ECE 280L – Fall 2020

Mike Keohane (mbk28)

Image Processing

Contents

1 Exercise 1

Changing p changes the rate at which the color fades from the center. The larger p faded to black quicker than the smaller one.

2 Exercise 2

The image is a pixelization of random colors. It looks like static. The colors change randomly each run, but it still just looked like a colorful ”static”.

3 Exercise 3

The 5 point moving average filter is smoother than the two point filter which makes sense because it is averaging over more points. The two point filter is only creating lines with two points which is why the edges are so ’sharp’.

4 Exercise 4

The 51 point approximation resembles the graph more closely. The 11 point approximation has unacceptable error because the slope of the graph changes a lot between each point in the approximation.

5 Exercise 5

The 10x10 blur is just a consistent blur in all directions. The 50x2 blur is blurred mostly vertically where is looks like the coins slid up/down in the picture. The 2X50 blur is blurred horizontally where it looks likes the coins are sliding horizontally in the picture.

6 Exercise 6

When using ’valid’ for convolution, only the values where there is full overlap with the shifted and flipped h are returned which results in some cropping of the images. ’same’ on the other hand creates filler 0s on the edge of x in order to keep the size of the return the same as x when convolved with h.

7 Exercise 7

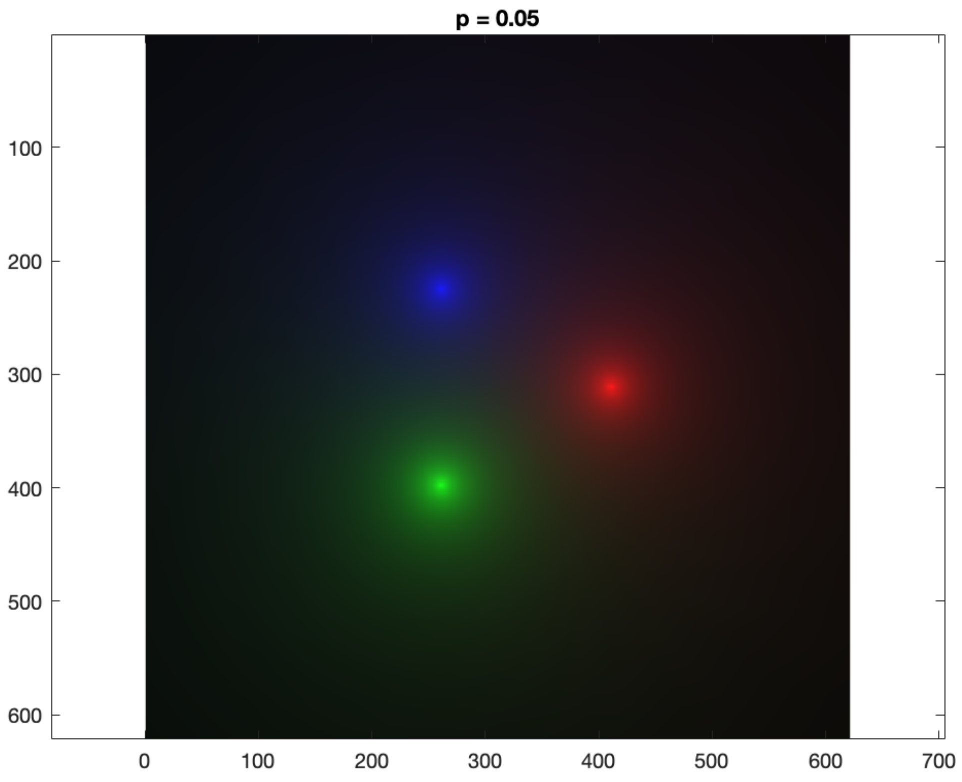
1. Coin Gaussian Blur- Blurs the entire image in a translucent like lens.
2. Coin Vertical Edges- Highlight the left side and shadow’s the right side of the vertical edges which because the coins are circular, are semi-circular edges.
3. Coin Horizontal Edges- Highlight the top side and shadow’s the bottom side of the coin.
4. Coin Edges- Highlights the edges of the coins and shadows the rest in black. This method found the edges more clearly than the Alternate Coin Edges.
5. Alternate Coin Edges- Highlights the edges but shadows the rest in grey. This method found the edges very easily but doesn’t display them as clearly as the last one.

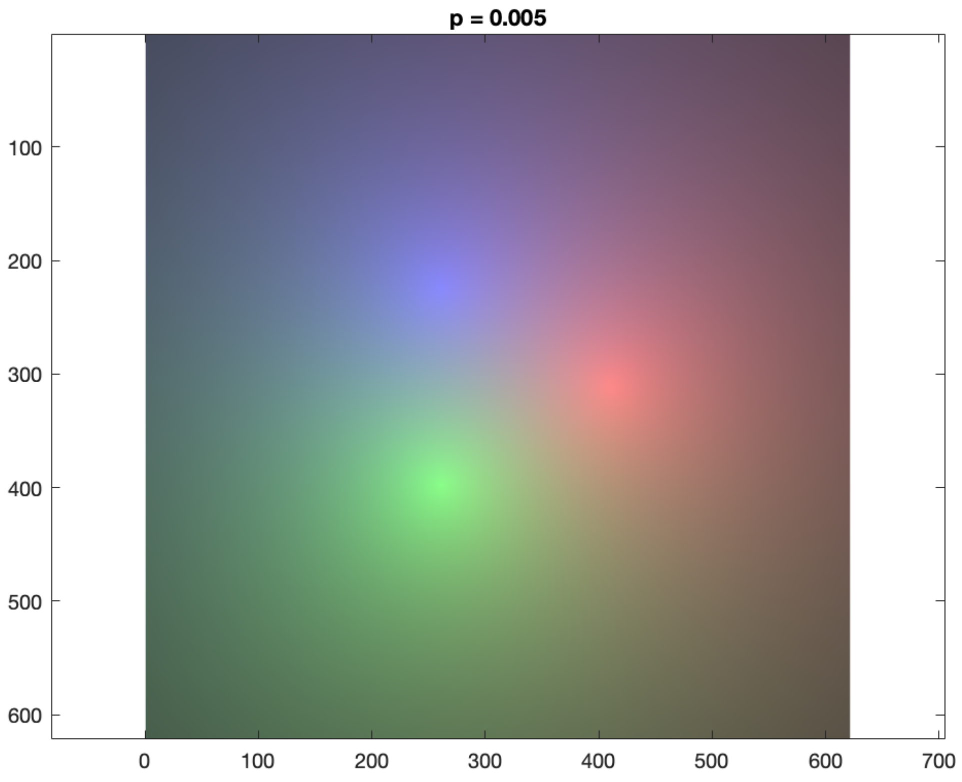
8 Exercise 8

The box blur blurs the the test card evenly causing a clearly blurred image. The Gaussian Blur on the other hand doesn’t appear as Blurred which is because it blurs with a low standard deviation. The Sobel vertical edges and horizontal edges are clearly shown in each case. The normalized absolute Sobel Edges clearly shows all the edges of the original test card. It looks like it un-filled all of the colors.

