TASK 2

Task 2 has been completed in 3 parts:

- **PART 1**
- **PART 2**
- **PART 3**
  - **#1**
  - **#2**

Part 1: The data provided for the task comprised of images of digits and letters. The data was extracted from the file appended to data structures. The labels corresponding to the was fit into images were extracted from the folder names by programming. The images were pre-processed in accordance with the MNIST dataset to standardize it before usage. The images were resized to (28,28). The image colours were

inverted so that only the digit part could be obtained and unrequired data i.e. black pixels were discarded as they were not deciphering any useful information for the model to learn. The image was padded with black pixels if required for sizing it to (28,28). Some pre-processing steps have been obtained from Medium blogs. They have been explicitly mentioned in the notebook. After pre-processing the data was shuffled and appropriately reshaped before sending to the model.

Then, CNN model was developed with Keras Tuner for hyper-parameter tuning.

The model architecture is as follows:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 24, 24, 112)       2912
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 20, 20, 48)        134448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 10, 10, 48)        0
_________________________________________________________________
dropout (Dropout)            (None, 10, 10, 48)        0
_________________________________________________________________
flatten (Flatten)            (None, 4800)              0
_________________________________________________________________
dense (Dense)                (None, 112)               537712
_________________________________________________________________
dense_1 (Dense)              (None, 63)                7119
=================================================================
Total params: 682,191
Trainable params: 682,191
Non-trainable params: 0
_________________________________________________________________
```

Random search method was used for hyper-parameter tuning which searched for the best fitting hyperparameters by running for 3 epochs each for 3 trials. When the best set of parameters were obtained, it was trained for 17 more epochs.

The model was then saved.

Part 2: For the second part of the task, the same pre-processing steps as part-2 were performed. The labelling of data was the only variant, which has been represented in the notebook. After pre-processing the data was shuffled before sending to the model. A CNN model was created as above using Keras Tuner with Random Search. The optimal parameters were used to train the data for a total of 15 epochs. The model, so formed was saved.

Then, the pre-trained model was used to train on MNIST dataset. The model gave a testing accuracy of **98.4%.** (Refer notebook)

A randomly initialized CNN was trained on the dataset and it rendered a testing accuracy of **98.6%,** which is quite comparable. However, the convergence time was more for

randomly initialized model as compared to the pre-trained model.

Part 3:

Qualitative analysis of the dataset-

For this part of the task, the provided training data did not have any labels. Nor could they be produced by programming or using folder names because they were in no specific order. The images were randomly present across the folders, unlike other datasets. So, a model was trained on MNIST training data. Then, this model was used to generate predictions for the given dataset, which were used as labels for the training set.

This data also consisted of more images which appeared as difficult test cases for the model as many of the digits also resembled some letters. Next, after the data was labelled, it was trained on a CNN by using Keras Tuner for hyperparameter tuning. Then, this model was tested on the MNIST test data. The accuracy reported was: **98.29%**

The next challenge was to use pre-trained network in Part 1 to find test accuracy on MNIST dataset. I created a dictionary to map the labels as the dataset in Part 1 network was trained on letters and numbers both. Shockingly, the test accuracy on MNIST turned out to be 0 as all of the digits were wrongly attributed. Zeros were recognized as 'O,' '2' was recognized as 'z'. 4 as 'h', etc.

CONCLUSION

- **In PART 2 the pre-trained network performed better (or comparable) than randomly initialized network.**
- **Whereas in PART 3, the randomly initialized network performed better than pre-trained network.**

Reasons to explain this:-

1. **The dataset contains a lot of digits which look like numbers.**
2. **Network used here i.e. Part 1 network, consisted of both digits and letters, which have been misclassified.**

3. **We generated predictions and used them labels, that can cause discrepancy as some would have been mis-classified.**