INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 4)
Available online at: https://www.ijariit.com

# Handwritten character recognition

Addala Tejaswini
tejaswini.ira@gmail.com
BVRIT Hyderabad College of Engineering for Women

Y. Lakshmi Pratyusha
ylpratyusha5@gmail.com
BVRIT Hyderabad College of Engineering for Women

A. Rajashekar Reddy
raja.akuthota@gmail.com
BVRIT Hyderabad College of Engineering for Women

## ABSTRACT

***Character recognition is one of the most important research fields of image processing and pattern recognition. Character recognition is generally known as Handwritten Character Recognition (HCR) or Optical Character Recognition (OCR). HCR is the process of electronic translation of handwritten images or typewritten text into machine editable text. It becomes very difficult if there are lots of paper-based information on companies and offices. Because they want to manage huge volume of documents and records. Computers can work much faster and more efficiently than human. It is used to perform many of the tasks required for efficient document and Content management. But computer knows only alphanumeric characters as ASCII code. So computer cannot distinguish character or word from a scanned image. In order to use the computer for document management, it is required to retrieve alphanumeric information from a scanned image. There are so many methods, which are currently used for OCR and are based on different languages. The existing method like Artificial Neural Network (ANN based on English Handwritten character recognition needs the features to be extracted and also the performance level is low. So Convolutional Neural Network (CNN) based English handwritten character recognition method is used as a deep machine learning method for which it doesn't want to extract the features and also a fast method for characterrecognition.***

***Keywords:*** *Convolutional Neural Network, Optical Character Recognition, EMNIST*

## 1. INTRODUCTION

This project, 'Handwritten Character Recognition' is a software algorithm project to recognize any handwritten character efficiently on computer with input is either an optical image or currently provided through touch input, mouse or pen. Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editabletext.

Optical Character Recognition uses the image processing technique to identify any character computer/typewriter printed or hand written. A lot of work has been done in this field. The time used in entering the data and also the storage space required by the documents can be highly reduced by the use of OCR or in other words it can be retrieved fast. OCR in advance can be inferred in two ways based on type of the text and document acquisition.

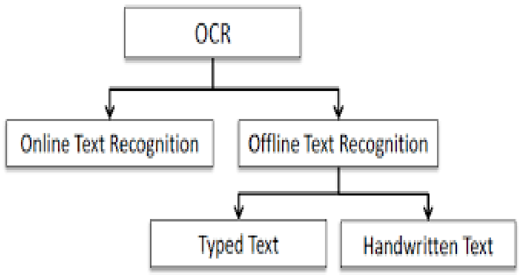


**Fig. 1: OCR Types**

Many a times the writing style of same individual is different at times. Further OCR is characterized into two forms as Offline and Online recognition systems based on acquiring of the documents. Offline System deals with recognizing the pre written document acquired through various input methods. But in Online recognizing system, the writing is recognized the moment it is written. The device used for the online system is Electric pen where it isused for writing the letters or words on the device called as digitizer and on the basis of the pen movement the input isrecorded.

## 2. DATASET AND METHODOLOGY

The dataset is taken from the EMNIST (Extended Modified National Institute of Standards and Technology) is a large database of handwritten digits and characters that is commonly used for training various image processing systems. "re-mixing" the samples from NIST's original datasets created it.

### 2.1 DatasetSummary

There are six different splits provided in this dataset. A short summary of the dataset is provided below:

- EMNIST ByClass: 814,255 characters. 62 unbalancedclasses.
- EMNIST ByMerge: 814,255 characters. 47 unbalancedclasses.
- EMNIST Balanced: 131,600 characters.

 

47 balancedclasses.

- EMNIST Letters: 145,600 characters.
  26 balancedclasses.
- EMNIST Digits: 280,000 characters.
  10 balancedclasses.
- EMNIST MNIST: 70,000 characters.
  10 balancedclasses.

The full complement of the NIST Special Database 19 is available in the ByClass and ByMerge splits. The EMNIST Balanced dataset contains a set of characters with an equal number of samples per class. The EMNIST Letters dataset merges a balanced set of the uppercase and lowercase letters into a single 26-class task. The EMNIST Digits and EMNIST MNIST dataset provide balanced handwritten digit datasets directly compatible with the original MNIST dataset. We used ByClass split of EMNIST dataset, which contains 697, 932 training images and 116, 323 testing images. See figure: 2.

