IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 3)
Available online at: www.ijariit.com

A machine learning approach for handwritten recognition

Vanamoju Vandana
vanamojuvandana130699@gmail.com
Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh

Yedlapalli Divya Manasa
divyavenkat799@gmail.com
Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh

Gedela Gayathri
satyagayathri123@gmail.com
Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh

Komakula Sai Lakshmi Priyanka
priyanka.kmkl@gmail.com
Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh

M. Sion Kumari
msmondru@gmail.com
Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh

ABSTRACT

Handwritten recognition tends the user to process the input to the machine (which is trained) so that this well-trained machine recognizes the input and generates the corresponding output. There are various input sources like paper documents, texts, images, etc., that can be processed in emerging areas like banking, industrial, educational sectors. In this paper, we designed an expert system that process the input from user and generate the output basing on various classification algorithms and choosing the best algorithm meeting the criteria of achieving highest accuracy.

Keywords— *Neural Networks, Random Forest Classifier, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Classification*

1. INTRODUCTION

Handwriting recognition is the ability of an expert system to receive the handwritten input from various input sources like paper documents, photographs, touch screen devices etc. Recognition of handwritten characters is most widely used application in many areas and finds its vast applications in banks, offices and industries. The main aim of this project is to design an expert system (i.e. trained) for, "Hand Written Recognition" that can fruitfully recognize a particular character or digit of type format using the Artificial Neural Network approach and various other classification algorithms. Neural computing is used for classification method (using neural networks) and it has less design components. Neural computers will perform data parallelism. Neural computing is a way different from that of normal computing. They are trained (not Programmed), so that given a data input (starting state); they either classify the input data into any of the classes or make sure that certain property is optimized.

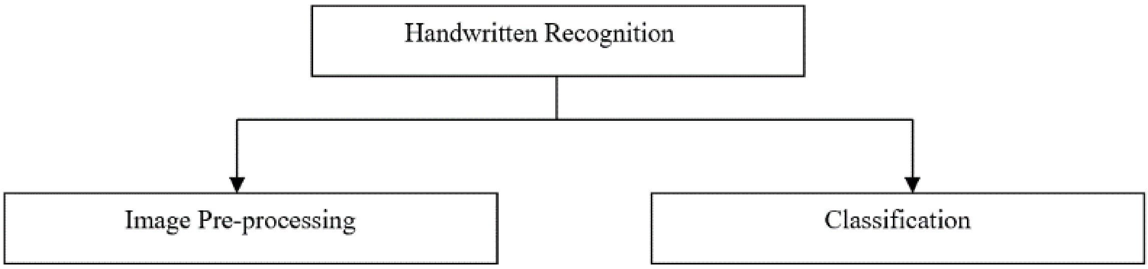
2. EXISTING SYSTEM

In the existing system, only the digits are recognized and the accuracy is also less .This system developed as an application and cannot be able to recognize characters. For this system, they used python, openCV and sklearn to run classification and read the dataset. MNIST dataset for training and evaluation for classification. MNIST problem is a dataset provided for evaluating machine learning models on the handwritten digit classification problem. The dataset was constructed from a number of scanned documents dataset that is available from the National Institute of Standards and Technology (NIST). Each image in this dataset is a 28X28 pixel square (748 pixels total). The limitations of this existing system can be achieved by implementing the proposed system. There are 70000 images in the dataset that can be used for training and evaluate the system.

3. PROPOSED SYSTEM

For this system, we used python, Keras, TensorFlow and sklearn to run classification algorithm and read the dataset. We used MNIST and Extended-MNIST(EMNIST) dataset for training, evaluation, classification and prediction. MNIST problem is a dataset for evaluating machine learning models on the handwritten digit classification problem. EMNIST problem is a dataset for evaluating machine learning models on the handwritten character classification problem. The MNIST dataset was constructed from a number of scanned documents dataset available from the National Institute of Standards and Technology (NIST). Each image in this dataset is a 28X28 pixel square (748 pixels total). There are 70000 images in the dataset that can be used for training and to evaluate the system. The EMNIST dataset consists of character images for training and testing. In this proposed system we used various classification algorithms for the recognition the characters which is Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Convolution Neural Network (CNN). The accuracy is eventually increased in the Proposed system.

