IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 17, Issue 3, Ser. II (May – June 2022), PP 17-22
www.iosrjournals.org

Number Script Recognizer Using Cnn

KVBL Deepthi[1], K. Rupesh Sai Chand Reddy[2], G. Keerthi[3], Y. Sushma[4]
[1](Assistant professor, ECE Department, MVSR Engineering College, India)
[234](Graduate student,ECE Department, MVSR Engineering College, India)

Abstract: *Accurate text identification and recognition are critical in image processing and document analysis.Some numeric symbols with similar appearances are hard to differentiate.As a result the best deep learning and convolutional neural networks are employed.In this project we use MNIST dataset to train the model.A digit from the selected dataset is given to deep learning architecture in which it undergoes convolution,pooling and dense layers.Dense layers are typical neural network layers also called as decision making layers.This project can handle a variety of numerals and provide an accuracy upto 97.24%*

Key Word: *Handwritten Digit Recognition, Convolutional Neural Networks (CNN), MNIST Dataset, OpenCV, Tensorflow, Adam Optimizer, Backpropagation, Hidden layers, Raspberry Pi*

Date of Submission: 13-06-2022 Date of Acceptance: 29-06-2022

I. Introduction

Handwritten digit recognition is the process to provide the ability to machines to recognize human handwritten digits. It is not an easy task for the machine because handwritten digits are not perfect, they vary from person-to-person.

Developers are putting all their strength to make machines more intelligent, and smarter than humans. Deep learning is one such technique that contributes to developers enhancing machines. Various types of neural network architectures are used by deep learning algorithms to solve different types of problems.In this project we are using Convolutional neural networks.

This whole project is going to be written in python language, which has the privilege of providing libraries namely OpenCV and TensorFlow to precisely train the CNN model for Recognition. Then we are going to use the data from the learning stage to allow the Pi Camera to read and recognize digits.Training a model for various handwritten digits of this world is unthinkable as handwriting is novel to each person. So we can prepare a model utilizing an enormous dataset of manually written digits like the MNIST dataset and test them on other handwritten digit styles.

II. Convolutional Neural Networks

CNN is a type of deep learning model for processing data that has a grid pattern, such as images and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers.

a) Convolution Layer

A convolution layer is the fundamental part of the CNN architecture which performs feature extraction, it normally comprises linear and nonlinear tasks, i.e., convolution operation and activation function.

A convolution layer transforms the input image by convoluting it with a kernel (or filter). A kernel means a small matrix whose width and height are smaller than the image that is to be convolved. Kernel is also called a convolution matrix. Fig:2.1 and Fig2.2 demonstrate the sample kernel which slides across the width and height of the input image and dot product of the kernel and the image is computed at every spatial position.

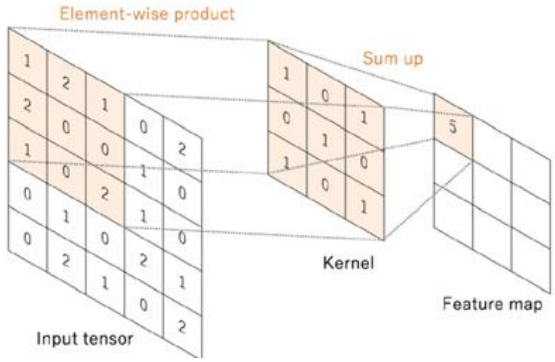

Fig:2.1

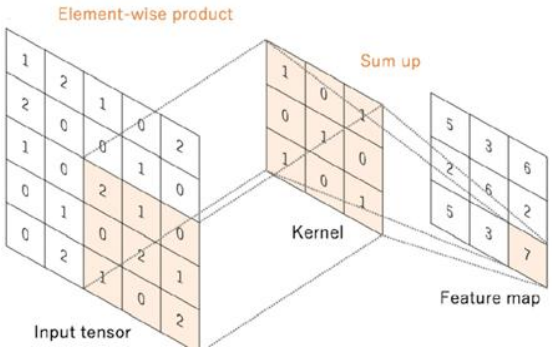

Fig:2.2

An activation function is the next component of the convolutional layer to increase the non-linearity of the output. Here we are using the ReLu function as an activation function in a convolution layer.

