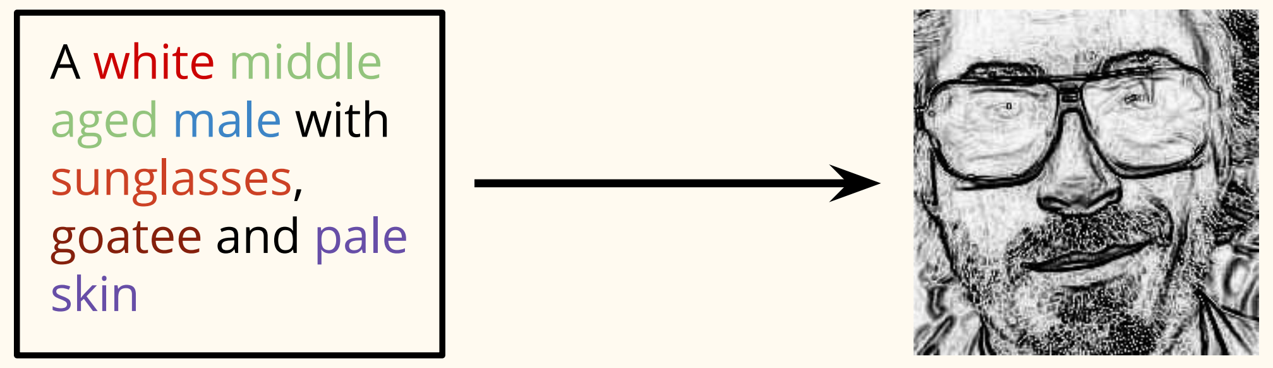
DeepFace: Face Generation using Deep Learning

Hardie Cate, Fahim Dalvi, Zeshan Hussain

Motivation

Convolutional Neural Networks (CNNs) are powerful tools for image classification and object detection, but they can also be used to generate images. We construct a system to generate faces from sparse descriptions. This model can be applied in many contexts, including law enforcement settings, medical applications, and art design. The techniques we employ can be applied to more general image generation settings.


A white middle aged male with sunglasses, goatee and pale skin

Data

Figure 1: Example cropped images in the training set; each image is reshaped to 224x224

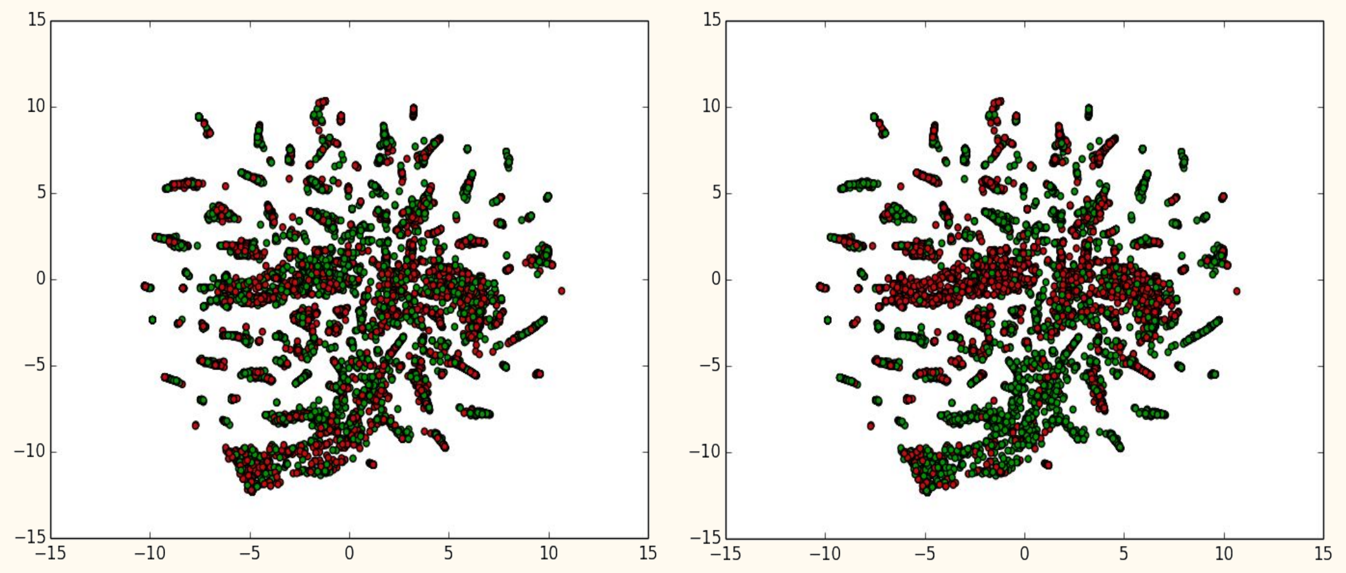


Figure 2: t-SNE visualization of FC7 activations with labeling of data points according to presence of a given attribute. The left plot shows "soft lighting" and exhibits poor clustering. The right plot depicts the attribute "youth" and is an example of good clustering. These activations come from the VGGNet before fine-tuning.

