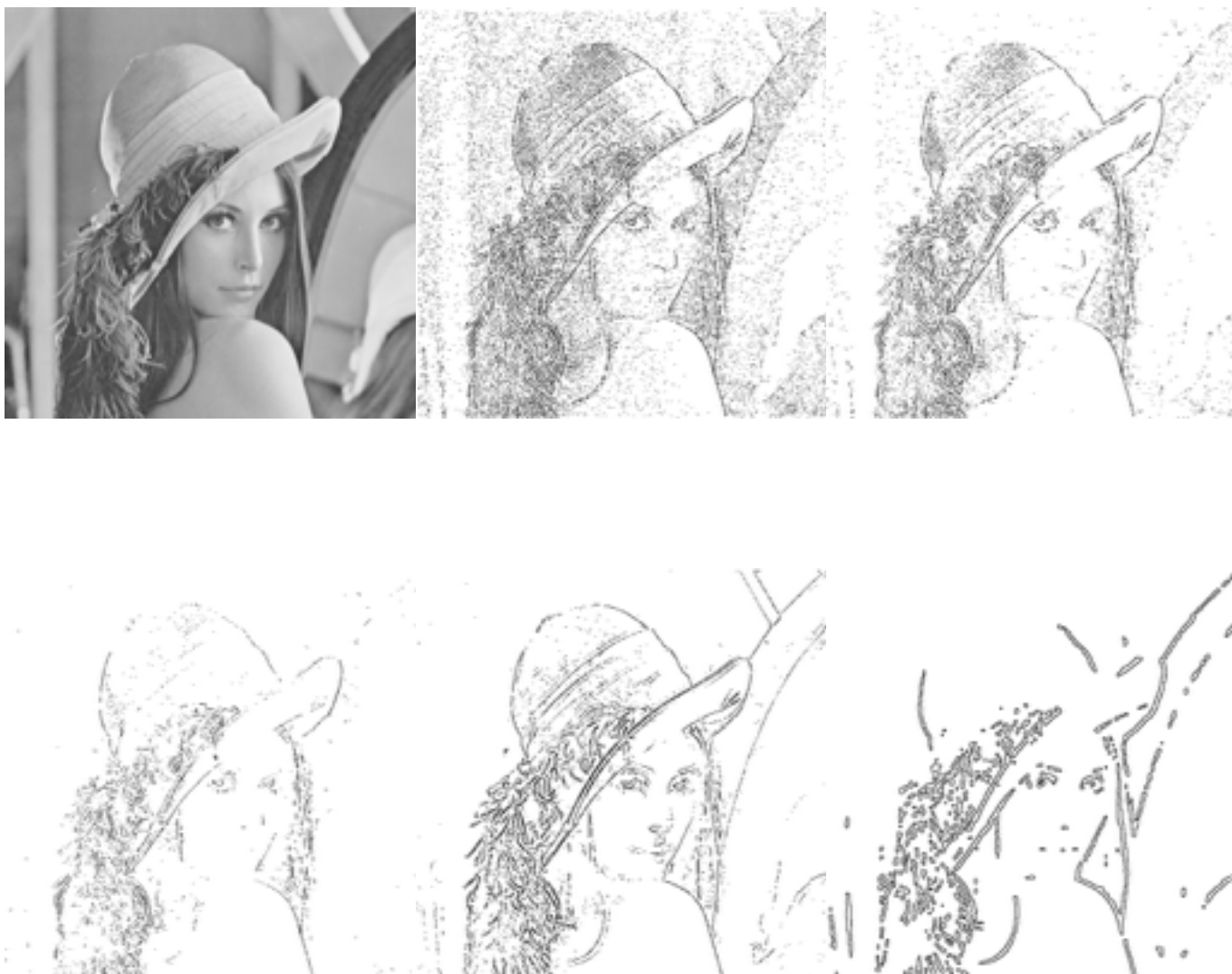# Computer Vision HW10

National Taiwan University

R02944002 王瀚宇

Description

I implemented Laplacian, Minimum Variance Laplacian, Laplacian of Gaussian, and Difference of Gaussian(inhibitory sigma=1, excitatory sigma=3, kernel size 11x11 [1][1])

Parameters

int headerLength = 172;
int imageWidth = 512;
int imageHeight = 512;

