Handwritten Character Recognition Using Deep Learning

**Navaneeth Chinthakayala, Hema Sri Chirmani,
Uday Kiran Sangepu, Deekshitha Nandhi**

Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India
Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India
Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India
Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, India

*navaneethchinthakayala123@gmail.com, hemasrichirumani@gmail.com,
udaykirans2810@gmail.com, deekshithanandi.in@gmail.com*

1 Abstract

Handwritten character recognition is the detection of characters from images and documents and changes them in machine-readable shape for further processing. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize handwritten digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning techniques. Several machine learning algorithms namely, Convolutional Neural Network, Multilayer Perceptron, Support Vector Machine and K-Nearest Neighbor have been used for the recognition of digits. The main focus of this work is to investigate CNN capability to recognize the characters from the image dataset and the accuracy of recognition with training and testing.

Keywords*: Pattern Recognition, Handwritten Recognition, Character Recognition, Machine Learning, Off-Line Handwritten Recognition, MNIST Dataset, Machine Learning Algorithm, Neural Network, Classification Algorithm*

2 Introduction

Machine Learning and deep learning plays an important part in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and numerous further areas [1].
This composition presents recognizing the Handwritten Characters from the notorious MNIST dataset, comparing classifiers like KNN, SVM, ANN and complication neural network on base of performance, accuracy, time, sensitivity, positive productivity, and particularity with using different parameters with the classifiers [2].
To make machines more intelligent, the developers are diving into machine learning and deep learning ways. A human learns to perform a task by rehearsing and repeating it again and again so that it memorizes how to perform the tasks. Also, the neurons in his brain automatically spark and they can snappily perform the task they've learned. Deep learning is also veritably analogous to this. It uses different types of neural network infrastructures for different types of problems [3].
The Handwritten Character recognition is the capability of computers to recognize Handwritten Characters. It's a hard task for the machine because handwritten Characters aren't perfect and can be made with numerous different flavors. The handwritten Character

recognition is the result to this problem which uses the image of a Character and recognizes the Character present in the image [4].

2.1 Character Recognition System

Character recognition system is the working of a machine to train itself or recognizing the Characters from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of , numeric entries in forms filled up by hand and so on.

2.2 Problem Statement

The goal of this project is to create a model that will be able to recognize and determine the handwritten Characters from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the Characters, it can be extended to letters and an individual's handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

3 Literature Survey

In 1959, Research from Grimsdale in the field of word recognition, was the earliest endeavor to perceive the handwritten character. This research exhibited the utilization of examination by combination strategy being proposed by Eden [1]. He demonstrates that the role of individual handwriting is limited to the number of schematic highlights. This hypothesis was later used as a part of almost all strategies to support the methodologies in the field of text recognition. Amit Choudhary [2] demonstrated an Off-Line Handwritten Character Recognition using Features Extracted by using Binarization Technique. It helps to extract features obtained by Binarization technique for recognition of English language handwritten characters. This algorithm delivers outstanding classification accuracy of 85.62 %. Sonu Varghese Ketal [3] demonstrated Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition. In the first step we will start setting up character groups in different classes based on the number of corners, loops, bifurcations and endings. In the second step we identify the exact character carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptron in each hidden layer. Because of these disadvantages, a Character recognizer using the MLP structure may not produce the desired low error rate.

In the final step we are checking the probability of occurrence of the character in the given position on the basis of defined rules for the making of words. Recognition conducted in different stages improves the efficiency, rate of recognition and accuracy of the system. Parshuram M. Kamble [4] demonstrated handwritten Marathi character recognition using R-HOG Feature. The system has been tested with a large quantity of handwritten Marathi characters. From the results it can be concluded that the use of R-HOG based feature extraction method and FFANN based classification with high processing speed and accuracy is more accurate.

4 Methodology

Our proposed method is mainly separated into stages, preprocessing, Model Construction, Training & Validation, Model Evaluation & Prediction. Since the loading dataset is necessary for any process, all the steps come after it.

Figure 1. Steps in System Development

4.1 Import the Libraries

Libraries required are Keras, TensorFlow, NumPy, Pillow, Flask.

Keras: It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano.

TensorFlow: It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems.

NumPy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

Pillow: Pillow is a free and open source library for the Python programming language that allows you to easily create & manipulate Characteral images. Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports a wide variety of images such as “jpeg”, “png”, “bmp”, “gif”, “ppm”, “tiff”.

