Task 6: Image Processing

Name: Jenice Zong **Degree:** DE-DE 22261024

Problem 1

a) Compute G using scan2d:

```
clear * filter2d_gx=
% Filter kernel
F = [-1 0 1; -2 0 2; -1 0 1];

% Apply filter using scan2d
G = apply2d(scan2d(F));

b) Compute G using for loops:

% Initialize output image
[M, N] = size(I);
G = zeros(M, N);

% Apply Sobel filter
for i = 2:M-1
    for j = 2:N-1
        sum = 0;
        for m = -1:1
            for n = -1:1
                sum = sum + double(I(i+m, j+n)) * F(m+2, n+2);
        end
    end
    G(i, j) = sum;
end
end

% Display results
figure;
imshow(G, [min(G(:)), max(G(:))]);
title('Sobel filter');
imwrite(G, 'sobel_result.jpg');
```

Resulted image G:

c) Compute G using for loop:

```
% Initialize output image
[M, N] = size(I);
G = zeros(M, N);

% Apply Sobel filter
for i = 2:M-1
    for j = 2:N-1
        sum = 0;
        for m = -1:1
            for n = -1:1
                sum = sum + double(I(i+m, j+n)) * F(m+2, n+2);
        end
    end
    G(i, j) = sum;
end
end

% Display results
figure;
imshow(G, [min(G(:)), max(G(:))]);
title('Sobel filter');
imwrite(G, 'sobel_result.jpg');
```

Resulted image G:

The two images look the same.

Problem 2

Original image:

a) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

b) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

c) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

The four images look the same.

Problem 3

a) Create Gaussian kernel G:

```
G = fspecial('gaussian',[7,7]);
sigma = 1.4;
G = G./max(G(:));

% Gaussian filter
G_size = size(G,1);
G = G./sum(G(:));
G = zeros(G_size);

for m = -(G_size-1)/2:(G_size-1)/2
    for n = -(G_size-1)/2:(G_size-1)/2
        x = -(G_size-1)/2:(G_size-1)/2;
        y = -(G_size-1)/2:(G_size-1)/2;
        G(m+(G_size+1)/2,n+(G_size+1)/2) = sum(sum(exp(-(x.^2+y'.^2)./(2*sigma^2)).
*G.*I(m+x,n+y)));
end
end

% Display results
figure;
imshow(G, [min(G(:)), max(G(:))]);
title('Gaussian filter');
imwrite(G, 'gaussian_result.jpg');
```

b) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

c) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

The four images look the same.

Problem 4

a) Original image:

b) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

c) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

d) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

The four images look the same.

Problem 5

a) Gaussian filter:

```
% Gaussian filter
G = fspecial('gaussian',[7,7]);
sigma = 1.4;
G = G./max(G(:));
G_size = size(G,1);
G = G./sum(G(:));
G = zeros(G_size);

for m = -(G_size-1)/2:(G_size-1)/2
    for n = -(G_size-1)/2:(G_size-1)/2
        x = -(G_size-1)/2:(G_size-1)/2;
        y = -(G_size-1)/2:(G_size-1)/2;
        G(m+(G_size+1)/2,n+(G_size+1)/2) = sum(sum(exp(-(x.^2+y'.^2)./(2*sigma^2)).
*G.*I(m+x,n+y)));
end
end

% Display results
figure;
imshow(G, [min(G(:)), max(G(:))]);
title('Gaussian filter');
imwrite(G, 'gaussian_result.jpg');
```

b) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

c) Convolution using sift:

```
Matlab's built-in filter (sift) for comparison
G = sift(I, F);
```

The four images look the same.

Problem 6

a) Binarize image:

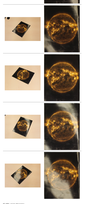
```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```

b) Binarize image:

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```



c) Binarize image:

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```

d) Binarize image:

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```

Problem 7

Original and cropped images:

Original Image	Cropped Image

```
% Binarize if input image
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 8

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 9

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 10

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 11

Original Image	Result	Processed Image

Problem 12

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 13

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 14

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 15

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 16

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 17

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 18

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 19

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 20

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 21

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 22

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 23

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 24

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 25

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 26

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 27

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 28

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 29

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 30

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 31

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 32

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 33

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 34

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 35

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 36

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 37

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 38

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 39

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 40

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```


Problem 41

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos(floor(i*cos(k*pi/180)+j*sin(k*pi/180)),k) = 1;
            end
        end
    end
end
cos = cos(1:M,1:N);
imshow(cos,[min(cos(:)),max(cos(:))]);
imwrite(cos, 'cos_result.jpg');
```

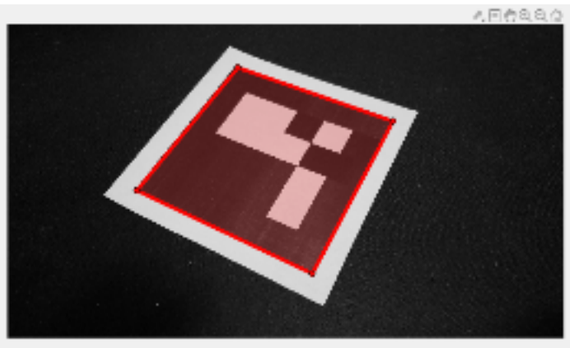

Problem 42

