Creating Face Caricatures

Anubhav Mittal, Vivek Singh

EE604 course project

Supervised by

K S Venkatesh, Department of Electrical Engineering
Govind Sharma, Department of Electrical Engineering

November 16, 2019

Caricatures

- A caricature is a rendered image showing the features of its subject in a simplified or exaggerated way.
- Facial caricature is an art form of drawing faces in an exaggerated way to convey humor or sarcasm.

Our Approach

In this project, we focuses on specific characteristics of the face to create the face caricature. The whole process is divided into 2 parts

- Cartoonifying the image
- Distortion of the image

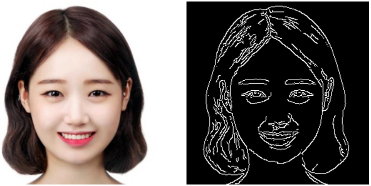
Cartoonifying of the image

Two main characteristics of a cartoon:

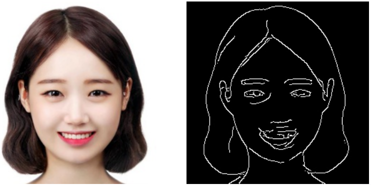
- Really clear edges
- Homogenity in colour

Detecting Edges

- First, we tried using Canny edge detector which resulted in getting to much edges.

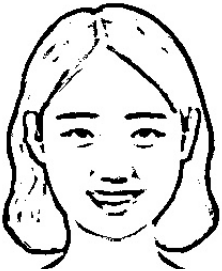


- Then, we tried to blur the image using 9x9 median pooling operation, we got better images but the lines were too thin to use for practice. So this method was discarded.



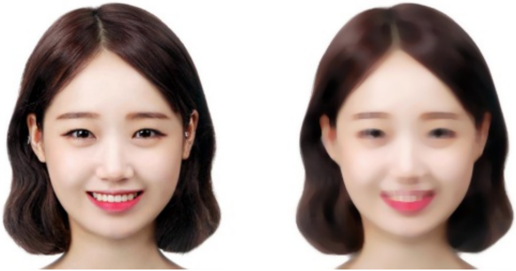
Detecting Edges cont.

- Then, we used adaptive thresholding. The algorithm determines the threshold for a pixel based on a small region around it. So we get different thresholds for different regions of the same image which gives better results for images with varying illumination. Again, blurring helps to reduce the no. of edges. We get much thicker edges with the ability to control the edges too, by changing the size of the region.



Homogenity on colours

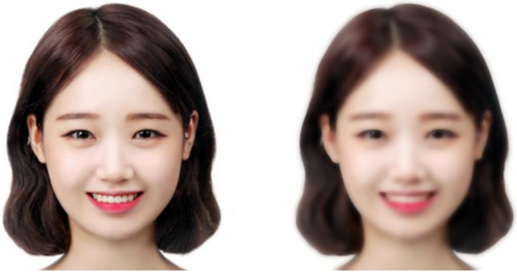
- We start with the most intuitive method - median pooling operation. We obtain the result as shown below:



We don't want to mix colors across different regions, in which this approach fails.

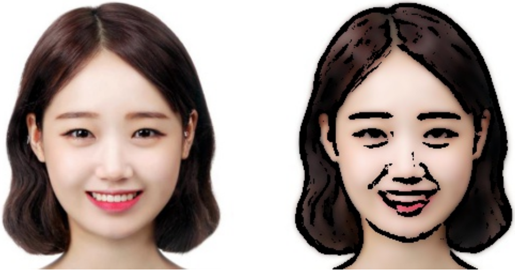
Homogenity on colours

- To overcome this problem, we used bilateral filtering. It is highly effective in noise removal while keeping edges sharp. Bilateral filter takes 2 gaussian filters: one in space, and the other which is a function of pixel difference. Gaussian function of space make sure only nearby pixels are considered for blurring while gaussian function of intensity difference make sure only those pixels with similar intensity to central pixel is considered for blurring. So it preserves the edges since pixels at edges will have large intensity variation.



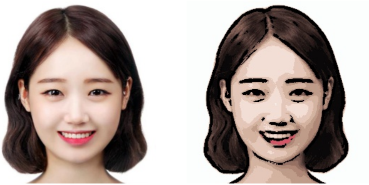
Final Cartoonified Image

- We then combine the edges and bilateral filtered image to get the final cartoonified image.



K means method for cartoonification

- We use colour quantization - reducing the number of distinct colors in an image. The intent is to preserve the color appearance of the image as much as possible, while reducing the number of colors.
- For doing this, we use K means on the LAB conversion of the RGB image. In the LAB color space, the euclidean distance between colors has actual perceptual meaning this is not the case for the RGB color space. Given that k-means clustering also assumes a euclidean space, were better off using LAB rather than RGB.
- We try for different values of k from 2 to 32. The following conversion was obtained for k=16.



