Github Repo: https://github.com/Digital-Image-Processing-IIITH/project-mandelbrot

Style Transfer for Headshot Portraits

Team : **Mandelbrot**

Team Members :

Amarthya Sasi Kiran
Sai Krishna Charan Dara
Sai Deva Harsha Annam
Venkata Susheel Voora

Mentor : **Soumyasis Gunn**

Motivation

- Headshot portraits are commonly taken in professional settings, and a lot of time and effort is spent in styling these photographs, through various editing methods.
- Each feature - eyes, skin, eyebrows, mouth, hair, all require specific treatment. This project uses a combination of various techniques to style an amateur input image using other, professionally taken styled portraits.

*Original image

*Styled image

Preview

- Users provide an input portrait photo and an example stylized portrait, and our algorithm processes the input to give it same visual look as the example.
- The output headshot that we seek to achieve is the input subject, but as if taken under the same lighting and retouched in the same way as the example.
- We also support the case in which an artist has produced a collection of portraits in a consistent style. In this case, our algorithm automatically picks a suitable example among the collection, e.g., matching beardless examples to beardless inputs.

Approach

- Foreground-Background mask generation
- Dense Correspondence
- Multiscale Transfer of Local Contrast
  - Multiscale Decomposition
  - Local Energy
  - Robust Transfer
- Additional Post-processing
  - Eye Highlights
  - Background

Dense Correspondence

To obtain the correspondence between input and example image we use the following approach

1. Facial feature point detection - First align the eyes and mouth of the example image with those of input image using affine transform.
2. Morphing - Morph the example to input using segments on the face template.

Facial feature point detection

We find 66 facial feature points on the faces of the input and example. We mirror the facial points corresponding to the jawline, and compress them, to approximately find forehead points. This procedure works in most cases since a human face generally satisfies this property. We also add feature points along the borders of the image so the entire image stays intact after morphing

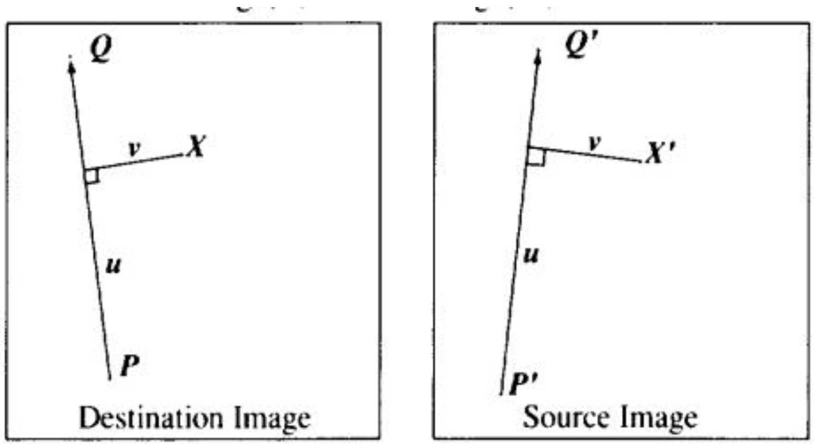
Morphing

Using the sets of points from the previous step, we find the Delaunay triangulation for each, and use a geometric morphing algorithm on these triangles, to morph the Example to the Input.

Reference - https://www.cs.princeton.edu/courses/archive/fall00/cs426/papers/beier92.pdf

Morphing (Beier and Neely '92)

For each pixel X in the destination
  $DSUM = (0,0)$
  $weightsum = 0$
  For each line $P_i\,Q_i$
    calculate u,v based on $P_i\,Q_i$
    calculate X'_i based on u,v and $P_i'Q_i'$
    calculate displacement $D_i = X_i' - X_i$ for this line
    $dist$ = shortest distance from X to $P_i\,Q_i$
    $weight = (length^p \,/\, (a + dist))^b$
    $DSUM \mathrel{+}= D_i * weight$
    $weightsum \mathrel{+}= weight$
  $X' = X + DSUM \,/\, weightsum$
  destinationImage(X) = sourceImage(X')



Multiscale Transfer of Local Contrast

- Our goal is to match the visual style of the example without changing the identity of the input subject
- That is, we want the output to represent the same person as the input with the same pose and expression, but with the color and texture distribution and overall lighting matching the example.
- Working at multiple scales allows us to better capture the frequency profile of these elements
- Technique builds upon the notion of power maps to estimate the local energy in each image frequency subband.
- We first assume grayscale images and that the region of interest is the entire image. We later explain how to adapt our algorithm to deal with colors and to use a mask.

