International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

Convolution Neural Network: Implementation of a Handwritten Digit Recognition System

Akshit Gambhir

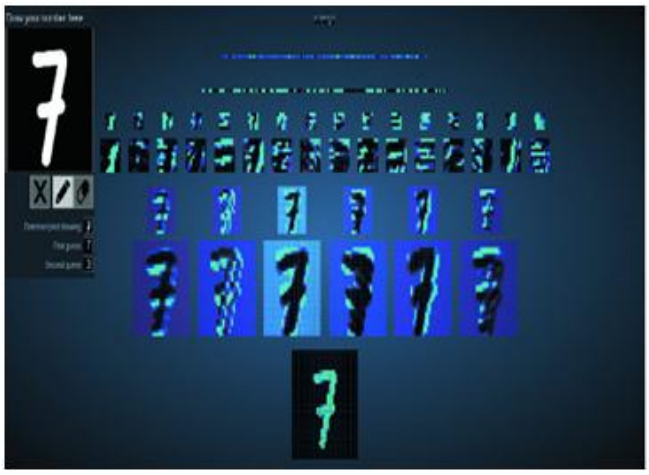
Abstract: ***Digit recognition is one of a famous problem in today's era of different fields like deep learning, machine learning, and computer vision applications. Different techniques are implemented to solve the problem of handwritten digit recognition but this paper focuses on the approach of the neural networks and specifically about the convolution neural network (CNN) approach. The CNN model in this paper is evaluated on many different factors like loss during validation, the accuracy obtained during validation and training. The model trained has a Training Loss = 0.0195 and Validation Loss = 0.0235 due to the similar shapes of some digits like (3, 5), (1, 7), (8, 5), (3, 8) and (6, 9). And the Training Accuracy and Validation Accuracy are calculated as 99.37 and 99.22 respectively.***

Keywords: Handwritten digit recognition, Convolution Neural Network, TensorFLow, Image Pre-processing, MNIST Dataset

1. Introduction

Handwritten digit recognition comes amongst one of the most recognized and basic issues in pattern recognition applications. Xuan Yang [4] worked on the classification of human written digits using the Convolutional Neural Network.

There are various applications of a digit recognition system which include automated speeding ticket generation, postal mail sorting, form entry without less human intervention, and signature processing at banks. The main problem is to focus on developing an efficient algorithm that can classify images that come from various sources. Also, the handwritten digit recognition system is a basic model to understand most of the image and pattern recognition problems. Digits that are handwritten vary from one individual to the other depending upon the handwriting, size of the character also the boldness of the character, so the general problem arises while classifying the similar digits such as 5 and 6, 1 and 7, 3 and 8, 2 and 5, 9 and 6, etc.



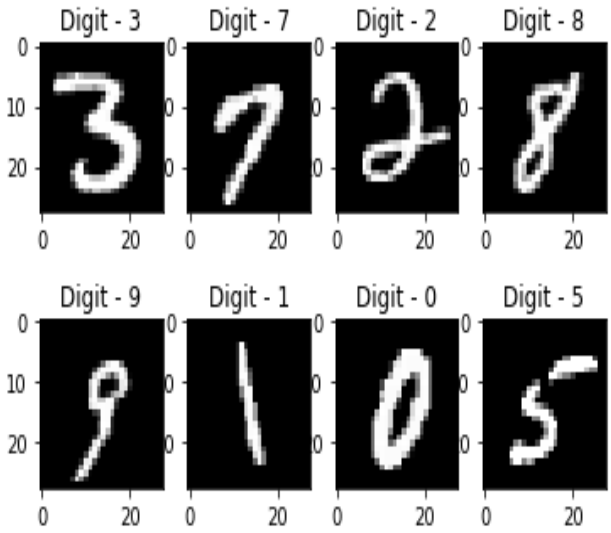
Convolution Neural Network: This neural network consists of different layers namely the pooling and convolution layers as a combination and also the dense layers of a neuron. Each neuron consists of some input weights known as connections, biases, and applied activation functions and these connections help the network to determine which neuron is to be givens more preference and in which layer. In general, a neural net involves two processes the feed-forward process that involves feeding the inputs to the model and the back propagation process that involves back propagation of the loss but the CNN works into two parts the first part is responsible for the feature extraction and the second is responsible for the classification part. The image's feature extraction is done by combining the two layers the Convolutional layer and the max or min pooling layer whereas the classification part is performed by the dense layers i.e. the fully connected layers followed by the softmax activation at the output layer.

