AML Mini Project Report

The mini project is about object classification. Each image contains one object. Likewise, there are 100 different types of images in the dataset.

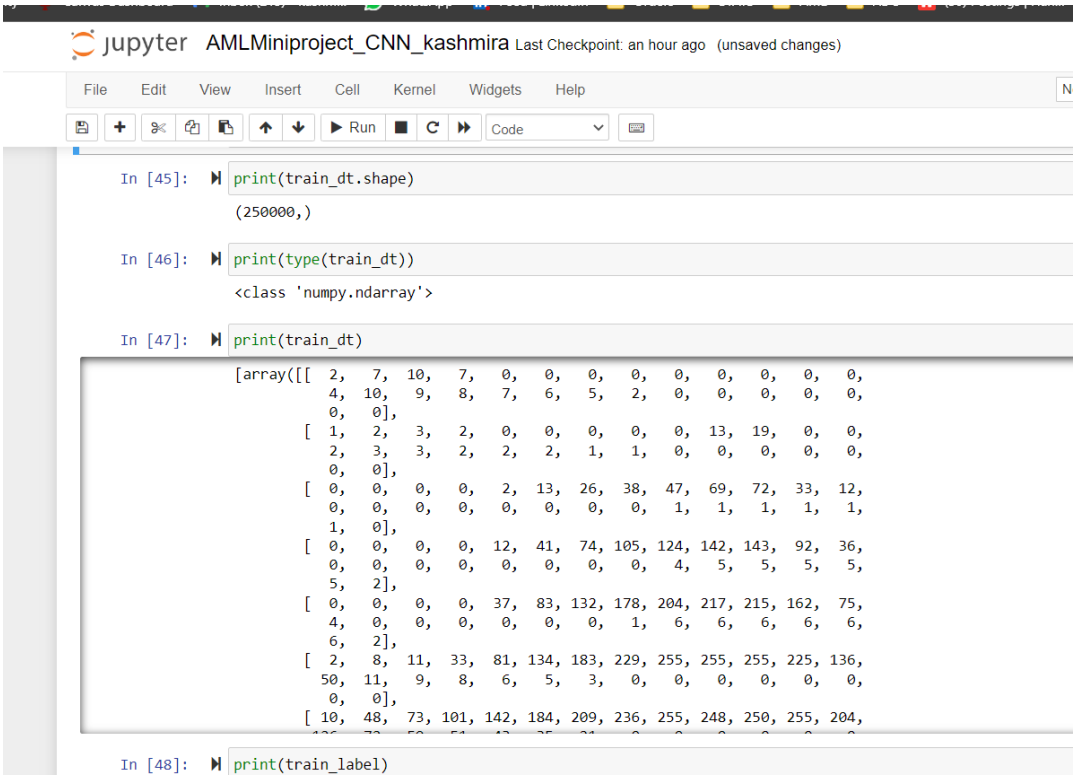
About the training dataset:
The training dataset contains two columns namely – data i.e the pixels and target (0-99).

```
In [58]: print(train_df)
```

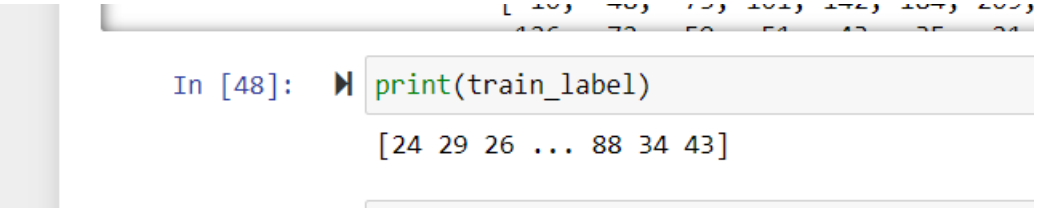
```
                                                     data  target
0       [[46, 132, 11, 1, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0...       0
1       [[0, 0, 3, 0, 116, 78, 0, 1, 1, 0, 0, 0, 0, 0,...       0
2       [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4,...       0
3       [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...       0
4       [[0, 0, 0, 69, 165, 202, 199, 214, 204, 170, 1...       0
...                                                   ...     ...
499995  [[0, 0, 0, 0, 1, 4, 7, 8, 6, 4, 5, 1, 0, 0, 0,...      99
499996  [[0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0,...      99
499997  [[0, 0, 0, 0, 2, 0, 25, 26, 0, 1, 0, 0, 0, 0, ...      99
499998  [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...      99
499999  [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,...      99

[500000 rows x 2 columns]
```

