**Bangla Handwritten Digit Recognition**

| Ahmad Umar Mahdi  193-15-1071 (PC-A)  ahmad15-1071@diu.edu.bd  Department of CSE  Daffodil International University | Bristy Saha  193-15-1077 (PC-A)  bristy15-1077@diu.edu.bd  B.Sc. of CSE  Daffodil International University | Jannat Ara Shormee  193-15-1080 (PC-A)  Jannat15-1080@diu.edu.bd  B.Sc. of CSE  Daffodil International University | Shoaba Razzak  193-15-2958 (PC-A)  shoaba15-2958@diu.edu.bd  B.Sc. of CSE  Daffodil International University |
| --- | --- | --- | --- |

**Abstract:** Handwritten digit recognition is a technique for automatically recognizing and detecting handwritten digital data via Machine Learning models. To meet the demand for paperless offices and to greatly improve work efficiency, a proper Bangla handwritten digit recognition system needs to be researched and implemented. It is not easy to recognize Bengali handwritten digits due to differences in shape, size and writing style. In this paper, we will use Machine Learning algorithms to recognize the Bangla handwritten digit. Plentiful works have already been done in English, Arabic, Chinese, Japanese handwritten script. Some work on Bangla also has been done but there is space for development.

**Introduction:** Handwritten digital recognition development research is rapidly evolving and redesigning automation fields such as automatic check reading, automatic number plate reading, digital postal service, optical image recognition (OCR), etc. Due to the different aspects of its use, computer vision researchers really feel the need to work on and improve on it - actually quality and performance. But handwriting identification is more challenging than typed letters. Because different people write differently and that creates a high level of diversity in the writing style. Also, there are some similarities between the shapes of the different characters. Overwriting situations make it even more challenging to properly classify handwritten numbers. There are many more applications that can use this Bangla Digit Recognition System. Such as Bangla Handwritten Letters Base OCR (Optical Character Recognition), Picture to Text to Speech, Bangla ID Card Reading, Number Plate Reading, Vehicle Tracking, Post Office Automation etc.

**Research Methodology:** Our proposed method is mainly separated into three stages. Those are-

1. Dataset Description
2. Data Preprocessing
3. Model Training

**I. Dataset Description:**

In this paper, we used the BHaND dataset from GitHub as a primary dataset to train the model which contains 70000 handwritten Bangla digits. From which 50000 digits are used for training and validation. 20000 digits are used for testing. BHaND data is represented in the png file format and looks like in figure 1. The images are gray-scale and the dimension is 32 × 32. Each row of these sets has two dimensions, the first dimension is the image itself and the second dimension is the label. Here the label means which digit (0 to 9 in Bengali) this image represents. Thus the training set has 50,000 rows and two columns. Other sets are also similar in nature.



Fig. 1: Dataset Visualization

Each of the samples are in gray-scale form normalized to the intensity range [0,1] and the intensity were inverted to make black pixels 1 and white pixels 0. The images were then reshaped to a single dimension making 1024 (32\*32) features in total (fig 2). Thus the first dimension of each row in each set is a 1024 dimension vector and the second dimension of each row in each dataset is a regular integer number (0 to 9).

All these samples were then grouped together in a single pickle file using python's serializer Pickle, then they were compressed by GZIP compression algorithm for faster transmission.

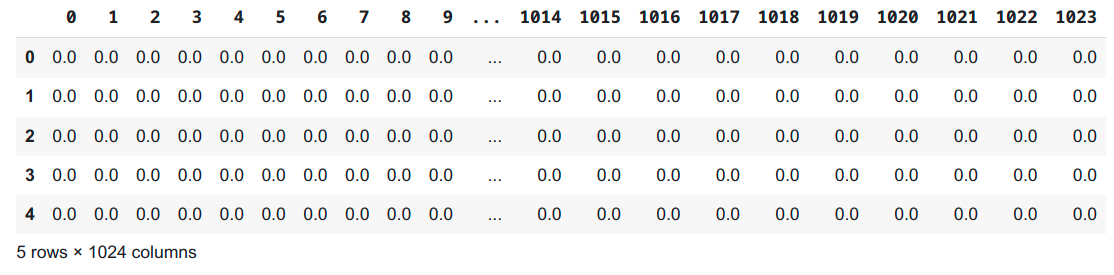
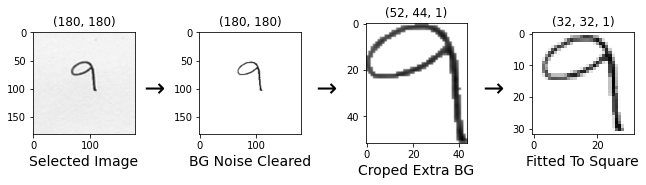


Fig. 2: DataFrame

**II. Data Preprocessing:**

First we have to invert the image so that black pixels are close to 1 and the white pixels are close to 0. Then we have to normalize the image by converting values that are less than 0.5 to 0. Then we can crop the extra background to get better results. Now we shape the data to our required square shape 32x32. Now, we have to compress the data to a GZIP file to use it easily in our project.