Multiscale Decomposition

- The first step of our algorithm is to decompose the input and example images into multiscale Laplacian stacks
- Procedure for the input image I; the same procedure applies for the example image E.
- Construction uses a 2D normalized Gaussian kernel $G(\sigma)$ of standard deviation σ.
- Using $\otimes$ as the convolution operator, we define the level $L\ell$ at scale $\ell \geq 0$ of the input Laplacian stack as:

$$L_\ell[I] = \begin{cases} I - I \otimes G(2) & \text{if } \ell = 0 \\ I \otimes G(2^\ell) - I \otimes G(2^{\ell+1}) & \text{if } \ell > 0 \end{cases} \tag{1}$$

Multiscale Decomposition (Contd..)

- For a stack with n ≥ 0 levels, we define the residual as:

$$R[I] = I \otimes G(2^n) \quad (2)$$

Local Energy

- We estimate the local energy S in each subband by the local average of the square of subband coefficients.
- Intuitively, this estimates how much the signal locally varies at a given scale.
- Concretely, since we do not downsample the Laplacian layers, we adapt the size over which we average the coefficients to match the scale of the processed subband.

For the ℓth subband, this gives:

$$S_\ell[I] = L_\ell^2[I] \otimes G(2^{\ell+1}) \tag{3}$$

Local Energy (Contd..)

- For the example image E, we account for the correspondence field that we have computed previously . Using W(·) for the warping operator defined by this field, we compute:

$$\tilde{S}_\ell[E] = W(S_\ell[E]) \tag{4}$$

where we compute $S_\ell[E]$ with Equation 3. Estimating the energy before warping the data avoids potential perturbations due to distortion and resampling.

Robust Transfer

- Using these two estimates (Eq. 3 and 4), we modify the input subbands so that they get the same energy distribution as the example subbands.
- Letting O be the output image, we formulate a first version of our transfer operator as:

$$L_\ell[O] = L_\ell[I] \times \mathrm{Gain} \tag{5a}$$

$$\text{with Gain} = \sqrt{\frac{\tilde{S}_\ell[E]}{S_\ell[I] + \epsilon}} \tag{5b}$$

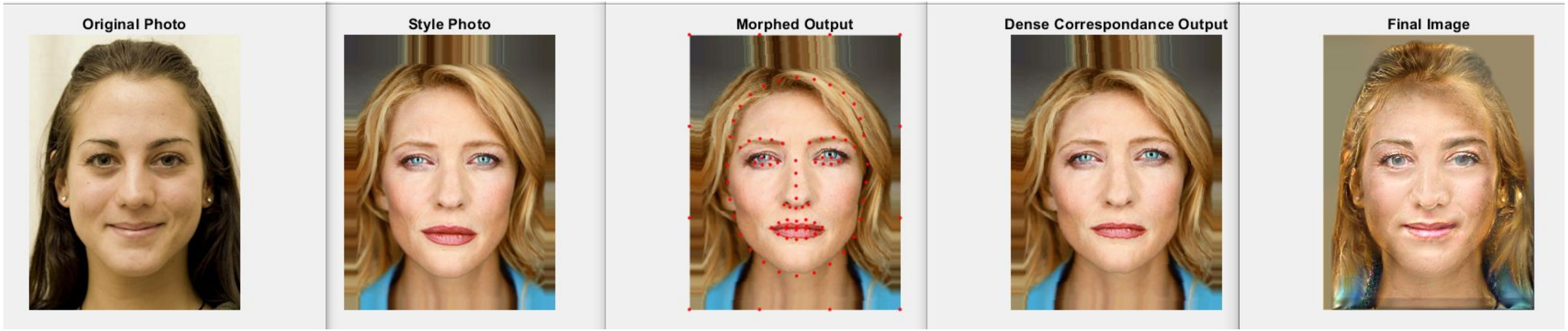
where ε is a small number to avoid division by zero (ε = 0.012,) is between [0,1]) and the square root compensates for the square used to define the energy in Equation 3.