4. WORKING PROCEDURE



4.1 Image Pre-processing

After training the Neural network model, an accuracy of 94 % was obtained. Now let us discuss the various image processing operations which are performed on the image to be recognized. Following steps are involved in processing of images: This is the first step performed in image processing. In this step the noise from the image is removed by using median filtering. It is one of the most widely used noise reduction technique. This is because in median filtering the edges in image can be preserved while the noise is still removed.

4.1.1 Conversion to Gray-Scale: After the removal of noise from image, the image is converted into grayscale. Conversion into grayscale is necessary because different writers use pens of different colors with varying intensities. Another advantage is the complexity of the system can be reduced. The preprocessing stage enhances the quality of the input image by excluding the unwanted data from it. It is also termed as pixel level or low-level processing. Handwritten Recognition Image Pre-processing includes various activities like removal of dust, spots, dots, or lines, which are termed as noise that may affect the recognition results to a large extent. Image enhancement methods are applied to improve the quality of the image in both machine and human perspective. Smoothing and non-linear operations, such as morphological operations, are used for noise removal.

4.1.2 Thresholding: Soon after the image is converted into grayscale, the handwritten text is darker as compared to its background. With the help of thresholding, we can separate the darker regions of the image from the lighter regions. Thus because of this thresholding the image is separated from background.

4.1.3 Image Segmentation: A user can write text in the form of lines. Thus, the threshold image is first segmented into individual lines. Then each individual line is segmented into individual words. Finally, each word is segmented into individual letter (character or digit).

4.1.4 Binarization: The conversion of the grayscale image to black and white is called binarization. Binary images are also called as Bi-level or two-level. First, the original RGB image (input image) should be converted to grayscale and then the image is converted to black and white image. Most of the OCR packages work on these binarized images. The conversion is done using the criteria threshold value (the values which are higher than the threshold is white and the values which are lower than this threshold are black). Otsu's method is used to perform threshold based on cluster i.e. from gray level image to binary image. The threshold value 0.5 yields better for all type of images. The laser printer, fax machines can handle the binarized images.

4.1.5 Thinning: Thinning is a pre-process which results in single pixel width image to recognize the handwritten character easily. It is applied repeatedly leaving only pixel-wide linear representations of the image characters.

4.2 Classification

4.2.1 Recognition: Recognition of handwritten characters or digits is extremely difficult problem. The alphabets and digits might be scripted in various size, direction, width, arrangement and measurement. This may lead to provide differences. The potential of neural network (NN) is to simplify the input and might be extremely helpful in handwritten alphabets and digits. The feature vector is written as X where X = (f1, f2,...,fd) where f indicates features and d is the no. features removed from English alphabets. On the basis of similarity of feature vector, English alphabets are powerfully categorized into suitable category and acknowledged. Classifiers works on basis of two types of learning techniques:

- **Supervised learning:** In supervised learning, training information with accurate specification of category is to be given to train a novel model. This novel model is utilized to test information for appropriate categorization. Training information comprises both the input and the required results. The novel model suffers learning procedure and on this basis it learns & classifies test information. For example: HMM, SVM etc.

- **Support Vector Machine:** It is one of the classification algorithms which is used for input handwritten recognition. It makes use of hyper plane which consists of margin that can classify the data points into various classes.

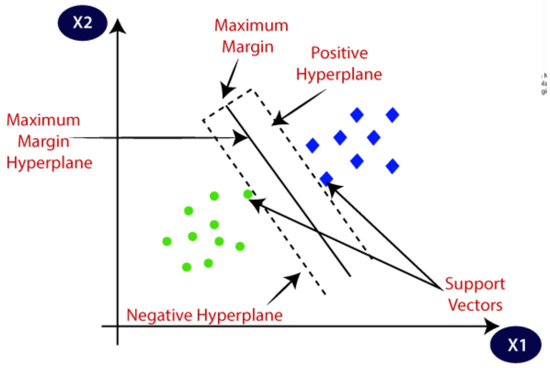
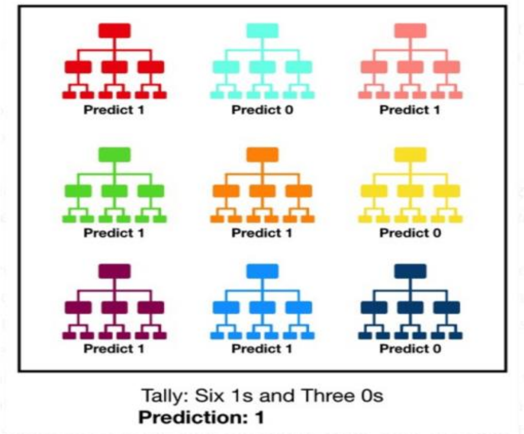


