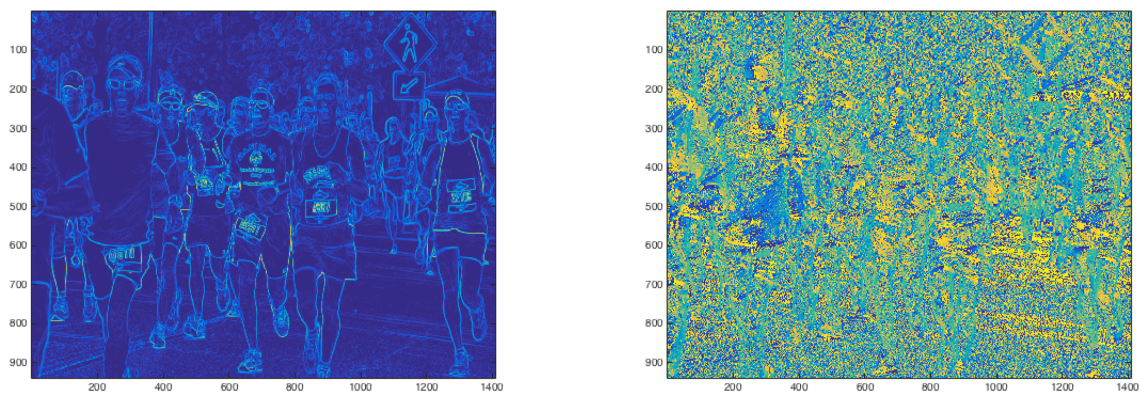
16-720 Computer Vision: Homework 5
Image Understanding

Anbang Hu
anbangh@andrew.cmu.edu
Due: April 12, 2016

1 Image Detection Using Histogram of Gradients

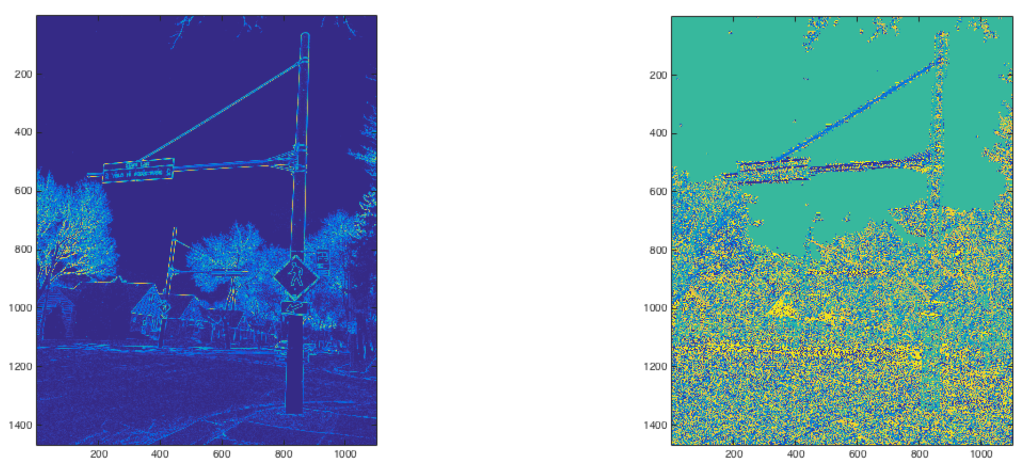
1.1 Image Gradient (5 points)

Implementation is in `mygradient.m`. Instead of using `imfilter` with `'replicate'`, I chose to use `imgradient` function to obtain magnitude and orientation directly. Magnitude and orientation of images `test0.jpg` and `test3.jpg` are shown in Figure 1,2. `q1_1.m` is used to generate the visualization.



