Pedro Oporto and Matt McDonald

Although we are confident that our code will work after running many tests on smaller images/datasets, neither of us have computers powerful enough to run these tests with the data provided. The highest test we attempted was the full 24x24 images with only 10 of the 2000 images of the dataset, and even this took just under 28 minutes to complete the first round of testing predictors. Some quick math tells us that even the 400-picture dataset (40 times the size) through 5 rounds would take around 93 hours on our machines, which isn't particularly practical especially considering we both need our computers for other classes and daily tasks. With some better planning, we could have tried to find another system to run the code on, but as it stands we were unable to complete the accuracy testing or visualization steps.

We accomplished all of the tasks through the use of numpy functions and believe we have achieved efficient vectorization (tested both with apply_along_axis and vectorized functions), but were unable to achieve anything close to the 1 hour time you quoted in the assignment description. There is a possibility of memorizing some of the information returned by the predictors about the pixels, which by our best estimation would cut the time down by around 60% if it works as we suspect it might, but this still leaves the 2000 picture dataset taking around 150 hours!

We understand that this is almost definitely on us and there's something obvious that we're missing that makes it exponentially faster, but we could not come up with anything in time for it to help.