Problem 1.

```
image = [0 0 2 4;4 3 3 2;4 3 3 3];
[row col] = size(image);
horizontal = [-1 -2 -1;0 0 0;1 2 1];
vertical = [-1 0 1;-2 0 2;-1 0 1];
Ix = conv2(image,horizontal,'same');
Iy = conv2(image,vertical,'same');
gradient_mag = zeros(row,col);
for r = 1:row
    for c = 1:col
        gradient_mag(r,c) = sqrt(Ix(r,c)^2 + Iy(r,c)^2);
    end
end
```

The result, gradient_mag (gradient magnitude), is [11.4 13.3 13.0 9.9;14.2 11.0 4.5 11.0;14.2 13.3 11 11.4]

1. (10 points) Apply 3x3 Horizontal $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ and Vertical $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ Operator to the image below. Then compute the gradient magnitude at each pixel. Show all your work and assumption.

1 → 4

1 ↓ 3

0	0	2	4
4	3	3	2
4	3	3	3

$$H_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow H_x(-x,-y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow H_y(-x,-y) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$F_x = I(x,y) * H_x(x,y) = \begin{bmatrix} 0 & 0 & 2 & 4 \\ 4 & 3 & 3 & 2 \\ 4 & 3 & 3 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$F_y = I(x,y) * H_y(x,y) = \begin{bmatrix} 0 & 0 & 2 & 4 \\ 4 & 3 & 3 & 2 \\ 4 & 3 & 3 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$F_x = \begin{bmatrix} -11 & -13 & -11 & -7 \\ -11 & -11 & -4 & 1 \\ 11 & 13 & 11 & 7 \end{bmatrix} \#$$

$$F_y = \begin{bmatrix} -3 & -3 & -7 & 7 \\ -9 & 1 & -2 & 11 \\ -9 & 3 & 1 & 9 \end{bmatrix} \#$$

$$\therefore |\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}$$

$|\nabla f(1,1)| = \sqrt{11^2+3^2} = 11.4018$

$|\nabla f(1,2)| = \sqrt{11^2+9^2} = 14.2127$

$|\nabla f(1,3)| = \sqrt{11^2+9^2} = 14.2127$

$|\nabla f(2,1)| = \sqrt{13^2+3^2} = 13.3417$

$|\nabla f(2,2)| = \sqrt{11^2+1^2} = 11.0454$

$|\nabla f(2,3)| = \sqrt{13^2+3^2} = 13.3417$

$|\nabla f(3,1)| = \sqrt{11^2+7^2} = 13.0384$

$|\nabla f(3,2)| = \sqrt{4^2+2^2} = 4.4721$

$|\nabla f(3,3)| = \sqrt{11^2+1^2} = 11.0454$

$|\nabla f(4,1)| = \sqrt{7^2+7^2} = 9.8995$

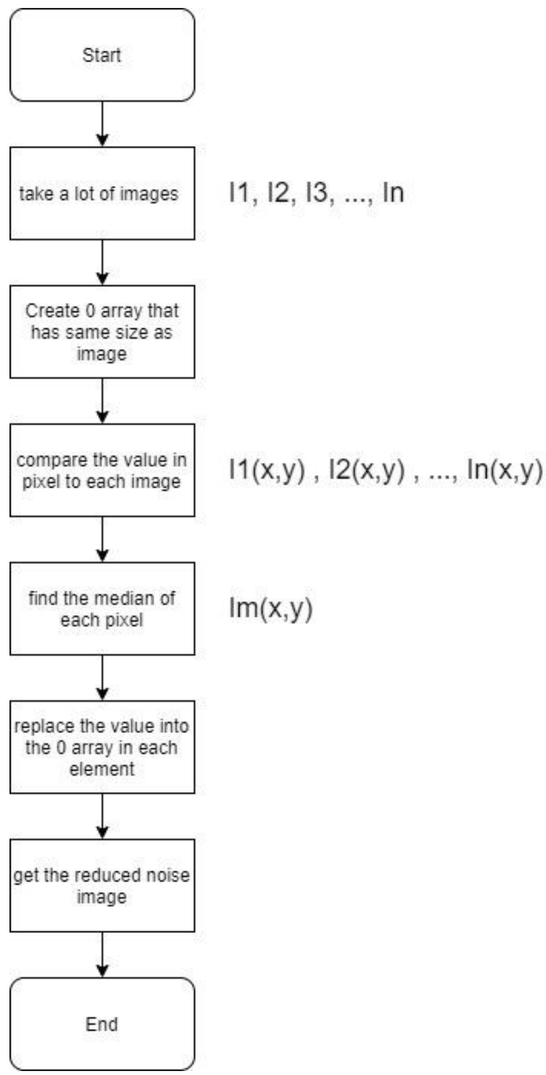
$|\nabla f(4,2)| = \sqrt{1^2+11^2} = 11.0454$

$|\nabla f(4,3)| = \sqrt{7^2+9^2} = 11.4018$

$$\therefore |\nabla f(x,y)| = \begin{bmatrix} 11.4 & 13.3 & 13.0 & 9.9 \\ 14.2 & 11.0 & 4.5 & 11.0 \\ 14.2 & 13.3 & 11.0 & 11.4 \end{bmatrix} \#$$

Problem 2.

