Repo URL :

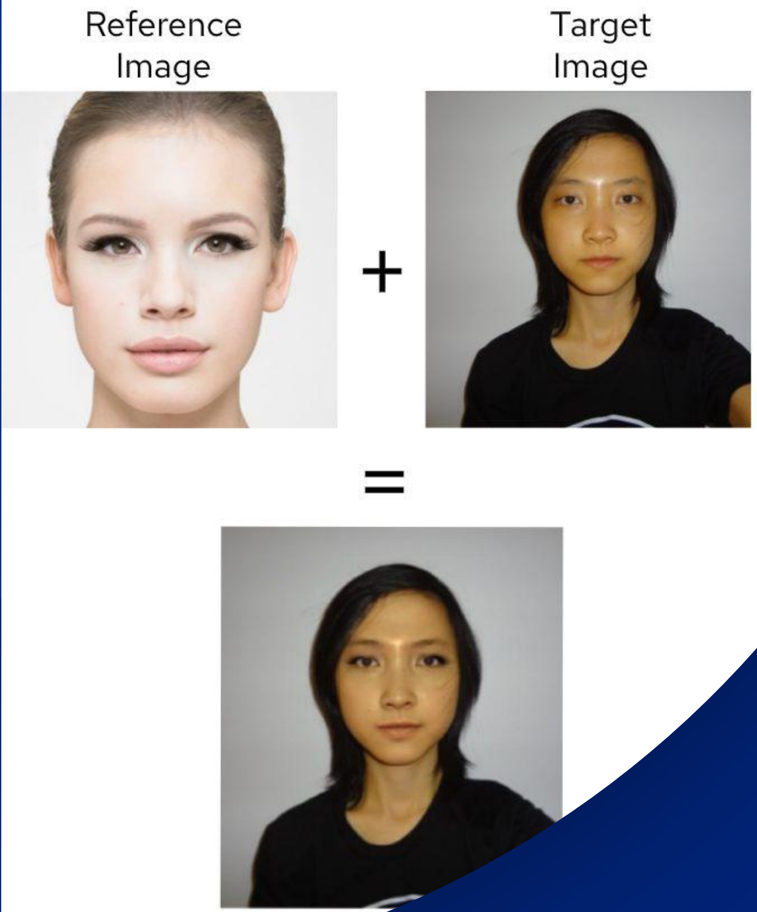
https://github.com/Digital-Image-Processing-IIITH/project-adobe-lite-room

Digital Makeup Face Generation

Team Adobe Literoom

TA - Surendra Kumar Reddy

Shaunak Badani 20171004
Manav Bhatia 2018102009
Shantanu Agarwal 2018102040
MD Kalesha 2018102047



Aim

Transfer makeup from a particular reference image to a person (target image).

How do we achieve this ?

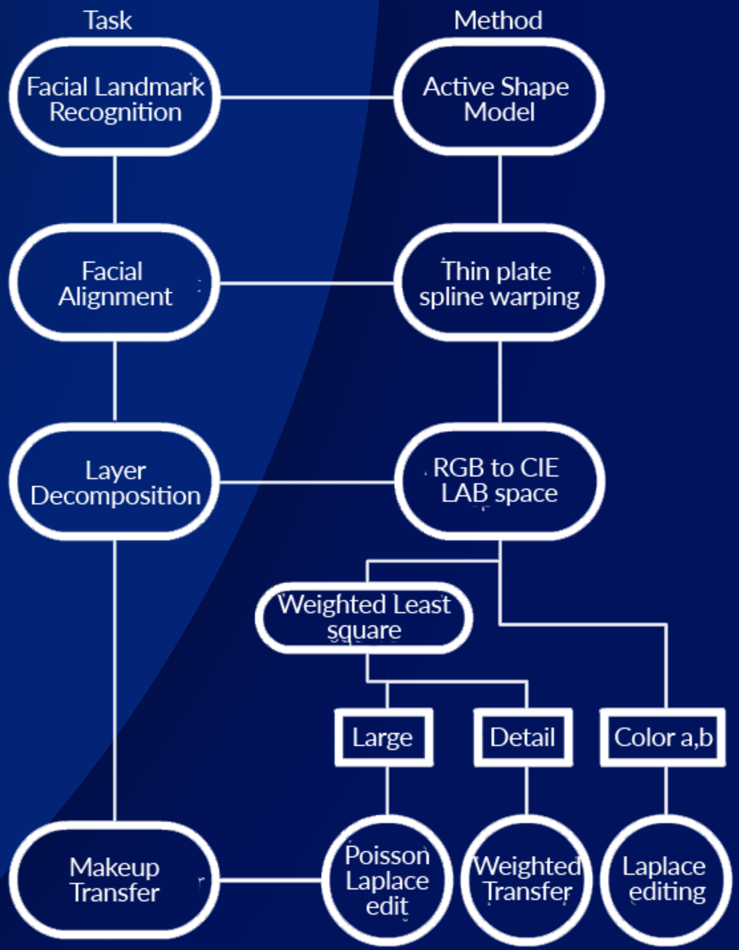
1. Recognize important facial landmarks in both pictures, e.g. boundary of nose, boundary of eye.

