International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Special Issue - NGEPT - 2019

Handwritten Digit Recognition

Ms. Suma Y, Student, EPCET, Bangalore, India, suma.yg22293@gmail.com

Dr. Anitha N, Professor, EPCET, Bangalore, India, anitha.mhp@gmail.com

Ms. Maria Supreetha, Student, EPCET, Bangalore, India, maria276maria@gmail.com

Ms. Shilpa H, Student, EPCET, Bangalore, India, shilpahnaik97@gmail.com

Ms. Babitha M, Student, EPCET, Bangalore, India, babithamohan810@gmail.com

Abstract: Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, I compare the results of some of the most widely used Machine Learning Algorithms like SVM, KNN & RFC and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow. Using these, I was able to get the accuracy of 98.70% using CNN (Keras+Theano) as compared to 97.91% using SVM, 96.67% using KNN, 96.89% using RFC.

Keywords – CNN, Deep Learning, KNN, MNIST data set, RFC, SVM

I. INTRODUCTION

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas are signature verification, bank check processing, postal address interpretation from envelopes etc. A lot of classification techniques using Machine Learning has been developed and are used for this like K-Nearest Neighbors, SVM Classifier, Random Forest Classifier etc. but these methods although have the accuracy of 97% this is not enough for the real world applications. One example of this is, if you send a letter with addressee named as “Anuj” and the system detects it and recognizes it as “Tanuj” then it will not be delivered to “Anuj” but it will be delivered to “Tanuj”. Although eventually it may come to the right address but if the mail is important, this delay can cost us a lot. In short, the accuracy in these applications is very less but these techniques do not provide the required accuracy due to very little knowledge about the topology of a task. Here comes the use of Deep Learning, In the past decade, deep learning has become the best tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine learning tools have been developed like scikit-learn, scipy-image etc. and pybrains, Keras, Theano, Tensorflow by Google, TFLearn etc. for Deep Learning. These tools makes the applications robust and therefore much accurate. The Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field.

II. LITERATURE SURVEY

Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis (August 2003)

- This paper illustrates a set of conjugate best practices that document analysis researchers can use to get better results with artificial neural networks.
- The most important practice is to train data set as large as possible. We can expand the training set by adding a new form of distorted data.
- The next important practice is that convolutional neural networks are well suited for visual document tasks than the fully connected networks.
- We will illustrate our claims on the MNIST data set of English digit images.

Unconstrained handwritten numeral recognition using majority voting classifier. In Parallel Distributed and Grid Computing (2012)

- This paper presents a simple profile of combined local and global features and majority voting scheme classifier essential for unconstrained handwritten numeral recognition.
- This simple profile feature is computed by using the left, right, top and bottom profile of an image. A feature vector of length 112 is formed when all the profiles being combined.
- A feature vector of length 80 is formed when only 64 local and 16 global features are combined. The feature vector is the intensity of a pixel in the third level approximation of the component of an image.

 DOI : 10.18231/2454-9150.2019.0595

- The system is based on two stages, the first stage is to extract the roots from text and the second stage is to group the text according to predefined categories. The linguistic root extraction stage is composed of two main phases.

III. EXISTING SYSTEM

A lot of classification techniques using Machine Learning have been developed and used for this like K-Nearest Neighbors, SVM Classifier, Random Forest Classifier etc. but these methods although having the accuracy of 97% are not enough for the real world applications. One example of this is, if you send a letter with addressee name as "Anuj" and the system detects and recognizes it as "Tanuj" then it will not be delivered to "Anuj" but "Tanuj". Although eventually it may come to the right address but if the mail is important, this delay can cost a lot. In short, the accuracy in these applications is very critical but these techniques do not provide the required accuracy due to very little knowledge about the topology of a task

IV. PROPOSED SYSTEM

Proposed system consists of database modelling, preprocessing and classification using machine learning techniques. Following are the Database modelling steps

1. Read the dataset files from the MNIST dataset using python as „rb". 2. For each file, there will be a specific magic number. Take this files one by one and read them if the condition for the magic number is satisfied i.e. if the magic number matches the type of the file. For example, to read the Training Set Image data file, you need to check first if the magic number is equal to 2051 else do not read the file for training labels. 3. Read the number of rows and columns provided in the data file in next row to the magic number. 4. The using this information, read the 28x28 data corresponds to the respective label provided in the row-wise format. 5. Follow the above steps for the rest of the files and put the respective data in the variables. The function that reads the image data returns the image information and the labels.

