1) Download a color image from the web. Run the SLIC superpixel segmentation algorithm provided in Matlab or Python and experiment with the target number of superpixels and compactness. Display and discuss your results. [3 pts]

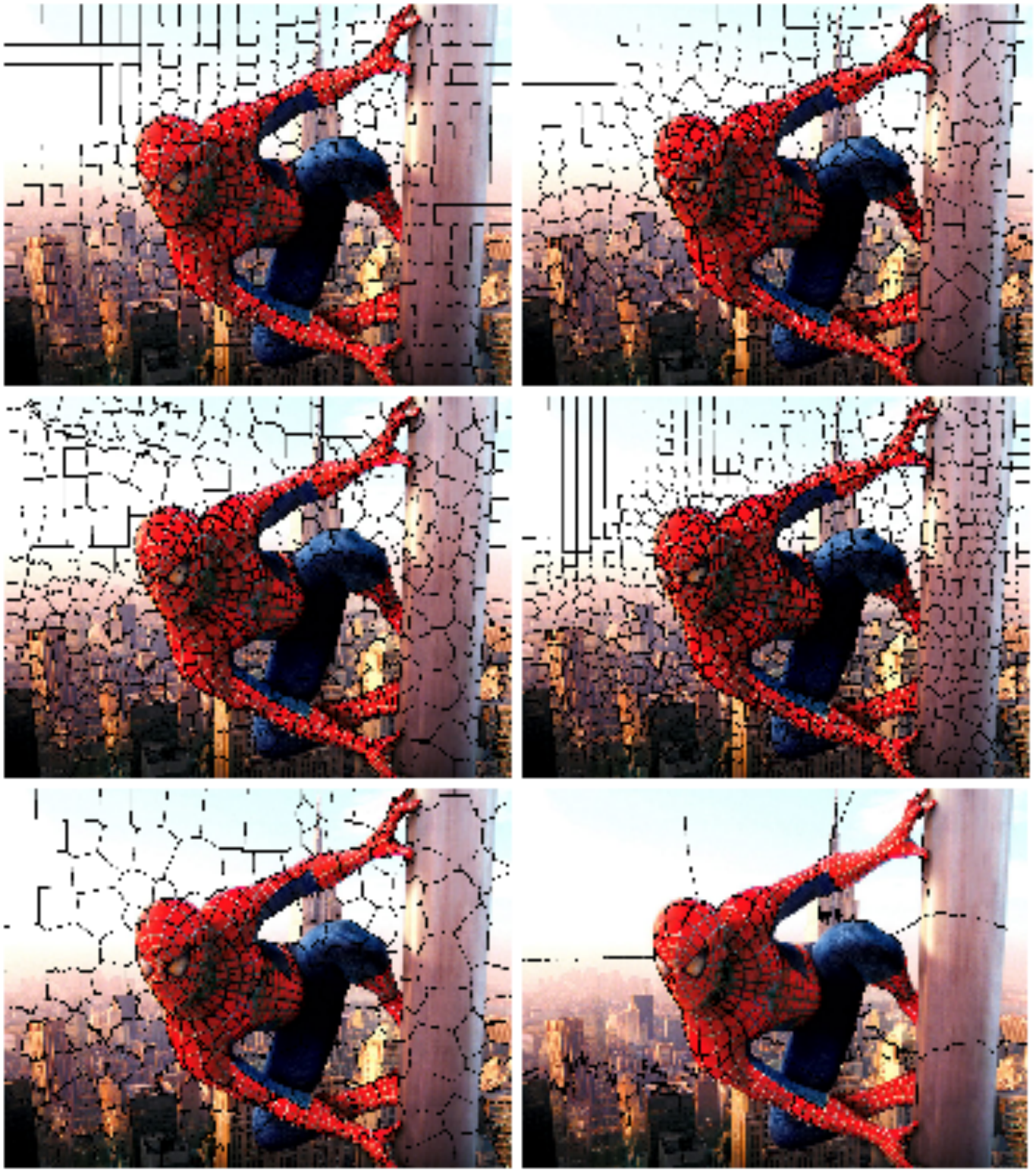
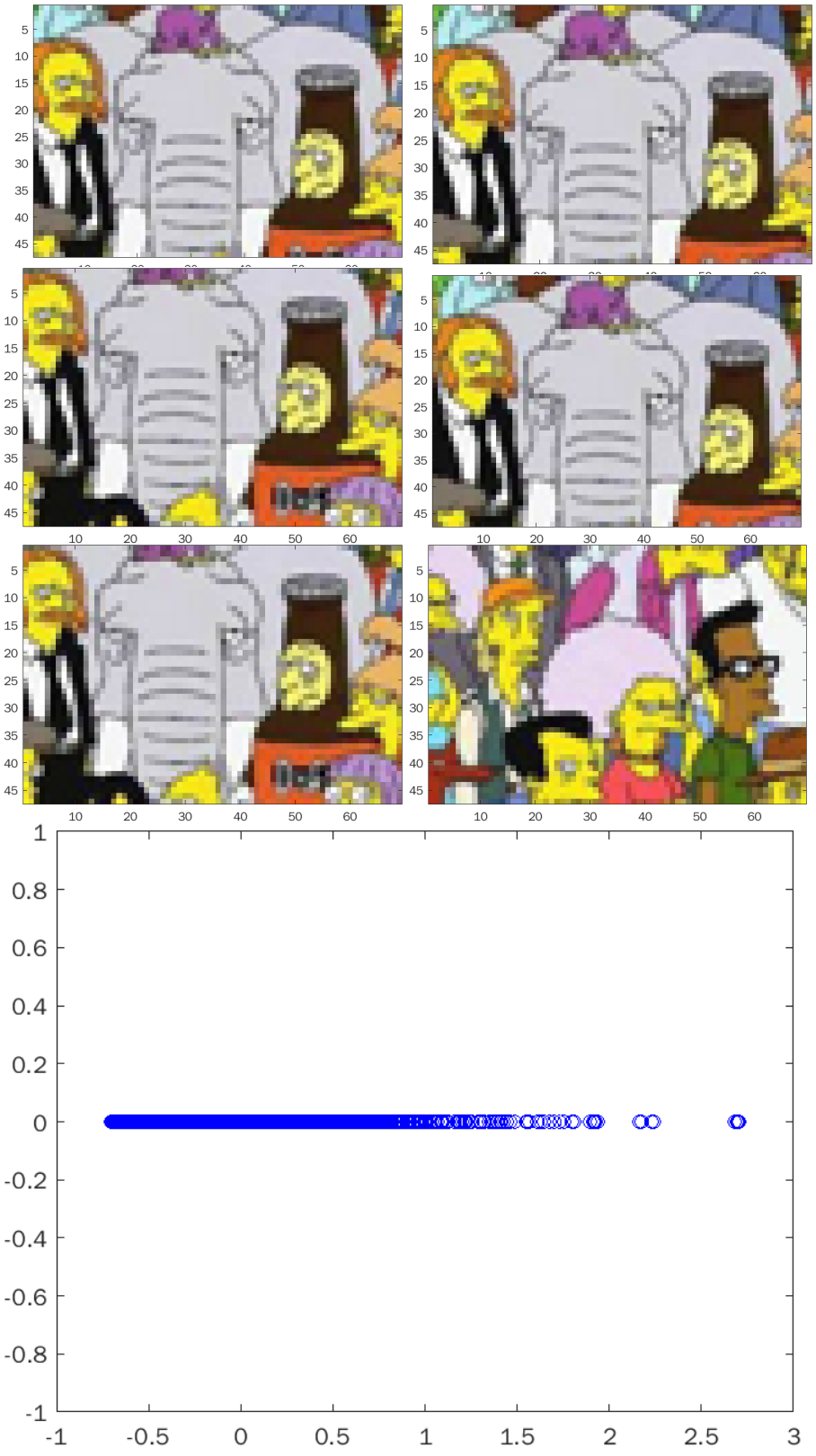


