Mason DiMarco Computational Vision HW2

There does not seem to be many visualizations within this homework, so the majority of this write up will just be descriptions of my code. The process itself is standard for most uses of an SVM. Essentially:

1. Load Data
2. Process / Clean Data
3. Shuffle Datapoints
4. Split into train/test Data
5. Build SVM model based on train features / labels, cross-validating data to reduce overfitting
6. Test SVM and record accuracies
7. Report result

The results I received were interesting. I achieved ~97% accuracy, which means either the SVM has found an outstanding linear separation, or, more likely it is grossly overfitting the datapoints. This could certainly be possible, given only 230 datapoints with 1200 features. Adjusting the slack variable should help, although 10x datapoints would more likely be even better.

And interestingly enough, this picture to the left came up very frequently

Image Pixel Intensities Model:

My guess with why the pixel intensities model is not as good as that of the Gabor Filters is that the model for image pixel densities is simply not as good of a fit. Pixel intensities tells much less about orientation and averages and more than likely creates a bit of an overfit based on similarities in intensities in the photos. I'd also like to point out that with a sample size of 230, there may be an overfit in general.

I will admit however that I do believe the gabor-filtered model has an overfit on it, perhaps due to how jpeg's deal with edges based on different cameras, it is able to find a clear distinction due to that. The problem here of course is that there are only 230 images, which is a relatively small sample size for 1200 features.

Tuning:

Tuning was hard on my machine with running such computationally intensive tasks, but I found that with such a high initial accuracy from the default parameters, the only way it could go was down. The initial parameters seemed to have found a good balance between what would be helpful or not, which I believe was increasingly done so by ensuring a good series of theta’s, as the main point of the gabor filter is to capture oriented edges.