```
% Parameters
BwThreshold = 0.1 * 255;% % of
% Convert to grayscale if input is already image size
if ndims(I) == 3
    I = rgb2gray(I);% Convert to grayscale
end
BW = I > BwThreshold;

% Processing
[M, N] = size(BW);
cos = zeros(floor(sqrt(M^2+N^2)),180);
for i = 1:M
    for j = 1:N
        if BW(i,j) == 1
            for k = 1:180
                cos
```

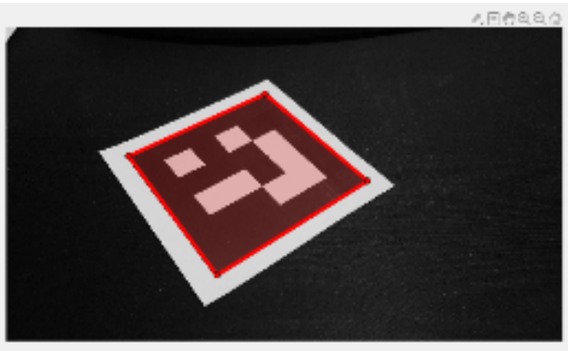
```
% Identify marker number
i=1;
while i <= size(aruco_marker,3)
     if markerImg == aruco_marker(:,:,i)
         marker_num = i;
         break
     elseif rot90(markerImg,1)==aruco_marker(:,:,i)
         marker_num = i;
         break
     elseif rot90(markerImg,2)==aruco_marker(:,:,i)
         marker_num = i;
         break
     elseif rot90(markerImg,3)==aruco_marker(:,:,i)
         marker_num = i;
         break
     else
         i = i+1;
     end
end
end
```

img(1):



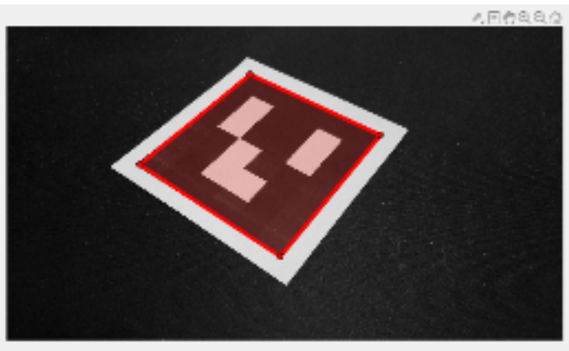
```
Aruco Marker Number: 2
```

img(2):



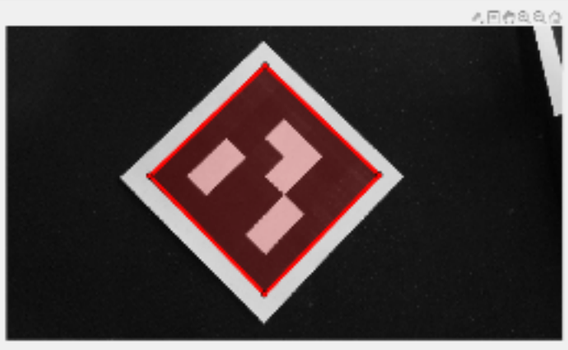
```
Aruco Marker Number: 4
```

img(3):



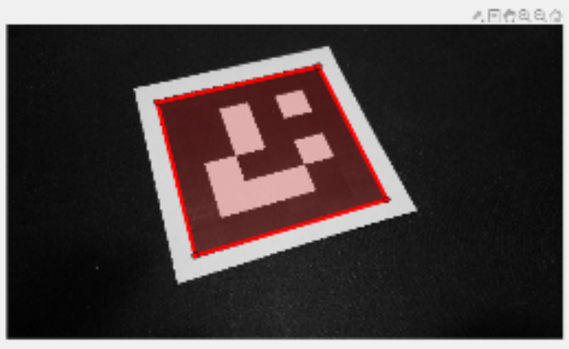
```
Aruco Marker Number: 3
```

img(4):



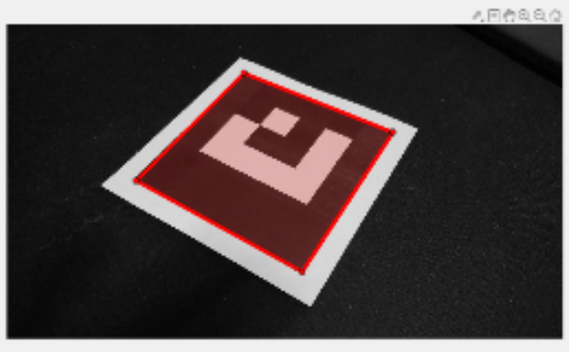
```
Aruco Marker Number: 3
```

img(5):



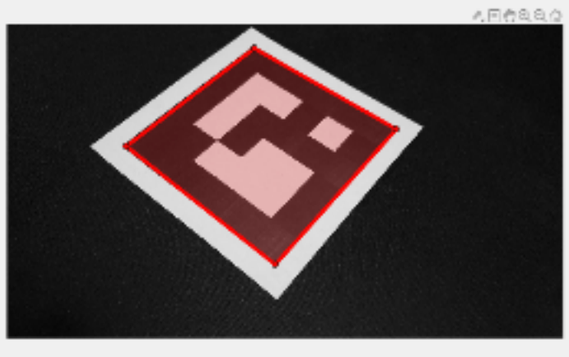
```
Aruco Marker Number: 4
```

img(6):



```
Aruco Marker Number: 1
```

img(7):



```
Aruco Marker Number: 5
```