Facial Attractiveness Prediction

Tran Quoc Lap
20194443 - DSAI K64
lap.tq194443@sis.hust.edu.vn

Assoc. prof. Nguyen Linh Giang
Faculty advisor
giangnl@soict.hust.edu.vn

Abstract

Facial beauty assessment is a natural behavior of humans. In this project, I view this perception as a regression problem to predict human beauty. To solve the problem, I use the SCUT-FBP5500 dataset and deep convolutional neural network. The SCUT-FBP5500 dataset has 5500 frontal faces of different genders (female/male) and races (Asian/Caucasian). Experiments have been done on three different approaches: predicting on the original image, predicting on image with face extracted, predicting on face embedding based upon face recognition task. Having faces extracted from the images considerably improves model performance, while using face embedding increases model stability. My experiments are comparable to state-of-the-art results produced by related works. Once completed, the models can be used in tasks such as make-up suggestion, digital entertainment, content-based image retrieval, face collection or assessment on generated faces.

1. Introduction

Assessing human beauty is a natural behavior of humans, and the attractiveness of a face does influence many social aspects. Establishing a model to assess appearance attractiveness is a sensitive work. Then, in my project, I view it as a usual computer vision problem, where the highest importance is put on how to solve it efficiently and effectively. Besides, from a positive perspective, applying a beauty assessment model real-life might provide some practical benefits. For example, once completed, the model can be used to evaluate an artificially generated face for robots or virtual assistants, or can be used to assess make-up quality. I believe that, in a future not far when the standard of living steadily increases, humans will be more concerned with aesthetics and experience. At that time, aesthetic perception models like this, whether related to face or not, will be very popular. Also add that, facial beauty perception is not a brand-new problem, previous works have been done on it, and this problem is not in center attention recently. However, as I have mentioned earlier, my main focus in this project is the approach to solve the problem.

Recently, deep learning is a wonderful tool to solve computer vision tasks. It has shown to provide more impressive results than any other hand-engineering methods so far. So, in this project, I decided to use deep learning as a skeleton for the solution.

2. Related Work

There were many attempts to solve the problem of facial attractiveness. Indeed, the first step in a project is to investigate related works to see state-of-the-art achievements in the field. However, this is my first experience in a specialized project, so I had actually done the research on previous works too late - when I almost finalized everything. At the time of writing, there is no more time to research. So, in this section, I could just list papers that I investigated before or while running the project.

In 2015, Xie *et al*. [4] noveled the SCUT-FBP dataset containing face portraits of 500 Asian female subjects with attractiveness ratings. Images contain front-on face portraits of Asian female subjects with neutral expressions, simple backgrounds, and minimal occlusion. The average number of raters per image of the SCUT-FBP dataset is 70. The authors performed on this dataset with different combinations of facial geometrical features and texture features using classical statistical learning methods and the deep learning method.

In 2018, Liang *et al*. [2] propose a new diverse benchmark dataset, called SCUT-FBP5500. The SCUT-FBP5500 dataset has 5500 frontal faces with diverse properties (male/female, Asian/Caucasian) and face landmarks, beauty scores distribution. They evaluated the SCUT-FBP5500 dataset using different combinations of feature and predictor, and deep learning methods. They observed that all the deep CNN models are superior to the shallow predictor with hand-crafted geometric features.

3. Data

Among facial datatset labeled with beauty scores used in previous works, I have found AVA dataset, SCUT-FBP and

