Computational Beauty Analysis

Libby Day and Fawntia Fowler

1 Background

Our project was a continuation of the recent research published by Yael Eisenthal, Gideon Dror, and Eytan Ruppin entitled: *Facial Attractiveness: Beauty and the Machine* [1]. The authors used a variety of machine learning techniques to predict facial attractiveness ratings from photographs. They had two data sets, each consisting of ninety-two photographs of young, Caucasian women. All images in the first data set were taken by the same photographer under the same lighting conditions and with the same orientation. These high-resolution photographs were of Americans with neutral facial expressions who were wearing no glasses or jewelry. The second data set was of somewhat lower quality. The women in these photographs were Israelis, some of whom were wearing jewelry or smiling with closed lips.

Eisenthal, Dror, and Ruppin used two distinct representations of the women's faces: the *feature representation* and the *pixel representation.* The feature representation consisted of distance measurements between *feature landmarks* on the face and ratios between these distances. Examples of feature landmarks include the centers of the pupils, the corners of the mouth, and the endpoints of the eyebrows. All distances were normalized by the distance between pupils. Additionally, average hair color, skin color and skin smoothness values were included as part of the feature representation. The image representation was simply the original photograph converted to grayscale and concatenated by columns to become a vector.

The authors of [1] used both classification and regression techniques. For classification, they retained only the photographs with average attractiveness ratings in the top or bottom quartile. They labeled these photos as "attractive" or "unattractive" accordingly. Their best classification results are summarized in Fig. 3. They attempted several kernels but found that a linear kernel worked best for SVM. They also tried K-nearest neighbors (KNN).

Besides classification, [1] used the regression version of SVM to predict attractiveness ratings. They were able to achieve a 0.65 correlation on a test set by combining the predictions from the feature and pixel representations, but the feature representation alone performed nearly as well with a 0.6 correlation. A very promising result was the shape of the learning curve. (See Fig. 4) By extrapolation, it appears that significantly higher correlations can be achieved by using a moderately larger data set. This expectation was a big motivator for this project, which leads to our objective: to repeat the work of Eisenthal, Dror, and Ruppin, but using a larger training set.

2 Data Collection

Practically unlimited numbers of photos of young women rated according to attractiveness are available on HotOrNot.com. We received permission from HotOrNot to use their photos for our project. We originally acquired 280 color photographs of women between the ages of 18 and 25 with corresponding ratings from the HotOrNot web site, after sorting through more than a thousand other photos that were deemed unusable due to poor resolution, poor lighting, bad angle, et cetera. (We ended up keeping about one in ten perused photographs in the end.) Initial selection requirements were that both sides of the nose had to be in view, the woman had to be looking reasonably straight ahead, and both eyes had to be visible. Also, no pictures of women showing a lot of skin were used, as this would likely bias the ratings. (We avoided full body shots in general.) Upon recording feature landmarks, we realized the need for being even more particular, and further reduced the size of our data set to 200 photos. However, this is still more than twice the size of the original dataset from [1].

All 200 photos were rotated so that the head was vertical and cropped around the face. We recorded 42 feature landmark locations on each photo using a Java applet we wrote that outputs the coordinates of mouse clicks to a file. The 42 feature landmarks are shown in Fig. 1. They include all the feature landmarks used in [1] and an additional four points – one on the arch of each eyebrow and on the sides of the chin. Another applet averaged the RGB color values of pixels in a selected region. We used this applet to collect average hair, eyebrow, skin, eye, lip, and teeth colors. (In contrast, Eisenthal, Dror, and Ruppin used only hair and skin color.) Photos without visible teeth were assigned a teeth color equal to the mean of the teeth colors in the other photos.

We wrote a Matlab program to convert the feature landmark locations into the feature representation. The feature representation is made up of several distance measurements, normalized by the distance between pupils. It includes a few ratios between distances as well. See the appendix of [1] for details. In addition to the distances and ratios in [1], we also measured average chin slope and the height of the eyebrow arch. However, our feature representation did not include the symmetry indicator used by [1], because the validity of their symmetry indicator relies on the lighting conditions being the same in every photo,[1] which was not the case for our data set. Also, because many of our photos had relatively poor resolution, an edge detector found edges created by pixelation rather than by imperfections in the skin. Thus, we were unable to use a skin smoothness indicator as in [1].