Visualization of Image transformation representations undergoing a Convolutional model [7]

- Convolution layer – For a pixel, this layer is responsible for taking the weighted average of the neighboring pixels, and hence determining and learning new features, unlike the fully connected layers it only depends on the neighboring pixels and not all of them. This layer can have one or more filters the more the number of filters the more diverse our image feature can become, major advantages of the convolution layer over fully connected layers is the significant reduction in the number of parameters due to weight sharing i.e. a common kernel for all the neurons. Apart from that, it learns the areas of interest without human intervention.
- Pooling layer - this layer is applied to down-sample the inputs to minimize the computations power required and also for avoiding over fitting problems, the common techniques used are max pooling, average max pooling, min pooling.
- Dense layer – Also Known as a typical feed-forward network layer. Once the features are extracted from the convolution layers those features are flattened and fed to the Dense Layer and hence contain maximum trainable parameters.

Dataset used

The proposed system uses the MNIST data set [7]. This dataset carries 70,000 images in total out of which 10,000 images are used for testing and 60,000 are used for training the model [9]. It is the modified version of the NIST special database, having 10 classes that belong to digits 0 to 9.



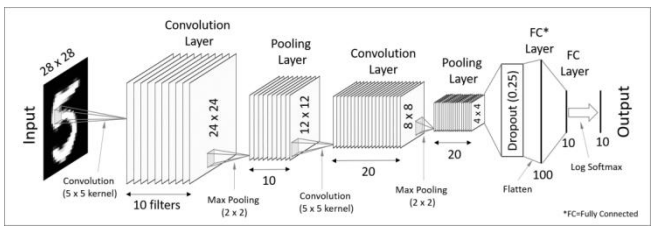
All the digit Images of this dataset are pre-processed i.e. they can be used directly and also every Image of this dataset contains homogenous properties like an image is represented as an array of 28*28 and the pixels are calculated to 784, all the images are gray scale i.e pixel with value 0 signifies black and the pixel with value 1 signifies white [6].

2. Methodology

- First, an image is taken as an input [8], then that image is preprocessed and converted into an array and that array is passed as input to the convolution model.
- We get the regions of interest to classify the objects in the image.
- All these regions are then reshaped as per the configuration of the trained model, and then each of these regions is passed to the Model.
- The trained model then extracts necessary features for each one of these regions and uses a fully connected layer that uses softmax as an activation function to interpret the output as probabilities.
- Finally, the maximum of the probability is classified as the output using np.argmax().