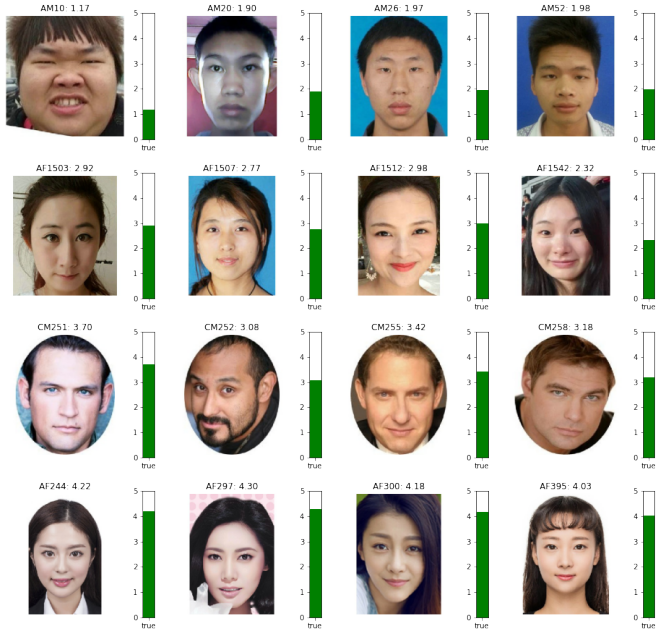

Figure 1. Images and beauty scores from the SCUT-FBP5500 dataset.

SCUT-FBP5500 publicly available with the highest diversity. However, the AVA database is originally designed for aesthetic analysis of entire images and not facial attractiveness. The SCUT-FBP dataset, though having a large and centered frontal face, contains only 500 Asian female images, which is not supportive in terms of generalization for males or Caucasian.

In the meantime, the SCUT-FBP5500 dataset has 5500 color images of frontal faces collected from the internet and some databases. All faces are scored by 60 volunteers. The dataset includes 2000 Asian females (AF), 2000 Asian males (AM), 750 Caucasian females (CF) and 750 Caucasian males (CM). This diversity makes the SCUT-FBP5500 the best choice for my problem. Figure 1 demonstrates some of the images in the SCUT-FBP5500.

Each image is of size 350×350, labeled with a continuous beauty scores ranging from 1 to 5. Figure 2 describe density distribution of the beauty score. It is clear that the score is distributed unequally where there are less Caucasian or faces scoring in range $(1, 2)$. So far, imbalanced dataset might lead to some issues such as inefficiency in training progress of some algorithms. However, at the moment, it is hard to determine if this imbalance negatively affects the attractiveness measurement and if 5500 data point is sufficient to eliminate that negative effect.

One should know that even ground truth labels might have some noise itself. While labeling, volunteers might be affected by personal taste, culture, momentary emotion, age, etc.

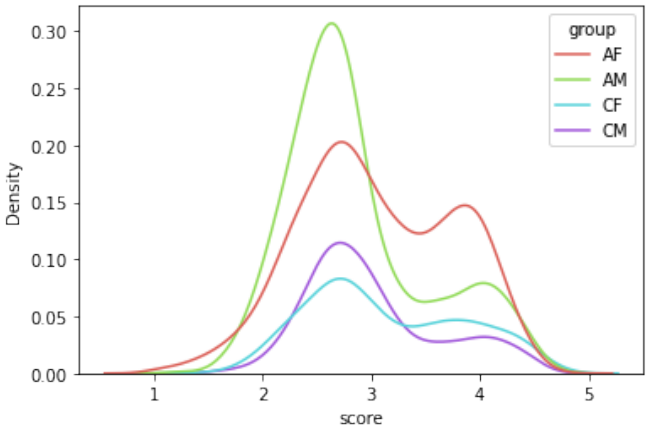

Figure 2. Distribution of Asian females (AF), Asian males (AM), Caucasian females (CF) and Caucasian males (CM).

4. Method

The predecessors' works have shown that using deep learning models gives superior results compared to hand-engineering methods. So in this project, I will employ deep neural network to solve the problem. The general architecture for the network is inspired by some famous architecture such as MobileNet or GoogleNet, and is pre-trained on ImageNet. For the convenience of end-users, the model can receive input as an image without any manual processing, but can be transmitted directly over the network to return an output of a real number in the range 1 - 5 corresponding to the beauty score of the face in the image. Partly to get an insight how a machine learning project advances, partly to see if I can solve the problem better than predecessor.

One possible way is to convert the image to grayscale before feeding it into the network. The advantage of this approach is that the face might be evaluated regardless of make-up or skin color and more focus on structure of the face. Besides, it is less computationally expensive as the image is reduced from 3 RGB channels to just one channel, therefore requires less training time. However, reducing the image's number of channels also means that the project will be less applicable in some cases, such as makeup suggestions or face synthesis. Therefore, I decided to use the RGB color image.