(a) magnitude of `test0.jpg` (b) orientation of `test0.jpg`

Figure 1: `imagesc` visualization of magnitude and orientation of `test0.jpg`



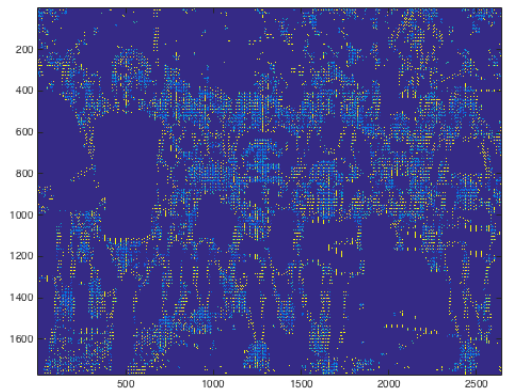
(a) magnitude of `test3.jpg` (b) orientation of `test3.jpg`

Figure 2: `imagesc` visualization of magnitude and orientation of `test3.jpg`

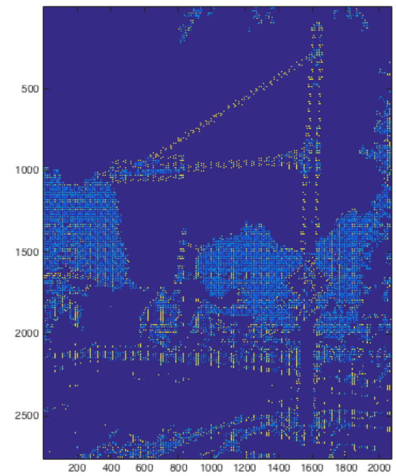
Anbang Hu
anbangh@andrew.cmu.edu

1.2 Histograms of Gradient Orientations (20 points)

Implementation is in `ohist.m`. I chose to implement the "unsigned" binning. Visualization of HOG features of `test0.jpg` and `test3.jpg` are shown in Figure 3. `q1_2.m` is used to generate the visualization. Figure 4 shows the visualization of HOG features of pedestrian crossing sign in `test3.jpg`.



(a) Visualization of HOG features of `test0.jpg`



(b) Visualization of HOG features of `test3.jpg`

Figure 3: Visualization of HOG features

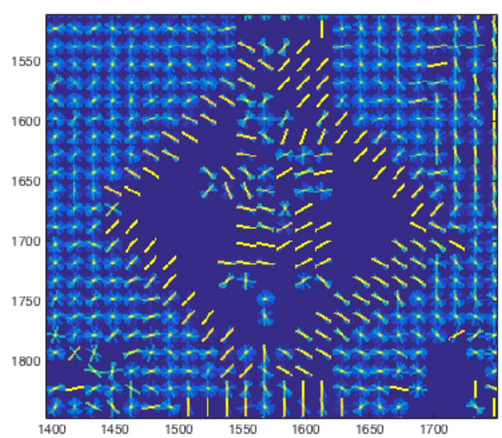


Figure 4: Visualization of HOG features of pedestrian crossing sign in `test3.jpg`

Anbang Hu
anbangh@andrew.cmu.edu

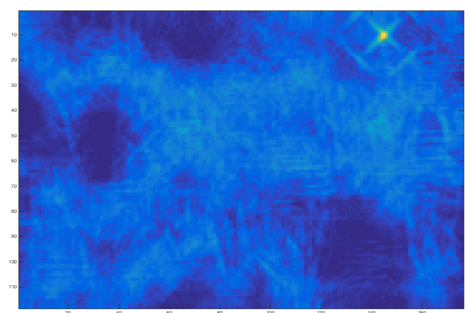
1.3 Detection (20 points)

The implementation is in `detect.m`. Using `test2.jpg` as training image, pedestrian crossing sign in `test2.jpg` as the template (Figure 5 shows the training image and the template extracted from it), the heatmap visualization of correlation and detected window of all other test images are shown in Figure 6,7,8,9,10, 11. `q1_3.m` is used to generate the visualization. In this question, only single detection is shown.

(a) Training image `test3.jpg`

(b) Template from training image `test3.jpg`

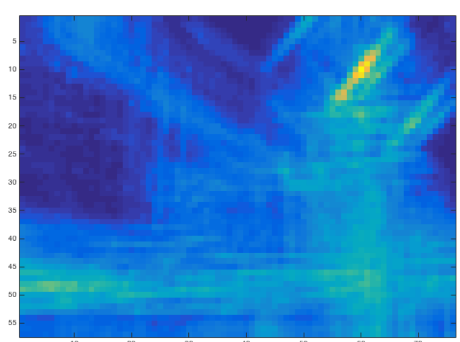
Figure 5: Training image `test3.jpg` and template from it



(a) Heatmap visualization of correlation of `test0.jpg`

(b) Detected window of `test0.jpg`

Figure 6: Heatmap visualization of correlation and detected window of `test0.jpg`