TASKS:

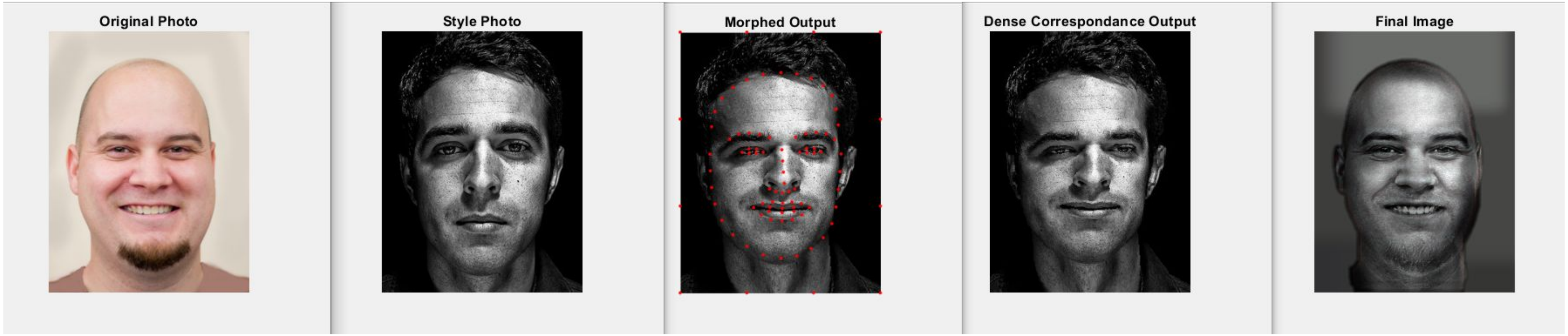
- Implement Grabcut for Mask generation.
- Identify facial features or keypoints
- Implement morphing as described in the paper, Beier and Neely, ‘92
- Multi-scale decomposition
- Local Energy
- Transfer using gain.

RESULTS

Result 1:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

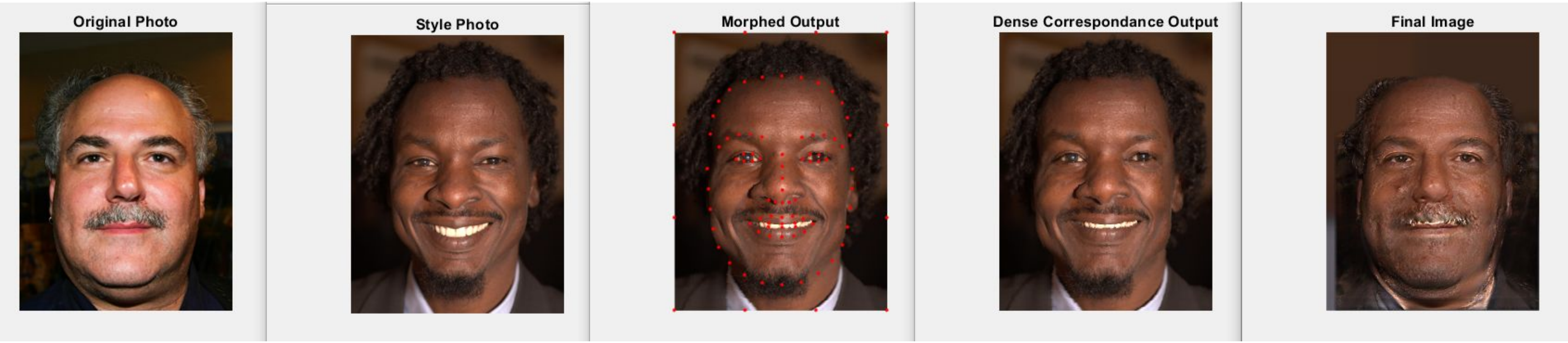
Result 2:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

Result 3:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

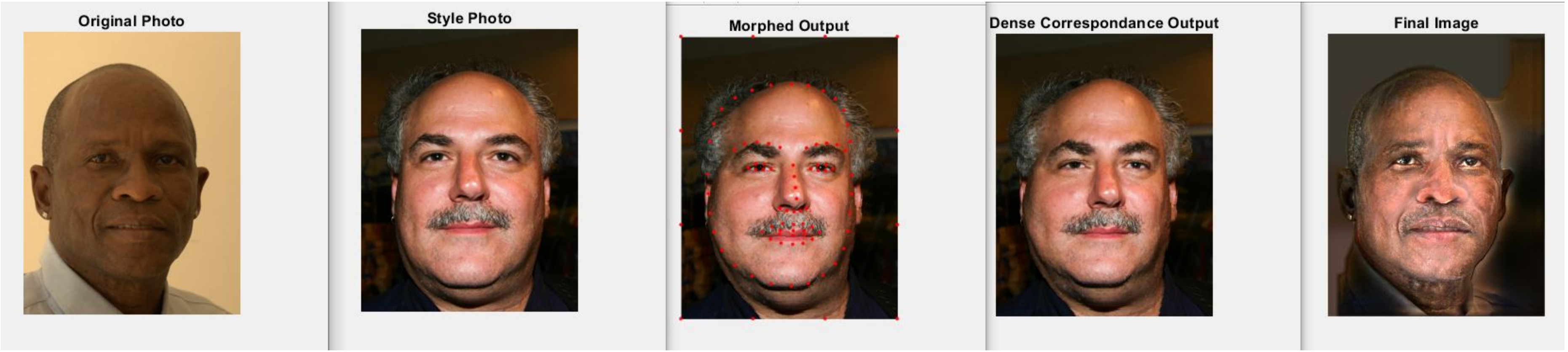
Result 4:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

