Effective Handwritten Digit Recognition using Deep Convolution Neural Network

Vishal Jorwal
Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070089@iitb.ac.in

***Abstract*—This article presents a straightforward method for identifying handwritten digits using a neural network that utilizes convolution. The difficulty in recognizing digits using machine learning techniques such as KNN, SVM, and SOM stems from the diverse range of handwriting styles. To overcome this challenge, the authors employed Convolution Neural Networks on the MNIST dataset, which consists of 70,000 digits from 250 different handwriting styles. The proposed method achieved an accuracy of 98.51% in real-world scenarios for identifying handwritten digits, with less than 0.1% loss during the training process using 60,000 digits, with 10,000 used for validation.**

***Index Terms*—Convolution neural networks, MNIST dataset, TensorFlow, OCR, Segmentation, Cross-Validation**

I. Introduction

Recent advancements in computer vision have drawn significant attention from artificial intelligence practitioners, particularly in the realm of deep neural networks.[1] One notable project that utilizes deep learning is Object Character Recognition (OCR), a tool that converts printed or documented letters into encoded text. OCR scans a document to extract information and store it in a digital format using either pattern recognition or segmentation. Handwritten Digit Recognition (HDR) is a subset of OCR that detects digits and is more lightweight and faster than OCR [2]. This technology is flexible and can be applied in various fields such as medical, banking, student management, and taxation processes [3].

While the human brain can interpret sophisticated images and extract data, computers view images as collections of pixels. Information in images is extracted from the pixel values, and features such as shape, size, and color are identified and sent to the visual cortex for analysis. Neural networks, inspired by the human brain, utilize layered architectures and mathematical functions to learn and identify patterns. For image classification, Convolution Neural Networks (CNNs) are optimal, whereas Long Short-Term Memory Networks (LSTMs) are better suited for speech recognition [4].

In the proposed method, a convolution neural network architecture with ReLU and sigmoid activation functions is used to predict real-world handwritten digits by training the network with the MNIST dataset.[3] The input layer consists of 784 neurons corresponding to the 28x28 greyscale image's grey level values, and the output layer contains neurons equal to the number of target classes (ten for handwritten digit recognition) [5]. The hidden layers, which can be changed depending on the task, lie between the input and output layers and are arbitrary. The highest activation neuron in the presentation layer is the network's choice of class.

Fig. 1: Sample MNIST data

II. Methodology

The focus of this study was the development of a model for handwritten digit recognition, which was trained using the MNIST dataset. This dataset contains 70,000 raster images of handwritten digits from 250 sources, and it was divided into a training set of 60,000 images and a validation set of 10,000 images. The MNIST data was presented in the IDX file format and is illustrated in Figure 1. The proposed method was divided into several stages, as shown in Figure 2: pre-processing, data encoding, model construction, training and validation, and model evaluation and prediction. All of these steps came after loading the dataset, which was a necessary step for the entire process.

A. Pre-Processing

After loading the data, we separated the data into X and y where X is the image, and y is the label corresponding to X.

Fig. 2: Flowchart

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 flatten (Flatten)           (None, 1600)              0

