Graham Skeats
Project 4 Report

For Project 4 I continued with my work to identify images that contained signs of agricultural activity using HOG descriptors and a support vector machine. To me the defining feature of agriculture are straight lines and corners where the forest has been cleared away to make a field for crops. The HOG (histogram of gradients) algorithm will similarly look for large changes (gradients) that are present along edges and other areas of abrupt changes in intensity. This will again exploit these straight lines, edges and corners, that otherwise wouldn't really exist in nature without human intervention. Potential issues that I think this approach might have is that rivers might be able to create straight or nearly straight lines naturally, so there might be additional preprocessing or feature extraction needed in order to prevent false positives.

The first of several changes to the work that I did for Project 3 was to decrease the size of the images that I was working with. While in Project 3 I worked with 750x750 images, I decreased the size to 250x250 which seems about the size of most of the fields and other agricultural features in the image. This served two important purposes, firstly it decreased the size of the set of features that HOG would generate for each patch which hopefully made any interesting features more salient for subsequent classification algorithms. In addition, it also allowed me to increase the size of my data set which is critical for supervised learning approaches. While in Project 3 I had 22 labeled images of non-agricultural jungle and agricultural landmarks, for this stage I was able to increase the size of my labeled dataset to 85 images. While selecting and labeling these images was tedious, it gave me the ability to have an appropriate number of training images to create a useful classification model and enough images to test that model's performance. Importantly, I made sure to prevent class imbalance by paying attention to the amount of each class (non-agricultural and agricultural) that was included in the dataset. I was able to split it 45-40 non-agricultural to agricultural.

After I had my data set I was extracted HOG descriptors from each image and split the data set approximately 77%-23% training images to test images. Passing these features to a support vector machine I hoped that it would be able to effectively classify these images to show whether or not they contained evidence of agricultural activity. Support vector machines are a machine learning classification algorithm that seeks to find a hyperplane (line) that separates two classes with the maximum margin of error between the two. In other words, it tries to pick the path of greatest difference between the classes. The initial performance of our SVM model with no regularization and a linear kernel hovered between 65-70% but would occasionally swing wildly and even dip below random guessing (50%). Interestingly since I shuffled the data randomly each time the model ran, the classification performance of the model fluctuated depending on what particular images where selected. This is indicative of overfitting, where the model becomes overly complex in order to fit all the training data points and loses its ability to generalize to the rest of the data. It is also indicative that if this models performance is going to improve it needs more data that is more representative or it needs some tuning.

The first thing I changed was the HOG extraction itself. Since the model wasn't performing especially well I thought that the HOG features may not be forming linearly

separable data sets. I first changed the cells_per_block parameter which sets how many local histograms are collected per area of normalization called a block. I hoped that increasing this parameter would increase the information gathered per unit of space but it actually degraded the performance of the model so I left it at 1,1 for further experimentation. Next I tried to decrease the pixels_per_cell again with the idea that it would increase the information gathered per unit of space and that extra information might let me separate the two classes. I found that this change drastically increased the time required to train the model but otherwise had little effect. Increasing pixels_per_cell again seemed to degrade the performance of the model with it fluctuating between 40% and 60% accuracy.

Running out of ideas to change my feature set, I turned my attention to the classifier itself. First, I altered my SVM by change my regularization or slack variable to see if that would help with classification. Slack variables work by relaxing the constraints the classifier needs to meet, I thought that by increasing my slack variable I could combat my overfitting problem and the model would generalize to the test set more accurately. This did improve the performance of the model with most runs at 75% accuracy and some reaching a high-water mark of 86.66%. I also confirmed this my decreasing the slack variable to .5 which increased overfitting and decreased the performance of the model to below random guessing at around 35%. I found that the right slack value was between 5 and 10. Next I tried applying different kernels to see taking the data to a different dimension would allow it to be more easily separated. Poly, rbf, and sigmoid kernels all produced similar results to the linear and none offered an improvement over the performance demonstrated so far.

Since I was confident that HOG should be able to provide a feature set for accurate image classification using SVM, I wrote a function oneatatime that iterates over the original satellite image in 250x250 images chunks and outputs a prediction based on the model I trained from the labeled data set. What I found was that the false positive rate for our classifier was extremely high, many images that contained only forest were being classified as agriculture. My explanation for this was that certain areas of the forest are composed of only one type of tree and as a result look ‘flat’ since the canopy is only one color. These ‘flat’ canopy areas are interpreted by the classification algorithm in the same way that the managed fields are and as a result give false positives.

Figure 1 You can see in this image how parts of the canopy look flat where they are all the same color and there are no distinctions between trees, these pattern trick the HOG and cause false positives.

This discouraged me from decreasing the size of my images further since I thought that would only exacerbate the problem as the 'flat' canopy areas would fill more of the frame. Interestingly, the classifier did any excellent job discriminating against areas containing the river which was surprising since rivers are by nature all one color and essentially 'flat' in the eyes of the HOG. To try and fight these false positives I altered my SVM so that it could compute probabilities that a particular sample (image) belonged to either class 1 (agriculture) or class 2(non-agricultural). Next, I implemented a check to see if the difference between the probabilities warranted a positive prediction. As you can see below, if the difference in probabilities wasn't greater than .25, the classifier wasn't very sure how to classify the image, so I threw it out. Additionally, since I focused on positively classifying the agricultural images I

```
def oneatatimeprob(svm_hog,list_images):
    for image in list_images:
        fd = get_hog(image_array=image)
        prediction = svm_hog.predict_proba([fd])
        if abs(prediction[0][0]-prediction[0][1]) > .25 and prediction[0][0]<prediction[0][1]:
            print(prediction)
            displayimage(image, "1")
```

Figure 2 Using probabilities that a sample belongs to a class to decrease the false positive rate.

included the constraint that as long as the classifier thought the image contained agriculture it could produce a positive result and throw out all the others. Since these images are mostly jungle, and most agricultural activity is concentrated together I was willing to trade some accuracy on fields that looked disused or overgrown for decreasing the number of false positives I got when considering the images which this strategy was able to effectively do.

This probability approach allowed me to reduce the rate of the false positives but it also reduced the rate that I identified true positives. As a result, accuracy didn't increase significantly and stayed around 80%. Next I tried using a random forest classifier to see if that would improve my results. Unfortunately the random forest classifier had similar to slightly worse performance than the SVM hovering in the 60-65 percent range and again fluctuating with each run. This finally forced me to come to the conclusion that HOG wasn't a useful feature extraction for this problem in the end. However, since I already had several weak classifiers working for HOG the last thing that I tried was to use majority vote, or ensemble classification. Basically, I just allowed several classifiers to run on the same image then whatever the majority decided would be the classification. Unfortunately the third classification algorithm I tried to use, k nearest neighbors, was so bad that I ended up ignoring it in the vote and only looking at the random forest and as a result it performed worse than the SVM or random forest since both had to agree with each other or it defaulted to a prediction of non-agriculture.

Even though I wasn't able to use HOG features to classify images very well I still wanted to return to my original problem of identifying illegal gold mines. Interestingly even though I had the same false positive issues that I had with agriculture, the same model was able to identify some illegal gold mines since, similarly to maintained fields, they represented 'flat' images in a jungle that is otherwise full of gradients. Given more time I'd like to revisit my choice of using HOG features as my feature extraction and spend more time developing an even larger and more comprehensive dataset. Similarly annotating the area that a particular

landmark is discovered would be a useful and interesting extension of this classification problem.