EasyChair Preprint

№ 3364

Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame

K. Senthil Kumar, Suman Kumar and Aabhash Tiwari

May 10, 2020

REALTIME HANDWRITTEN DIGIT RECOGNITION USING KERAS SEQUENTIAL MODEL AND PYGAME

[1]K. Senthil Kumar, [2]Suman Kumar, [3]Aabhash Tiwari

[1,2,3]Computer Science and Engineering,
[1,2,3]SRM Institute of Science and Technology, Chennai, India
[1]senthilkumar.k@ktr.srmuniv.ac.in , [2]sk2257682@hotmail.com,
[3]abhashtiwari.97@gmail.com

Abstract. Handwritten recognition has received the greater attention in deep learning research community due to its vast applications and ambiguity in learning methods. CNN in deep learning is now becoming one of the most appealing approaches and has been a crucial factor in the variety of recent success and challenging machine learning applications such as object detection. This research paper is about the extended application of handwritten digit recognition i.e. Realtime detection of the handwritten digits. CNN is used as the Model for the classification of the image and more specifically Keras Sequential Model is used as a classifier. The interface is created by pygame. The layout` of the interface is kept simple and is divided into two frames one for input and other for output. The image pre-processing is the most important step which has done with the help of OpenCV and Scipy. MNIST is the dataset used for training and testing. Handwritten Digit Recognition has various real-life time uses. It is used in the detection of vehicle number, banks for reading cheques, post offices for arranging letter, and many other tasks.

KEYWORDS*:* Sequential Model, Convolutional neural network (CNN) ,MNIST

1 Introduction

Handwritten digit recognition is a technology that is very much needed in this world as of today. It is used in the detection of vehicle number, banks for reading cheques and post office for arranging letters.In this paper, the extended application of handwritten recognition i.e. Realtime recognition of handwritten digits is implemented with the help of CNN and pygame. The classifier used in this project is a Sequential model with a 4-layer CNN. The dataset used in this paper is MNIST dataset. The MNIST dataset has 70000 handwritten digits. Among these 60000 are training example and 10000 are testing examples. Each image is this data is set to represented as an array of 28*28. The aim of this proposed endeavour was to make path toward digitization comprehensible by giving high accuracy and speedy computation for recognizing digits. That's why this problem was approached using CNN as they provide better accuracy over such tasks. To make Realtime recognition an interface is needed to draw, so to build that interface pygame was used as a tool. The input will be drawn on the left side of window and final output will be appeared on the right side.

2 Literature Survey

Numerous researchers blatant their contribution in the field of digit recognition. Based upon their outcome and features weights were assigned and it was executed on character recognition system by Hanmandlu and Murthy [1].

Graves and Schmidhuber[2] used a Hidden Markov Model that used a recurrent neural network. This helped in determining the sequence of characters in scripts that are handwritten. This was implemented for classifying the handwritten Arabic words. This gave an accuracy of about 91%. Multilayer perceptron was used by Pal and Singh[3] for recognizing English characters that are handwritten and an accuracy of 94% was successfully achieved. It was also able to improve the computation time for training the dataset. Neves [4] implemented the Support Vector Machine algorithm and compared this with the MLP algorithm. This recognized the offline handwritten characters with a much better accuracy. The tests and training were done on the standard dataset NIST SD19. Although MLP is a better classifier for all nonlinear classes segmentation, it unfortunately gets trapped in a local minima. This lets the support vector machine get a better accuracy. Younis and Alkhateeb used the MNIST Dataset and extracted features without any pre-processing. It addressed the handwritten OCR problem by

