Compsys 302 project

Handwriting Recognition DNN

Gayeon Kim

1st May, 2022

Contents

Chapter 1

Introduction

Nowadays, handwriting recognition plays an important role in this digitalised world. Our group named SWAT (team 31) has made a contribution to make a handwritten recogniser using python. Our group consists of Jimmy Wong and Gayeon Kim.

Jimmy Wong
jwon345
jwon@aucklanduni.ac.nz
Student of university of Auckland
Specialisation in electrical engineering

Gayeon Kim
gkim902
gkim902@aucklanduni.ac.nz
Student of university of Auckland
Specialisation in computer system Engineering

Chapter 2

Planning

2.1 Literature review

There has been a lot of work done on handwritten recognition systems. Mahmoud M. Abu Ghosh has used the three most famous NN approaches which are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). [5] In this paper, Mahmoud compared and evaluated many factors such as accuracy and performance among these three NN approaches. The result shows that DNN was the most accurate model with the accuracy rate of 98.08%.

Also, based on the experiment done by Akkireddy Challa, the research shows that CNN performance has the most accurate recognition rate among other methods. SVM performance showed 39%, ANN performance showed 37% and CNN performance 71%. DNN, which is a multiple layer of CNN, shows better accuracy than the others. [4]

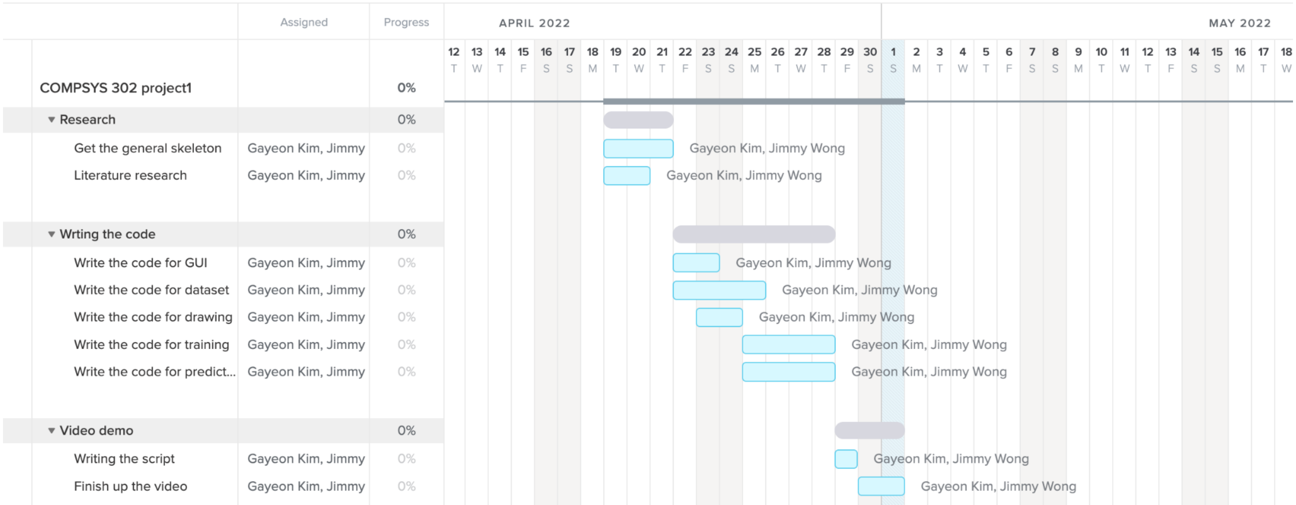
From the experiment done by Rahul Chauhan, the NN algorithms such as DNN, CNN, and RNN are implemented for the classification of handwritten digits. The algorithms are implemented on various deep learning frameworks and the performance is evaluated in terms of accuracy of the models. The best accuracy is of CNN 99.6% model and the error rate of algorithms ranges from 0.2–3%. [7]

2.2 Setup for the project

The implementation has been carried out by Python 3.8 version and pyqt which is python GUI (graphic user interface) widgets toolkit is used. We have used the torch and sklearn which are machine learning libraries, Numpy which is a fundamental package for scientific computing and some image manipulation tools such as skimage and PIL. We have chosen torch over TensorFlow that has several options from which the user may choose since torch has an object-oriented approach.

