

# Phone Detector Project

Shusen Lin, 1/10/2023

## 1. Models Requirements:

Numpy, OpenCV, Scikit-learn, Scikit-image, Pqdict

## 2. Approach

For this project, the database only has no more than 200 samples, which is not suitable for training the deep learning model. Instead, I choose to solve this problem with the traditional machine learning method. The first approach I used was the color classification and image segmentation to detect the phone. However, when I tried the grayscale or HSV color format, the classification result was not good because the phone does not have a distinguished color like a red stop sign, and the black screens are easily messed up with the dark background. Hence, I chose the feature that can significantly represent the phone: the shape. And **it led me to extract the phone features such as HOG (Histogram of Oriented Gradients), and I learned this classic HOG combined with the SVM method from N Dalal (2005) and made a simpler version for this project**, the model can detect the phone successfully and the accuracy is around 98.8%

## 3. Implementation

### 1. Generate samples

The good things about the labeled data are we know the ground truth center position of the phone, the phone size is relatively the same in the image, and the phone color is the same. **Hence, our model will focus more on the position of the phone.** To generate the positive samples, I located each phone position in the image and cropped the shot with 44x44 pixels as the positive samples (The largest crop I can do without exceed the boundary). I also rotated each positive sample in four directions to obtain more data. To generate the negative samples, I randomly crop 30 44x44 pixels windows in each image in the meantime avoiding the phone area.

### 2. HOG extraction SVM model

*Once the samples are cropped from the original image, I extracted the HOG features with parameters: `pixels_per_cell=(6, 6)`, `cells_per_block=(2, 2)`, then feed them into SVM model with default settings. And save the model as "trained\_model.sav"*

### 3. Bounding box and sliding window

A critical feature of HOG+SVM is applying the sliding window. The sliding window is usually combined with Gaussian pyramid scaling to capture objects of different sizes. As I mentioned before, the phones' sizes are similar, so we can only apply the window with the size 44x44 to detect the object without scaling down the image. When we chose the bounding box, **we only kept the box with the most considerable prediction accuracy.**

#### 4. Result

I randomly picked 26 images from the database and put them into the folder “find\_phone\_test\_images”, and train the model based on the rest data. **The final model accuracy is about 98.82%. And all the phones are detected.**

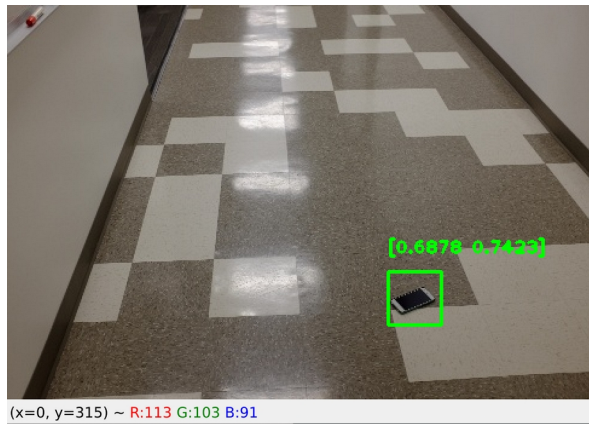


Fig 1. bounding box 54.jpg



Fig 2. bounding box 117.jpg

We can observe the model overcame the black strip in Fig 1 and the dark mat in Fig 2.  
*Model:*

positive samples: 420  
negative samples: 3150  
Image process took: 2.5293 sec.  
**training took: 6.9451sec.**  
**Single image detection took: 5.1516 sec.**

#### 5. Discussion

In general, the goal of this project achieved, the model can successfully detect the phone with a high accuracy. The further improvements can be:

1. Generate the CNN model with the HOG layer and even more to increase the accuracy.
2. The detection time is quite long. Consider scaling down the test image while predicting.
3. Consider adding the pyramid for phone size variation and any image preprocessing method can be explored to help train the model.

#### 6. How to run

```
> python train_phone_finder.py ./find_phone
```

The **train\_phone\_finder.py** will take the find\_phone folder to train and generate the model named “trained\_model.sav” which will be used later.

```
> python find_phone.py ./find_phone_test_images/51.jpg
```

The **find\_phone.py** can either take the single image to detect the phone position, or run the whole test folder to return the model accuracy.

**Further instructions are included in two .py files.**