Now we load that GZIP compressed file in our project and we decompress the file. After decompressing we load the pickle file to three two dimensional Tuple sets (trainSet, validSet, testSet). Then we separate that tuple to (X\_train, y\_train) Arrays. Here, X\_train is our png data and y\_train is our label of that data. Then we convert those two Arrays to Pandas DataFrame. We do similar operations for validSet and testSet.

Now, we have to reshape our data by transforming X\_train sets from a 50000x1024 dataframe to a 50000x32x32x1 4D tensor for Keras modeling. Here, we have 50000 images in the train set which are in 32x32 pixels and the 4th dimension is for color profile. Our color profile is 1 means those are gray-scale images.

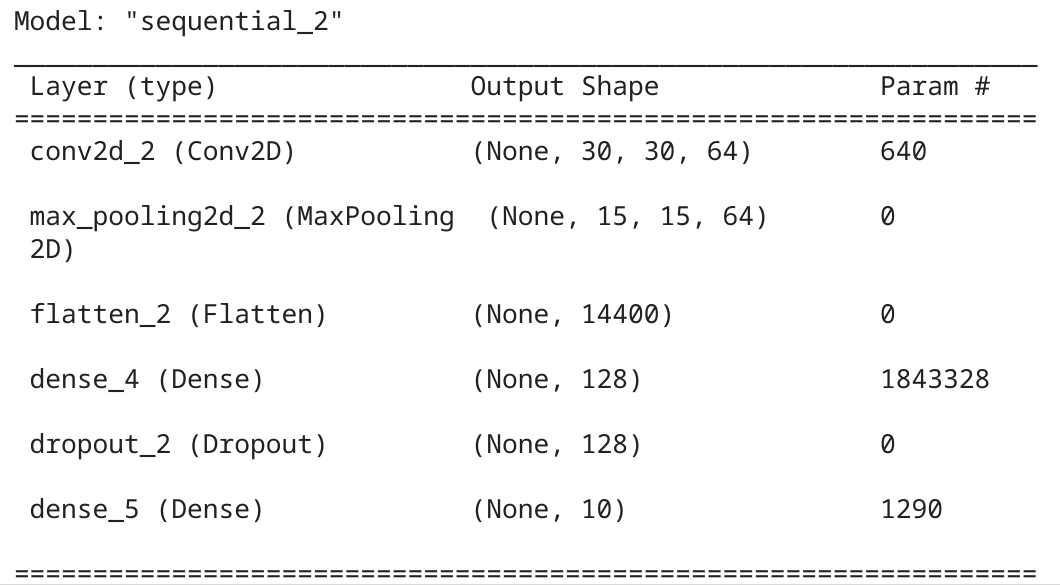
**III. Model training:**

To train our model we used Sequential Modeling. A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. It is a way of creating deep learning models where an instance of the Sequential class is created and model layers are created and added to it.