To show the working accuracy of Machine Learning algorithms, Here we are using three classifiers:

1. Random Forest Classifier [RFC]

2. K-Nearest Neighbors [KNN]

3. Supervised Vector Machine [SVM]

Classification using Random Forest Classifier (RFC)

For classification of the MNIST data, the RFC algorithm works as follows: 1. Load the MNIST data set. 2. Divide the data and label it as Training and Testing Image and labels. 3. Use cross validation technique to divide the training data into training and testing data to train the classifier. 4. Train the Classifier using RFC algorithm. Provide training data and labels as input to train the classifier. RFC requires the number of trees in forest, number of features to look for best split, maximum depth of the tree etc. as the input. 5. The digit recognized using RFC is then matched with the provided Training Labels to get the score/accuracy of the trained classifier. 6. This trained classifier is pickled again to use it on the testing data. 7. The Test Image data is used to predict the labels of digits and is compared with the provided test labels to see for the accuracy of the algorithm. 8. The Confusion Matrix is printed that provides the percentage of accuracy with which each digit that has been recognized.

Classification using K-Nearest Neighbors (KNN)

K-Nearest Neighbors is an algorithm in which the best estimate among all the values is the value that has maximum number of neighbors with smallest Euclidian or Hamming distance.

For classification, the KNN algorithm works as follows: 1. Load the MNIST data set. 2. Divide the data and label it as Training and Testing Image and labels. 3. Use cross validation technique to divide the training data into training and testing data to train the classifier. 4. Train the Classifier using KNN algorithm. Provide training data and labels as input to train the classifier. KNN uses the difference in distance between the actual points and the points provided to classify the digit in the image. 5. The digit recognized using KNN is then matched with the provided Training Labels to get the score/accuracy of the trained classifier. 6. This trained classifier is pickled again to use it on the testing data. 7. The Test Image data is used to predict the labels of digits and is compared with the provided test labels to see for the accuracy of the algorithm. 8. The Confusion Matrix is printed that provides the percentage of accuracy with which each digit that has been recognized.

Classification using SVM Classifier

In machine learning, support vector machine is an important model. It is a supervised learning model used for classification and regression

1. Load the MNIST data set. 2. Divide the data and label it as as Training and Testing Image and labels. 3. Use cross validation technique to divide the training data into training and testing data to train the classifier. 4. Train the SVM Classifier. Provide training data and labels as input to train the classifier. SVM uses the existing labelled data to learn and then classifies the unlabeled data based on that learning. 5. The digit recognized using SVM is then matched with the provided Training Labels to get the score/accuracy of the trained classifier. 6. This trained classifier is pickled again to use it on the testing data. 7. The Test Image data is used to predict the labels of digits and is compared with the provided test labels to see for the accuracy of the algorithm. 8. The Confusion Matrix is printed that provides the

 DOI : 10.18231/2454-9150.2019.0595

percentage of accuracy with which each digit that has been recognized.