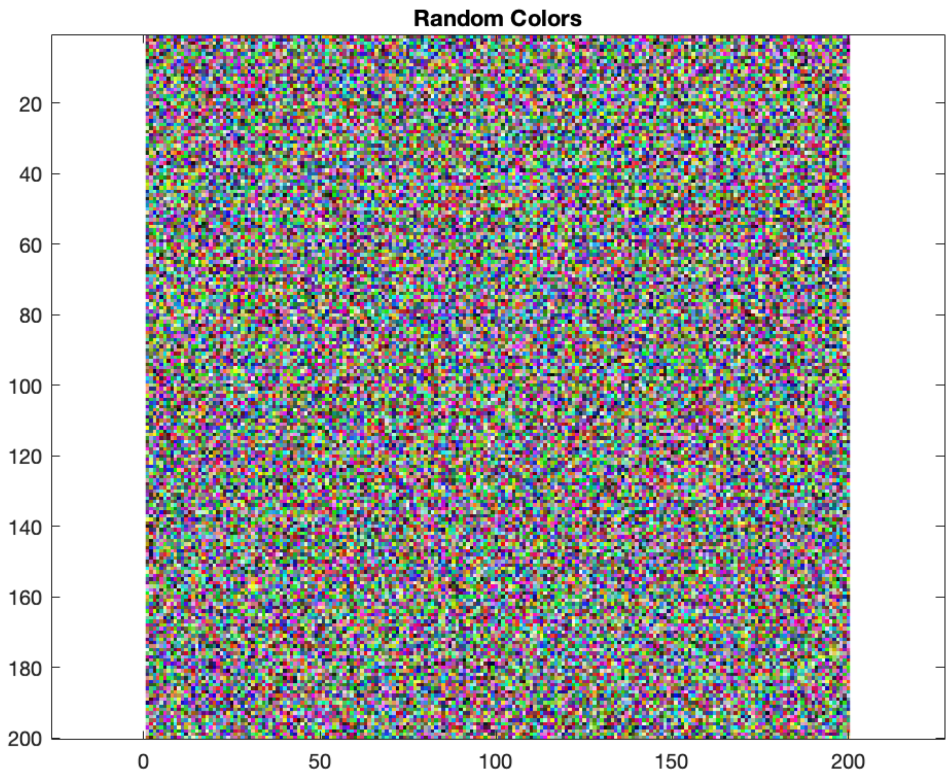
9 Graphs

9.1 Exercise 1

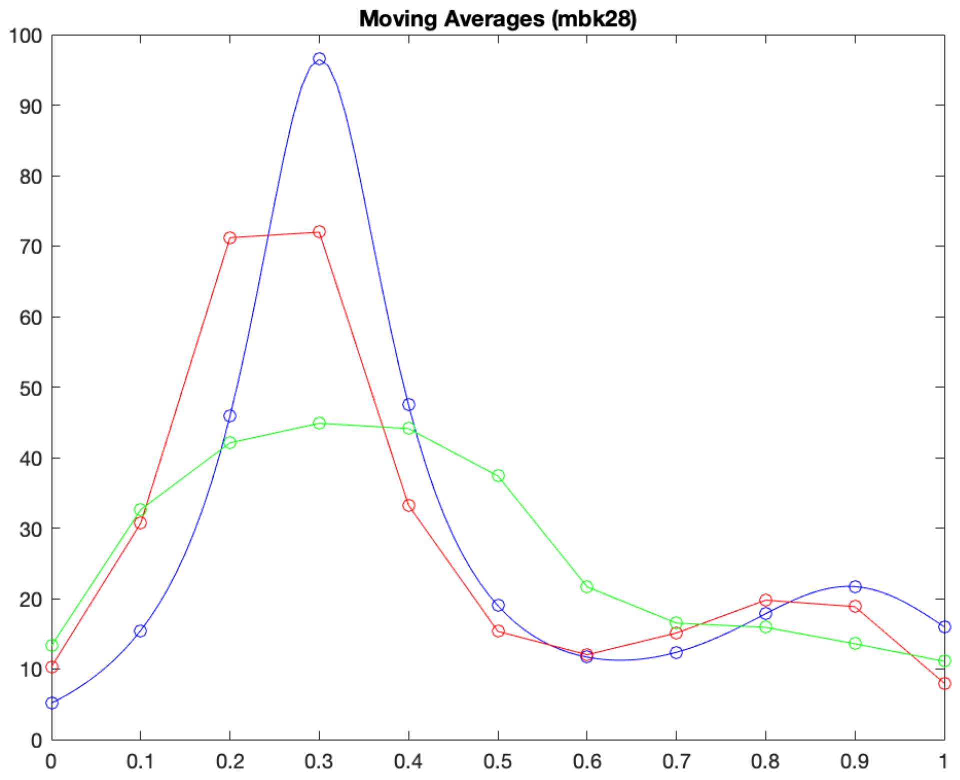




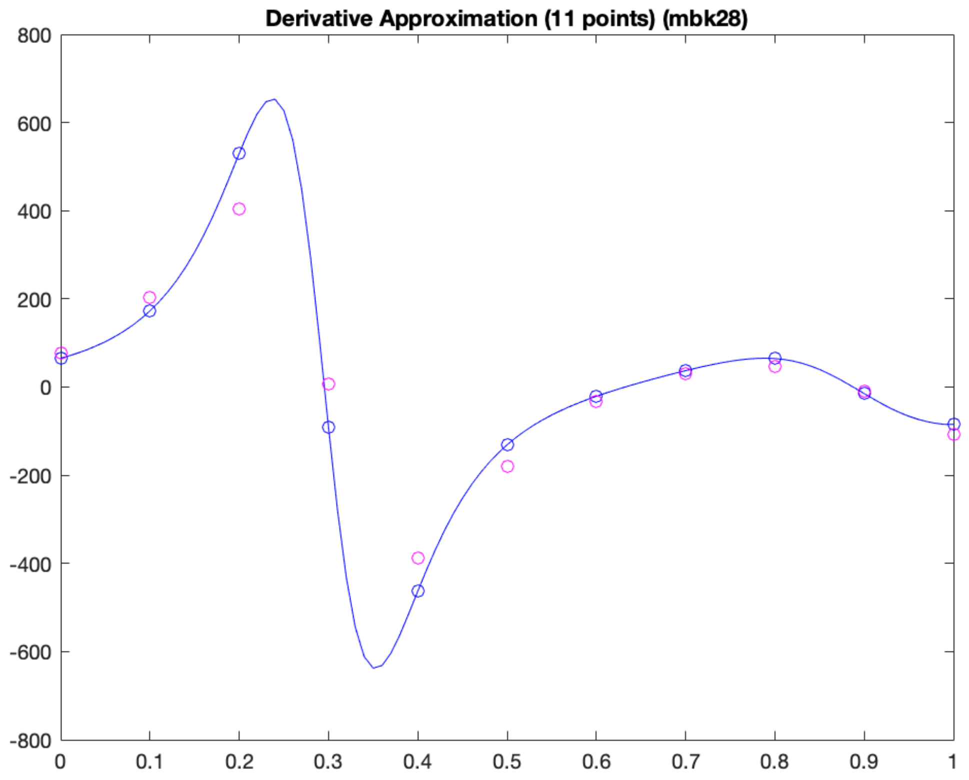
9.2 Exercise 2