Layer Structure for CNN Model

The Layered structure for my model is presented below:-



Model Training

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/15
 - 6s - loss: 0.2416 - accuracy: 0.9247 - val_loss: 0.0601 - val_accuracy: 0.9803
Epoch 2/15
 - 6s - loss: 0.0772 - accuracy: 0.9765 - val_loss: 0.0400 - val_accuracy: 0.9866
Epoch 3/15
 - 6s - loss: 0.0599 - accuracy: 0.9809 - val_loss: 0.0386 - val_accuracy: 0.9872
Epoch 4/15
 - 6s - loss: 0.0492 - accuracy: 0.9849 - val_loss: 0.0272 - val_accuracy: 0.9900
Epoch 5/15
 - 6s - loss: 0.0428 - accuracy: 0.9862 - val_loss: 0.0305 - val_accuracy: 0.9902
Epoch 6/15
 - 6s - loss: 0.0376 - accuracy: 0.9883 - val_loss: 0.0247 - val_accuracy: 0.9922
Epoch 7/15
 - 6s - loss: 0.0360 - accuracy: 0.9886 - val_loss: 0.0226 - val_accuracy: 0.9925
Epoch 8/15
 - 6s - loss: 0.0325 - accuracy: 0.9900 - val_loss: 0.0236 - val_accuracy: 0.9926
Epoch 9/15
 - 6s - loss: 0.0291 - accuracy: 0.9906 - val_loss: 0.0250 - val_accuracy: 0.9916
Epoch 10/15
 - 6s - loss: 0.0269 - accuracy: 0.9914 - val_loss: 0.0241 - val_accuracy: 0.9927
Epoch 11/15
 - 7s - loss: 0.0243 - accuracy: 0.9923 - val_loss: 0.0244 - val_accuracy: 0.9914
Epoch 12/15
 - 7s - loss: 0.0238 - accuracy: 0.9923 - val_loss: 0.0271 - val_accuracy: 0.9917
Epoch 13/15
 - 6s - loss: 0.0214 - accuracy: 0.9931 - val_loss: 0.0220 - val_accuracy: 0.9929
Epoch 14/15
 - 6s - loss: 0.0211 - accuracy: 0.9931 - val_loss: 0.0263 - val_accuracy: 0.9918
Epoch 15/15
 - 6s - loss: 0.0195 - accuracy: 0.9937 - val_loss: 0.0235 - val_accuracy: 0.9922
10000/10000 [==============================] - 1s 96us/step
Test loss: 0.023492475468767225
Test accuracy: 0.9922000169754028
```

Model Summary

The model is built by using the Keras framework on top of Tensor Flow [5] and it involves two Convolution Layers with two max-pooling layers and two fully connected layers, the total number of parameters to be trained is 38390 and the time is taken to train the model has been significantly reduced and is approximately 91 seconds for the 15 epochs by utilizing the GPU of the machine. The Trained model summary is as follows:-

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None,24,24,10)	260
activation_9 (Activation)	(None,24,24,10)	0
max_pooling2d_5 (MaxPooling2	(None,12,12,10)	0
conv2d_6 (Conv2D)	(None,8,8,20)	5020
activation_10 (Activation)	(None,8,8,20)	0
max_pooling2d_6 (MaxPooling2	(None,4,4,20)	0
dropout_3 (Dropout)	(None,4,4,20)	0
flatten_3 (Flatten)	(None,320)	0
dense_5 (Dense)	(None,100)	32100
activation_11 (Activation)	(None,100)	0
dense_6 (Dense)	(None,10)	1010
activation_12 (Activation)	(None,10)	0

```
Total params: 38,390
Trainable params: 38,390
Non-trainable params: 0
```

Model Output

OUTPUT

4
1
8
9
5
6
7
3
2
8

Issues with Previous Models

The working accuracy of the previous models can be expanded by tuning hyperparameter tuning and the proposed model works on the same and excels over the previous models [1-3] in metrics like accuracy and loss by hyperparameter tuning, the accuracy has now been significantly increased in comparison to previous model [2] and now the accuracy comes to about 99.3 % and loss is calculated as 0.0235 which is also low.

3. Conclusion

The main objective of this investigation is to find a representation of an algorithm which can suitably classify a system of handwritten digits. In this paper, we used a neural network approach for the identification of human written digits. In image segmentation models, the key issue is to address the correct segmentation and feature extraction methods. The solution proposed in this paper attempts to address both aspects in terms of accuracy and loss. The validation accuracy of 99.3% is achieved in the classification process by Convolution Neural Network. This paper pursuit to Implement this classification task without the use of standard classification techniques.

References

[1] Hand Written Digit Recognition using Convolutional Neural Network (CNN) Nimisha Jain, Kumar Rahul, Ipshita Khamaru, Anish Kumar Jha, Anupam Ghosh IJIACS ISSN 2347 – 8616 Volume 6, Issue 5 May 2017.
[2] A Comprehensive Data Analysis on Handwritten Digit Recognition using Machine Learning Approach Meer Zohra, D.Rajeswara RaoSeewald, A. K. (2011). On the brittleness of handwritten digit recognition models. ISRN Machine Vision, 2012.
[3] Plamondon, R., & Srihari, S. N. Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 63-84.
[4] Yang, Xuan Jiang. “MDig: Multi-digit Recognition using Convolutional Neural Network on Mobile.” (2015).
[5] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758.
[6] Li Deng, "The MNIST Database of Handwritten Digits images for Machine Learning Research", MIT Press, November 2012.
[7] Visualizing and Understanding Convolution Networks 2013 by Matthew D Zeiler, Rob Fergus.
[8] Image Preprocessing Loading in your own data - Deep Learning basics with Python, Sentdex YouTube This video helped in the preprocessing of the image which was taken as input from the user.
[9] Source: https://www.youtube.com/watch?v=j-3vuBynnOE&list=PLQVvaa0QuDfhTox0AjmQ6tvTgMBZBEXN&index=2
[10] The MNIST DATABASE by Yann, LeCun, Corinna Cortes, Christopher J.C. Burges http://yann.lecun.com/exdb/mnist/