Assessing Handwritten Digit Recognition Using Neural Networks

Alex Dineshbhai Christian
Lakehead University - Computer Science
Email: christiana@lakeheadu.ca
Student ID: 1133629

Abstract— **The ability of a computer to receive and interpret legible handwriting input from sources such as paper documents, pictures, touch-pads and displays, and other devices is referred to as handwritten recognition.The applications of optical character recognition (OCR) and handwritten character recognition (HCR) are numerous. Character recognition has been addressed in a variety of ways in the character recognition application.Character recognition has been addressed in a variety of ways in the handwriting recognition application. Handwritten Recognition, which uses different ways to transform textual content from a paper document into machine-readable form, has spawned a slew of studies and papers.The handwriting recognition problem was solved using this machine learning and computer vision software. As a result, character recognition systems will become increasingly important in the future as a means of scanning and processing existing paper documents in order to achieve a paperless environment.In this project, preprocessing methods, segmentation, and feature extraction will be discussed in this article. In this research project, different methods have been used such as Convolution Neural Network (CNN), Feed Forward Neural Network (FFNN), and Multi-layer Perceptron to implement and make handwritten digit recognition system. This problem is of Supervised Learning. I have expected outcome for it is to get the maximum accuracy and perfect prediction. I have used famous dataset MNIST.**

Keywords - Optical Character Recognition, Handwritten Digit Recognition, Handwritten Character Recognition, Neural Network

I. Introduction

One of the most challenging areas of machine learning and pattern recognition is the recognition of handwritten characters. Signature verification, mailing bank check processing, address comprehension, zip code recognition, documentation analysis, and many other applications are available. The process of detecting and recognizing characters in an image and translating them to a machine-readable representation is known as character recognition [1][2]. Computer systems called handwriting recognition systems can recognize anything written by hand, whether it's a letter, a symbol, or a digit. A considerable deal of effort has gone into developing effective techniques for approximating recognition from data by researchers working in machine learning and data mining [3]. One of the most difficult challenges in the field of digit identification is identifying the digit from which the greatest discriminating characteristics may be derived. In pattern recognition, several types of region sampling approaches are employed to find such areas [4].

There are two parts of handwritten recognition which is offline handwriting recognition and online handwriting recognition. If some handwriting is scanned and then interpret by the machine, that is called offline handwriting recognition.

By taking an example, when reading addresses on a mail piece, the first effort is focused on recognizing the postal code, which also happens to be the easiest to recognize using classic OCR approaches. Once this is accomplished, the recognized postal codes (an ordered list of top choices) are used to generate a list of states (provinces), cities and a lexicon of potential street names. Since addresses are written in a structured manner (beginning with the addressee, followed by street number and name, followed by city, state and postal code), recognition can be applied in a sequential manner. The sequence is typically postal code, state name, city name, street number, and street name. Only in very rare instances, addressee recognition is attempted. Similarly, a bank check has multiple fields that are highly connected. The courtesy recognizing the postal code, which also happens to be the easiest to recognize active payees for the specified account are all included. Words related to the legal limit are also limited to a tiny subset of the wider language. Correct extraction of the various fields, followed by accurate recognition of the data in those fields, are the main issues. Many commercial banks have had significant success with computerized check processing [5].

Handwritten digit identification has become quite popular in recent years. Developing such a model entail creating a computer that can recognize and categories pictures of handwritten digits as 10 digits (0–9), and the performance of such models has vastly improved as the error rate has been reduced using various classifiers and parameters.

This project implements different methodologies as:

1) Convolution Neural Network (CNN)
2) Multi-layer Perceptron
3) Feed Forward Neural Network (FFNN)

These methods elaborate different processing on singular Data-set resulting in the variations in the result depending upon the parameters and factors.