- Training set includes roughly 20,000 faces
- Test set contains 8,000 faces
- No overlap between two sets

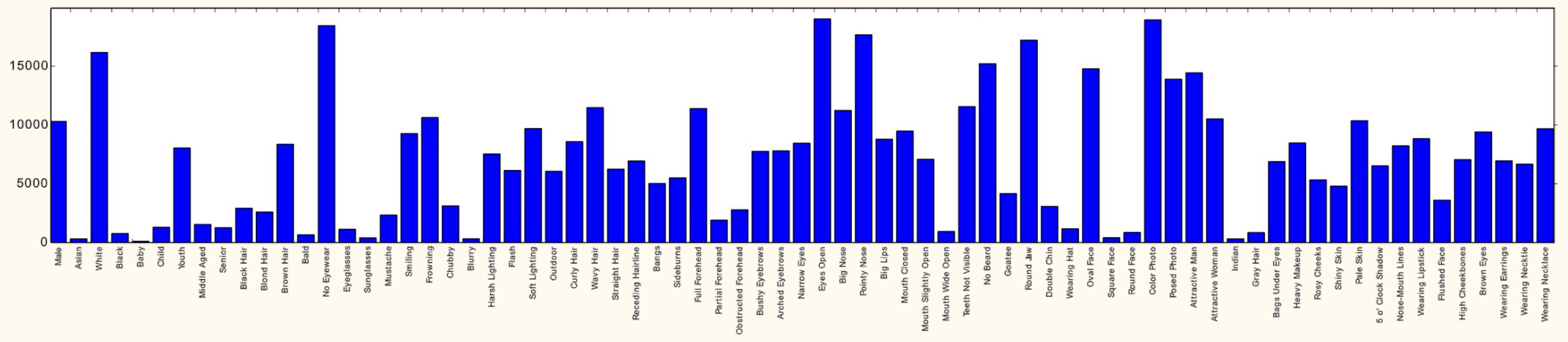
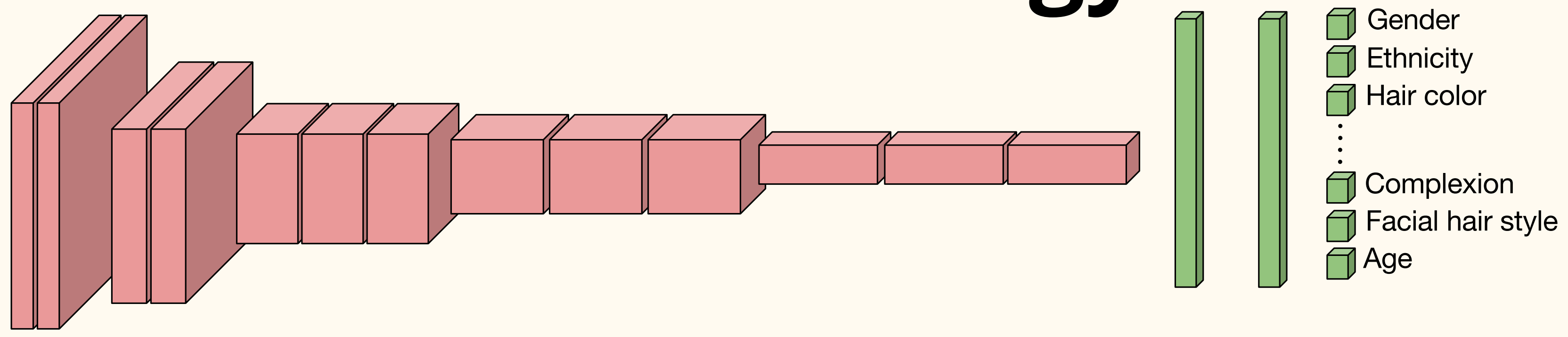


Figure 3: Bar graph of facial attribute distribution; displays the number of training images that show a given facial characteristic

Methodology



1 Fine-tuning
VGGNet weights adjusted using 42 softmax loss heads representing different attributes

2 Image Generation
Generate images using the fine-tuned network by boosting certain attributes

Baseline

- **Class visualization**
  - Random noise added to mean image; boosted toward certain attributes

