谭树杰 11849060

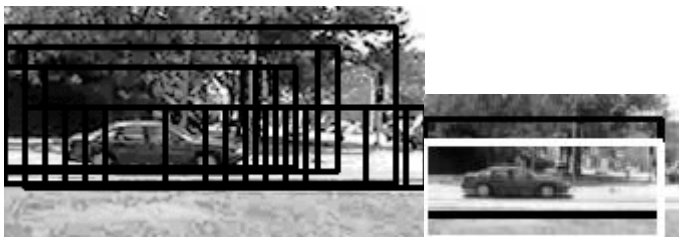
Pre Lab #2 SVM Report

Exercise

The code is shown below, you can see more details in SVM pre-Lab.ipynb

```
run extract_features.py -p ./CarData/TrainImages/pos/ -n ./CarData/TrainImages/neg
run train_classifier.py
run test_classifier.py -i ./CarData/TestImages/test-0.pgm -v
```

The detection results before NMS and after NMS are shown below respectively.



Question

1. HOG method is one of the famous techniques for object recognition and edge detection. Please illustrate characters of HOG and how it can be used for feature descriptors.

Answer: A feature descriptor is a representation of an image which only extracts useful information for our purpose and for computation consideration. The HOG uses the distribution of directions of gradients as features. As edges and corners' magnitude of gradients is large and they have many information about object shape, the gradients of the image are very useful. To get HOG, we should:

1. Preprocessing: Normalized color and gamma values is not necessary in HOG

2. Calculate gradient: we should compute both the vertical and horizontal gradient images

3. Compute the HOG of 8*8 cells. First, calculate the magnitude and direction by two images in step 2. Second, we could use 9 bins such as 0, 20, ⋯, 160 (the 20 and -20 are considered as identical. Then we could vote for the 9 bins by magnitude and direction images. For example, If the direction is 15 and magnitude is 4. We should add 1 to bin 0 and 3 to bin 20, proportionally to the reciprocal of the distance between 0, 15 and 20. Finally, we get a vector of 9 bins.

4. Normalization. We use 24 * 24 blocks, that means 9 cells per time, to form a vector of

size 9 * 9 = 81. Then normalize the vector.

5. The 81 * 1 vectors are concatenated into HOG feature vector.

2. Please illustrate how sliding-window algorithm and non-Maxima Suppression works in the task of object detection respectively.

sliding-window algorithm:

The sliding-window is a window to process the test image. We can a patch of images by sliding the window through the input image. Then we process returned images to detect the object. The code is shown below.

```
# from test-classifier.py
for y in range(0, image.shape[0], step_size[1]):
    for x in range(0, image.shape[1], step_size[0]):
        yield (x, y, image[y:y + window_size[1], x:x + window_size[0]])
```

If the step_size is too small, the algorithm would become costly. Otherwise, if the step_size is too large, the algorithm may have bad performance.

non-Maxima Suppression

The algorithm is used to merge all detections belong to the same object. We could first sort the detections according to their confidence score in descending order. Then we create a unique list to store processed detection, adding first detection to it. Compare other detections to the detections in the unique list in sorted order. If the area of overlap of two detections is greater than the threshold value, the detection with lower confidence score will be removed.

Reference:

1. https://www.learnopencv.com/histogram-of-oriented-gradients/