Our group used github which is a source code management to share the work and work as a group. For the Project environment setting, we have used conda which is an open- source package management system and environment management system. We have also used Pip as a package installer for python

2.3 Gantt chart



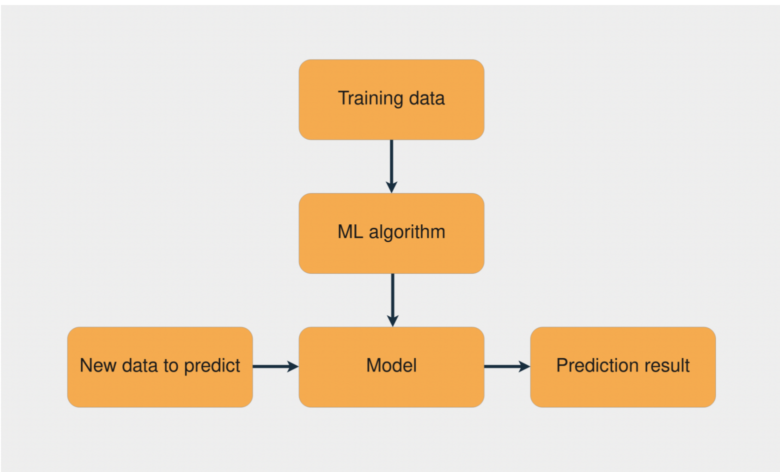
Chapter 3

Design software architecture

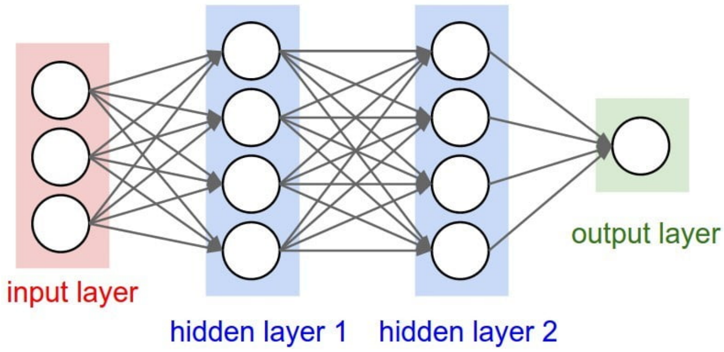
3.1 Purpose of the system

3.1.1 Background

Machine learning is a technique in which you train the system to solve a problem instead of explicitly programming the rules. The data should be gathered and a model should be trained to solve the problem using machine learning. It is common to have a dataset with inputs and known outputs. The task is to use this dataset to train a model that predicts the correct outputs based on the inputs. The image below presents the workflow to train a model using supervised learning [1]



The learning portion of creating models spawned the development of artificial neural networks. ANNs utilise the hidden layer as a place to store and evaluate how significant one of the inputs is to the output. The hidden layer stores information regarding the input's importance, and it also makes associations between the importance of combinations of inputs. Deep neural nets, then, capitalise on the ANN component. So, the deep net has multiple hidden layers. 'Deep' refers to a model's layers being multiple layers deep. [2]



3.1.2 Reason to develop our own recognizer

Today, OCR technology provides higher than 99% accuracy with typed characters in high-quality images. [3] However, a user's handwriting style varies from time to time and is uneven. In addition, strokes have a lot of variation and ambiguity depending on the diversity in human writing types. These factors cause less accurate handwriting recognition. To cover and challenge this problem, we decided to develop our own handwriting recognition system, which stores thousands of different handwriting images, trains itself and predicts the user's handwriting based on the trained model.

3.1.3 Benefit to society

Since the world is becoming digitalized, it will benefit society in various ways. For example, this handwritten recognition system could be used for reading postal addresses, bank check amounts and forms in the real world. Also, OCR (optical character recognition) plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods.

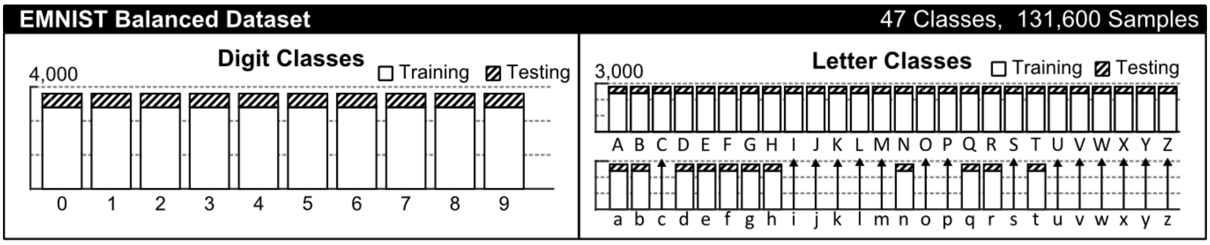
3.1.4 The goal of the project

The goal of the project is to develop a recogniser which takes the user's handwriting or a selected image as an input, trains the model then tests the input by the trained model. As an output, it gives out the predicted results and a percentage.

