IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 5.731

Volume 2, Issue 1, January 2022

OCR Using Convolution Neural Network in Python with Keras and TensorFlow

Sandipta Bhadra[1], Kritika Aneja[2], Satyaki Mandal[3]
Department of Computer Science and Engineering[1,2,3]
Vellore Institute of Technology, Chennai, Tamil Nadu, India
sandipta.bhadra2019@vitstudent.ac.in[1], kritika.aneja2019@vitstudent.ac.in[2], satyaki.mandal2019@vitstudent.ac.in[3]

Abstract: *We aim to design an expert system for," OCR using Neural Network" that can effectively recognize specific character of type style using the Artificial Neural Network Approach. We are pre-processing the input image, extracting the features, and then using the classification schema along with training of system to acknowledge the text. During this approach, we have trained the system to seek out the similarities, and also the differences among various handwritten samples. It takes the image of a hand transcription and converts it into a digital text. The extension of MNIST digits dataset has been used and A-Z characters in both uppercase and lowercase to detect handwritten text and convert it into digital form using Convolutional Neural Networks model, abbreviated as CNN, for text classification and detection also we are using keras graph to predict alphanumeric characters drawn using a finger and linked our handwriting text recognition program using keras and TensorFlow librar.*

Keywords: Handwritten Digit Recognition, Epochs, Convolutional Neural Network, MNIST dataset, Hidden layers

I. Introduction

Handwritten Text Recognition is a technology that is much required in this world as of today. Before proper implementation of this technology, we've relied on writing texts with our own hands which can end in errors.It's hard to store and access physical data efficiently. Manual labour is required in order to maintain proper organization of the data. Throughout history, there has been severe waste of knowledge due to the normalmethod of storing data. Modern day technology is letting people store the info over machines, where the storage, organization and accessing of knowledge iscomparatively easier. Adopting the utilization ofHandwritten Text Recognition software, it's easier to store and access data that was traditionally stored.

Handwritten Text Recognition is a technology that is much required in this world as of today. Before proper implementation of this technology, we've relied on writing texts with our own hands which can end in errors.It's hard to store and access physical data efficiently. Manual labour is required in order to maintain proper organization of the data. Throughout history, there has been severe waste of knowledge due to the normalmethod of storing data. Modern day technology is letting people store the info over machines, where the storage, organization and accessing of knowledge iscomparatively easier. Adopting the utilization ofHandwritten Text Recognition software, it's easier to store and access data that was traditionally stored.

II. Literature Survey

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

2.1 Handwritten Text Recognition using Deep Learning

Batuhan Balci, Dan Saadati, Dan Shiferaw

This project seeks to classify an individual handwritten word so that handwritten text can be translated to a digital form. The researchers have used two main approaches to accomplish this task: classifying words directly and character segmentation.

For the former, they have used *Convolutional Neural Network (CNN)* with various architectures to train a modelthat can accurately classify words. For example they have found a previous CS 231N project to be helpful in guidingthem with their task. A Faster R-CNN model to identify individual characters within a word and for classification.This uses a sliding window across the image to first determine whether an object exists within the boundaries.That bounded image is then classified toits corresponding character. They have also implemented edit distance which allows for making modifications to the classified word to determine if another classified word is more likely to be correct (for instance xoo vs zoo).

For the latter, we use *Long Short Term Memory networks (LSTM)* with convolution to construct bounding boxes for each character and various other data augmentation and preprocessing techniques to the dataset (IAM HANDWRITTEN DATASET), to make the data more compatible withthe models and to make the dataset more robust to real life situations.

- PADDING: Words segmented from different sentences are of different lengths sopadding must be provides for easier inputinto the CNN.
- Zero-centering of Data: to avoidunnecessary amplifivation and de- amplification ofcertain data points

They have then passed the segmented characters to a CNN for classification, and then reconstructeach wordaccording to the results of classification and segmentation.