2. Warp the reference image wrt these points, such that these points are at the same position when both images are superimposed.

3. Cut eyes and lips from reference. Decompose the makeup warped image into lightness and color layers.

4. Combine these layers in some form to get the desired output.

A more formal explanation



Step 1: Facial Landmark Recognition

Shape

A **N x 2** vector, representing points on image containing high curvature or are distinct corners.

Active Shape Model

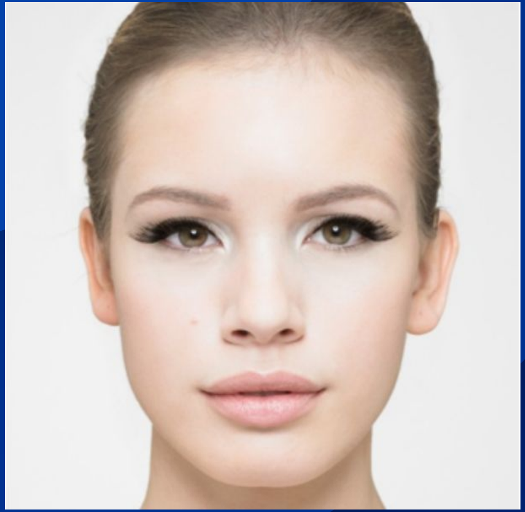
A model containing pre-trained data of faces which on input, an image representing some face, returns the shape.

Advantage

Generates face contours very effectively, eliminating the need of face segmentation.

Step 1 : Facial Landmark Recognition

Image



Shape

[[127. 276.]
 [133. 346.]
 [145. 406.]
 [164. 464.]
 [199. 521.]
 [249. 564.]
 [302. 576.]
 [355. 566.]
…….]

Image with landmarks

Step 2: Facial Alignment

Problem statement

Aligning the two images w.r.t shapes (landmarks).

Arsenal

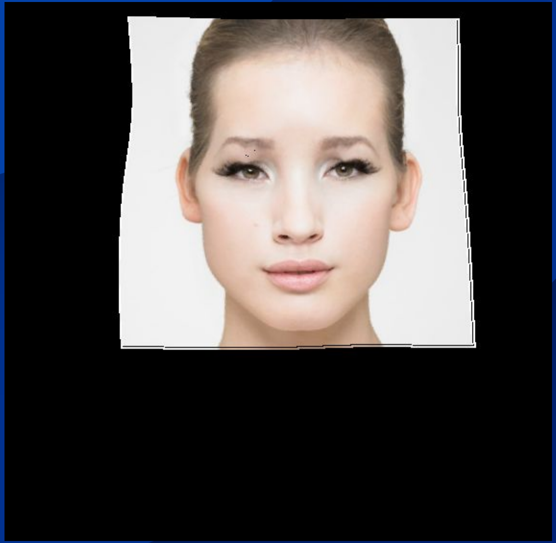
A warping method called thin plate spline warping, which uses the inverse distance weighted interpolation method to warp the reference w.r.t target