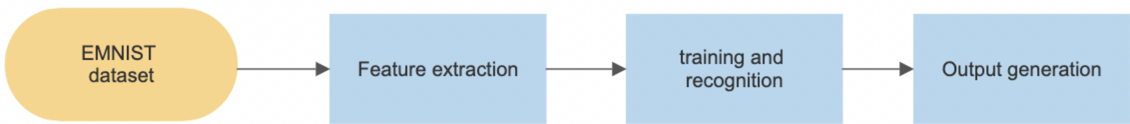
3.2 Database and methods/model used

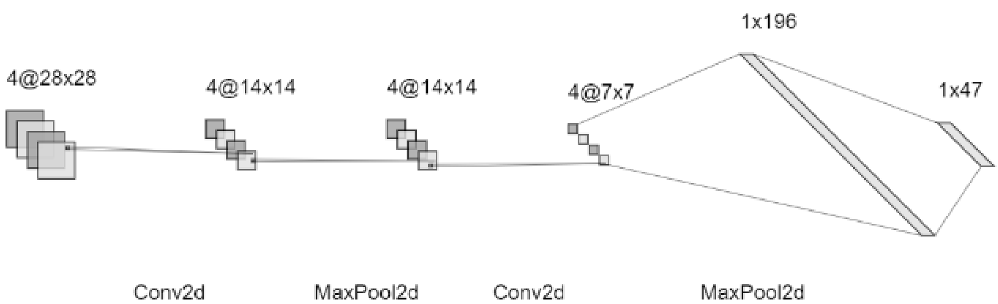
3.2.1 Name of the database and reference

The model we used in this project is the EMNIST dataset. Handwritten Recognition from the MNIST dataset is well known among scientists by utilising different classifiers for various parameters. [8] It is an existing model. Its simplicity and ease of use are what make this dataset so widely used and deeply understood and therefore, we used this dataset. EMNIST Balanced: 131,600 characters. 47 balanced classes.



3.2.2 System diagram for the software





3.2.3 Explanation of each component and the benefit

Firstly, the EMNIST dataset is downloaded and the feature extraction is processed using the DNN method. It extracts features from the images which classifies the objects in that image. It starts by extracting low dimensional features from the image, and then some high dimensional features like the shapes. [6] The model is then trained for predicting the output. Deeper understanding of the neural network is explained below.

The network is built with 2 Convolutional layers and 1 fully connected layer. It was developed with a reference to the PulkitS model with a few modifications to fit the dataset. [6] We chose this model since cuda couldn't be used, its processing time was limited as cuda.

The image is one channel as it's just black and white. The first convolutional layer splits the image into 4 features. Then the Maxpool takes the maximum value of a 2x2 square pixel, reducing the size of the features and therefore the processing time. Another convolution is processed and its output features are then reduced again by a 2x2 max. Leaving 196 features to analyse in the fully connected layers. The fully connected layer then with weighting properties determines the link between features and output representing the predicted character. With the increase of epoch, accuracy could be improved through repetitive training increasing the results.

The structure of the neural network is contained within the training.py file within the /Scripts folder. In the file a class named Net contains the structure of the neural network that can be modified to another structure that could prove to be more accurate. Addicational layers can be added or removed in both the convolutional layers or the fully connected layers. The first 2 parameters of the Conv2d represent the input and output features of the layers respectively. They can be modified to fit a different number input and output features.

Since this architecture has less feature layers and less fully connected layers compared to others, it allows the processing time taken to be shorter.

