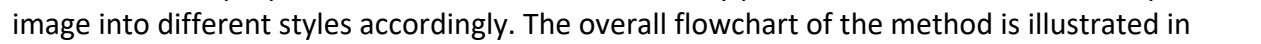


Final Project Report

Introduction

In this project, I explored using deep learning-based method to detect facial key points and then categorize human faces into different styles empirically.

Methods

In this section, I proposed a method to detect facial key points and then cartoonize the portrait image into different styles accordingly. The overall flowchart of the method is illustrated in . The four main stages are (1) face detection using OpenCV, (2) face landmarks detection based on FCN, (3) style categorization based on face morphology calculated using detected facial key points and (4) photo to cartoon transformation. In the first stage, I first decide whether a human face exists in an image. If yes, I extract the face region as region-of-interest (ROI). In the second stage, I transform the detection problem to a region segmentation problem and utilize FCN to obtain the result. In the third stage, I category the portrait into different classes based on face morphology information. Different cartoon styles will be suitable for faces within different categories. In the final stage, a well-developed photo to cartoon transform network is used.

2.1 Face recognition

Considering the difficulty of detecting facial landmarks directly from portrait images with different conditions, a face recognition is needed. The task of recognition is to detect whether a face exists in the image, and box the region of interest if there is a face. In this project, a detection method based on Haar feature-based cascade classifiers[1] is used. The features are calculated using all possible sizes and locations of each Haar convolutional kernel. In order to save computation time and resource, classifiers are cascade. For most of the image blocks that do not contain face information, the a few features are examined in the first few stages before achieving failure decisions.

In this project, I utilize the pre-trained cascade classifiers in OpenCV[2].

2.2 Face key points detection

Given the face region, I introduce a fixed detection stage based on the FCN to further detect facial key points.

2.2.1 Data Preparation

Facial landmarks are the representations of facial structures, including eyebrow, eye, nose, moth and others. In the whole set of key points, a few (usually 2 or 3) points are usually grouped together to illustrate the location and morphological shape of the facial components. In the research field of facial key points detection, there are many datasets with different definitions of landmarks. For example, 15, 21, or 68 landmarks are all used to represent one face. In this project, I use the facial key points dataset from Kaggle [3]. For each face, 15 points are included to demonstrate. For example, two points are used to define the left and right tip of eyebrow. One eye is represented using three points illustrating the left, right corner and the center of eye. One point is to mark the center of nasal. The rest four points are grouped

together to show the upper and lower lips. In total I have 15 annotated points for each face image.

In the Kaggle dataset, there are in total over 7000 training images with facial key points annotated. However, for a few training images, some facial landmarks are missing. In this study, I only use the images with full lists of key points. Since I don't have ground truth for the testing data in Kaggle, I do not use them. In this project, I only use the data with annotations which are available in Kaggle training folder and further split those data into training and validation datasets (428 samples). To tune the hyper parameters

In order to further enlarge the number of training images, and to increase the variety of face patterns, 2 data augmentation process is included: (1) random affine transform and (2) horizontal flip. In this section, I randomly generate a shift $(\Delta x, \Delta y)$ where $\Delta x, \Delta y$ are independent uniformly distributed between 0 and 5 pixels. Then each pixel in the transformed image is defined as

$$I_t(x, y) = I_0(x + \Delta x, y + \Delta y),$$

where I_0 represents the intensity of original image at a specific location. Each pixel in the horizontal flipped image is defined as

$$I_f(x, y) = I_0(96 - x, y).$$

Note that after coordinate of facial key points should also be transformed for the augmented images. The raw image (left), affine translated image (center) and flipped image (right) are illustrated in Figure 1.