If the noise appeared randomly. So we can take a lot of images and then compare each image in every pixel. Then find median value in each pixel. And replace the value in the output image.

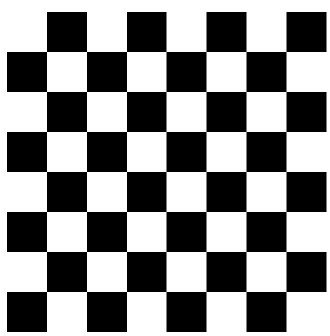


Ex. Value in pixel(x,y) in each image ; 5 6 6 5 0 4 7 5 6 7 0

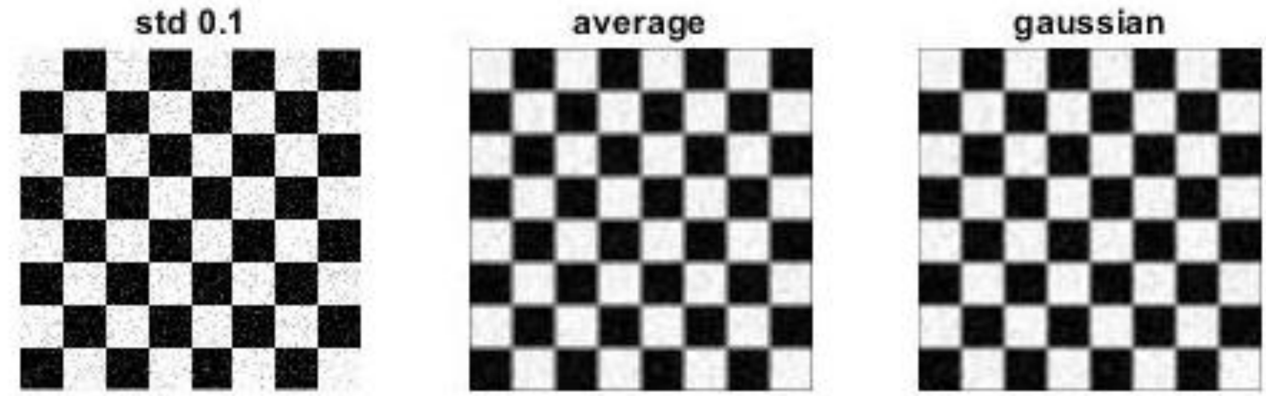
In this method will get the median, 5, which might be not a noise value. Then do it again in other pixels.

Problem 3.

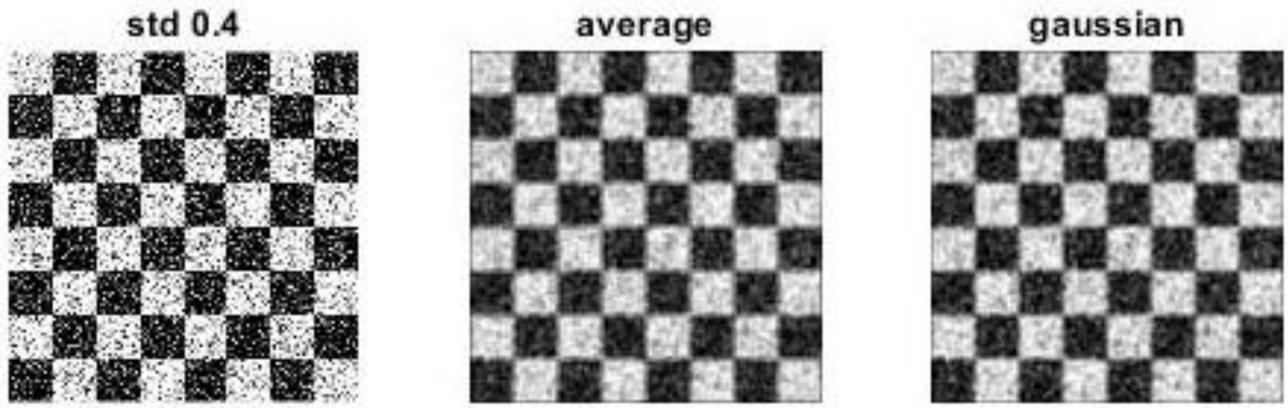
```
[chess map] = imread('chessboard.bmp');
I = ind2gray(chess,map);
c01 = imnoise(I,'gaussian',0,0.01);
c04 = imnoise(I,'gaussian',0,0.16);
average = (1/9)*[1 1 1;1 1 1;1 1 1];
sigma = 1; %standard deviation
gaussian_kernel = zeros(3); %build 3*3 kernel
w = 0; %sum of element
for r = 1:3
    for c = 1:3
        sq_dist = (r-2)^2 + (c-2)^2; %square of distance of any pixel from
center
        gaussian_kernel(r,c) = exp(-1*(sq_dist)/(2*sigma));
        w = w + gaussian_kernel(r,c);
    end
end
gaussian_kernel = abs(gaussian_kernel/w);
chess_01_a = imfilter(c01,average);
chess_01_g = imfilter(c01,gaussian_kernel);
chess_04_a = imfilter(c04,average);
chess_04_g = imfilter(c04,gaussian_kernel);
figure;subplot(131);imshow(c01);title("std 0.1");
subplot(132);imshow(chess_01_a);title("average");
subplot(133);imshow(chess_01_g);title("gaussian");
figure;subplot(131);imshow(c04);title("std 0.4");
subplot(132);imshow(chess_04_a);title("average");
subplot(133);imshow(chess_04_g);title("gaussian");
```