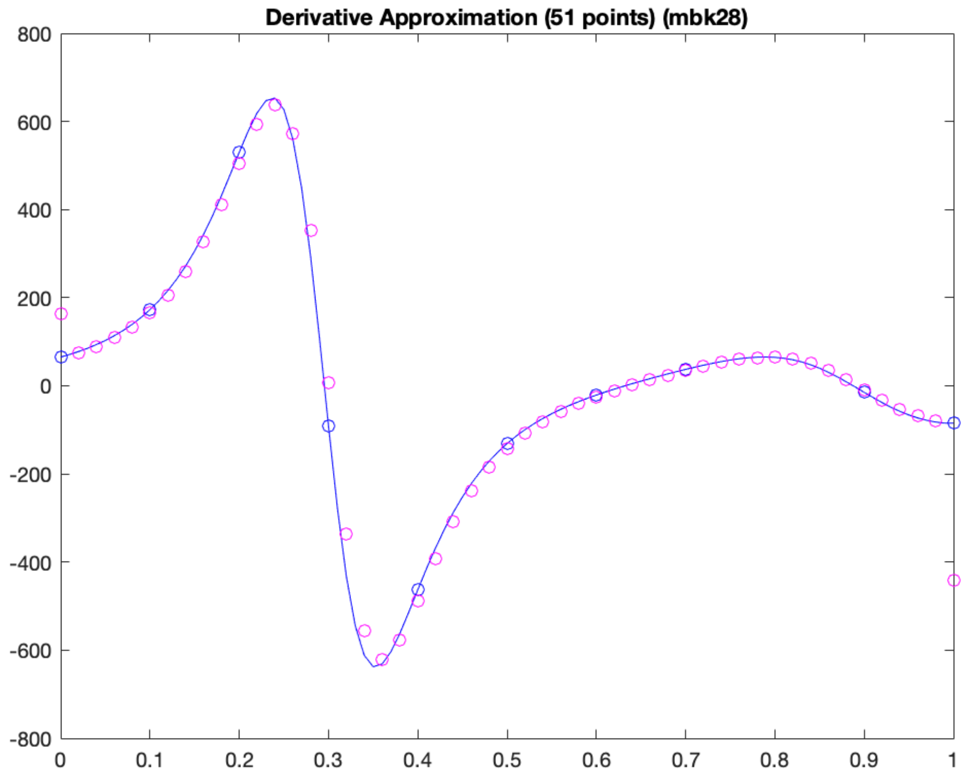


9.3 Exercise 3

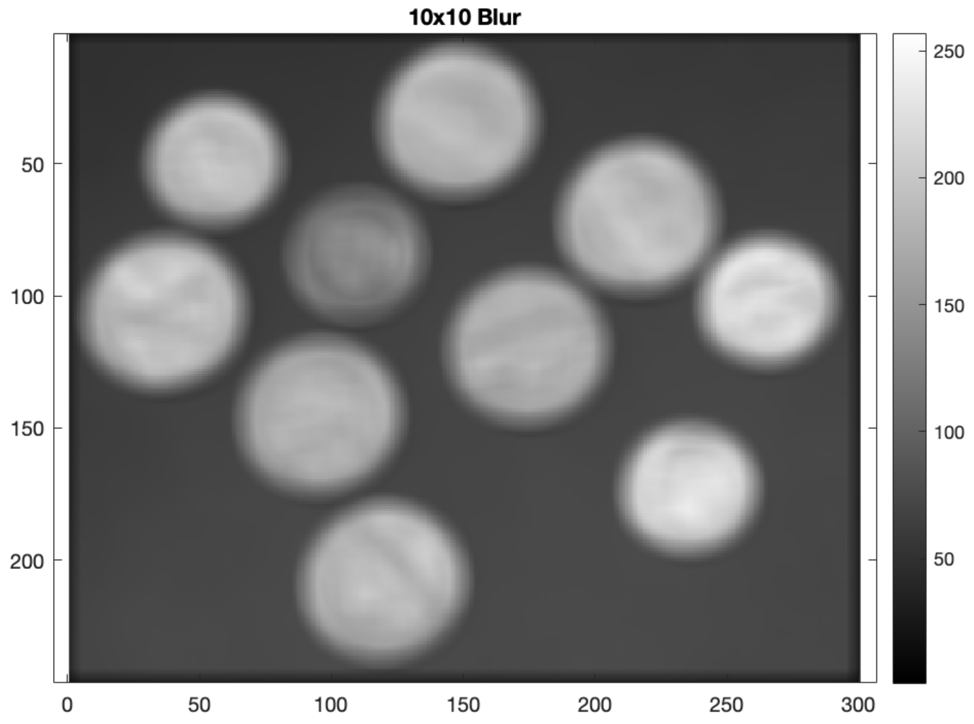


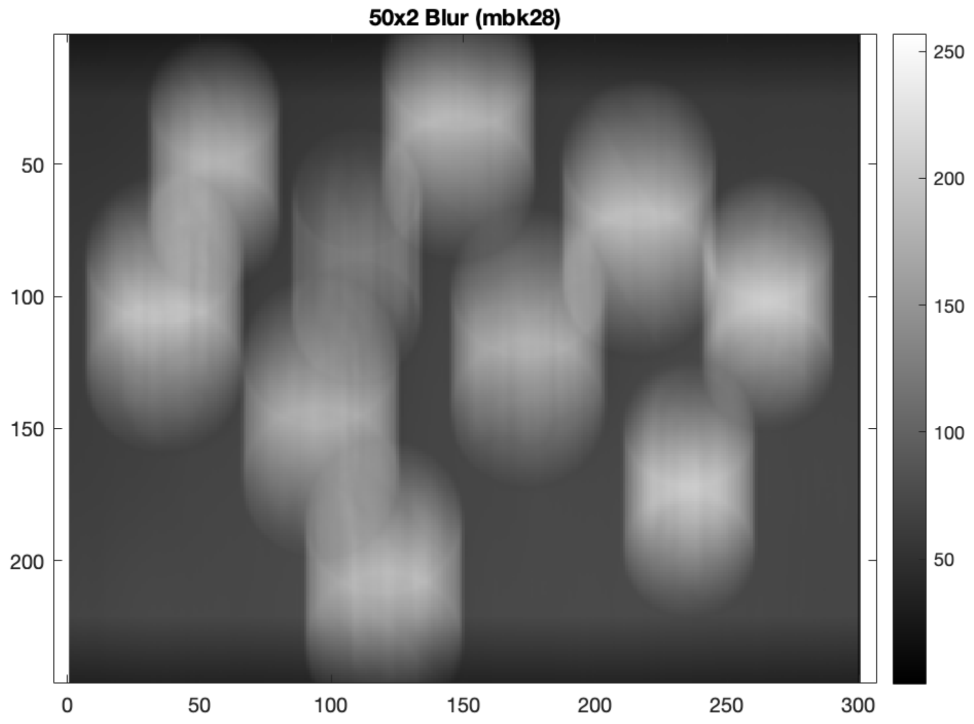
9.4 Exercise 4



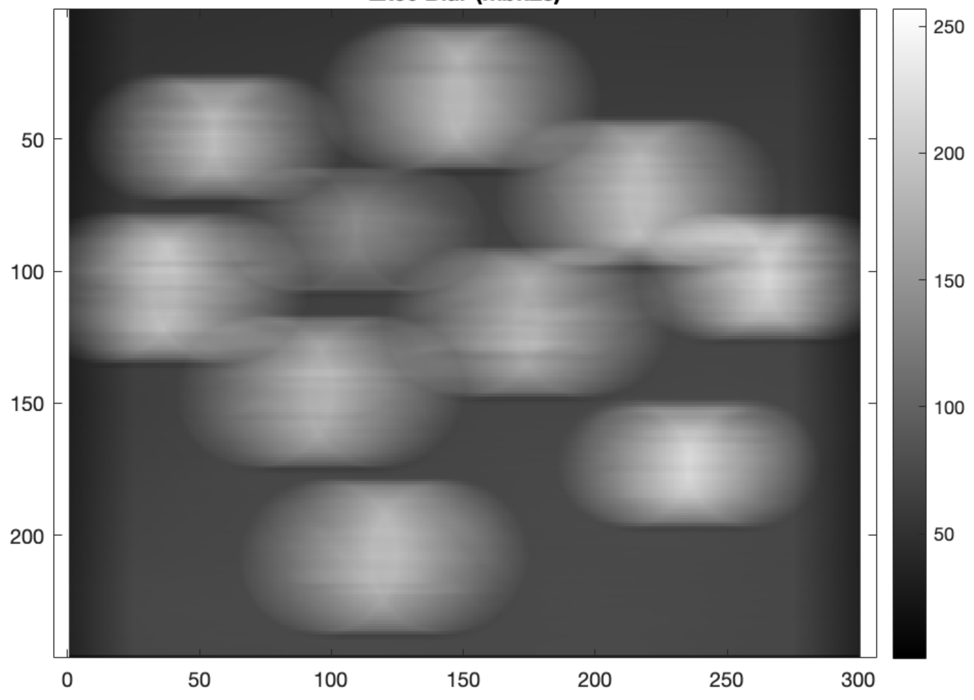


