Front Facial Reconstruction for Deep Fake Applications

Mu-Ruei Tseng
Texas A&M University
133007868

Meetansh Gupta
Texas A&M University
534002289

Matthew Chang
Texas A&M University
828004952

1 Introduction

Deepfake technology has gained significant attention in recent years. With the advancements in generative models, state-of-the-art techniques can now generate high-fidelity forged images from a single source image paired with a corresponding driving video or audio. However, we've observed performance limitations when the provided source image isn't in a frontal position (facing towards camera), which can lead to distortions during animation with a driving video. Moreover, acquiring frontal face images in the wild is often challenging and very rare.

Our project aims to design a model that reconstructs a person's face in frontal position from side-view images. We believe that leveraging multiple image inputs will enhance the synthesis quality when the source image isn't of a frontal face. This can be done by extracting spatial features of a side view face and estimating the 3D transformation matrix that would transform it into a frontal position. Applying the transformation and adding the corresponding color to the features would yield a reconstructed front-facing view of the individual.

2 Methodology

Our method can be separated into three parts: transformation matrix estimation, image bank generation, and front face recovery. We use the VGG dataset(2), which contains images of several thousand individuals captured in various poses and various angles of tilt and pan (including frontal and side views). We extract spatial feature landmarks as geometric information from each image using Google's Mediapipe model (1).

Blending these features with the colors of the image as texture information, we design a pipeline that translates these features in a 3-dimensional space, representing the frontal face position using existing features extracted from an image with a front-facing individual. In the end, our algorithm can reconstruct the frontal view of a person's face using side profile images. This may have significant applications in deepfake technology. The overview of our implementation can be seen in Figure1

2.1 Finding the Transformation Matrix

Assume we two 3D facial landmarks, $L_s, L_t \in \mathbb{R}^{N\times 3}$. We want to compute a 3D transformation matrix M to convert L_s into the same space of L_t.

However, the depth information (z-axis) obtained from the 3D landmarks extracted by Mediapipe is not proportional to the x and y dimensions of the image. To address this disparity, it's necessary to normalize the landmarks to a consistent scale. We achieve this in two steps:

1. **Zero-mean the landmarks (L_o):**

$$l_o = l - \mu_L, \quad \forall l \in L \tag{1}$$

where l is the original landmark, μ_L is the mean of all landmarks L, and $L_o = \{l_o | \forall l \in L\}$ is the zero-meaned landmarks.

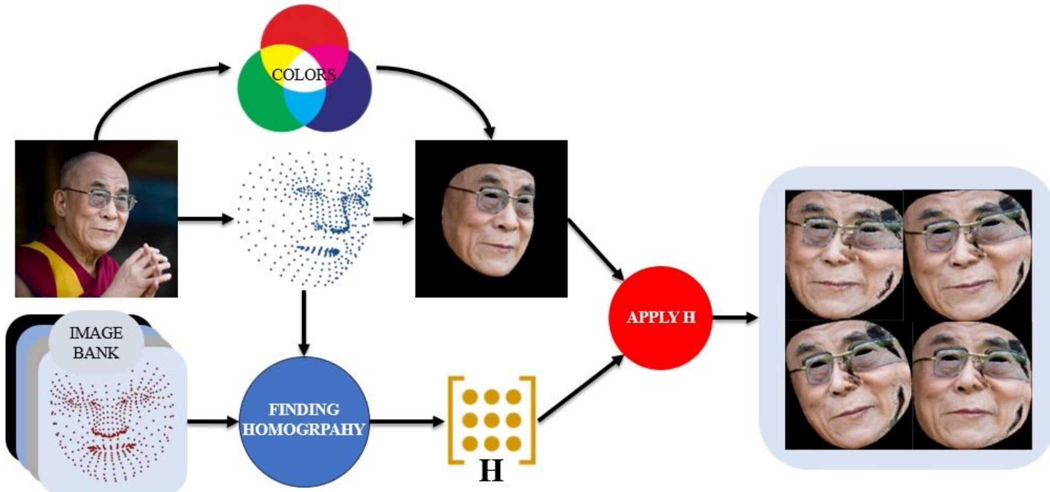

Figure 1: Overview Pipeline

2. **Rescale to the range** $[0, 1]$**:**

$$\hat{l} = \frac{l_o}{s_L} \tag{2}$$

where s_L is the maximum absolute value of each dimension in L_o and $\hat{L} = \{\hat{l} | \forall l_o \in L_o\}$ is the normalized landmark.