Implementation

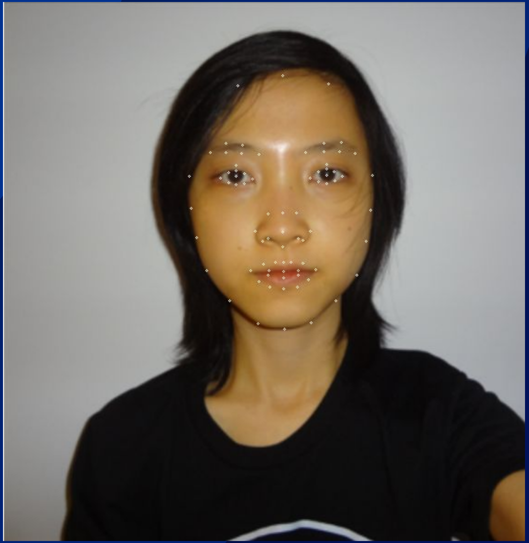
Writing python code for the same.

Step 2: Facial Alignment

Warped image wrt reference landmarks



Target (dotted)



Superimposed image

Step 3.1: Preprocessing Images

Cutting and Masking

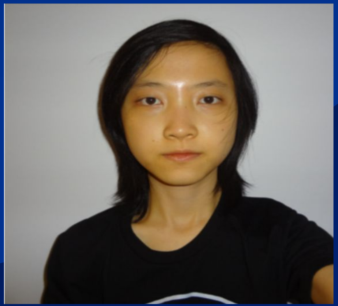
Since the images are aligned, we now used the points generated in step 1 and cut-off the eyes and the lips of the image.

Face Cutting and Background Masking

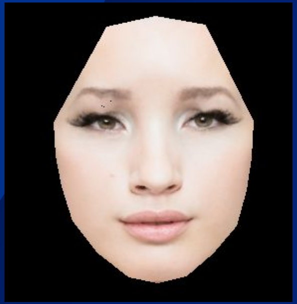
We used the convex hull of the boundary points of the face image and generated a binary mask to mask the background and extract the facial features, sans the eyes and lips.

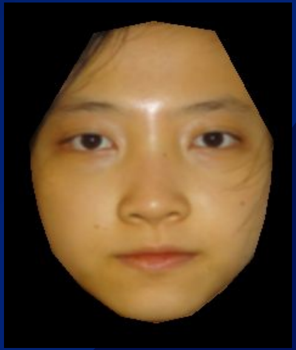
Preprocessing Images

Warped image wrt reference landmarks



Face cut out





Eyes and Lips removed

Step 3.2 : Layer decomposition

First step

Convert warped image obtained into Lightness and color (a, b) layers.

This can be easily achieved using cv2's inbuilt function, cv2.cvtColor(img,cv2.COLOR_RGB2LAB).

Second step

Further decompose the lightness layer into large scale layer and a detail layer.