A. Dataset

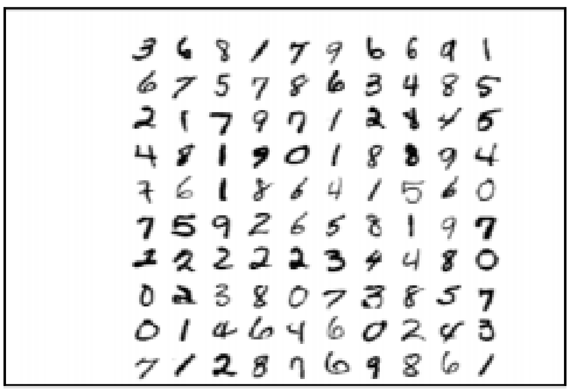

Fig. 1. MNIST Dataset Example

In this project, the MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. The MNIST database contains 60,000 training images and 10,000 testing images. Each sample image is 28x28 and linearized as a vector of size 1x784. So, the training and test data-sets are 2-d vectors of size 60000x784 and 10000x784 respectively.

B. Literature Review

In this paper, A. Challa [6] focuses on to build an "Automatic Handwritten Digit Recognition" method for the recognition of connected handwritten digit strings. The numerals were originally separated into separate digits. Then, to complete the handwritten digit string recognition problem, a digit recognition module is used to classify each segmented digit. Here, they have used different machine learning methods those are Support Vector Machine (SVM), Artificial Neural Network (ANN), and Convolution Neural Network (CNN) model with Histogram of Oriented Gradient (HOG) feature vectors and deep learning methods. Here, data is processed through the various preprocessing steps such as noise reduction, document skew correction, slant correction, normalization, smoothing, and skeletonization. This preprocessing output can be used as an input for feature generation, various extraction approaches are utilized in the feature generator to represent an image as a vector feature. To maintain it clear and free of noise, an algorithm is used to shrink the image, which minimizes the amount of noise in the image, then it followed by the classification and at the end it goes for the recognition part. After applying SVM, ANN and CNN machine learning model on the dataset, it can be shown that SVM and ANN have not performed well compared to CNN. SVM gives 39% accuracy, 37% accuracy level given by the ANN and CNN has the highest accuracy which is 71%, which is much efficient.

Arora S. *et al* [7], do their experiment of Handwritten character recognition on Devnagari languages uses multiple classifier combination. In their proposed method, they first performed are conversion of handwritten character to bit-mapped binary images, scaling of bitmap character and after that extracted three different features. Here, these researchers use three different feature sets. The features used are intersection, shadow feature and chain code histogram features. Shadow features are computed globally for character image while intersection features and chain code histogram features are computed by dividing the character image into different segments. After that, Multi-layer Perceptron with 3 layers applied on the extracted feature dataset to check the accuracy and error rate. This classifier simply trained with standard back-propagation technique, that minimizes the sum of squared errors for the training samples by conducting a gradient descent search in the weight space. By using combination of classifiers and MLP, model observed overall recognition rate of 92.16% on a dataset of 4900 samples.

Velappa Ganapathy and Kok Leong Liew [8] proposed a method in which multi-scale neural training with input training vector modifications is used to gain an advantage in training higher resolution character images, and then selective thresholding using the minimum distance technique is used to improve character recognition accuracy. The characters in a simulator software (a GUI) can be located on any point on the blank paper on which they are written. The results show that such methods with moderate level of training epochs can produce accuracies of at least 85% and more for handwritten upper case English characters and numerals.