Chapter 4

Results

4.1 Details of the tool

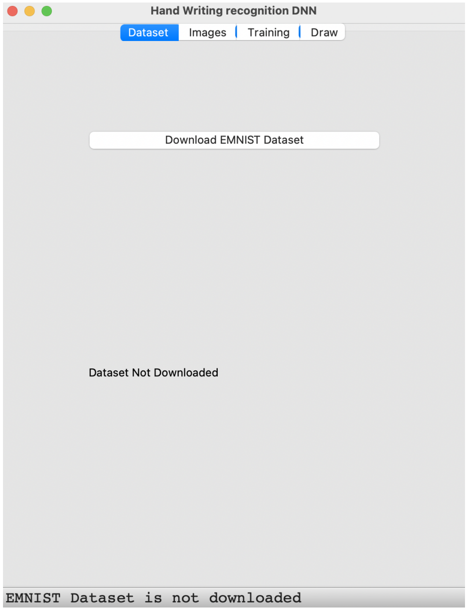


Fig. 1

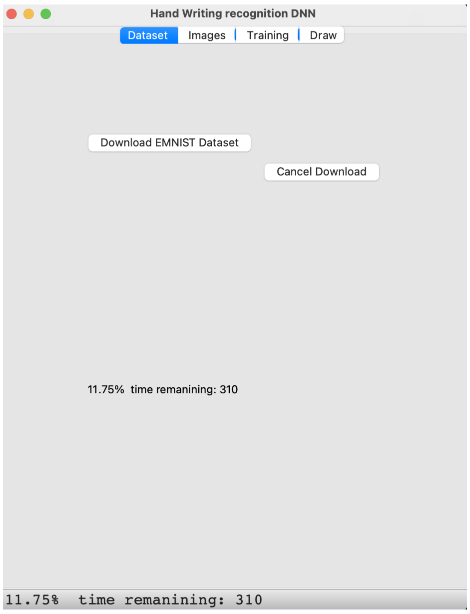
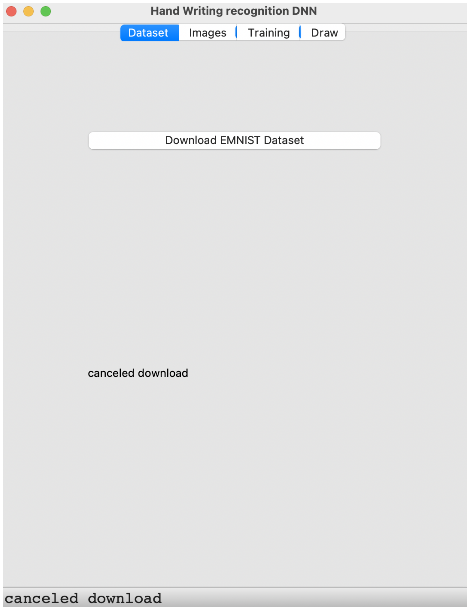


Fig. 2

The widget with a title of “Handwritten recognition DNN” has four different tabs, dataset tab, images tab, training tab and draw tab. (Fig.1)
In the dataset tab, the user can click the “Download EMNIST Dataset" button to download the dataset. When the download begins, 2 threads are started. One to update the percentage and one to download the models. The progression is shown as a percentage with a time remaining information on the widget. This is calculated by the file size and the rate of the download speed. When the download is started,



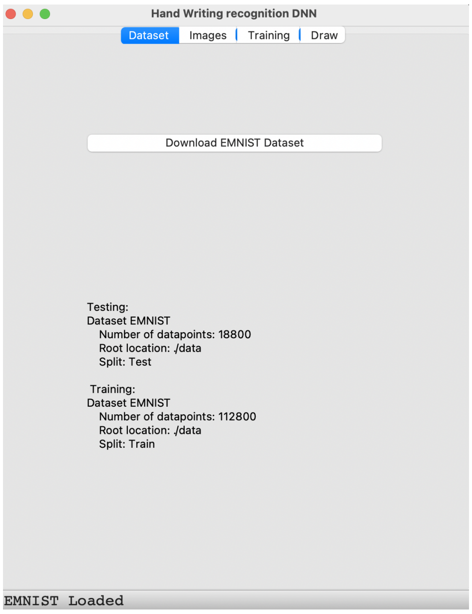
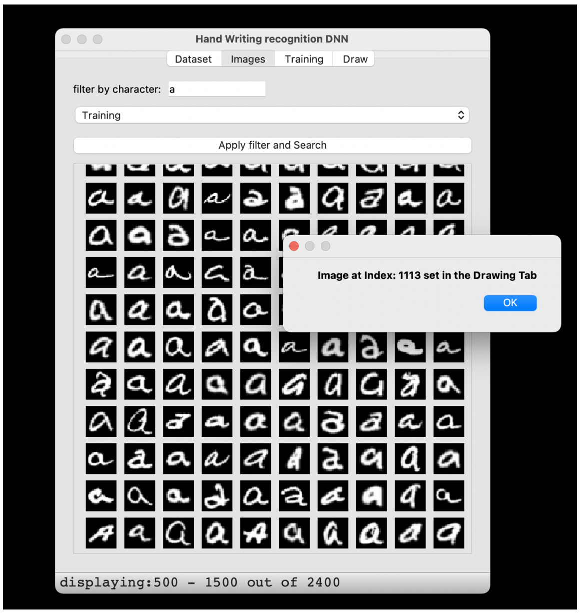


