Multimedia Computing Project 1

Topic: Handwritten digit recognition using pre-trained CNN

Members:

Mareva Zenelaj	150150906
Fidan Khalilbayli	150150905

1. *Description*

The human eye can easily identify handwritten digits, our brain can predict the designated number even if the writing is sloppy or not distinguishable enough. In this project we will use pre-trained CNN models to portray how machines are taught to do the same. We plan to use different CNN models and compare the different networks by accuracy rates. Like other machine learning researchers, we will use Modified National Institute of Standards and Technology database (MNIST) [1] on handwritten digits identification in order to complete the project.

The database contains 60,000 training images and 10,000 testing images each of size 28x28 [2]. An example of Sample Digits from MNIST dataset is given in figure 1. There will be a training phase but also a prediction one where we will show if the models can detect the correct number.

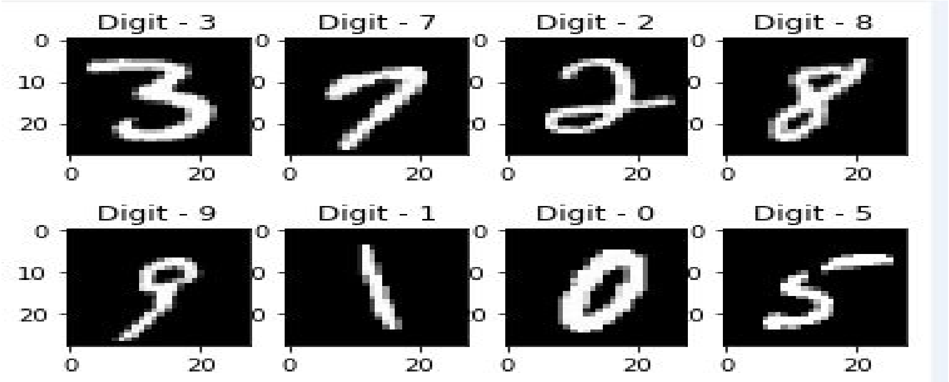

Figure 1: Sample Digits from MNIST dataset

2. *Design*

At first, the MNIST database will be loaded through the keras api. The training and test images' data and their corresponding labels will be held in defined variables. In order to prepared the data, certain modifications such as resizing the images (since CNN only accepts 4D vectors), pixel normalization will be applied on the images. Also, the label data will be converted into binary class matrices. Then the model's architecture will be designated. In this project, 2 convolution layers, a pooling layer, a fully connected layer, softmax layer and dropout layer will be utilized [2]. The function of each layer can be indicated as:

- Convolutional layers: Passing the result to the next layer [3].
- Pooling layer: Combines the outputs of neuron clusters at one layer into a single neuron in the next layer [4].
- Fully connected layer: connect every neuron in one layer to every neuron in another layer.
- Softmax loss layer: Predict a single class of K mutually exclusive classes.
- Dropout layer: Regularize the model to reduce the over-fitting.

Categorical_crossentropy loss function will be used to compile the model [5]. For model parameters optimization AdaDelta technique will be implemented [5]. Later the model is trained with the training data and evaluated in terms of accuracy percentage. A visualization of the workflow is given in Figure 2.

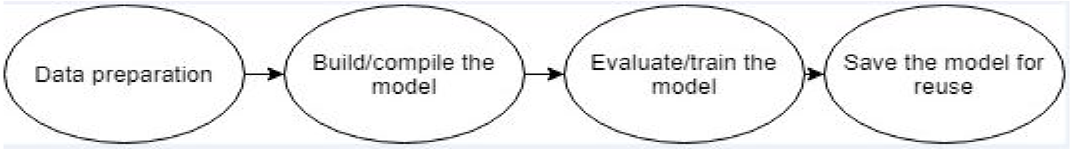

Figure 2: Workflow

3. Components Of The System

When it comes to the components of the system, we will divide the work into n-many parts where n denotes the number of different model there will be used to compare. In order to train the models we will be using Tensorflow, at some a Keras capsule, and most importantly GPU's. The dataset will be retrieved from MNIST and the programming language used will be Python. OpenCV, sklearn and Numpy will be used as well in order to aid the workload.

The models will have two main parts: the training and the prediction. Moreover, if needed we will implement a k-fold cross validation technique in order to improve the accuracy of predictions.

4. Deliverables

The deliverables of this project will be comparison report among different models that will be tested using MNIST's dataset retrieved by using Keras. Also, an out of scope deliverable would be building an user interface module that would let the user write a number in the black canvas and the model would classify it from 0 to 9. However the latter might be unreachable given the workload.

5. References

[1]. Y. LeCun, C. Cortes and C. Burges, "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges", *Yann.lecun.com*, 2019. [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed: 22- Feb- 2019].

[2]. Mahapatra, S. (n.d.) A simple 2D CNN for MNIST digit recognition
Retrieved from:
https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a

[3].*"Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation"*. *DeepLearning 0.1*.Retrieved 31 August 2013.

[4].Dan, C., Meier, U., Masci, J., Gambardella, L.,; Schmidhuber, J. (2011). *"Flexible, High Performance Convolutional Neural Networks for Image Classification" Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two*. **2**: 1237–1242. Retrieved 17 November
2013

[5].Dwived, A.(2019) *Handwritten Digit Recognition with CNN*
Retrieved from:https://datascienceplus.com/handwritten-digit-recognition-with-cnn/

5. Appendix

Various implementations of the project can be found in the following links:

1. https://github.com/asanka94/Handwritten-Digit-Recognition/blob/master/app.py
2. https://github.com/anujdutt9/Handwritten-Digit-Recognition-using-Deep-Learning/blob/master/CNN_Keras/CNN_MNIST.py
3. https://github.com/DenimaMab/PredictHandwrittenDigits/blob/master/python/convolutional-model/model.py
4. https://github.com/anushuk/CNN-digit-classifier/blob/master/handwritten_digit_recognition.ipynb
5. https://github.com/ardamavi/Digit-Classifier
6. https://github.com/gary30404/convolutional-neural-network-from-scratch-python
7. https://github.com/Curt-Park/handwritten_digit_recognition