



Handwritten Character Recognition Using Keras

Akash S¹, Anu T², Nasseena N³, Nitheesh T S⁴, Vineetha V C⁵, Vinod P R⁶

U.G. Student, Department of Computer Engineering, College Of Engineering, Chengannur, Kerala, India^{1, 2, 4, 5}

Assistant Professor, Department of Computer Engineering, College Of Engineering, Chengannur, Kerala, India^{3, 6}

ABSTRACT: In our project, we are implementing Handwritten Character recognition (HCR) system on the EMNIST dataset using **Keras** deep learning model for Python and **TensorFlow** at the backend. Keras is designed to be a simple, fast, minimal structure, and consistent Python-based library, which is best for deep learning applications. Here we are recognizing 26 uppercase and 26 lowercase English alphabets plus digits from 0-9. We used Convolutional Neural Networks since it always shows better accuracy for classification purposes compared to traditional methods. 90.35% test accuracy is obtained after training the model.

KEYWORDS: Handwritten Character Recognition, Convolutional Neural Network, relu Activation Function, Softmax Activation Function, Deep learning.

I. INTRODUCTION

Handwritten Character Recognition methods can be generally classified as two types- **Offline and Online**. In the Offline HCR system, the input is a scanned image. In it, the computer understands the image of the handwritten script automatically. But in Online HCR systems, the input is interpreted by the movements of a pen on a digitizing tablet or we can give input using mouse or touch. In this project, we are implementing Online Handwritten character recognition.

i. Keras Deep Learning Model

Keras is a simple, fast minimalist, highly modular neural networks library, written in Python and it is capable of running on top of TensorFlow. **TensorFlow** is a software library developed by Google Brain Team for deep learning and artificial intelligence researches. It combines the computational algebra of compilation optimization techniques, which makes the calculations of mathematical expressions easy.

ii. Convolutional Neural Network (CNN)

In this project, we are using the Convolutional Neural Network which consists of an input layer, one output layer, and multiple hidden layers. To build a CNN, we use three important layers- Convolutional, Pooling, and Fully Connected Layers.

The first layer of CNN is always a **Convolutional Layer**. A **Convolutional Filter**, as it is sliding, or **convolving**, around the input, it is multiplying the values in the filter with the original pixel values of the image. After sliding the filter over all the locations, we are left with an array of numbers usually called an **activation map** or **feature map**.

After each convolutional layer, we are using a *nonlinear layer*. **ReLU (Rectified Linear Unit)** immediately afterward. The ReLU activation function is a piecewise linear function. If the input is positive, it will output the input as itself, otherwise, it will output zero. After some ReLU layers, it is customary to apply a **pooling layer**. It progressively reduces the spatial size of the representation to reduce the number of parameters. One of the common **pooling** operation-Max-pooling calculates the **maximum**, or largest, the value in each patch of each feature map. Fully connected layers are given before the classification output of a CNN and are used to flatten the results before classification, which is practically a multilayer perceptron.

iii. The Dataset

We use the EMNIST (Extended MNIST) dataset for character recognition which is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and the dataset structure that directly matches the MNIST dataset. [8]. Figure(1) shows the EMNIST dataset.