After finding the normalized landmarks $\hat{L}$, we can calculate the transformation matrix. Here, we restrict the transformation matrix to a similarity transformation with 7 degrees of freedom (scaling, translation, and rotation) to preserve the shape of our source landmarks. This gives us:

$$R = \begin{bmatrix} cos(c) & -sin(c) & 0 & 0 \\ sin(c) & cos(c) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} cos(b) & 0 & sin(b) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(b) & 0 & cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(a) & -sin(a) & 0 \\ 0 & sin(a) & cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$D = \begin{bmatrix} s & 0 & 0 & x \\ 0 & s & 0 & y \\ 0 & 0 & s & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\hat{M} = R * D \tag{5}$$

We can recover $\hat{M}$ using the least square approach given the corresponding normalized landmarks $\hat{L_s}$ and $\hat{L_t}$. Finally, we can obtain M as:

$$M = A^{-1}\hat{M}A \tag{6}$$

, where $\mu_L = [\mu_x, \mu_y, \mu_z]$, $s_L = [s_x, s_y, s_z]$, and $A = \begin{bmatrix} 1/s_x & 0 & 0 & -\mu_x \\ 0 & 1/s_y & 0 & -\mu_y \\ 0 & 0 & 1/s_z & -\mu_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the normalization matrix to convert L_s to $\hat{L_s}$.

2.2 Image Bank Generation

Since every person's face might be different, we generated the image bank that contains the mean frontal face landmarks for several people. These mean frontal face landmarks are used to estimate a similarity transformation matrix from any given source landmark. Assume M_b is the similarity transformation matrix estimated from the source face landmark L_s and subject L_b in the image bank, we define the error as:

$$E_b = ||wL_s - L_b||^2 \tag{7}$$

, where

$$wL_s = \text{warp}(L_s, M_b) \tag{8}$$

The best M will be defined as:

$$M = \text{argmin}_b\, E_b \tag{9}$$

To build the image bank, we first manually select several front-facing images for each subject from the training set. For each image, features are extracted as landmark points using Mediapipe (1), and a global mean is calculated to generate an average set of landmark points. This average set per subject is stored as the sample image bank B_s.

With the sample image bank established, we can automate the process of finding a front face and expanding the size of the image bank by estimating the 3D similarity transformation matrix between a given landmark L and all the average front-face landmarks from the sample image bank. This process of expanding the image bank resembles semi-supervised learning, where a pseudo-front face is classified if the average rotation (r_x, r_y, r_z) between the two landmarks is smaller than 10 degrees. The expanded image bank is denoted as B.

2.3 Front Face Recovery

For recovering the front face image, we experimented with two methods: a naive approach, which performs a simple 2D homography, and our proposed approach, which leverages a 3D similarity transformation.

2.3.1 Naive Approach

In the naive approach for front face recovery, we warp the landmarks in 2D from the side image to the front-facing image using a 2D homography. This results in a distorted image where facial features appear more appropriately sized relative to each other, but it fails to achieve the 3D rotation of the face that we aim for.

2.3.2 Our Approach

After finding the best similarity transformation matrix M, we reconstruct the front face by querying the color from the source image. For each 3D landmark, we use the color at its corresponding 2D position in the image by disregarding the z-axis. However, since the landmarks extracted from Mediapipe are relatively sparse, we augment them to create a denser set.

Landmark Augmentation Inspired by (3), we started with the Parallelogram Method, a well-known approach for randomly selecting points from a triangle. However, this method does not ensure that points are evenly spaced throughout the triangle. To address this, we combine the Parallelogram Method with the R_2 quasirandom sequence, resulting in a method that yields better spatial distribution.

For every group of three adjacent points in the mask, we interpolate points within the triangle formed by these three points using this combined method, until a specified density d is reached. This ensures that our augmented landmarks are sufficiently dense and well-distributed, allowing better reconstruction while performing color querying. We can denote the dense landmark as L^d, where d represents the density we achieved.

Color Querying Once we obtain the dense landmarks L_s^d from the source image, we apply the transformation matrix M to convert them into the front face domain. Each point retains the color from the query image. As the transformation is applied, the colors warp in alignment with the transformed landmarks, resulting in a reconstructed front-facing image of the queried individual.