I started by dividing the dataset into train set and test sets with a ratio of 4:1, that is, 4400 images for the train set and 1100 images for the test set. Data is shuffled before division to ensure generality. For the train set, I chose batch-size of 32.

This project uses a deep convolutional network. So the first step is to define a loss function. There are a few options: MAE, MSE, RMSE. TensorFlow does not support RMSE loss and previous works use MAE as evaluation metrics, so, in order to make comparison with state-of-the-art results, I

selected MAE as loss function as well as evaluation metric.

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|$$

For hyperparameters, I'm interested in learning rate and optimizer. The general approach for hyperparameters optimization is to search in coarse ranges and then, depending on where the best results are turning up, narrow the range. In the first stage, I perform the initial coarse search for just 5 training epochs, then select the best hyperparameter configurations of them and perform a second narrowed search for a few more epochs. eventually I pick one best configuration of learning rate and optimizer, then perform a complete training. During training, I observe the training loss and reduce the learning rate by tenth if the loss is on plateau.

To prevent overfitting, I employ 2 approaches. On one hand, early stopping is activated whenever the validation loss is about to increase. On the other hand, image augmentation is performed, such as flip, rotation, zoom, translation, noise addition, quality reduction, brightness adjustment.

Also, to generate an effect of ensemble learning, I apply dropout layers between fully-connected layers. The value of drop-out rate is 0.5.

In fact, hyperparameters tuning and overfitting prevention is a basic procedure in any machine learning project, so my major description in this report is not put on these steps. Instead, I focus on the different approaches to solve the problem. The details for these approaches are presented in section Experiments. For each approach, after a complete training and validation, I test with images captured from usual camera as well as with real-time video stream to get a sense how stable the model is.

5. Experiments

In this section, I describe different approaches I had tried in my experiments. Before digging into details, I need to clarify that for the first approach, I had tried many optimizing algorithms: SGD, Adam, AdaDelta, RMSprop. For each optimizer I also initialize the learning rate with varying values. Generally I found that there is no significance between these optimizers - all of them have the CNN model converge to the same training loss. So the final configuration for all approaches in my experiments is: SGD optimizer with Nesterov momentum 0.9 and initial learning rate $5\text{e}{-3}$ or $1\text{e}{-2}$.

5.1. The first approach

The first approach, I use MobileNet v3 Large architecture [1] as the underlying feature extractor. The feature extractor is then plugged with one 256 fully-connected layer and one 128 fully-connected layer respectively, finally with an output layer of 1 neuron. Input shape is $(224 \times 224 \times 3)$

Figure 3. Variation of face position and conditions.

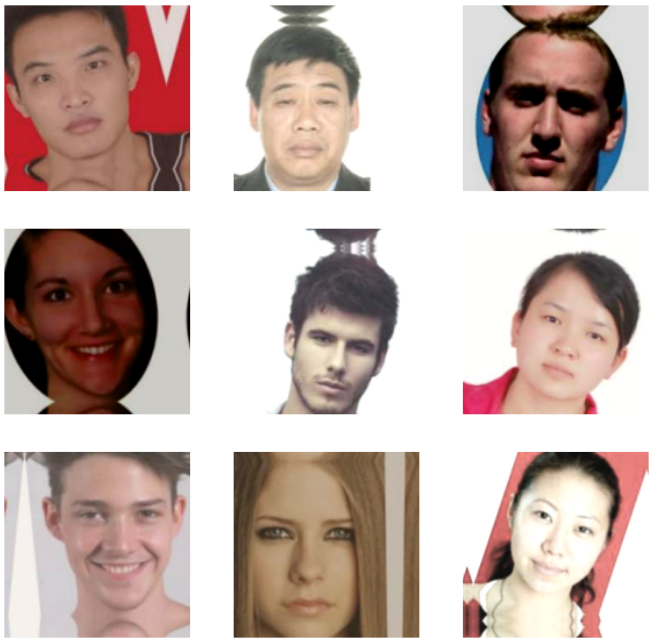

Figure 4. Image augmentation for the first approach.

and raw batches of original image are fed directly to the network. My desire is that the model comes in handy for the end-user, which means it should be able to handle face of different scales, faces with background clutters, or faces with different viewpoints (see Figure 3). In fact, original images in SCUT-FBP5500 are faces already cropped and centered, so it turns out image augmentations such as zoom or translation is highly concerned (See Figure 4).