 dense (Dense)               (None, 128)               204928

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 225,034
Trainable params: 225,034
Non-trainable params: 0
_________________________________________________________________
None
```

Fig. 3: Model sumary

As shown in Figure 4, the first layer/input layer for our model is convolution. Convolution takes each pixel as a neuron, so we need to reshape the images such that each pixel value is in its own space, thus converting a 28x28 matrix of grayscale values into 28x28x1 tensor. With the right dimensions for all the images, we can split the images into train and test for further steps [6] [7] [8].

B. Data Encoding

This step is not mandatory, but it is relevant since we are utilizing the cross-categorical entropy as the loss function. Therefore, we need to inform the network that the provided labels are categorical in nature.

C. Model Construction

Once the data is encoded, we can input the images and labels into our model. Figure 3 shows a summary of our model [9] [10]. Our model has two parts: feature extraction with convolution and binary classification. To extract features from the image, we use convolution and max-pooling. First, we apply 32 convolution filters of size 3x3 to a 28x28 image. Then, we use a max-pooling layer with a pooling size of 2x2. After that, we apply another convolution layer with 64 filters of size 3x3. Finally, we get 7x7 images that are flattened into a series of 128 values. These values are then mapped to a dense layer of 128 neurons, which is connected to the categorical output layer of 10 neurons. It's worth noting that this step is optional, but we need to specify to the network that the given labels are categorical in nature since we are using cross-categorical entropy as a loss function.

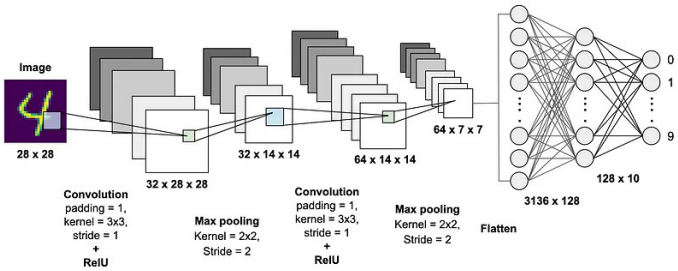

Fig. 4: Proposed model

(***Source**:https://towardsdatascience.com/mnist-handwritten-digits-classification-using-a-convolutional-neural-network-cnn-af5fafbc35e9)*

D. Training & Validation

Once the model was constructed [11], we used the standard approach of compiling it with an Adam optimizer and cross-entropy loss function for convolution neural networks. Training of the model was carried out on the training data for 100 iterations; however, overfitting can occur with an increase in the number of iterations. Finally, to validate the model, we utilized the test data.

E. Model Evaluation & Prediction

To predict real-world image classification, we need to preprocess the images a bit since the model was trained using grayscale raster images. The image preprocessing steps include:

1) Load the image
2) Convert the image to grayscale
3) Resize the image to 28x28
4) Convert the image into a matrix format
5) Reshape the matrix into 28x28x1

After pre-processing the image, we predict its label by passing the pre-processed image through the neural network. The output obtained is a list of 10 activation values ranging from 0 to 9. The position in the list that has the highest value corresponds to the predicted label for the image [12].

III. RESULTS AND DISCUSSION

The model we created is designed to handle real-world data. However, real-world images are quite different from the raster images in the MNIST dataset. Thus, we had to perform extensive pre-processing to transform a real-world image into a raster image-like format.