3 Results

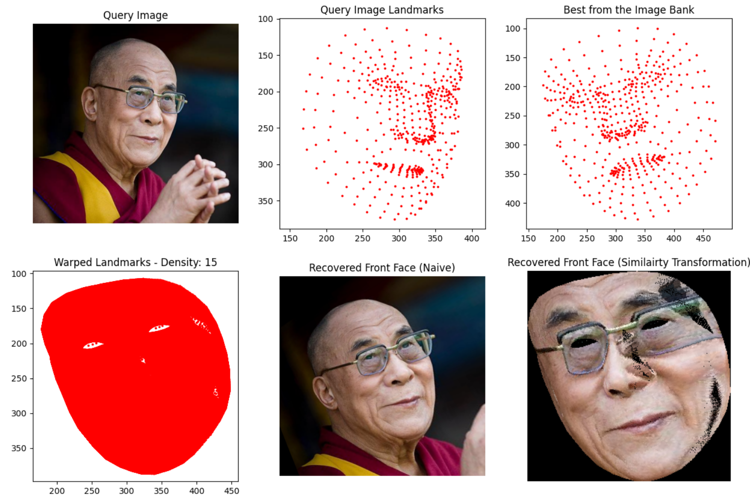

Figure 2: Result 1

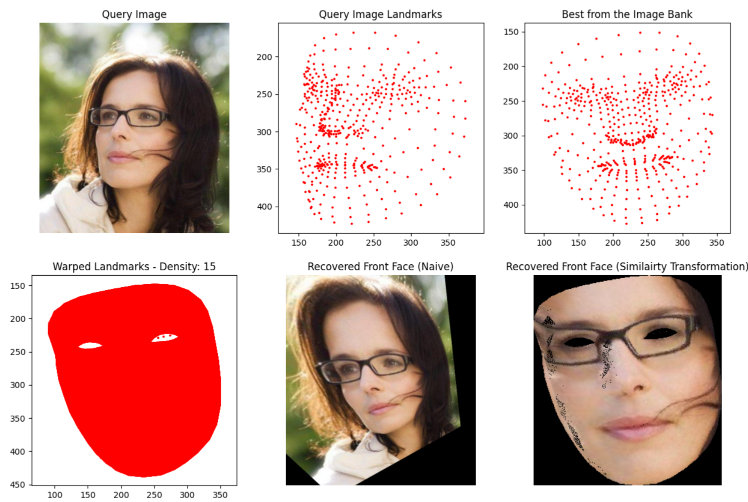

Figure 3: Result 2

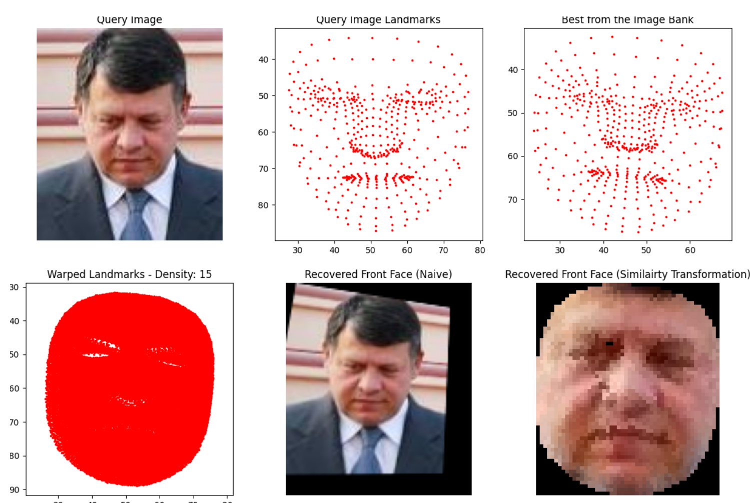

Figure 4: Result 3

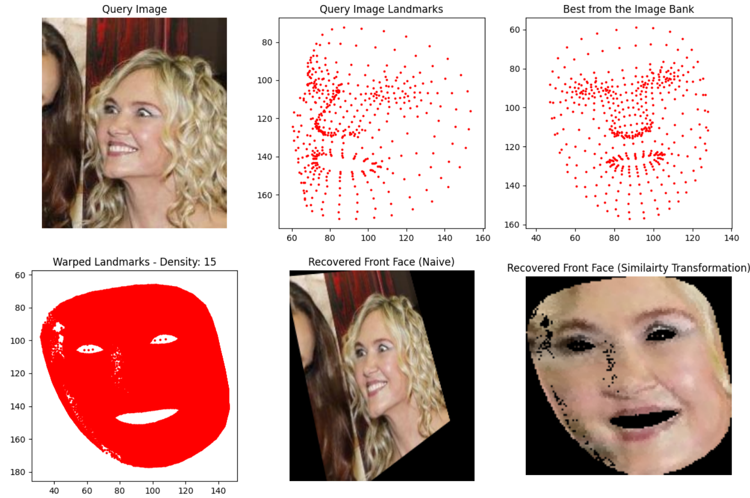

Figure 5: Result 4

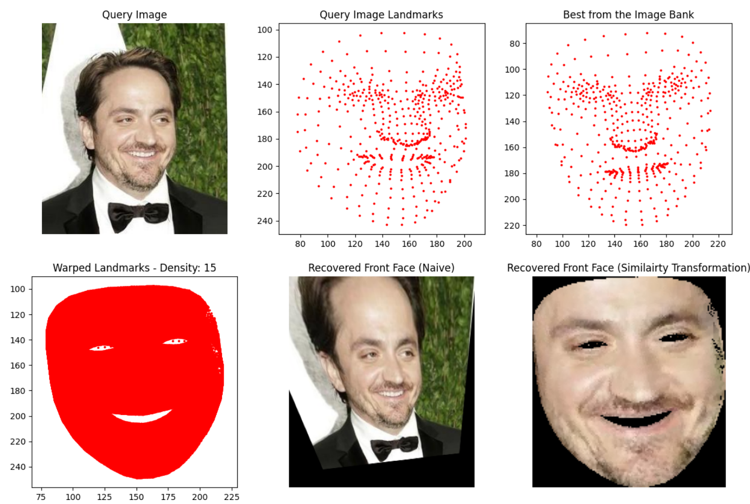

Figure 6: Result 5

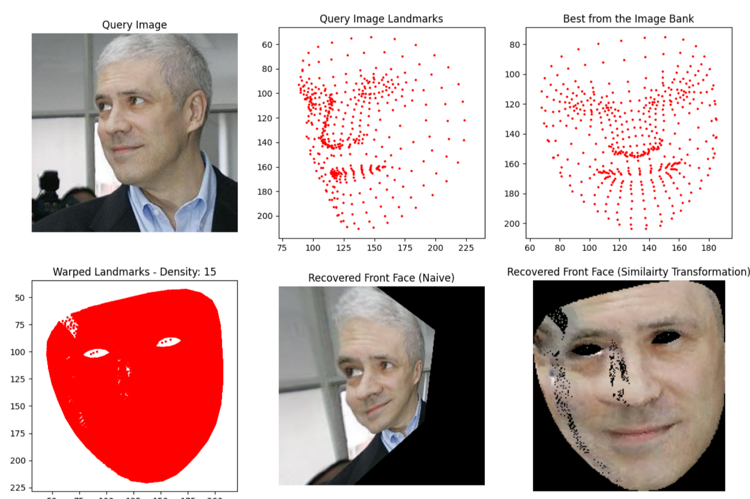

Figure 7: Result 6

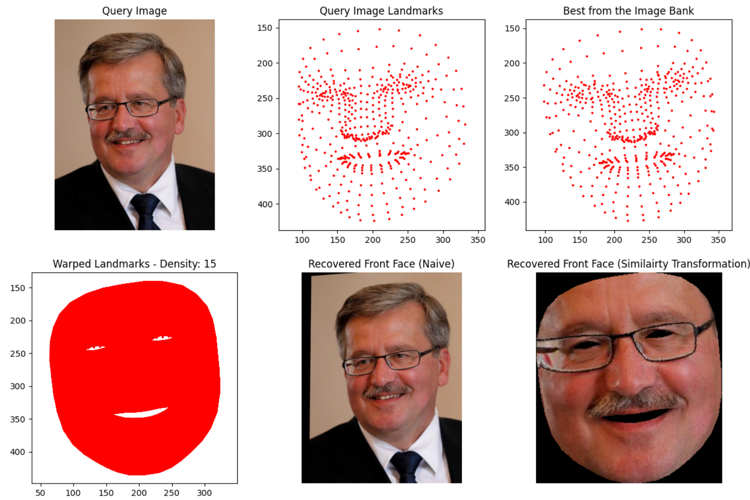

Figure 8: Result 7

4 Discussion

The results above show the effectiveness of performing the face landmark transformation in 3D. Instead of warping the face into a certain shape, we simulate the result of the person in the image actually turning his or her face towards the camera. We observe that the results that have a slightly turned face come out better than faces that are more fully turned away from the camera. Since we do not have as much information about the side of the face that is turned away from the camera, the recovered front face comes out noisier on that side.

One limitation includes filling in the eyes and mouth region. Both of these regions did not have any points from the Mediapipe face landmarks, so colors were not queried in these areas. Another limitation, also caused by the landmarks, is the cropping of hair and features surrounding the subject's face. Future work may include warping the eyes and Poisson blending them into the face, or expanding the region around the face to include hair and other external features like ears.

5 Code

The repository for the project can be found at https://github.com/Morris88826/face_reconstructure.

References

[1] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M., Lee, J., Chang, W.T., Hua, W., Georg, M., Grundmann, M.: Mediapipe: A framework for perceiving and processing reality. In: Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019 (2019), `https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf`

[2] Parkhi, O.M., Vedaldi, A., Zisserman, A.: Vgg face dataset. University of Oxford, Department of Engineering Science (2015), `https://www.robots.ox.ac.uk/~vgg/data/vgg_face/`

[3] Roberts, M.: Evenly distributing points in a triangle. (Feb 2024), `https://extremelearning.com.au/evenly-distributing-points-in-a-triangle/`