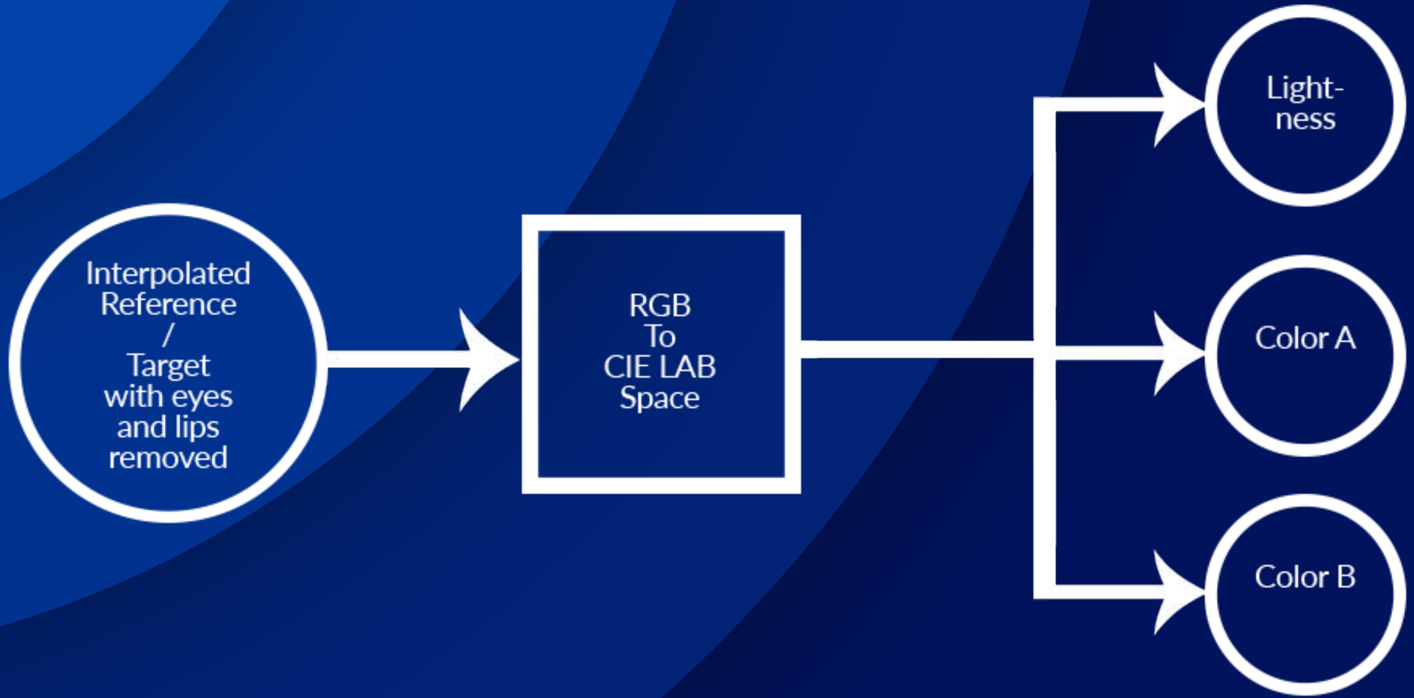
$$s + d = L$$

$$E = |L - s|^2 + H(\nabla s, \nabla l)$$

$$H(\nabla s, \nabla l) = \sum_p \lambda(p)\left(\frac{(\frac{\delta s}{\delta x})_p^2}{(\frac{\delta l}{\delta x})_p^\alpha + \epsilon} + \frac{(\frac{\delta s}{\delta y})_p^2}{(\frac{\delta l}{\delta y})_p^\alpha + \epsilon}\right)$$

Symbol	Meaning
s	Large Scale Layer
d	Detail Layer
E	Function to be minimized

RGB2LAB


Interpolated Reference / Target with eyes and lips removed
RGB To CIE LAB Space
Light-ness
Color A
Color B