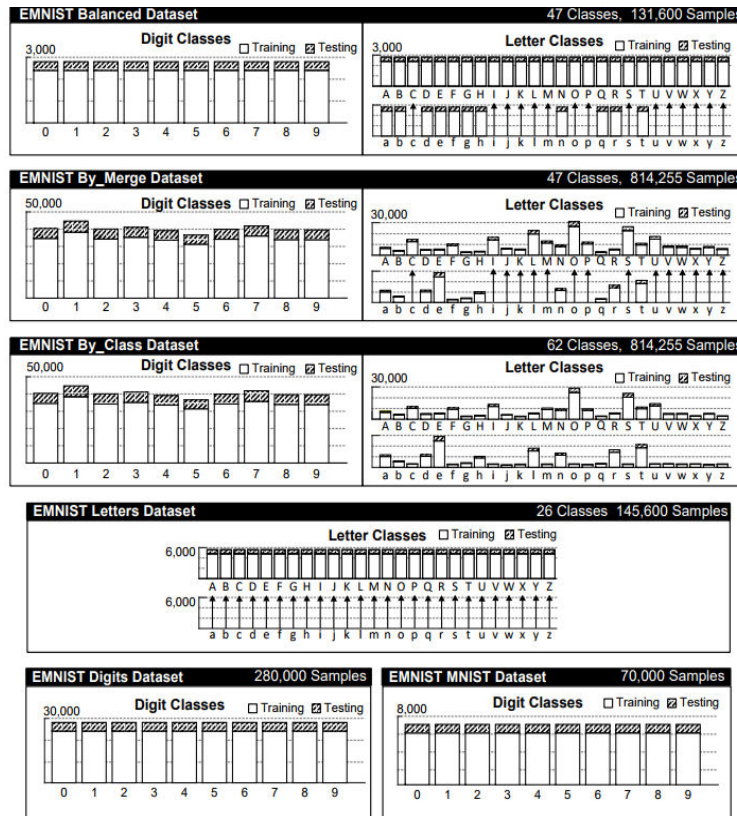


Figure (1): EMNIST dataset

This dataset consists of six splits. Given below is a summary of the EMNIST dataset

- EMNIST ByClass: 814,255 characters. 62 unbalanced classes.
- EMNIST ByMerge: 814,255 characters. 47 unbalanced classes.
- EMNIST Balanced: 131,600 characters. 47 balanced classes.
- EMNIST Letters: 145,600 characters. 26 balanced classes.
- EMNIST Digits: 280,000 characters. 10 balanced classes.
- EMNIST MNIST: 70,000 characters. 10 balanced classes.

For the proposed model, we are using the EMNIST By Merge split of the dataset.

II. RELATED WORK

[1] The paper explains about various steps like image acquisition, pre-processing, segmentation, feature extraction, etc. Each character is resized. Then the feature extraction phase takes place. The extracted features are given to this neural network for training.

[2] This paper is all about handwritten character recognition in Malayalam scripts. It also defines the components of a general character recognition system. This paper explains both offline and online character recognition system.

[3] This paper is about predicting Hindi Numerals using SVM. Support vector machine classifiers are used here. mSVMs are statistical classifiers. They have gained immense popularity in recent years providing excellent recognition results in various applications. From the collected data two features named directed angle and curvature are extracted. Different schemes are used for their calculations. The final result is obtained by comparing these schemes.

[4] This paper mainly focuses on predicting the Telugu letters and Vothulu composed on a digital tablet or any display screen. It includes two substantial degrees. In this work, the handwritten character recognition (HWR) model for Telugu [South Indian language] has been developed with high accuracy, less training time, and time of classification. In this paper, certain classifiers are used like Bayesian, MLP, and CNN. Among them, CNN gives the highest accuracy. This implementation is very useful for south India's online, offline Telugu character recognition system.

[5] This is a survey about the Handwritten Character recognition techniques for the English alphabets. Here online character recognition is used. This paper clearly explains the general methods of character recognition. The different stages in text recognition are pre-processing, Line segmentation, Normalization, Feature Extraction, Recognition, Transcriptions. This paper tells about various methods that have been used to analyze alphabets from the previous decade.

III. METHODOLOGY

The online handwritten character recognition system is designed using a Convolutional Neural Network(CNN). Convolutional neural network use filters which are simply like a matrix of weights in each of its layers. The CNN has same weight over its filters and this is the advantage of CNN over other artificial neural networks. This eventually reduces the computational complexity of the model.

In the proposed model, the sequential CNN model is used. The Sequential model allows us to add layers sequentially from input to output. There are 8 layers which include 2 convolutional layers, 2maxpooling layers, 2 dense layers, 1 dropout layer. The final layer is the dense layer. The input layer is the convolutional layer with output shape (28, 28,128) and 'relu' activation function. The dense layer is taken as the output layer with softmax activation.

a) ReLU(Rectified Linear Unit)

It gives 0 if the input is lower than the output. Otherwise, the output is equal to the input. Figure (2) shows the relu activation function.