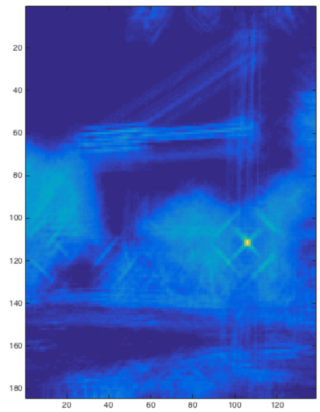


(a) Heatmap visualization of correlation of `test1.jpg`

(b) Detected window of `test1.jpg`

Figure 7: Heatmap visualization of correlation and detected window of `test1.jpg`

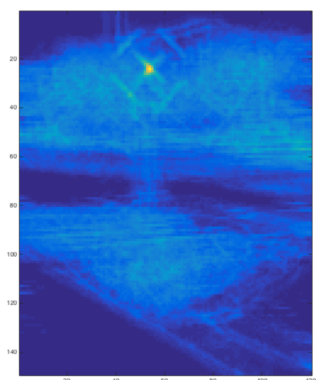
Anbang Hu
anbangh@andrew.cmu.edu



(a) Heatmap visualization of correlation of `test3.jpg`

(b) Detected window of `test3.jpg`

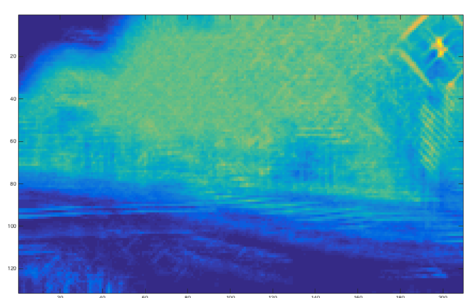
Figure 8: Heatmap visualization of correlation and detected window of `test3.jpg`



(a) Heatmap visualization of correlation of `test4.jpg`

(b) Detected window of `test4.jpg`

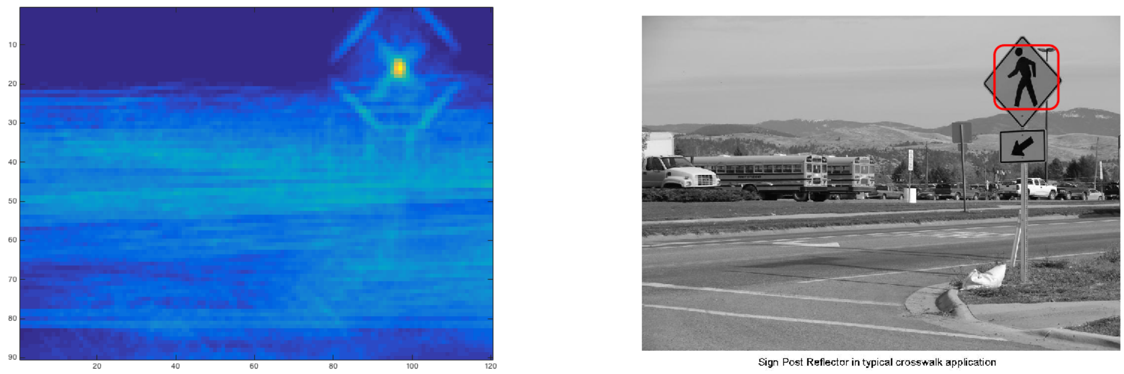
Figure 9: Heatmap visualization of correlation and detected window of `test4.jpg`



(a) Heatmap visualization of correlation of `test5.jpg`

(b) Detected window of `test5.jpg`

Figure 10: Heatmap visualization of correlation and detected window of `test5.jpg`



(a) Heatmap visualization of correlation of `test6.jpg`

(b) Detected window of `test6.jpg`

Figure 11: Heatmap visualization of correlation and detected window of `test6.jpg`

1.4 Extra Credit - Multiple detection (5 points)

Still using `test2.jpg` as training image and pedestrian crossing sign as template, multiple detections (5 detected windows in this case) in `test3.jpg` are shown in Figure 12. `q1_4_1.m` is used to generate the visualization.

