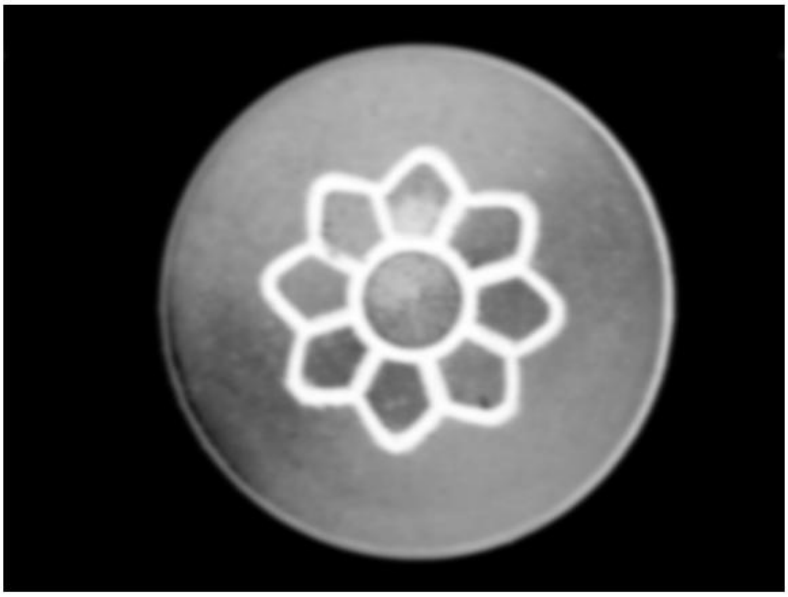
นายพีรณัฐ ธารทะเลทอง 550610530

Digital Image Processing (261453)

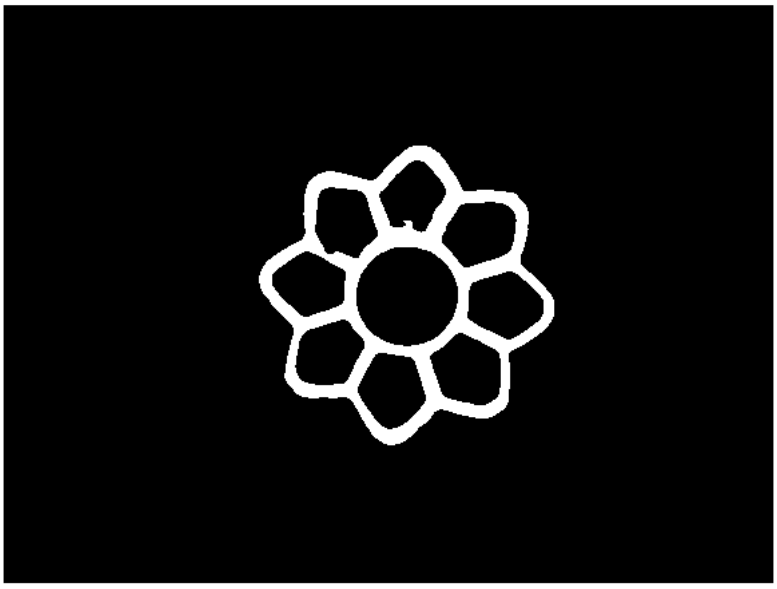
Computer Assignment 3

1. คำนวณหาพื้นที่ส่วนรูปดอกไม้สีขาวของรูป Flower_Snack.pgm

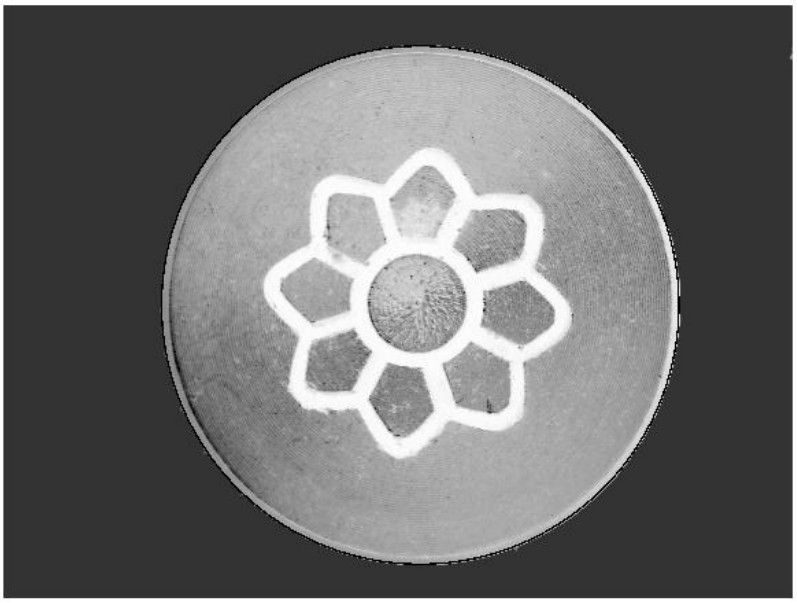
หาภาพ object ด้วยการเบลอภาพ ด้วย Gaussian low-pass filter ที่ cutoff = 36 เพื่อให้แยกวัตถุได้ดีขึ้น ได้ดังรูป



