<descriptors.py>

Image class labels

1. Grass
2. Ocean
3. Redcarpet
4. Road
5. Wheatfield

Algorithm

- Generate descriptors for each image class in the training data
- Generate descriptors for each image class in the test data
- Stack the descriptors to form a full descriptor

Steps

- For every image in each image class
- Form blocks of 4 x 4 & calculate indices of the blocks corners
- Create histogram using Histogramdd for the blocks
- Form a one contiguous descriptor vector for the image
- Form a one contiguous descriptor vector for the entire image class

- Finally, use the Pickle module to output descriptors for train set and test set

<svm.py>

Used Linear Support Vector Classification (LinearSVC) and GridSearch
Linear SVC has a parameter C
The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

Steps

- Import train.pkl file, permutate and normalize the data (pseudorandom using seed)
- GridSearch over range C = [0.1, 0.2, …, 10.0]
- Get 5 different classifiers initialized with C that best classifies only one image class over the other classifiers
- Calculate normalized weight vectors and offset for each classifier
- Train on the test set with each classifier
- Make prediction on all images, calculate training error by compare predicted and actual
- Using descriptor from test.pkl, make prediction on test data
- Calculate percentage correct for each image class

Final results

img class	Training error	Test accuracy
Grass	0.197142	0.706667
Ocean	0.134285	0.746667
Redcarpet	0.018571	0.946667
Road	0.142857	0.746667
Wheatfield	0.215714	0.766667

Overall precision: 0.7882640759418699
Overall accuracy: 0.7826666666666666

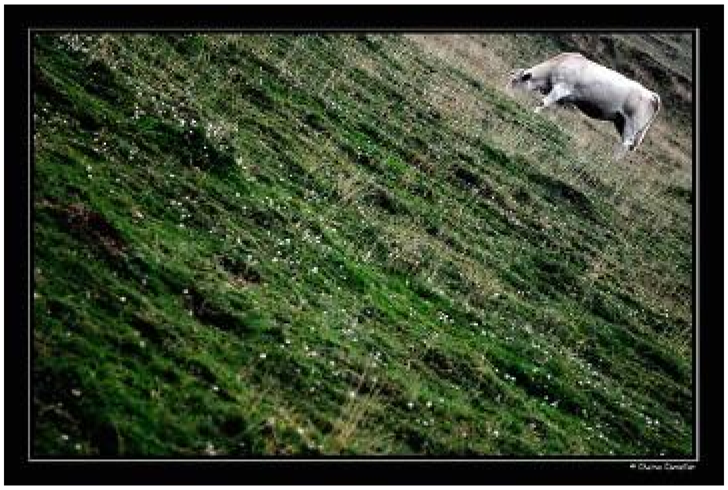
Confusion matrix (sklearn.metrics.ConfusionMatrixDisplay seems to be deprecated)

106	11	1	10	22
2	112	3	23	10
2	0	142	5	1
7	11	4	112	16
12	3	2	18	115

Strengths
Overall precision and accuracy are pretty high.
It performs especially well with the red carpets given the error and accuracy.

Weaknesses
The algorithm has a hard time distinguishing between grass and wheatfield
For instance, the image below should be classified as a wheatfield not grass

Below is an image of grass that might look like grassfield



I think it’s an inherent problem with this way of approaching the problem as it uses color histograms.
Also, the image below probably has more greens than reds when it should be classified as Redcarpet