After around 30 epochs, the MAE loss starts to converge to 0.27 (see Figure 5). Evaluation on the test set gives Pearson correlation 0.87. A few of the predicted results on test set is illustrated in Figure 6.

Now the questions are: What has the model learnt? What does the output score represent? Whether that score is a beauty representation of face only or something else, such as the background? Is the output score affected by hair style or emotion expression? So, in order to get a sense of how

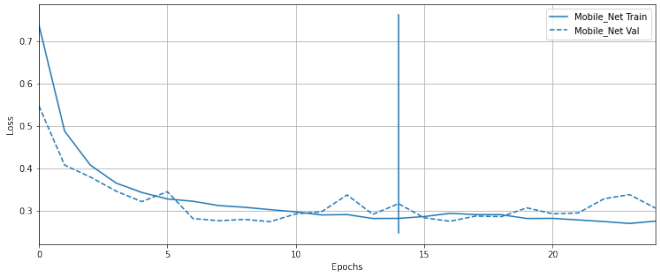

Figure 5. Learning curve of the MobileNet v3 Large model with original image input (the first approach).

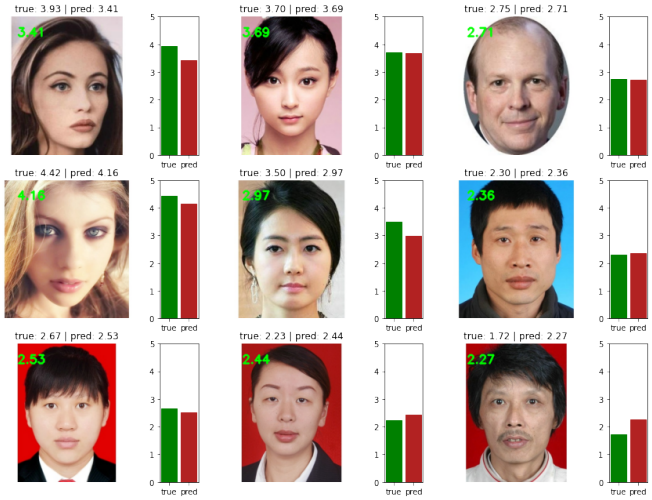

Figure 6. The first approach's prediction on test set.

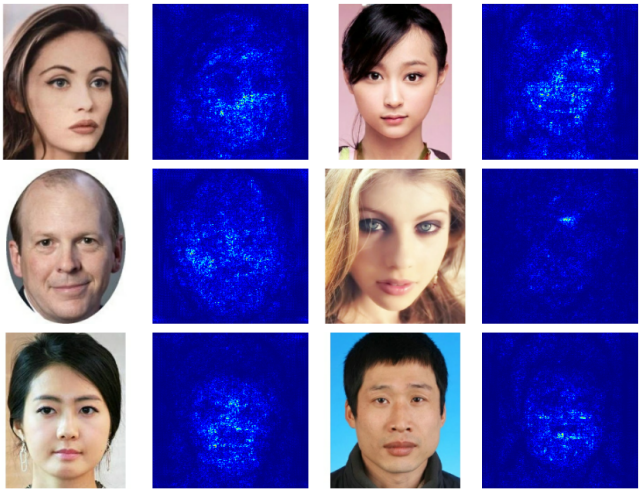

Figure 7. Saliency map on test images. The brighter pixel is, the more attention it receives from the model.

the model scores images, I visualize the model's attention on a saliency map using gradient ascent (see Figure 7).

Given the saliency map, we see that the face is in focus, so we are confident that the model has learnt something from the face. However, we also notice that the face, though is the brightest in saliency map, is not the unique part of the image that receives attention. There are some patches around the face that also get attention from the model, which means the background and hair, etc. also contribute to the beauty score. This is undesirable as we just want only face matters.