The original image



Zero mean gaussian noise with std = 0.1

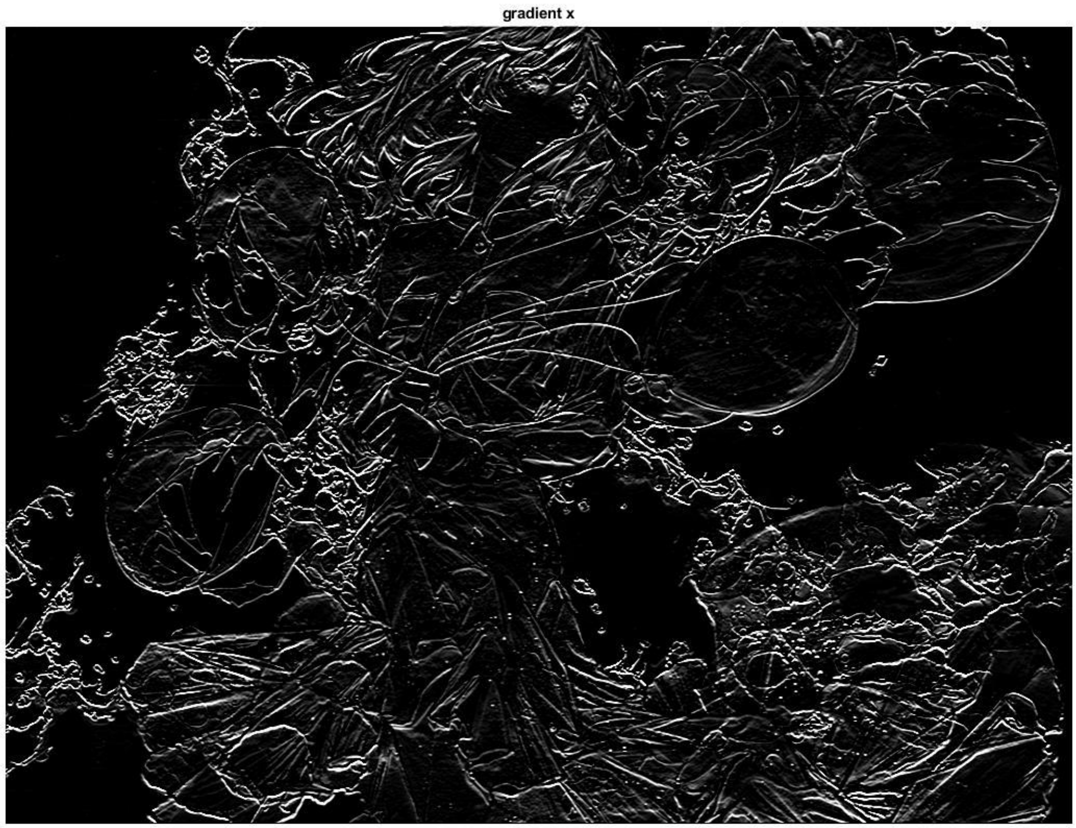


Zero mean gaussian noise with std = 0.4

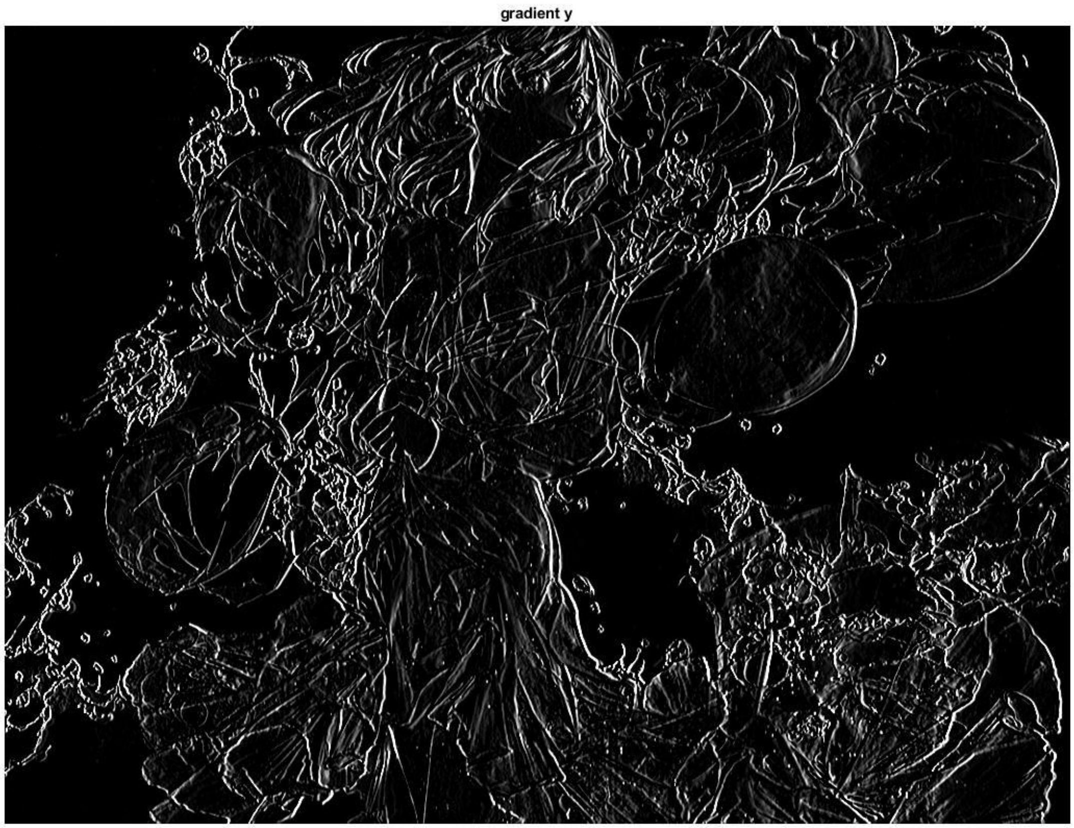
Problem 4.

```
Im = imresize(rgb2gray(imread('test_problem3.png')),0.25);
Im = im2double(Im);
[row col] = size(Im);
horizontal = [-1 -2 -1;0 0 0;1 2 1];
vertical = [-1 0 1;-2 0 2;-1 0 1];
Imx = imfilter(Im,horizontal);
Imy = imfilter(Im,vertical);
subplot(131);imshow(Imx);
subplot(132);imshow(Imy);
gradient_mag = zeros(row,col);
for r = 1:row
    for c = 1:col
        gradient_mag(r,c) = sqrt(Imx(r,c)^2 + Imy(r,c)^2);
    end
end
subplot(133);imshow(gradient_mag);
```