implementing a deep neural network. The accuracy achieved was 98.46%. [5] .Dutt[6] also used the MNIST Dataset. But they also used the Keras and Theano libraries and used multilayer CNN and got an accuracy of 98.7%. Ghosh and Maghari[7] received 98.08% and demonstrated that DNN is the best algorithm, after completing a comparative study of 3 neural networks. But there may be similarity in the shape of digits, and for this reason some amount of error rate will be present. CNN should be the best classifier in this case as compared to the other algorithms such as Support Vector Machine, KNN and the Random Forest Classifier for HDR. [8] implemented a DeepLearning4j framework which was incorporated with a rectified linear units activation which was never used before. The major aim of the paper was to recognize and determine digits that are handwritten with a higher accuracy and low computation time as well.
CNN is used for fault detection and also for Classification purposes. Recognition of handwritten digits is a big issue of interest in the community of researchers. Many research has already been done on this topic and the flow of papers is still on. This topic is a big interest topic in the field of Machine Learning. Highest accuracy has been achieved using deep learning algorithms and also be using different libraries such as theano and keras. Tensorflow has also been used. This enables the accuracy and performance to be better than other algorithms. Convolutional Neural Network is being used in various areas such as Natural Language Processing, Video analysis and many other research areas as well. Sentiment analysis has also gained traction among researchers and people are trying to do more and more research in this field and move deeper into the research area.
The research will go deeper in the coming days as well.

3 Dataset

A huge set of vision dataset which is used for training and testing is [Modified National Institute of Standards and Technology (MNIST)].

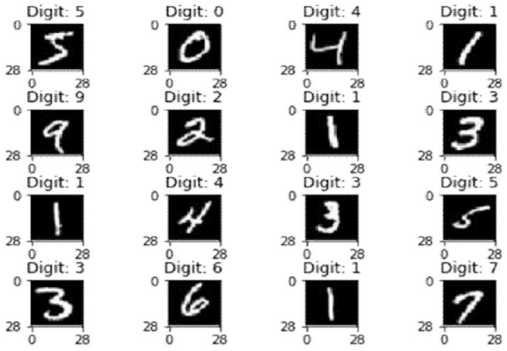

Fig. 1. Sample images of MNIST dataset

The dataset contains 60000 images for training and 10000 images used for testing. All images are grayscale and arranged in a explicit style of 28*28 pixels. Then it can be compressed into a 28*28=784-dimensional vector. Each element of the vector has a binary value which specifies pixel intensity.

4 Implementation

4.1 Convolutional Neural networks for Digit Recognition

In pattern recognition and image processing problems neural network are often used. The most promising tool to carry out this is CNN. It is a deep learning technique which is inspired by the neuron connectivity pattern in animal visual cortex. CNN is mainly used in image recognition, object recognition etc. The CNN model for recognizing the digits is constructed using the KERAS library of python and TensorFlow as a backend. Sequential Model is used as classifier which consists of linear stack of layers.
In CNN the neurons have learnable weights and biases. Compared to others CNN requires minimum pre-processing. The input is always a vector in neural network but in CNN the input will be a multi channelled image. The CNN comprises of an input layer, hidden layers and an output layer. The hidden layer constitutes the convolutional layer, Rectified layer unit (ReLU) i.e. acti-vation function, pooling layers, normalized layers and fully connected layers.

