

# Description of Major Project

Presented by B18

Our team project was **Human Image Classification** where we had to classify images as Indian human faces or other country(foreign) and build a model.

## Tools and strategy(phase 1)

Machine learning, Big data and data analysis projects require data to work with. There are many ways in which we can acquire the data required to build the models.

Download manually the images from search engines like Google, or a dataset from Kaggle.com, or use web scrapers like parsehub, etc.

**And our approach was to Download images manually.**(since it had the most accuracy of collecting appropriate dataset)

## Dataset Collection

We had collected images of people from different regions of India, as India is a vast country with different cultures, etc. Also the images of foreign people from different regions of the world manually.

## Dataset cleaning and building the model

Our target is to remove unwanted pictures(not human pictures), which can be done by cleansing the dataset using a python code. For this we used OpenCV, which contains haarcascades package from which we used face cascade and eye cascade, which are used to detect the face and eyes.

We have written the code which basically detects whether the image has a face and 2 eyes, if yes crop it and save it in the new folder called **Cropped**, else discard it. So this code discards unwanted images and we get a dataset which can be used to train our model.

Now we have to build a model using the dataset which is already split into indian and non-indian(foreign) folders. We also have to preprocess the dataset by resizing and flatten the images(from 3D to 1D array). Then use those flattened preprocessed data as input for our model. So all we have to do is take those pictures as input X, Y. Using

train\_test\_split data we split the data into train and test data. So 80% data to train our model, and 20% to test our model.

Now for our model the best algorithm was SVM(kernel used was 'poly') and gave us the most accuracy. So now we had to feed the model with dataset and then use it to test our outputs based on the inputs test dataset. So we got an accuracy of 75-80%.

Now we also saved the file in pickle, so that we don't have to run it everytime we open it.

Now when we want to test our model, all we have to do is give the input:

- 1) An **URL** of the image you are giving as input.(mostly jpg format)
- 2) The datapath of the image you have stored in your pc.(ex:C://users//XYZ//Downloads//image1.jpg)

Now the output will be the image that we chose presented, and also stating whether the person is an Indian or foreign. And also store the image into the respective folder(if indian face store it in indian folder containing indian faces, and vice versa).

**This model which we created has an accuracy of 75-80%.**