The rectified linear activation(ReLu) is the default activation when developing multilayer Perceptron and convolutional neural networks.

b) Pooling Layer

Pooling layer is utilized to reduce the size of the input image.In a CNN, a convolutional layer is normally trailed by a pooling layer.

Pooling layer is usually added to accelerate computation and to make the detected features more robust.Max Pooling results the maximum value from the portion of the image covered by the Kernel. Max Pooling likewise proceeds as a Noise Suppressant. It also discards the noisy activations altogether and furthermore performs de-noising alongside dimensionality decrease.

On the other hand, Average Pooling just performs dimensionality decrease as a noise suppressing mechanism. Subsequently, we can say that Max Pooling plays out significantly better compared to Average Pooling. Fig 2.3 gives the illustration of how max pooling works

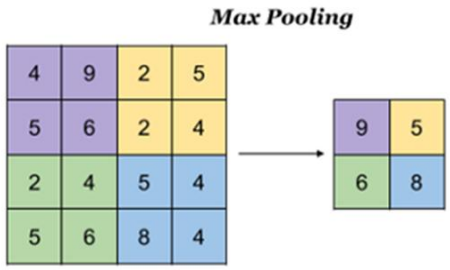

Fig 2.3

The Convolutional Layer along with the Pooling Layer, form the i-th layer of a Convolutional Neural Network. Contingent upon the complexities in the images, the amount of such layers may be extended for getting low-level details. However, at the cost of more computational power.

c) Fully Connected layers

The Fully-Connected layer also known as dense layers which contains neurons is learning a possibly non-linear function in that space.

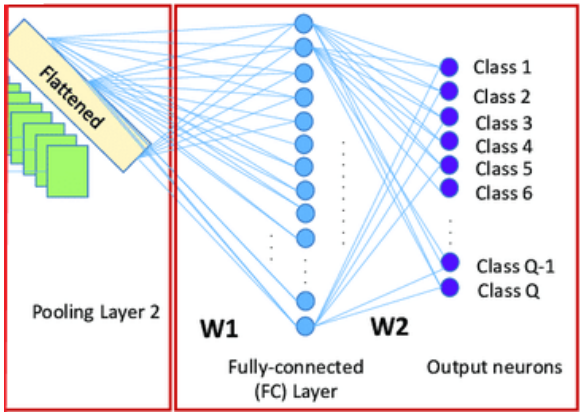

Fig2.4

This successfully enabled model will be able to understand the features. The final output is flattened as shown in Fig 2.4 and then fed to a regular Neural Network for classification purposes.

Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

The softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution.

Softmax is used for multi-classification in the Logistic Regression model, whereas Sigmoid is used for binary classification in the Logistic Regression model.

$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for j} = 1,\ldots,\text{K}$$

The output layer of the Neural Network classifier is a vector of raw values.For instance consider raw output values from our neuron network as: [-0.5, 1.2, -0.1, 2.4].

SoftMaxoutput values: 0.04, 0.21, 0.05, 0.70

SoftMax input values: -0.5, 1.2, -0.1, 2

By using Softmax function the outputs are interrelated. The Softmax probabilities will always sum to one by design: 0.04 + 0.21 + 0.05 + 0.70 = 1.00. In this case, the likelihood of one class is increased, the other has to decrease by the same amount.

III. Literature Survey

M. Hanmandlu, O.V. Ramana Murthy has given within their study the popularity of written Hindi and English numerals by representing them in the kind of exponential membership functions that function a fuzzy model. popularity is administered by modifying the exponential membership functions fitted to the fuzzy sets. These fuzzy sets are units derived from options consisting of normalized distances obtained from the Box approach. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numerals.

R. Bajaj, L. Dey, S. Chaudhari used 3 completely different styles of options, namely, the density options, moment options and descriptive part options for classification of script Numerals. They projected multi classifier connectionist design for increasing the popularity of dependableness and that they obtained 89.6% accuracy for written script numerals.