In this paper, H.M.Balaha *et al* [11], proposed a handwritten character recognition on Arabic language. Here, they have taken large and complex Arabic Handwritten characters' dataset (HMBD) for their experiment. Deep learning system with two convolution neural network (CNN) architectures named as HMB1 and HMB2, with the appliance of optimization, regularization and dropout techniques have been used. There are sixteen experiments were applied to the described system using HMBD, and another two datasets: CMATER, and AIA9k. The outcomes of the experiments were recorded and compared to see how weight initializers, optimizers, data augmentation, and regularization affected over-fitting and accuracy. The highest accuracies were recorded by the "He Uniform" weight initializer and the AdaDelta (An adaptive learning rate method) optimizer. The accuracy of the data was improved by data augmentation. Using augmentation on HMBD, HMB1 reported testing accuracy of 98.4 percent with 865,840 records. The generalization was tested using the CMATER and AIA9k data-sets. The greatest outcomes for testing accuracies and data augmentation were 100 percent and 99.0 percent, respectively. In two rounds, a cross-over validation was done between the described architectures and a previous state-of-the-art architecture and dataset. For starters, the preceding control architecture does not apply to the dataset at hand. Second, the study found that the structures presented in the control data set generalize with better accuracies (97.3 percent , and 96.8 percent for HMB1, and HMB2, respectively)

II. Methods

The goal of this paper is to look at the architecture, block diagram, sequence diagram, data flow diagram, and user interface design options for the proposed system in order to describe stages such pre-processing, feature extraction, segmentation, classification, and digit recognition.

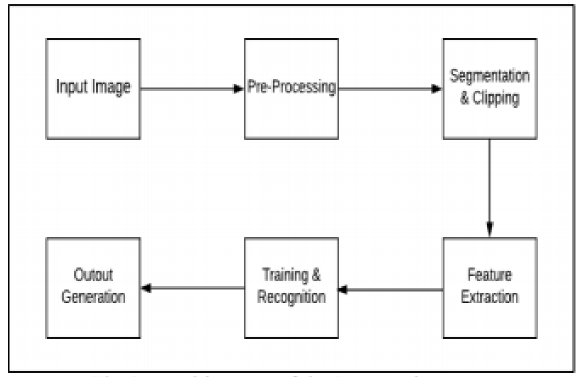

Fig. 2. Architecture

The above Fig. 2 illustrates the architecture diagram of the proposed system. The proposed model contains the four stages in order to classify and detect the digits:

A. Pre-Processing

The purpose of the pre-processing stage is to perform various tasks on the provided image. It effectively enhances the image by making it segmentation-ready. Pre-main processing's objective is to eliminate an intriguing as an example from the background. Filtering, smoothing, and standardising are the most common noise reduction techniques.The pre-processing also distinguishes a more compact representation of the case. Binarization is the process of converting a gray-scale image to a binary image.

B. Segmentation

The series of photos is utilised to produce sub-images of individual digits once the input photographs have been pre-processed. Digit images that have been pre-processed are split into sub-images of individual digits, each of which is given a number. A pixel is created for each digit. Images from a dataset are segmented using an edge detection method at this step.

C. Feature Extraction

The pre-processed images are represented in the form of a matrix comprising exceptionally large image pixels once the pre-processing and segmentation steps are completed. In this approach, depicting the digits in the photos that convey the essential information will be helpful. Feature extraction is the term for this process.

D. Classification and Recognition

Throughout the classification and recognition process, the gathered feature vectors are utilised as independent inputs to each of the subsequent classifiers. Extracted features are combined and defined using the three classifiers given below to showcase the working system model:

1) Convolution Neural Network (CNN): Convolution neural networks categorise and create output from visual input, generally 2D data, by using convolution, pooling, and fully connected layers. Convolution layers employ the convolution process between the input picture and the filter or kernel. The filter, or kernel, is also a 2D matrix that is in charge of creating feature maps utilising the local receptive field. The local receptive field is a tiny localised area of the input picture that is linked to a single neuron in the feature map. The number of feature maps required is proportional to the number of features to be categorised. After the feature map identifies the features, the kernel works as a weight matrix and learns the weights. The pooling layers simplify the output after convolution. There are two types of pooling: Max pooling and L2 pooling. Maximum activation output is pooled into a 2 x 2 input region in max pooling, and L2 pooling takes the square root of the sum of squares of the activation 2x2 region. Finally, the completely connected layers link the output neurons to each layer of the max pooling layer.