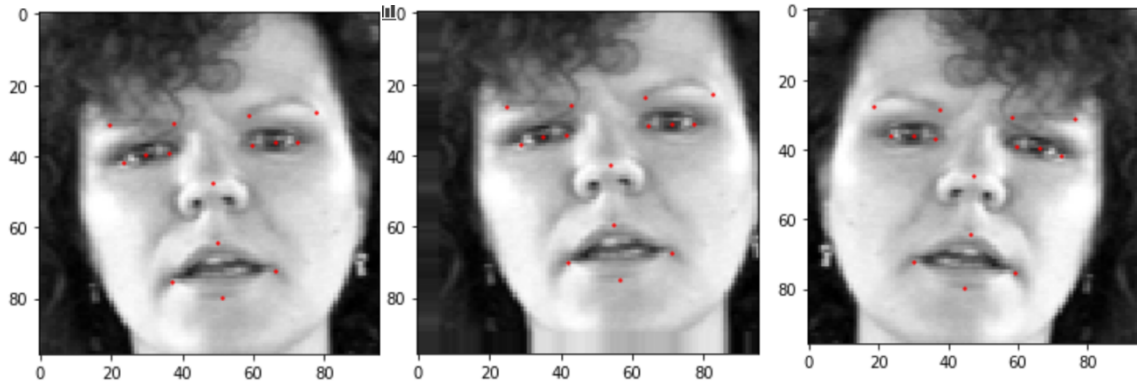


Figure 1 Raw input image (a), and affine translated image (b), and flipped image (c)

After data augmentation, in total 3852 samples are used for training. I further split the training set into train (3081 samples) and validation (771 samples). The hyper parameters are tuned on the validation sets.

Considering the difficulty of detecting a single point in the patch, I extend the label to the heatmap generated using 2D gaussian kernel. Suppose given the example landmark (x_0, y_0) , the weight values of heatmap at (x, y) is given by

$$(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x - x_0)^2 + (y - y_0)^2}{\sigma^2}\right],$$

where σ is the standard deviation of the gaussian kernel. In this project, σ is a hyper parameter that need to be tuned.

The point detection problem is then transformed to the segmentation task of the soft bordered circular region. Compared to the binary circular mask with a radius r , the weights introduced by the gaussian kernel contributes more to the center information. As a result, the effect of imbalanced labeling is decreased and additional neighborhood information is learned to distinguish the landmarks from other small structures.

2.2.2 Network architecture

The fully connected network (FCN) has been well developed and widely used in the domain of sematic segmentation. Thus, the detection stage in our method is based on a simplified version of FCN-8, as shown in Figure 2.

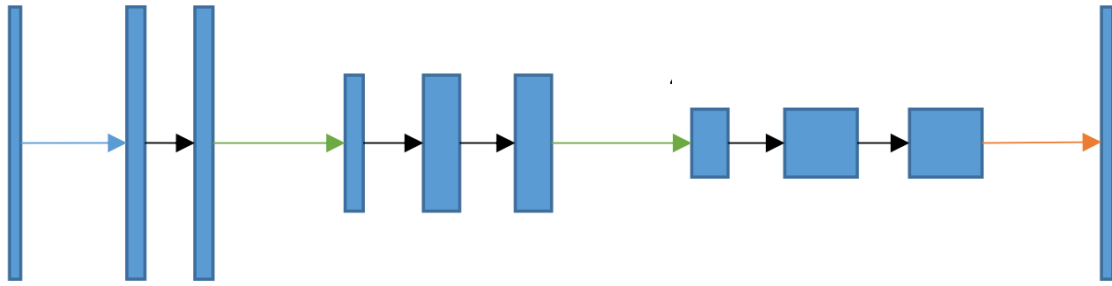


Figure 2 FCN architecture in the method

1. The blue blocks represent the arrangement of its neurons in three dimensions (width \times height \times channels), as visualized in one of the layers. Every layer of an FCN transforms the input tensors to an output volume of neuron activations.
2. The black arrows represent convolution operation with 3×3 kernels followed by the rectifier (RELU) activation function.
3. The green arrows represent down-sampling operation along the spatial dimensions (width, height) using the max operator. In this model, every max operation is taking a max over 4 numbers (little 2×2 region in some depth slice), discarding 75% of the activations. The depth dimension remains unchanged.
3. The red arrows represent up-sampling operation through transposed convolution operations with 3×3 filters.

In this project, I use the mean squared error as losses.

$$MSE = \frac{1}{n} \sum_x (P_x - H_x)^2,$$

where P_x is the predicted map and H_x is the mask heatmap. n is the total number of pixels in the mask.

A combination of soft dice and mean squared error has also been investigated but failed. I use the Adam optimizer with adaptive learning rate. The learning rate will be decreased according to the loss changes in training and validation loss.