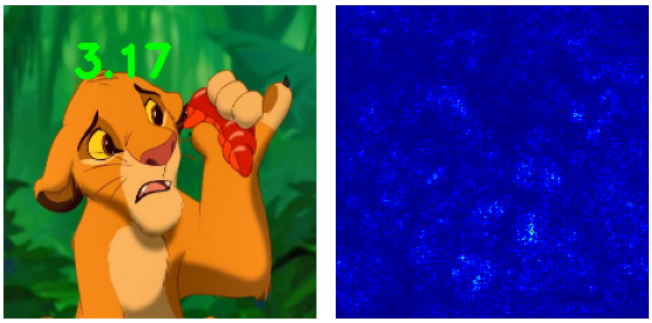

Figure 8. The first approach's prediction on a non-face image and the corresponding saliency map.

Another problem with this approach is that, even with an image with no face, the model still gives a beauty score. Though the model's attention is distributed everywhere and does not ensemble to any particular place in the image, the image should not be evaluated.

5.2. The second approach

Given that the first approach produces a model capable of face evaluation, but with some noise drawn by the background behind the face, I take a second approach: before feeding any image into the CNN model, the image is cropped to yield face only (see Figure 9). The chance is that the model can pay full attention to the face, and image with no face won't not evaluated.

Implementing this second approach requires 2 stages: extracting face and predicting beauty score. The second stage is similar to whatever in the first approach. As for the first stage, there are many traditional algorithms to detect faces in an image, such as Haar Cascade. This algorithm is very fast, but the problem is that when applying it with some training images, I notice that sometimes it detects no face or 2 faces (each training image has only one face), despite my effort to configure. So eventually, I employ a pre-trained Multi-task Cascade Neural Network [5] (MTCNN). Though using a neural network for face detection is slower than Haar Cascade, it's more accurate. MTCNN is very popular because it achieves very high accuracy on many benchmark datasets.

The network uses a cascade structure with three networks: P-Net, R-Net and O-Net. The three models are trained on three tasks: face classification, bounding box regression, and facial landmark localization. The three mod-

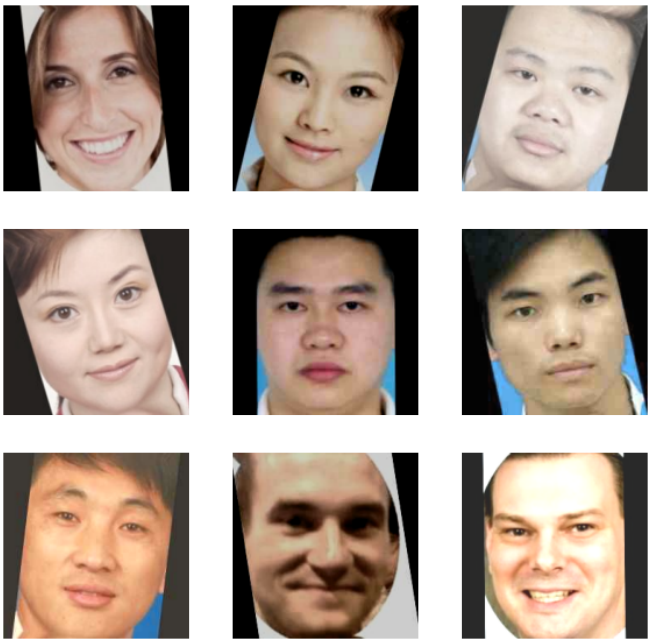

Figure 9. Training images with face extracted by MTCNN after augmentation.

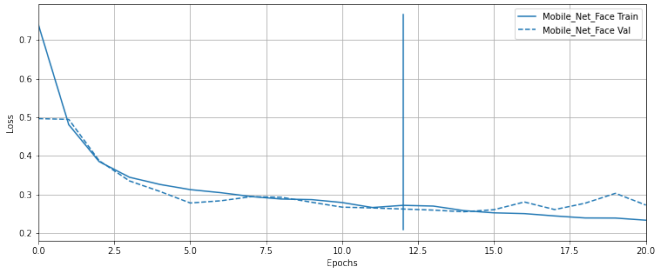

Figure 10. Learning curve of the MobileNet v3 Large model with face extracted by MTCNN (the second approach).

els are not connected directly; instead, outputs of the previous stage are fed as input to the next stage. This allows additional processing such as non-maximum suppression between P-Net and R-Net to be performed between stages. The MTCNN architecture is complex to implement, so I employed a pre-trained models that can be used directly for face detection.