Alternate Strategies

RNN Variational Autoencoder + Custom Gaussian Mixture Model (cGMM):

- Pair of RNNs
  - Encoder to compress images and decoder to generate images
- cGMM
  - Feature vector estimation via cGMM + image construction using feature inversion

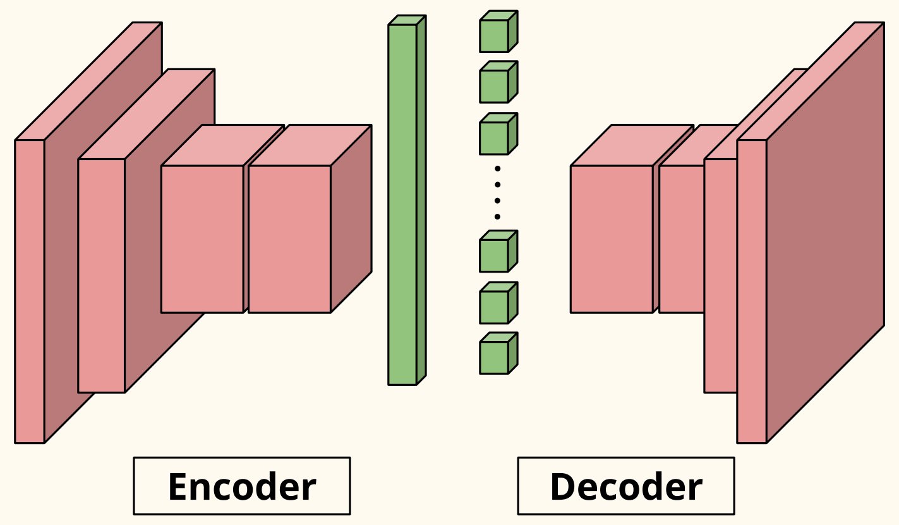


Figure 4: Alternate strategy for image generation that utilizes an autoencoder structure; the encoder learns the "codes" of the images while the decoder reconstructs images from their codes

Final Approach

Custom GMM

- Models CNN features as weighted sum of Gaussian samples
- Gaussians estimated by sampling subset of of activations across images
- Gaussian weights learned to minimize difference between predicted and observed features

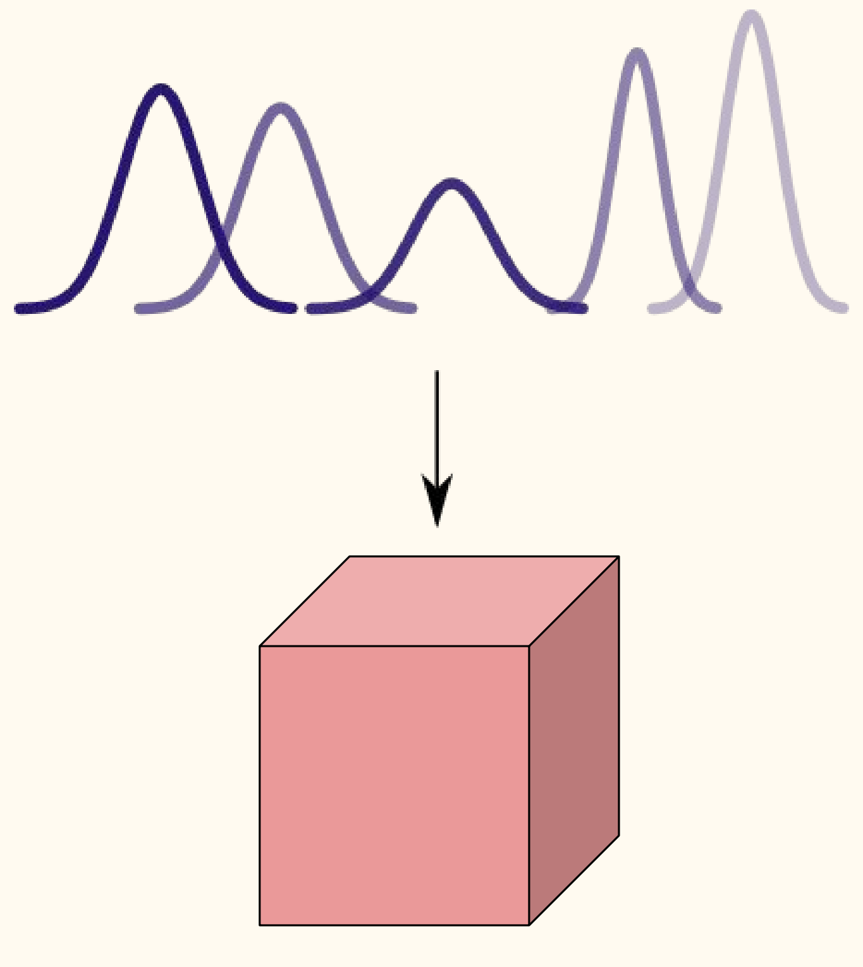


Figure 5: Weighted sum of Gaussian mean features produces activations for target layer