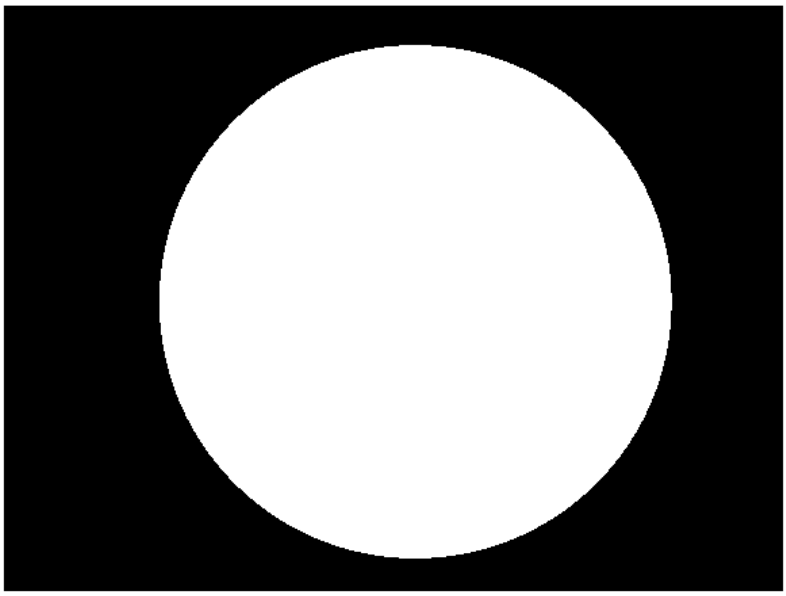
และตัด Threshold ที่ 225 เพื่อแยกวัตถุออกมา จะได้วัตถุดังรูป



และหาวงกลมด้วยการลูปหาตำแหน่งที่ทำให้รัศมีมีค่ามากสุดและครอบพื้นที่วงกลมมากที่สุด
จะได้ ตำแหน่งที่ (1218, 875) และรัศมีขนาด 757 pixel และมาวาดเป็นขอบวงกลม (เส้นสีดำ) ดังรูป



