Importing Libraries

os library is used to read the files and folders from a specific path.
numpy, csv, Image and pandas are used to read and process the data.
Matplotlib is used to visualize the given data by using various plots present in the library.

Structuring Train Data

Creating Labels

path = the path where you stored the train data. This will help to process all the folders in the train data
All the names of the folders are stored in the **filenames** which is a list. The last word after _ is split and taken into **labels** list which is the name of the images in the folder.

Creating .csv file

Each file name is retrieved from the filenames list and all the image file names are stored in the list. Using this file path image is converted into pixel data and this data is written into the csv along with the name of the letter that will form the label.
Note in the function with open **you should replace the file path given with the path of the folder where you want to store the csv.**

Reading .csv file

All the data in the csv file is loaded into the data frame

Data verification from .csv file

The first five entries are printed to check whether the data is loaded correctly of not.

Dividing Train data into labels and attributes

Here the train data is divided into two data frames, one data frame containing all the attributes of the data and the other one containing all the labels.

label conversion from strings to int

Since all the labels that were created are in string format we assigned each string a unique integer. All the labels present in the Y are replaced by these integers.

Training the algorithm

We used RandomForestClassifier from sklearn library to train the model with estimators=100 to 1000. We have observed that the model accuracy hasn't changed very much from 100 to 1000 hence it is better to use 100 instead of 1000 as it could save a lot of time.

Structuring the Test Data

Creating Labels

Please replace the path_test with the folder name of the test data. (**you can shift+right-click and select option copy as path and paste it here**).
The processing of the files of test data is the same one as that of the train data.

Creating .csv file

Creating the csv and and appending the labels to the csv file is same as that of the train data. Please replace the path in the **with open function with the path of the location where you want to store the test csv file.**

Reading .csv file

The data is loaded into the data frame from the csv. **Please update the path with the file location of the test csv.**

Data verification from .csv file

To verify whether the data is correctly read into the data frame, the first five entries of the data frame are printed.

Dividing Test data into labels and attributes

Similar to the train data the test data is also split into two data frames one containing all the attributes and the other containing all the labels.

Label conversion from strings to int

All the labels in the test data label data frame are replaced with integers in the same way as that of the training data.

Predicting the Test Data

The test data is run through the predict function of the model trained using the train data and the results were stored in an array.

Calculating Accuracy

The results of the predictions are then compared with the actual labels obtained from the test data and the accuracy is determined.

Plot between n_estimators and respective accuracies

A plot is drawn between the accuracies and the estimators used. By doing this we can determine what estimator is producing the best accuracy and train the model using that particular estimator.