Face Detection and Recognition on Video Stream

This code combines the task of face detection with face recognition. The model is given a set of known faces and it basically compares the input face with the known faces. Both the operations here are performed frame by frame on the video. Face Recognition algorithms pick out specific, distinctive details about a person's face, such as distance between the eyes or shape of the chin, which are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. This code does the same step by step process making use of the predefined models. This code uses OpenCV combined with face_recognition and dlib. Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It also provides an API for python.

The code first reads the input video and gets the frame rate for the same and creates an output file called "output.avi" with the same resolution and fps as the input file. Video here is a clip from the movie "The Big Short". The model then reads a set of known faces to compare read faces with. Just like Task 1, here also each frame is read and first converted to greyscale to get a single channel image. Then the model detects a face in the current frame and compares it with the set of known faces to see if it recognizes any, in our case, Ryan Gosling and Steve Carell. It simultaneously draws a bounding box around the faces, writes the name beneath and also writes the frame to the output file "output.avi". The progress can be tracked with the help of print statement just before section 2.6 which shows written frames with the total number of frames. As we're using openCV, there are some improvements needed like the number of training images, orientation of the faces, make-up of actors, etc.