After sampling the 50% data we get the training dataset – data attribute(X) as follows:



About the testing dataset:
The test dataset contains only the values of the images i.e pixels, without any labels.



Aim: Our aim is to find the labels in the test dataset by improving the accuracy of the model (KNN). This can be achieved by applying various algorithms.

Formulation of the problem:
As the deep learning techniques are highly effective when working with images, I have used CNN model. As it takes inputs of the images and extracts the features of it and classifies based on their attributes

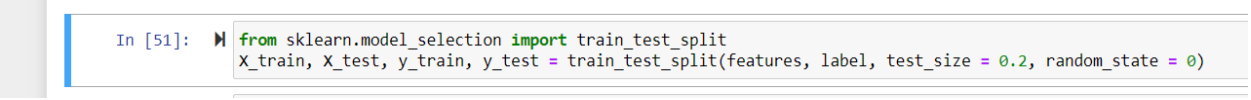
Convolutional Neutral Network:

The CNN model consists of 4 stages, namely - Input layer, Convolution layer + Activation function, Pooling layer and Fully Connected Layer.

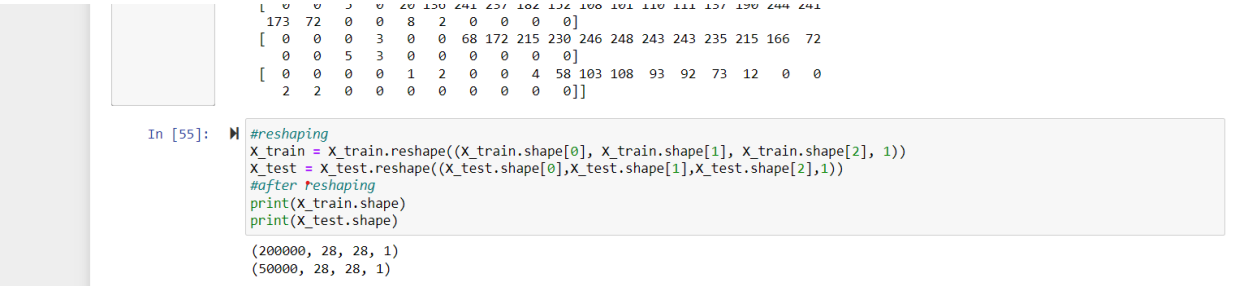
Illustration of preprocessing and model design:

Input layer:
We have a dataset containing 500000 images. Sampling will be required as the dataset is too large. Hence, we will be using 50% of the training set, around 100000 images. After sampling is performed, we will further reshape our dataset into 28, 28, 1 format for uniformity and in turn normalize it further.

Applying the train and test data split on 20% of test data:



Reshaping X_train and X_Test:



Normalizing the pixels:

```
In [56]: #normalizing the pixels
         X_train=X_train/255
         X_test=X_test/255
```

Convolutional layer:

