Computational Vision

Sheet 1 — CSCI 4270

Homework 5: Problem 1 Discussion

1. When considering the design of the SVM, I used the built-in functionality from scikit-learn. I decided to create a numpy file for each set of train, validation, and test image descriptors. After reading in the descriptors from these files and normalizing the data, I decided to just run the scikit-learn with a c=1 for the training data where I got a success rate of 69.9%.

When designing the code specifically, I created an SVM training function that can be called for training and validation purposes. For validation, I decided to adjust the single parameter of c for the SVM. The values I used were [.001, .01, 0.1, 0.5, 0.9, 5, 10], it was suggested we use a lowest value of c=0.1, but I expanded it since my validation was constantly selecting c=0.1 and I wanted to expand to see if I could get better results (which I did with c=0.01 which yielded a success rate of 79.6% on the validation data set). Then I simply used the selected model with c=0.01 for the test data, and found a test success rate of 67.3%.

Now analyzing the confusion matrices where the indices are in the order of ["grass", "ocean", "redcarpet", "road", "wheatfield"] classifications. We find that generally the models are very good at recognizing a redcarpet background (probably because it is very distinct in comparison to the others). It is also interesting to see how there are more false positives for grass than for any other classification. I wonder if this may be due to the way argmax functions where it selects the first index in the case of a tie breaker (see the final confusion matrix below, these trends are evident in all confusion matrices though).

```
Final Validation Success Rate: 0.796
Training Parameter: 0.01
Test Success Rate: 0.673
Confusion Matrix for C = 0.01
[[129   7   1   5   8]
 [ 33  96   1  13   7]
 [ 15  15 116   3   1]
 [ 33  12   4  88  13]
 [ 52   4   6  12  76]]
```

Also, I noticed a huge 15% increase in successful predictions when I standardized the data, which helped improve my results drastically.

Overall I am happy with these reasonable results. After speaking with some other students, they had a slightly higher final success rate on the test data (at almost 80%), which is definitely still comparable to my 67.3%.

Nicholas Lutrzykowski: lutrzn@rpi.edu