I. System Configuration

1. Hardware

   Intel Core i5 1.9GHz

2. Software

   Windows 10 pro (also tested on Ubuntu 16.04 to verify proper operation of absolute path variable)

   Python 2.7.14 Anaconda

   - Sklearn.(svm, cluster)
   - Numpy
   - OpenCV
   - Matplotlib
   - random
   - Sys, OS

II. Algorithm Architecture

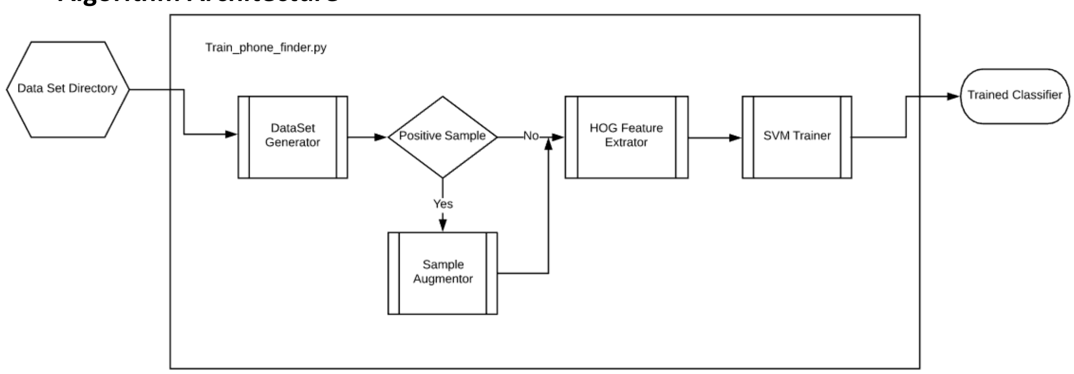

Figure 1. Process Model Classifier Training

For this problem I went with a Hog feature based SVM classifier. To make the dataset fit this method I wrote a script to cut a 50x50 pixel slice from the images centered at the location of the label associated with that image. I then augmented the positive sample set by rotating the image to generate a total positive set of approximately 44,000 images.

I randomly sampled 500 slices of the image background to generate a total negative sample set of 61500 images. I held 6 randomly selected images aside from this process as a validation set.

