# ‘Find Phone’ Notes – Brain Corp Aaron Penne – October 2017

For this exercise, a Haar feature-based cascade classifier was used. This method detects objects by training a model to identify Haar-like features (line, edge, etc.). A sliding window is used to create what Viola/Jones [1] call an integral image which enables the speed of detection that this method is known for. For efficiency, the feature type in the submitted runs has been changed to Local Binary Pattern (LBP) [2], which has similarities to the Haar method.

Dataset preparation for this method involves creating a positive and negative dataset. The positive dataset is provided and contains images with the object. The negative dataset is generated by looping through each positive image, determining a rectangle that covers the object, and filling it with an adjacent portion of the same image.

Each object in each positive image is annotated using the provided labels.txt file and enforcing a similar sized rectangle across all images. The rectangle size is based on the assumption that all objects in the provided images are roughly the same size. The positive images, negative images, and annotations are the inputs needed for the OpenCV Haar cascade detector tools.

# Assumptions

* OpenCV is installed and is in the path. This can be installed using pip install –c menpo opencv
* Pillow (fork of PIL) is installed. This can be installed using pip install Pillow
* The object is roughly the same size in all images (for automated annotation).

# Results

A test set of 5 images was reserved during development. After successfully training the detector, these 5 were tested with reasonable results: 3/5 detected the phone correctly, 1/5 detected 2 phones in the image, and 1/5 detected no phone at all. Additionally, 5 images of my own iPhone were taken in various locations with a similar spread: 4/5 correct and 1/5 no detections.

# Possible Improvements

* A larger dataset of positive and negative images will improve accuracy.
* When making the negatives images from positives, a wider mask with feathered edges would give a more realistic appearance.
* All processes here are automated, at the cost of some accuracy. Assumptions that are made could be invalidated easily with a different dataset. A more robust solution would involve manually annotating the positive images.
* When making positive images, 10-20 clear pictures of the phone at various angles on a white background would be ideal. These images could then be pasted at various locations on a large benign set of negative images. This would bring the count of positive and negative images up as high as needed, allowing for greater accuracy in detection. This would also remove the need to annotate images manually.
* A naïve approach to selecting detected objects was used: the first detected region is selected as the estimate. It would be better to implement a weighted selection, or polling method to increase accuracy.

# References

[1] Paul Viola, Michael J. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 511-518.

[2] Shengcai Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. Li. *Learning Multi-scale Block Local Binary Patterns for Face Recognition*. International Conference on Biometrics (ICB), 2007, pp. 828-837.