Project Report (Project Term jan-may 2022)

On

Improving Video call Experience

Submited by : Yaswant Aditya

Regno. : 11902268

Section : KM015

Course code : INT247

Under the Guidance of

Dr. Sagar Pande

School of computer science and engineering

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in **CSE (Programme Name)** from Lovely Professional University, Phagwara.

Signature and Name of the Mentor : Dr. Sagar Pande

Name of student : Yaswant aditya

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.
Date :

DECLARATION

We hereby declare that the project work entitled Improving **Video call Experience** is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in **CSE (Programme Name)** from Lovely Professional University, Phagwara, under the guidance of **Dr. Sagar Pande**, during january to may 2022. All the information furnished in this project report is based on our own intensive work and is genuine.

Name of Student : Yaswant aditya
Registration Number: 11902268

Signature of Student : Yaswant
Date:

Index

Abstract

We know that some people are afraid to show them selfs in video calls as they fell embarrassed or shy in front of camera, therefore I came up with an fun soultion. The solution is face swapping, In this the Face was used as one of the normal behaviors for identifying humans. I have made this project to help people who feel comfortable at the same time have fun while on video calls. The entire project is divided into three stages, the first stage extracts the frames from a video file. The Second stage includes face detection and landmark extraction from frames for which we have used the get_frontal_face() and predictor() function of dlib library in Python which tells us whether a face is present in an image and also provides us the landmarks of the facial features. In the third stage, we use Delaunay triangulation and affine transformation to find the triangles by their index values and the coordinates of the triangles then these triangles are matched and swapped from one face to another. Finally, the application will generate frames with swapped faces, which are further combined into a video file.

Introduction

Most of you would have used or seen many filters that let you swap your face with your friends, celebrities or even animals. These filters are very popular in social media platforms such as Instagram, Snapchat and face app. Have you ever wondered how this is done? With Python and OpenCV it is actually very simple to build this application. The concept behind this is to detect certain points on the face and then replace it with the swapping image.

How face swapping is done?

Ever since the emergence of advanced computer vision techniques, there have been multiple attempts at developing face swapping techniques. Since then, many software, like adobe, have integrated the tools to perform face swapping in their photoshop software.

Similarly, people in the field of artificial intelligence have developed models for the implementation of face swapping. DeepFacLab's implementation of deep learning models for the purpose of face swapping is one example that provides an integrated framework to perform face swapping between pictures.

Aside from these, the method used in dlib for face swapping is also based on a machine learning based solution. It uses an ML model for landmark detection in a picture which aids in facial recognition. This facial recognition using landmark detection can then be extended to perform face swapping between images. In summary the methods for face swapping include.

- Classic Photoshop in Adobe software.
- Artificial intelligence eased methods
    - DeepFaceLab
    - Landmark detection with dlib

The landmark detection gives the facial recognition part of the swapping process. After the recognition, we extract the face from one image by deleting the part except the face and in the second picture, we delete the face.

We then place the extracted face from one image on the image with the deleted face in the place where the deletion occurred. This obviously does not provide a good result because the outline of the new face would not match or merge with the new image. So, we have to smooth out the boundary of the face and process both images a little at the boundary so that the new face merges well with the new image and it looks realistic.

Recent days have seen a revolution in human computing systems. Detection and identifying of faces from the collection of images is often simple for a person. Nonetheless, it should be a difficult part for a machine because it should be properly programmed in such a way that it would be able to detect a human face when data is provided by the user. The toughest part of the system is recognizing the faces irrespective of the lighting conditions. To overcome this problem in OpenCV uses underexposed image correction to properly illuminate the image, to identify an individual in an image it uses the Haar Cascades classifier's get_frontal_face_detector() function by evaluating and comparing specific features of the face region with the saved data collection in the dlib module. Once the faces are detected then the shape predictor 81 face landmarks is used to identify the facial landmarks like eyes, nose, eyebrows, lips, jaw-line which are then stored in a text file. These landmarks are then used to calculate the Delaunay triangles, which are then further transformed and superimposed on the second image.

The Delaunay Triangulation/Voronoi Diagram technique well known in computational geometric is applied on the edge enhanced binarized facial image. Facial features are classified and extracted in terms of various types of Delaunay triangles and the dual of a subset of the Delaunay triangles, Voronoi edges form the skeleton of facial skin. That is, facial feature's shape is described by Delaunay Triangulation/Voronoi Diagram. Furthermore, the facial features can be identified. The method succeeds in locating facial features in the facial region exactly and is insensitive to face deformation. The method is executable in a reasonably short time.

The various applications of this system includes the entertainment industry for example it can be used in stunt scenes to swap the face of stuntman with an actor, to create deep fake videos.