Flask: Flask is a web framework, it’s a Python module that let us develop web applications easily. It has a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

4.2 MNIST Data Set

We used MNIST as a primary dataset to train the model, and it consists of 70,000 handwritten raster images from 250 different sources out of which 60,000 are used for training, and the rest are used for training validation. Modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is extensively used for training and testing different systems. All the Characters are grayscale and positioned in a fixed size where the intensity lies at the center of the image with 28×28 pixels. Since all the images are 28×28 pixels, it forms an array which can be flattened into 28*28=784 dimensional vector. Each component of the vector is a binary value which describes the intensity of the pixel. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of the neural networks.

Figure 2. MNIST Data Set

4.3 Pre-Processing

Data pre-processing plays an important part in any recognition process. Data preprocessing is a data mining technique which is used to transfigure the raw data in a useful and effective format. To shape the input images in a form suitable for segmentation pre-processing is used. Data preprocessing is an essential step before building a model with these features. It generally happens in stages:

- Data quality assessment
- Data cleaning
- Data transformation
- Data reduction

4.4 Model Construction

Now, comes the delightful part where we eventually get to use the strictly set data for model building. Depending on the data type (qualitative or quantitative) of the target variable (generally referred to as the Y variable) we are moreover going to be building a classification (if Y is qualitative) or regression (if Y is quantitative) model.

Models that can be used for the project:

Support Vector Machine: Support vector machine is supervised learning model with associated learning algorithm that analyze data for classification and regression analysis. Though we say regression problems as well its best suited for classification. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

SVM can be of two types:

- Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line.
- Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line.

K-Nearest Neighbor: The k-nearest neighbor algorithm is a supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. KNN can be used for both classification as well as regression predictive problems. The following two properties would define KNN well:

- It is also called a lazy learner algorithm because it does not learn from the training set immediately rather it stores the dataset and at the time of classification, it performs an action on the dataset.

- KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Convolutional Neural Network: A Convolutional Neural Network is a Deep Learning algorithm that can take in an input image, assign biases to various aspects/objects in the image, and be able to differentiate one from the other. Convolutional neural networks are more often utilized for classification and computer vision tasks. A CNN is specifically used for image recognition and tasks that involve the processing of pixel data. The Convolutional Neural Network has five layers called as Input Layer, Convolution Layer, ReLU Layer, Pooling Layer and Fully Connected Layer. The output of particular layer will be taken as input by the successive layer. Finally, the output of fully connected layer will be the final output of a image in which the initial input is recognized by the system.

4.5 Training & Validation

After the construction of the model, the model has to be computed to train it with the available data set. Optimizers are used to compute the model. Optimizers are used to solve optimization problems by minimizing the function and controls the learning rate. For compiling the model, we are using Adam optimizer in our system. The Adam optimizer is used to reduce learning rate after specific epochs. The model only sees validation data for evaluation but does not learn from this data, providing an objective unbiased evaluation of the model. If the training and testing data is increased, we can get better validation. The training is limited up to 98% accuracy because, we are using real-world data for prediction. For validation of the model, the test data is used.

4.6 Model Evaluation & Prediction

For real-world image classification prediction, we need to do a little image pre-processing on the real-world images as model training was done with greyscale raster images. The steps of image pre-processing are :

- Loading image
- Convert the image to greyscale
- Resize the image to 28x28
- Convert image into a matrix form
- Reshape the matrix into 28x28x1

After pre-processing, we predict the label of the image by passing the pre-processed image through the neural network. For evaluation of a model's performance, the methods are divided into 2 categories: Holdout and Cross-Validation. Both methods use a test set to evaluate model performance.

5 Results and Discussion

The implementation of this model only depends on NumPy, OpenCV and TensorFlow imports. The input images are a gray-scale images. The 5 layers of CNN map the input images to a feature sequence. The CNN has more accuracy when compared to SVM and KNN classifiers for both the trained data and test data. Moreover, the KNN is giving less accuracy in comparison with the SVM and CNN. The error rate of KNN classifier on the test data which is higher when compared to both SVM classifier and CNN. An epoch is one complete forward and backward passage of data in the neural network. With the change in the number of epochs, the difference in the trained data accuracy, test data accuracy and cross-entropy loss can be observed.
We will get output as N/A, if that particular word is not in the trained data.

