Follow the Sun: Learning North from Images

Brandon Minor & Jack Morrison

December 9, 2013

Abstract

Humans have navigated by the sun for millennia, relying on its predictable path across the sky to determine their own heading relative to north. We have applied Machine Learning algorithms to this practice in an attempt to do the same through images with only RGB data. Data was collected from three different areas in the US, all with different latitudes. A number of classification and regression algorithms were run on the different data sets, with varying success.

1 Experiment Setup

1.1 Data Collection

Datasets consisted of a series of images taken from one spot while rotating at least a full 360° . To train our parameters, our datasets need to include the heading and timestamp of every image. We produced an Android application to facilitate this process. Images were captured as fast as possible by the phone's camera and saved on external memory. Every photo taken had a related orientation vector that the phone registered at that timestamp. This vector was produced by the phone's magnetometer and registered the difference, in degrees, of the heading of the phone with respect to gravity and magnetic north. Images were only captured if the phone was nearly vertical, i.e. pitch of the phone was within ±3° .

1.2 Programming

All Machine Learning algorithms used (described in Section 3) were part of the scikit-learn toolkit for Python. scikit-learn also performed cross-validation on datasets provided. All plotting was done in the matplotlib library.

2 Learning

2.1 Features

Brightness was the key component for all features. First iterations of testing were performed on full images, but this method was changed to analyzing images scaled down by a factor of 70 to increase performance and give a better ratio of features to datapoints. For these initial tests, the sum of the brightness of each row/column were used.

2.2 Cross-validation and Labeling

Cross validation was performed on all datasets. After experimenting with both n-fold and leave-one-out validation, we used leave-one-out for all of our tests.

Magnetometer data registered magnetic north as 0° of yaw, with the degree increasing clockwise around the compass rose. The yaw of each image served as the training label used in every machine learning process.

2.3 Classifier Algorithms

The main objective of this project is to formulate a reliable classifier; if we can label the orientation of one set of images with low range of error, we would hope the classifier would work similarly for other data sets. As such, each classifier breaks the 360 compass degrees into twelve 30-degree increments before analysis. We used two different algorithms for classification:

1. Support Vector Machines - We used polynomiall radial basis function, and linear kernels. Since there were twelve different incriments, the SVM treated the dataset as having 12 distinct classes.
2. Decision Forests

2.4 Regression Algorithms

Beyond this comparison of algorithms, we also performed regression function training. Our regression functions take our feature vectors and output an angle in degrees, with zero degrees being magnetic north. Our initial experiments used Linear Regression, using both LASSO and Ridge loss functions.

3 Testing and Analysis

3.1 SVM - Linear kernel

Linear kernel proved to be very unsuccessful for our purposes. Due to the

3.2 SVM - Ponynomial kernel, degree 3

3.3 SVM - RBF kernel

3.4 Linear Regression - LASSO vs. Ridge

3.5 Decision Trees

4 Further Work

4.1 More Features

1. Histogram of Oriented Gradients (HOG): Bins of pixels are formed into histograms representing the gradient of brightness levels in the bin. Feature count is dependent on the size of the bin.

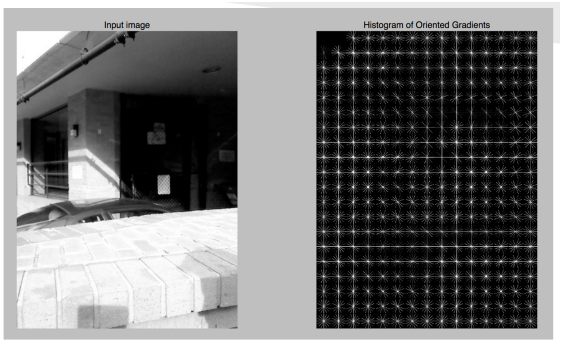

Figure 1: HOG

2. RANSAC and shadow detection: using a point cloud of dark pixels, RANSAC is performed to get a basic linear function that fits the trend. This is then compared to other photos, and similar trends are classiied similarly.

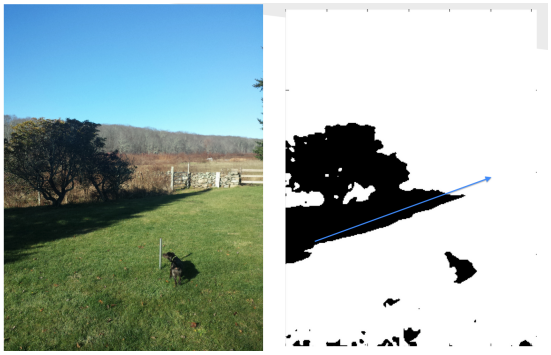

Figure 2: RANSAC (proposed)