For the pixel representation, we scaled the (already rotated and cropped) photos so that they were all the same size. Then we converted them to grayscale. Unfortunately, due to time constraints, we did not align the mouths and eyes as in [1].

3 Data Processing and Results

We performed principal component analysis (PCA) on all 200 grayscale images to obtain eigenfaces. The eigenfaces form a basis for the original images, with eigenfaces corresponding to larger eigenvalues capturing global information, and eigenfaces corresponding to smaller eigenvalues capturing fine detail. (See Fig. 2.) We ran SVM (with linear kernel) to classify the top and bottom quartiles of the data set as attractive or unattractive. This resulted in a training error of zero and a generalization error of 0.60, as estimated by hold out cross validation by training on 70 randomly chosen images and testing on the remaining 30. We then ran filter feature selection to lower the number of eigenfaces used for prediction in order to reduce the amount of variance and improve generalization error. The scores we used for filter feature selection were the magnitudes of the correlations between the coefficients of the eigenfaces and the human ratings. Again, we used 70-30 hold out cross validation to estimate generalization error during filter feature selection. (See Fig. 7.) The minimum generalization error occurred at about 100 eigenfaces. Thus, we chose to use eigenfaces with the 100 top scores. To get a better estimate of generalization error with the 100 eigenfaces, we used 10-fold cross validation. The result was a generalization error of 0.15. The training error was still zero. The learning curve for the pixel representation using the top 100 eigenfaces appears

[1] Their symmetry indicator is the sum of squares of the differences in pixel values when the pixels are reflected across a vertical axis.

in Fig. 6.

We also performed PCA on the feature representation before performing classification SVM. The best generalization error was about 0.35 with a 0.17 training error. Since the training error was lower than the generalization error, we again performed filter feature selection as on the pixel representation (prior to PCA). The resulting plot is shown in Fig. 5. The minimum generalization error occurred at 31 features; however, performing PCA and then SVM on just the top 31 features did not improve generalization error significantly. The generalization error (using 10-fold cross validation) was 0.33, while the training error was 0.24.

4 Conclusions

Looking back to [1], our results are comparable. We both were able to accurately predict attractiveness classes about $\tilde{8}$5% of the time, but by using different representations. While [1] obtained their best results with the feature representation, our best results came from the pixel representation. This seems very reasonable since our data set was far more varied than either of the training sets used by [1]. Presumably, the very sensitive feature representation works much better on uniform data sets. The values of measurements and proportions in the feature representation can vary significantly with only a slight head tilt. Very rarely was a woman in a photo from our data set looking precisely straight ahead. Furthermore, our lack of consistent photo quality and lighting could also have an impact. Perhaps our biggest source of error was the variety in facial expressions (serious versus large smiles with teeth, et cetera). On the other hand, our much larger data set was able to improve notably upon the pixel representation results, despite the poor quality of our data. In the end, it seems that while the feature representation prefers quality to quantity, the pixel representation will perform well, despite relatively poor quality, as long as the data set is large enough. We predict that for data sets where both quality and quantity are present, both the pixel representation and the feature representation will lead to accurate predictions.

5 Ideas for Future Work

Due to the time-consuming nature of our data collection process, we did not have enough time to try out everything we thought of during our research. In the future, we would like to perform a regression version of SVM on our data to see if our larger sample size would increase the correlation achieved by [1]. Additionally, we would like to align the mouths and eyes in the photos, for potentially better results using the pixel representation. We also considered testing different kernels for SVM. Although [1] determined that a linear kernel worked best, that might not be the case for our data set. Additionally, we would like to test both representations using KNN.

Another idea is to change the feature filter so that it can account for nonlinear relationships. Many facial features are considered to be most desireable at neither extreme. For example, a huge nose is not often thought well of, but if it is too small, that can also be undesireable. If we were able to design such a feature filter, it would probably have a positive effect on our generalization error. It is also possible, considering the learning curve in Fig. 6, that adding still more photos to our data set could lead to better classification and regression.

References

[1] Dror, G., Eisenthal, Y., and Ruppin, E. (2006). Facial Attractiveness: Beauty and the Machine. *Neural Computation*. 18, 119-142.

And thanks to HOT or NOT.