9.5 Exercise 5

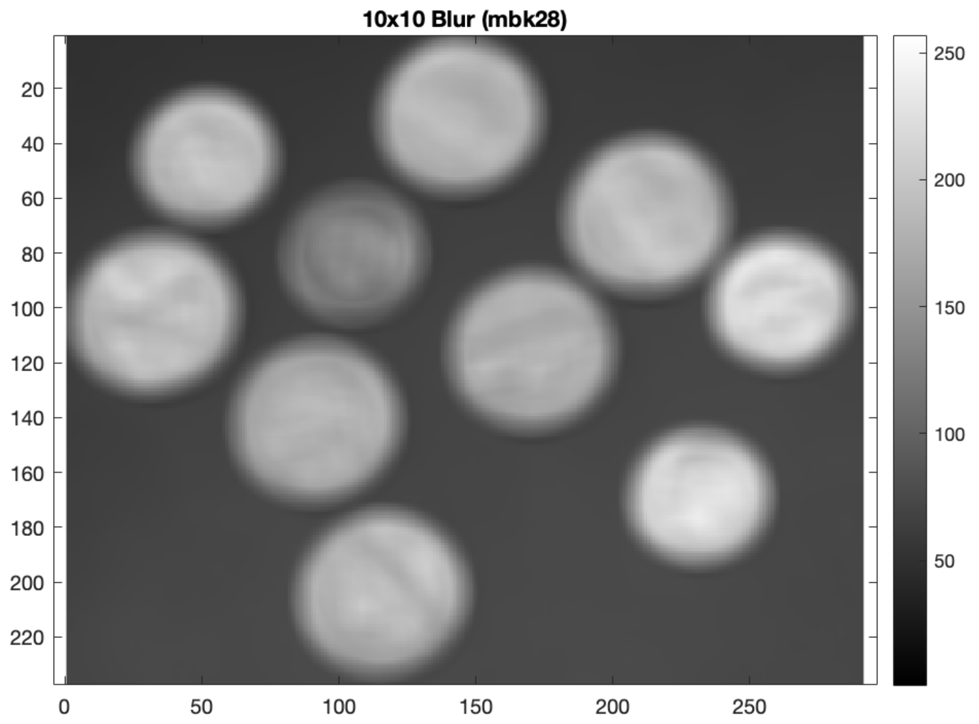


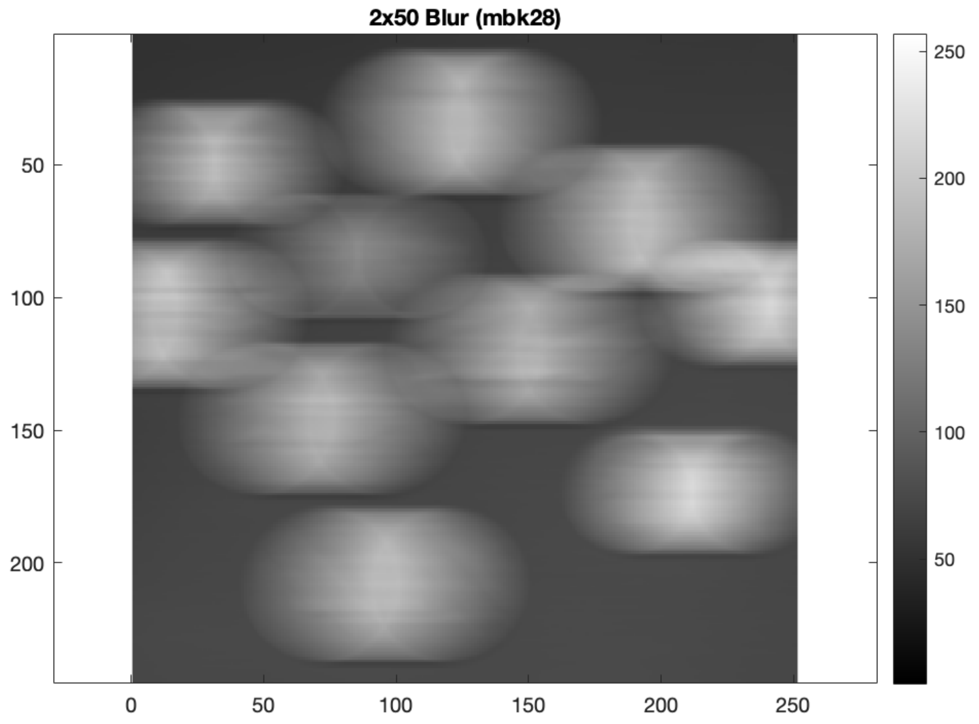


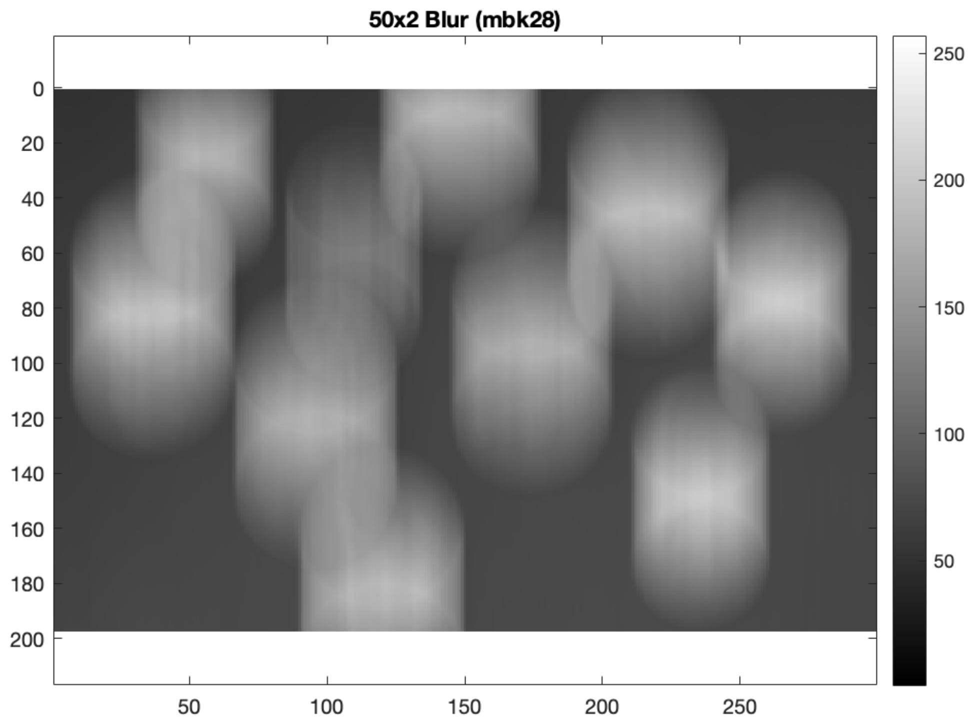
2x50 Blur (mbk28)