A. Accuracy score

Our model reached 99.91% training accuracy after 100 epochs and 98.86% validation accuracy with 0.0622 training loss after 100 epochs and 1.4198 validation loss as shown in Fig 5:-

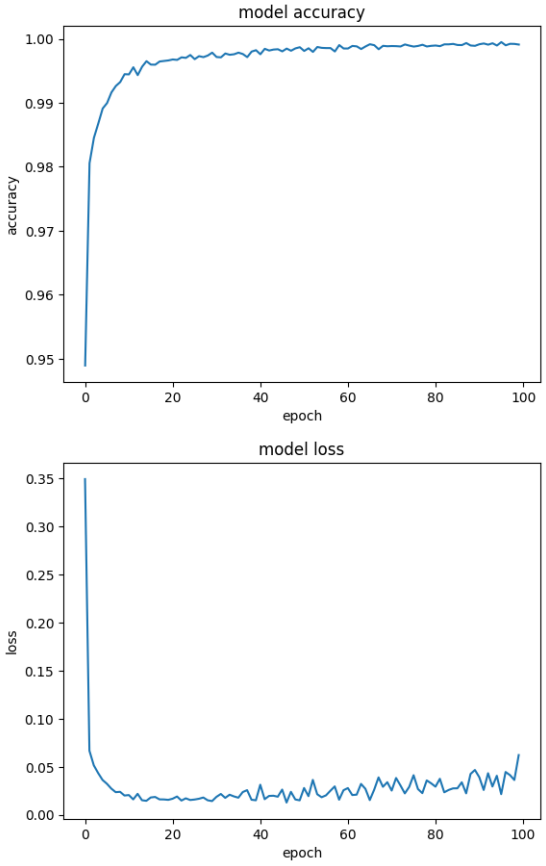

Fig. 5: Loss and Accuracy Learning Curves

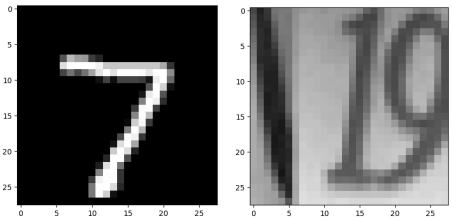

Fig. 6: Raster vs. Real image prediction

B. Prediction

Our model is able to recognize computer-generated digits as well as handwritten digits. Computer-generated digit prediction is more accurate compared to real-world digit prediction.Example as follows in Fig 6

IV. CONCLUSION

The CNN performed well in recognizing handwritten images, achieving an accuracy of 98.86% and successfully identifying real-world images. The loss percentage during both training and evaluation was negligible, less than 0.1. The only challenge is the presence of noise in real-world images, which requires attention. The learning rate of the model is strongly influenced by the number of dense neurons and the cross-validation metric.

REFERENCES

[1] Yin, Yue, et al. "Deep learning-aided OCR techniques for Chinese uppercase characters in the application of Internet of Things." IEEE Access 7 (2019): 47043-47049.
[2] Sudhakar, S., et al. "Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires." Computer Communications 149 (2020): 1-16.
[3] Venkateswarlu, N. B. "New raster, adaptive document binarisation technique." Electronics Letters 30.25 (1994): 2114-2115.
[4] Lee, Seong-Whan, and Sang-Yup Kim. "Integrated segmentation and recognition of handwritten numerals with cascade neural network." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 29.2 (1999): 285-290.
[5] Aly, Saleh, and Ahmed Mohamed. "Unknown-length handwritten numeral string recognition using cascade of pca-svmnet classifiers." IEEE Access 7 (2019): 52024-52034.
[6] Prakash, Kolla Bhanu. "Information extraction in current Indian web documents." Int. J. Eng. Technol.(UAE) 7.2 (2018): 68-71.
[7] Prakash, Kolla Bhanu, M. A. Dorai Ranga Swamy, and A. Raja Raman. "A Neuron Model for Documents Containing Multilingual Indian Texts (ICCCT 2010)." (2010).
[8] Kolla, Bhanu Prakash, and Arun Raja Raman. "data engineered content extraction studies for Indian web pages." Computational Intelligence in Data Mining: Proceedings of the International Conference on CIDM 2017. Springer Singapore, 2019.
[9] Prakash, Kolla Bhanu, and M. A. Dorai Rangaswamy. "Content extraction studies using neural network and attribute generation." Indian Journal of Science and Technology 9.22 (2016): 1-10.
[10] Prakash, Kolla Bhanu, MA Dorai Rangaswamy, and Arun Raja Raman. "Text studies towards multi-lingual content mining for web communication." Trendz in Information Sciences & Computing (TISC2010). IEEE, 2010.
[11] Prakash12, Kolla Bhanu, and MA Dorai Rangaswamy. "Content Extraction of Biological Datasets Using Soft Computing Techniques." (2016).
[12] Sudhakar, S., et al. "Optimizing joins in a map-reduce for data storage and retrieval performance analysis of query processing in HDFS for big data." International Journal of Advanced Trends in Computer Science and Engineering,(IJATCSE) 8.5 (2019): 2062-2067.
[13] Bharadwaj, Y. S., et al. "Effective handwritten digit recognition using deep convolution neural network." International Journal of Advanced Trends in Computer Science and Engineering 9.2 (2020): 1335-1339.