After training for around 35 epochs with early stopping, the CNN model achieved 0.24 validation loss (see Figure 10) and 0.89 Pearson correlation . This is a remarkable improvement as compared to the first approach. Besides, as face is extracted before feeding into the network, image with no face is not evaluated.

At this point, the second approach has achieved good performance. However, when I test it with different images of the same person, the beauty score varies significantly, depending on the level of difference of the face in these images. For example, in Figure 11, although 5 faces are of the same person but with different viewpoints and expression, the model predicts them with significantly different beauty scores. This is a huge problem because the model has shown to be very unstable.

Figure 11. Beauty score prediction of the second approach on the same person. From left to right: $3.07, 3.11, 2.87, 3.11, 2.52$.

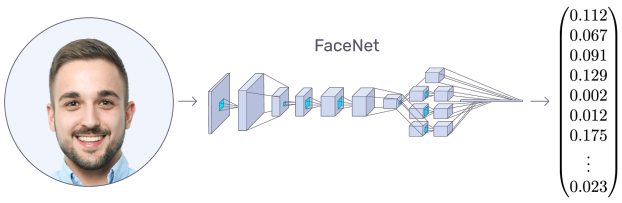

Figure 12. Illustration of FaceNet.

Now as I look back on the SCUT-FBP5500 dataset, I notice that among the whole dataset, there is no pair of images of the same person. This could be the reason why the model is not stable.

5.3. The third approach

So the question at this point is: How can the model learn the consistent characteristics of a face, given only one image per face available in the training set? In 2015, Schroff *et al.* [3] published a system called FaceNet that directly learns a mapping from face images to an embedding space (see Figure 12) where distances between embedding vectors directly correspond to a measure of face similarity. Faces of the same person have small distances and faces of distinct people have large distances. This embedding vector is consistent whatever invariances in pose, illumination, and other variational conditions.

I believe that my model will benefit from FaceNet, because if I capture one face various times, each time from a different viewpoint and conditions; his face embedding should oscillate around a fixed point, and therefore produce a similar beauty score. This is not the case with casual CNN models. To verify this, I make a comparison between my second model with a FaceNet model. As for my second model, I remove top classification layers, and get an output of 1280 dimensions. As for the FaceNet system, I employ a pre-trained Inception ResNet v1 from Hiroki Taniai, which has been trained on MS-Celeb-1M dataset, and get an output of 128 dimensions. I obtain embedding vectors of many

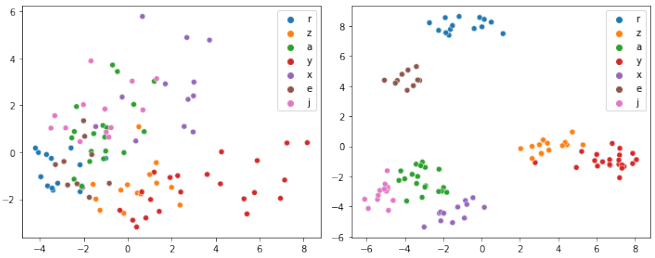

Figure 13. Embedding vectors of face compressed into 2D using PCA. Dots with the same color represent the same person. *Left*: using the last flattened feature map of the second model. *Right*: using FaceNet model.

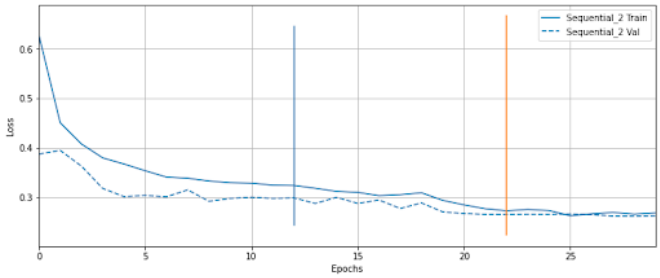

Figure 14. Learning curve of the model with face extracted by MTCNN and embedded by FaceNet (the third approach).

faces, use PCA to compress them to 2 dimensions, then plot them as in Figure 13. Clearly, the embedding vectors produced by FaceNet form clusters, while embedding vectors produced by the second model distribute everywhere and overlap with others' face.