Kernels

```
Kernel Laplacian1 = new Kernel(new float[][]{
                { 0, 1, 0},
                { 1,-4, 1},
                { 0, 1, 0}
        },1,1);


Kernel Laplacian2 = new Kernel(new float[][]{
                { 1.0f/3, 1.0f/3, 1.0f/3},
                { 1.0f/3,-8.0f/3, 1.0f/3},
                { 1.0f/3, 1.0f/3, 1.0f/3}
        },1,1);


Kernel MinimumVarianceLaplacian = new Kernel(new float[][]{
                { 2.0f/3, -1.0f/3, 2.0f/3},
                { -1.0f/3,-4.0f/3, -1.0f/3},
                { 2.0f/3, -1.0f/3, 2.0f/3}
        },1,1);
```

```
Kernel LoGKernel = new Kernel(new float[][]{
                    {  0,   0,   0,  -1,  -1,  -2,  -1,  -1,   0,   0,   0},
                    {  0,   0,  -2,  -4,  -8,  -9,  -8,  -4,  -2,   0,   0},
                    {  0,  -2,  -7, -15, -22, -23, -22, -15,  -7,  -2,   0},
                    { -1,  -4, -15, -24, -14,  -1, -14, -24, -15,  -4,  -1},
                    { -1,  -8, -22, -14,  52, 103,  52, -14, -22,  -8,  -1},
                    { -2,  -9, -23,  -1, 103, 178, 103,  -1, -23,  -9,  -2},
                    { -1,  -8, -22, -14,  52, 103,  52, -14, -22,  -8,  -1},
                    { -1,  -4, -15, -24, -14,  -1, -14, -24, -15,  -4,  -1},
                    {  0,  -2,  -7, -15, -22, -23, -22, -15,  -7,  -2,   0},
                    {  0,   0,  -2,  -4,  -8,  -9,  -8,  -4,  -2,   0,   0},
                    {  0,   0,   0,  -1,  -1,  -2,  -1,  -1,   0,   0,   0},
        },5,5);


Kernel DoGKernel = new Kernel(new float[][]{
                    { -1,  -3,  -4,  -6,  -7,  -8,  -7,  -6,  -4,  -3,  -1},
                    { -3,  -5,  -8, -11, -13, -13, -13, -11,  -8,  -5,  -3},
                    { -4,  -8, -12, -16, -17, -17, -17, -16, -12,  -8,  -4},
                    { -6, -11, -16, -16,   0,  15,   0, -16, -16, -11,  -6},
                    { -7, -13, -17,   0,  85, 160,  85,   0, -17, -13,  -7},
                    { -8, -13, -17,  15, 160, 283, 160,  15, -17, -13,  -8},
                    { -7, -13, -17,   0,  85, 160,  85,   0, -17, -13,  -7},
                    { -6, -11, -16, -16,   0,  15,   0, -16, -16, -11,  -6},
                    { -4,  -8, -12, -16, -17, -17, -17, -16, -12,  -8,  -4},
                    { -3,  -5,  -8, -11, -13, -13, -13, -11,  -8,  -5,  -3},
                    { -1,  -3,  -4,  -6,  -7,  -8,  -7,  -6,  -4,  -3,  -1},
        },5,5);
```

Threshold Values listed below:

Laplace Mask Type1: 15
Laplace Mask Type2: 15
Minimum variance Laplacian: 20
Laplace of Gaussian: 3000
Difference of Gaussian: 1

Principal code

```java
public static ArrayList<Integer> CrossingEdgeDetector(ArrayList<Integer>
origin,int headerLength, int width, int height,Kernel kernel,int
threshold)
    {
        ArrayList<Integer> results =
InitWhite(origin,headerLength,width,height);
        ArrayList<Integer> temp =
InitWhite(origin,headerLength,width,height);

        for(int y = 0 ; y < height; y++)
        {
            for(int x = 0 ; x < width ; x++)
            {
                float tempValue =
CalculateKernel(origin,headerLength,width,height,kernel,x,y);
                temp.set(headerLength+y*width+x,(int)tempValue);
            }
        }

        for(int y = 0 ; y < height; y++)
        {
            for(int x = 0 ; x < width ; x++)
            {
                for(int y2 = -1; y2 < 2 ; y2++)
                {
                    //System.out.println(x+":"+y);

                    for(int x2 = -1; x2 < 2 ; x2++)
                    {

if(isDifferenceGreaterThan(temp,headerLength,width,height,x,y,x2,y2,thre
shold))
                        {
                            results.set(headerLength+y*width
+x,0);
                        }
                    }
                }
            }
        }

        return results;
    }
```

```java
    public static boolean isDifferenceGreaterThan(ArrayList<Integer>
origin,int headerLength, int width, int height,int x,int y,int x2,int
y2,int threshold)
    {
        int newIndexX = x + x2;
        int newIndexY = y + y2;

        if(newIndexX < 0) return false;
        if(newIndexY < 0) return false;
        if(newIndexX >= width) return false;
        if(newIndexY >= height) return false;

        int originValue = origin.get(headerLength+width*y+x);
        int nearValue = origin.get(headerLength+width*newIndexY
+newIndexX);

        if(originValue > threshold && nearValue < -threshold)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
```

Results

Laplacian Mask Type 1 - Threshold : 15

Laplacian Mask Type 2 - Threshold : 15

MinimumVarianceLaplacian - Threshold : 20

Laplace of Gaussian - Threshold : 3000

Difference of Gaussian - Threshold : 1