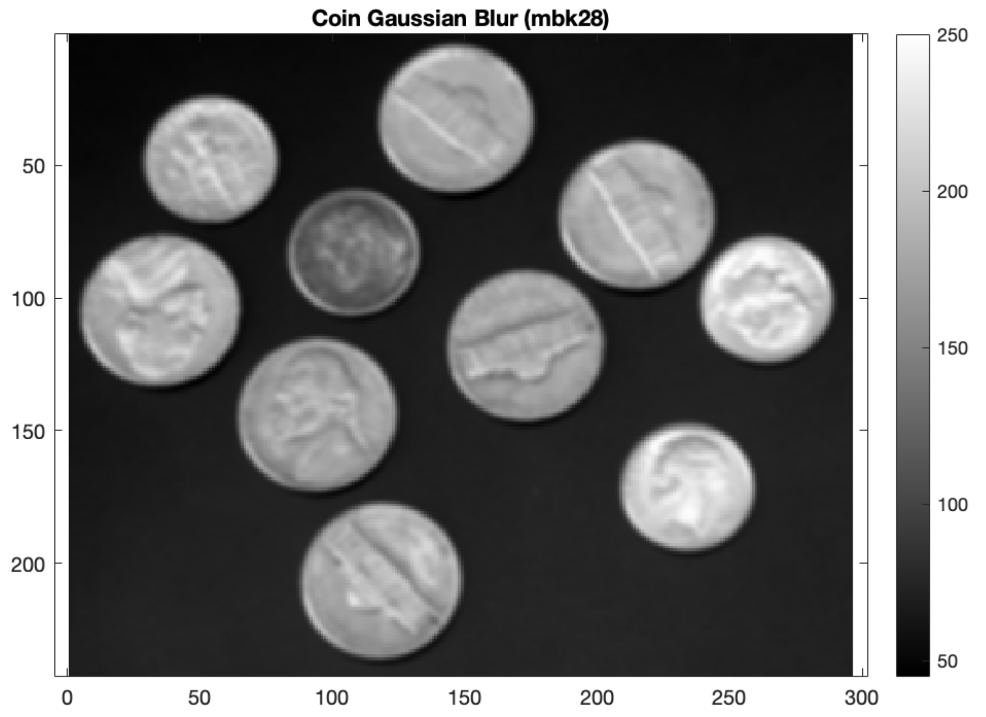
9.6 Exercise 6

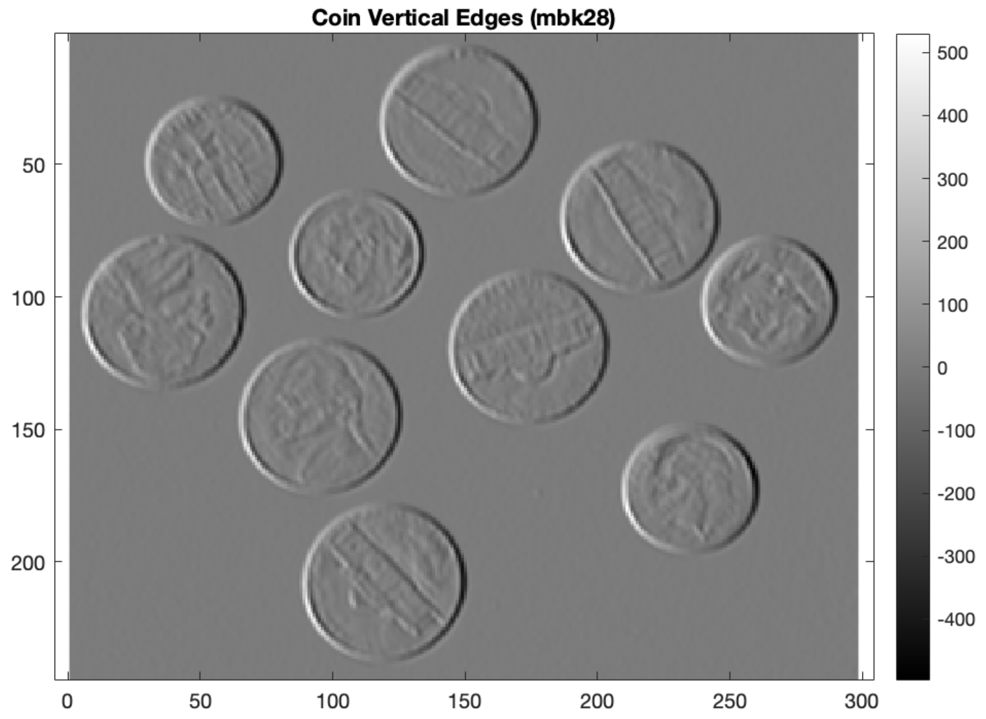


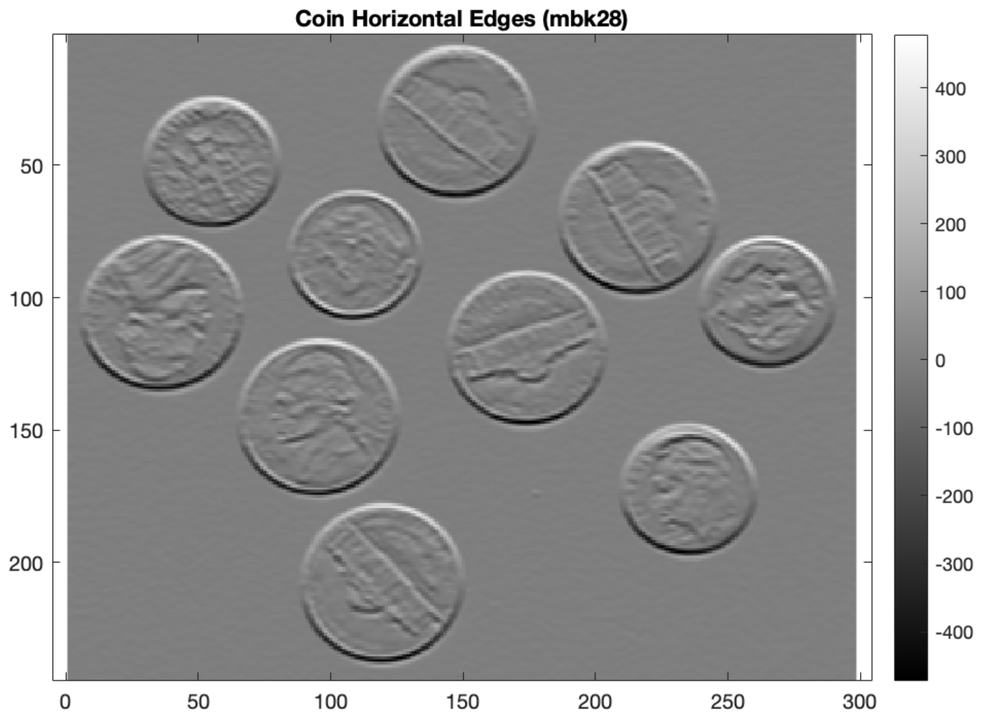


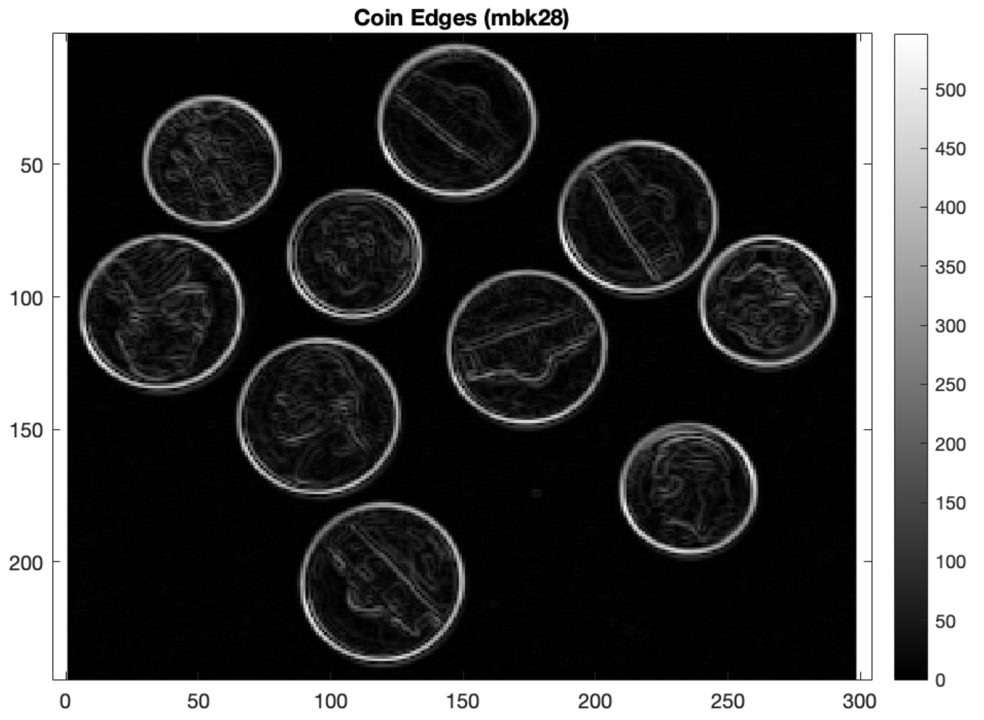

50x2 Blur (mbk28)
0
20
40
60
80
100
120
140
160
180
200
50
100
150
200
250
250
200
150
100
50