V. SYSTEM ARCHITECTURE

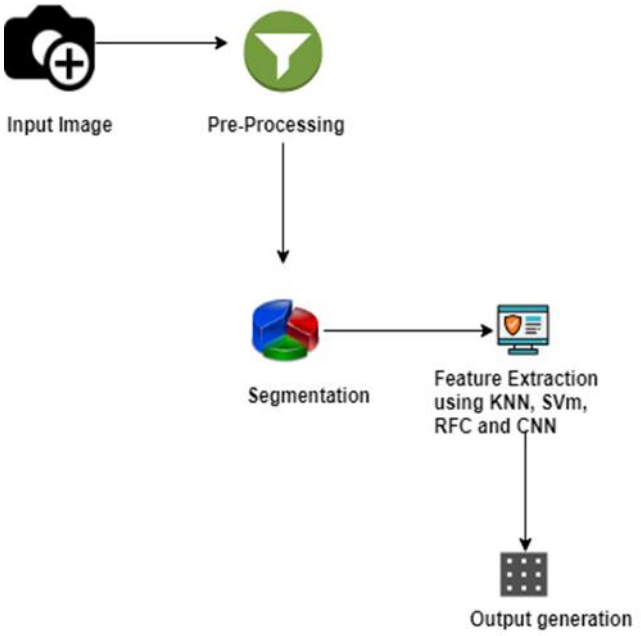

Fig 1: System Architecture

Input: The MNIST dataset, is a subset of a larger data set of NIST. It is a database of 70,000 handwritten digits, which are divided into 60,000 training examples and 10,000[2] testing samples. The images in the MNIST dataset are in the form of an array consisting of 28x28 values representing an image along with their labels.

Preprocessing: Pre-processing images are most important part while making programs that are related to image or data. When we are using learning networks to recognize images, The image must be in the same format for which we have trained the network. So pre-processing is very much important to make the image more precise and accurate. Steps in pre-processing can be resizing, cropping, changing hue, making black and white etc.

Segmentation and Feature extraction: A CNN consists of a lot of layers. These layers when they are used repeatedly, lead to a formation of a Deep Neural Network. Three main layers used to build a CNN are:

1. Input: This layer holds the raw pixel value of an image.

2. Convolutional Layer: This layer gets the results of the neuron layer which is connected to the input regions. We define the number of filters to has be used in this layer. Each filter may be a 5x5 window that is slider over the input data and gets the pixel with the maximum intensity as the output.

3. Rectified Linear Unit [ReLU] Layer: This layer applies an element wise activation function on the image data. We know that a CNN uses back propagation method. So in order to retain the same values of the pixels and not being changed or altered by the back propagation we use ReLU layer..

4. Pooling Layer: This layer perform a down-sampling operation along the spatial dimensions such as width and height that results in volume.

5. Fully Connected Layer: This layers is used for computation of the score classes i.e., which class has the maximum score corresponding to the input digits.

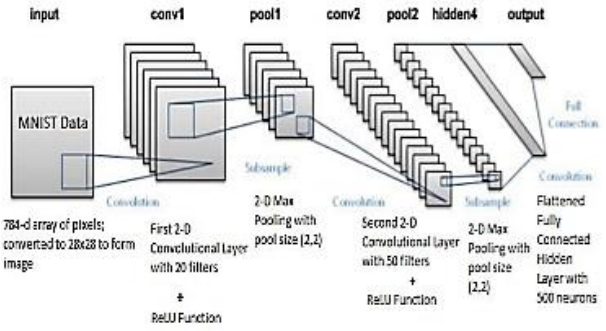

Fig 2: CNN Layers for Handwritten Digit Recognition

Convolutional Neural Network (CNN): Convolutional Neural Network (CNN) is a type of feed-forward Artificial Neural Network in which the connectivity pattern between its neurons is inspired by the way by which animal visual cortex is organized. Convolutional Neural Networks consists of neurons that have learnable weights and biases. Each neuron receives some input, performs a dot product operation and optionally follows it with a non-linearity.

Classification and recognition: The Test Image data is used to predict the labels of digits and it is compared with the provided test labels to see how accurate the algorithm is. We apply classifier that returns a list of probabilities for each of the 10 class labels. The class label with the largest probability is chosen as the final classification from the network and it is shown in the output. The output received is used to make the confusion matrix for the model. In this we add more number of layers but adding more layers might affect the accuracy of the system. Since, it uses multiple layers in the network.

VI. FUTURE ENHANCEMENT AND APPLICATIONS

We have restricted our study to only handwritten digit recognition. We can further enhance our study to handwritten characters with standard datasets.

A. Application to real world

- Bank check processing
- Extracting business card information in contact list
- Data entry for business documents like passport, invoice, bank statement and receipt.

 DOI : 10.18231/2454-9150.2019.0595

- Automatic insurance documents key information extraction.
- Number plate recognition

VII. RESULTS

The most accurate recognition of the digits, characters etc. along with the speed of recognition is of much great interest. The image below shows the accuracy comparison of the various techniques used by us for handwritten digit recognition.

