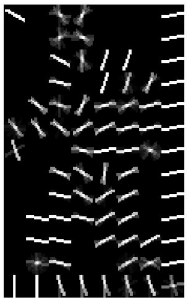
CS216 HW4 Name: Yifan Tian ID:78921267

The demo on detection of signs:

(a) Single scale detector output

- Template extracted from 1 positive example.
  The original image patch and HOG templates build on it:

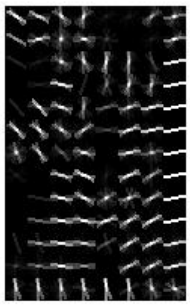


The detection using this template. Notice the signs on each part of image and the detection on edge of the image.

When I subtract the average of negative examples (remove background affect). It improves a lot.

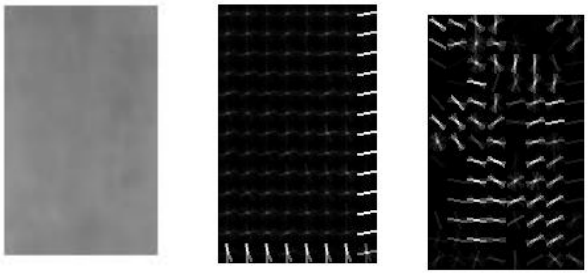
- **Template which is average of 5 positive examples**

The average image, and the average of HOG templates:



Template which is an average of 5 positive examples minus average of 100 negative examples

The typical average of negative patches, and improved template by subtracting negative average.

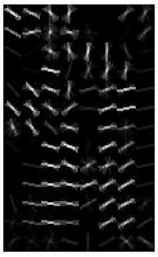


Below are detection using this template. It detects all signs at similar scales.

(b) Multi-scale detector output using whichever template works the best

Detection 1.

The positive images:



The template I use:

Detection 2. To detect small sign images on “test3.jpg”. First trial.

Here I use templates built by using positive examples from different images.

The positive examples I use:

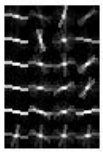
The improved templates from

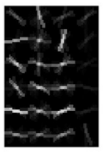
subtracting negative examples: 

By tring different scales, I get this detection. So we can detect the signs at different scales. **Notice the small people on top and on the left distant signs.**

Detection 2, second trial.

All positive examples are from the original images. Images below are positive examples, the original template and improved template by subtracting average of 100 negative examples. Because this time I use very small positive examples, so the template is smaller than the template I use in first trial.





Because the template is very small, so if I do not subtract the average negative examples, the affect from background and noise is very obvious. I cannot any examples:

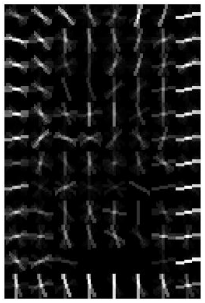
The demo on detection of faces:

(a) Single scale detector output

- **template extracted from 1 positive example. The face and HOG template built on it**

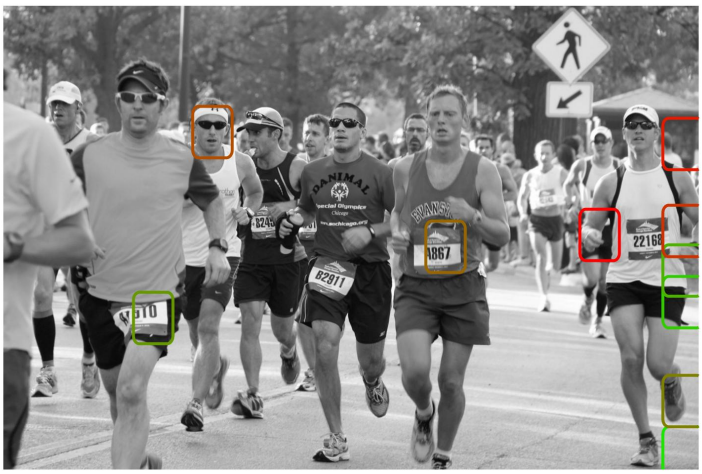
When subtract negative examples. It becomes better, can detect more faces.

- **template which is average of 5 positive examples.**
  The average image and the average of HOG templates：



The detection using above template

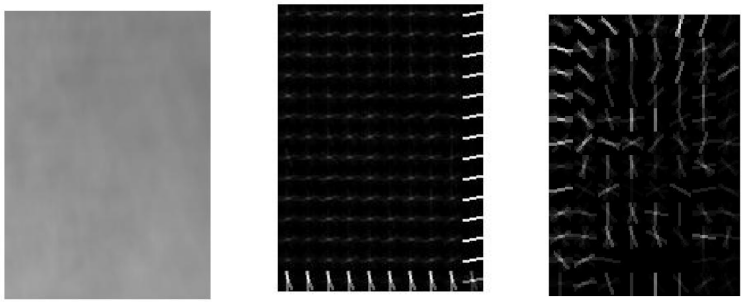
Because the weak features due to **average and** strong background noise effect and the trivial features- strong feature along the edge). The raw average template works poorly. Even can not detect any face on original image by 5 detections. When I increase detects to 10. The first face detection (positive detection) shows up (image below).