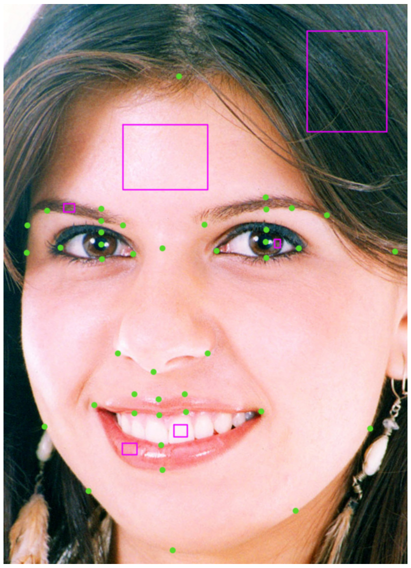


Figure 1: The feature representation is derived from 42 points and six color samples.

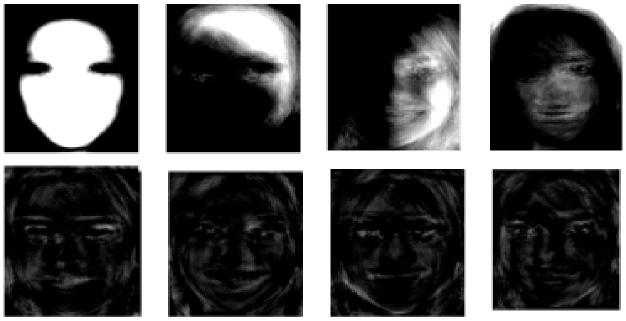
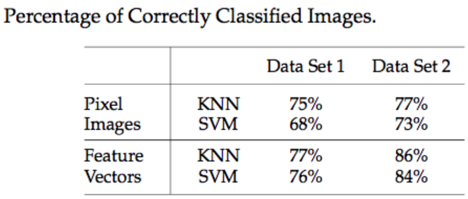


Figure 2: Eigenfaces obtained from our data set.



Percentage of Correctly Classified Images.

		Data Set 1	Data Set 2
Pixel Images	KNN SVM	75% 68%	77% 73%
Feature Vectors	KNN SVM	77% 76%	86% 84%

Figure 3: Classification results of [1].

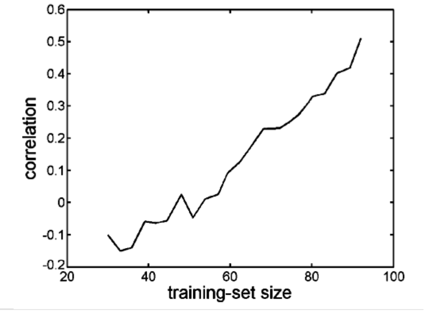


Figure 4: Correlation learning curve obtained by [1] using regression SVM.

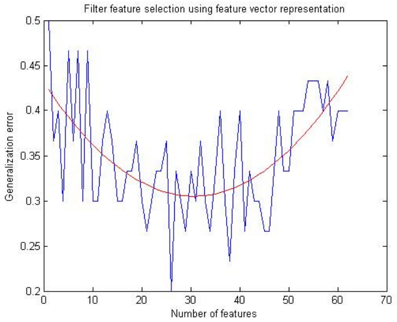


Figure 5: Generalization error as a function of the number of features used in the feature representation, as estimated by 70-30 hold out cross validation.

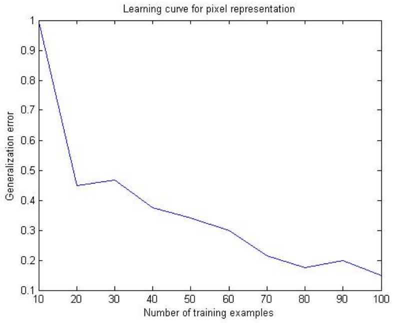


Figure 6: Estimated generalization error for the pixel representation as a function of the number of photos in the training set according to 10-fold cross validation.

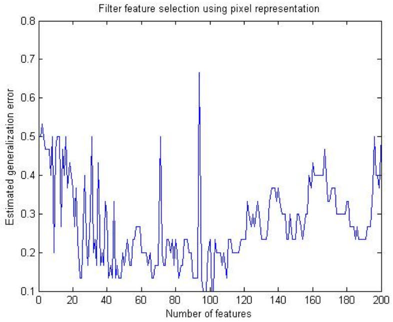


Figure 7: Estimated generalization error as a function of the number of eigenfaces used.