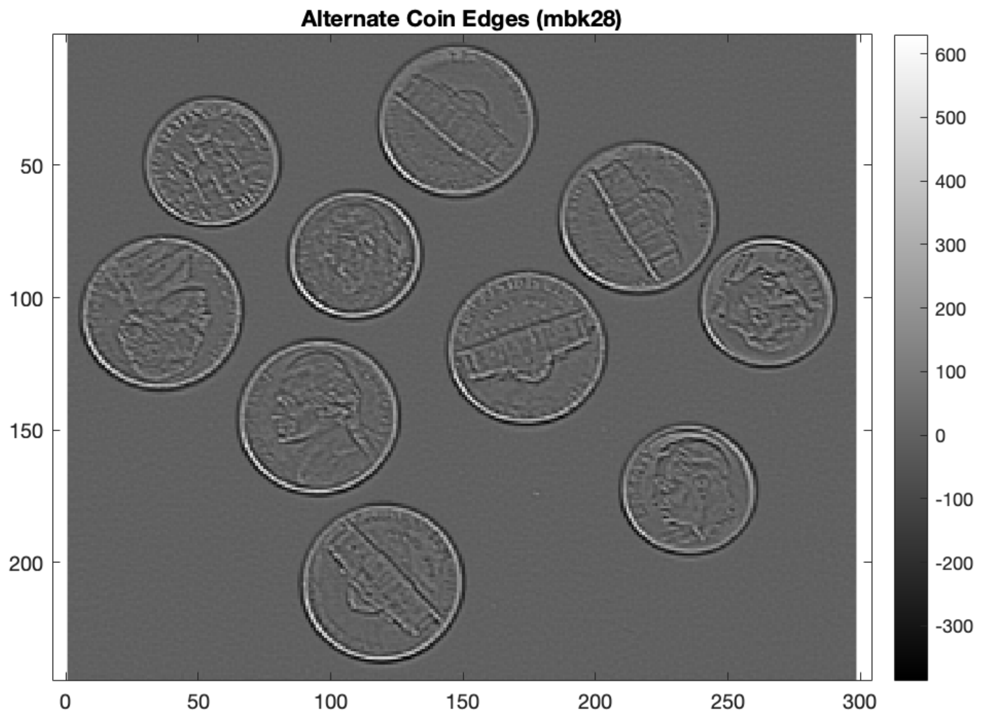
9.7 Exercise 7



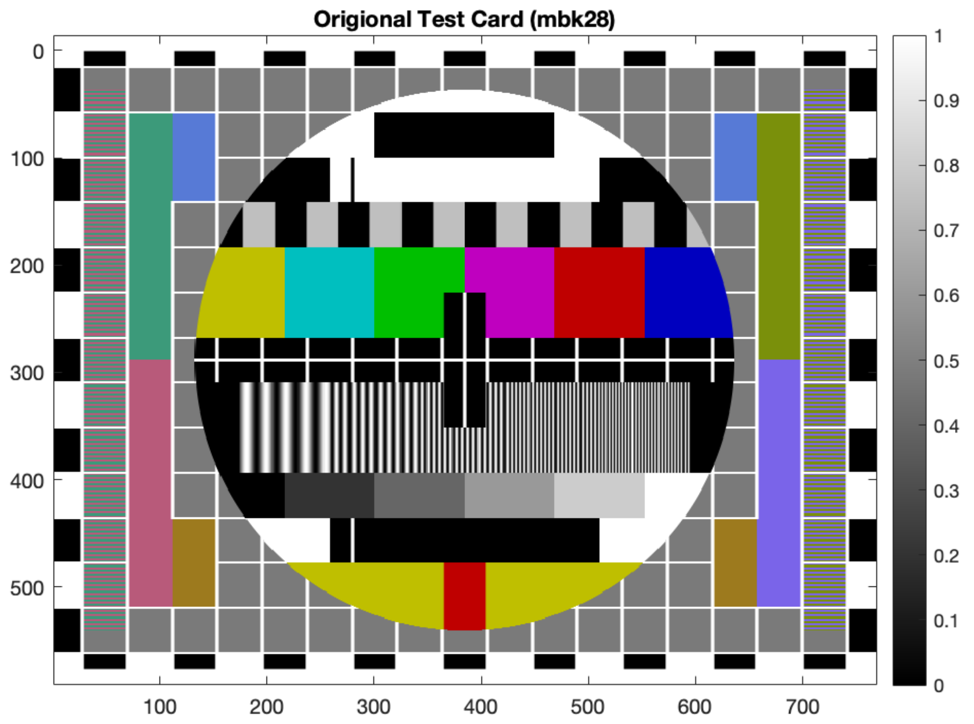


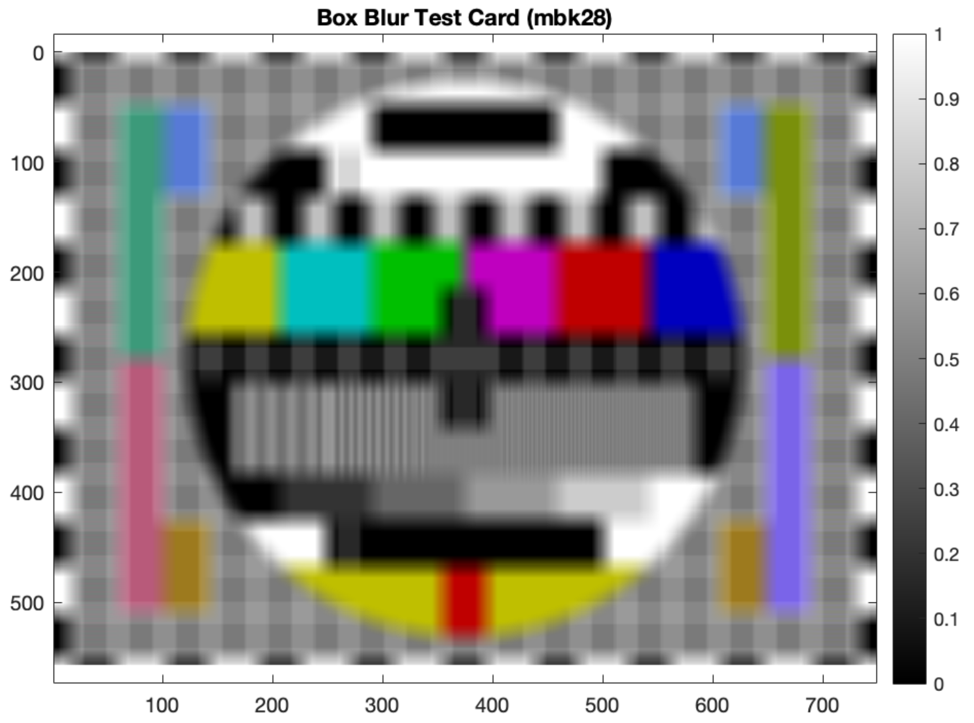

Coin Horizontal Edges (mbk28)
50
100
150
200
0
50
100
150
200
250
300
400
300
200
100
0
-100
-200
-300
-400