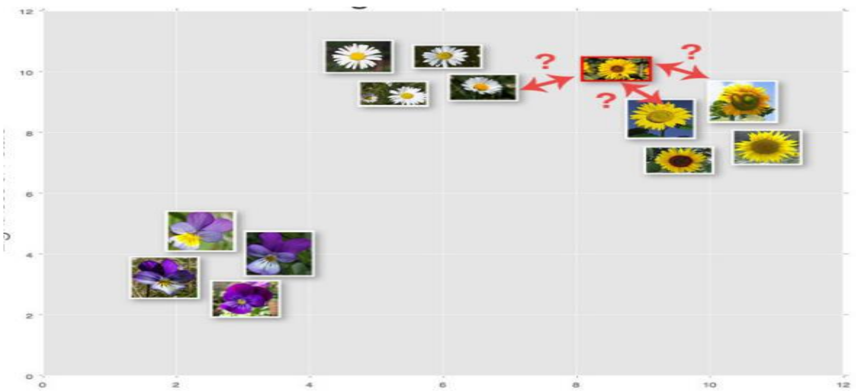
Fig. 1: SVM Classification

To separate the two classes of data points, there are many possible hyperplanes that can be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some benefit so that future data points can be classified with more confidence. The SVM in scikit-learn support both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. Scikit-learn have three classes that capable of performing multiclass classification on the given dataset which are SVC, NuSVC and LinearSVC. In this system we will use LinearSVC class to perform the classification of MNIST dataset. LinearSVC or Linear Support Vector Classification that use linear kernel and implemented in terms of liblinear that has more flexibility in the choice of penalties and loss functions and should scale better large numbers of samples.

Random Forest Classifier: Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees implies more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.



K-Nearest Neighbors (KNN): K-nearest neighbours (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. KNN classifier is the simplest image classification algorithm. KNN classification doesn't actually learn anything. This algorithm relies on the distance between feature vectors. KNN algorithm classifies unknown data points by finding the most common class among the k closest examples using Euclidean distances. Each data point in the k closest provides a vote and the highest category number of votes may have the chance to win.



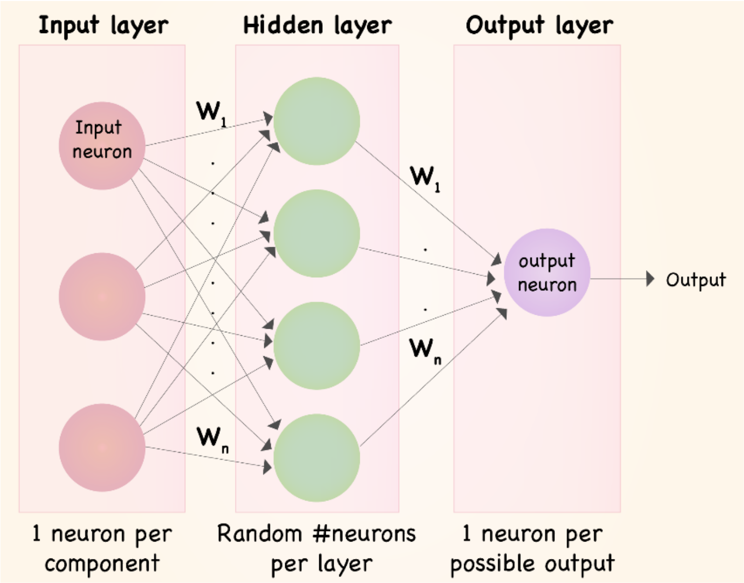
KNN Classification

- **Unsupervised learning:** Unsupervised learning technique is not granted for training data. It doesn't need learning. The technique categorizes test information on basis of statistical characteristics and by their spatial alliance and consider their closest neighbour. For example: k means, Clustering, etc.

Neural Networks: A neural network is defined as a computing system that consist of a number of simple but highly interconnected elements or nodes, called ‘neurons’, which are organized in layers which process information using dynamic state responses to external inputs.

