Automated Face Detection And Swapping In Video

Jayant P. Patel[1], Amruth M. Shetty[2], Amar N. Vishwakarma[3] ,Mrs.Vishakha Shelke[4]
1jayantpatel275@gmail.com[2]amruthshetty437@gmail.com,[3]amvkind@gmail.com,

[1,2,3] BE Student ,Department of Computer Engineering, Universal college of Engineering, Mumbai
[4] Assistant Professor, Department of Computer Engineering, Universal college of Engineering, Mumbai

Abstract. Face was used as one of the normal behaviors for identifying humans. It generally stays a challenge to cause a computer framework to carry on like how a human senses things. We have made our application to help and simulate the visualization of a human's imagination of Deep Fake videos. The entire face swapping 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.

Keywords: Automated Face Swapping, Face Replacement, Face Detection in video, Face swapping in video, Deep fake video.

1. Introduction

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 [7] 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[9] 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 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. To the best of our knowledge this paper is the follow-up work "Ying Zhang"[1] which discusses the importance of face swapping detection and proposes an automated solution. The system proposed enhances the accuracy of the present system.

2. Literature Survey

Ying Zhang et al. have suggested the Automatic Face Swapping System and Its Identification. In This 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 [1].

Zhang Xingjie, Jong-Il Park and Joong Seok Song. A blending approach has been suggested that can resolve the question of frontier lines in face detection, which can lead to unrealistic faces by using an adaptive weight value and demands low computational expenses. This paper verifies that during image synthesis, the proposed approach can effectively erase the boundary line and retain the main parts of the human face, such as eyes, nose, and mouth. In this paper the technique of blending is based on the alpha blending process suggested by Thomas Porter and Tom

ICASISET 2020, May 16-17, Chennai, India

DOI 10.4108/eai.16-5-2020.2303961

Duff [12] to blend the images and remove the boundary line of the face effectively. An efficient method of blending is suggested in this paper by using the adaptive weight value for natural face swapping [2].

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].

Richard R. Schultz has developed a method for removing frames from video sequences in high resolution. It is an observation model that was proposed for a video sequence based on motion compensated subsampling,It uses the Bayesian restore algorithm to obtain a high-resolution video for the specified short low-resolution sequence with a discontinuity-preservation of the prior image model. It calculates from a low-resolution image sequence with a panshow of a subpixel camera, dramatic visual and quantitative improvements over bilinear, cubic B-spline, and single-frame Bayesian interpolations [5].

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 [6].

3. Implementation of proposed system

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 1. In the further examples we have used bollywood actor Mr. Akshay Kumar's Image [9] for representation of the results.

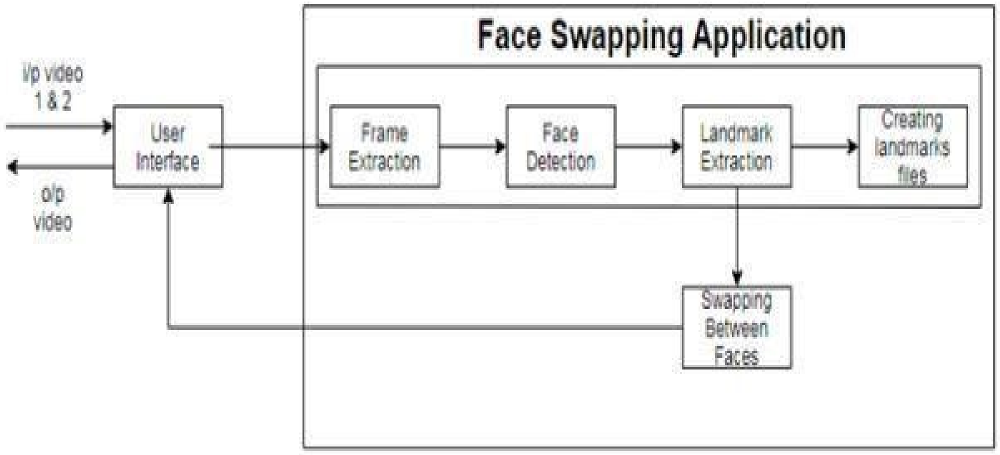


Fig. 1. System block diagram

3.1 Extracting Frames from Videos

We take the Subject and Base videos and then extract the frames from both the videos using cv2.VideoCapture() function of OpenCV which is implemented by using a while loop. The while loop extracts a certain number of frames and then saves those frames in separate folders.

3.2 Detecting faces in an image

To detect faces we use the get_frontal_face_detector() function of the dlib library to check whether a face is present in an image or not. As shown in Fig. 2.

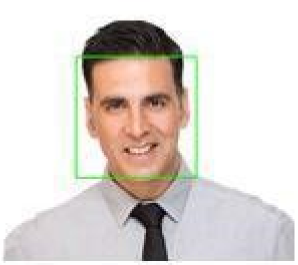


Fig. 2. Detected Face

3.3 Finding Landmarks from Frames

In this we extract the 81 landmarks per face using the dlib predictor function, as shown in Fig 3. These landmark points are stored in a 2 dimensional matrix which are then written in a text file.