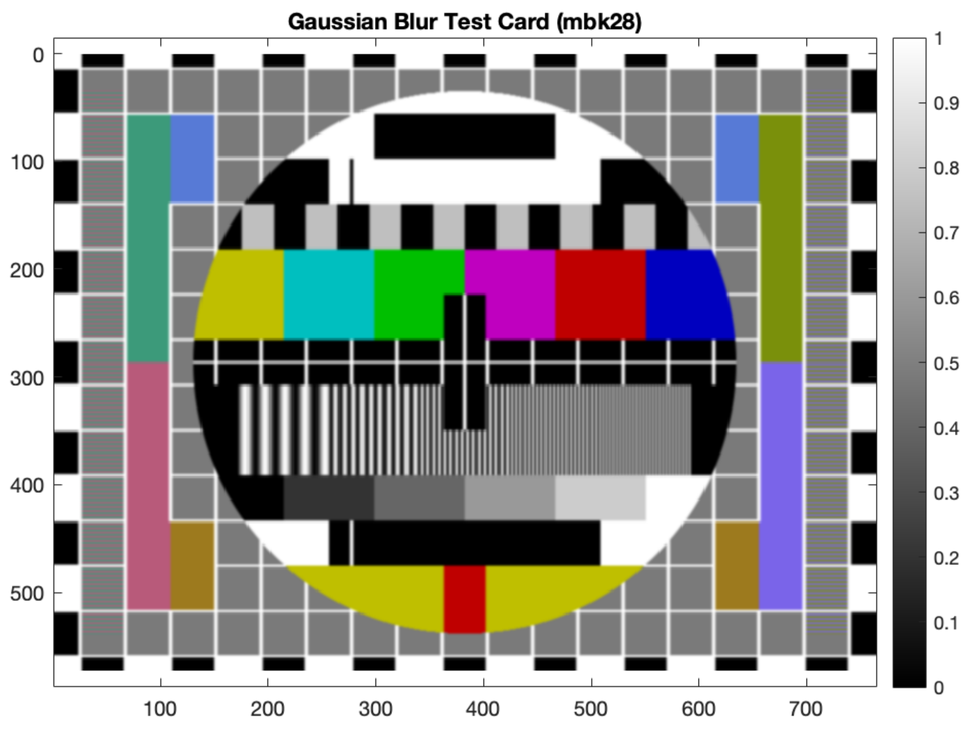

Coin Edges (mbk28)
50
100
150
200
0
50
100
150
200
250
300
500
450
400
350
300
250
200
150
100
50
0

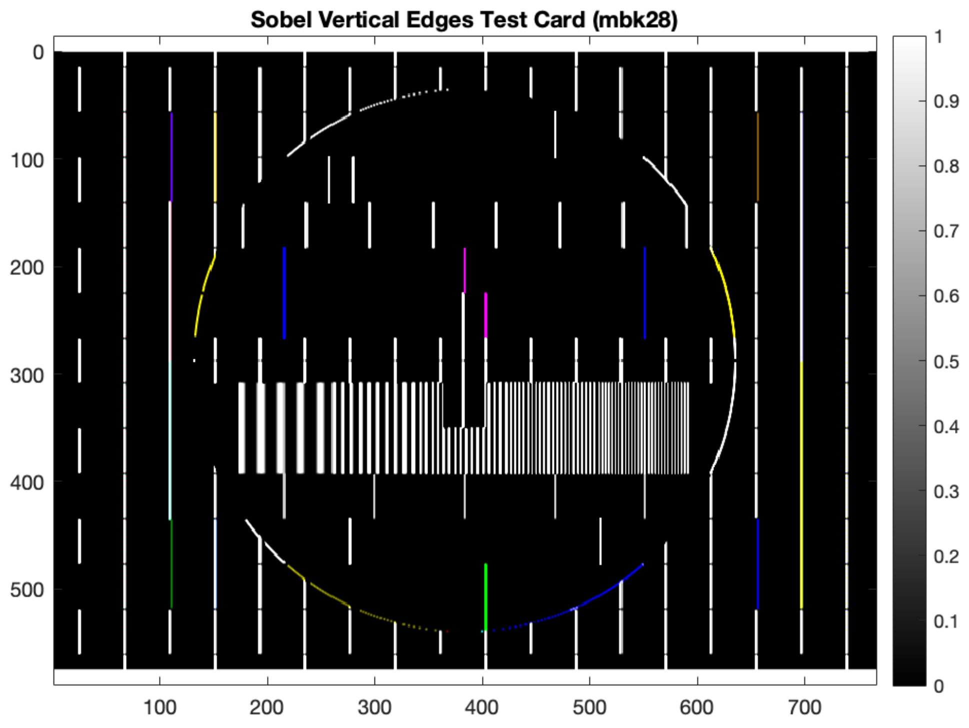

Alternate Coin Edges (mbk28)
50
100
150
200
0
50
100
150
200
250
300
600
500
400
300
200
100
0
-100
-200
-300