**Fig. 2: EMNIST Dataset**

### 2.2 Methodology

Our goal is to detect the character, which is given as input, in whatever style the input text might be. In this project, we develop a model for handwritten character recognition. We also present the algorithms for recognizing the characters, which is given as the input, and give the correct output for the user. In this recognition process there will be very less wrong recognition of characters. Instead it recognizes the input and gives the correct output for the user, with accuracy of 86.31% and in a less time while compared to existing once.

### 2.3 Convolutional Neural Network

Using a fully connected neural network to make an image classification requires a large number of layers and neurons in the network, which increases the number of parameters leading the network to over-fitting (memorizing the training data only). The input image may also lose its pixels correlation properties since all neurons (carrying pixels values) are connected to each other. Convolutional neural networks have emerged to solve these problems through their kernel filters to extract main features of the input image and then inject them into a fully connected network to define theclass. The chosen architecture in our application is convolutional neural network. It contains 6 layers of convolution and simplification functions made by 5x5 kernel filters, Batch Normalization and a max-pooling filter of 5x5 to reduce at last the input image of 28x28. The feature images carry most important features to define a specified age and gender classes by processing them into fully connectednetwork.

In deep learning, a CNN is a class of deep neural network most commonly applied to analyzing visual imagery. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product.

The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. A CNN typically has three layers:

a) ConvolutionalLayer,
b) PoolingLayer
c) Fully ConnectedLayer.

## 3. CONVOLUTIONALLAYER

The convolution layer is the core building block of CNN. It carries the main portion of the network’s computational load. The main objective of convolution is to extract features such as edges, colors, and corners from the input. As we go deeper inside the network, thenetworkstarts identifying more complex features such as shapes, digits, as well. This layer performs a dot product between two matrices, where one matrix (known as filter/kernel) is the set of learnable parameters, and the other matrix is the restricted portion of the image. If the image is RGB then the filter will have smaller height and width compared to the image but it will have the same depth (height x width x 3) as of the image. At the end of the convolution process, we have a featured matrix, which has lesser parameters (dimensions) than the actual image as well as more clear features than the actual one.

## 4. POOLINGLAYER

This layer is solely to decrease the computational power required to process the data. Decreasing the dimensions of the featured matrix even more does it. In this layer, we try to extract the dominant features from a restricted amount of neighborhood. The pooling technique is MAX-pooling. In MAX-pooling, we just take the maximum amongst all thevalues lying inside the pooling region. So, after pooling layer, we have a matrix containing main features of the image and this matrix has even lesser dimensions.

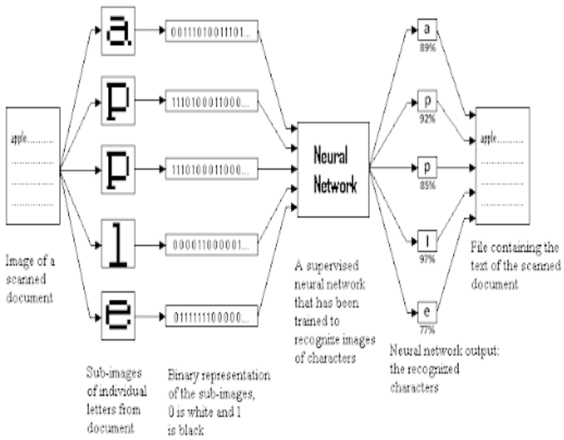


**Fig. 3: CNN Architecture**

## 5. FULLY CONNECTEDLAYER

Now that we have converted our input image into a suitable form for our Multi-Level fully connected architecture, we shall flatten the image into one column vector. The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them.

