Style Transfer for Headshot Portraits

https://github.com/Einsteino/HeadshotStyleTransfer/

Overview

Introduction

Quite often, it is not feasible to capture studio-like quality headshot portraits, since we might not have the required lighting conditions, studio setup, etc. By transferring local statistics, we can transfer these feature-specific lighting styles and retouching effects onto the input image from a selected example image.

In this manner, we would be able to achieve studio-quality portraits from photos captured by a casual photographer.

Problem Statement

Given a subject portrait (input image) and a professionally taken headshot portrait (example image), match the appearance/style of the subject to that of the example.

Problem Statement

Input Image

Example Image

Proposed Approach

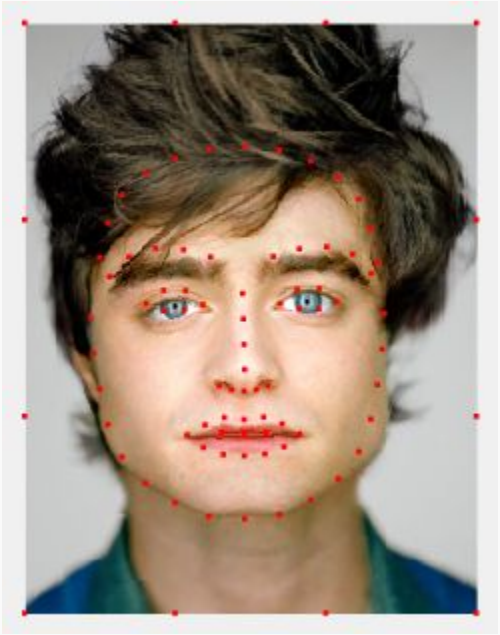
- Facial feature points detection
- Triangular morphing using feature points
- Grabcut foreground/background segmentation
- Pyramid based energy transfer for foreground
- Replace background from style image
- Finishing touches

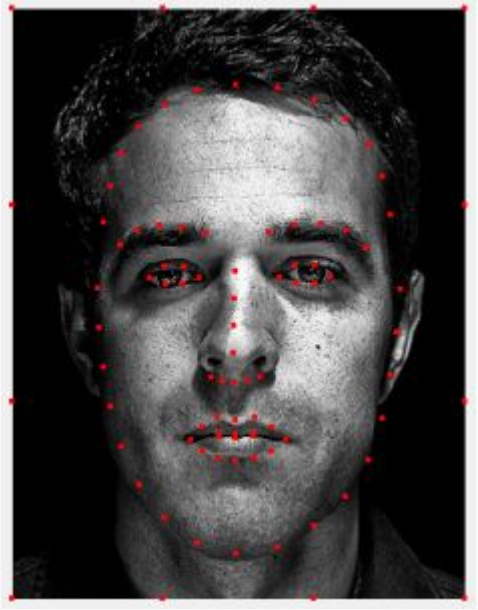
Workflow specifics

Facial feature points detection

- Detect facial landmarks using facial templates.
- Using aasthana facial landmark detection gives us a set of 66 feature points on the face. These points cover the eyes, nose, mouth and the jawline/chin.
- Extend jawline features by mirroring and compressing them, to get an approximate forehead boundary line.
- Add feature points along the borders of the image to ensure entire image gets morphed.

Facial Feature points





Triangular morphing using feature points

- Find the Delaunay triangulation of the given set of feature points.
- Morph the Example (style) image to the input image, using the triangulation.
- This ensures that the chin, jawline, eyes, mouth and the nose are aligned perfectly.

Morphing of example images

Grabcut foreground/background segmentation

- Use grabcut algorithm from Rother et al.[2004], on the Input image and the morphed example image, to segment out the backgrounds from their respective foregrounds.
- The foregrounds would include the head, neck, shoulders.
- Create masks according to this segmentation, and store the morphed Example’s background.