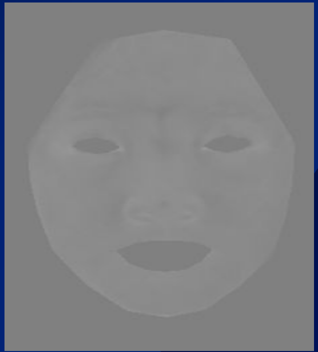
Results of the Conversion to LAB Space

REFERENCE IMAGE
LIGHTNESS LAYER

REFERENCE IMAGE COLOR
LAYER A

REFERENCE IMAGE COLOR
LAYER B

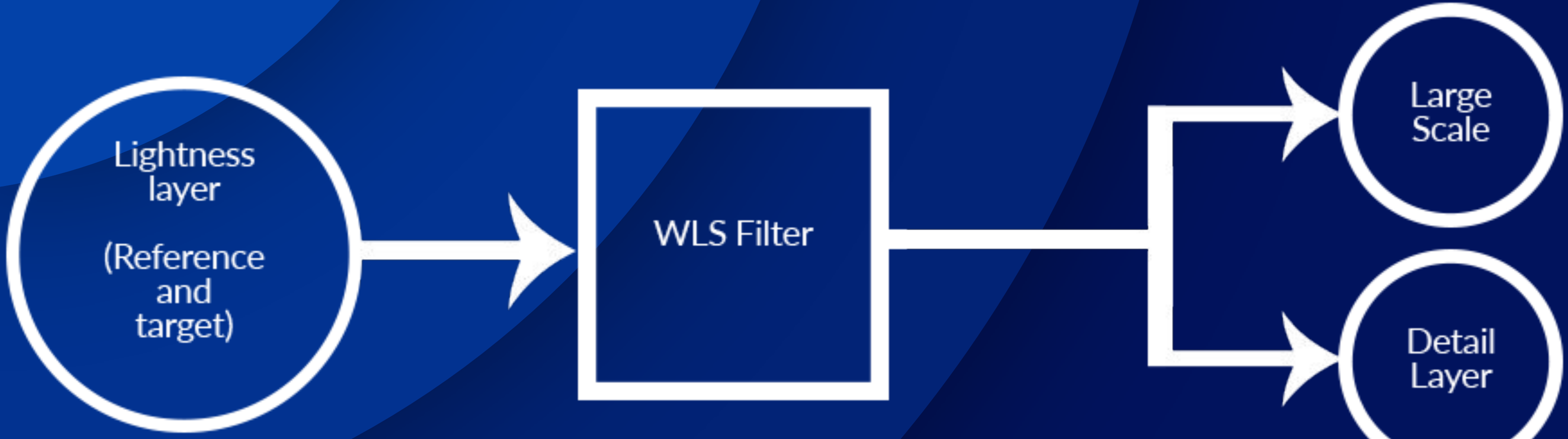
TARGET IMAGE
LIGHTNESS LAYER



TARGET IMAGE COLOR
LAYER A

TARGET IMAGE COLOR
LAYER B

WLS Filter at a glance



Results of the WLS Filter

Lightness Layer | **Large Scale Layer** | **Detail Layer**






Step 4 : Makeup transfer

Large scale layer

Contains the smoothed face image with only highlights and contours.

Detail layer

Contains moles, skin texture, wrinkles and etc.

