**109590004 呂育瑋** Machine Vision HW#5

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。

步驟如下：

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 GassianFilter

如果要有較佳的edge operator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 operator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 operator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。

// 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。

// 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。

// 取得特定方法過濾圖像

    void getFilterImageByMode(Mat& source, Mat& dest, const string& mode, int kernelSize)

根據提供的原圖像source以mode方法的filter和視窗大小kernelSize2輸出圖像至dest，利用函式指標依照mode指向特定的get{mode}Value()以不同方法處理像素值，而mode = gaussian會需要先呼叫過setGaussianKernel()。

* Bonus: Discuses the result after repeat 7 times
* Mean filter 3\*3 能得到比較平均的變更值，在去雜訊中還是有些雜訊沒被濾除，這跟kernel太小有關。
* Mean filter 7\*7 能得到比Mean filter 3\*3更平均的變更值，且沒有明顯雜訊，但圖像會看起來更模糊，許多細節被濾除，這跟kernel太大有關。
* Median filter 3\*3 能得到比較合理的變更值，在去雜訊有顯著的效果，灰階差距較大的地方能看到強烈的邊緣，kernel並不會太大，是蠻理想的處理方式。
* Median filter 7\*7 的效果與Median filter 3\*3相似，去雜訊時圖像細節被嚴重忽略，較不理想的處理結果，這跟kernel太大有關。
* Gaussian filter 5\*5，我使用的σ= 1.414，是在逐漸調整下得出比上述filter效果還要好的參數值，特色是能盡量保留細節、圖像不會過度模糊，5\*5 的kernel有較佳的去雜訊效果。
* Result images ( total = 140 pics, document imports 40 pics, 1 times and 7 times)

\*\*At next page\*\*

Lena\_gray (repeat 1, repeat 7)

* Mean 3\*3

* Mean 7\*7

* Median 3\*3

* Median 7\*7

* Gaussian 5\*5 σ= 1.414

House256\_noise (repeat 1, repeat 7)

* Mean 3\*3

* Mean 7\*7

* Median 3\*3

* Median 7\*7

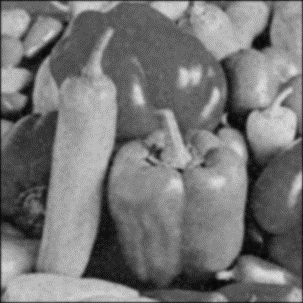
 

* Gaussian 5\*5 σ= 1.414

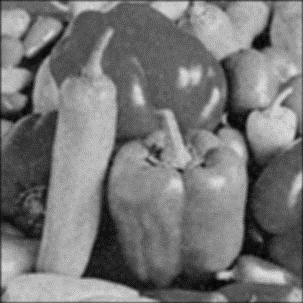
 

Peppers\_noise (repeat 1, repeat 7)

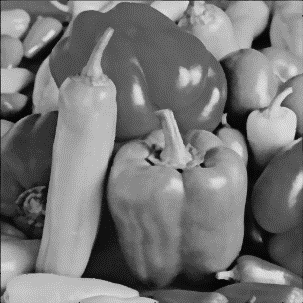
* Mean 3\*3

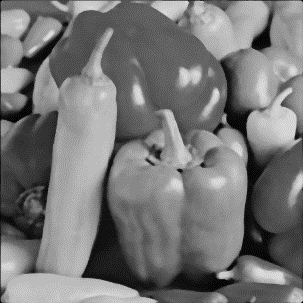
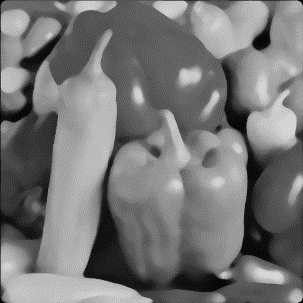
* Mean 7\*7

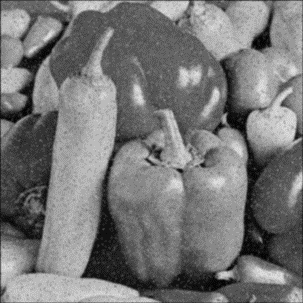
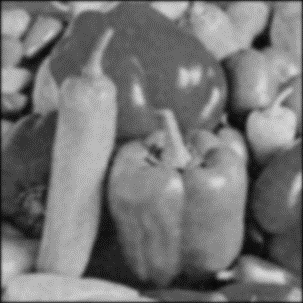
* Median 3\*3

* Median 7\*7

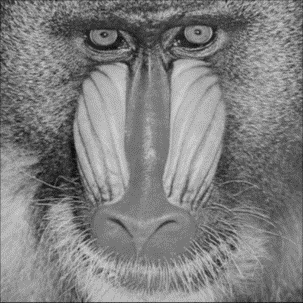
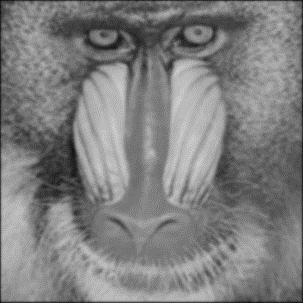
 

* Gaussian 5\*5 σ= 1.414

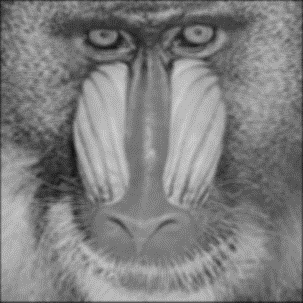
 

Mandrill\_gray (repeat 1, repeat 7)

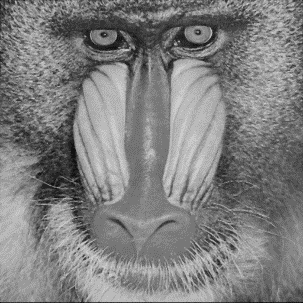
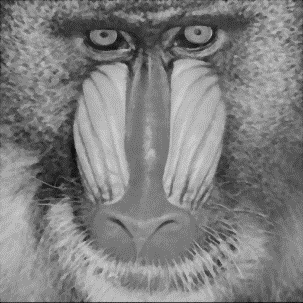
* Mean 3\*3

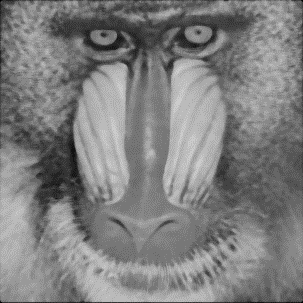
* Mean 7\*7

* Median 3\*3

* Median 7\*7

* Gaussian 5\*5 σ= 1.414

