Taylor Thurlow
Dr. Hao Ji
CS 4990 Fall 2018

# Project 3 - Developing Image Segmentation Models

I started project 3 with Option 1, hair segmentation. I used the Keras U-Net starter by Kjetil Åmdal-Sævik as a start point. I wanted to be able to use my model and code for the skin segmentation option as well, so I chose to use the '.ppm' files provided. Because I wanted to separate my training data into a training and validation myself, I wrote a small script to move all validation images into the testing data folder, ensuring that the indices for each image were not duplicated. Use of the '.ppm' files only required a slight tweak to the provided starter code, selecting the proper channel containing the mask of the type of image segmentation I was doing. I used a 90/10 ratio for training and validation. The intersection over union metric used in the starter was good for illustrative purposes, but searching through the comments on the page yielded some discussion on the shortcomings of the implementation, as well as a revised implementation which solved some of the issues the original one had, including a metric value which tends to increase over time as training proceeds - an increasing problem at higher epoch counts. I replaced the algorithm with the revised one.

The model itself required little to no tweaking, and any experimentation I did yielded either negative or neutral results. I tried using different optimizers with different learning rates, including SGD. Adam, with the default learning rate, ended up being the best option, when combined with a "reduce learning rate on plateau" callback enabled. For this callback I set patience to 10 epochs, with good results. I trained with a maximum of 200 epochs, but early stopping meant that training rarely exceeded 50 epochs. As a side note, it seems that the training masks aren't all that accurate - they get the general shape but there are a lot of extra mask spots that shouldn't be there. My

model seems to have largely accounted for them, but undoubtedly the training masks themselves could use some work.

The starter and provided RLE algorithm were written assuming masks would be written to disk, and reloaded. This seemed a little counter-intuitive, so I modified the provided CSV generation code to better fit a numpy array of masks. This resulted in less, more simple code. In general, the results were good, but I expect that there is some room for improvement in fixing up the training masks, and spending more time trying different model architectures.