Results & Analysis

Training images are annotated with up to 73 different labels, each representing a facial characteristic. Because some characteristics are mutually exclusive, we condense the 73 labels to 42 categories.

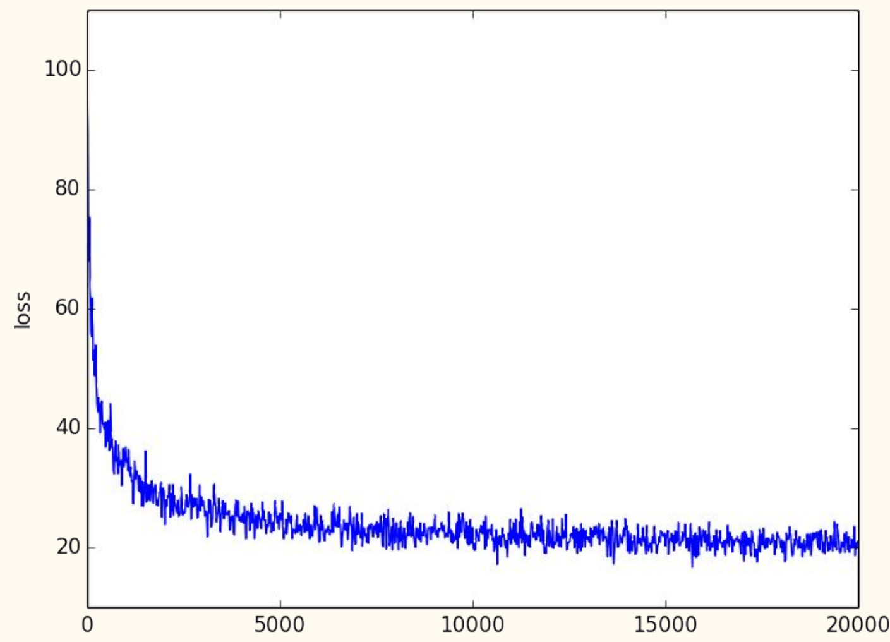


Figure 6: Graph of loss history during fine-tuning of VGGNet weights

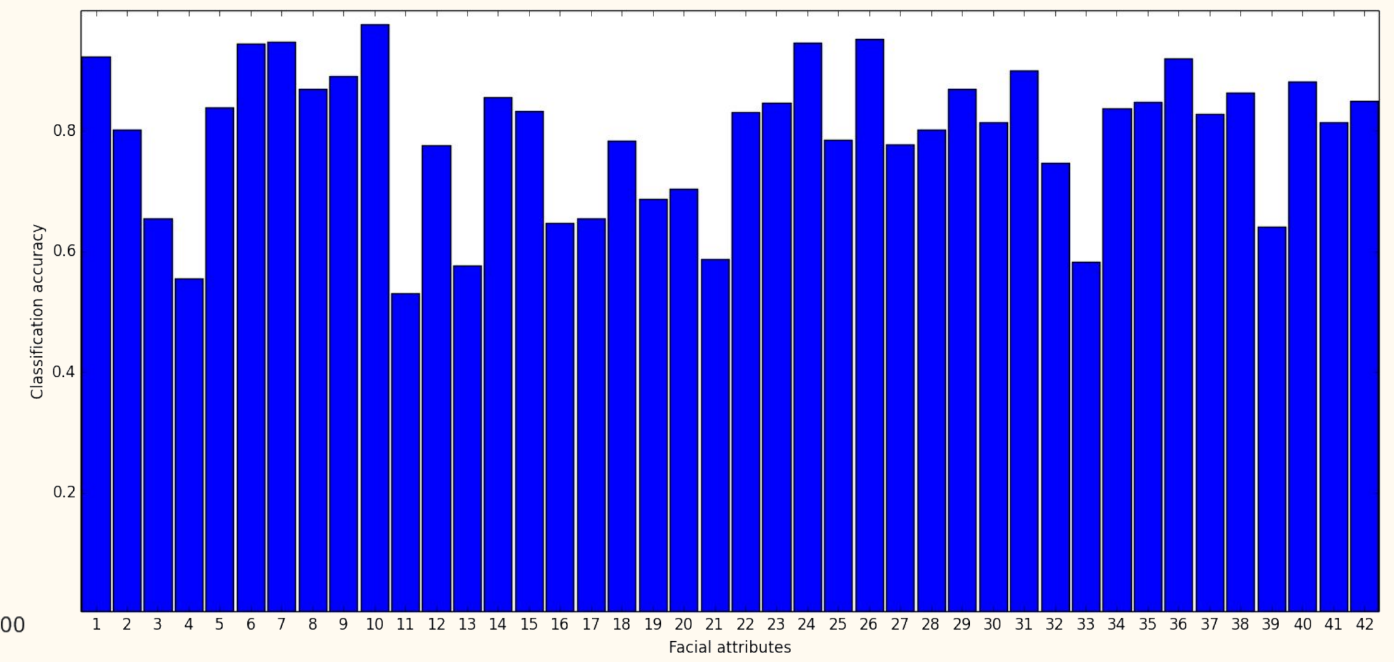


Figure 7: Training accuracies of the 42 multilabel classifiers attached at the end of the CNN; all accuracies are greater than 0.5 and most are consistently above 0.8

Baseline Visualization

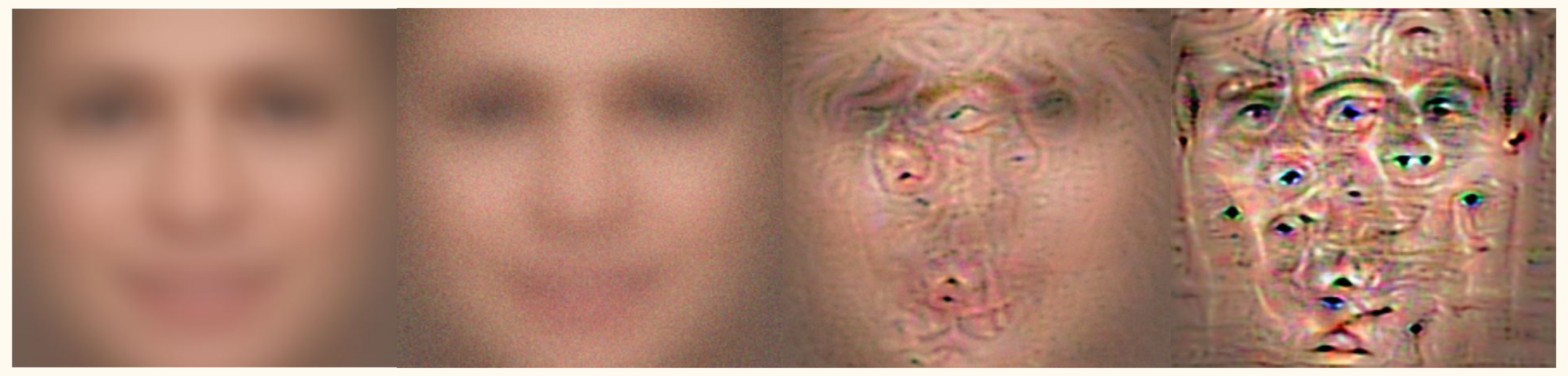


Figure 8: Pixel-wise mean image (far left); Mean image with added Gaussian noise (middle left); Image after several iterations of class visualization (middle right); Generated image of black male youth with black hair

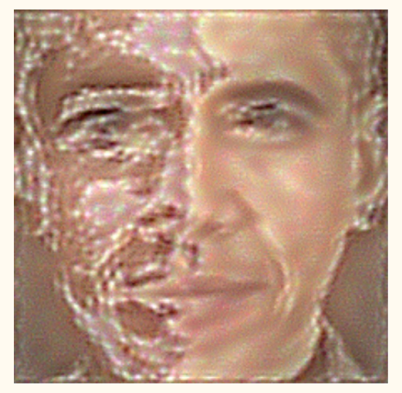

"Barack-inator"

Figure 9: Image of Barack Obama altered using feature inversion on Obama's true attributes

- Baseline visualization naively boosts features in image that "look" like desired attributes
- cGMM method attempts to approximate intermediate layer representations given a set of attributes

Future Work

We plan to tune hyperparameters for our cGMM training so that we can estimate more accurate feature activations for a desired set of attributes. Further, we will potentially use a variational autoencoder, as described in the alternate strategies section. Current state-of-the-art methods employ RNN autoencoders.