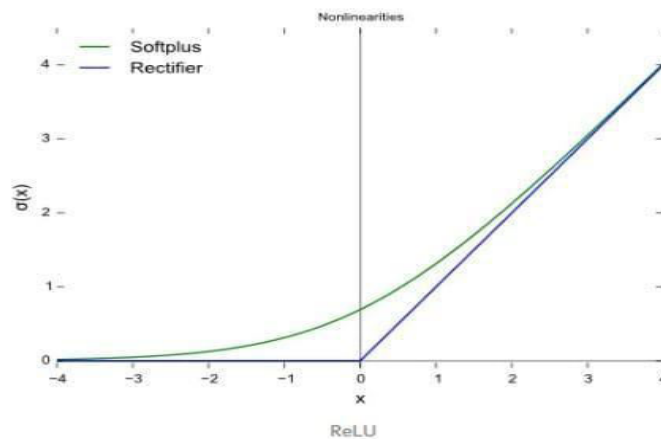


Figure (2): relu activation function



b) Softmax activation

Softmax maps each output to range between [0, 1] and map each output so that the sum is equal to 1. Figure (3) shows the softmax activation function.

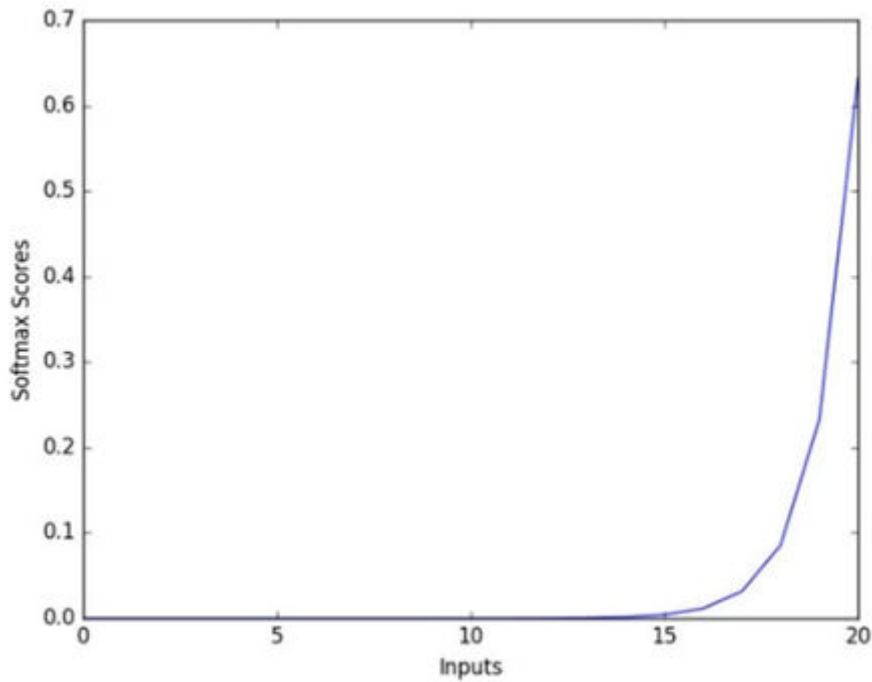


Figure (3): Softmax activation function

A total of 10 epochs is used with validation accuracy of 90 percent. Training is carried out on 628137 samples and validation on 69794 samples. The training and testing samples have the size (28, 28). Figure (4) shows the model layers and its number of parameters.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 128)	3328
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 128)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	73792
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 64)	0
flatten_1 (Flatten)	(None, 3136)	0
dense_1 (Dense)	(None, 128)	401536
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 47)	6063
Total params: 484,719		
Trainable params: 484,719		
Non-trainable params: 0		

Figure (4): CNN model



Tensorflow is used at the backend. Open-source neural network library 'Keras' is used on top of TensorFlow. Keras provides a high-level API's while TensorFlow provides both low-level and high-level libraries. Python which is compatible with both TensorFlow and Keras is used as the programming language. A simple GUI is designed to collect user input online.