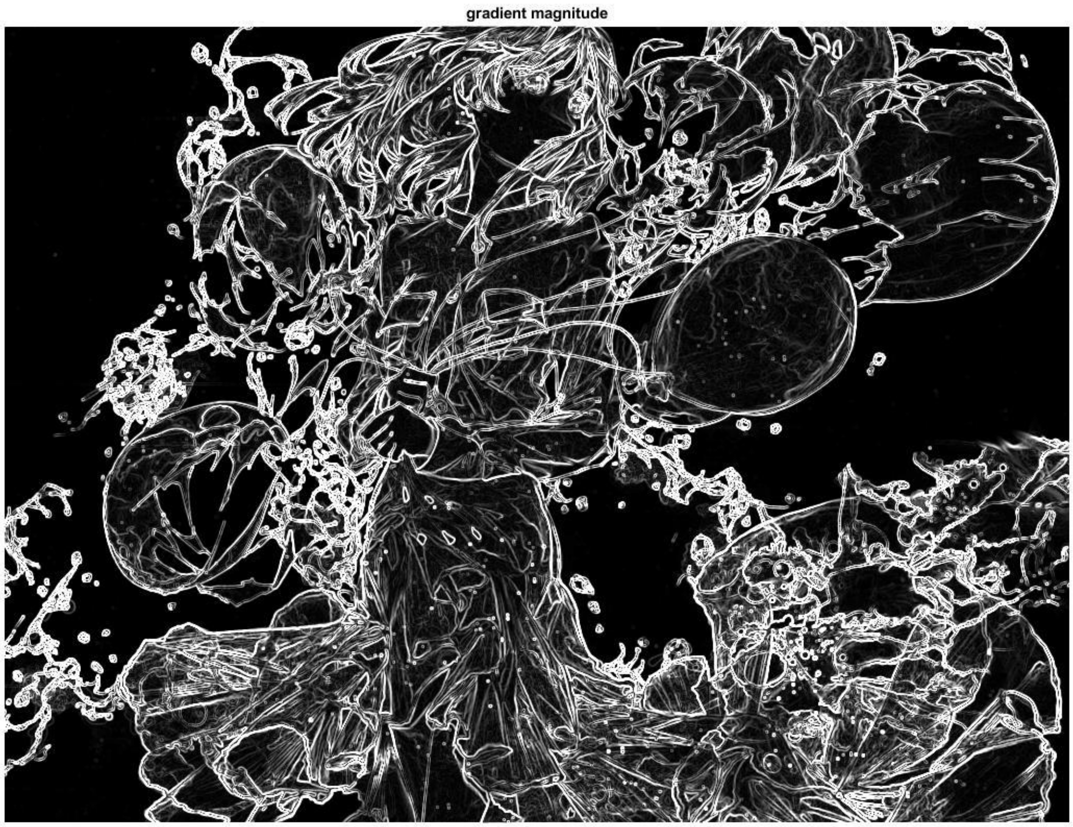
The original anime image



Gradient Ix

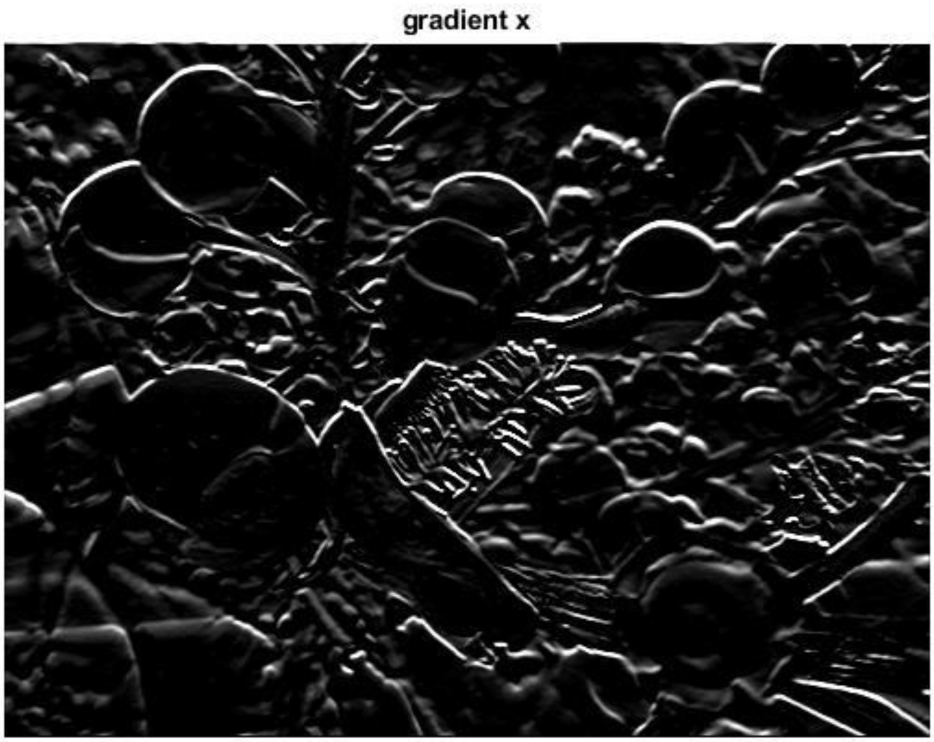