2.2 Handwritten Text Recognition: with Deep Learning and Android

Shubham Sanjay Mor, Shivam Solanki, Saransh Gupta, SayamDhingra, Monika Jain, Rahul Saxen

This research paper offers a new solution to traditional handwriting recognition techniques using concepts of Deep learning and computer vision. An extension of MNIST digits dataset called the Emnist dataset has been used. It contains 62 classes with 0-9 digits and A-Z characters in both uppercase and lowercase. An application for Android, to detect handwritten text and convert it into digital form using Convolutional Neural Networks, abbreviated as CNN,for text classification and detection, has been created. They have approached the problem using CNN as it provides better accuracy over such tasks.

1. **Deep Learning:** finds out complex structure in massive data sets by using the back propagation algorithm to indicate how a machine should changeits internal parameters used in each new layer from the representation in the previous layer. CNN image classifications take an input image, process it and classify it under certain categories (E.g., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension).
2. **Data Pre-Processing:** Prior to training of sequential model with the train images, it was required to apply some pre-processing techniques to the data in order to make it more fitting for the model and for app (The android app provides a 1-D array as an input).

- **Normalization** - In image processing, changing the range of pixel intensity values is known as normalization. Some of its applications include correcting photographs with poor contrast dueto glare, for example. Here they have used the range of float values between 0 and 1.
- **Rotating and Reversing the data** –The Emnist dataset contains 697932 train images and 116323test images, all rotated and reversed. In this section, they reverse the images and later rotate them 90 anticlockwise. They achieve both these operations using an inbuilt NumPy transpose function.
- **Input image filtering** –Firstly, they have resized and image in the ratio 4:5(using inbuilt python libraries) and convert it into grayscale. Next, they performed thresholding on the image. After this the image is converted to binary format because contours can be found easily in binary images.
- **Model:** The CNN model for recognizing handwritten characters is constructed using the KERAS library of python and tensorflow [9] backend. Keras is a high-level neural networks API, written in Python and capable of running ontop of TensorFlow [9], CNTK, or Theano as backend. The equation for the SoftMax function is given below.

$$\left\{ \sigma(z)_{\{j\}} = \frac{\{e^{\{z_{\{j\}}\}}\}}{\{\sum_{\{k=1\}}^{\{k\}} e^{\{z_{\{k\}}\}}\}} \quad \text{for } j=1,\ldots,K \right\}$$

SOFTMAX will calculate the probabilities of each target class over all possible target classes. The target class can be later easily identified using the calculated probabilities for the inputs. SOFTMAX provides with arange of 0 to 1 with the sum of all probabilities equal to1.

3. **Result:** The highest accuracy obtained while training the models was when Adamax was used for experiment. So, they decided to use it for the purpose of research on the Emnist dataset.
4. **Android Application:** The android application developed for the purpose of easy hands on usage of this project was built with SDK tools 27.0.1 and gradle version 4.4. The android application has a simple layout with a draw view, two buttons and a text view and is inspired by the project uploaded by Sourcell: Link - https://github.com/llSourcell/A_Guide_to_Running _Ten sorf low_ Models_on_Android.

2.3 Handwritten Text Recognition using Deep Learning with TensorFlow

Sri. Yugandhar Manchala, Jayaram Kinthali, Kowshik Kotha

Character recognition is one in all the emerging fields within the computer vision. The most abilities of humans are they will recognize any object or thing. The hand transcription can easily identify by humans. Different languages have different patterns to spot. Humans can identify the text accurately. The hand transcription cannot be identified by the machine. It's difficult to spot the text by the system. During this approach, the system is trained to seek out the similarities, and also the differences among various handwritten samples. This application takes the image of a hand transcription and converts it into a digital text. Image processing could be a manipulation of images within the computer vision. The character recognition involves several steps like acquisition, feature extraction, classification, and recognition. Handwriting recognition isthe ability of a machine to receive and interpret the handwritten input from an external source like image. the most aim of this project is to style a system that may efficiently recognize the actual character of format employing a neural network. Neural computing could be a comparatively new field, and style components are therefore less well-specified than those of other architecture. Neural computers implement data parallelism.

NN for our task. It consists of a convolutional neural network (CNN) layers, recurrent neural network(RNN) layers, and a final Connectionist Temporal Classification (CTC) layer. taken 5 CNN (feature extraction) and a pair of RNN layers and a CTC layer (calculate the loss). first, we've to pre processthe pictures in order that we are able to reduce the noise. view the NN in an exceedingly more formal way as a which maps a picture (or matrix) M of size W×H to a personality sequence (c1, c2, …)with a length between 0 and L. As you'll see, the text is recognized on character-level, therefore words or texts not contained within the training data is recognized too.

