INSTRUCTIONS

BACK-END PART:

Platform Used: Jupyter Notebook
Language Used: Python

ABSTRACT

Our back-end part comprises of four models. The first three are the respective detection models i.e wrinkle detection, dark spot detection and puffy eye detection. The fourth model is our face detection model. We are basically deploying the three models into the face detection model and as an output we are getting an image which is outlined by a rectangle box on the face region along with the percentages of the prediction made by the models.

Various steps taken to make our wrinkle, dark spot and puffy eye detection model are as follows:

STEP 1: Import the necessary packages

You first need to install the required packages and library in your anaconda environment(if not present), so that our code can get executed.

Name of the packages and library:
Tensorflow, keras, openCV, sklearn, numpy, matplotlib, imutils.

STEP 2: Initialize the number of epochs to train for, and batch size

STEP 3: Grab the list of images in our dataset directory, then initialize the list of data(i.e. images) and labels

STEP 4: Perform one-hot encoding on the labels

STEP 5: Splitting the data into training and testing dataset

STEP 6: Construct the training image generator for data augmentation

STEP 7: Load the EfficientNetB0 network

STEP 8: Specify the architecture

STEP 9: Compile the model

STEP 10: Train the model

STEP 11: Make predictions on the testing set

STEP 12: Serialize the model to disk

STEP 13: Plot the training loss and accuracy

INFERENCE:
Our models are ready to make prediction and now can be deployed to the face detection model.

Various steps taken to make our face detection model are as follows:

STEP 1: Import the necessary packages

STEP 2: Get the frontal face cascade classifier

You can get this xml file from the git hub openCV repositary

STEP 3: Load the 3 detection models from your disk

STEP 4: Make a function that gonna do the face detection and then gonna make prediction according to the respective models loaded

STEP 5: Call the “detection” function and show the output

INFERENCE:

Now our all the four models have been deployed properly and we are able to make the suitable detection on any images.