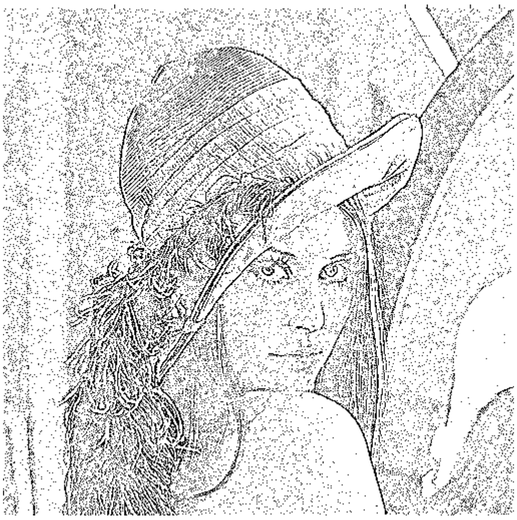
資工所碩一 R07922164 林群崴

本次作業採用 C++配合 OpenCV 完成

```
Mat convolve(Mat a, Mat b) {
    Mat ans(a.rows - b.rows, a.cols - b.cols, CV_64F, Scalar(0));
    for (int i = 0; i < ans.rows; i++) {
        for (int j = 0; j < ans.cols; j++) {
            double result = 0;
            for (int k = 0; k < b.rows; k++) {
                for (int z = 0; z < b.cols; z++) {
                    result += b.at<double>(b.rows - k - 1, b.cols - z - 1) * a.at<uchar>(i + k, j + z);
                }
            }
            ans.at<double>(i, j) = result;
        }
    }
    return ans;
}
```

1. Laplace Mask1 (0, 1, 0, 1, -4, 1, 0, 1, 0): 15

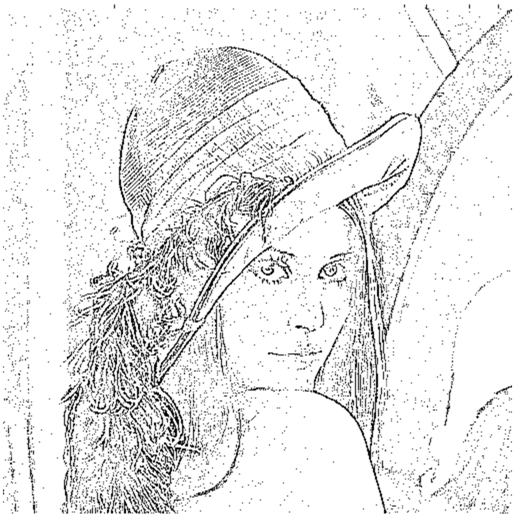
```
Mat LaplaceMask1(Mat img) {
    Mat mask(3, 3, CV_64F, Scalar(0));
    mask.at<double>(0, 0) = 0; mask.at<double>(0, 1) = 1; mask.at<double>(0, 2) = 0;
    mask.at<double>(1, 0) = 1; mask.at<double>(1, 1) = -4; mask.at<double>(1, 2) = 1;
    mask.at<double>(2, 0) = 0; mask.at<double>(2, 1) = 1; mask.at<double>(2, 2) = 0;
    Mat result = binary(convolve(img, mask), 15);
    return result;
}
```



採用 15 作為 threshold value

2. Laplace Mask2 (1, 1, 1, 1, -8, 1, 1, 1, 1)

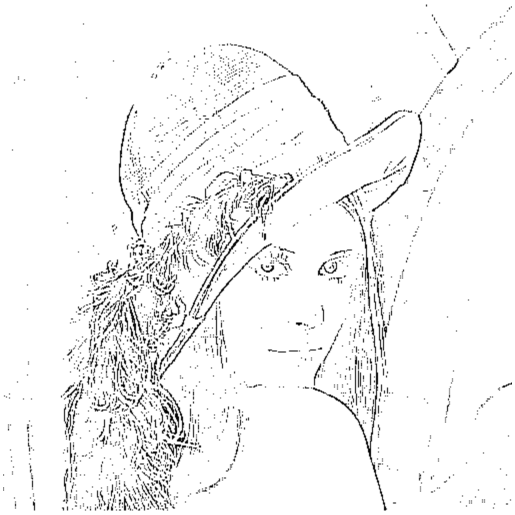
```
Mat LaplaceMask2(Mat img) {
	Mat mask(3, 3, CV_64F, Scalar(0));
	mask.at<double>(0, 0) = 1. / 3; mask.at<double>(0, 1) = 1. / 3;
mask.at<double>(0, 2) = 1. / 3;
	mask.at<double>(1, 0) = 1. / 3; mask.at<double>(1, 1) = -8. / 3;
mask.at<double>(1, 2) = 1. / 3;
	mask.at<double>(2, 0) = 1. / 3; mask.at<double>(2, 1) = 1. / 3;
mask.at<double>(2, 2) = 1. / 3;
	Mat result = binary(convolve(img, mask), 15);
	return result;
}
```