2.2.3 Get coordinate from predicted heatmap

To extract the coordinate of predicted landmark from the heatmap, I calculate the weighted geometric center of largest n^2 pixels in the heatmap.

$$C_x = \sum_i P_i x_i, C_y = \sum_i P_i y_i,$$

where (x_i, y_i) indicates the pixel location of the i -th largest pixel in the predicated heatmap, and its prediction value is P_i , i.e., $P_i = \max_i P$, for $i = 1, 2, \dots, n^2$.

2.3 Face categorize

In recent literatures, researchers have discovered the relationships between genetic, environmental and racial factors [4]. In this section, I simply classify the detect face into two categories: (1) Asian and (2) Western style. I think that the Japanese animation style will be more suitable for the Asian group and American style will be suitable for the Western group. In this project, I simply calculate the following facial morphological information and group the subject accordingly.

Table 1 Face Morphological Information

Feature	Calculation
Eye distance	Pixel distance between center of left and right eye
Height of upper and lower lips distance	Vertical distance between upper/lower center lips points and mean of two tip points of mouth
Width of the mouth	Distance between two tips points of mouth

Right now, the thresholding is selected empirically, referencing to the publications[5]. In the future, I can prepare more face images with ethnical information, and learn those features from those images using the machine learning methods, such as decision trees.

2.4 Face carbonization

In this section, I use the developed tools to transform photo to cartoon image. The proposed method is based on GAN, in which a generator network and a discriminator network are utilized to transform the input image to another style.

Results, and limitations

The training and validation loss are illustrated in the Figure 3. The history is truncated if it does not change for 10 epochs. The total number of epochs is around 50.

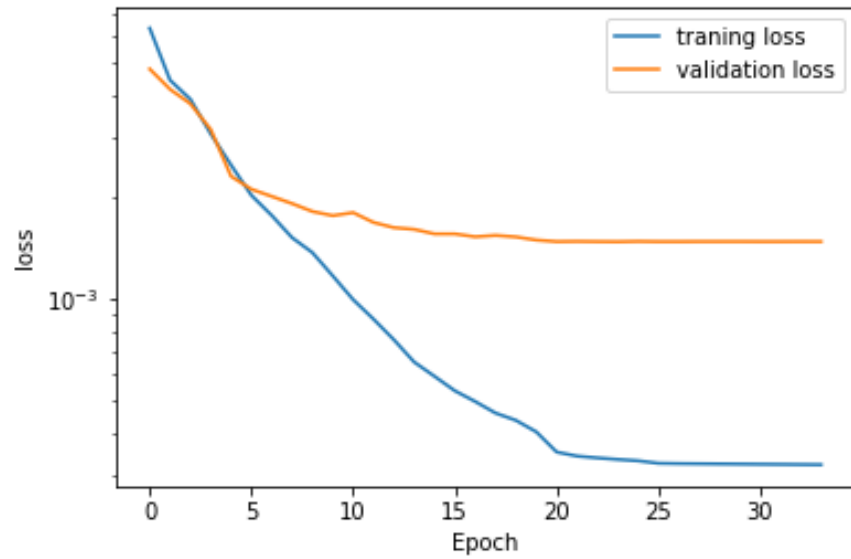


Figure 3 Training history

The facial key points detected using the proposed FCN is illustrated in Figure 4. One sample testing image and the 15 predicted heatmaps are included. The final 15 predicted key points are extracted and displayed in Figure 5. I can see that all the landmarks are detected.

