

這次作業實作以不同的 edge detector 做影像邊界偵測處理，並探討其中差異。對彩色圖像依序進行灰階、高斯濾波後，以要求的 Prewitt/Sobel/Laplacain edge detector 分別處理並輸出圖像，探討不同方法之間的差異。

➤ Describe the main part of your method.

```
// Gaussian 處理的類別  
class GaussianFilter
```

如果要有較佳的 edge detector 結果，先執行 Noise reduction/filter operation 預處理是必要的，這樣會讓結果圖有較分明與清晰的邊緣曲線，我使用 gaussian filter 作為每個 edge detector 的預處理，而結果也與未進行預處理差別很大，filter 的方法與上次作業相同並寫在 Gaussian class 中，使用的 kernel 大小為 3x3，sigma 值為 1.414。

```
// Edge 偵測的類別  
class EdgeDetector
```

所有的 edge detector 都在這個 class 中，是這次作業的主要方法撰寫類別。

```
// 用 3*3 卷積回傳鄰近的像素點  
vector<uchar> getConvolutionList(const Mat& source)
```

取得中心點附近 3x3 的像素值，中心點的值由其他函式設定存入類別成員中。

```
// 取得目前計算 edge 方法的單點值  
int getEdgeValue(const Mat& source, const vector<uchar>& pixelList, const vector<int>& kernel)
```

根據中心點的值（由其他函式設定）、getConvolutionList 取得的 pixelList 和特定方法設計的 kernel 給予 edge 特徵值。

```
// 回傳是否超過邊界值的布林值  
bool isOverThreshold(const int& threshold, const int& value1, const int& value2)
```

回傳是否超過邊界值的布林值函式，用途為查看給予的 value 經過運算是否超出 threshold，公式：兩變數 =  $|value1| + |value2|$ 、單變數 =  $|value1|$ 。

```
// 輸出 edge 偵測影像  
void detectEdge(const Mat& source, Mat& dest, EdgeOperatorName edgeOperator, int threshold)
```

根據提供的 edgeOperator 選項，進行邊緣偵測的處理，並輸出處理後的圖像，其中用到了設定中心點的類別成員、getConvolutionList、getEdgeValue、isOverThreshold。

```
// 取得特定方法過濾圖像  
void getEdgeImageByMethod(const Mat& source, Mat& dest, EdgeOperatorName edgeOperator, int threshold)
```

main()函式只有呼叫這個函式來完成邊緣偵測的輸出要求，這個函式主要根據 edgeOperator 來決定要使用 Prewitt/Sobel/Laplacain 哪一種 kernel 做運算，之後呼叫 detectEdge 來完成工作。

- Discuss horizontal edge、vertical edge and both edge differences.

Horizontal edge：能偵測到橫向的顏色強度變化。

Vertical edge：能偵測到縱向的顏色強度變化。

Both edge：能偵測到對角的顏色強度變化，類似於將 Horizontal 和 Vertical edge 結合呈現出來的圖形。

- Discuss Laplacian kernels differences.

L1：

0	1	0
1	-4	1
0	1	0

L2：

1	1	1
1	-8	1
1	1	1

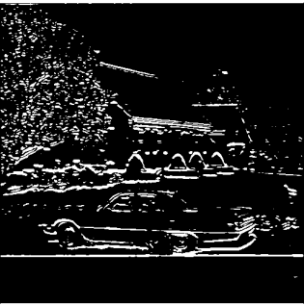
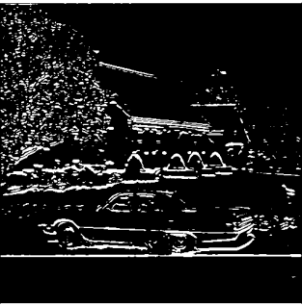
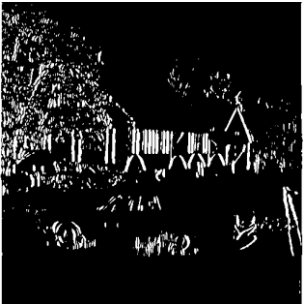
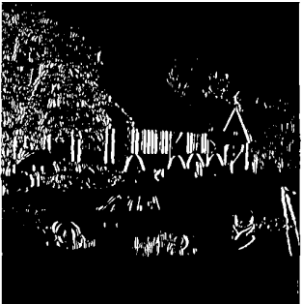




對於 L1 來說，對於雜訊較不敏感，邊緣的線會比 L2 來的粗。

對於 L2 來說，對於雜訊較敏感，邊緣的線會比 L1 來的薄。

- Result images.









**\*\*Next page\*\***

House512.png

edge	Prewitt	Sobel
Horizontal		
Vertical		
Both		
	Laplacian 1	Laplacian 2
		



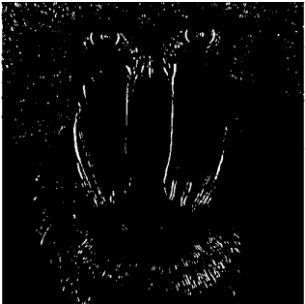
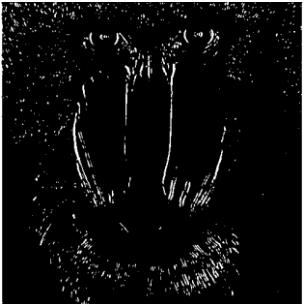




	Threshold
Prewitt	60
Sobel	80
Laplacian 1	10
Laplacian 2	25

Lena.png

edge	Prewitt	Sobel
Horizontal		
Vertical		
Both		
	Laplacian 1	Laplacian 2
		

	Threshold
Prewitt	80
Sobel	95
Laplacian 1	10
Laplacian 2	25

Mandrill.png

edge	Prewitt	Sobel
Horizontal		
Vertical		
Both		
	Laplacian 1	Laplacian 2
		

	Threshold
Prewitt	90
Sobel	110
Laplacian 1	8
Laplacian 2	25