IV. EXPERIMENTAL RESULTS

The training of 628137 samples and testing of 69794 samples of EMNIST ByMerge dataset on 10 epochs gave **90.35% validation accuracy**. The validation loss is only about 26.64%. Figure (5) shows the training of the model.

```
Epoch 8/10
628137/628137 [=====] - 1526s 2ms/step - loss: 0.3124 - accuracy: 0.8899 - val_loss:
0.2681 - val_accuracy: 0.9010
Epoch 9/10
628137/628137 [=====] - 1544s 2ms/step - loss: 0.3054 - accuracy: 0.8918 - val_loss:
0.2700 - val_accuracy: 0.9012
Epoch 10/10
628137/628137 [=====] - 1621s 3ms/step - loss: 0.2998 - accuracy: 0.8934 - val_loss:
0.2664 - val_accuracy: 0.9035
```

Figure (5): Training of the CNN model

By plotting a graph showing the **model accuracy** with the number of epochs on x-axis and accuracy on the y-axis we will get a plot as in figure (6). Another graph showing the testing and training loss is given in figure (7).

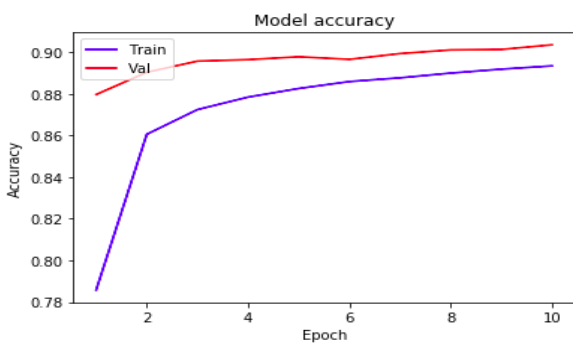


Figure (6): Model accuracy

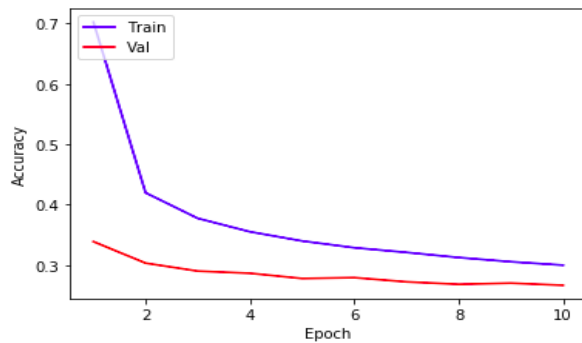


Figure (7): testing and training accuracy

The **Handwritten character recognition app** that we created to help in finding the prediction of any character or digit the user wants to give. We can draw the alphanumeric character in the **Canvas area** on the left side of the window either using a mouse or touch. The application window is provided with a **'Clear'** and a **'Run'** button on the bottom. By clicking the Run button after giving the input, we can see the character **Predictions** on the right side of the window. The predictions of some of the characters are given in figures (8) to (11). We can clear the Canvas area and the responses of user input by just clicking on the Clear button.