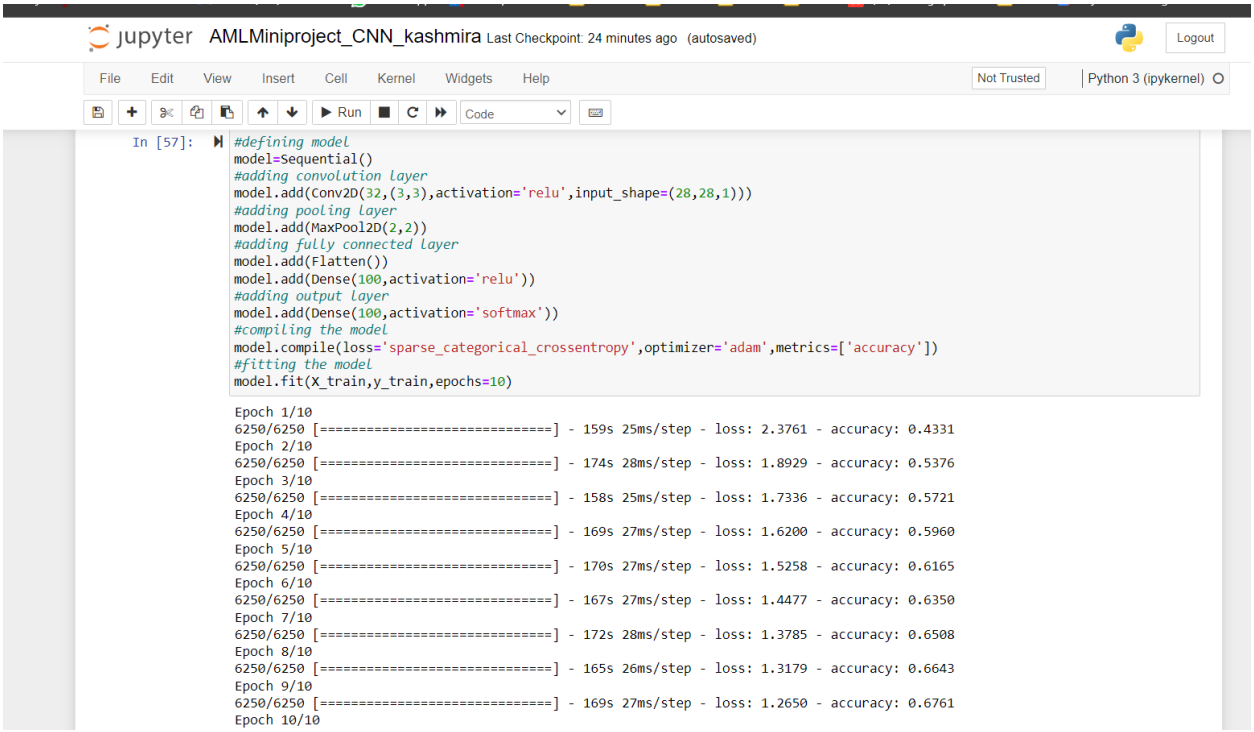
In this layer we will apply filter on our image input. I have applied filter of size 3X3. So a filter a size 3X3 will be applied and multiplied to each pixel in the image and will be shifted by one column in every step. Likewise, a filter will be applied to an image several times. After the filter has completed applying over an entire image, our feature map is ready.

Pooling layer:

This layer is applied after the previous layer is completed. The main goal of this layer is to reduce the dimensions of the feature map. To help in preserving the important attributes of the image and will also reduce the processing time. There are two types of techniques in pooling – Max and Average Pooling. I have used the max pooling technique with 2X2 strides, which will directly consider the max values of each stride, instead of taking average of the entire stride.