Literature Review / Related works

Ying Zhang et al. have suggested the Automatic Face Swapping System and Its Identification. In his paper the author has defined a system for face recognition and replacement based on machine learning algorithms being applied. For face detection, They had used SURF (Speeded up robust features) algorithm. It is a local descriptor used for identifying key points from an image. (key points like eyes, nose, etc.) by key points machine will understand if the person present in the image is watching toward the right or left.

Existing face swapping algorithms can be roughly divided into three categories: replacement-based, model-basd,and learning based. The replacement based method usually replaces the face region in the reference image with the input face region and then applies some image processing techniques to enchance the real sense of the synthesized image.

In the model based approach, a two dimensional or three-dimensional parametric feature model is established to represent human face, and the parameters and features are well adjusted the model parameters.

In the learning based models, the reference image is converted into a synthesized face image by training a generative neural network that contains the information of the input image. This model needs to build while slightly degrading the quality of the synthetic picture.

In the replacement based approach is simple and fast but sensitive to the variation in posture and perspective. The model based method can effectively solve the perspective problem; however, it usually needs to collect three-dimensional face data, and robustness is not somethin to be satisfied. The learning based approach can produce quite real and natural synthetic face

image, while usually requiring a large number of training data and having more restriction on the input and reference faces. Based on the comprehensive consideration of the charateristics of the above three methods, a face swapping alogorithm supported by the facial landmark aligment is proposed uner the replacement based framework.

Adding to this , other algorithm has been used in this project to achieve results, such as facial landmark detection, face warping.

N. M. Arar, N. K. Bekmezci, and F. Güney, H. K. Ekenel suggested a face swapping method, which replaces the face of the consumer with the face of a chosen famous individual in the database. The device communicates with the user through a user interface that allows it to pick the replacement face and represents the modified appearance directly[3].

Mangayarkarasi Nehru utilized Viola Jones Face Detection Algorithm. The key concept perceived here is to make the computers recognize the faces in addition to the non-facial structures in an image and finally to render the image with the selected portion [4].

Zhang Xingjie, Joongseok Song, Jong-Il Park used Image blending method to totally remove the boundary line during image synthesis and preserve the major parts of the human face like eyes, nose and mouth. They also demonstrated that the method can measure the algorithm in close to real-time with a screen resolution of 600 X 600

Overview of the Project

1. We will first install the numpy, opencv,dlib, pandas,time libraries.
1. And now we will use imread function of opencv to read source image.
2. I created a function that we need later on to extract the indexes of the triangles from the landmarks point array.
3. Then we will convert the images to grayscale images. Then we will intialzie a matrix with the size same as the shape of image with the face we will use.
4. We will intialize open cv's web cam enabling function.
5. Now we will use the pre-trained face landmark detector we downloaded to recognize the face in the first image. As seen in the for loop, we detect 68 landmarks to recognize the face in the image. We extract the coordinates of each of the 68 landmarks in the for loop.
6. Begin For outer foor loop
7. We will find the Delaunay Triangulation of the landmarks point of the first face.
8. Next, we use the convexhull module of the OpenCV library to draw the contour around the face that is detected from the landmark detection model.
9. We will find the indexs of each triangle. In other words we want to know not only the coordinates of the triangles, but what specific landmarks points each triangle connects.
10. Outer for loop END
11. Now we will start while loop to capture each frames from web cam
12. And we we will perform same steps on web cam frame as we done on source image

13. After creating Delaunay Triangulation for both faces then we will perform some operation to match the triangles of source image to match the triangles of destination image to perform swapping.
14. After performing the operation we will seprately join all the triangles
15. Then we will run swap function to swap both faces
16. After that we will perform seamless cloning to make swapped face appear more realistic by matching the face color and perfectly mrging the face.




Circuit Description

Our proposed system detects faces from images and swaps those faces, to create a new image these images are then combined to form a video file. To swap the face into video we have to follow these steps- frame extraction, face detection, landmark extraction, and creating landmark files according to the system block diagram shown in Fig.

