**Image Classification**

Use the data in the compressed file named, Data.7z (this is a compressed file, you will need to uncompress it with a utility such as 7 zip). Once you extract the file, you will get access to 1 folder, **train**. This folder contains, subfolders with images of handwritten devnagri characters and digits. Your task is to build a CNN based image classifier, which will be able to distinguish between different handwritten devnagri characters. The first thing that you need to do is, read all the images in the train folder and build a model. To read images make use of opencv3. If you haven’t installed opencv3, you can follow instructions here: <https://gist.github.com/Gunnvant/bed6ed28de512abfa1fc6d730a2ee406>

**Q1.**  Before building the image classifier, you will need do some data munging. You will need to understand your data first. Keeping this context in mind, answer the questions raised below (12):

1. In the train folder, images of how many different classes are present? (Hint: use os module) (2) (Text input)

**46**

1. How many total images are present in the train folder? (3) (Text output) (Hint: use os module)

**78200**

1. Read all the images (in the train folder) and their associated labels in to two separate lists. While reading the images make sure that, images are in grayscale colorspace (use opencv to achieve this). Do all the images have a dimension of 28 by 28 pixels. (2) (True/False) (Hint: use os and opencv)

**False**

1. If you flatten all the images (in the train folder) and create a dataframe, with each row of the dataframe representing a flattened image, then, how many columns this dataframe will have? (2) (Text input)

**1024**

1. How many images are there per class in train data. (Each folder inside **train** contains images corresponding to specific devnagri alphabets or digits) (3) (Text input)

**1700**

**Q2.** You now will need to build an image classifier. Use a CNN architecture to build an image classifier. Use the train data to get a model. Then use the test\_X.csv to make predictions. This file contains, flattened images of devnagri characters. The images in test\_X.csv are of same dimensions as images in train folder, hence when you will reshape these images, you will get images with same dimensions as the ones you used to train the model. You need to make predictions on the test\_X.csv. Make sure that your predictions are in the same format as given in the file submission\_format.xlsx. Essentially you need to predict the class of each image in test data. Make sure the labels are spelt the same way as they are in the train data or as suggested by submission\_format.xlsx (names of sub-folders in train data).

1. Submit your python script here (Formats accepted, .ipynb, .py) (File upload) (10)
2. Submit your predictions (name your file as pred.csv, make sure it is in the correct format, as suggested above) (File upload) (8)