In the structure of neural networks, there are neurons present at each cell. The input layer contains the individual neurons that are to be classified into their respective classes by traversing through various hidden layers. The actual process takes place in these hidden layers. At each neuron, they possess the properties weights and biases. At each neuron in the hidden layers, we calculate the sigmoid function, that takes the weights and biases as their values to process. At the last hidden layer, these values for the sigmoid function can be used to find their respective outputs. A perceptron uses a function to learn the classifier so that many input nodes can point to a single output node and it can also be used in supervised learning. In this context, the perceptron follows these steps:

1. Each input is multiplied to their corresponding weight.
2. Add all those calculated multiplications and sum up them. (i.e. ***weighted sum:*** $\sum w_j x_j$)
3. Apply the activation function, i.e. determining whether the weighted sum is greater than a threshold value, where -threshold is equivalent to bias, that ultimately assigns an output if exists.



The perceptron function is:

$$f(x)=\begin{cases} 1 & \text{if } w.x + \overset{\text{bias = - threshold}}{b} \geq 0 \\ 0 & \text{if } \underbrace{w.x}_{\sum_j w_j x_j} + b < 0 \end{cases}$$

Here b is the bias and is equivalent to -threshold, w.x is the dot product of w (a vector whose component are the weights) and x (a vector consisting of inputs). The output for the formula is:

$$\sigma = \frac{1}{1 + e^{-\sum_j w_j x_j - b}}$$

4.2.2 Training: In the training phase, the correct class for each record is determined so that if the input node that is present in the correct class is termed as 1 and the input that is not present in the class is termed as 0. The system must be well trained so that the input nodes must always leads to the correct classes rather than to non-correct classes. We must provide the training data in bulk amounts in order to obtain a system which leads to generate the outputs through an efficient manner.

4.3 Performance

The results are as we expected and the best outputs are being generated by comparing the various algorithms and their respective accuracy so as to choose the best technique to generate the rich outputs.

Model	Accuracy
Support Vector Machine (SVM)	97%
Random Forest Classifier	93.4%
K-Nearest Neighbors (KNN)	98.2%
Neural Networks	99%

5. CONCLUSION

There are many developments possible in this system in the future. As of now the system cannot recognize cursive handwritten text. But in future we can add support for recognition of cursive text. Currently our system can only recognize text in English languages. We can add support for more languages in the future. Presently the system can only recognize letters and digits. We can add support for recognition of Special symbols in the future. There are many applications of this system possible. Some of the applications are Processing of cheques in Banks, helping hand in Desktop publishing, Recognition of text from business cards, Helping the blind in recognizing handwritten text on letters. This Neural network method gives the accuracy of 99.17%. The paper provides a useful method for the recognition of handwritten characters to a great extent. The proposed method has been applied on different unknown characters. It is hoped that this insight will be beneficial into various concepts involved, and boost further advances in the area. The accuracy of recognition is depending on the nature of the material to be read and its quality. This project establishes a system that converts scanned images of handwritten characters to text document.

6. REFERENCES

[1] [1] Hands-on Machine Learning with Scikit-Learn&Tensorflow, concepts, tools, and techniques to build intelligent systems powered by Jupyter by AurelienGeron, published by O'REILLY Media

[2] Introduction to Machine Learning with Python, A Guide for scientists, powered by Jupyter, by Andreas C. Muller & Sarah Guido, October 2016, published by ccO'REILLY Media

[3] https://pypi.org/project/scikit-learn/ sklearn and scikit for module references

[4] https://www.kaggle.com/ for few datasets.

[5] Wei Lu, ZhijianLi,Bingxue Shi . "Handwritten Digits Recognition with Neural Networks and Fuzzy Logic" in IEEE International Conference on Neural Networks, 1995. Proceedings.,

[6] SihamTabik, Daniel Peralta, Andrs Herrera-Poyatos, Francisco Herrera. "A snapshot of image Pre-Processing for convolutional neural networks: Case study of MNIST" in International Journal of Computational Intelligence Systems 10(1):555 January 2017.

[7] "An open-source machine learning framework for everyone" https://www.tensorflow.org/,[Online] Available: https://www.tensorflow.org/. [Accessed 05 March 2018]

[8] "NIST Special Database 19" https://www.nist.gov, [Online]. Available: https://www.nist.gov/srd/nist-special-database-19. [Accessed 05 March 2018].

[9] OpenCV" https://en.wikipedia.org,[Online] Available: https://en.wikipedia.org/wiki/OpenCV.