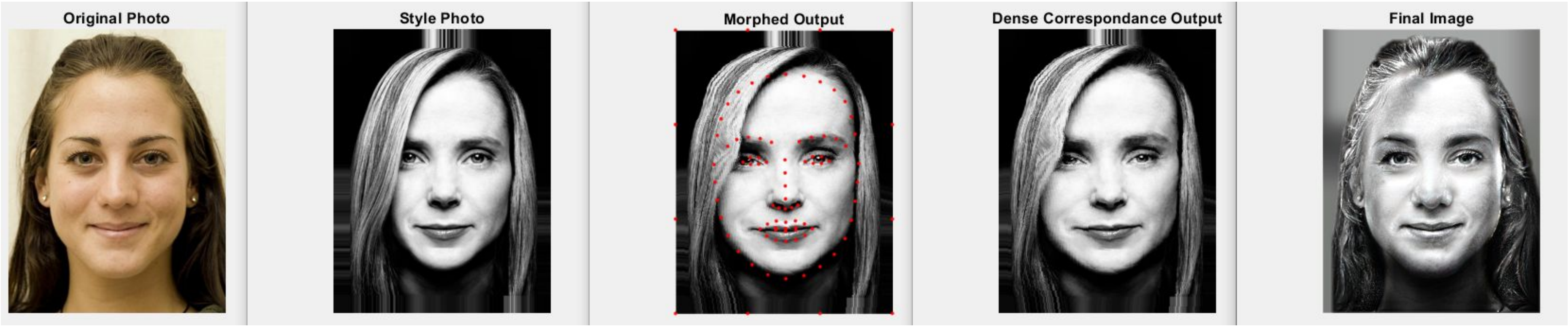
Result 5:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

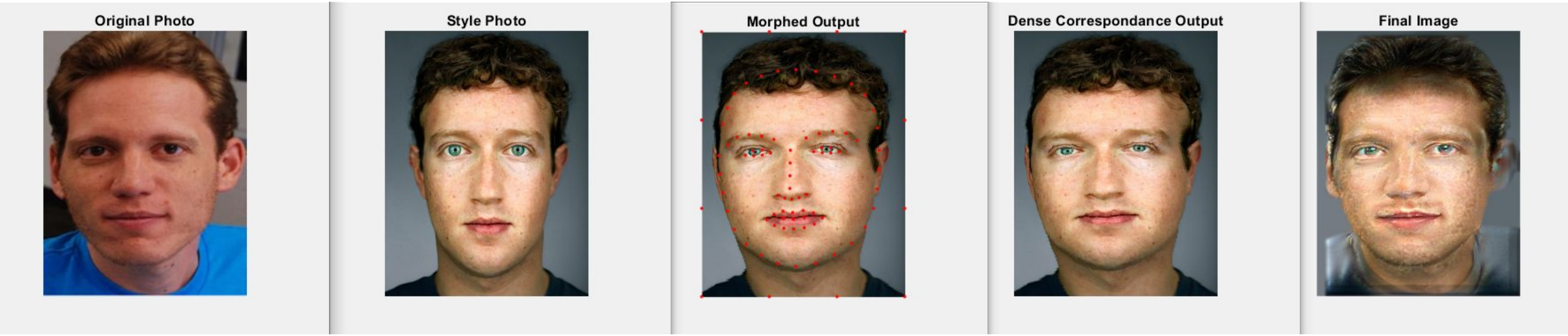
Result 6:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