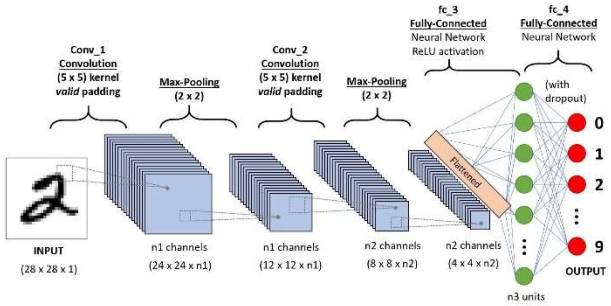

Fig. 2. A seven-layered CNN architecture for digit recognition

The input layer consists of 28 by 28-pixel images which means it contain the network of 784 neurons as input data. The layer next to input layer is convolutional layer which receives the output from the input layer. A convolution operation is performed on the input in convolutional layer. The operation performed in this layer helps in reducing the number of free parameters. And design each filter such that it slides above the input data to have pixels with the utmost intensity.
The Rectified Linear Unit (ReLU) is a activation function which is a linear function, easy to use and achieves better performance. Also, it does not saturate.
The concept of pooling layer is to merge the product of neuron at one layer into single neuron at the next level. The fully connected layer reckon the score of input digits. At the end a SoftMax classifier is applied at the end which returns the probabilities of all output classes. From all these values, the class with the largest value is selected as a final classification. To work on Keras API, it is converted into 4 dimen-sional NumPy arrays. The normalization is done by dividing RGB code by 255. Adam optimizers is used to update the neuron weights and it require little memory.

4.2 Image Pre-processing

After training the model it will be saved as a classifier. Now the image pre-processing is done on the input image obtained by the interface. The image processing is done with the help of Open Cv2 and Scipy.
We are reshaping the data to suit the neural network that is being built .After the reshaping process the data is now converted into categorical data.
The steps taken are: -
1.Read image: - In this, the path of our image is stored into a variable and then a function is created that converts images into array.
2.Resize image: -Some images captured by camera vary in size. The base size needs to be established for all images fed into our system.
3.Remove Noise: -By using Gaussian Blur () function image noise may be used.
4.Segmentation: - More techniques to smoothen and reduce further noise is done.
5.Scaling and Padding: - The image is then scaled and padded to make it suitable for prediction.

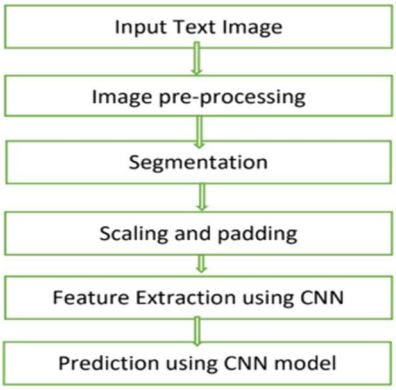

Fig. 3. Flow Diagram of proposed method

4.3 Application Interface

The application interface is created with the help of Pygame. Pygame is a group of modules in python which is highly portable and can be used for building video games.
For opening the pygame window, run the interface. On the left side the digits will be drawn by the user and the corresponding output will be displayed on the right side. To draw the digit left click your mouse and once finished release the left click. To reset the screen, click right button.
The output will be displayed using a bounding box and the recognized digit will be displayed in small letters at the top. Multiple numbers can be recognized as well in real time.

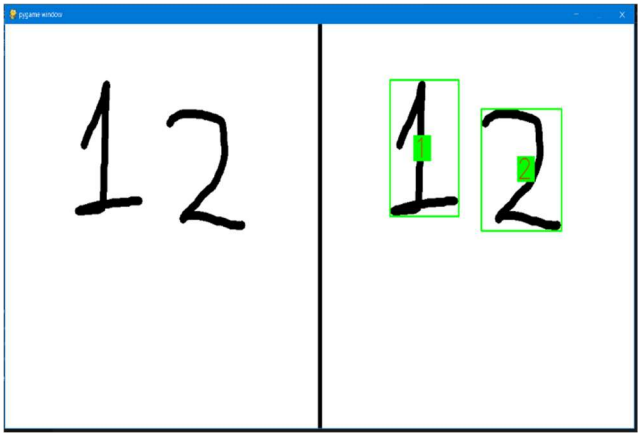

Fig. 4. pygame application interface

The image obtained after the drawing the input is now fed into the pre-processing stage and classifier predicts the digit and classifies it into one of the 10 classes.

5 Result

The experimental parameters for which study is performed are the 2 CNN layers vs 3 CNN layers vs 4 CNN layers. Also, result has been compared for various epochs. Performance has been measured through accuracy and loss for every CNN layers.