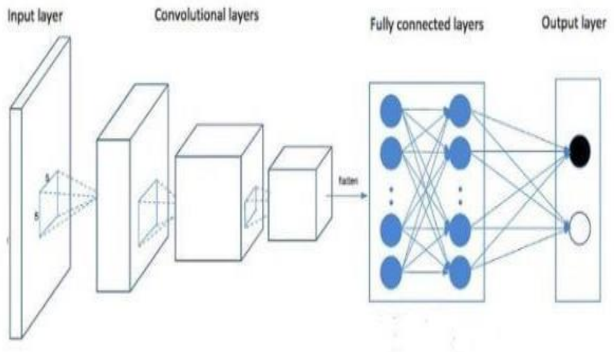


**Fig. 4: Fully connected layer inside CNN**

## 6. IMPLEMENTATON

### 6.1 Preprocessing the data

The raw data is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Pre-processing aims to produce data that are easy for the handwritten character recognition systems to operate accurately. Here we applied normalization technique, which is shown in the figure: 5.0. In imageprocessing, normalization is a process that changes the rangeof pixel intensity values. Applications include photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion. We need to reshape our dataset inputs (X_train and X_test) to the shape that our modelexpects when we train the model. The first number is the number of images (60,000for X_trainand 10,000 for X_test). Then comes the shape of each image (28x28).The last number is 1, which signifies that the images are greyscale.

## 7. HANDWRITTEN CHARACTER RECOGNITION USING CNN

In this section, we discuss the proposed method to recognize handwritten character. The framework involves several stages to accurately identify character. Figure 7 describes the proposed schematic overview of our framework. It has severalphases:

**Step i**: Collected English Handwritten characters from EMNICTDataset.

**Step ii**: Collected samples are pre-processed to scale the images by 28X28 in gray scale format and normalize them between 0 and 1.

**Step iii**: The dataset is divided into two groups as train and test phase. In training phase, we have considered 814, 255 individual samples i.e 697, 932 characters for training and for testing phase 116, 323 samples.

**Step iv:** We implemented a Convolutional Neural Network (CNN) model in Python using Colab. The training samples are input to CNN and that adjusted the weights to accurately recognize characters.

The proposed CNN method is evaluated on test dataset to measure the overall accuracy and individual digit and character recognition rate. We ran different iterations by varying the neural network parameters that can observe in lot of variation in accuracy.

The overall accuracy is captured of 15 epochs is shown in Fig. 6. At the 15th epoch, we measured heights accuracy of 86.96% is captured. In our experiments, we also captured the each digit and character recognition rate that gives the insight of how each digit can be accurately identified. Most of the digits are recognized more the 84% except letter e, digit 2, digit 5 and letter r are very low in accuracy due to ambiguity in patterns. These digits and letter are closely resembling each other that can successfully recognize individual patterns.

```
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.01568627 0.01568627 0.01568627 0.
 0.         0.         0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.00392157 0.07843137 0.1254902
 0.14509804 0.2        0.44705882 0.49019608 0.44705882 0.17647059
 0.01960784 0.         0.         0.        ]
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.00784314 0.03529412 0.18431373 0.62352941 0.79215686
 0.85098039 0.87058824 0.96078431 0.97647059 0.96078431 0.76078431
```