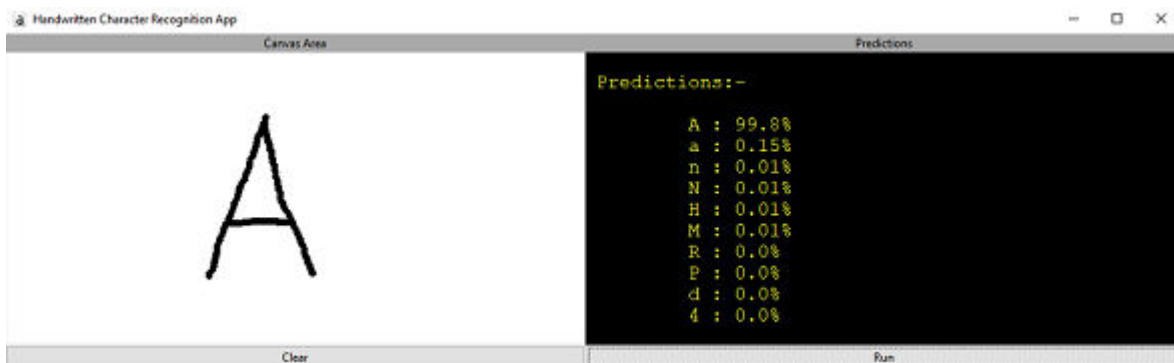


Figure (8): Prediction for capital letter A

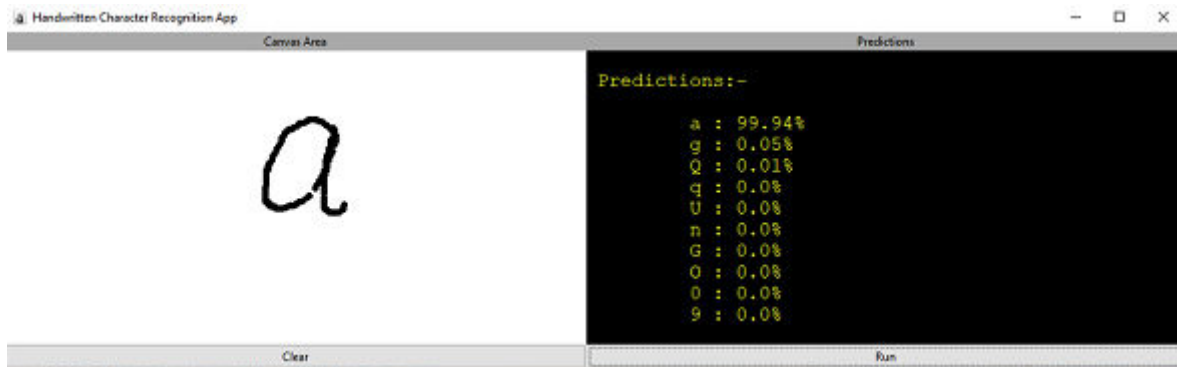


Figure (9): Prediction for small letter a

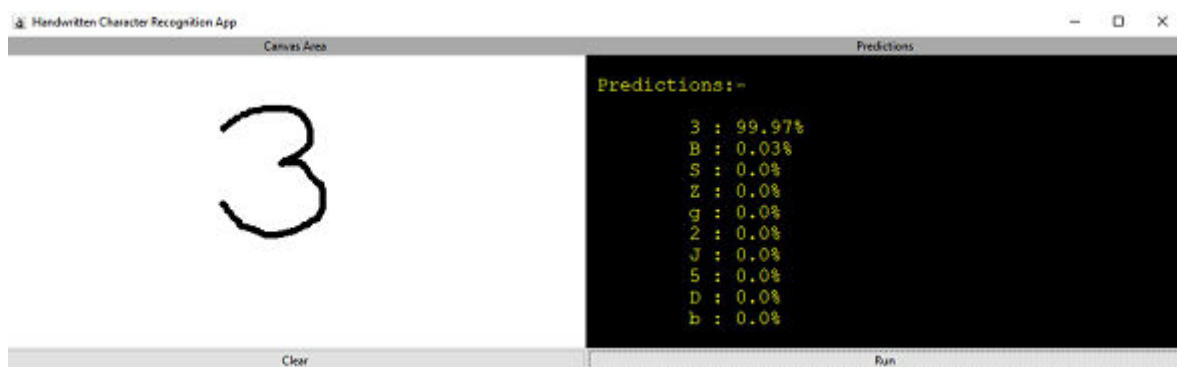


Figure (10): Prediction for number 3

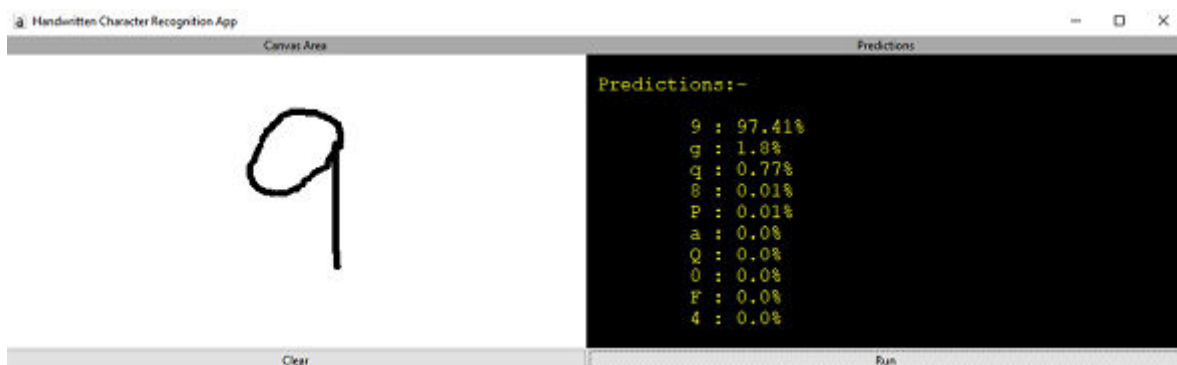


FIGURE (11): PREDICTION FOR NUMBER 9

V. CONCLUSION

The automatic recognition of handwritten characters is a difficult process as the writing styles of different people are different. Handwritten Character Recognition has a wide variety of applications like reading handwritten text, names or documents, reading bank check amounts, forms, and postal addresses. We have done the character recognition of English characters (uppercase, lowercase, and digits from 0-9) in this project. The testing accuracy for the EMNIST dataset which we used to train our Keras deep learning model is found to be 90.35%. We also demonstrate the output prediction of each user given character in a Graphical User Interface (GUI).



REFERENCES

- [1] J.Pradeep, E.Srinivasan and S.Himavathi, "DIAGONAL BASED FEATURE EXTRACTION FOR HANDWRITTEN ALPHABETS RECOGNITION SYSTEM USING NEURAL NETWORK, International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011"