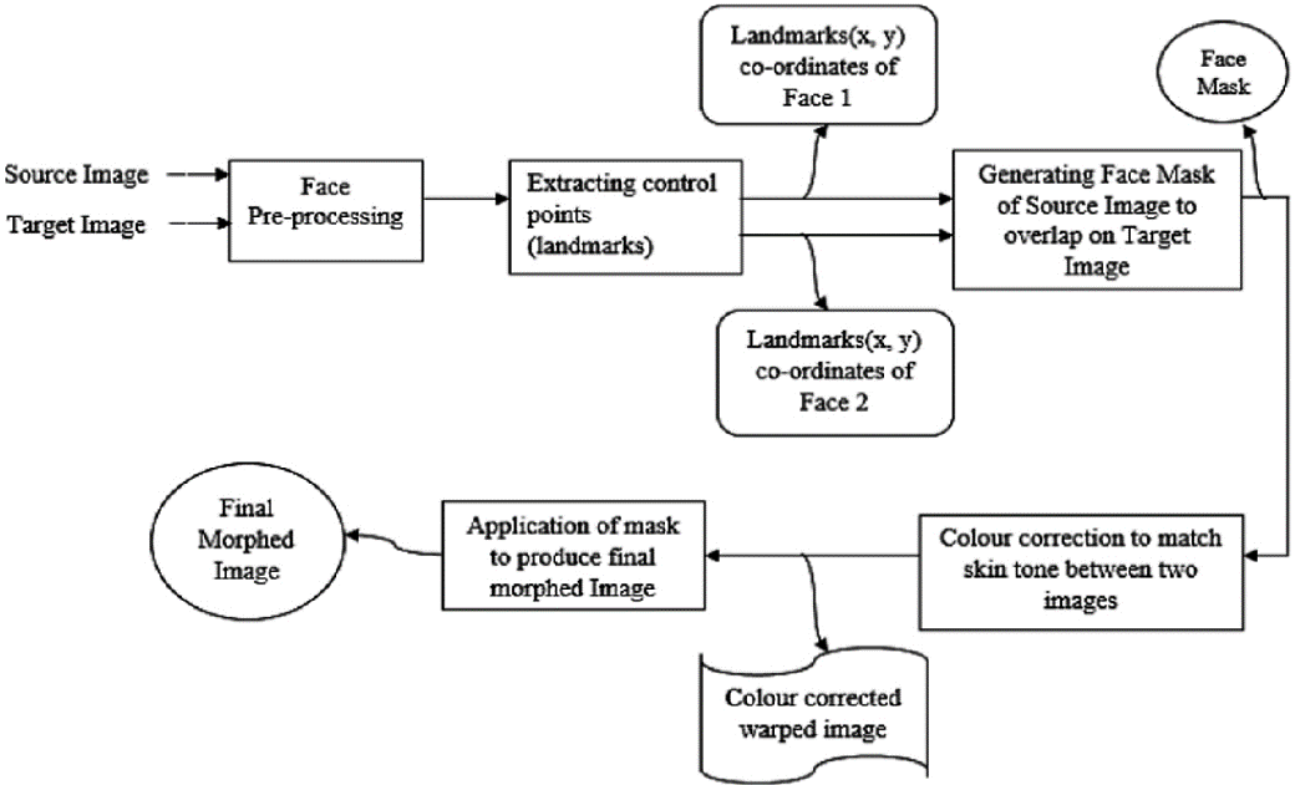

Fig. Block diagram

Face Pre-processing :- converting source and Target image into greayscale image and also detect facial landmarks. we use the pre-trained face landmark detector we downloaded to recognize the face in the first image.

Extraction of control points :- Setting up environment for facelandmarks and extracting the face landmarks points. we use the convexhull module of the OpenCV library to draw the contour around the face that is detected from the landmark detection model.

Landmarks(x,y) coordintaes of face 1 :- Extracting delaunay triangulation landmarks coordinates of face 1 and storing it into the indexes.

Landmarks(x,y) coordintaes of face 2 :- we can find the same triangles starting from the landmarks points of the second face. Extracting delaunay triangulation landmarks coordinates of face 2 and storing it into the indexes.

Generating face mask of source image :- We are adjusting the source image triangle parts to match to traget image parts, and then joining the adjusted source image triangles to generate face mask of source image to overlap on target image.

Color correction :- We convert the image from Grey to BGR

Applictation of mask :- After that we swap the source mask to destination mask then we perform seamless cloning.

Final morphed Image :- then we acquire our final morphed image.

Software Tools

Numpy :- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

Pandas :- Data analysis requires lots of processing, such **as** restructuring, cleaning **or** merging**,** etc. There are different tools are available for fast data processing, such as Numpy, Scipy, Cython**,** and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.

Open CV :- OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++.

Time :- The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Dlib :- According to dlib's github page, dlib is a toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings.

Results

Conclusion

Computer vision is an extensive domain with a vast number of applications. Face swappng is one them. Modern computer vision based on neural networks and deep learning is the most efficient way of developing a face-swapping model. Dlib and OpenCV can be used to create a face swapping model.

Dlib is an extensive library on C++ that can be used to develop machine learning models. It provides a pre-trained model for machine landmark detection for facial recognition.

The approach which we are using introduces the training results of existing learning models directly. The method proposed does not require any training data, because it uses no learning model for new training. The experimental results show that the composite image obtained by our model has a great reality and strong adaptability to the difference of skin color occlusiong while retaining most of the facial features of the input image.

After implement this method we can see that the source and target image will be approx. 70-80 % identical. Which is great result in case of this method for image and video synthesis.

Future Scope

- This project method can be used in video calls for fun.
- Most of all it has huge scope in animation, gaming and film industries.