Grabcut foreground/background segmentation

Pyramid based image transfer - 1

- We first construct a laplacian pyramid of the image, except that we don't downsample the image as we proceed through the levels.
- Energy of a given image is defined as the square of the image convolved with a gaussian filter => S[l] = conv(L^2[l], G(l+1)) where l is the current level and G(l+1) is the gaussian at the next level.

Pyramid based image transfer - 2

- At each level we perform the following steps.
  - Calculate the energies of both the images.
  - We then calculate the gain between the energy maps of both the morphed example image and the input image.
  - This gain is nothing but ⇒ sqrt(energy(example) / (energy(input) + ε))
  - We set the output image at this level to be Input image at current level multiplied with the gain ⇒ L[out] = L[content] * gain for all levels except the residual where we set L[out] = L[example].

Pyramid based image transfer - 3

- At times there are issues with the gain maps, so we clamp it's limits between 0.1 and 2.9 and then apply a gaussian averaging to smoothen the map.
- Once we have the output images at each of the layers we will merge them to form our final image.
- We use the CIE-Lab color space to work with each channel independently in all the steps above. For each of the images we assume them to be a double representation (0.0 to 1.0 range for each pixel).

Pyramid based image transfer - 3 (No background)

Background from example image

- We take the background from the mask we computed earlier (on the morphed Example image), and extrapolate to find what the background at other points in the image would be.
- We then merge this background to the segmented foreground of the Input image.

Transferring the background from the example image

Finishing touches - 1

- Apply high boost filtering to ensure details are boosted.
  - This is optional and could be done to ensure images are sharp.
  - For certain kinds of styles this gives great results but for others it's not really necessary and also it ends up highlighting any artefacts that pop up in the results which is something we don’t want.

Finishing Touches sample - 1

Finishing Touches sample - 2

Related Resources - 1

1. Face Alignment through Subspace Constrained Mean-Shifts
2. Personal Photo Enhancement Using Example Images
3. Feature-Based Image Metamorphosis
4. SIFT Flow: Dense Correspondence across Scenes and its Applications
5. Preattentive texture discrimination with early vision mechanisms

Related Resources - 2

1. “GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts
2. A Closed Form Solution to Natural Image Matting
3. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing

Code Components Used

- Facial Feature points, morphing - http://ibug.doc.ic.ac.uk/resources/drmf-matlab-code-cvpr-2013/, https://github.com/GabriellaQiong/Face-Morphing/
- Grabcut - https://github.com/xiumingzhang/grabcut