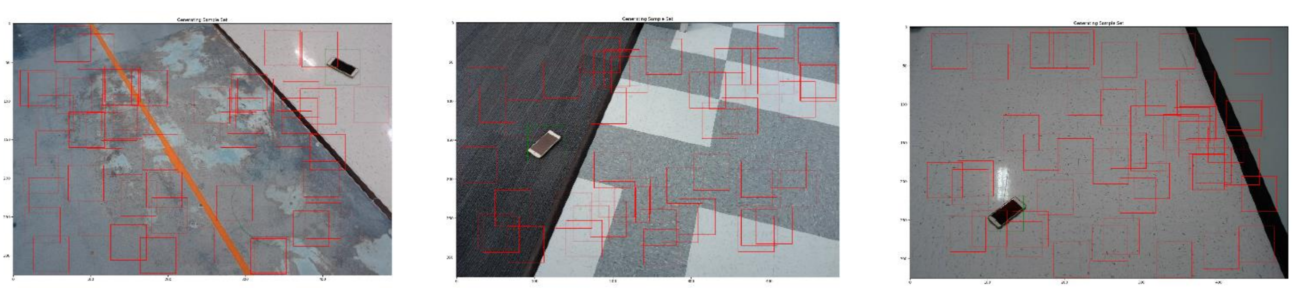

Figure 2. Examples of Training Sample Collection

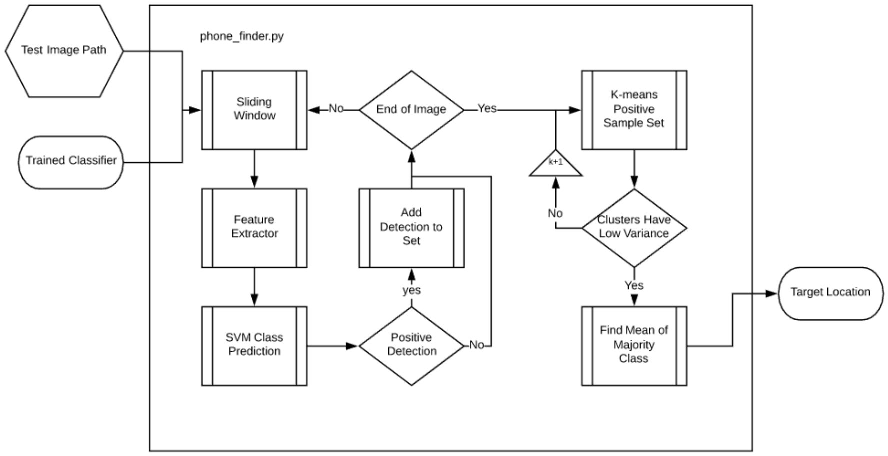

Figure 3. Process Model Classification

I choose to use HOG features because the nature of the problem lent itself well to geometric features easily optimized for using a gradient based extractor. This method however also catches false positives on areas of the image that have artifacts in the background of similar dimensions to that of the phone. Figure 4 displays the raw output of the sliding window based classifier.

Figure 4. Examples of Detection Sets

To zero in on a single detection I used the k-means clustering algorithm to determine the mean location of sets of detections. Then choosing the cluster associated with the majority class provides a single location for the positive detection. Figure 5 displays examples of final detections from the validation set. Figure 5(a) shows the only example from the validation set of false positive, as discussed above we see that an artifact in the background on the location of the detection is of a similar geometric structure to that of the phone.

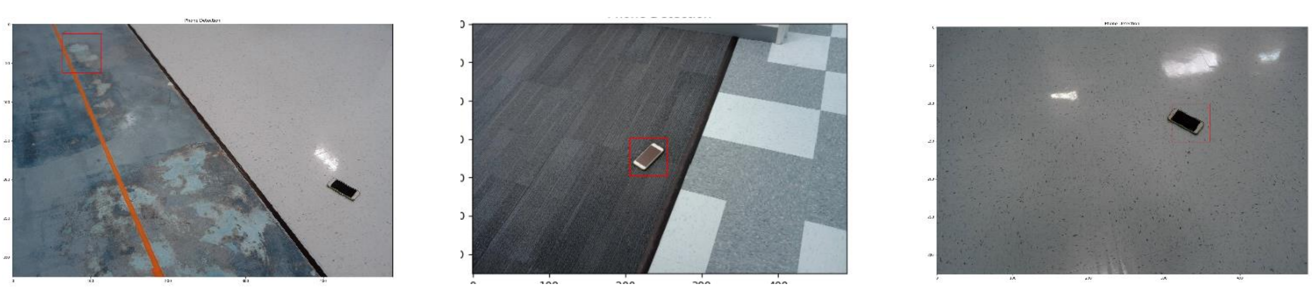

Figure 5. Examples of Final Detections

III. Drawbacks, Robustification

The main drawback is this is not at all generalizable. The dataset contains so few examples and the methods of positive set augmentation is such that this classifier is only suitable to images that are very similar to these examples. It is intentionally overfit to a very narrow scope problem and would require serious reworking of the method to accommodate statistical variations. If the dataset was larger and contained more classes I would have chosen a neural net pretrained on the imagenet dataset. From that method the last layers could be retrained to the classes specific to the problem.

Another drawback is that this method includes an inherent assumption that the test set contains targets of similar size relative to scene. This is due to a single size sliding window and sample set generator chosen for computational efficiency. Mitigating the second drawback is much easier than doing so for the first. Conducting a pyramid style sliding window would increase robustness to dimensional variations of a broader dataset.

There was also an assumption that isn't clearly stated in the problem statement that the test set contains only a single example of a target. The classifier chosen is SVC with radial basis kernel providing single [0,1] class decision and clustering algorithms used intentionally draws down to a single majority class such that if the image contains two or more positive examples of targets all but one are automatically discarded as false positives. This is also a solvable problem, if the classifier chosen was logistic regression based, a probabilistic threshold could be parameterized such that multiple examples could inherently be detected in each image and the clustering algorithm wouldn't be needed at all.