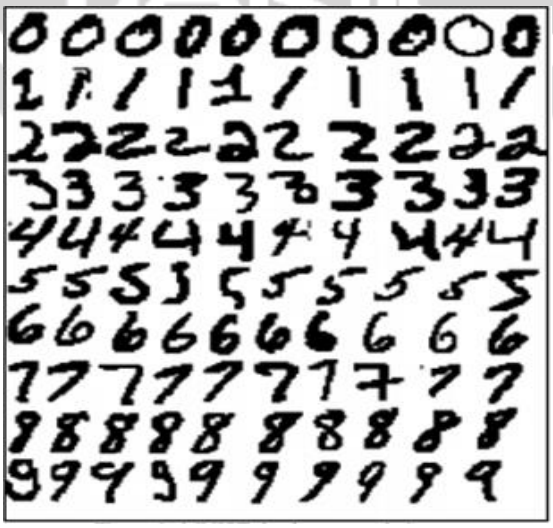

Fig3: MNIST Database Sample Images

IV. ALGORITHM

The overall classification design of the MNIST digit database is shown in following algorithm. Algorithm: Classification of Digits Input: Isolated Numeral images from MNIST Database Output: Recognition of the Numerals Method: Structural features and K-nn classifier.

Step1: Convert the gray level image into Binary image

Step2: Preprocessing the Binary Image

Step3: Convert the Binary Image into a single Dimentional Array of [1,n]

Step4: Keep the label of each Array along with it.

Step5: Feed the classifier with the train_data set.

Step6: Repeat the steps from 1 to 5 for all images in the Sample and Test Database.

Step7: Estimate the minimum distance between feature vector and vector stored in the library by using Euclidian distances.

Step8: Classify the input images into appropriate class label using minimum distance K-nearest neighbor classifier.

Step9: End.

V. CLASSIFICATION

The proposed method uses k-nearest neighbor (knn) classification algorithm for classifying the MNIST digit images in test set using the feature vector of training database. The k-nearest neighbor algorithm (k-NN) is a classification technique which classify the objects base on training features space. The functionality of k-NN algorithm is to define the computations until classification is done irrespective of the learning techniques.

Generally k-NN has two learning techniques. They are 1. Instance-based and 2. Lazy learning techniques. K-nearest neighbor algorithm is simplest classification technique because of computations are simple. The classification of objects based on votes of its neighbors which represented by by k. In K-nn object is classified to a particular class which has majority of votes.

In pattern recognition, the **k-nearest neighbors algorithm** (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

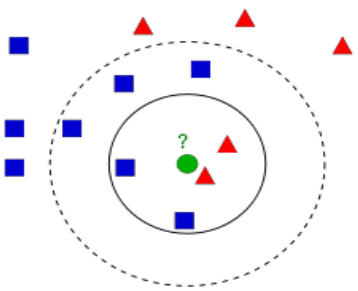

Fig2. K-NN Classification Diagram

Our data should be a floating point array with size NUMBER OF TEST DATA X NUMBER OF FEATURES. Then we find the nearest neighbours of new-comer. We can specify how many neighbours we want. It returns:The label

given to new-comer depending upon the kNN theory we saw earlier. If you want Nearest Neighbour algorithm, just specify *k=1* where k is the number of neighbours.The labels of k-Nearest Neighbours.Corresponding distances from new-comer to each nearest neighbour.

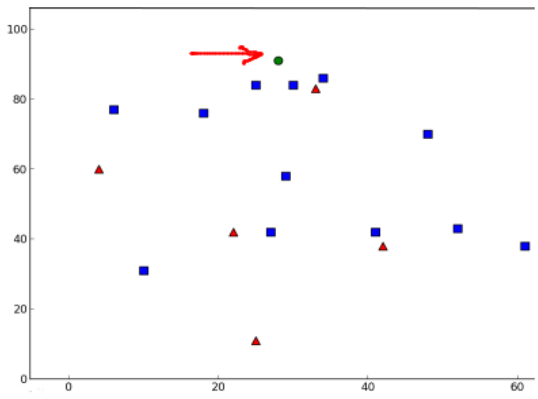

Fig4: Graph of K-NN Algorithm

VI. CONCLUSION

In this paper we used ten (10) features for recognition of handwritten numerals. In any recognition process, the important problem is to address the extraction of feature and correct classification approaches. The proposed algorithm tries to deal with both the factors and well in terms of accuracy and time complexity. The Overall accuracy of 97.67% is achieved in the recognition process. The novelty of this method is that free from size normalization and accurate ,independent of size of the digit and writer style independent, fast and accurate. This work is carried out as an initial attempt, and the aim of the paper is to facilitate for robust English OCR. It is our future Endeavour to modify this algorithm and design a still robust handwritten English OCR for high recognition rate and also recognition of offline handwritten digit recognition with less number of features and without using any standard classification algorithm.

VII. REFERENCES

[1] Yann LeCun,"THE MNIST database of handwritten digits" Courant Institute, NYU Corinna Cortes, Google Labs, NewYork .http://www.research.att.com/yann/exdb/mnist/index.html

[2] " *Optical character recognition using template matchingand back propagation algorithm"* Swapnil Desai, Ashima Singh.

[3] U Pal and P.P.Roy, "Multi-oriented and curved text lines extraction from Indian documents", IEEE Trans on system, Man and Cybernetics-Part B, vol.34, pp.1667-1684, 2004.

[4]B.V.Dhandra, R.G.Benne, Mallikarjun Hangarge, " Handwritten Kannada Numeral Recognition Based on Structural Features" Int. conference on Computational Intelligence and multimedia Applications 2007.

[5] Y. LeCun, et al., Comparison of learning algorithms for handwritten digit recognition, in: F. Fogelman-Souli e, P. Gallinari (Eds.), Proceedings of the International Conference on Artificial Neural Networks, Nanterre, France, 1995, pp. 53–60.

[6] P.Nagabhushan, S.A.Angadi, B.S.Anami, "A fuzzy statistical approach of Kannada Vowel Recognition based on Invariant Moments", Proc. Of NCDAR-2003, Mandy, Karnataka, India, pp275- 285, 2003.

[7]L.Heutte, T.Paquest, J.V.Moreau, Y.Lecourtier, C.Oliver, "A structural/ statistical feature based vector for handwritten character recognition", Pattern Recognition, p.629-641, 1998.

[8] Object Recognition Using K-Nearest Neighbor Supported By Eigen Value Generated From the Features of an Image**,** Dr. R.Muralidharan

[9]A.L.Koerich, R. Sabourin, C.Y.Suen, "Large off-line Handwritten Recognition: A survey", Pattern Analysis Application 6, 97-121, 2003.

[10]"Handwritten Digit Recognition Using K-Nearest Neighbour Classifier": U Ravi Babu, Dr. Y Venkateswarlu, Aneel Kumar Chintha.