Gradient Iy



Gradient magnitude

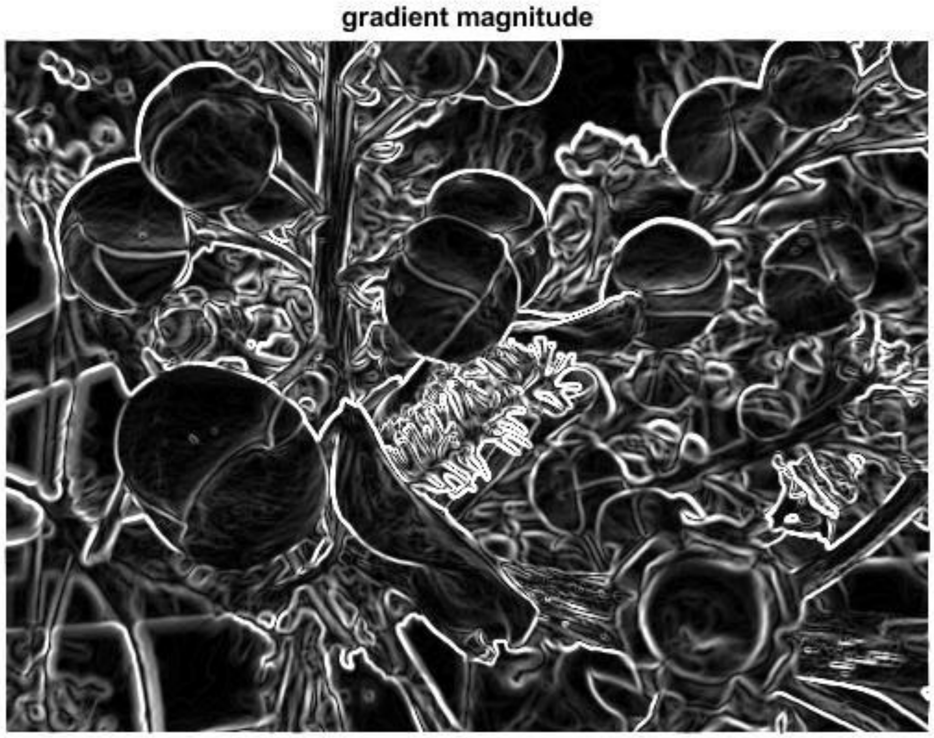
The original flower image



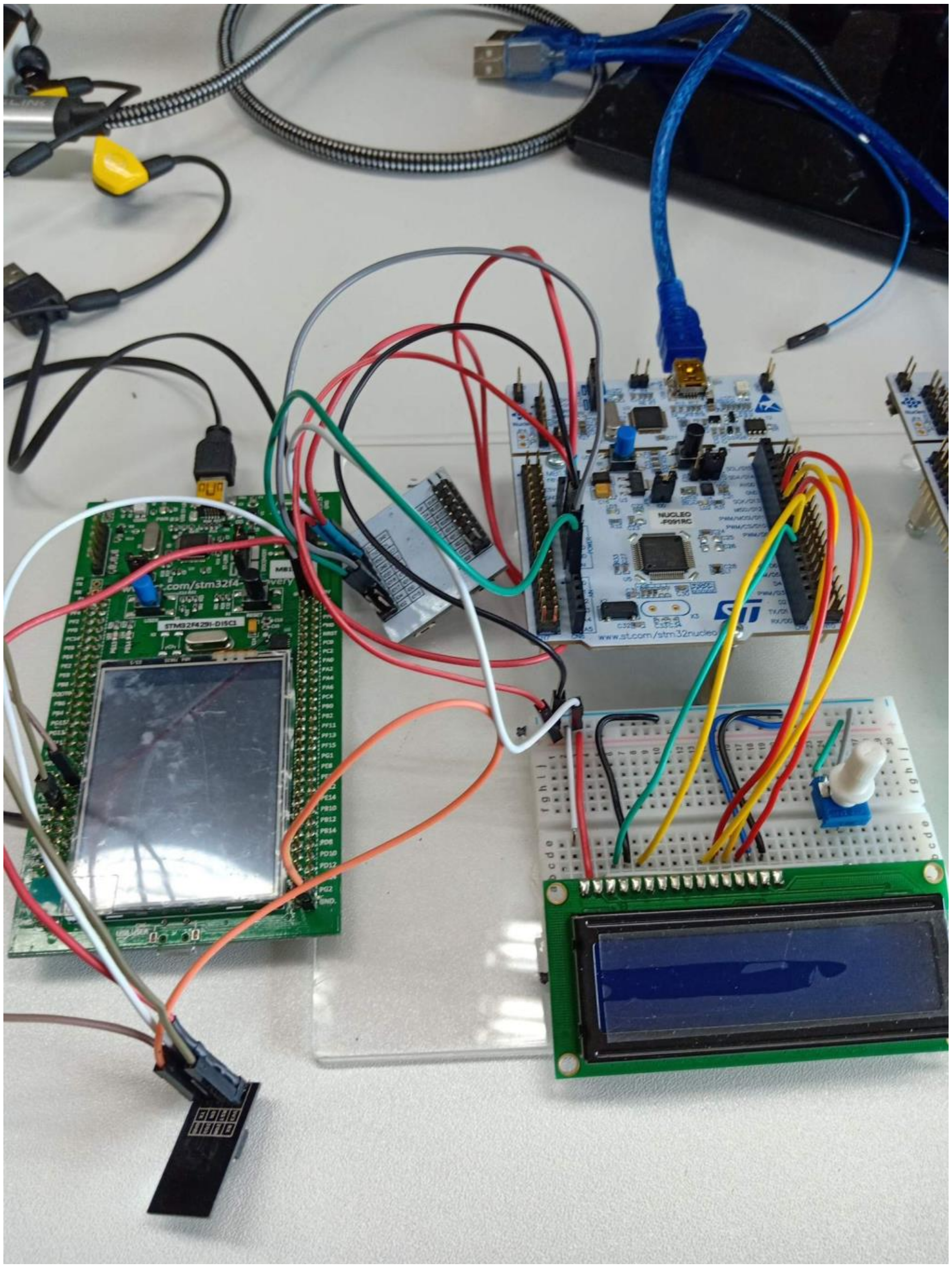