Figure 3. User Interface

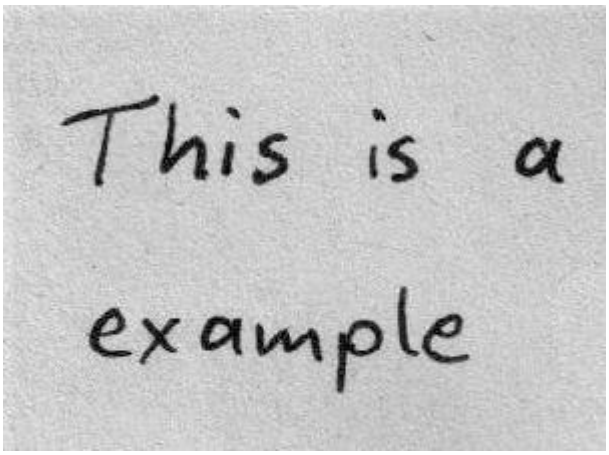

Figure 4. Input 1

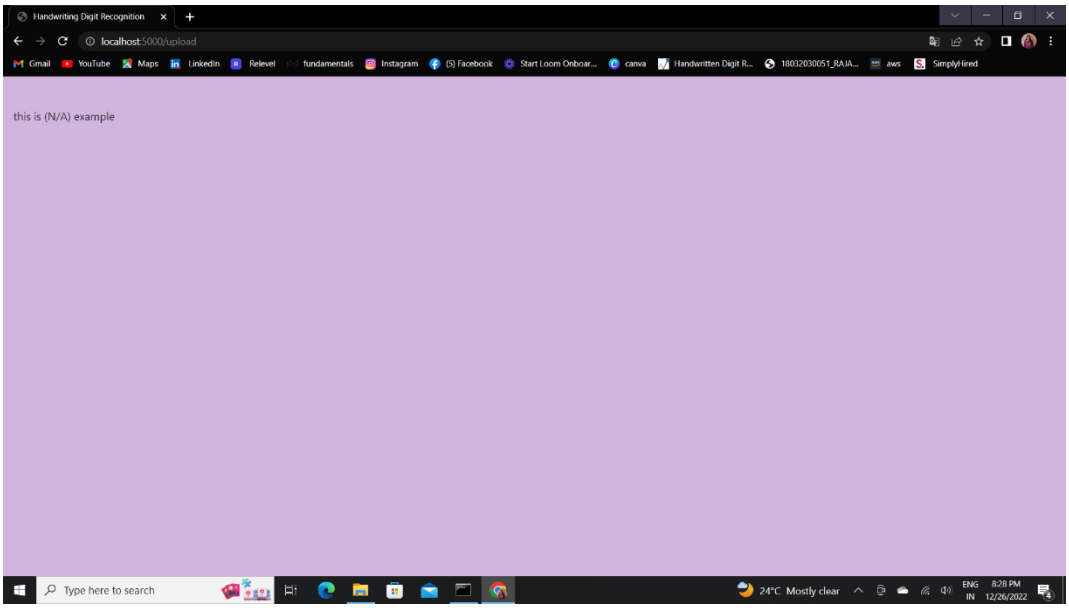

Figure 5. Output 1

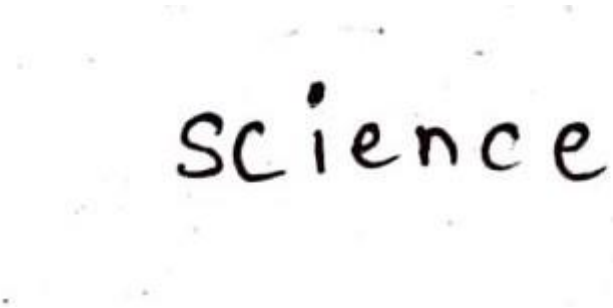

Figure 6. Input 2

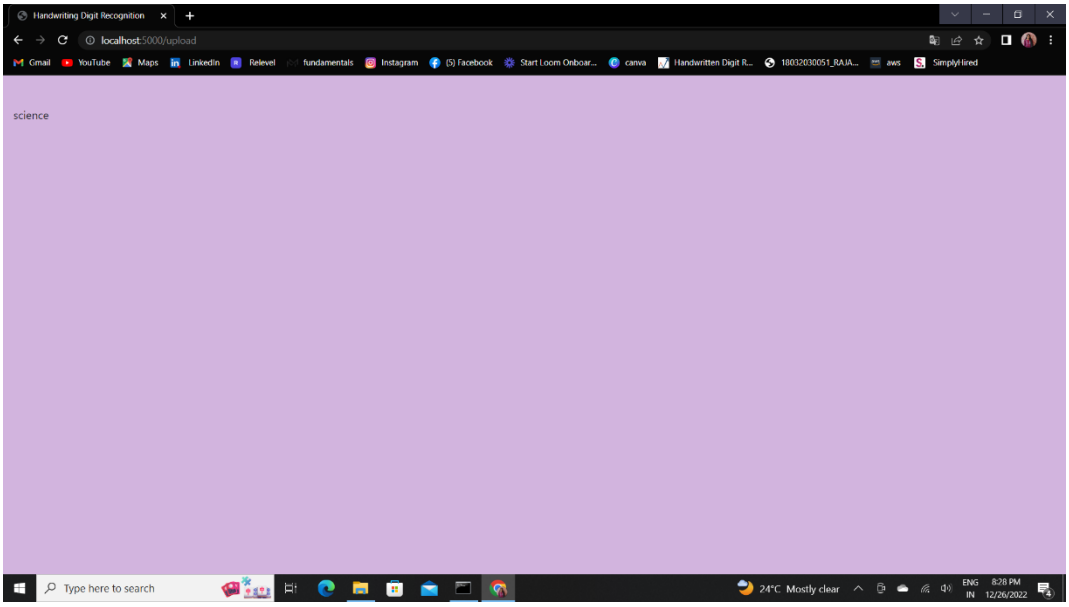

Figure 7. Output 2

6 Conclusion

The accuracy of text recognition fully depends on the quality and the nature of the image to be read. Current research does not deal with the cursive handwriting because it needs a high supervised system. The Convolutional Neural Network (CNN) model takes less time for training and error-rate is also less when compared to other models. The main purpose of this project is to build an automatic handwritten character recognition method for the recognition of handwritten character strings. In this project, different machine learning methods, which are SVM (Support Vector Machine), ANN (Artificial Neural Networks), and CNN (Convolutional Neural Networks) architectures are used to achieve high performance on the Character string recognition problem.

7 Acknowledgement

Thank you very much, Mrs. P Vijaya Lakshmi, Assistant Professor, Department of Computer Science a person who helped us in every difficult situation we faced during the creation of the project. I would also like to thank Department Faculty, Principal and the Management of Sreyas Institute of Engineering and Technology for giving us an opportunity.

8 References

[1] Shubham Sanjay Mor, Shivam Solanki, Saransh Gupta, Sayam Dhingra, Monika Jain, Rahul Saxena, “Handwritten text Recognition: with Deep Learning and Android”, Dept. of Computer Science and Engineering, Manipal University Jaipur, International Journal of Engineering and Advanced Technology (IJEAT), 2019, pp 819.

[2] Megha Agarwal, Shalika, Vinam Tomar, Priyanka Gupta, “Handwritten Character Recognition using Neural Network and Tensor Flow”, Computer Science and Engineering, SRM IST Ghaziabad, India, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019, pp 1445.