Fig. 3

Fig. 4

To cancel the download of the EMNIST dataset, the user can push the cancel download button which will terminate the downloaded thread. The label shows “cancelled download” (Fig.3)
When the download is completed, the information of training and testing appears, such as number of data points and root location. (Fig.4) Also, the label shows “EMNIST dataset Loaded” so that the user can check the status of the dataset download.



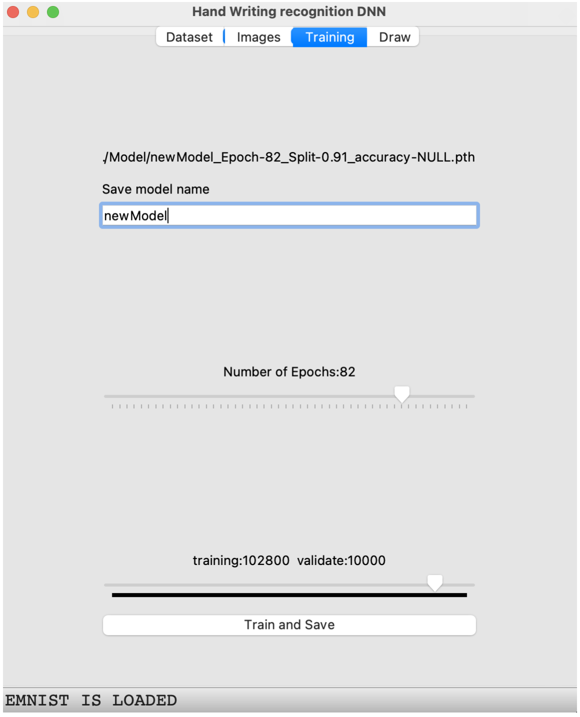


Fig. 5

Fig. 6

In the images tab, there are two options of dataset, training and testing. The user can type in the letters (lowercase and uppercase) and numbers to see the filtered images. The user can also input nothing to see all the dataset images without being filtered. In the label, the range of the index of the images could be seen. For example, “displaying 500- 1500 out of 2400”. If the image is clicked, the index of the image could be seen. (Fig. 5). Also when the image is clicked, it is copied to the draw tab so that the image of the dataset could also be predicted. In the training tab, the user can type in the file name where the model is saved. The user gets to choose the number of Epochs then push the “train and save button” to train the dataset and create a model. The training dataset is splitted into characters to be trained and validated. Validation is used for training. The relevant information of the trained model is saved in the file’s name such as the number of Epochs and the accuracy. (Fig. 6)