Gradient x

gradient y

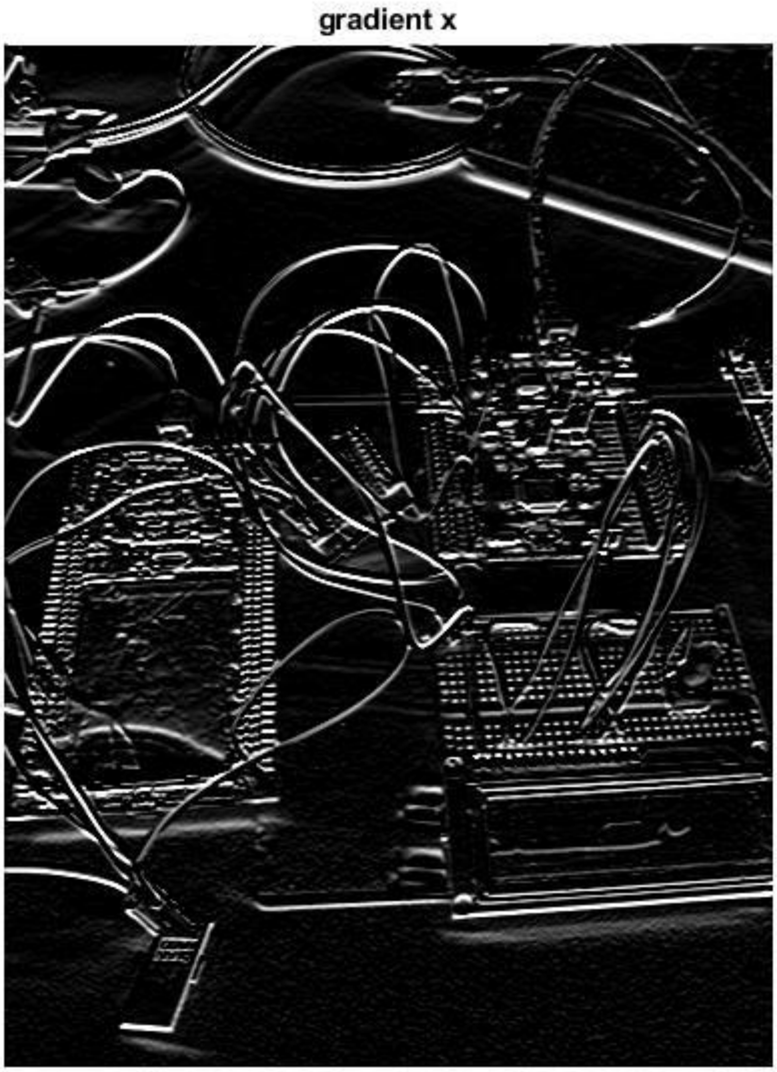
Gradient y



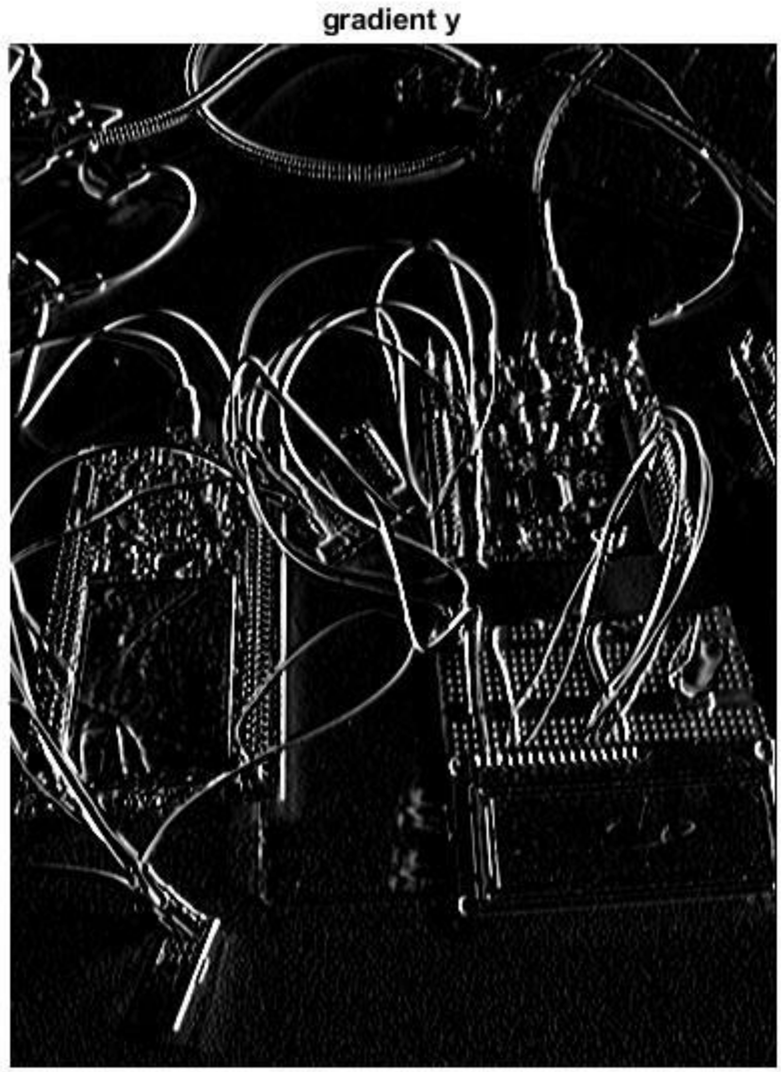