Table 1. Test Accuracy and Loss for 15 epochs

CNN Layers	Accuracy	Loss
2	98.68%	0.068
3	98.97%	0.032
4	99.25%	0.025

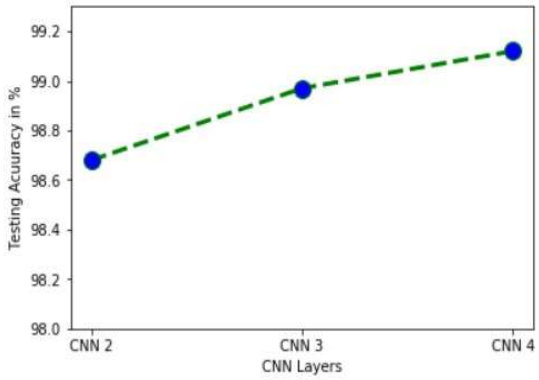

Fig. 5. Testing Accuracy for 15 epochs

The above graph shows the accuracy of the test data when trained with 15 epochs. The 4 layered CNN.

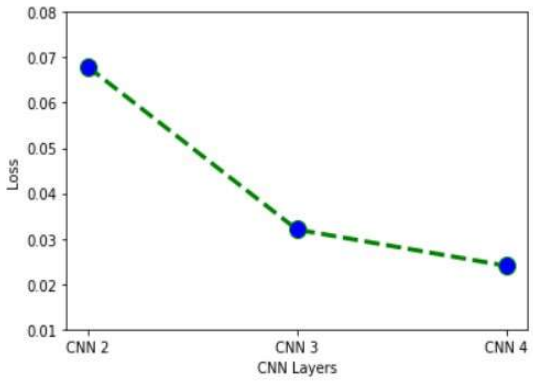

Fig. 6. Loss for 15 epochs

The above graph shows us the losses when the number of epochs is 15.

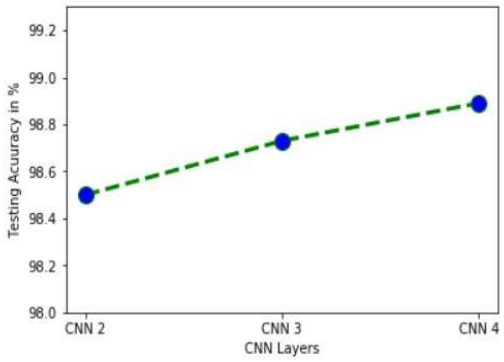

Fig. 7. Testing Accuracy for 5 epochs

The above figure shows us the accuracy of the test data when the number of epochs is 5.
In the below 3 figures the graph has been made for testing accuracy against the training accuracy for 15 epochs with different layered CNNs.

