FACIAL FEATURES DETECTION

-Bhavya Kohli (20d070021)
-Amruta Parulekar (20d070009)
-Tanmay Joshi (200070027)
-Sameep Chattopadhyay (20d070067)

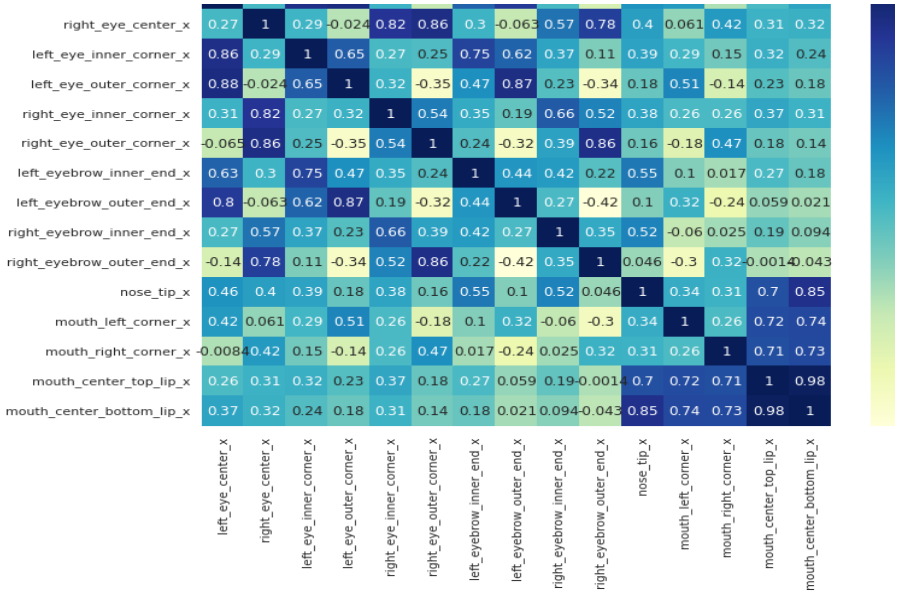
In this project, we aim to demonstrate the usage of the *fastai* library as a faster and more efficient computational tool for feature extraction in order to perform facial recognition and detect whether any two given images belong to the same person or not.

For facial recognition, we are implementing a two-stage algorithm.

Firstly, we performed facial detection on the given image and cropped the face portion in order to extract certain facial features. After that, we extracted 15 points from each face, including the locations of the centers and corners of the eyes, nose, and mouth using fastai. Then these 15 points were used to compare the facial features of any two images and return a single value, 0 or 1, which predicted whether the images belonged to the same person or not. For our project, we have used convolutional neural networks - the cnn_learner method.

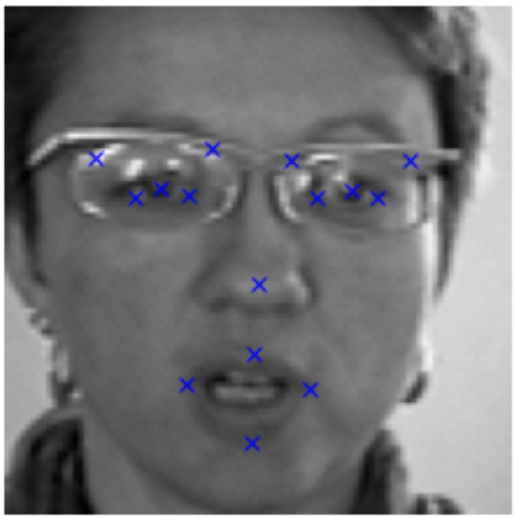
Step 1: Exploratory Data Analysis

The data were thoroughly analyzed and various attributes of the variables were printed and heatmaps of the x coordinates and y coordinates of different features were made along with other plots to see the general location of important points like the centers of the eyes and the tip of the nose.



Step 2: Feature extraction and image augmentation

Fifteen facial features such as nose tip, centers & corners of eyes, and mouth are detected from the images from the given dataset as shown below-



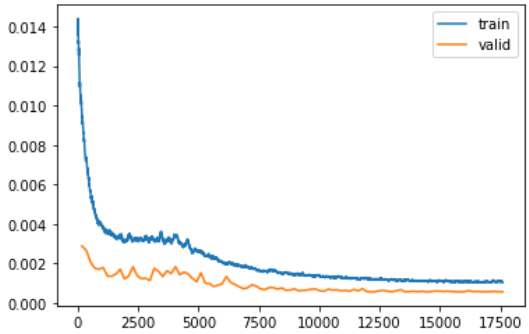
The images were also augmented (Shifted and/or rotated) in order to make the model more generalized and capable of identifying faces that are not straight.

Step 3: Learner model (Using cnn_learner method in fastai)

A loss function was defined for the learner model, using mean square error on the flattened image. Then we used transfer learning for our further task.

Transfer learning is a technique where you use a model trained on a very large dataset and then adapt it to your own dataset. We have used Resnet18 for our model along with the *cnn_learner* method in *fastai* which was initialized with *resnet18* and our *data loader*.

Step 4: Analysis and Prediction



Plot of Loss function of Learner with iterations

The actual points on the training images, along with the predicted points

Step 5: Feature comparison

We have used the LFW dataset of Massachusetts university for the second stage of our project. The images of the dataset were first cropped and preprocessed in order to just have the faces, and stage 1 steps were applied to obtain feature points. Then we fed 13 points to a function that calculated the distances between each possible pair of these 13 points eg- the distance between eyes, length of lips, etc.

Now, for any pair of training images of the same person, the l_2 norm of every such corresponding distance pair was calculated and then they were used to further calculate The root mean square of these norms for every image pair.

We used these RMS values to calculate a certain threshold, if the root mean square value is more than the threshold value, the image is not a match and otherwise it is a match

Conclusion

- We successfully created a model that compares two images and identifies whether they belong to the same person or not.
- There is a possibility of further improvement in the model by training the model on images of people of different ages and ethnicities
- We can use this model in making face locks, locating people in crowds, etc

Execution Sequence: Same as the sequence in which the code has been written in th ipynb file