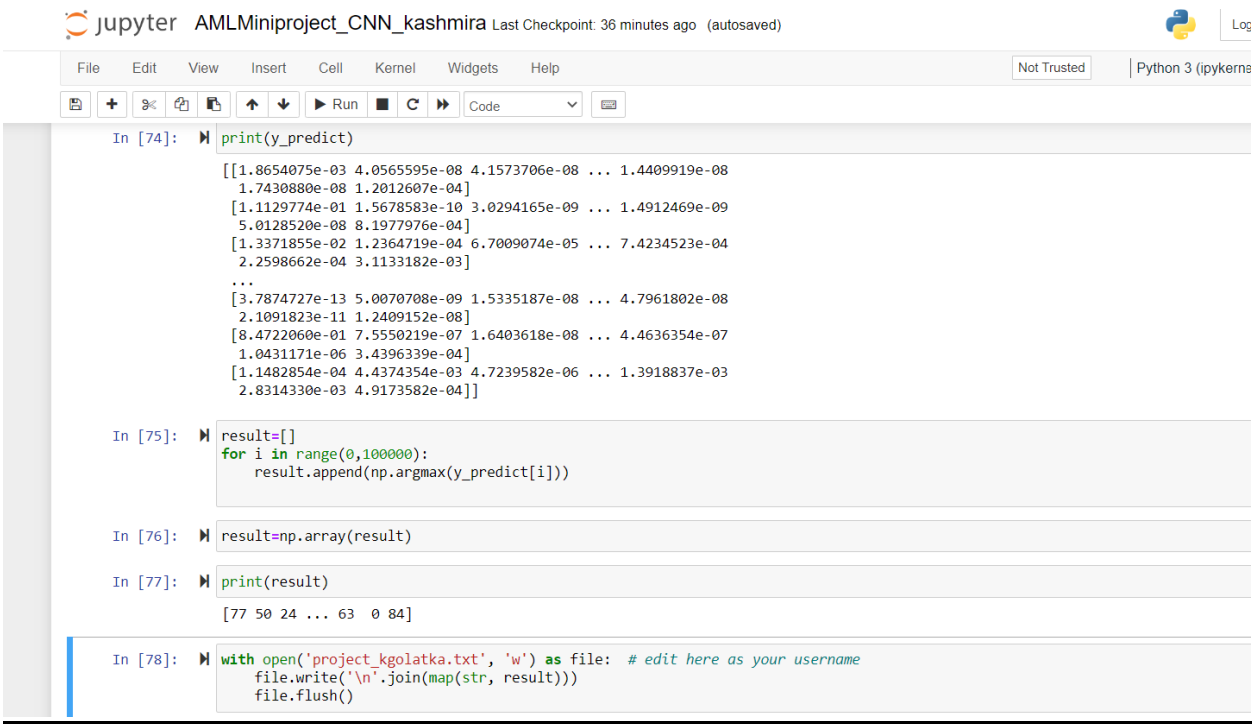
Fully Connected Layer:

This last layer merges the results of all the previous layers and uses them to classify the labels of the input image in the output layer.



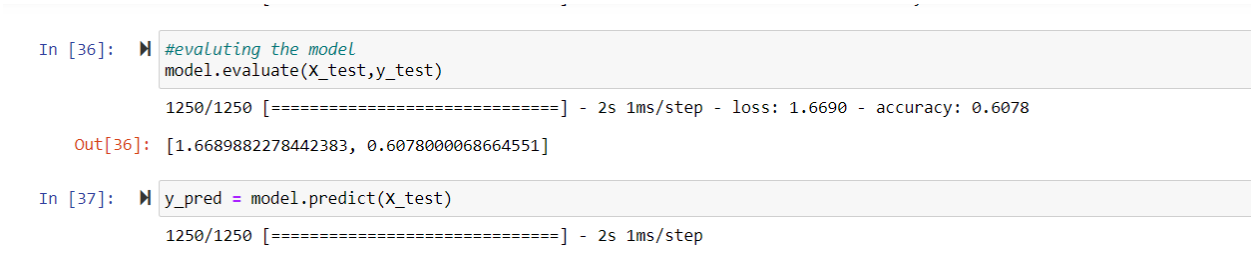
Applying the model on test the data:

As we can see the predicted output is in exponential form, we need to convert it to a numerical form by using np.argmax from the possible 100 outcomes. After the result is acquired in the numerical format, append the results in the text file



Results:

After applying the CNN model on 50% sampled training data, the accuracy is 60%.



Ways to improve accuracy:

The accuracy can be improved by training the data on a larger training set i.e. more than 50% of the training data.

It can also be improved by applying other different machine learning models like SVM and Random Forest.