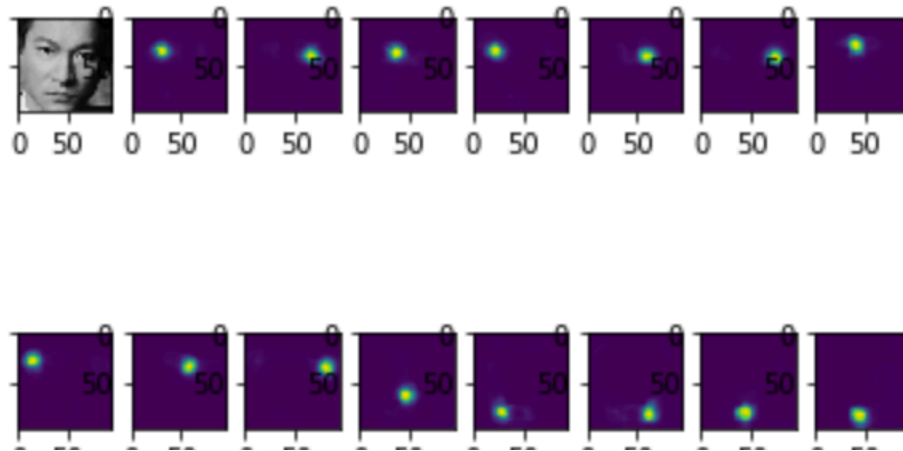


Figure 4 Test image and predicted heatmaps

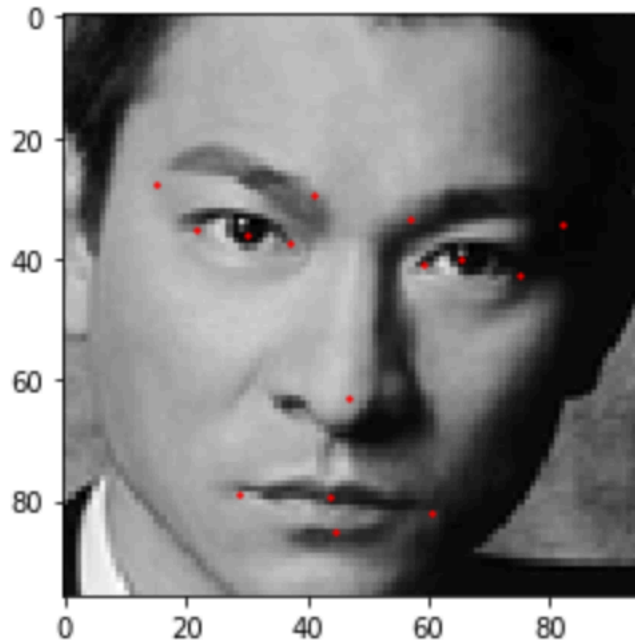


Figure 5 Sample test results (red dots indicate the detected landmarks)

In the proposal of the project, I plan to train two separate models using the developed method: one for Japanese animation style and the other for American comedy book style.

However, due to the limitations, I am unable to deliver the two models.

1. In our second stage, the FCN is developed using Tensorflow framework. The method in the fourth stage requires Tensorflow, Pytorch, and dlib as frameworks and dependencies. However, I cannot install the dlib module in my computer. So, I am not able to test the whole package. I tried to run the training and testing section without pre-processing, but it still failed.
2. At least hundreds of human face images and cartoon images with desired style are needed in order to train the transform model. It is easy to prepare the cartoon images which contain Japanese animation faces. To my knowledge, there is no available database for American cartoon faces only. It takes time and effort to prepare the datasets. That is why I skip the last stage.

Conclusions, and future plan

In this project, firstly the human face is localized in a portrait. In the second stage, a deep learning-based facial key points detection model is developed. A gaussian kernel is applied to transform the difficult point detection problem to a segmentation problem. In the third stage, I proposed to classify the suitable styles for different human faces based on the knowledge in the field of social science.

In the future, I will firstly finish two model training listed in the fourth stage. And I can further improve the landmarks detection by combining the first and second stage.

Reference

- [1]. Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. *In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.
- [2]. Bradski, G. (2000). The OpenCV Library. Dr. Dobb. *Journal of Software Tools*.
- [3]. Yoshua B. (2016). Facial Key points Detection from <https://www.kaggle.com/c/facial-keypoints-detection>.
- [4]. Tsagkrasoulis, D., Hysi, P., Spector, T., & Montana, G. (2017). Heritability maps of human face morphology through large-scale automated three-dimensional phenotyping. *Scientific reports*, 7, 45885.
- [5]. Wen, Y. F., Wong, H. M., Lin, R., Yin, G., & McGrath, C. (2015). Inter-ethnic/racial facial variations: a systematic review and Bayesian meta-analysis of photogrammetric studies. *PloS one*, 10(8), e0134525.