Fig.
3. Landmarks plotting

3.4 Delaunay Triangulation and Transformation

This is performed by taking the landmark points of the subject image and then using the subdiv.getTriangleList(), as shown in Fig 4. function to get all the triangles coordinates, the triangles from the first image are swapped on the second image after transformation using the affine transform method.

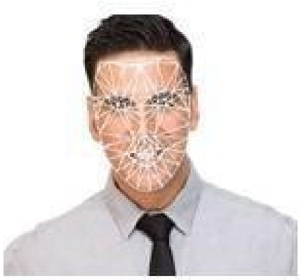


Fig.
4. Plotting of Delaunay Triangles

3.5 Colour Correction

To generate realistic faces, we should ensure the skin-tone and lighting between the two images is similar and will not cause obvious discontinuity around the edges of the overlaid region. We applied a Gaussian smooth filter to the mask for this purpose, which is used to integrate two facial images.

3.6 Merging of the Frames to obtain a Final Video

The Swapped frames which are obtained using the system are then merged together to create a final output video file.

4. Results and Discussions

In this section the comparative analysis is done with existing system and proposed system.

4.1 Existing System

The previous system working on shape_predictor_86 model [10] is shown in Fig. 5. We identified the research gap that the model is limited to identifying face coordinates till eyebrows the forehead landmarks of the face were not covered affecting the accuracy of the results.

Fig. 5. Existing System Output

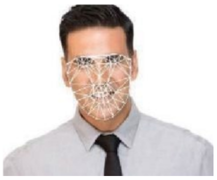
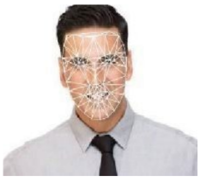
4.2 Simulation result of Proposed System

To fulfill the research gap discussed in the result of the existing system we have used the shape_predictor_81 model on the image [9]. It can be observed that, shape predictor 81 facial landmarks can recognize a full face along with the forehead. It also provides better accuracy as compared to the existing system.

Fig. 6. Simulation of Result

The Comparison of the existing and the proposed system are shown in the below Table 1.

Table 1. Comparison of Systems

System	**Author**	**Model**	**Result**
Existing System	Ying Zhang [1]	Shape Predictor 86 Facial Landmarks	
Proposed System	Nikolay Feldman [10]	Shape Predictor 81 Facial Landmarks	

The landmarks are used to calculate the delaunay triangulations which are further transformed according to the base image triangulations to form the final output video.

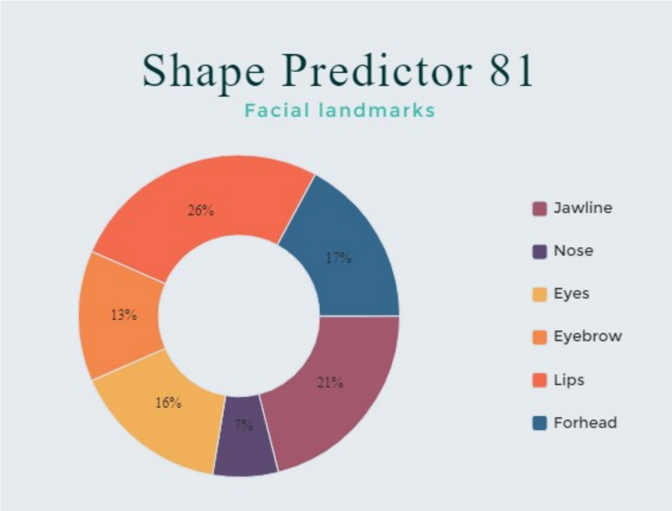


Fig. 7. Graph of shape_predictor_81 model

The distribution of landmark points i.e. jawline, nose, eyes, eyebrows, lips and forehead according to the points on the face is shown in the Fig 7.

The range of the landmarks are used to align the subjects face onto the base image face by matching the triangulations and transforming according to alignment of the base image face. Coordinates are calculated and placed on the landmark points respectively, Table 2. gives a brief description about the ranges of the landmark points on the face used in the proposed system.

Table 2. Details of landmark position

Facial Features	Landmarks Range
Jawline	0 to 16
Eyebrow	17 to 26
Nose	27 to 35
Eyes	36 to 47
Lips	48 to 68
Forehead	69 to 81

Steps to obtain result in our proposed System:

We have implemented all the experimental work discussed above in our final year B.E project. We have implemented a face swapping concept in the videos. We have presented the swapped video in the final output.

The steps to get the result is as follow

1. Select Subject and Base video from the Browse button from the interface.
2. Click on the Upload button to start the Swapping process, this may take between 2 to 5 minutes.
3. The result is shown in the third section of the page.