และวาดวงกลมด้วยตำแหน่งและรัศมีนั้น ได้ดังรูป

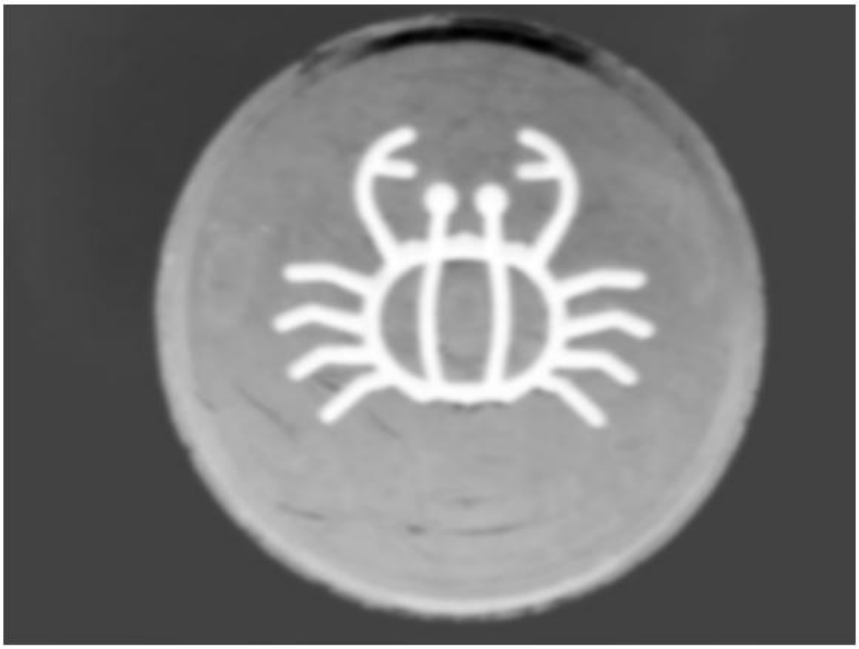


และหาจำนวน pixel ของทั้งสอง object และนำมาคำนวณหาพื้นได้ จะได้

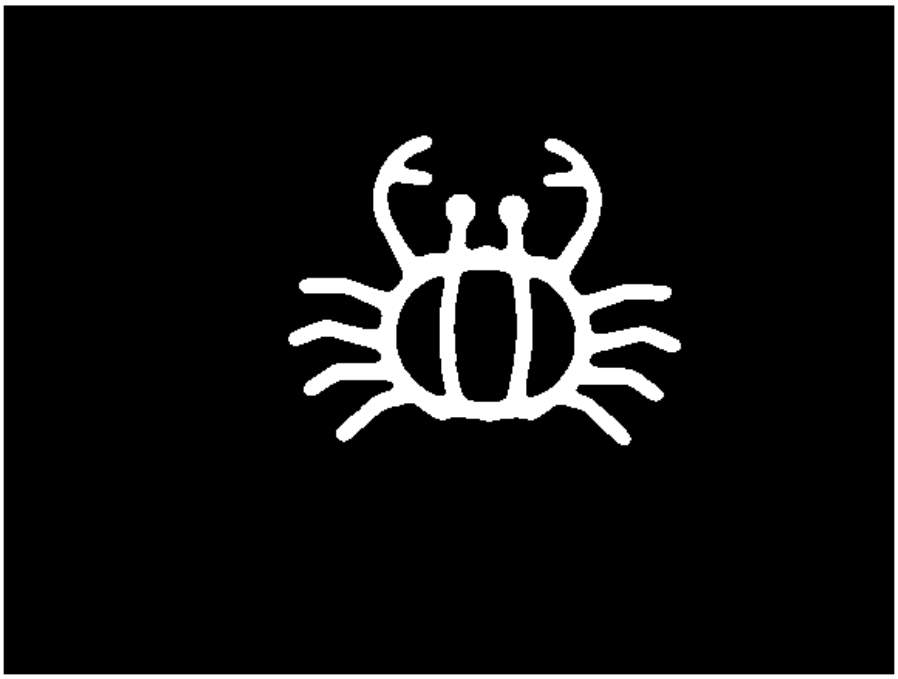
Area = 2.6 * (195244 / 1785473) = 0.2843 cm^2

2. คำนวณหาพื้นที่ส่วนรูปปูของรูป Crab.pgm

ใช้หลักการเดียวกันกับข้อแรก โดยการเบลอภาพ ด้วย Gaussian low-pass filter ที่ cutoff = 36 จะได้ดังรูป

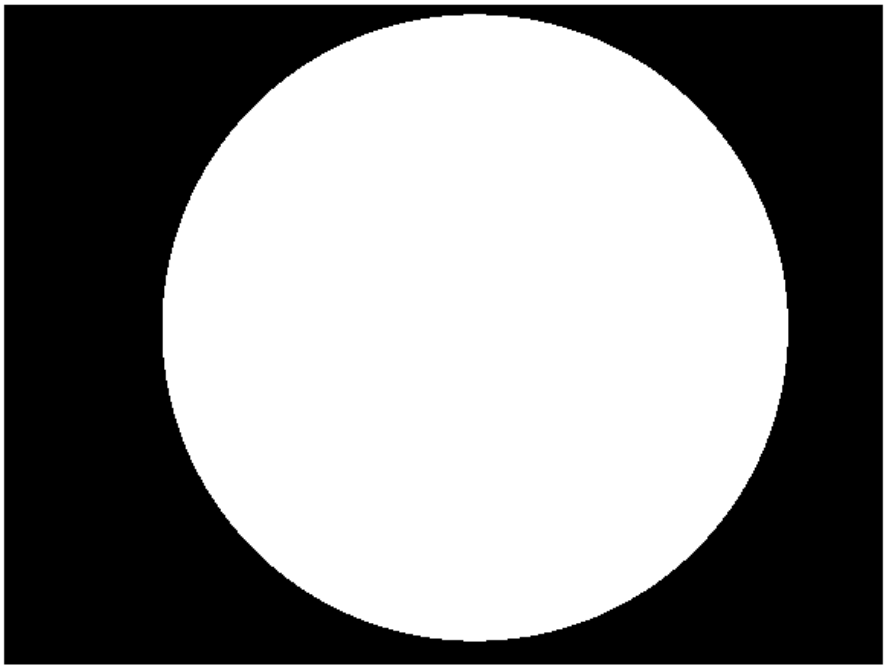


และแยกวัตถุออกมา โดยตัด Threshold ที่ 215 จะได้วัตถุดังรูป



จากนั้นหาขอบของวงกลม จะได้ตำแหน่งที่ (1236, 847) รัศมีขนาด 820 pixel ซึ่งจะได้ขอบเส้นสีดำดังรูป จะได้