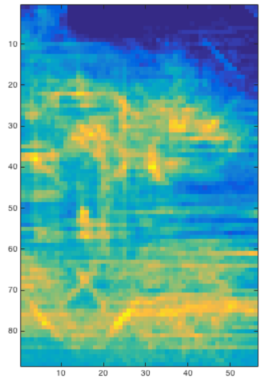
(a) 5 detected windows in `test3.jpg`

(b) 5 detected windows in `test5.jpg`

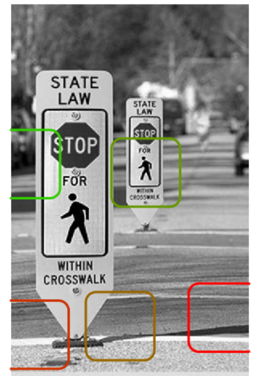
Figure 12: Multiple detections in `test3.jpg` and `test5.jpg`

I found and tested some image from internet. The results are shown in Figure 13. `q1_4_2.m` is used to generate the visualization.

I used pedestrian crossing signs from `test0.jpg` and `test2.jpg` as templates. Multiple detected windows (though not all correct) in `internet2.jpg` are shown in Figure 15. `q1_4_3.m` is used to generate the visualization.



(a) Heatmap visualization of correlation of `internet2.jpg`



(b) Detected window of `internet2.jpg`

Figure 13: Heatmap visualization of correlation and detected window of `internet2.jpg`

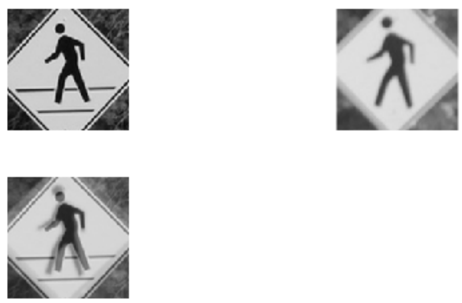
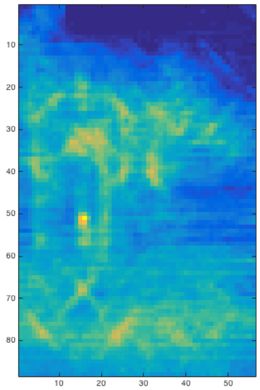


Figure 14: First two are templates (one from `test2.jpg`, one from `test0.jpg`), the third one is average of the first two templates

2 Learning Templates

I chose to build template for pedestrian crossing sign. Since there is only `test3.jpg` that contains multiple pedestrian crossing signs, I used another image `internet2.jpg` from online as another test image (shown in Figure 16). In section 2 (excluding extra credit), I use `test3.jpg` and `internet2.jpg` as test images.



(a) Heatmap of `internet2.jpg`

(b) Detected windows of `internet2.jpg`

Figure 15: Templates and detected windows of `internet2.jpg`

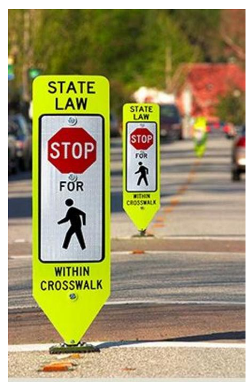


Figure 16: Test image `internet2.jpg` from internet

2.1 Select Patches (5 points)

Implementation is in `select_patches.m`. Positive template images are stored in `template_images_pos.mat`. Negative template images are stored in `template_images_neg.mat`. The five positive images and templates are shown in Figure 17,18. `q2_1.m` is used to generate this figure.