採用 15 作為 threshold value

3. Minimum variance Laplacian: 20

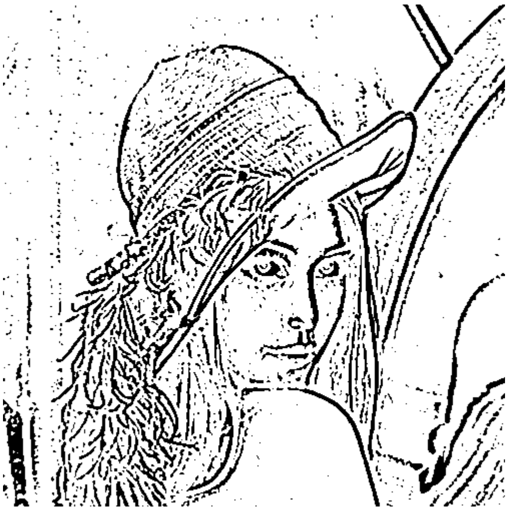
```
Mat MinimumVarianceLaplacian(Mat img) {
	Mat mask(3, 3, CV_64F, Scalar(0));
	double m[11][11] = { {2, -1, 2},
			{-1, -4, -1},
			{2, -1, 2} };
	for (int i = 0; i < 3; i++) {
		for (int j = 0; j < 3; j++) {
			mask.at<double>(i, j) = m[i][j] / 3;
		}
	}
	Mat result = binary(convolve(img, mask), 20);
	return result;
}
```



採用 20 作為 threshold value

4. Laplace of Gaussian: 3000

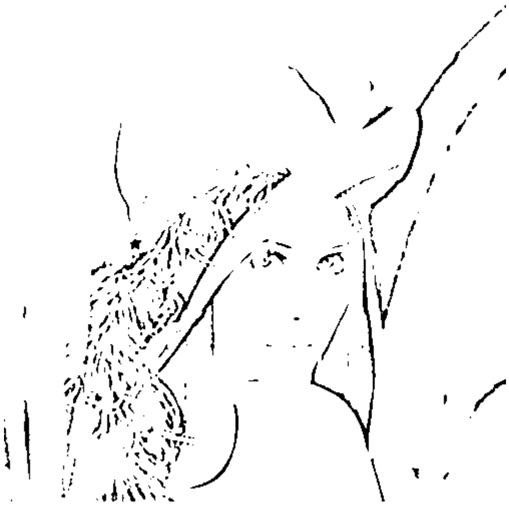
```
Mat LaplaceOfGaussian(Mat img) {
    double m[11][11] = { {0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0},
        {0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0},
        {0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0},
        {-1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1},
        {-1, -8, -22, -14, 52, 103, 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} };
    Mat mask(11, 11, CV_64F, Scalar(0));
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            mask.at<double>(i, j) = m[i][j];
        }
    }
    Mat result = binary(convolve(img, mask), 3000);
    return result;
}
```



採用 3000 作為 threshold value

5. Difference of Gaussian: 1

```
Mat DifferenceOfGaussian(Mat img) {
    double m[11][11] = { {-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}, };
    Mat mask(11, 11, CV_64F, Scalar(0));
    for (int i = 0; i < 11; i++) {
        for (int j = 0; j < 11; j++) {
            mask.at<double>(i, j) = m[i][j];
        }
    }
    Mat result = convolve(img, mask);
    for (int height = 0; height < result.rows; height++) {
        double *data = result.ptr<double>(height);
        for (int width = 0; width < result.cols; width++) {
            data[width] = data[width] < 1 ? 0 : 255;
        }
    }
    return result;
}
```



採用 1 作為 threshold value

檔案名稱說明：

依照所作的操作為名字的.bmp 檔為其結果