- [2] Anitha Mary, M.O. Chacko and Dhanya P.M., "HANDWRITTEN CHARACTER RECOGNITION IN MALAYALAM SCRIPTS – A REVIEW, International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 5, No. 1, January 2014"
- [3] Deepika Wadhwa, Karun Verma, "ONLINE HANDWRITING RECOGNITION OF HINDI NUMERALS USING SVM, INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, (0975 – 888) Volume 48– No.11, June 2012"
- [4] P.V. Ramana Murthy, CH. G.V.N. Prasa, "RECOGNITION OF ONLINE HANDWRITTEN TELUGU LETTERS FOR DIFFERENT DOMAINS AND ORGANIZATIONS, Journal of Critical Reviews, Vol 6, Issue 6, 2019"
- [5] Manoj Sonkusare and Narendra Sahu, "A SURVEY ON HANDWRITTEN CHARACTER RECOGNITION (HCR) TECHNIQUES FOR ENGLISH ALPHABETS, An International Journal (AVC) Vol.3, No.1, March 2016"
- [6] Ankita Wankhade, Kalpana Malpe, "DEEP NEURAL NETWORK FOR HANDWRITTEN CHARACTER RECOGNITION WITH FOCUS ON USING DOMAIN-SPECIFIC KNOWLEDGE– A SURVEY Vol. 7, Issue 5, May 2018"
- [7] Meghna Agarwal, shalika, Vinam Tomar, Priyanka Gupta, "Handwritten Character Recognition using Neural Network And Tensorflow, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-6S4, April 2019"
- [8] Cohen, G.Afshar, S.Tapson, J. & VAN Schaik, A.(2017) "EMNIST : AN EXTENSION OF EMNIST TO HANDWRITTEN LETTERS, <http://arxiv.org/abs/1702.05373>"