ได้วงกลมที่ตำแหน่งและรัศมีนั้น ดังรูป



และหาจำนวน pixel ของทั้งสอง object และนำมาคำนวณหาพื้นได้ จะได้

Area = 2.766 * (228254/ 2092035) = 0.3018 cm^2

Code (โค๊ดทั้งหมดอยู่ที่ https://github.com/porpeeranut/Digital_Image_Processing_Assignment3)

Flower_Snack.m

```
flowerPath = 'Flower_Snack.pgm';
im = imread(flowerPath);
bgTH = 65;
flowTH = 210;
[h, w] = size(im)
bg = im;

% Gaussian Lowpass Filter
FFT = fftshift(fft2(im)); % shift to center
y = (w-1)/2;
x = (h-1)/2;
cutoff = 36;
[u,v] = meshgrid(-y:y,-x:x);
D = sqrt(u.^2+v.^2);
HG = double(exp((-(D).^2)./(2.*((cutoff).^2))));
% multiple wit fourier transform
GHG = HG.*FFT;
% inverse fourier transform
imObj = round(real(ifft2(ifftshift(GHG))));
figure('name','Gaussian Lowpass Filter');
imshow(real(imObj),[]);

% binary image to find obj
objPixel = 0;
TH = 225;
for i = 1:h
    for j = 1:w
        if (imObj(i,j) >= TH )
        imObj(i,j) = 255;
        objPixel = objPixel + 1;
        else
        imObj(i,j) = 0;
        end
    end
end
figure('name','obj');
imshow(imObj);

% black white bg
for r = 1:h
    for c = 1:w
        if (bg(r,c) > bgTH) && c < w-100
            bg(r,c) = 255;
        else
            bg(r,c) = 140;
        end
    end
end

% find cycle
max = 0;
ansX = 0;
ansY = 0;
ansR = 0;
cycleIm = bg;
for x = 1190:1240
    for y = 855:905
        for r = 740:770
            sum = 0;
            for degree = 1:360
                if (bg(y+round(r*sind(degree)), x+round(r*cosd(degree))) > 200)
```

```
                    sum = sum+1;
                end
            end
            if (max < sum) && (ansR < r)
                max = sum;
                ansX = x;
                ansY = y;
                ansR = r;
            end
        end
    end
end
max
ansX
ansY
ansR
max = 273;
ansX = 1218;
ansY = 875;
ansR = 757;

% set to black
for r = 1:h
    for c = 1:w
        cycleIm(r,c) = 0;
    end
end

%   show cycle border
for degree = 0:0.01:360
    im(ansY+round(ansR*sind(degree)), ansX+round(ansR*cosd(degree))) = 0;
    im(ansY+round((ansR-1)*sind(degree)), ansX+round((ansR-1)*cosd(degree))) = 0;
    im(ansY+round((ansR-2)*sind(degree)), ansX+round((ansR-2)*cosd(degree))) = 0;
    im(ansY+round((ansR-3)*sind(degree)), ansX+round((ansR-3)*cosd(degree))) = 0;
    im(ansY+round((ansR-4)*sind(degree)), ansX+round((ansR-4)*cosd(degree))) = 0;

    cycleIm(ansY+round(ansR*sind(degree)), ansX+round(ansR*cosd(degree))) = 255;
    cycleIm(ansY+round((ansR-1)*sind(degree)), ansX+round((ansR-1)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-2)*sind(degree)), ansX+round((ansR-2)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-3)*sind(degree)), ansX+round((ansR-3)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-4)*sind(degree)), ansX+round((ansR-4)*cosd(degree))) =
255;
end
figure('name','ori with border');
imshow(im);

% fill cycle obj
cyclePixel = 0;
for r = 120:h-100
    start = 0;
    filled = 0;
    for c = 1:w
        %   found 1st border
        if (start == 0) && (filled == 0) && (cycleIm(r,c) == 255)
            start = 1
            cyclePixel = cyclePixel + 1;
            continue
        end

        %   fill
        if (start == 1) && (filled == 0) && (cycleIm(r,c) == 0)
            filled = 1
            cycleIm(r,c) = 255;
```

```
            cyclePixel = cyclePixel + 1;
            continue
        end
        if (filled == 1) && (cycleIm(r,c) == 0)
            cycleIm(r,c) = 255;
            cyclePixel = cyclePixel + 1;
            continue
        end

        %   found 2nd border
        if (filled == 1) && (cycleIm(r,c) == 255)
            cyclePixel = cyclePixel + 1;
            break
        end
    end
end
figure('name','cycle');
imshow(cycleIm);

objPixel
cyclePixel
area = 2.60;
area = (area*objPixel)/cyclePixel
```