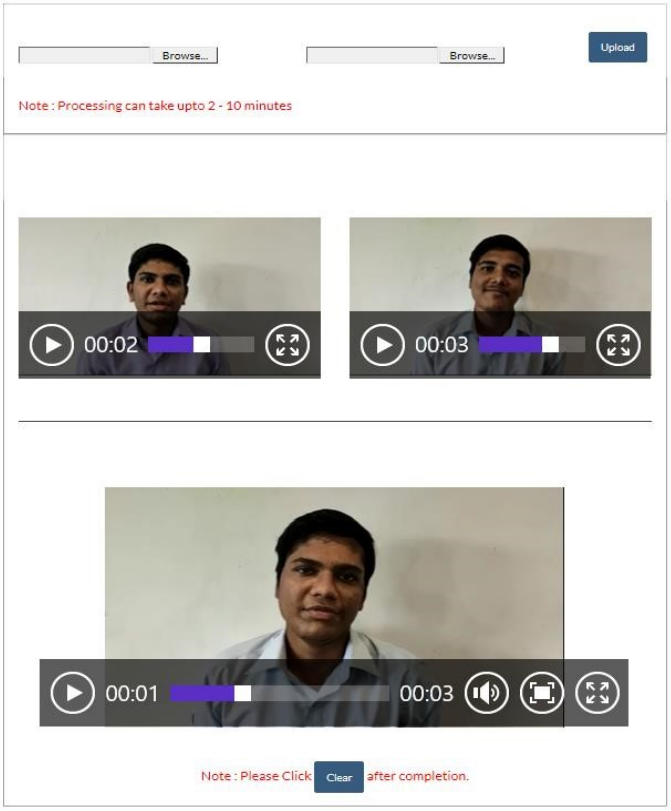
4. The Clear button is used to clear the memory so we can upload new videos to prevent wastage of memory space.

The frame rates of the videos used in the proposed system is shown in Table 3. Frame rate of a video describes the number of frames a video puts out in one second, higher frame rate is used to reduce the jitter in a video. In our proposed system we have kept the frame rate to 24fps in order to optimize the system to get quick and smooth videos.

Table 3. Video Frames Description

Video	Frames
Base	60fps
Subject	60fps
Simulation Output	24fps

The final implementation of the proposed system is presented using a GUI interface shown in Fig 8. Where we have successfully demonstrated the face swapping in the video.



Note : Processing can take upto 2 - 10 minutes

Note : Please Click Clear after completion.

Fig. 8. Interface of Simulated System

5. Conclusion and future work

This paper presents a brief introduction to the process of swapping facial regions between videos of different individuals. We have made our application to help and simulate the visualization of a human's imagination of Deep Fake videos. It can be applicable in the film industry for entertainment purposes. Object recognition can be a good improvisation of our proposed system. The system lacks options for texture identification that makes it tough for the system to spot objects. The output is extremely addicted to factors like noise within the sketch, accuracy of the fed sketch and it's well outlined boundaries. This generally leads to an output that's not fascinating by the user, all these factors can be looked and improved upon in the near future.

References

1. Ying Zhang, “Automated Face Swapping and Its Detection” , **IEEE 2nd International Conference on Signal and Image Processing, 2017**
2. Richard R. Schultz, “Extraction of High-Resolution Frames from Video Sequences”, **IEEE Transactions on Image Processing**

Vol 5 , 1996

3. N.Murat Arar, “Real-time Face Swapping in Video Sequences: Magic Mirror” , **IEEE 19th Signal Processing and Communication Applications Conference , 2011**

4. M. Nehru, “Illumination Invariant Face Detection Using Viola Jones Algorithm”, **ICACCS , 2017**

5. Changxing Ding, “Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition” , **IEEE , 2017**

6. Zhang Xingjie , Joongseok Song , Jong-Il Park , “The image blending method for face swapping” , IC-NIDC , 2014
7. “Face Detection using Haar Cascades” **[Online] Available; https://opencv-pythontutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/pyfacedetecti on/py_face_detection.html**

8. “Dlib library” [Online] Available; https://pypi.org/project/dlib/

9. “Akshay kumar Image” **[Online] Available; https://www.bollywoodhungama.com/wpcontent/uploads/2019/01/CONFIRMEDAkshay-Kumar-to-playPrithviraj-Chauhan-film-goes-on-floors-this-year.jpg**

10. Nikolay Feldman, “Shape Predictor 81 Landmarks” **[Online] Available; https://github.com/codeniko/shape_predictor_81_face_landmarks 11. “Shape Predictor 86 Landmarks” [Online] Available;https://github.com/nicoTrombon/faciallandmarks/blob/master/facial_landmarks.py**
12. Porter, Thomas, and Tom Duff. Composting digital images. **ACM Siggraph Computer Graphics. Vol. 18. No. 3. AC**