- **Template which is an average of 5 positive examples minus then average of 100 negative examples.**

Following shows One example of sampling negative examples, we need to avoid positive examples. Here I use 1000 just to show the effect:

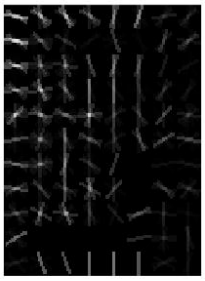
Following images are the average of negative examples. The HOG feature of the average, the improved template by subtracting this negative average.



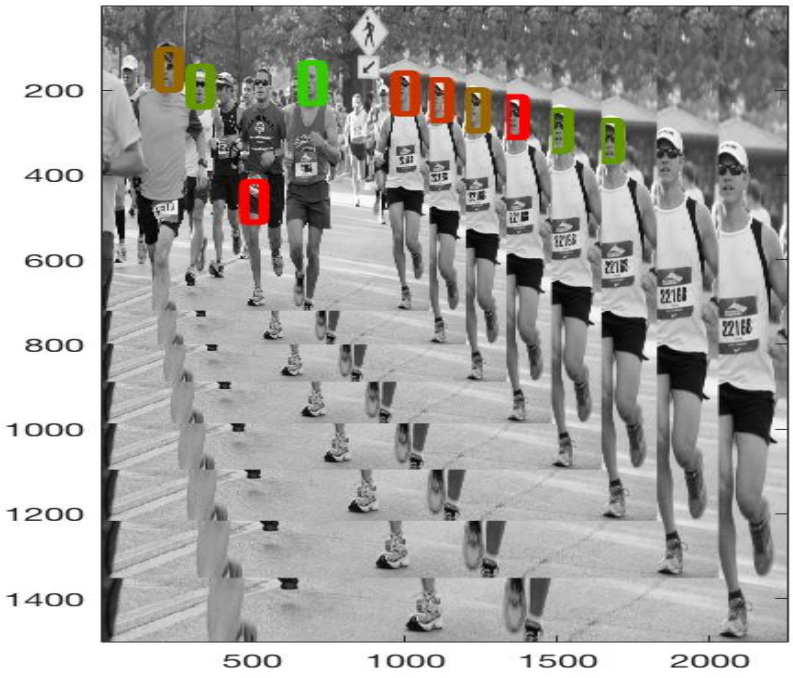
By subtracting the negative examples, I think it helps get rid of the affect from background and noise and trivial features (the direction features along the edge.) Now the template works much better. And also better than just using one positive example (image below).

(b) Multi-scale detector output using whichever template works the best

Positive examples and improved template:



Below shows the sequence of scales and detection on each scale, we can see that each scale has different detection result. We need to combine them and choose stronger detection.



To improve the detection performance, I finely tune the scales. To detect small faces, I first amplify image by a factor (such as 1.5), then we do down sampling for several loops until the image is so small that we know for sure that at this scale there should be no detection. For the image below. I first amplify image by 1.5, then I use 7 loops, every time scale down by 0.9. To finely detect a small range of images. Because from the picture we know that all faces all much small than the whole image. Then I got following detection results:

Why are we able to use cross-correlation here:

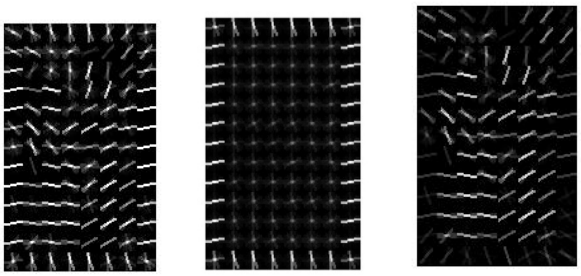
Answer: previously, we just use pixels as features, Then the feature is affected by the absolute brightness of images a lot. So we need to use SSD to improve the detection. Here the main difference and modification is that we use **the Histograms of Gradient Orientations** as features to feed into our cross-correlation. The Histograms of Gradient Orientations provides better description of features. First, it is a high order features, it uses gradient to build histograms, the histograms contain information about edges and corners, so the average and statistics I think is more stable. For example, the sky or road with small changes will not contribute to features as they do not contain edges or corners. Second, by subtracting negative averages, we can further improve the template, because it gets rid of the affect of noise and trivial features that all examples shares. Third, by using multi-scaling, the information of edges and corners is still useful in different scales. So we can use one effective template to detect objects in different scales. Also, this structure is similar to the most powerful objection detection techniques-convolutional neural networks. First we use filters to calculate gradients, then

we use gradients as feature to get Histograms (another layer of convolution), we use pooling(scaling) to detect object in different scales. So I think the powerfulness of CNN is consistent with the effectiveness of our strategy here.

Appendix:

Here I use raw derivative filter: X: [1 -1], Y: [1; -1]. If I use Gaussian derivative filter: I got very similar result for Histograms of Gradient Orientations. Thus detection result. The only difference is that Gaussian derivative give you more symmetric HOG on edges of template. But seems it does not matter to much as it will be removed after subtracting negative average. And you can change the choice of derivative filter in "mygradient.m" easily.

The HOG from Gaussian derivative filter:



The detection result for Gaussian derivative filte

Supplements:

1. I place all detection results in directory for reference.
2. In order to have several sign instances on one image, I create test8.jpg which is a combination of several images that contains sign instances. But actually all detectors work pretty well when there is an obvious instance. Here I show detection result on images when there is one obvious instance(ndet=1)

Also with ndet = 5