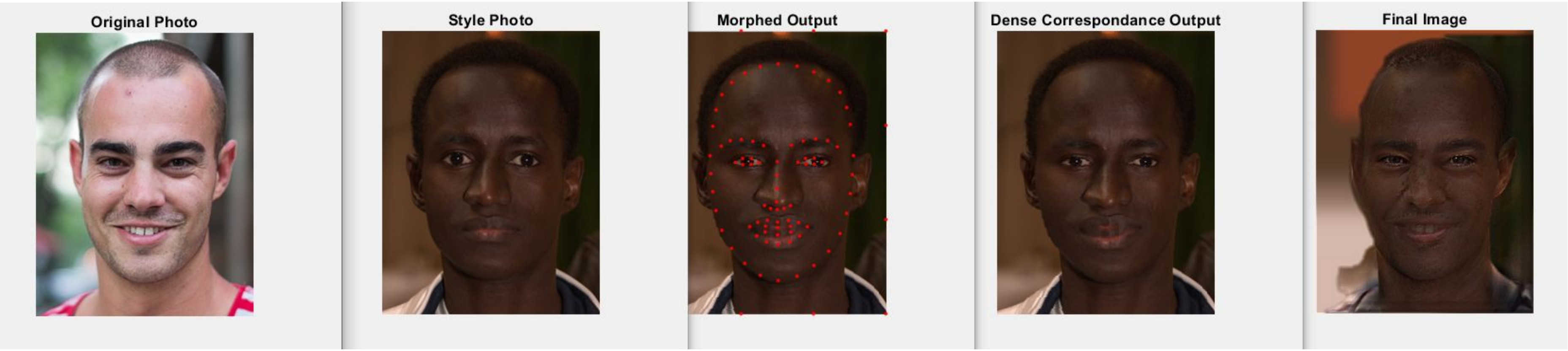
Result 7:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

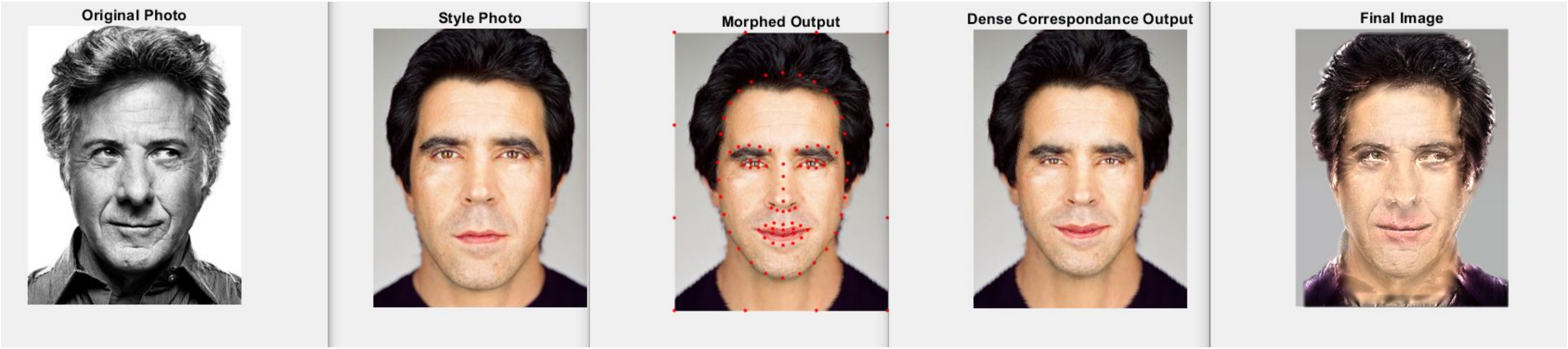
Result 8:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

Result 9:

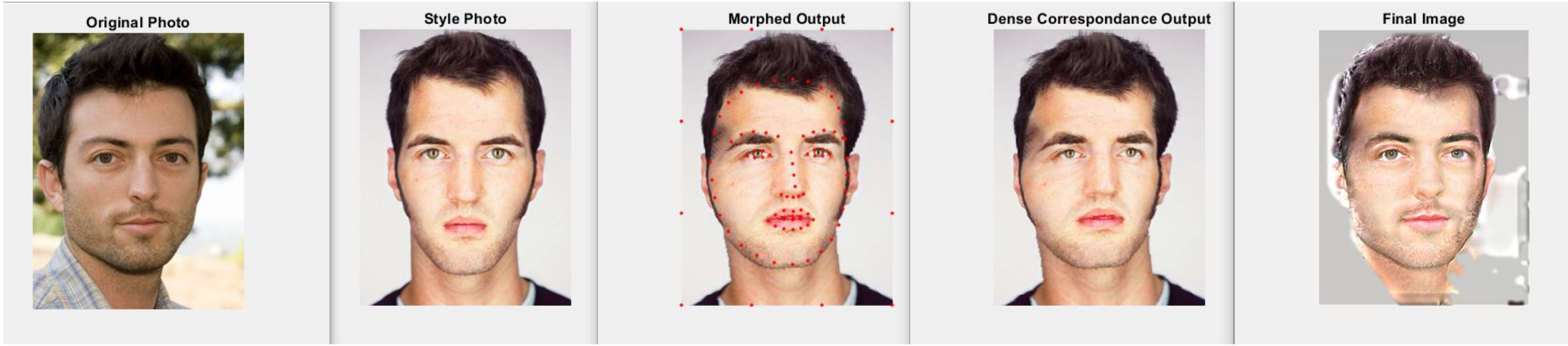

Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

Result 10:



Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

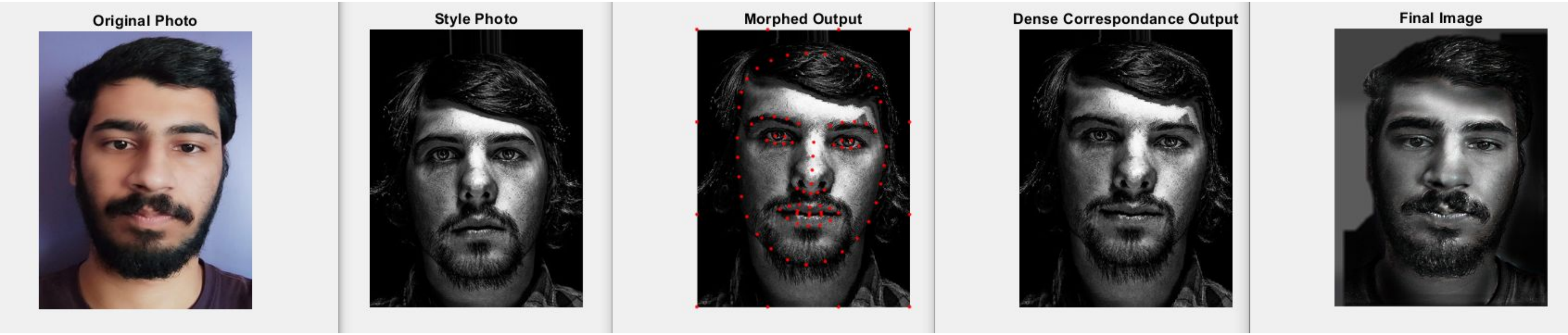
Result 11:



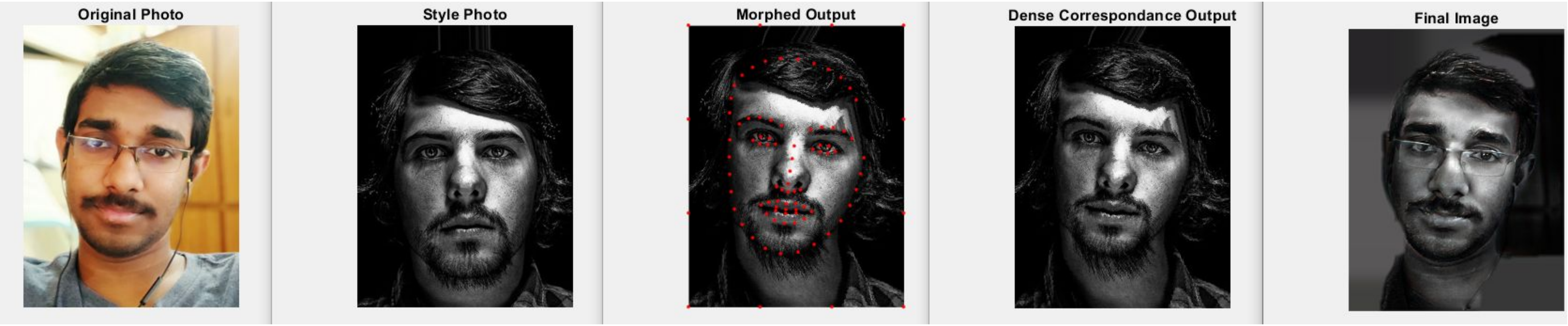
- Reason: Improper Mask

RESULTS (Our Data):