Final Results

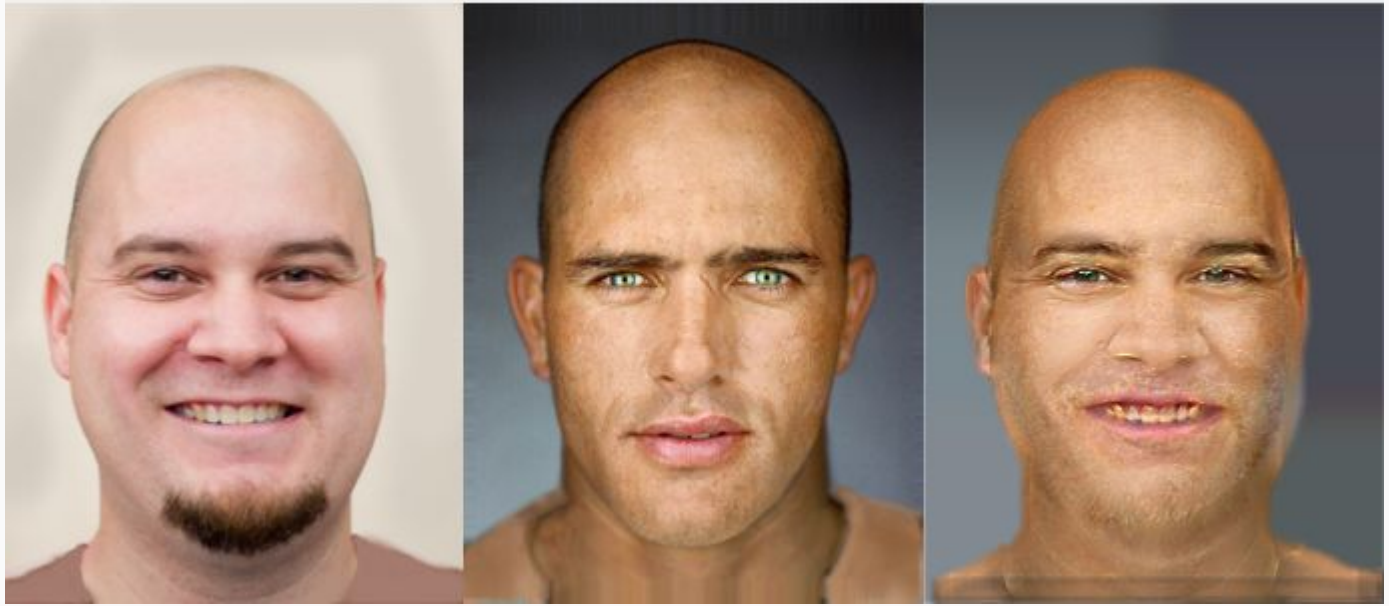
Result - 1

Result - 2

Result - 3

Result - 4

Result - 5



Result - 6

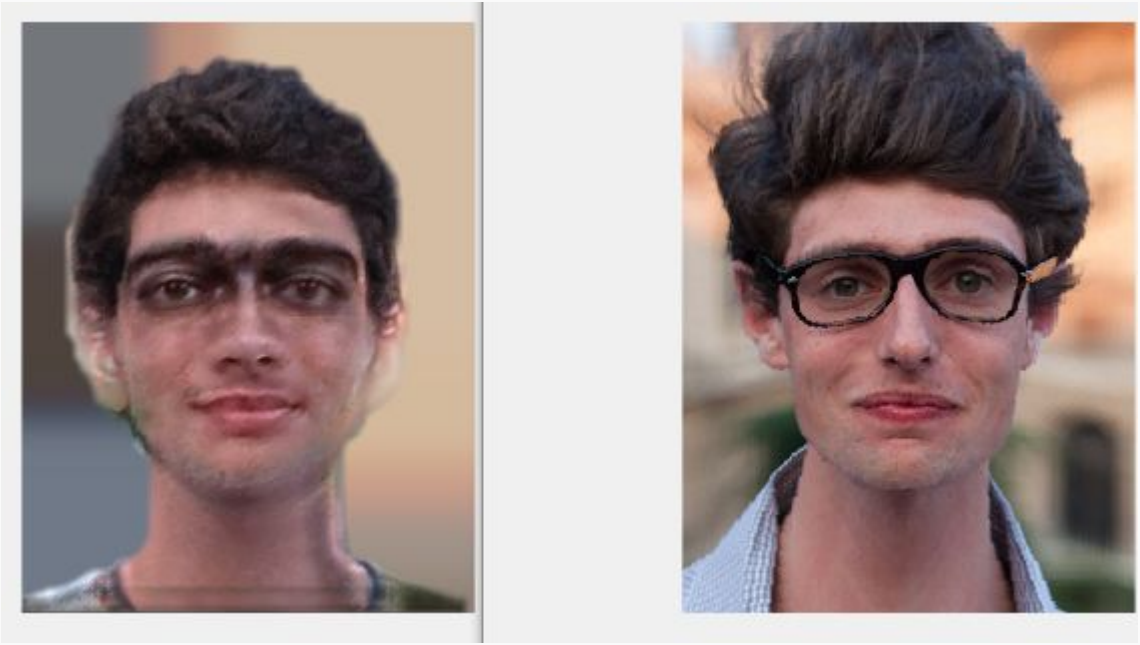
Result - 7

Result - 8

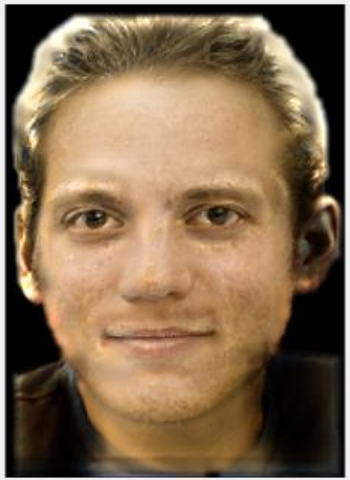
Result - 9

Failure cases

Issues with Glasses

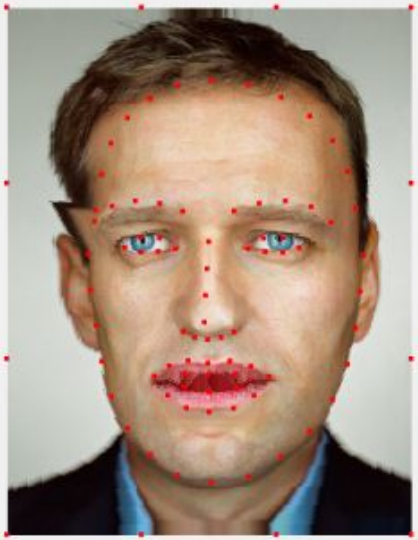


Incorrect hair and ear morphing

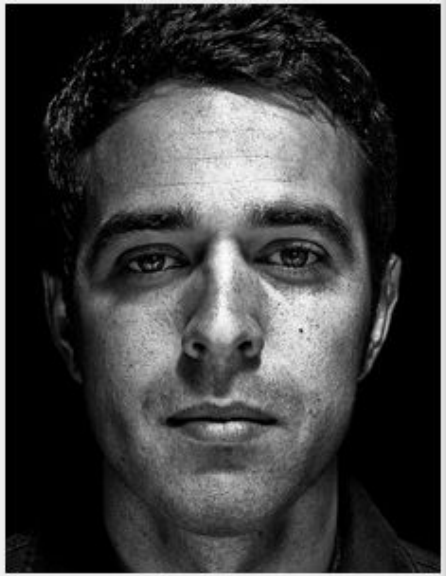


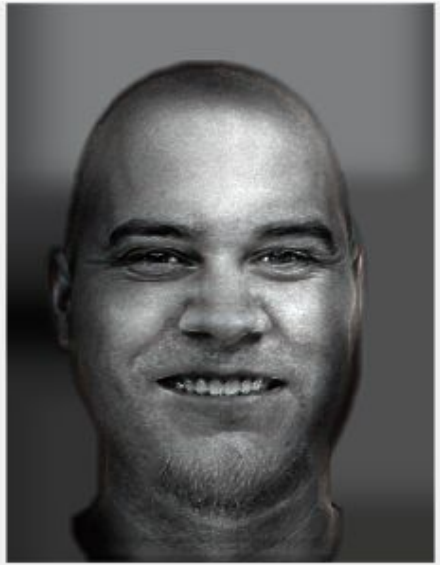
Large Illumination Changes

Incorrect Morphing



Grab-cut T-Shirt Selection





Further extensions to the project - 1

- Adding feature points - Once we have segmented output from Grabcut, we can add feature points along the boundary, for better alignment, especially of the hair/ears. This would prevent failure cases from arising, where ears are covered/ears are pulled back/hair is absent.
- In-Painting of Background - Instead of using max, mean filters to propagate background to blank parts of segmented image, we can use in-painting to automatically fill in blank regions with the nearest match, for better results.

Further extensions to the project - 2

- Extension to videos and real-time: Re-use segmentation results computing flows near the boundaries from frame to frame, and rather than transfer the style from example to input frame every frame, propagate style from frame to frame, and ‘reset’ by transferring from the example every few frames.

Team

Vishal Batchu

Sri Aurobindo Munagala

Thank you!