Gradient magnitude



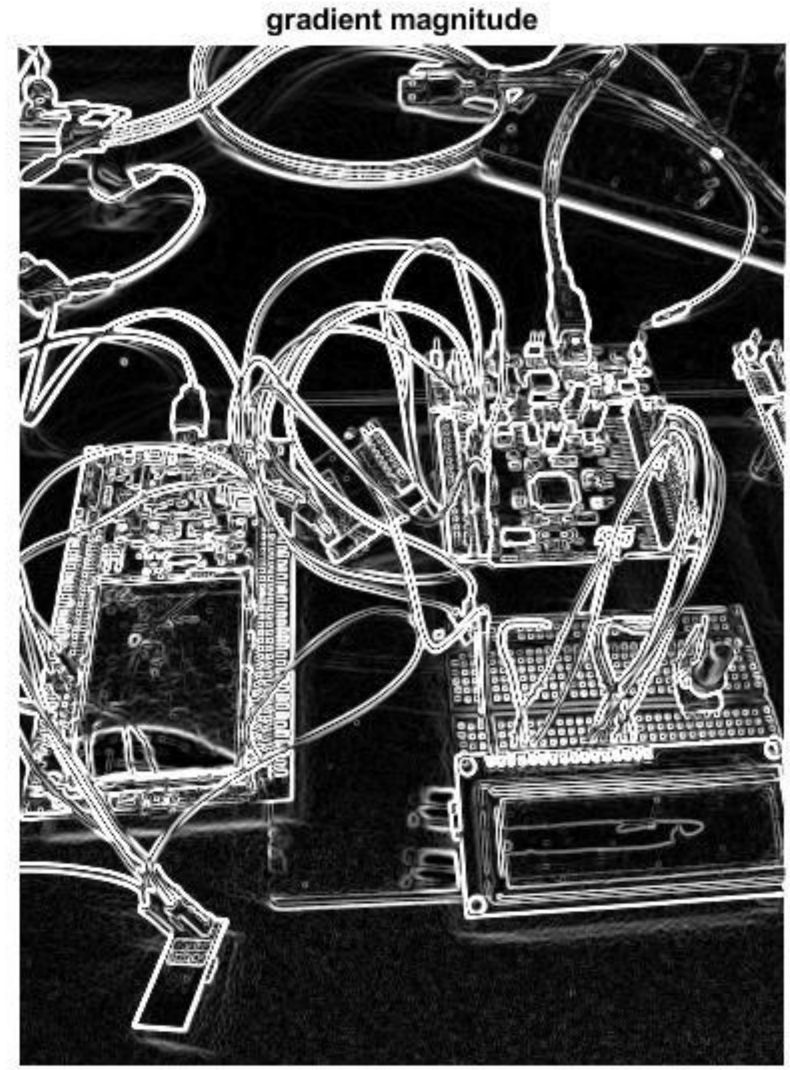
The original circuit image



Gradient x



Gradient y



Gradient magnitude