III. Proposed Work

The recognition process of the handwritten digits includesthe following steps:

1. To collect the MNIST handwritten digit images.
2. To split the input images into training and test images.
3. To apply the pre-processing method to both the trainingdataset and the test dataset.
4. To normalize the data to make it ranges from 0 to 1.
5. To split up the training dataset into batches of a suitablesize.
6. To train the CNN model and its variants applying the labelled data.
7. To use a trained model for the classification.
8. To examine the recognition accuracy and processingtime for all the variants.

3.1 Dataset Used

To train our custom Keras and TensorFlow model, we first implemented two helper utilities that will permit us to load both the Kaggle A-Z datasets and the MNIST 0-9 digits from disk.
I/O helper functions are:

- load_az_dataset: for the Kaggle A-Z letters
- load_mnist_dataset: for the MNIST 0-9 digits

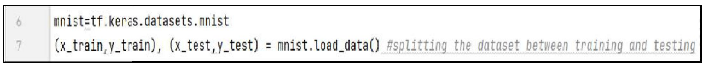
3.2 Model Used

1. **Sequential:** The sequential model contain linear stack of layers. The same has been used for developing the handwritten text recognition model. Defining a model was ourprimary task. Defining a model basically means adding different layers to the stack. Layer 1 is the first layer which we have added is the reshape layer that took an input of (784,1) and reshaped it into (28,28,1). Layer 2 and Layer 3 arethe Convolutional layer - The function of convolutional layer is to discover unique features from the given input matrix which is an image of handwritten text which has been already normalized. The convolutional layer uses a filter matrix whichis a combination of zeros and ones. The filter matrix run over the input matrix and keeps the count of elements which matched with the elements of the filter matrix. The matrix so created is called FEATURE MAP. This convolutional layer takes input matrix of shape of (28,28,1).
2. **CNN:** Convolutional neural networks are deep artificial neural networks. We used it to categorize images, cluster themby similarity (photo search) and perform object recognition within scenes. It is used to recognize faces, street signs, individuals, tumours and various other aspects of visual data. The layer's parameters consist of a set of learnable filters (or kernels) which have a small receptive field but reach out the full depth of the input volume. During the forward pass, each filter is convolved over the width and height of the input volume and calculates the dot product, producing a 2- dimensional activation map of that filter. As an outcome, the network learns and when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a down sampling layer, this method is applied one patch at a time, similar as convolutions.CNN has also fully connected layer that categorize outputwith one label per node Convolutional Neural Network (CNN) is at the centre of spectacular advances that blends Artificial Neural Network (ANN) and up to date deep learning strategies. We have used CNN to categorize handwritten digits using various numbers of hidden layers and epochs. CNN include three dimensions where all the neurons are not fully connected rather each neuron is associative to the local receptive field. In order to train the network, a cost function is generated which link the output of the network with the desired output.

3.3 Data Pre-Processing

A. Digit Recognition: version 1

We have loaded MNIST dataset using keras.



```
mnist=tf.keras.datasets.mnist
(x_train,y_train), (x_test,y_test) = mnist.load_data() #splitting the dataset between training and testing
```

The training dataset is structured into 3-dimensional arrayof instance, image width and image height. For a multi-layer perceptron model we reduced the images down into a vectorof pixels. In this case the 28×28 sized images will be given as input which makes up 784 pixels. Initially the images were normalized to 20*20 pixels and centred, also their aspect ratio was maintained

After this they were resized to 28*28 pixels in order to get better accuracy and distinction between black and white pixel of image. The input to the model is pixel data in the form of PNG filecreated using MS Paint

B. Optical Character Recognition: version 2.1

Depending on the space between groups of characters, a string of characters (a sentence) is broken down into individual groups of character(words) and then depending on the spaces between individual characters, a word is broken down inti individual letters. These individual blocks are then scanned and are identified by their specific structural features by mating them from the dataset of characters, to generate the most probable character matching the inputted image.