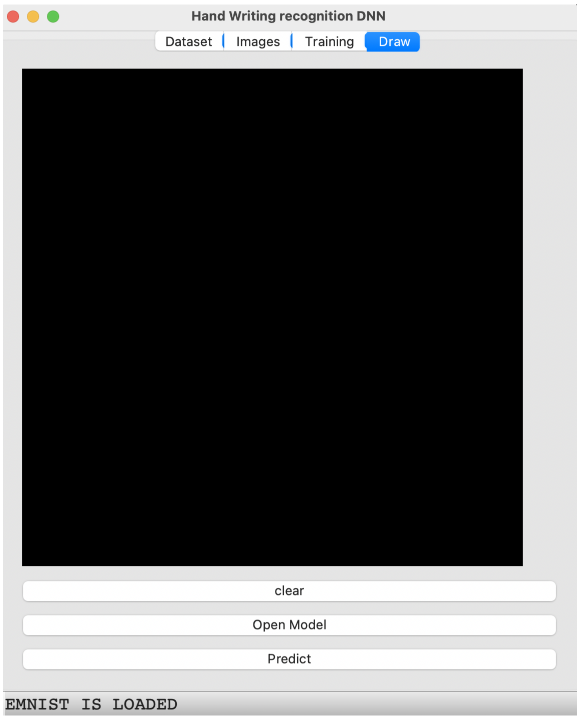


Fig. 7

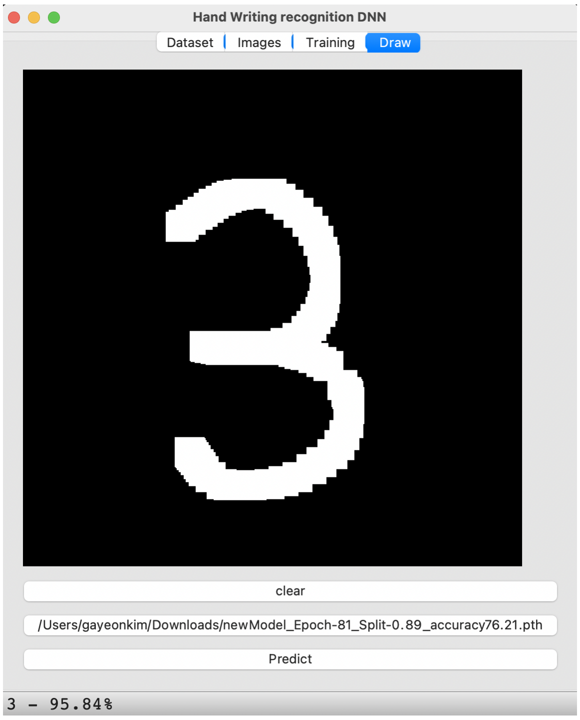


Fig. 8

The draw tab includes a canvas which allows the user to draw a character that wants to be predicted. The user can open the file of the model that is previously saved by the training process. Then the predict button predicts the character on canvas based on the trained model. The user can also clear the canvas by clicking the clear button and redraw another character after. Also the user could get the image from the images tab and predict it.

4.2 Statistical analysis, recognition results and discussion

With our neural network system, the highest accuracy we could get using the epochs number of 81 was 76.2%. This percentage shows that our model is well trained with reasonably high accuracy. Using the model trained by this epochs number, with the proper drawing input in the canvas and using the images chosen in the images tab, our system gives out the correct predicted result. Since the segmentation process is not implemented in our system, it cannot recognise multiple characters at once. This is the only shortage in our recognising system, we would like to work further later on.

Chapter 5
Conclusion

5.1 Conclusion

In this project, we developed a handwriting recognition system using python. From our widget, the user could download the EMNIST dataset, train the model choosing the file name and the number of epochs. Then the user could make a prediction on user input such as an image or drawing. For the classification, we used the DNN method with 2 convolutional layers, 1 fully connected. For the performance, the highest accuracy we could get for training the model was 76.2% for the number of epochs of 81. Moreover, the accuracy is a bit lower than other techniques but it compromises with a shorter processing time. If the increased number of convolutional layers was used, we could get higher accuracy results.

5.2 Future works

There are some limitations in our project and improvements to be implemented.
The method to perform the segmentation could be implemented so that multiple characters can be predicted separately. The accuracy of the prediction could also be improved by increasing the number of epochs or increasing the number of convolutional layers. Furthermore, to improve the accuracy, the number of convolutional layers and number of filters in each convolutional layer could be increased.

Chapter 6
References

[1] Déborah Mesquita Python AI: How to Build a Neural Network & Make Predictions
[2] bmc blogs. What's a Deep Neural Network? Deep Nets Explained
[3] Cem Dilmegani. High tech industry analyst AIMultiple.Handwriting Recognition in 2022: In-depth Guide. 2022
[4] Akkireddy Challa. Dept. Computer Science & Engineering Blekinge Institute Of Technology Se--371 79 Karlskrona, Sweden. 2019
[5] Mahmoud M. Abu Ghosh. A Comparative Study on Handwriting Digit Recognition Using Neural Networks.2017
[6] PulkitS. Build an image classification Model using convolutional neural networks.2019
[7] Rahul Chauhan. Recognition of Handwritten Digits Using DNN, CNN, and RNN: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I
[8] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017) EMNIST: an extension of MNIST to handwritten letters. https://arxiv.org/abs/1702.05373

Link to the video demo
https://webdropoff.auckland.ac.nz/cgi-bin/pickup/072c663fb73b71d6d0ccaf63b476cfcb/179724