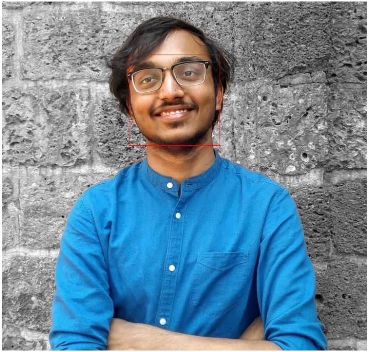
Distortion of the Image

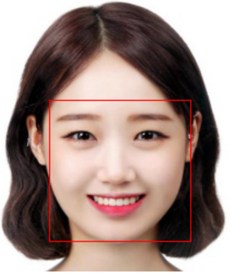
The Distortion process comprises of 2 steps

- Face detection and pose detection. We then use pose to define specific points on the face.
- Distortion in specific part of the face.

Face detection

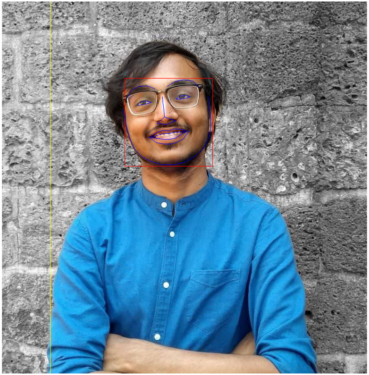
- In this process, first we take an image (.jpg) as input and then use dlib face detection to find frontal human face in an image.
- This face detector is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme.

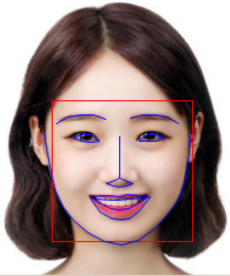




Pose detection

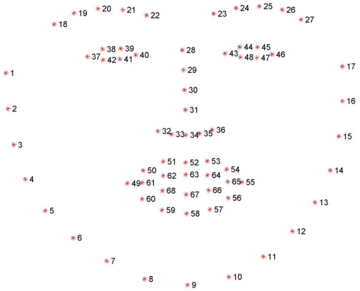
- The pose estimator was created by using dlib's implementation of the paper: One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014 and was trained on the iBUG 300-W face landmark dataset.
- Parameters are defined according to the landmark points. Some points collectively define a specific part of the face.

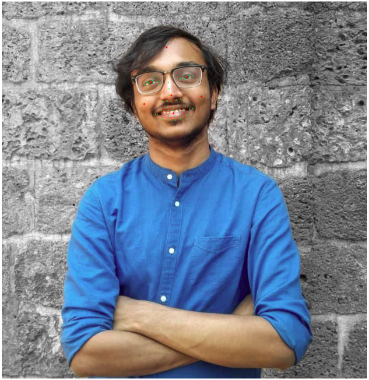




Parameter defining

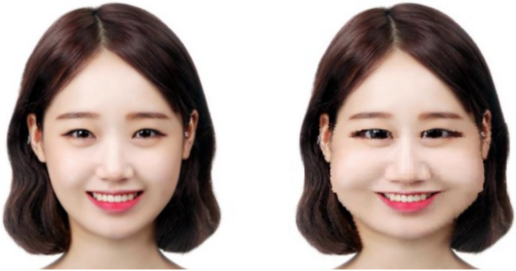
- Pose parameters in terms of the points is shown in the left figure.
- Using these points, we find the location of specific parts of the face, like nose, left/right cheek, eyes, etc. In the right image, red points denote specific parts of the face we are going to distort:





Landmark Distortion

- In this process, the image is distorted by stretching the specific sections of the image. It uses the parameter function to change the shapes.
- In this project, we have distorted the different parts of the face in an sequential way, so the changes can be noticed explicitly. The degree of change can be tuned easily as well.



Combining the two approaches

- We finally pass the distorted image throught two cartoonifiers:

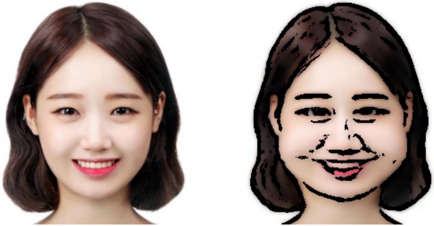


Figure: Using Adaptive thresholding + Bi-linear filter

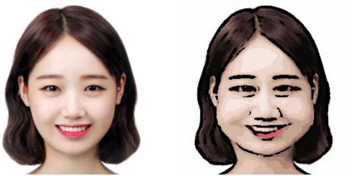


Figure: Using K means color quantization