C. Optical Character Recognition: version 2.2

The basic preprocessing is the same for this version too, apart form that, the individual character images are also turned into greyscale for better memory management and image processing, also a Gaussian blur effect is applied on eachimage to reduce any sort of noise on the image.

3.4 Approach

A. Digit Recognition: version 1

We have used DNN(deep neural network). A perceptron is a node which takes the input processes it and gives out the single output. Single layer of perceptron is neural network and multiple layers of perceptron is called Deep Neural Network. This gives better accuracy compared to algorithms like linear regression, decision tree as they have limitation of processing high dimensional data DNN model used: Initially we used 3 hidden layereach containing 500 nodes and ran for 10 epochs Thenwe used same model and ran for 15 epochs. Then we used 4 hidden layers each containing 500,1500,1500,500 nodes respectively for 10 epochs Then we used the same model and ran for 20 epochs

B. Optical Character Recognition: version 2.1

We used Keras-OCR, which is a bit polished and packaged version of the Keras CRNN implementation and the published CRAFT text detection model. It provides a high-level API for training a text detection and OCR pipeline. Keras-OCR will automatically download pretrained weights for the detector and recognizer. Each list of prediction in prediction_group is a list of (word, box) tuples. Boxes will be represented as Nx4x2 array of box quadrangles, whereN represents the number of detected text boxes.

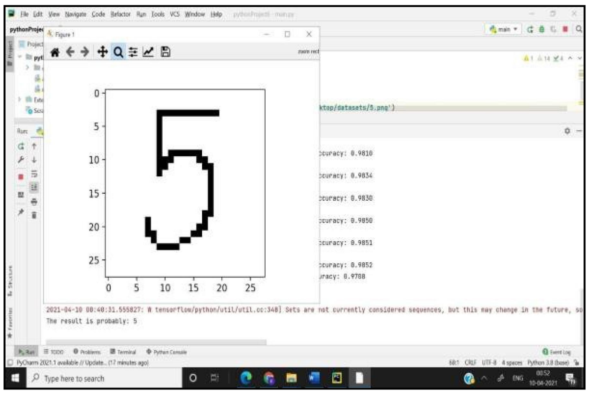
C. Optical Character Recognition: version 2.2

We have examined the handwriting dataset that we will be using to train our model. From there, we have executed couple of helper functions that will help us in loading our handwriting datasets from disk and then pre-processing them. Given these helper functions, have created our own custom OCR training script with Keras and TensorFlow. Next, we have trained the network, defined the label names, and evaluated the performance of the network. We have imported ResNet from our deeplearning model. model, which contains our own custom implementation of the popular ResNet deep learning architecture. then we have set up the training parameters for ResNet and load our digit and letter data using the helper functions and prepare our data and labels to be well suited with our ResNet deep learning model in Keras and TensorFlow. As we combine our letters and numbers into a single characterdata set, we desire to remove any ambiguity wherethere is overlap in the labels so that each label in the combined character set is unique. For this ,We added ten to all of our A-Z labels so they all have integer label values greater than our digit label values. Now, we have a unified labeling schema for digits 0-9 and letters A-Z without any intersection in the values of thelabels.

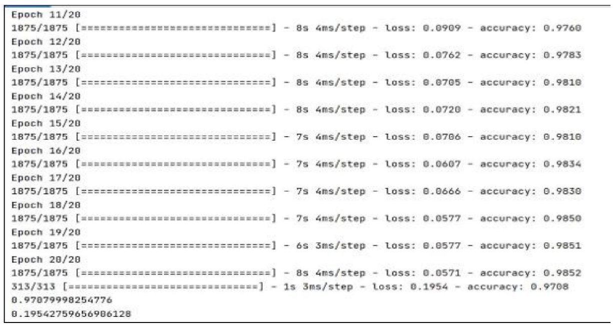
Our ResNet architecture input dimensions of 32 x 32was essential, but our input images currently have asize of 28 x 28. We resize each of the images using cv2.resize also we added an extra “channel” dimension to each and every image in the dataset to make it compatible with the ResNet model in Keras/TensorFlow. Finally, we scaled our pixel intensities from a range of [0, 255] to [0.0, 1.0]