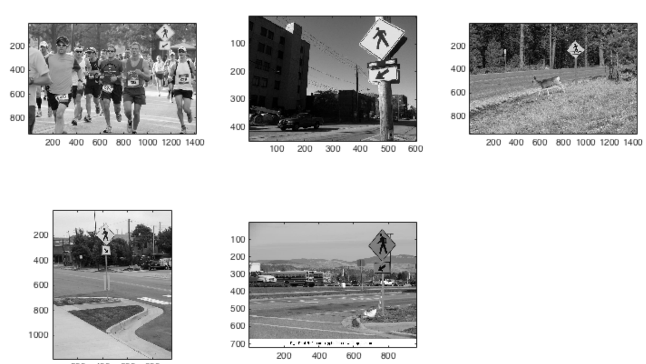


Figure 17: 5 positive images

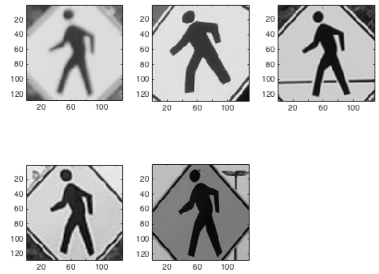


Figure 18: 5 corresponding positive templates

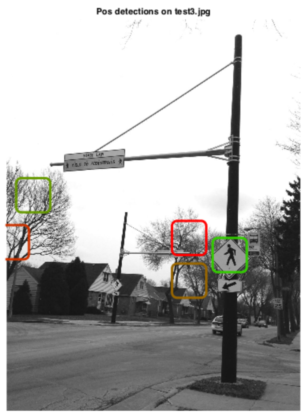
2.2 Positive Template Learning (5 points)

Implementation is in `tl_pos.m`. Detection results with 1 positive template are shown in Figure 19. Detection results with 5 positive template are shown in Figure 20.

(a) Detection results for `test3.jpg` using 1 positive template

(b) Detection results for `internet2.jpg` using 1 positive template

Figure 19: Detection results using 1 positive template



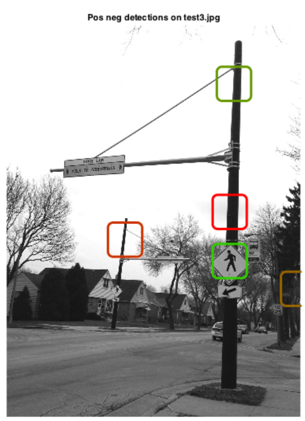
(a) Detection results for `test3.jpg` using 5 positive templates

(b) Detection results for `internet2.jpg` using 5 positive templates

Figure 20: Detection results using 5 positive templates

2.3 Positive Negative Template Learning (5 points)

Implementation is in `tl_pos_neg.m`. Detection results are shown in Figure 21.



(a) Detection results for `test3.jpg` using 5 positive templates + 100 negative templates

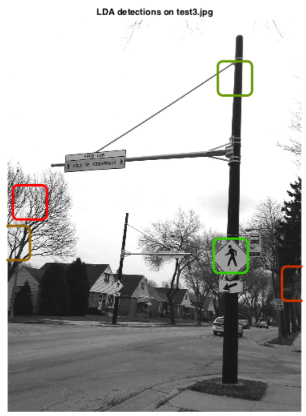
(b) Detection results for `internet2.jpg` using 5 positive templates + 100 negative templates

Figure 21: Detection results using 5 positive templates + 100 negative templates

Anbang Hu
anbangh@andrew.cmu.edu

2.4 LDA Template (20 points)

Implementation is in `tl_lda.m`. Detection results are shown in Figure 22. λ is chosen to be 0.05.



(a) Detection results for `test3.jpg` using LDA with 5 positive templates + 100 negative templates

(b) Detection results for `internet2.jpg` using LDA with 5 positive templates + 100 negative templates

Figure 22: Detection results using LDA with 5 positive templates + 100 negative templates

2.5 Multi-scale Detection (20 points)

Implementation is in `multiscale_detect.m`. Detection results on `test3.jpg` and `internet2.jpg` are shown in Figure 23. We can see that all targets (including every small targets) in `test3.jpg` are successfully detected; the red box, i.e. least confident window, captures something else. Since in `internet2.jpg`, there are only two targets of slightly different sizes, the most confident windows capture these targets and other less confident windows capture something else.

Note: Because the detector needs to search over a pyramid of images and do NMS to obtain the final result, it takes longer time to return.

(a) Detection results for `test3.jpg` using multiscale detection with template from LDA

(b) Detection results for `internet2.jpg` using multiscale detection with template from LDA

Figure 23: Detection results using multiscale detection with template from LDA

2.6 Extra Credit - Mixture of Templates (10 points)

Implementation is in `q2_6.m` and `multiscale_detect_mixture.m`. 5 positive templates for front face are shown in Figure 24.

