Face-CNN

CS280 Final Project, Spring 2015

Michelle Nguyen 22761256, Allen Tang 22900725

1 Overview

Human attraction based on physical characteristics translates to a picky phenomenon. Individuals value a wide range of varying attributes, preferring certain facial shapes, hair colors, eye colors, hair lengths, etc. Due to advances in social technology, we are meeting more potential mates more quickly than ever. However, filtering out matches for physical compatibility can be a relatively time-consuming task.

We use classifiers built upon convolutional neural networks[2], support vector machines, and k-nearest neighbors to classify various attributes that are huge factors in human attraction. Some of these traits are commonly explored in prior works, such as race and hair color[1]. However, our method also allows us to consider more novel features, such as the presence of makeup. These classifiers are then run on ?real-world? image sets to create a database of photos indexed by attributes. Upon this database, we implement a simple search engine that allows users to easily filter for people of their preferred features.

2 Dataset

We created our own datasets using the Flickr API, collecting images based on a set of image tags, i.e. ?asian portrait?. By using this versatile method of querying Flickr, we can easily create data for features that have no available dataset online. After pulling these images from Flickr, we had to manually clean our data by removing falsely tagged photos, excessive repeats, grayscale images, bad pose, and more. This constraint required us to generate a reasonably large dataset. This was cumbersome to obtain, but ideally allowed us to generalize our classifiers on a more extensive set of images that capture a wide range of conditions, such as illumination and expression. This ability to generalize is very important given that our ultimate goal is to run our classifier on user-generated images. However, it is important to note that Flickr image data is not completely diverse, as it is heavily biased toward photos of females and certain styles of photos.

After collecting the images, we used the Viola-Jones algorithm in OpenCV to detect faces in the photos. Since one of the attributes we wanted to consider was hair color, we increased the vertical size of the cropped bounding box to include the person?s hair as well. Finally, each crop was resized to 100x200.

For our project, we considered three attributes: race, hair color, and the presence of makeup. Our dataset for race consists of four possible categories. For the races Caucasian, Asian, Hispanic, and African we have 1097, 1627, 1124, and 607 images respectively. We categorize hair color into three categories, blonde, dark (which includes black and brown), and red. For these categories, we have 572, 915, and 415 photos respectively. Finally, the problem of detecting the presence of makeup is a binary classification. We have NUM images of people with no makeup and NUM images of people with makeup.

3 Face Detection

SVM Our first approach was to use a SVM with a linear kernel. We tried a variety of techniques to standardize the images: No change Normalization to mean 0 and unit variance Normalization by dividing image mean We also experimented with using the raw pixel values versus converting the image to grayscale. Then, we applied a Gaussian blur with sigma of .7, and used a specific mask on the image according to the attribute we were classifying by. We experimented with parameters and found c=0.05 gave us best results.

CNN Before sending the images through our neural network, we first resized our photos to a size of 227x227. Then, we attempted a variety of CNN architectures, each trained for 10,000 iterations. Some of them were: 1) AlexNet with Softmax 2) Modified AlexNet, with convolutional layers 3, 4 removed 3) Modified AlexNet, with convolutional layers 3, 4, as well as fc6, relu6, and drop6 removed.

K-means and kNN Specifically for hair color, we tried an approach involving both k-means and kNN. The aim of this method was to determine the label for an image using the known labels for images that had a similar hair color. We first applied a mask to each image to extract the hair at the top of the person?s head. After applying the mask, we used k-means and vector quantization to find the most dominant color in that image region. The RGB values for that dominant color were used as the feature vector for that image in kNN. To predict an image, we found that using a large number of neighbors kept us from overfitting. We determined that our best value of k was 300.

Results recall and precision for everything. chance accuracy SVM for race: Our accuracy was 72.7nocrop = 74.918compare our results for SVM and CNN? CNN better because more invariance or something. mention crop vs nocrop is similar.

0.808959 cropped hair CNN full kNN talk about how lighting really affects it, pose such that it was hard to crop hair, hairstyle affects it.

something for makeup no makeup. probbly a hard problem because it was hard for us to tell as people too.subtle makeup etc.

Search Engine We then collected images by using the Flickr API with search queries that did not correspond to any of our attribute categories. For instance, we pulled images from Flickr corresponding to the generic tag, ?portrait?. Then, with the best classifiers for each of our attributes, we classified each image and stored them with their corresponding categories in a PostgreSQL database. Upon the database, we built a simple search engine using Ruby on Rails. This search engine idea is similar to the search engine created by Kumar et al. The search engine allows users to filter for attributes for a set of images that have been categorized by our classifiers. The images are displayed along with tags showing what attributes the photo?s subject has. We have future plans to make this search engine hook to the backend of our classifiers so that users can upload their own images for classification.

Conclusion Obtaining good data from Flickr was very challenging and tedious. Resorting to using Amazon Mechanical-Turk would have certainly helped in this regard. Due to these constraints, we were forced to run our classifier on a smaller dataset (orders of magnitude smaller in size than those used in previous approaches) than we would have liked. Due to our own human error, these datasets may also be slightly noisy as well.

Although our results, given our limited dataset, were not as good as those obtained via the region-specific SVMs by Kumar et. al, we have shown that neural networks can provide a decent framework for general classification problems involving attributes. We were able to get close to 75

future work? even more novel attributes? idk

References

[1] "FaceTracer: A Search Engine for Large Collections of Images with Faces," N. Kumar, P. N. Belhumeur, and S. K. Nayar, European Conference on Computer Vision (ECCV), pp.340-353, Oct, 2008.

[2] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In CVPR, 2013.