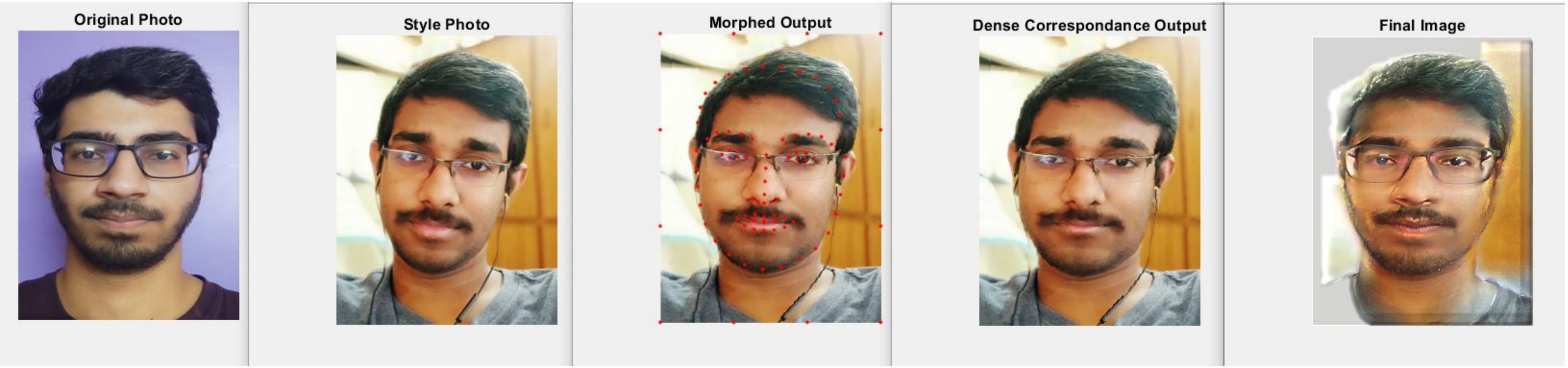
Result 1:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

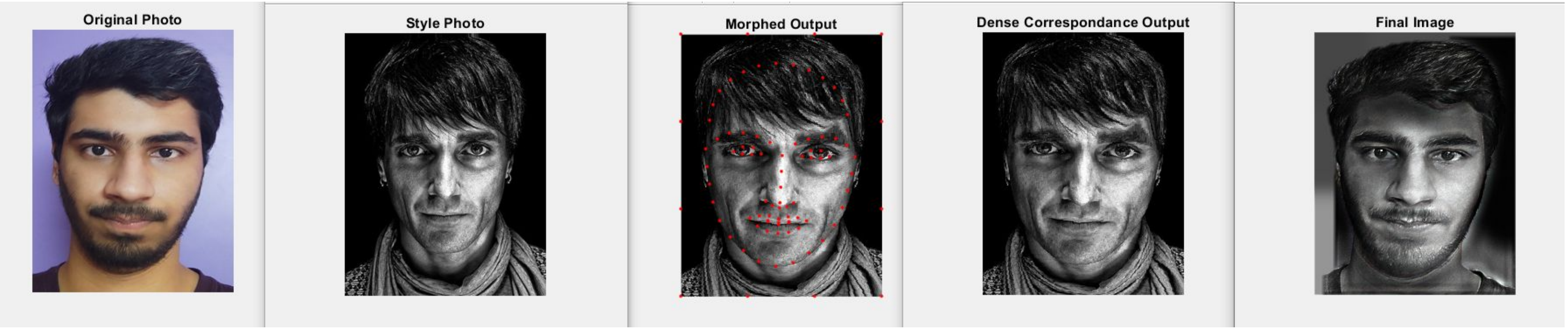
Result 2:


Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

Result 3:

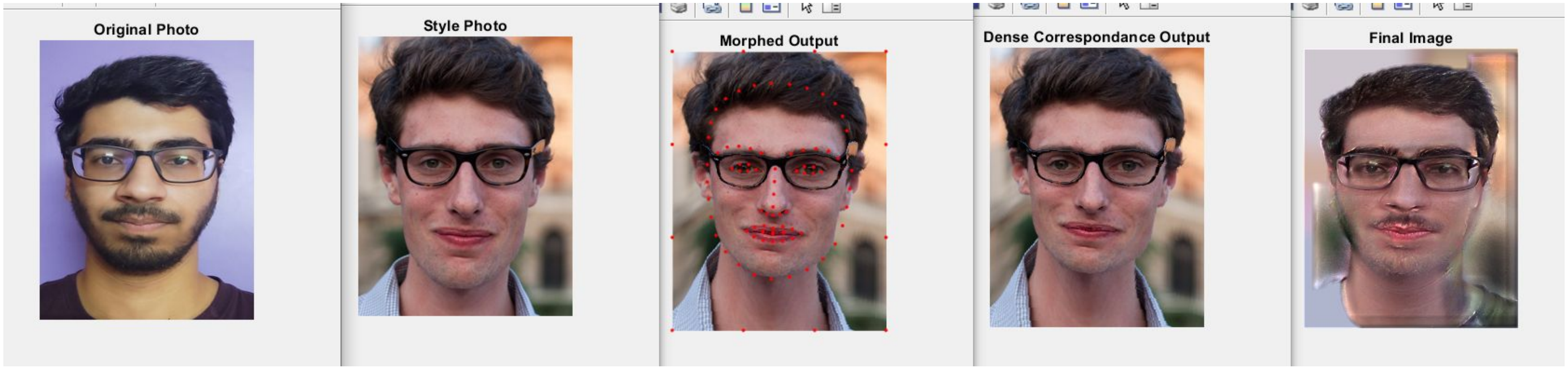

Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

Result 4:



Original Photo
Style Photo
Morphed Output
Dense Correspondance Output
Final Image

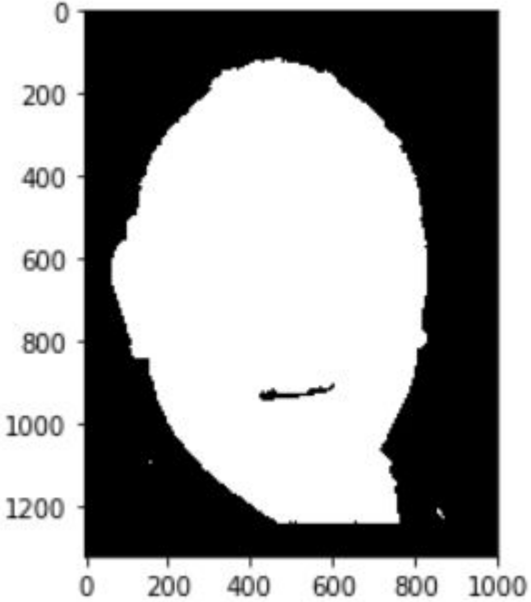
Result 5:



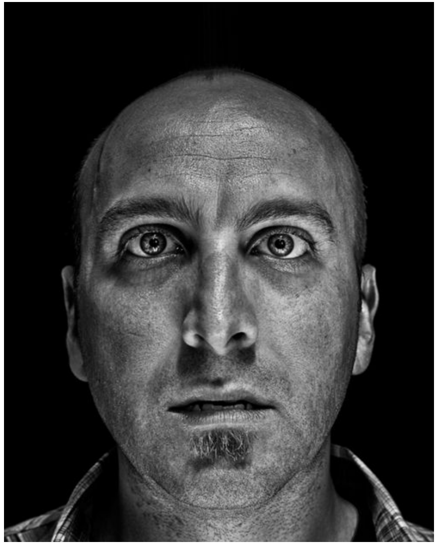
- Reason: Mismatching Features.(Here beard)

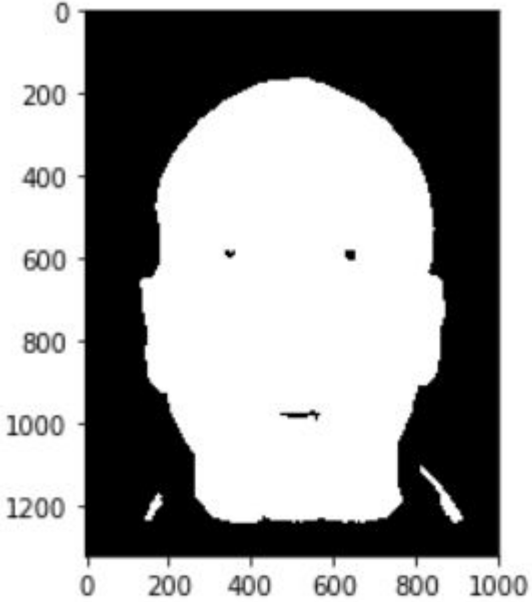

*Styled image
*Sasi kiran
*Mismatching
features

Grabcut mask results:



Grabcut mask results:





Work Division

Mask Generation	Sasi Kiran, Sai Krishna Charan
Dense correspondence	Sai Krishna Charan, Susheel
Laplacian Stacks	Harsha, Susheel
Local Energy	Sasi Kiran, Sai Krishna Charan
Gain Computation	Harsha, Susheel
Transfer of Gain	Sasi Kiran, Harsha
Reconstruction	Sasi Kiran, Sai Krishna Charan

Thank You