Yoshimasa Kimura gave a piece on a way to choose options for Digit Recognition victimization Genetic rule. The author proposes a unique technique of feature choice for Digit recognition victimization genetic algorithms (GA). The projected technique selects solely the cistrons that the popularity rate of coaching samples exceeds than the preset threshold as a candidate of the parent gene and adopts a discount magnitude relation within the range of options used for recognition because of the fitness price.

IV. Design Methodology

To make this project intact and minimize manual errors, Dataset for training the model is taken from online MNIST Database which had the courtesy of providing nearly 60,000 preprocessed images to fit the data into the model design. Later the data is introduced with further transformation techniques to make the model flexible and immune to distortions. The model is constructed using Keras from TensorFlow library which provides backend support for the Neural Network Model.

Further Hardware support is provided by the Raspberry Pi 4 Board with additional modules namely PiCamera V1.3, I2C Chip and LCD 16X2 Display .Program Code for these modules are written in Python 3.7 without violating library requirements of TensorFlow and OpenCV.

Algorithm

a)Training the Model

Step 1: Setting up local Variables
Step 2: Loading datasets from MNIST database
Step 3: Scaling, Reshaping and Normalizing the Data
Step 4: Creating a Skeleton for Model
Step 5: Training the Model
Step 6: Saving the Model

b)Recognizing Using the Trained Model

Step 1: Capturing Real Time Image into an array
Step 2: Gray-scaling,Scaling and Inverting the image
Step 3: Scaling, Reshaping and Normalizing the Data
Step 4: Feeding into Trained Model
Step 5: Predicting The Output

V. Implementation

Raspberry Pi 4 is booted with latest OS Debian Bullseye Release, has 4gb RAM, all the mentioned libraries are installed accordingly.

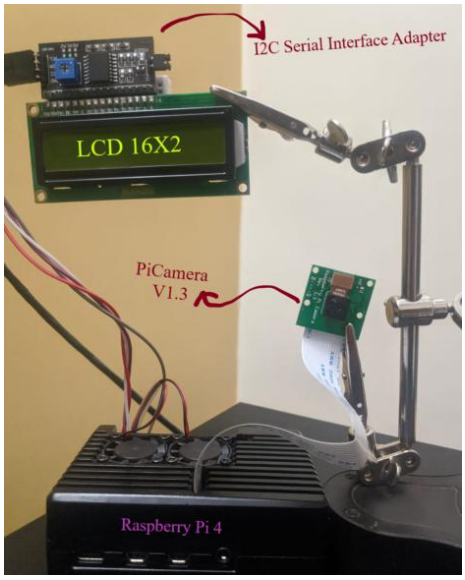

Fig:5.1

16X2 LCD is interfaced using I2C Serial to Parallel Interfacing Adapter. Raspberry Pi 4 Board is connected to the PiCamera for capturing input image frames and I2C Adapter for transmitting output data to LCD. Following Fig:5.1 displays the complete setup of the model.

Results

After an immense series of epochs, the trained model underwent real time testing, the Fig:6.1 depicts the frame captured by the Picamera and the Fig:6.2.a and 6.2.b shows predicted output displayed on the LCD which was integrated to the Raspberry Pi 4 Board.

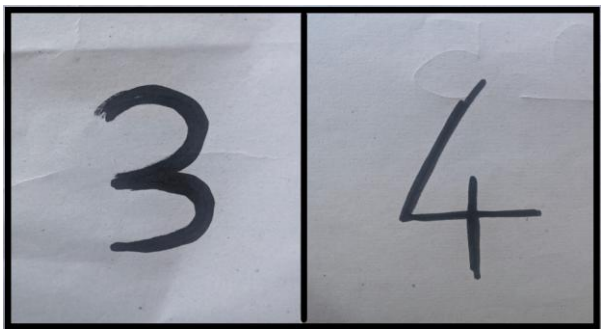

Fig:6.1

Fig:6.2.a Fig:6.2.b

And out of 30 real handwritten digit samples the Model could correctly recognize 28 digits with an accuracy of 97.24%