Image	Superpixels	Compactness	Numlabels
P1	500	100	494
P2	500	20	486
P3	500	1	472
P4	1000	20	995
P5	200	20	185
P6	10	20	12

Comparing P1, P2 and P3, compactness has little effect on Numlabels. When the compactness become large, the superpixels will be more like square.

Comparing P2, P4, P5 and P6, when superpixels is larger, the mesh patterns on spider-man's clothes are segmented better.

2) There's an elephant in the room. Can you find it? Search for the "template" **template.png** in the "search image" **search.png** using color-based NCC (make sure the standard deviation is "unbiased" with N-1). (Note: the template did NOT come from the search image.) Assume the origin is in the center of the template image for each approach (Note: there should be a border around the search image where the metrics cannot be computed). Sort the resulting scores from best to worst. Plot all of the sorted scores (1-D plot) and show the patches corresponding to the 1^{st}, 2^{nd}, 5^{th}, 10^{th}, 100^{th}, and 500^{th} closest matches. Compare the results. [5 pts]


5 10 15 20 25 30 35 40 45
10 20 30 40 50 60
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1 1.5 2 2.5 3

The 1st image is closest to what we're looking for. Although 2^{nd}, 5^{th}, 10^{th}, 100^{th} also contain an elephant, the position of elephant's hat and the position of the man on the left indicate that the center of the images has been shifted. The 500^{th} image doesn't include an elephant because it's too low on the list.

3) As usual, submit your material to Carmen.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 1
myIm = imread('hw7.jpg');
figure
%different campactness
N = 500;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 100);
BW = boundarymask(L);
subplot(2,3,1), imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

N = 500;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 20);
BW = boundarymask(L);
subplot(2,3,2),imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

N = 500;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 1);
BW = boundarymask(L);
subplot(2,3,3), imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

N = 1000;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 20);
BW = boundarymask(L);
subplot(2,3,4), imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

N = 200;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 20);
BW = boundarymask(L);
subplot(2,3,5), imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

N = 10;
[L, NumLabels] = superpixels(myIm, N, 'Compactness', 20);
BW = boundarymask(L);
subplot(2,3,6), imshow(imoverlay(myIm,BW,'black'))
disp(NumLabels)

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 2
```

```
clear; close all;
OriIm = double(imread('search.png'));
T = double(imread('template.png'));
[yO,xO,zO] = size(OriIm);
[yT,xT,zT] = size(T);
n = yT*xT -1;

Center = round([yT xT]/2) ;
i = 1;
for r = 1: (yO-(yT-1))
    for c = 1:(xO - (xT -1))
         P(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:zO);
         if r < Center(1) || c < Center(2)
            P_center_r = Center(1) - r;
            P_center_c = Center(2) - c;
         end
         sum = 0;
         for z = 1: zT
             meanP = mean(P(:,:,z),'all');
             meanT = mean(T(:,:,z),'all');
             denominator = std(P(:,:,z), 0,'all') * std(T(:,:,z), 0,'all');
             for y = 1: yT
                 for x = 1: xT
                        %numerator = (P(y, x, z) - mean(P(:,:,z),'all')) *
(T(y, x, z) - mean(T(:,:,z),'all'));
                        numerator = (P(y, x, z) - meanP) * (T(y, x, z) -
meanT);
                       % denominator = std(P(:,:,z), 1,'all') * std(T(:,:,z),
1,'all');
                        sum = sum + (numerator/denominator);

                 end
             end
         end
         resule_c(r,c) = sum/n;
         result_l(i) = sum/n;
         i = i+1;




    end
end



sorted = sort(result_l, 'descend');

fprintf('%.3f',sorted(1));
[r,c] = find(resule_c == sorted(1));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
imagesc(uint8(result_Im));
axis('image');
pause;


[r,c] = find(resule_c == sorted(2));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
```

```
imagesc(uint8(result_Im));
axis('image');
pause;



[r,c] = find(resule_c == sorted(5));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
imagesc(uint8(result_Im));
axis('image');
pause;

[r,c] = find(resule_c == sorted(10));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
imagesc(uint8(result_Im));
axis('image');
pause;

[r,c] = find(resule_c == sorted(100));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
imagesc(uint8(result_Im));
axis('image');
pause;

[r,c] = find(resule_c == sorted(500));
result_Im(1:yT,1:xT,1:zT) = OriIm(r:r+(yT-1),c:c+(xT -1),1:3);
imagesc(uint8(result_Im));
axis('image');
pause;


plot(sorted, 0 * sorted,'ob');
```