2) Feed Forward Neural Network (FFNN): The neurons in the input layer receive inputs and pass them on to the other layers. The number of neurons in the input layer should be the same as the data-set's characteristics or features. The anticipated feature is the output layer, which varies depending on the sort of model you're creating. Between the input and output layers is a hidden layer; the kind of model determines the number of hidden layers. A large number of neurons in the hidden layers make modifications to the inputs before passing them on. The weights are adjusted as the network is trained to make it more predictive. The intensity or amplitude of a connection between two neurons is referred to as neuron weights.You can compare weights on inputs like coefficients if you're familiar with linear regression. Weights are frequently set to tiny random numbers in the range of 0 to 1, for example.

3) Multi-layer Perceptron: To categorise the handwritten digits, a neural network-based classifier called Multi-Layer Perception (MLP) is employed. A multi-layer perceptron is made up of three layers: an input layer, a hidden layer, and an output layer.Each of the layers can have certain number of nodes also called neurons and each node in a layer is connected to all other nodes to the next layer [9]. As a result, it is also known as a feed forward network. The number of nodes in the input layer is determined by the number of characteristics in the dataset. The number of nodes in the output layer is determined by the number of visible classes in the dataset.It is difficult to establish the appropriate number of hidden layers or the convenient number of nodes in a hidden layer for a given task. However, in general, these numbers are chosen empirically. The link between two nodes in a multi-layer perceptron is made up of a weight.During training process, it basically learns the accurate weight adjustment which is corresponds to each connection [10]. It employs a supervised learning approach known as the Back propagation algorithm for learning purposes.

III. Results

Here, I have implemented Convolution Neural Network, Feed Forward Neural Network, and Multi-layer Perceptron for this project and I get different results each time. This paper shows, accuracy and loss of each neural networks and will see which one is outperform each other. I have run total 10 epochs on each model which will be discussed here.

A. CNN

In this approach, I have used convolution, Flatten, Dense layers for designing the CNN architecture. Here, I have used 128 neurons for creating a deeply connected layer in the neural network where each of the neurons of the dense layers receives input from all neurons of the previous layer and I have used ReLU activation function as an input layer whereas soft-max function is used for output layer. And model is being compiled using Adam optimization algorithm and using sparse categorical cross-entropy loss function to calculate loss. I have train the dataset and running 10 epochs on train data-sets and get model accuracy and loss on training and testing data-sets.

The Fig. 3 illustrates the how the training and testing accuracy increasing with the epochs and the Fig. 4 illustrates the how the training and testing Loss decreasing with the epochs.

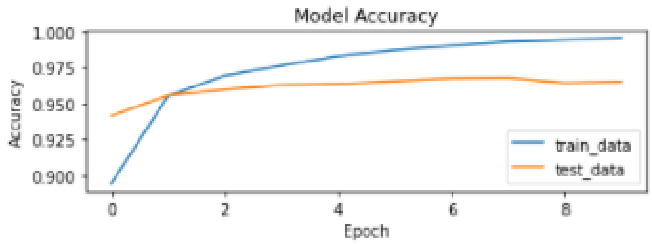

Fig. 3. CNN Accuracy

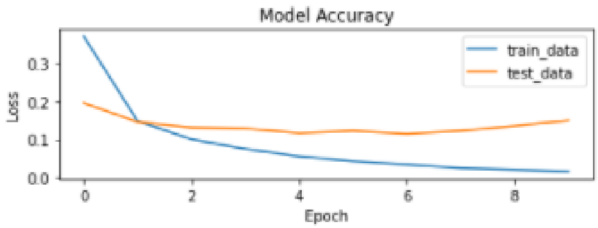

Fig. 4. CNN Loss

B. FFNN

In this approach, I have used sequential, Flatten, Dense layers for designing the FFNN architecture. Here, I have used 512, 256, 128, 64 and 10 neurons for creating a deeply connected layer in the neural network where each of the neurons of the dense layers receives input from all neurons of the previous layer and I have used ReLU activation function as an input layer whereas soft-max function is used for output layer. And model is being compiled using Adam optimization algorithm and using categorical cross-entropy loss function to calculate loss. I have train the dataset and running 10 epochs on train data-sets and get model accuracy and loss on training and testing data-sets.