**Fig. 6: Normalized Data**

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
reshape_3 (Reshape)          (None, 28, 28, 1)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 28, 32)        832
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 28, 28, 64)        51264
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
flatten_3 (Flatten)          (None, 3136)              0
_________________________________________________________________
dense_5 (Dense)              (None, 512)               1606144
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_6 (Dense)              (None, 62)                31806
=================================================================
Total params: 1,690,046
Trainable params: 1,690,046
Non-trainable params: 0
_________________________________________________________________
None
Train on 697932 samples, validate on 116323 samples
```

**Fig. 7: Model Summary of CNN**

Next, a diagnostic plot is shown, giving insight into the learning behavior of the model across each fold. In this case, we can see that the model generally achieves a good fit, with train and test learning curves converging. There is no obvious sign of over- or under fitting. The accuracy and loss graphs for the CNN model are shown in figure: 8 and figure 9 respectively.

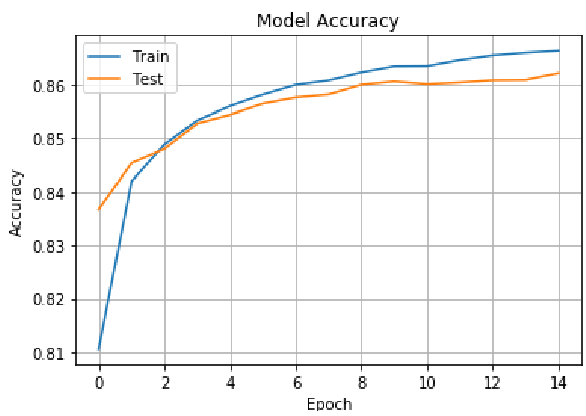


**Fig. 8: Model Accuracy**

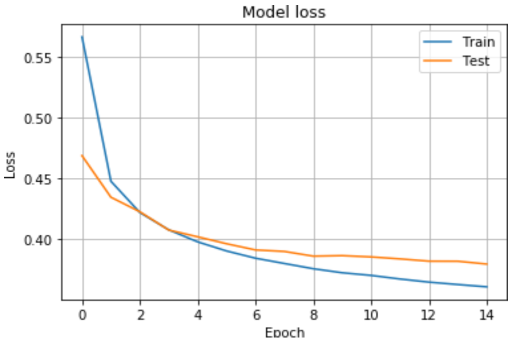


**Fig. 9: Model Loss**

After we found out accuracy and loss for our model, we created a graph to see which optimizer works well for model and it is shown in figure: 10.

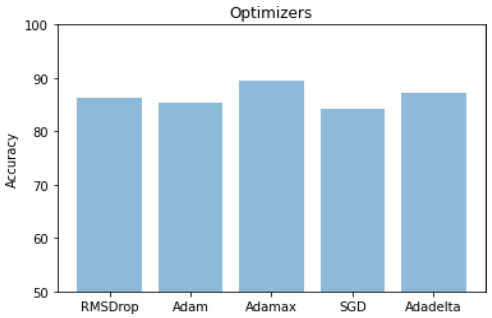


**Fig. 10: Optimizers**

After the completion of the shift row operation the next stage is mix column operation.

## 8. RESULT

The input is given as scanned imaged as shown in figure 11.

**Fig. 11: Imput**

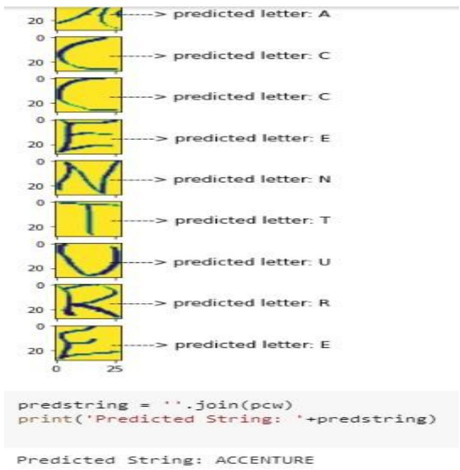


**Fig. 12: Output**

The final output of the CNN is shown in the following fig 12.

## 9. CONCLUSIONS AND FUTURE SCOPE

HCR plays an important role in optical character recognition and pattern recognition. It greatly contributes to automation of many things like medical prescriptions, tax returns etc. Earlier handcrafted feature methods were used for character recognition, which are inefficient and requires much effort and time. One of the best alternatives for conventional handcrafted features was to use deep learning techniques for character recognition. All the features used here are machine-generated features and produces much efficiency and accuracy to the result. The various deep learning techniques used are convolutional neural network, etc. As a part of future scope, we are trying to increase the complexity from character and word recognition level to sentence and paragraph recognition level. Apart from this we will implement front-end for this project-using HTML, Bootstrap so that our project can be user friendly.

## 10. REFERENCES

[1] MeghaAgarwal,Shalika, “HandWritten Character Recognition using neural network,”in International Journal of Innovative Technology and Exploring Engineering (IJITEE),Vol - 8, Issue- 6S4, April 2019,ISSN: 2278-3075.

[2] Joseph JamesS , C.Lakshmi, An Efficient Offline Hand Written Character Recognition using CNN, in International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol – 8, Issue-6,April 2019, ISSN: 2278-3075.

[3] BhagyasreeP V, Ajay James A Proposed Framework for Recognition of Handwritten Cursive English Characters usingDAG-CNN.

[4] Jagan Mohan Reddy D, A Vishnuvardhan Reddy, "Recognition of Handwritten Characters using Deep Convolutional Neural Network”International Journalof Innovative Technology and Exploring Engineering (IJITEE), ISSN:2278-3075, Vol - 8, Issue-6S4, April 2019, ISBN 3-540-42580-2.

## BIBLIOGRAPHY

**Rajashekar Reddy**
BVRIT Hyderabad College of Engineering for Women, India

**Addala Tejaswini**
BVRIT Hyderabad College of Engineering for Women, India

**Y. Lakshmi Pratyusha**
BVRIT Hyderabad College of Engineering for Women, India

 