We used 5 types of layers in our Model. Those are-

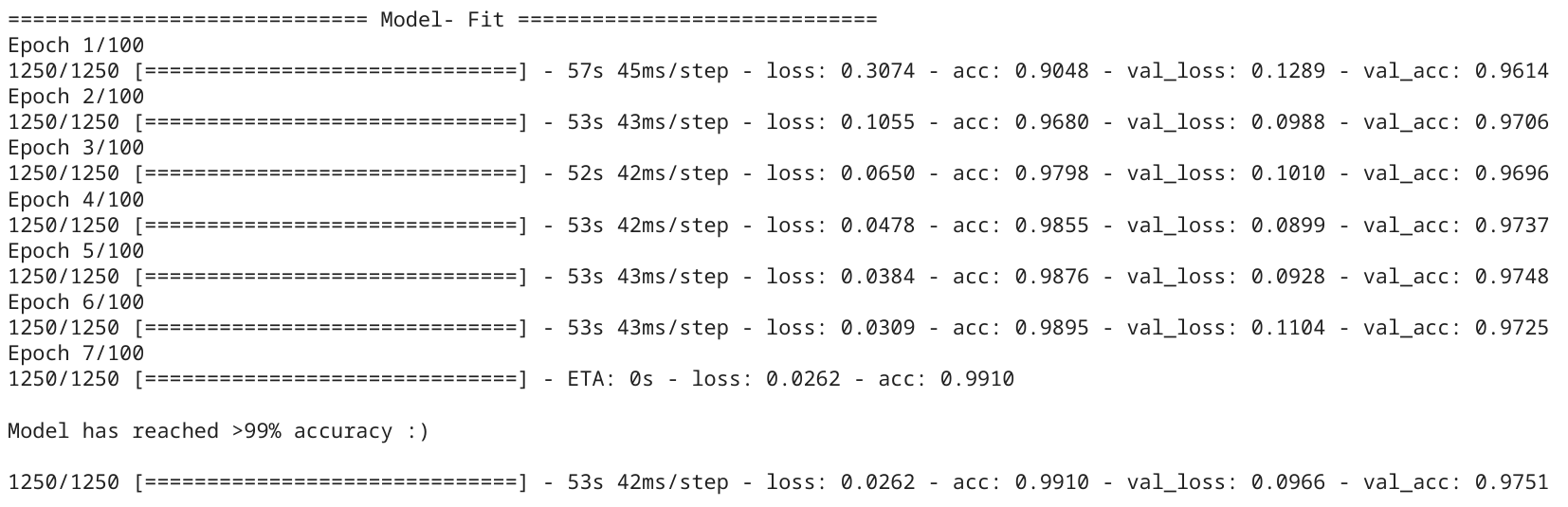
1. **Conv2D:** Keras Conv2D is a 2D Convolution layer. This creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.
2. **MaxPool2D:** Max pooling operation for 2D spatial data. Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool\_size) for each channel of the input.
3. **Flatten:** It flattens the multi-dimensional input tensors into a single dimension, so you can model your input layer and build your neural network model, then pass those data into every single neuron of the model effectively.
4. **Dense:** Dense layer is the regular deeply connected neural network layer. Each neuron in the dense layer receives input from all neurons of its previous layer. It is the most common and frequently used layer.
5. **Dropout:** The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.

Here is our Model Summary:



**Result and Discussion:**

We were able to get 99.1% accuracy after 7 Eproch in Model fitting. Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained.

****

We can visualize our accuracy and loss by showing the Confusion Matrix. Here is our Confusion Matrix:

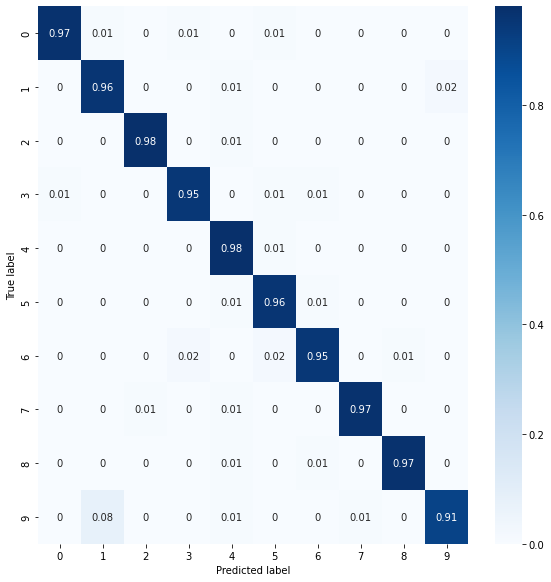


Fig. 3: Confusion Matrix

**Conclusion:**

In this research, we proposed to use deep learning approaches for handwritten Bangla digit recognition for lesser epochs and less computation time. We evaluated the performance of Keras Modeling. Among some of the observations, the maximum accuracy in the performance was found to be 99.1%. Research work is currently progressing to develop more.

**References:**

1. Al Mehdi Saadat Chowdhury and M. Shahidur Rahman :

BHaND Dataset: Towards optimal CNN parameters for bengali handwritten numerals recognition

1. AKM Shahariar Azad Rabby, Sheikh Abujar, Sadeka Haque and Syed Akhter Hossain :

Bangla Handwritten Digit Recognition Using Convolutional Neural Network (DIU)

1. Fathma Siddique, Shadman Sakib and Md. Abu Bakr Siddique :

Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers

1. Bhattacharya, U., Chaudhuri, B.:

Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. IEEE Trans. Pattern Anal. Mach. Intell. 31, 444–457

1. Biswas, M., Islam, R., Gautam, K.S., Shopon, Md., Mohammed, N., Momen, S., Abedin, A.:

BanglaLekha-Isolated: a multi-purpose comprehensive dataset of Handwritten Bangla Isolated

characters.

1. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Cmaterdb1:

A database of unconstrained handwritten Bangla and Bangla-English mixed script document images.

**Project Link (colab):** <https://colab.research.google.com/drive/1WLeofVHDImkWPpHXcyYkW2imAr7Fa55Y?usp=sharing>