3.5 Output and Accuracy

Version 1: Digit Recognition

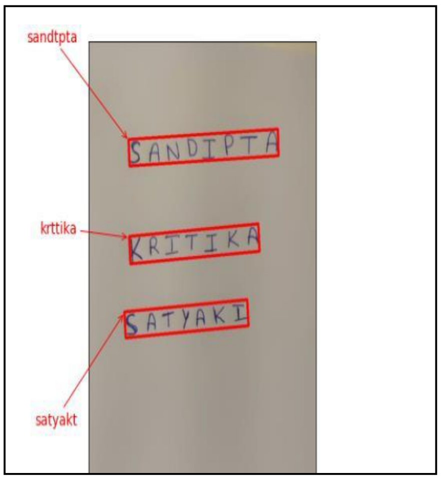


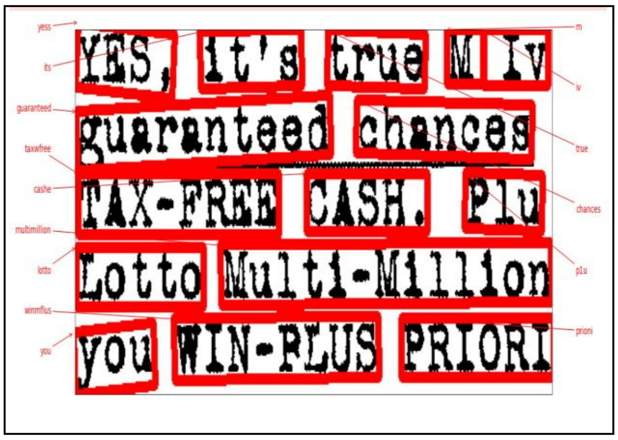
```
ktop/datasets/5.png')
ccuracy: 0.9810
ccuracy: 0.9834
ccuracy: 0.9830
ccuracy: 0.9850
ccuracy: 0.9851
ccuracy: 0.9852
uracy: 0.9708
2021-04-10 00:40:31.555827: W tensorflow/python/util/util.cc:348] Sets are not currently considered sequences, but this may change in the future, so
The result is probably: 5
```



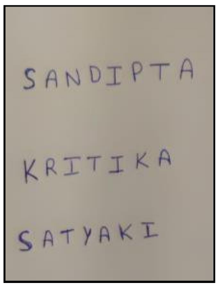
```
Epoch 11/20
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0909 - accuracy: 0.9760
Epoch 12/20
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0762 - accuracy: 0.9783
Epoch 13/20
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0705 - accuracy: 0.9810
Epoch 14/20
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0720 - accuracy: 0.9821
Epoch 15/20
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0706 - accuracy: 0.9810
Epoch 16/20
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0607 - accuracy: 0.9834
Epoch 17/20
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0666 - accuracy: 0.9830
Epoch 18/20
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0577 - accuracy: 0.9850
Epoch 19/20
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0577 - accuracy: 0.9851
Epoch 20/20
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0571 - accuracy: 0.9852
313/313 [==============================] - 1s 3ms/step - loss: 0.1954 - accuracy: 0.9708
0.97079998254776
0.19542759656986128
```

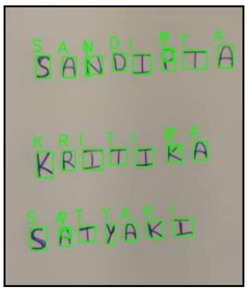
Version 2.1: Optical Character Recognition

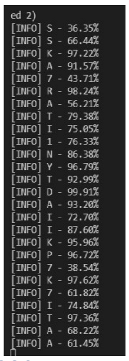




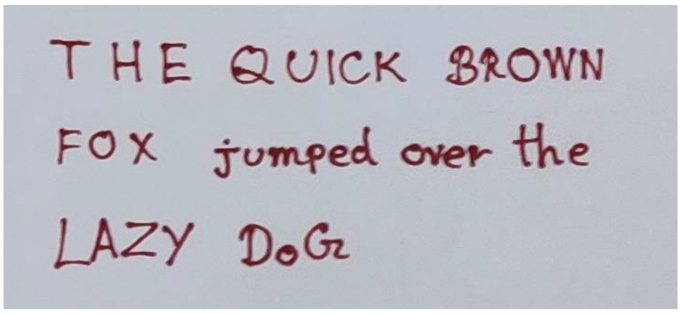
Version 2.2: Optical Character Recognition