Crab.m

```
crabPath = 'Crab.pgm';
im = imread(crabPath);
bgTH = 65;
flowTH = 210;
[h, w] = size(im)
bg = im;

% Gaussian Lowpass Filter
FFT = fftshift(fft2(im)); % shift to center
y = (w-1)/2;
x = (h-1)/2;
cutoff = 36;
[u,v] = meshgrid(-y:y,-x:x);
D = sqrt(u.^2+v.^2);
HG = double(exp((-(D).^2)./(2.*((cutoff).^2))));
% multiple wit fourier transform
GHG = HG.*FFT;
% inverse fourier transform
imObj = round(real(ifft2(ifftshift(GHG))));
figure('name','Gaussian Lowpass Filter');
imshow(real(imObj),[]);

% binary image to find obj
objPixel = 0;
TH = 215;
for i = 1:h
    for j = 1:w
        if (imObj(i,j) >= TH )
        imObj(i,j) = 255;
        objPixel = objPixel + 1;
        else
        imObj(i,j) = 0;
        end
    end
end
figure('name','obj');
imshow(imObj);

% black white bg
for r = 1:h
    for c = 1:w
        if (bg(r,c) > bgTH) && c < w-100
            bg(r,c) = 255;
        else
            bg(r,c) = 140;
        end
    end
end

% find cycle
max = 0;
ansX = 0;
ansY = 0;
ansR = 0;
cycleIm = bg;
for x = 1236:1252
    for y = 847:855
        for r = 820:828
            sum = 0;
            for degree = 1:360
                if (cycleIm(y+round(r*sind(degree)), x+round(r*cosd(degree))) > 10)
                    sum = sum+1;
                end
            end
```

```
            if (max < sum) && (ansR < r)
                max = sum;
                ansX = x;
                ansY = y;
                ansR = r;
            end
        end
    end
end
max
ansX
ansY
ansR
max = 360;
ansX = 1236;
ansY = 847;
ansR = 820;

% set to black
for r = 1:h
    for c = 1:w
        cycleIm(r,c) = 0;
    end
end

%   show cycle border
for degree = 0:0.01:360
    im(ansY+round(ansR*sind(degree)), ansX+round(ansR*cosd(degree))) = 0;
    im(ansY+round((ansR-1)*sind(degree)), ansX+round((ansR-1)*cosd(degree))) = 0;
    im(ansY+round((ansR-2)*sind(degree)), ansX+round((ansR-2)*cosd(degree))) = 0;
    im(ansY+round((ansR-3)*sind(degree)), ansX+round((ansR-3)*cosd(degree))) = 0;
    im(ansY+round((ansR-4)*sind(degree)), ansX+round((ansR-4)*cosd(degree))) = 0;

    cycleIm(ansY+round(ansR*sind(degree)), ansX+round(ansR*cosd(degree))) = 255;
    cycleIm(ansY+round((ansR-1)*sind(degree)), ansX+round((ansR-1)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-2)*sind(degree)), ansX+round((ansR-2)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-3)*sind(degree)), ansX+round((ansR-3)*cosd(degree))) =
255;
    cycleIm(ansY+round((ansR-4)*sind(degree)), ansX+round((ansR-4)*cosd(degree))) =
255;
end
figure('name','ori with border');
imshow(im);

% fill cycle obj
cyclePixel = 0;
for r = 32:h-66
    start = 0;
    filled = 0;
    for c = 1:w
        %   found 1st border
        if (start == 0) && (filled == 0) && (cycleIm(r,c) == 255)
            start = 1;
            cyclePixel = cyclePixel + 1;
            continue
        end

        %   fill
        if (start == 1) && (filled == 0) && (cycleIm(r,c) == 0)
            filled = 1;
            cycleIm(r,c) = 255;
            cyclePixel = cyclePixel + 1;
            continue
        end
```

```
        if (filled == 1) && (cycleIm(r,c) == 0)
            cycleIm(r,c) = 255;
            cyclePixel = cyclePixel + 1;
            continue
        end

        %   found 2nd border
        if (filled == 1) && (cycleIm(r,c) == 255)
            cyclePixel = cyclePixel + 1;
            break
        end
    end
end
figure('name','cycle');
imshow(cycleIm);

objPixel
cyclePixel
area = 2.766;
area = (area*objPixel)/cyclePixel
```