VI. Conclusion

For the recognition of handwritten numerals this research offered a convolutional neural network and deep learning algorithms.The proposed technique in this paper is to train the model first using MNIST dataset and applying deep learning algorithms subsequently.Using CNN in this project makes it accurate up to 97.24%. CNN makes this system more reliable, less complex and easy to understand as well.CNN helps in reducing the machine efforts to recognise the handwritten digits.Using Pi camera in this paper makes the job easy to understand.Based on obtained results we came to know that high rates of recognition made by our CNN model compared to others.So we recommend Using CNN is the best of all existing methods

References

[1]. Hanmandlu, M., & Murthy, O. R. (2007). "Fuzzy model based recognition of handwritten numerals. Pattern Recognition",40(6), 1840- 1854. doi:10.1016/j.patcog.2006.08.014

[2]. Cecotti, H., & Belaid, A. (2005). "Rejection strategy for convolutional neural network by adaptive topology applied to handwritten digits recognition". Eighth International Conference on Document Analysis and Recognition (ICD R05),765-769. doi:10.1109/icdar.2005.200

[3]. Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. Neural Computation",22(12), 3207-3220. doi:10.1162/neco_a_00052

[4]. Dalal, N., & Triggs, B. (n.d.). "Histograms of Oriented Gradients for Human Detection". 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05),886-893. doi:10.1109/cvpr.2005.177

[5]. Duerr, B., Haettich, W., Tropf, H., & Winkler, G. (1980). "A combination of statistical and syntactic pattern recognition applied to classification of unconstrained handwritten numerals. Pattern Recognition",12(3), 189-199.doi:10.1016/0031-3203(80)90043-6

[6]. Dinov, I. D. (2018). "Variable/Feature Selection. Data Science and Predictive analytics",557- 572. doi:10.1007/978-3-319-72347-1_17

[7]. Elleuch, M., Maalej, R., & Kherallah, M. (2016). "A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition". Procedia Computer Science,80, 1712-1723. doi:10.1016/j.procs.2016.05.512

[8]. Biswas, M., Islam, R., Shom, G. K., Shopon, M., Mohammed, N., Momen, S., & Abedin, A. (2017). BanglaLekha-Isolated: "A multi- purpose comprehensive data-set of Handwritten Bangla Isolated characters". Data in Brief, 12, 103-107. doi:10.1016/j.dib.2017.03.035

[9]. Hochuli, A., Oliveira, L., Jr, A. B., & Sabourin, R. (2018). "Handwritten digit segmentation: Is it still necessary? Pattern Recognition",78, 1-11. doi:10.1016/j.patcog.2018.01.00486

[10]. Karimi, H., Esfahanimehr, A., Mosleh, M., Ghadam, F. M., Salehpour, S., & Medhati, O. (2015). "Persian Handwritten Digit Recognition Using Ensemble Classifiers". Procedia Computer Science,73, 416-425. doi:10.1016/j.procs.2015.12.018

[11]. Kavallieratou, E., Likforman-Sulem, L., & Vasilopoulos, N. (2018). "Slant Removal Technique for Historical Document Images". Journal of Imaging,4(6), 80. doi:10.3390/jimaging4060080

[12]. Keysers, D., Deselaers, T., Gollan, C., & Ney, H. (2007). "Deformation Models for Image Recognition". IEEE Transactions on Pattern Analysis and Machine Intelligence,29(8), 1422- 1435. doi:10.1109/tpami.2007.1153
[13]. Lauer, F., Suen, C. Y., & Bloch, G. (2007). "A trainable feature extractor for handwritten digit recognition. Pattern Recognition",40(6), 1816-1824. doi:10.1016/j.patcog.2006.10.011
[14]. Li, F., & Gao, S. (2010). "Character Recognition System Based on Back-Propagation Neural Network." 2010 International Conference on Machine Vision and Human-machine Interface,393-396. doi:10.1109/mvhi.2010.185