# CMSC733: Project 2 - FaceSwap

Abhilash Mane
University of Maryland
Email: amane@umd.edu

Advait Patole
University of Maryland
Email: apatole@umd.edu

## I. Phase 1:Classical Approach

Fig. 1: Img for faceswap

In this part, we'll go through how to perform Face Swap in two different scenarios: (1) a face in a video with a face in an image, (2) a face in a video with a face in an image. Frame is extracted from video this frames are then processed. This frame undergoes different procedures for successful face swap.. The outcome of each phase will be shown on frames of each video in the subsections that follow. We mostly stick to the directions' steps. Three main phases make up the algorithm: (1) Detecting Facial Landmarks, (2) Face Warping 1)Triangulation or 2) Thin Plate Spline), and (3) Blending

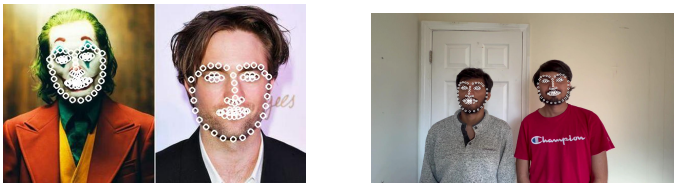

Fig. 2: Image for facial landmarks

### A. Detection of Facial Landmarks

The algorithm's initial phase is to recognize face fiducials/landmarks. The "dlib" library is used for this. We can get the position (x, y) of 68 key points of the identified face using this library. A pre-trained model is used to identify the spots. It returns the points in the order in which they were entered. This step's result may be seen below.

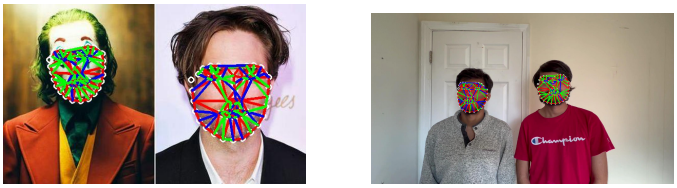

Fig. 3: Image for Delaunay triangle

### B. Warping or swaping face

In this section we talk about methods on how to warp the one image on another also known as face swaping. We implemented 2 methods 1) Warping using Triangulation and 2) Warping using Thin Spline Plate

Fig. 4: Image for Delaunay triangulation swap

Fig. 5: Image of face swap on blank canvas

*1) Warping using Triangulation:* The Delaunay Triangles will be created using the observed face landmarks in this stage. This method aims at maximizing the shortest angle in each triangle, and as a result, the source and target face images have the same triangulation. This resolves the issue of correspondence. Fig references The triangles are now warped from the target face to the source face. We perform inverse warping so that we don't get blank spots or holes. To ensure that the points are inside the triangle, we employ Barycentric coordinates. Then, using interpolation, we iterate and find the values for corresponding pixel using inverse warping. There is a bug in the opencv Delaunay triangle, where in it misses a triangle which can be seen in the figure.

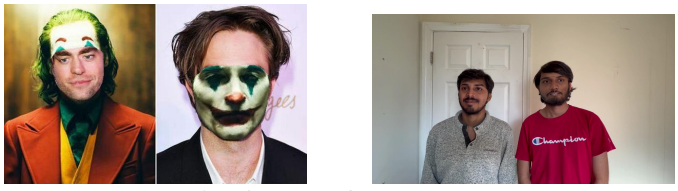

Fig. 6: Image for TPS swap

*2) Warping using TPS:* In TPS interpolation a relation in terms of spline is calculated between source and destination and this is used to do the inverse warping.To decrease the time consumption we increase the space complexity we convert 1d dimensions to 2d dimension to handle the repeating data.

Fig. 7: Image for blending of TPS swap

### C. blending

### D. Tracking face

In some of the the test videos at some time the face was getting miss by the dlib hog detector for this reason we implement tracker for the image. We implemented online Multiple Instance Learning tracker. It works successfully for small displacements.

## II. Phase 2: Deep Learning Approach

In our phase 2 implementation, we have used the code as provided in the instructions from the paper.For swapping two faces we have used dlib library to extract the region of faces in each frame of video and these faces were swapped using PRNet.The swapped faces were fixed at the respective regions from where they were extracted.For single face face swap we have directly used PRNet API and used other image which was supposed to be swapped.

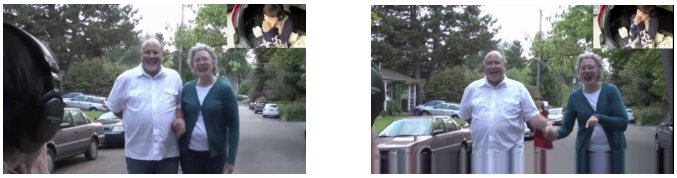

Fig. 8: Delaunay face swap for Test Set 2

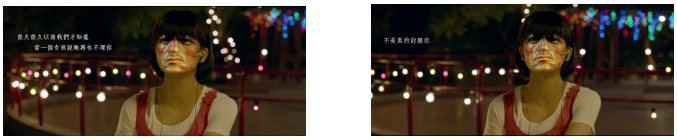

Fig. 9: Delauany face swap for Test Set 1

Fig. 10: Delaunay face swap for Test Set 3

Fig. 11: Delaunay face swap for Single Face

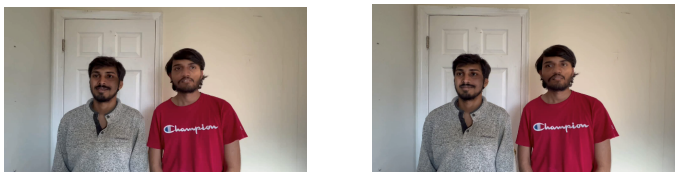

Fig. 12: Delaunay face swap for two faces

## III. Results and Conclusions

We evaluated the performance of both the traditional methods and the Deep Learning methods. PRNet gives the best result but it requires GPU and a lot of computation power to run it.Thin Plate Spline is faster than Delaunay Triangulation but the accuracy of Delaunay triangulation is much higher.We also presented the results for all the test files.


Fig. 13: TPS face swap for Test Set 2


Fig. 14: TPS face swap for Test Set 1


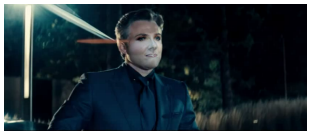

Fig. 15: TPS face swap for Test Set 3


Fig. 16: Output of TPS face swap for Single Face

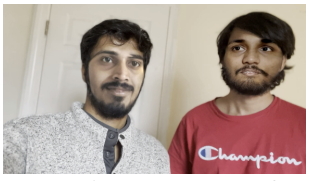


Fig. 17: Output of TPS face swap for two faces


Fig. 18: Output of PRNet face swap for Test Set 1


Fig. 19: Output of PRNet face swap for Test Set 2

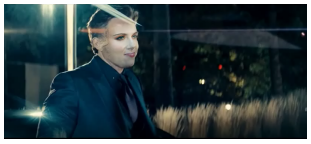


Fig. 20: PRNet Output face swap for Test Set 3


Fig. 21: Output of PRNet face swap for Single Face


Fig. 22: Output PRNet of face swap for two faces