MITA CAPSTONE PROJECT

*Handwritten Character Recognition*

**Introduction:**

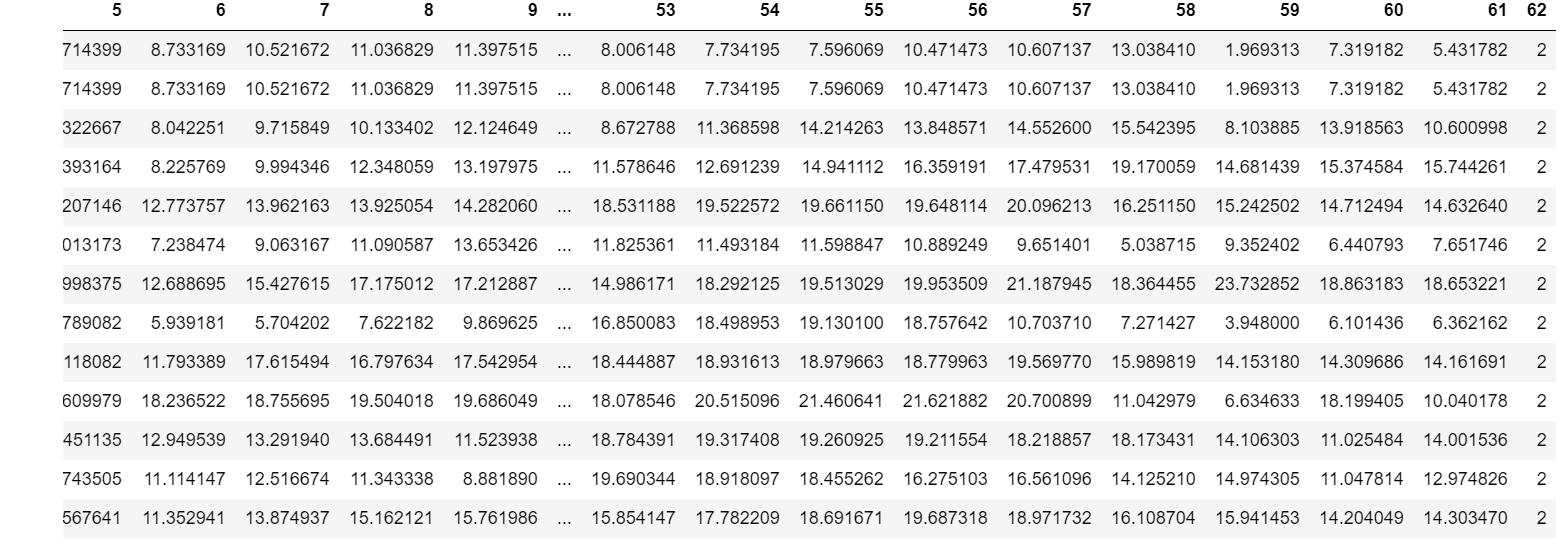
The existing handwriting character recognition process uses the MNIST data set and takes into consideration the pixel data for the character and then uses this to recognize the character using the convolutional neural network. The neural network model can be built and can be saved as .h5 file and then converted to a protobuf(.pb) file to be used in android studio and then can be exported to Android studio and can be used to recognize the character by feeding the pixels data generated using Android studio.

**Building the Dataset:**

In this project, not the pixel data but the time at which each of the point of the character is drawn, and the x and y coordinates are taken into consideration. And to make the values of 3 parameters as our predictors, b-spline(Cubic spline) has been used. In order to fit regression splines in python, dmatrix module from the patsy library has been used to fit regression splines in python. The regression splines can be fit by constructing an appropriate matrix of basis functions. The bs() function generates the entire matrix of basis functions for splines with the specified set of knots. To capture non-linearity in regression models, the predictors need to be transformed by appling family of transformations, and these transformations that can fit together are called basis function. So, here xi is fittedto ti and yi is fittedto ti to using a regression spline. And the assumption here that is made is that for this the degree of freedom is 30. So, in total there will be 31 transformed variables each for xi and yi. So in this way the x’s and y’s will be mapped on the basis of t.

**The Algorithm:**

Using android studio, for each character, these above-mentioned features can be captured, transformed and then the dataset was built using this and finally this dataset was the basis for the ANN Model i.e. Artificial Neural Network Model was built. A snapshot of the formed dataset is below:



After much observation and running the model, it was found that the neural network with **7 hidden layers** gives the maximum **accuracy (83%) and a batch size of 5 with 200 epochs**. This can further be improved by capturing more and more data and adding more rows in the dataset. Also, for now the numbers ‘2’ and ‘3’ are only taken into consideration and **a total of 53 samples** of these were taken processed and used for the recognition. Also, since this is a multiclass label classification problem, softmax function is used in the output layer of the neural network. Also, tanh function in the hidden layers was found out to be giving the optimal result.



The **advantage** of this model can be that it can turn out to be more accurate as compared to the existing models because in this case each points’ x’s and y’s are mapped according to the order (the time of their occurrence in the device screen where the character is being drawn) they appear, or they get displayed. So, this can define the shapes better. So, even if a person is drawing the shapes starting in opposite direction or in distorted manner, in that case this model is supposed to work better.

But the **disadvantages** could be that it’s more time consuming to build this model, like each and every points with the time needs to be transformed before feeding to the model, also when android studio will be used to predict the character then the data captured by android studio, all these have to be again transformed, so in android studio this code for the transformation needs to be written as it doesn’t have any predefined function/library that can be used like the bs() and dmatrix is there in Python/the code needs to be written in python and can be used in Android using some python Android Scripting Layer.

**Future Scope**: Of course, this can be extended to be used with every other numbers, then alphabets and special characters as well. Also, for characters which don’t contain all connected lines or while drawing the character, the finger needs to be taken off the screen (like small letter ‘i’ and many other characters like this), some discontinuous functions could be used and in the same process as described above, it can be noted if it gives good accuracy.