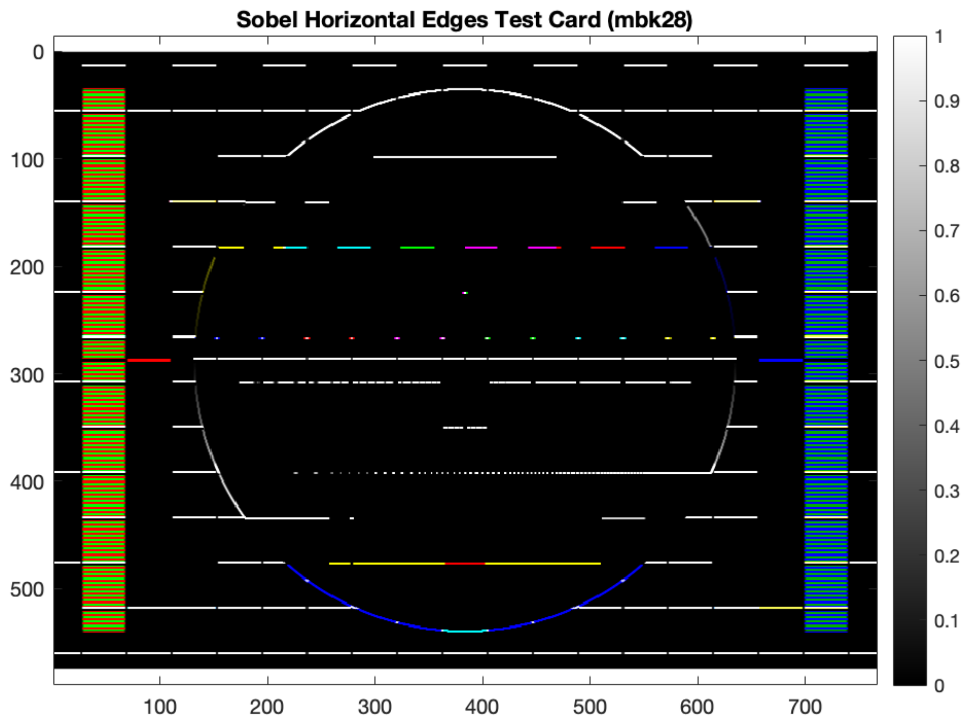
9.8 Exercise 8



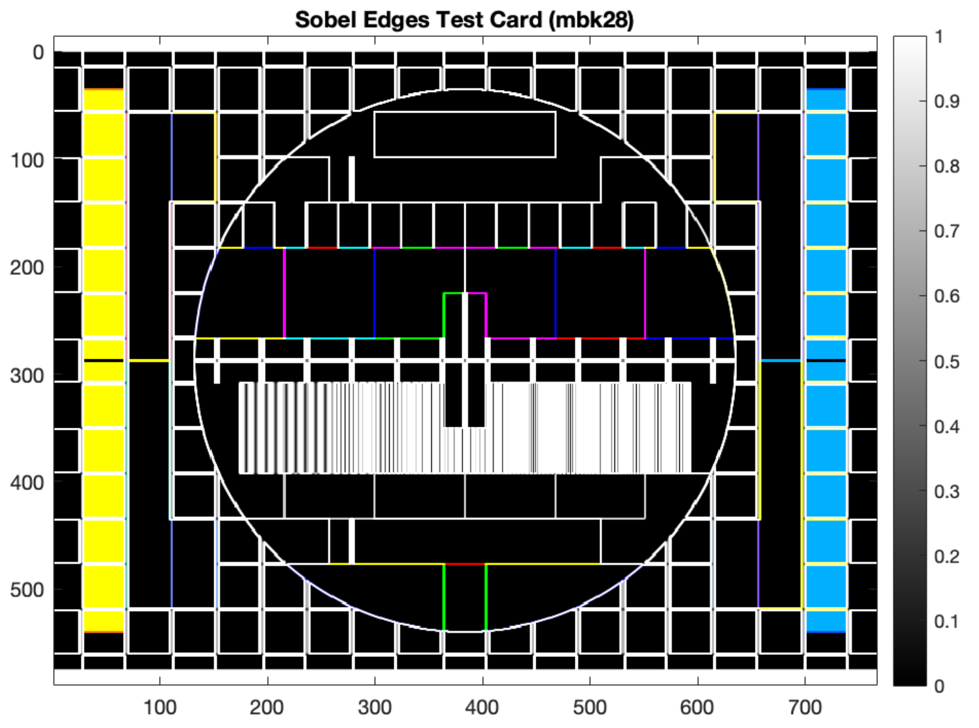



Gaussian Blur Test Card (mbk28)
0
100
200
300
400
500
100
200
300
400
500
600
700
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


Sobel Vertical Edges Test Card (mbk28)
0
100
200
300
400
500
100
200
300
400
500
600
700
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



Sobel Horizontal Edges Test Card (mbk28)
0
100
200
300
400
500
100
200
300
400
500
600
700
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0



Sobel Edges Test Card (mbk28)
0
100
200
300
400
500
100
200
300
400
500
600
700
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

10 Codes

10.1 Exercise 1

```
rad = 100;
del = 10;


[x, y] = meshgrid((-3*rad-del):(3*rad+del));
[rows, cols] = size(x);


venn_img = zeros(rows, cols, 3);
venn_img(:,:,1) = (dist(x, y, rad.*cos(0), rad.*sin(0)));
venn_img(:,:,2) = (dist(x, y, rad.*cos(2*pi/3), rad.*sin(2*pi/3)));
venn_img(:,:,3) = (dist(x, y, rad.*cos(4*pi/3), rad.*sin(4*pi/3)));



%fig
figure(1); clf
image(venn_img)
axis equal
%save('IP1_EX1_Plot2.png');

function ret = dist(x, y, xc, yc)
    p = 0.01;
    ret = 1 ./ (1 + p * sqrt((x-xc).^2 + (y-yc).^2));
end
```

10.2 Exercise 2

```

Ex2 = zeros(200, 200, 3);
Ex2(:,:,1) = rand(200, 200);
Ex2(:,:,2) = rand(200, 200);
Ex2(:,:,3) = rand(200, 200);

figure(1); clf
image(Ex2)
axis equal
```

10.3 Exercise 3

```

tc = linspace(0, 1, 101);
xc = humps(tc);
td = linspace(0, 1, 11);
xd = humps(td);
figure(1); clf
plot(tc, xc, 'b-')
hold on
plot(td, xd, 'bo')
hold off

h2 = [1/2, 1/2]
h5 = [1/5, 1/5, 1/5, 1/5, 1/5]

twopa = conv(xd, h2, 'same')
fivepa = conv(xd, h5, 'same')


hold on
plot(td, twopa, 'ro-')

plot(td, fivepa, 'go-')
hold off

title('Moving Averages (mbk28)')
```

10.4 Exercise 4

```
tc = linspace(0, 1, 101);
xc = humps(tc);
deltatc = tc(2) - tc(1);
td = linspace(0, 1, 11);
xd = humps(td);
deltatd = td(2) - td(1);

h = [1, 0, -1] ./ (2*deltatd)
con = conv(xd, h, 'same')

figure(1); clf
plot(tc, xc, 'b-')
hold on
plot(td, xd, 'bo')
hold off
title('Values')


figure(2); clf
twopointdiff = diff(xc)/deltatc;
twopointdiff(end+1) = twopointdiff(end);
plot(tc, twopointdiff, 'b-')
hold on
plot(tc(1:10:end), twopointdiff(1:10:end), 'bo');
hold on
plot(td, con, 'mo')
hold off
title('Derivative Approximation (11 points) (mbk28)')
```

10.5 Exercise 5

```
x = imread('coins.png');
h = ones(10 , 10) / 10^2;
h_25 = ones(2 , 50) / 10^2;
h_52 = ones(50 , 2) / 10^2;
y = conv2(x ,h , 'same');
y_25 = conv2(x ,h_25 , 'same');
y_52 = conv2(x ,h_52 , 'same');


figure(1); clf
image(y_25)
axis equal; colormap gray ; colorbar
title('2x50 Blur (mbk28)')
figure(2); clf
image(y_52)
axis equal; colormap gray ;colorbar
title ('50x2 Blur (mbk28)')
```

10.6 Exercise 6

```
x = imread('coins.png');
h = ones(10 , 10) / 10^2;
h_25 = ones(2 , 50) / 10^2;
h_52 = ones(50 , 2) / 10^2;
y = conv2(x ,h , 'valid');
y_25 = conv2(x ,h_25 , 'valid');
y_52 = conv2(x ,h_52 , 'valid');


figure(1); clf
image(y)
axis equal; colormap gray ; colorbar
title('10x10 Blur (mbk28)')
figure(2); clf
image(y_52)
axis equal; colormap gray ;colorbar
title ('50x2 Blur (mbk28)')
```

10.7 Exercise 7

```
x = imread('coins.png');

hGauss = [1, 4, 6, 4, 1; 4, 16, 24, 16, 4; 6, 24, 36, 24, 6;
                4, 16, 24, 16, 4; 1, 4, 6, 4, 1]./256;
yGauss = conv2(x ,hGauss , 'valid');

hx = [1, 0, -1; 1, 0, -1; 1, 0, -1]
CoinsEdgeX = conv2(x, hx, 'valid')

hy = [1, 1, 1; 0, 0, 0; -1, -1, -1]
CoinsEdgeY = conv2(x, hy, 'valid')

CoinsEdge = sqrt((CoinsEdgeX.^2)+(CoinsEdgeY.^2));

edgeDetect = [-1, -1, -1; -1, 8, -1; -1, -1, -1];
yED = conv2(x ,edgeDetect , 'valid');


figure(1); clf
imagesc(yED)
axis equal; colormap gray ; colorbar
title('Alternate Coin Edges (mbk28)')
```

10.8 Exercise 8

```
x = imread('TestCard.png');

hbb = ones(21,21)/441;
ybb = layerConv(x, hbb)

hGauss = [1, 4, 6, 4, 1; 4, 16, 24, 16, 4; 6, 24, 36, 24, 6;
                4, 16, 24, 16, 4; 1, 4, 6, 4, 1]./256;
yGauss = layerConv(x, hGauss)

hsoblv = [1, 0, -1; 2, 0, -2; 1, 0, -1];
ysoblv = layerConv(x, hsoblv)

hsoblh = [1, 2, 1; 0, 0, 0; -1, -2, -1];
ysoblh = layerConv(x, hsoblh)


for k = 1:3
    ysobl(:,:,k) = sqrt(conv2(x(:,:,k), hsoblv, 'valid').^2 + conv2(x(:,:,k), hsoblh, 'va
end
ysobl = uint8(ysobl)

image(ysobl)
axis equal; colormap gray ; colorbar
title('Sobel Edges Test Card (mbk28)')







function y = layerConv(x, h)
for k = 1:3
    y(:,:,k) = conv2(x(:,:,k), h, 'valid');
end
y = uint8(y);
end
```