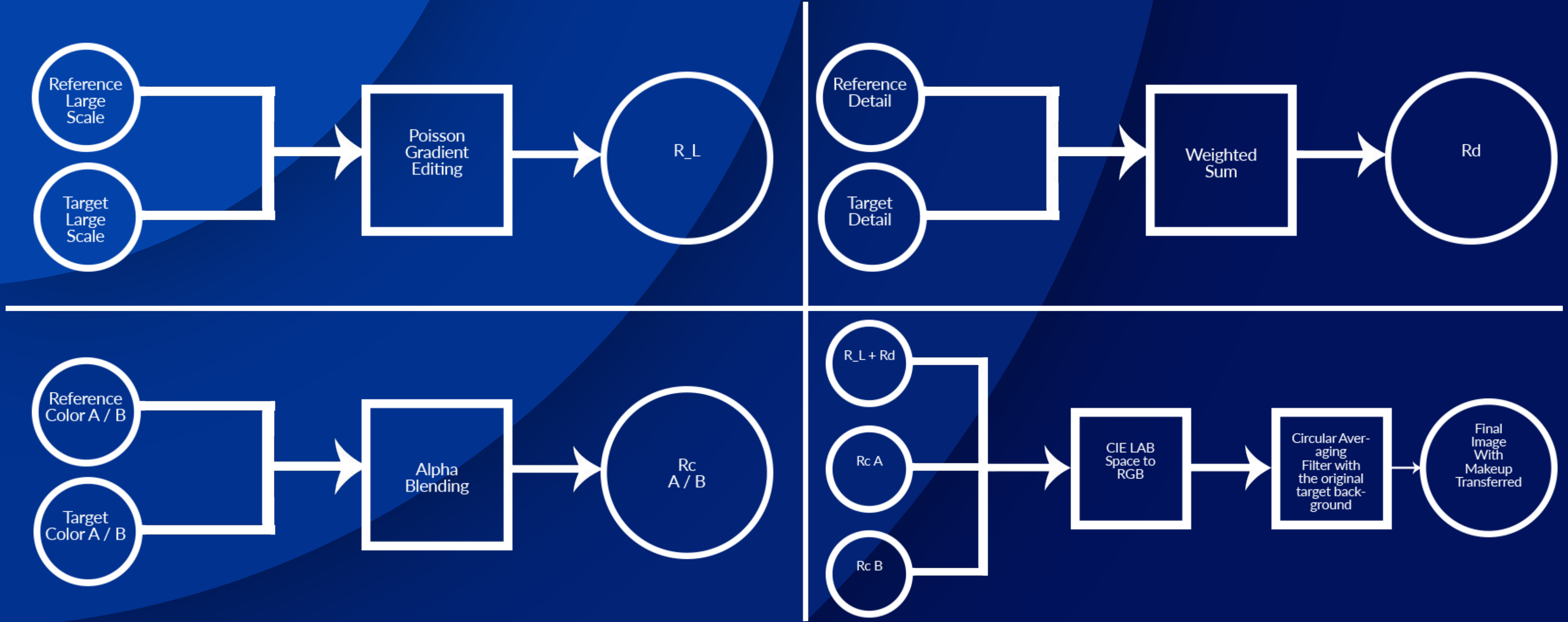
Color a,b layer

Apply laplace editing

Finally

Combine all the images using circular averaging filter.

Step 4 : Makeup transfer



Highlight Transfer /Resultant Large Scale Layer

Reference Large Scale

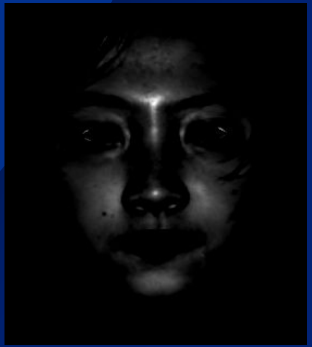
Target Large Scale

Resultant Large Scale (R_L)

Details Transfer/Weighted Sum

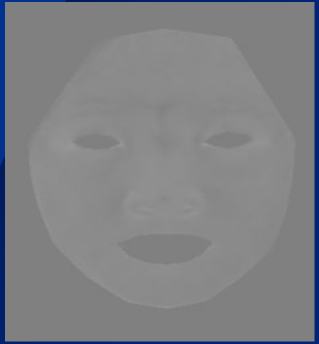
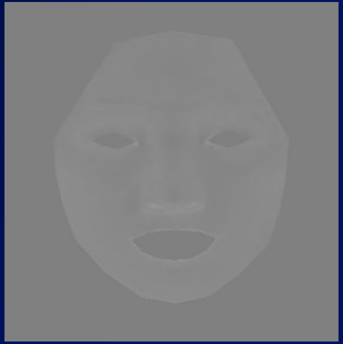
Reference Image Detail Layer

Target Image Detail Layer



Resultant Detail Layer

Color Transfer/ Alpha Blending

	Reference Image	Target Image	Resultant Color Layers
Color Layer A			
Color Layer B			