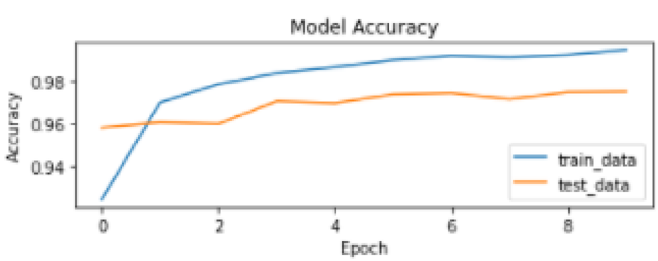

Fig. 5. FFNN Accuracy

The Fig. 5 illustrates the how the training and testing accuracy increasing with the epochs. The Fig. 6 illustrates the how the training and testing Loss decreasing with the epochs.

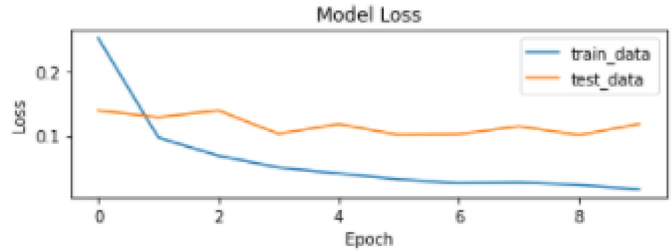

Fig. 6. FFNN Loss

C. Multi-layer Perceptron

In this approach, I have used sequential, Flatten, Dense layers for designing the FFNN architecture. Here, I have used 1000 with the dimension of 784, and 10 neurons for creating a deeply connected layer in the neural network where each of the neurons of the dense layers receives input from all neurons of the previous layer and I have used sigmoid activation function as an input layer whereas sigmoid function is used for output layer. And model is being compiled using Adam optimization algorithm and using categorical cross-entropy loss function to calculate loss. I have train the dataset and running 10 epochs on train data-sets and get model accuracy and loss on training and testing data-sets.

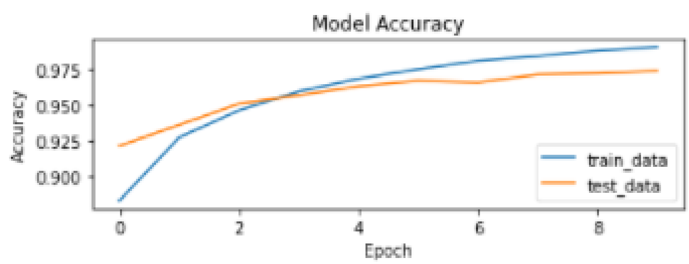

Fig. 7. MLP Accuracy

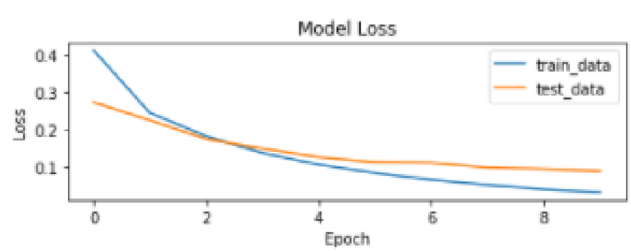

Fig. 8. MLP Loss

The Fig. 7 illustrates the how the training and testing accuracy increasing with the epochs. The Fig. 8 illustrates the how the training and testing Loss decreasing with the epochs.

IV. Discussion

If we talk about the Handwritten digit recognition, then there are many application and research papers available on it. I have reviewed few research papers and I am comparing with those with our own experiments.