[3] Jemimah K, “Recognition of Handwritten Characters based on Deep Learning with TensorFlow”, Research Scholar, School of Computer Vol 11, Issue 4 , April/2020.

[4] R. Vijaya Kumar Reddy, U. Ravi Babu, “Handwritten Hindi Character Recognition using Deep Learning Techniques”, Dept. of CSE, Acharya Nagarjuna University, Guntur, India, International Journal of Computer Sciences and Engineering, 2019, pp 1.

[5] V. V. Mainkar, J. A. Katkar, A. B. Upade and P. R. Pednekar, \"Handwritten Character Recognition to Obtain Editable Text,\" 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 599-602, doi: 10.1109/ICESC48915.2020.9155786.

[6] S. M. Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair "Handwritten digit recognition using machine learning algorithms." Global Journal Of Computer Science And Technology (2018).

[7] H. Zeng,(2020)An Off-line Handwriting Recognition Employing Tensorflow. International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE).

[8] A. B. Siddique, M. M. R. Khan, R. B. Arif, and Z. Ashrafi, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm," in 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018, pp. 118-123: IEEE.

[9] H. A. Shiddieqy, T. Adiono and I. Syafalni, \"Mobile Client-Server Approach for Handwriting Digit Recognition\",2019 International Symposium on Electronics and Smart Devices (ISESD), 2019, pp. 1-4, doi: 10.1109/ISESD.2019.8909448.

[10] R. Vaidya, D. Trivedi, S. Satra and P. M. Pimpale, \"Handwritten Character Recognition Using DeepLearning\",2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 772-775, doi: 10.1109/ICICCT.2018.8473291.

[11] H. Du, P. Li, H. Zhou, W. Gong, G. Luo and P. Yang, \"WordRecorder: Accurate Acoustic-based Handwriting Recognition Using Deep Learning,\"IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 2018, pp. 1448-1456, doi: 10.1109/INFOCOM.2018.8486285.

[12] Ahmed Mahdi Obaid, IIHazem M. El Bakry, IIIM.A. Eldosuky, IVA.I. Shehab "Handwritten text recognition system based on neural network." Int. J. Adv. Res. Comput. Sci. Technol.(IJARCST) 4.1 (2016): 72-77.