For our project, my main goals were to implement the [conv-relu-conv-relu-pool]xN -> [affine]xM -> [softmax or SVM] architecture, write a script to run cross-validation on our models and data, and create the ensemble to put our models together.

I used Keras to implement my model, and tested a variety of different layers and hyperparameters to fine tune my convolutional neural network. I followed the skeleton model provided for us, starting with one set of [conv-relu-conv-relu-pool], one [affine], and finally a [softmax] layer. While tuning the parameters for each of these layers and experimenting with what I could do to raise the overall accuracy, I tried to increase the number of times each of these setse appeared. However, adding more seemed to overfit on the training data so I ended up using what I originally had.

To further improve the performance, I wrote a script to cross-validate the model. I was able to get a minor improvement in testing accuracy when using five fold cross validation, however the improvement was not significant enough to warrant an almost five time increase in training time. We ultimately decided to not use cross-validation in our final product.

Lastly, I wrote an ensemble notebook where everyone could combine the results of their models. By averaging the predicted outputs of each model, I could create a final output array of our guesses for the classification of the image. This meant that errors in each of our individual models would likely be offset by the predictions from the others. Our ensemble managed to increase our test accuracy to a 0.9925.