A. Previous Research

- A. Challa's goal in this article is to develop a "Automatic Handwritten Digit Recognition" method for recognising linked handwritten digit strings.Authors utilised Support Vector Machine (SVM), Artificial Neural Network (ANN), and Convolution Neural Network (CNN) models using Histogram of Oriented Gradient (HOG) feature vectors along with deep learning methodologies in this study.Following the application of SVM, ANN, and CNN machine learning models to the dataset, it can be shown that SVM and ANN did not perform as well as CNN. SVM has a 39 percent accuracy level, ANN has a 37 percent accuracy level, and CNN has the greatest accuracy of 71 percent, which is considerably more efficient.

- Arora S. et al. employ many classifier combinations in their experiment of handwritten character recognition on Devnagari languages.They have used multi-layer perceptron with 3 layers applied on the dataset.By using the MLP they got overall 92.16% accuracy rate.

- Multi-scale neural training with input training vector modifications is used to gain an advantage in training higher resolution character images, and then selective thresholding using the minimum distance technique is used to improve character recognition accuracy, according to Velappa Ganapathy and Kok Leong Liew and they achieved at least 85% accuracy.

B. My Research

In this project, I have used CNN, FFNN, and Multi-layer perceptron on the inbuilt MNIST dataset of tensorflow. I found that no other research paper has used FFNN for handwritten digit recognition, I found that this method is much more efficient than other approaches that have been used in other research papers.

I also performed CNN and multi-layer perceptron on the same dataset. However, it is seen that MLP gives best results on the test dataset.

V. Conclusion

I have successfully implemented Handwritten Digit Recognition in this project. This project was all about recognizing any image from the dataset and gives one specific number.Hence I can conclude that I have successfully implemented the task of Handwritten Digit Recognition using the following 3 models Multi-layer Perceptron, CNN and Feed Forward Neural Network. From these three models, I get testing accuracy of 92.22% from Multi-layer perceptron, 96.48% has been achieved by Convolution Neural Network, and 97.61% got from the Feed Forward Neural Network. So, FFNN has been outperformed the other two models.This is a first attempt, and the goal of the article is to make it easier to recognise handwritten numerals by employing categorization algorithms.

Acknowledgment

I would want to convey my appreciation to Dr. Trevor Tomesh, without whose assistance, I would not have been able to complete this study. I am grateful to him since he provided appropriate research direction and information, as well as assistance by offering feedback on my research work.

References

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, "A Comparative study of GABOR feature and gradient feature for handwritten Chinese character recognition", International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2-4 Nov. 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp. 659-662, Jan-Feb 2012.

[3] Kloesgen, W., Zytkow, J. Handbook of Knowledge Discovery and Data Mining.

[4] Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., Basu, D. K. A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Applied Soft Computing, 12(5), 1592-1606, 2012.

[5] Shridhar, M., Houle, G. F., Kimura, F. (2009). Recognition strategies for general handwritten text documents. 2009 IEEE International Conference on Electro/Information Technology, 167–169.

[6] A. Challa, "Automatic Handwritten Digit Recognition on Document Images Using Machine Learning Methods," Dissertation, 2019.

[7] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D. K., Kundu, M. (2010). Multiple Classifier Combination for Off-line Handwritten Devnagari Character Recognition.

[8] Velappa Ganapathy, and Kok Leong Liew, "Handwritten Character Recognition Using Multi scale Neural Network Training Technique", Proceedings of World academy of Science, Engineering and Technology, vol. 29, ISSN 1307-6884, May (2007)

[9] Bhowmik, S., Roushan, M. G., Sarkar, R., Nasipuri, M., Polley, S., Malakar, S. Handwritten Bangla Word Recognition Using HOG Descriptor. In 2014 Fourth International Conference of Emerging Applications of Information Technology (EAIT), (pp. 193-197). IEEE, 2014.

[10] Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., Held, P. Multi-Layer Perceptrons. In Computational Intelligence (pp. 47-81). Springer London, 2013.

[11] Balaha, H. M., Ali, H. A., Saraya, M., Badawy, M. (2021). A new Arabic handwritten character recognition deep learning system (AHCR-DLS). Neural Computing Applications, 33(11), 6325–6367.