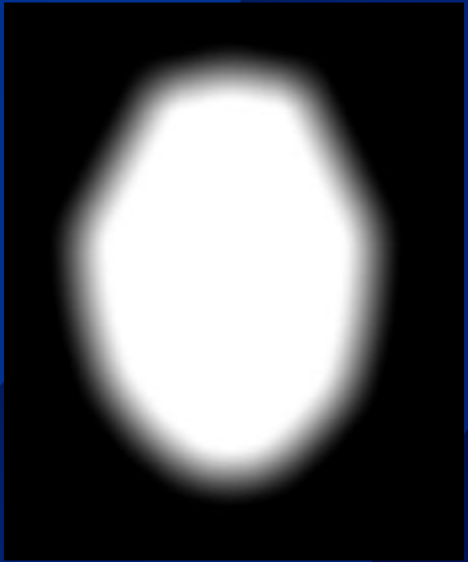
Blending/ Final Stitching

LAB Layers

Conversion to RGB Color Space

Blending/ Final Stitching

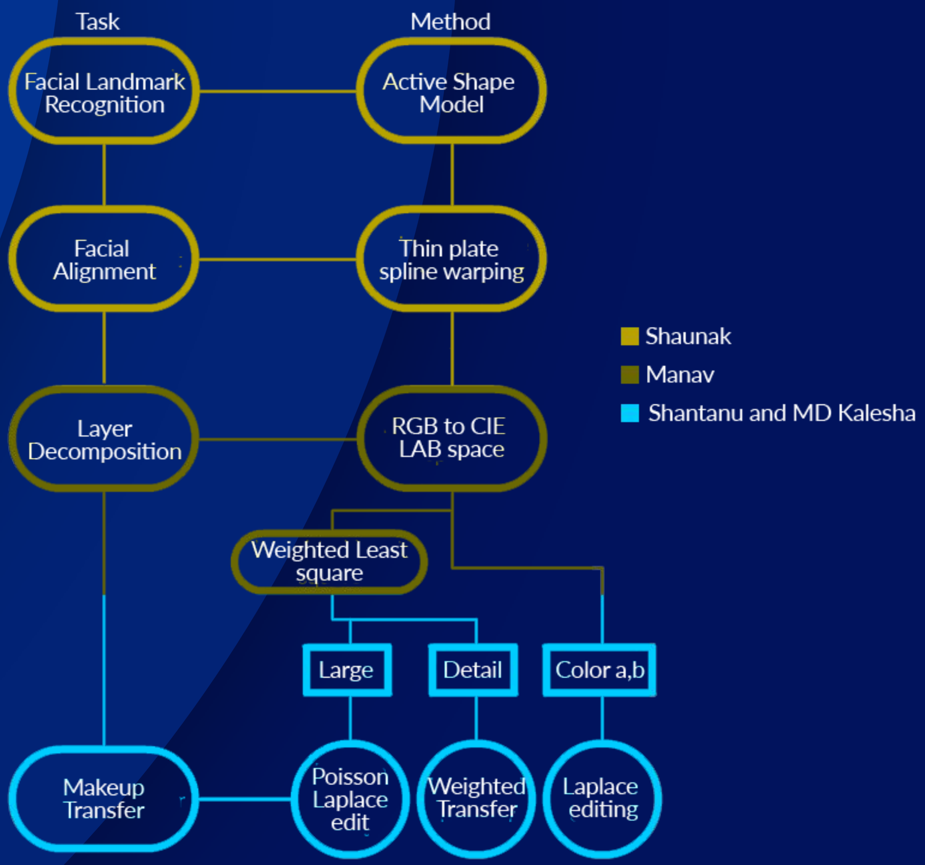
Circular averaging filtered binary mask with a 'disk' radius of 15



Blending/ Final Stitching

Final addition using similarly processed eyes and lips layer

Individual contributions



Appendix

Where the theoretical tuff i

Highlight Transfer /Resultant Large Scale Layer

- Face structure/Large scale layer contains **identity information**, we can neither directly **copy Source large scale layer over Target large scale layer nor blend them. Instead, we adapt a gradient-based editing method**.
- The idea is to add only **large changes of Source large scale layer to Target large scale layer as we assume that these changes are due to makeup**.
- So we used **Poisson image editing** method and the equation is solved by **Gauss-Seidel method**.

Poisson Image editing

Notation for the equation:-

- p is the pixel location,q is the location of one the 4-neighbours of p.
- N_p Is the set of the 4-neighbours of p
- Ω Is the area of interest in the source image.
- $\delta\Omega$ Is the boundary of the area of the interest.
- f_p^*, f_p Is the pixel value at location p from target and is the pixel value at the resultant image respectively.
- $\sum_{q \in N_p} v_{pq} = \begin{cases} \nabla f^*(p) & |\nabla f^*(p)| > |\nabla g(p)| \\ \nabla g(p) & otherwise \end{cases}$

Mixing gradients and here assigned element is the laplacian filter applied at p.

Poisson Image editing

The cost function is:-

$$\min_{f|\Omega} \sum_{<p,q>\cap \neq 0} (f_p - f_q - v_p)^2, with f_p = f_p^*, \text{for all p} \in \delta\Omega$$

Its solution satisfies the following simultaneous linear equations:-

For pixels which has boundary pixels as their Neighbours

For all p $\in \Omega, \|N_p\| f_p - \sum_{q \in N_p - \delta\Omega} f_q = \sum_{q \in N_p \cap \delta\Omega} f_q^* + \sum_{q \in N_p} f_q$

For pixels which has no boundary pixels as their Neighbours

$$|N_p| f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq}$$

Questions?

We’ll answer them, but hesitantly.