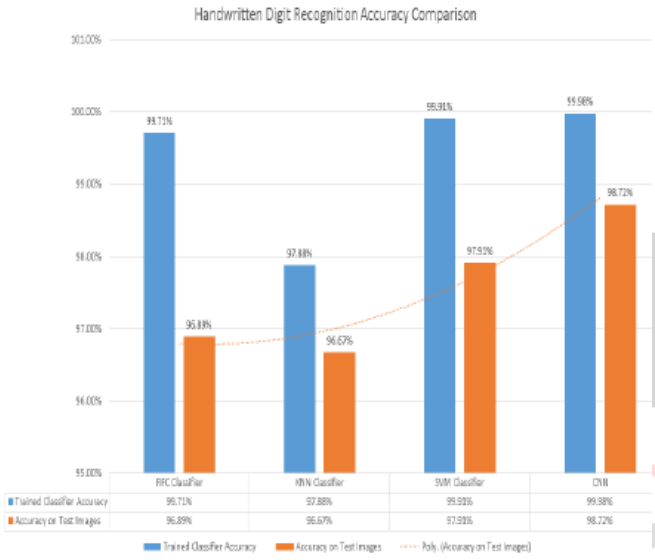

Fig 3: Accuracy Comparison of all Techniques

Fig 4: CNN Prediction Results on MNIST

	Trained Classifier Accuracy	Accuracy on Test Images
RFC	99.71%	96.89%
KNN	97.88%	96.67%
SVM	99.91%	97.91%
CNN	99.98%	98.72%

Table 1: Percent Accuracy of Each Classification Technique

We can see that using CNN with 3 hidden layers gives the accuracy of 98.72%. Although this accuracy might not be more optimal in today's world. Using Google's Tensorflow accuracy of 99.70% can be achieved.

VIII. CONCLUSION

We have implemented Handwritten Digit Recognition using Deep Learning. Additionally, some of the most widely used Machine Learning algorithms like RFC, KNN and SVM have been trained and tested on the same data to draw a comparison between deep learning methods in critical applications like Handwritten Digit Recognition. In this paper, we have shown that using Deep Learning techniques, a very high amount of accuracy can be achieved. Using the Convolutional Neural Network with Keras and Theano as backend we are able to get an accuracy of 98.72%. In addition to this, implementation of CNN using Tensorflow is given as an even better result of 99.70%. Every tool has its own complexity and the accuracy. Although, we see that the complexity of the code and the process is bit more as compared to normal Machine Learning algorithms but looking at the accuracy achieved, it can be said that it is worth it. Also, the current implementation is done only using the CPU. We have also implemented the same using CPU on EC2 instance using Amazon Web Service and we got similar results. For additional accuracy, to reduce training and testing time, the use of GPU's is required. Using GPU's we can get much more parallelism and we attain much better results.

REFERENCES

[1] Fatahi, M., 2014. MNIST handwritten digits.
[2] http://en.m.wikipedia.org/wiki/Support-vector_machine
[3]https://www.datacamp.com/community/tutorials/random-forests-classifier-python
[4] Jurgen Schmidhuber (2015). Deep Learning Scholarpedia
[5] Alsaad, A., 2016. Enhanced root extraction and document classification algorithm for Arabic text (Doctoral dissertation, Brunel University London).
[6] https://medium.com>technologymadeeasy
[7] Sebastian Raschka & Vahid Mirjalili, 2015. Python Machine Learning.
[8] Giuseppe Bonaccorso, 2018. Mastering Machine Learning Algorithms
[9] Charu C. Aggarwal, 2016. Neural Networks and Deep learning
[10] Mohamad H. Hassoun, 2012. Fundamentals of artificial neural networks
[11] https://towardsdatascience.com
[12] https://algorithmia.com>deep%20learning
[13] Aurelien Georn (2013). Hands-On Machine Learning with Scikit-Learn and Tensorfow: Concepts, Tools and Techniques to Build Intelligent System.

 DOI : 10.18231/2454-9150.2019.0595