The third approach includes 3 sequential stages: extract face, generate embedding, predict score. The first stage, extracting face, is the same as the second approach. The second stage takes face from the first stage, crop it to $(160, 160)$ and pass it through the FaceNet system to achieve a 128-dim vector. The third stage takes embedding from the second stage, passes it through a fully-connected layer of 256 neurons and finally to a beauty score. All layers in the FaceNet model of the second stage are freezed, only layers in the third stage are trainable.

After training for around 30 epochs with early stopping, the model achieved 0.26 validation loss (see Figure 10) and 0.86 Pearson correlation (see Figure 14). Though face embedding is more robust, the result is not as good as the second model, which achieves 0.24 validation loss and 0.89 Pearson correlation. So far, I can come up with one explanation: all the weights of the second model are trainable, so the output of the intermediate fully-connected layer is freely optimized to minimize the loss with no constraint. That's not the case for the third model, where FaceNet is not trainable, it constrains the embedding space and forces the rest of the predictor to minimize the loss under constraint. So the loss might not be as good as the second model.

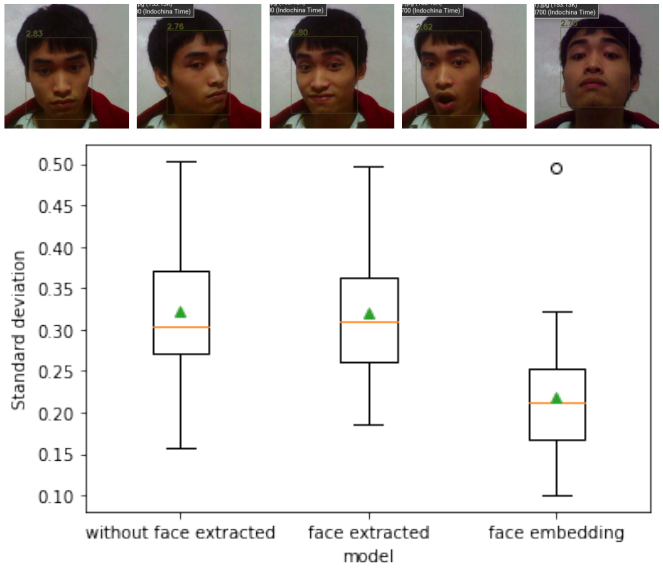

Figure 15. *a)* Beauty score prediction of the third approach on the same person. From left to right: $2.83, 2.76, 2.80, 2.62, 2.70$. *b)* Distribution of Standard deviation of 3 models.

Model pair	T-test p-value	Reject
('without face extracted', 'face extracted')	0.867	False
('face extracted', 'face embedding')	1.12e−6	True
('without face extracted', 'face embedding')	9.95e−7	True

Table 1. T-test result of models' standard deviation.

Nevertheless, my main concern is stability. To make a comparison, I collect many images of 34 individuals taken in the CMU Multi-PIE database. Each of them is along with a sample of 30 images with different poses, emotion expressions, under different lighting conditions. For each model, and for each individual's image sample, I compute the standard deviation of beauty score. Finally I do a T-test to check if there is any difference between standard deviation of 3 models. The result is shown in Figure 15 and Table 1 the third approach shown to be more stable as compared to the 2 previous approaches.

Given the T-test experiment result, it is confident that the approach based on face embedding is more stable than the 2 previous approaches.

Table 2 summarizes final results from previous works which I explored and my own results.

6. Summary

However, despite the dominant stability of the third model, there are two branches for application of my project.

Method	PC	MAE
Geometric feature + Gaussian regression (Liang *et al*. [2])	0.67	0.39
ResNeXt-50 (Liang *et al*. [2])	0.88	0.25
Without face extracted (mine)	0.87	0.27
Face extracted (mine)	0.89	0.24
Face embedding (mine)	0.86	0.26

Table 2. Comparison between my approaches and previous works.