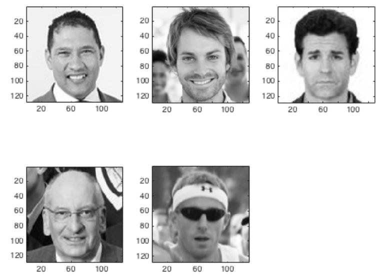


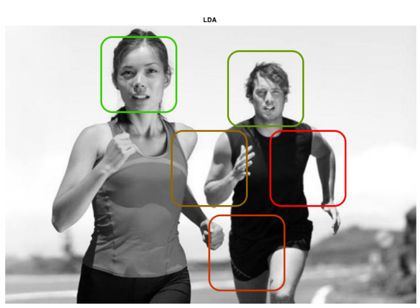
Figure 24: 5 positive templates for front face

First of all, I performed detection using positive template learning, positive negative template learning, LDA template learning, and multiscale template learning. The results are shown in Figure ??, which shows that there is a gradual improvement along the way.

(a) Detection results for `testface1.jpg` using positive template learning

(b) Detection results for `testface1.jpg` using positive negative template learning

Figure 25: Detection results using positive, positive negative template learning



(a) Detection results for `testface1.jpg` using LDA template learning

(b) Detection results for `testface1.jpg` using multiscale template learning

Figure 26: Detection results using LDA, and multiscale template learning

We can see that multiscale scheme works best. However, if we apply multiscale scheme to an image with side face, it fails to work (shown in Figure 27).

Figure 27: Front face multiscale fails to capture side face

Then I tried to build of mixture of models which contains templates of both front face and side face. 5 positive template for side face are shown in Figure 28.

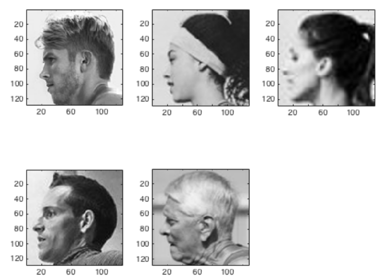


Figure 28: 5 positive templates for side face

Using NMS competing between side face and front face, I obtained the detection results as shown in Figure 29

Figure 29: Mixture multiscale captures side face

Although not as ideal as expected, the second best window captures the side face in test image. Based on the result, there is still room to improve. But due to lack of time, I left where it is.

3 A Few Words

Probably because the pedestrian crossing sign is a bit easy, positive template learning using 1 positive template has already performed well on objects of similar sizes in test images. When using 5 positive templates, the performance slightly improves in `test2.jpg` with more correct bounding box. Positive negative template learning and LDA template learning give similar performance as positive template learning using 5 templates. However, with fixed window size, smaller or larger object cannot be accurately captured. Therefore, multiscale method comes to rescue. using template from LDA, I upscale the image by factor of 1/0.7 up to 6x, and downscale the image by factor of 0.7 to right before the image is smaller than the template. Running detection algorithms on this pyramid of images and combining them with non-maximum suppression gives the best detector with a single template. The reason is that it not only includes all the advantages of LDA, but also exploits HOG features of different image sizes (object sizes) so as to detect object with different sizes.

More generally, positive template learning usually fire on background as well. Positive negative template learning improves upon positive template learning by subtracting the background information. The resulting detection performs relatively well. LDA template learning introduces covariance matrix into the equation, which helps remove the naturally occurring correlation between the HOG features, leaving behind just the discriminative gradients. This method should perform better. But since positive negative template learning usually does a good job, the improvement from using LDA is indistinguishable. Multiscale template learning builds a pyramid of images and run detection algorithm on each level of the pyramid. In the end, detected results are combined via NMS. Multiscale template learning is powerful in that it can detect object of different sizes. In extra credit, I built a template for human face, and carry out the series of detections performed previously. The detection results showed a better progression of improvement along the way. I also tried mixture model in extra credit. Although the result is not ideal as expected, one of the top five windows does capture the side face in the test image. Figure 29 shows that the best window include a patch that contains legs and background grass. This is probably because of the effect of variations in illumination.

The best detector in my implementation is multiscale detector. In comparison to using only cross correlation, this multiscale detector has several advantages. Averaging positive templates gives a more general template that can match to images patches with some variation. Incorporating negative templates adjusts pixel weights, reducing background noise and making the detector more robust. The detector uses templates generated from LDA method, which eliminates naturally occurring correlation, leaving behind useful discriminative information. The Gaussian pyramid in the detector enables it to find variable size objects if there are any.