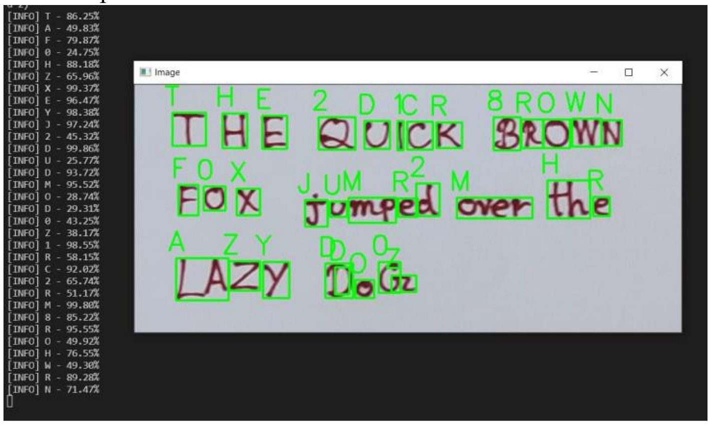




```
ed 2)
[INFO] S - 36.35%
[INFO] S - 66.44%
[INFO] K - 97.22%
[INFO] A - 91.57%
[INFO] 7 - 43.71%
[INFO] R - 98.24%
[INFO] A - 56.21%
[INFO] T - 79.38%
[INFO] I - 75.05%
[INFO] 1 - 76.33%
[INFO] N - 86.38%
[INFO] Y - 96.79%
[INFO] T - 92.99%
[INFO] D - 99.91%
[INFO] A - 93.20%
[INFO] I - 72.70%
[INFO] I - 87.60%
[INFO] K - 95.96%
[INFO] P - 96.72%
[INFO] 7 - 38.54%
[INFO] K - 97.62%
[INFO] 7 - 61.82%
[INFO] I - 74.84%
[INFO] T - 97.36%
[INFO] A - 68.22%
[INFO] A - 61.45%
```



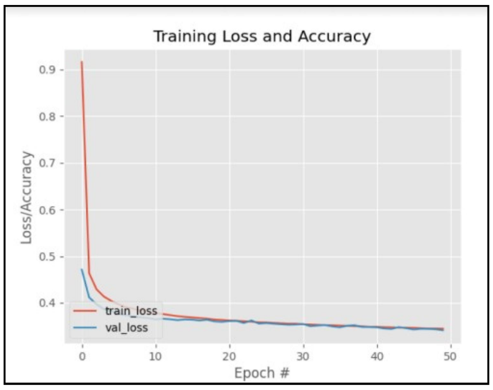
while training and testing the model, it felt like truelywe were teaching a child his/her first characters in language. Who would have thought, the end tail of capital "G" does look like a "z"



```
[INFO] T - 86.25%
[INFO] A - 49.83%
[INFO] F - 79.87%
[INFO] 0 - 24.75%
[INFO] H - 88.18%
[INFO] Z - 65.96%
[INFO] X - 99.37%
[INFO] E - 96.47%
[INFO] Y - 98.38%
[INFO] J - 97.24%
[INFO] 2 - 45.32%
[INFO] D - 99.86%
[INFO] U - 25.77%
[INFO] D - 93.72%
[INFO] M - 95.52%
[INFO] O - 28.74%
[INFO] D - 29.31%
[INFO] 0 - 43.25%
[INFO] Z - 38.17%
[INFO] 1 - 98.55%
[INFO] R - 58.15%
[INFO] C - 92.02%
[INFO] 2 - 65.74%
[INFO] R - 51.17%
[INFO] M - 99.80%
[INFO] 8 - 85.22%
[INFO] R - 95.55%
[INFO] O - 49.92%
[INFO] H - 76.55%
[INFO] W - 49.30%
[INFO] R - 89.28%
[INFO] N - 71.47%
```

IV. Conclusion

Among test cases, our model misclassifies some digits aftereight epochs which correspond to 96.15% recognition. The results are prettygood for such a simple model with CPU training and less training time (about 30 minutes).



