**COPY OF MY ORIGINAL REPORT: I wrote it in a bit of a rush; the writing is poor. I still have the original files, but the stored neural net is exceedingly large, if I recall – like a gig or so.**

**Face Generator – Written Report**

Sean Yanik

**What did I make?**

For my CS 240 final project, I made a customizable face generator.

**What can it do?**

The face generator allows the user total control of most all aspects of a human face. The user can **combine** two faces into one new face, **adjust specific features** of a face (“race”, nose size, width between eyes, gender, etc) and **save faces** that they have generated. These features are fun on their own, but become much more interesting when used with the **face loader**. The user can load an image of a face from their computer (any face, their own, a president’s, a celebrity’s…) into the program, and then edit that face. It can be very fun to see your own face reimagined in different ways. There are a number of applications for a program like this. It could be used as a replacement for police sketching, it could be used before plastic surgery (or facial reconstruction surgery following an accident) to pick the specific features someone wanted for their face. And, of course, it could be used for fun.

**Images:**

|  |
| --- |
|  |
| *Combining two faces* |

|  |
| --- |
|  |
| *Adjusting specific features (In order: Original, different “race”, different emotion, different eyebrows and nose)* |

|  |
| --- |
|  |
| *It Is very easy to load faces, though you must crop the image that you’re trying to load beforehand.* |

|  |
| --- |
|  |
| *Non face images can also be loaded with interesting results* |

|  |
| --- |
|  |
| *Faces can be saved as PNGs with the click of a button* |

|  |
| --- |
|  |
| *Specific facial features can be customized with simple sliders. I have done my best to label each slider for what it does (and group them) but because a Neural Network chose them, and not me, the labels may not be entirely accurate.* |

|  |
| --- |
|  |
| *There are eight collections of adjustable features, and 90 features in total. They can all be adjusted to the user’s preference.* |

|  |
| --- |
|  |
| *This is what the program looks like when it is initially opened. The “Default face”, when all of the sliders are set to the middle, is shown.* |
|  |
| *Different sub menus can be opened and closed.* |

**What are some design decisions that I made?**

I chose to:

Use a simplistic UI design, with several opening and closing menus, to avoid visual clutter.

Spend a couple of hours moving each slider up and down, and trying to figure out what they were doing (as I said, the computer chose these features, not me), so that I could label and group them. Otherwise, there would just be a list of 90 sliders, which would be quite confusing to a user.

Generate Faces instead of generating art, as I originally intended to do. As it turned out, art was too abstract a concept to get working within the time limit.

**What things have I utilized from the course?**

Algorithm runtimes and optimization have been very important for me throughout this process. As AI can be fairly resource intensive, it was very important to consider the runtimes of the code I was using.

I used a number of searching and sorting algorithms to process the dataset of faces I used to train this network.

I used thread synchronization in this project.

**How did I test my program?**

I tested new features as I added them, making sure that they worked properly before going forwards. This led to few errors ever occurring. Before submitting the project, I was careful check all of the buttons to make sure they functioned as was intended.

**Did I meet my goals?**

Originally, I intended to use a Generative Adversarial Network (a type of neural net) to generate art. I ended up with an Autoencoder (a different type of neural net) that generates human faces. My final project wasn’t what I set out to make, but I learned a huge amount in the process, and am very happy with what I made.

**Future Work?**

As I mentioned, my program uses an Autoencoder to generate faces. This is not the optimal solution; I could get better results (in less time, actually) if I use the more advanced “Principle Component Analysis” method. This would fundamentally change the structure of the program, and would take a lot of time to get working, but would provide much better results.

**Who did what?**

As I was the only member of my “team”, I did all of the work.

**What are my final thoughts?**

I have always dreamed of learning how to use AI and have finally done just that. All in all, this has been a very challenging subject to work with, but has been just as rewarding and fun. In summary, I greatly enjoyed creating this Face Generator.