Facial Attractiveness Prediction for Online Dating

Aditya Patel and Chris Westendorf

**Motivation and Use Case**

We plan to build a Deep Learning model that predicts the attractiveness of people’s faces. Such a model can be used by online dating companies to decide which user profiles to show to which other users E.g: when a brand new user Bob signs up on a dating website and uploads his photograph, that photograph can be fed to our model which outputs a number from 1 to 5 indicating how attractive his face looks. If the model rated him 2.2, the dating company could potentially decide that his profile would not be shown to the upper echelon of users (those rated 4s and 5s).

**Supervised learning**

Approach: We will use the Convolution Neural Net (CNN) architecture. CNN is a sequence of feature extracting layers (Convolution filters) followed by dense supervised Learning layers (conventional artificial neural net)

Dataset:We will use the SCUT-FPB5500 dataset of faces. It has 5500 frontal faces with diverse properties (male/female, Asian/Caucasian, ages) and diverse attractiveness labels from 1 to 5.

Evaluation Metrics: We shall measure performance with *Root Mean Square Error* (RMSE) between the actual labels and our model’s predicted attractiveness scores. The lower the RMSE, the better our model.

Visualization Tool: We may employ Tensorboard for tracking and visualizing metrics such as loss and accuracy, visualizing the model graph (layers), viewing histograms of weights, biases as they change over time and projecting embeddings to a lower dimensional space

**Unsupervised learning**

Automatic Feature Extraction

* CNN will automatically extract the features using Convolution layers and max pooling
* We may transfer-learn weights/features of pre-trained models like Google’s FaceNet, ResNet, Keras VGGFace-Library of models or other face embedding approaches.

Manual Feature Extraction:

We may use NMF (Non Negative Matrix Factorization) to generate face embedding manually to be used later for simple linear/polynomial regression or random forest regression. NMF is suitable for our data type (image) because each feature (in our case pixel) has a positive value which is a requirement to use NMF.

**Challenges:** Tensorflow, Keras, Tensorboard and Transfer Learning are pretty new to us. We foresee a short learning curve/mild turbulence before running into smooth air!

**Contribution**:Both Aditya and Chris are equal contributors of ideas and code in this project.

**Rough Timeline**

Week 2: Submit draft proposal.

Week 3: Get 1st baseline CNN working.   
Weeks 4, 5 & 6: Experiment to Improve performance (RMSE) using various approaches mentioned above.  
Week 7: Prepare and submit final report