Although there are some digits which are not a goodhandwriting, our model will be able to classify them correctly. Testing accuracy 96.15% implies that the model is trained well for prediction. Training set size affects the accuracy andaccuracy increases as the number of data increases. The more data in the training set, the smaller the impact of training error and test error, and ultimatelythe accuracy can be improved.

In our experiment we have maximum accuracy as 99.91% and maximum validation accuracy as 96.15%.the overall performance of network is found as 96.15%. Moreover, the overall lose range from 0.022036 to 0.023308

References

[1]. Gil Levi and Tal Hassner, "Offline Handwritten Digit Recognition Using Neural Network", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, no. 9, pp.

4373 -4377, 2013.

[2]. Nimisha Jain, Kumar Rahul, Ipshita Khamaru. AnishKumar Jha, Anupam Ghosh (2017). “Hand Written Digit Recognition using Convolutional Neural Network (CNN)”, International Journal of Innovations & Advancement in Computer Science, IJIACS,ISSN 2347– 8616,Volume 6, Issue 5

[3]. Dr. Kusumgupta2 ,"A Comprehensive Review On Handwritten Digit Recognition Using Various Neural Network Approaches", International Journal Of Enhanced Research In Management & Computer Applications, Vol. 5,No. 5, Pp. 22-25, 2016

[4]. Saeed AL-Mansoori, "Intelligent Handwritten Digit Recognition using Artificial Neural Network", Int. Journal of Engineering Research and Applications, vol. 5, no. 5, pp. 46-51, 2015.

[5]. Haider A. Alwzwazy1, Hayder M. Albehadili2, Younes S. Alwan3, Naz E. Islam4, "Handwritten Digit Recognition using Convolutional Neural Networks", International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 2, pp. 1101- 1106, 2016

[6]. Kussul, Ernst; Tatiana Baidyk (2004). "Improved method of handwritten digit recognition tested on MNIST database". Image and Vision Computing, 22(12): 971–981

[7]. "Handwritten Digit Recognition using various Neural Network Approaches", International Journal of Advanced Research in Computerand Communication Engineering, vol. 4, no. 2, pp. 78-80, 2015.

[8]. E. Tautu and F. Leon, “Optical Character Recognition System Using Support Vector Machines,” pp. 1-13, 2012.

[9]. R. Plamondon and S. N. Srihari, “On-line and off- line handwritten character recognition: A comprehensive survey,” IEEE. Transactions onPattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, 2000.

[10]. F. Bortolozzi, A. S. Brito, Luiz S. Oliveira and M. Morita, “Recent Advances in Handwritten Recognition”, Document Analysis, UmapadaPal, Swapan K. Parui, Bidyut B. Chaudhuri, pp 1-30.

[11]. U. Pal, T. Wakabayashi and F. Kimura, “Handwritten numeral recognition of six popular scripts,” Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007.

[12]. Rajavelu A, Musavi Mohamad T, Shirvaikar Mukul Vassant“ A neural network approach to character recognition Neural Network5, Elsevier (1989), p. 387393

[13]. Youssouf Chherawala, Partha Pratim Roy and Mohamed Cheriet, “Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network,” IEEE Transactions On Cybernetics, VOL. 46, NO. 12, December 2016

[14]. Faisal Tehseen Shah, Kamran Yousaf, "Handwritten Digit Recognition Using Image Processing and Neural Networks", Proceedings of the World Congress on Engineering, vol., 2007

[15]. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.

[16]. Wang, D.; Lihong, H.; Longkun, T. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. IEEE Trans. Neural Netw. Learn. Syst. 2017, 29, 3815– 3827.

[17]. Long, M.; Yan, Z. Detecting iris liveness with batch normalized convolutional neural network. Comput. Mater. Contin. 2019, 58, 493– 504

[18]. Ahlawat, S.; Rishi, R. A genetic algorithm based feature selection for handwritten digit recognition. Recent Pat. Comput. Sci. 2019, 12, 304– 316.

[19]. Pham, V.; Bluche, T.; Kermorvant, C.; Louradour, J. Dropout improves recurrent neural networks for handwriting recognition. In Proceedings of the 14th Int. Conf. on Frontiers in Handwriting Recognition, Heraklion, Greece, 1–4 September 2014.

[20]. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 2481–2495.