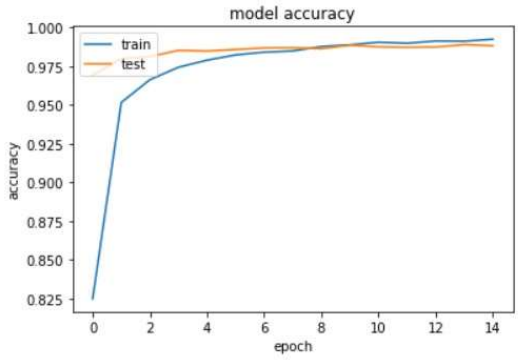

Fig. 8. Testing Accuracy vs Training Accuracy for 15 epochs in 2-layer CNN

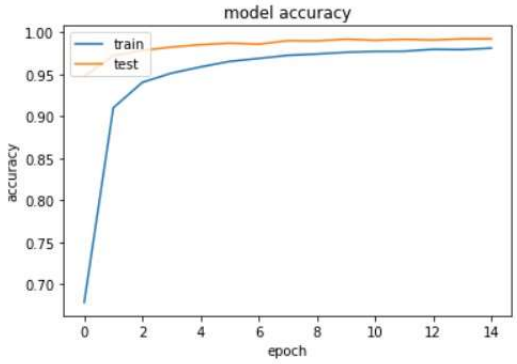

Fig. 9. Testing Accuracy vs Training Accuracy for 15 epochs in 3-layer CNN

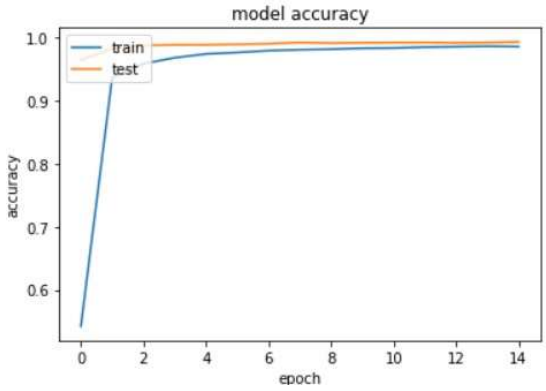

Fig. 10. Testing Accuracy vs Training Accuracy for 15 epochs in 4-layer CNN

6 Conclusion

The accuracy for the test data was checked for different CNN Layers, namely 2 layered, 3 layered and 4 layered CNN. The 4 layered CNN gave the most accuracy that was 99.25% with the least loss among all the layers compared. This was in case of 15 epochs.
A comparative study was also done with 5 epochs. The accuracy was highest with the 4 layered CNN which is 98.89%, but when compared with the 15 epochs, the accuracy has been less. This is mainly because when 15 epochs are being used, the algorithm is able to train itself more efficiently and thus a greater accuracy and a lesser loss. The final accuracy for 15 epochs is 99.25 percent which is slightly better than the previous work done and also the real time implementation of recognizing has been done with high accuracy and shorter time.

Besides this, the API built is also able to display the output perfectly for every digit that is written by the user.

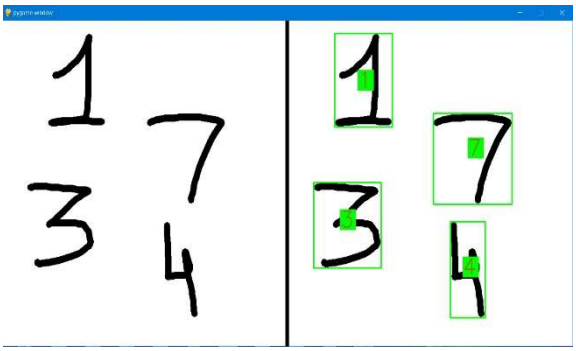

Fig. 11. Final Output

References

1. Hanmandlu M, Murthy OVR (2007) Fuzzy model-based recognition of hand-written numerals. Pattern Recogn 40(6):1840–1854
2. Graves A, Schmidhuber J (2009) Ofine handwriting recognition with multidi-mensional recurrent neural networks. In: Advances in neural information processing systems NIPS'22, vol 22. MIT Press, Vancouver, pp 545–552
3. Pal A, Singh D (2010) Handwritten english character recognition using neural network. Int J Comput Sci Commun 1(2):141–144
4. Neves RFP, Filho ANGL, Mello CAB, Zanchettin C (2011) ASVM based of-line handwritten digit recognizer. In: International conference on systems, man and cybernetics, IEEE Xplore, Brazil, 9–12 Oct 2011, pp 510–515
5. Younis KS, Alkhateeb AA (2017) A new implementation of deep neural networks for optical character recognition and face recognition. In: Proceedings of the new trends in information technology, Jordan, Apr 2017, pp 157–162
6. Dutt A, Dutt A (2017) Handwritten digit recognition using deep learning. Int J AdvRes Comput Eng Technol 6(7):990–997
7. Ghosh MMA, Maghari AY (2017) A comparative study on handwriting digit recognition using neural networks. In: International conference on promising electronic technologies, pp 77–81
8. Ali, S., Shaukat, Z., Azeem, M., Sakhawat, Z., Mahmood, T., & ur Rehman, K. (2019). An efficient and improved scheme for handwritten digit recognition based on convolutional neural network. *SN Applied Sciences*, *1*(9), 1125.