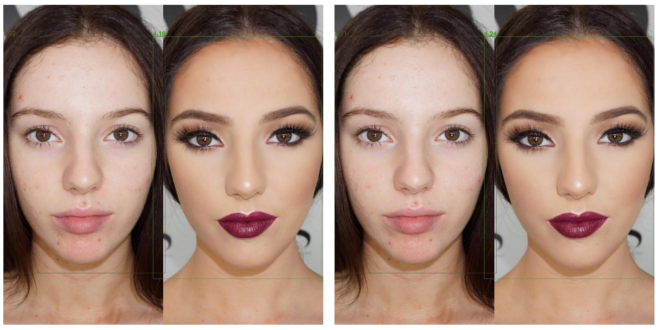

Figure 16. Model prediction after make-up. *Left* (Second model): 3.02 and 4.19. *Right* (Third model): 3.71 and 4.24.

On one hand, my project can be used to assess the consistent attractiveness of face structure, where factors like make-up, emotional expression do not contribute to a beauty score. In that case, the third model is the most preferred. On the other hand, my project can be used for make-up recommendations. In that case, the second model is more preferred. To illustrate, in figure 16, the same face is evaluated before and after wearing make-up. The second model shows a larger gap than the third model. It is dependent on specific problem to decide which model is more appropriate to use, but both of them are useful to give a relatively consistent estimation of beauty, given that noise in SCUT-FBP5500 is acceptable.

However, in terms of completeness, all these models have way more to be improved. First, the FaceNet system used in this project was pre-trained on the MS-Celeb-1M dataset, where the majority of people in this dataset are American and British actors. However, as described in Figure 2, the majority of SCUT-FBP5500 is Asian so the embedding clusters might not be very tight for the same person. Hence the third model might give better results if FaceNet is trained further. Second, so far all I know about is that there are 60 volunteers who label these images. There is no information about their culture, ages, gender, i.e there might exist some bias while labeling; so having more volunteers might give a more stable model. Third, one might use generative models to generate more facial expressions or condition variation of the same person, and then feed them all to the second model, which might serve as an alternative for

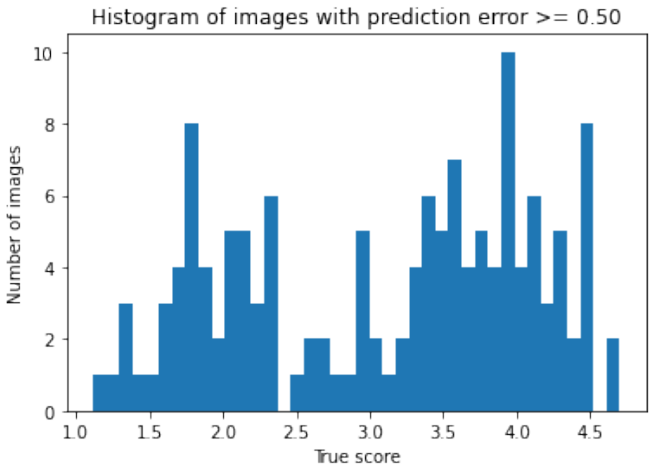

Figure 17. Histogram of prediction error larger than 0.5

the third approach using FaceNet. Last, but very important, is doing error analysis. In Figure 17, I plot the histogram of prediction error of the third model where the difference between prediction and the ground truth label is at least 0.5. There are 139 out of 1100 validation images, and most of them have a beauty score around 2 or 4. So there are 2 possible solutions: collecting more data of these cases, or doing hand-engineering to check whether they are wrongly labeled.

7. Supplementary Material

- Source code: `https://github.com/LapTQ/facial_beauty_perception`.
- Demo: `https://colab.research.google.com/github/LapTQ/mlapplications/blob/main/ML_applications.ipynb`

References

[1] Andrew Howard et al. “Searching for MobileNetV3”. In: (2019).

[2] Lingyu Liang et al. “SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction”. In: (2018).

[3] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: (2015).

[4] Duorui Xie et al. “SCUT-FBP: A Benchmark Dataset for Facial Beauty Perception”. In: (2015).

[5] Kaipeng Zhang et al. “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”. In: (2016).