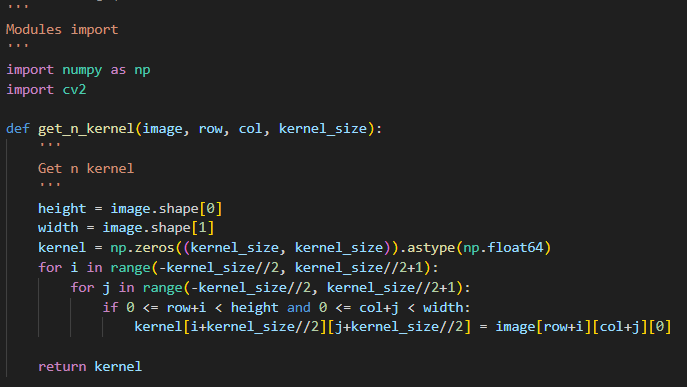
班級：資工三

學號：110590034

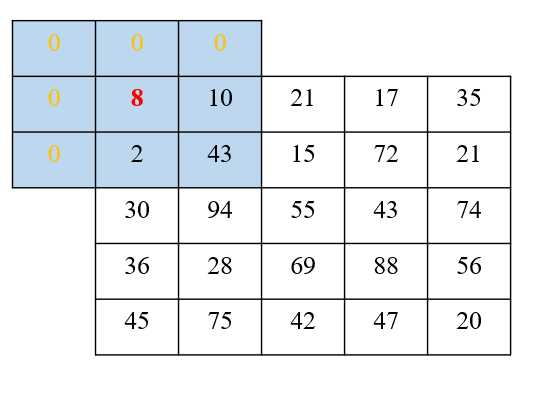
姓名：楊榮鈞

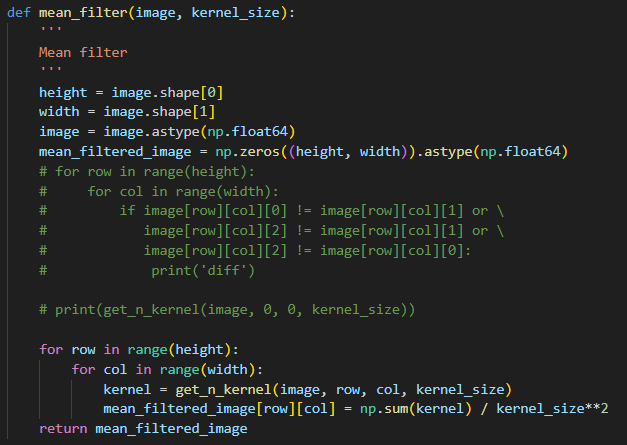


Function get\_n\_kernel:

先用image.shape找出height和width，再根據row和col(當前pixel的位置)抓取kernel\_size x kernel\_size大小的kernel。

如作業附圖的藍色的部分。



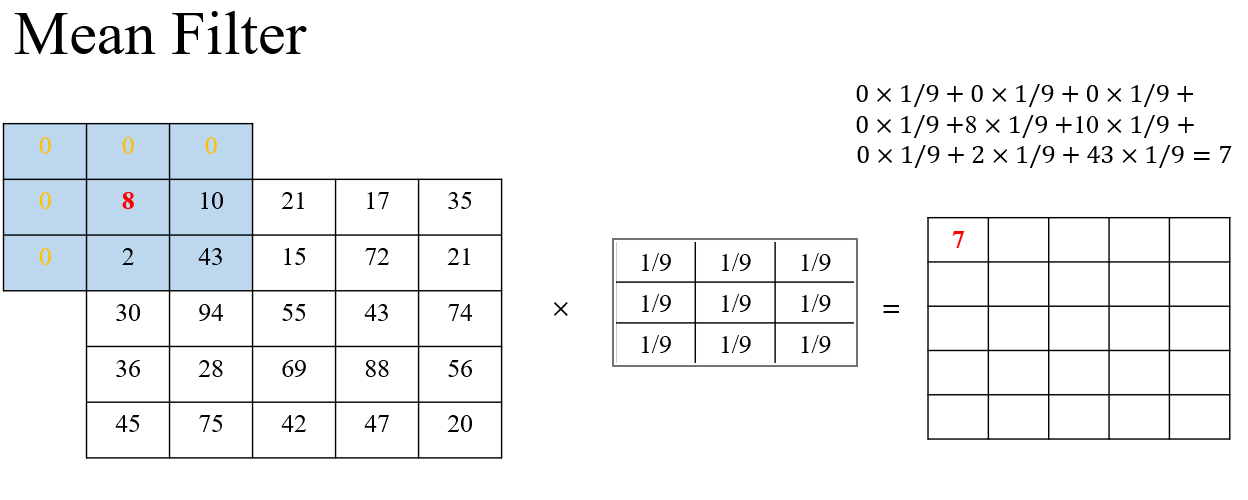


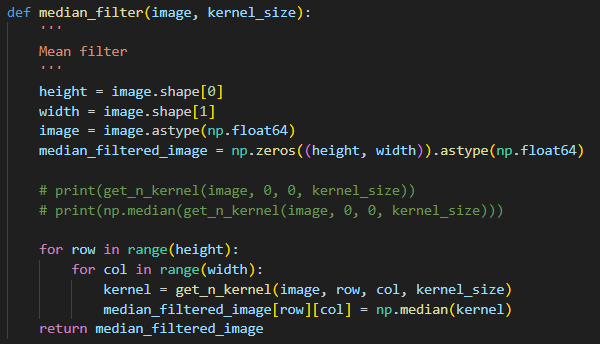
Function mean\_filter:

利用shape找出image的height和width，並利用astype設定image的type為np.float64，然後利用np.zeros創建mean\_filtered\_image的array。

註解的部分是測試image的三個通道的value是否皆相等，由於之前測試的結果皆為相等，因此有先註解使程式跑快一點。

接著利用for迴圈將計算每個pixel的value，其中會先用get\_n\_kernel抓取當前pixel的kernel，並利用np.sum(kernel) / kernel\_size \*\* 2計算mean filter，如同作業的PPT提到的算法，我將1/9的部分(1/ kernel\_size \*\* 2)提出來在最後相乘。

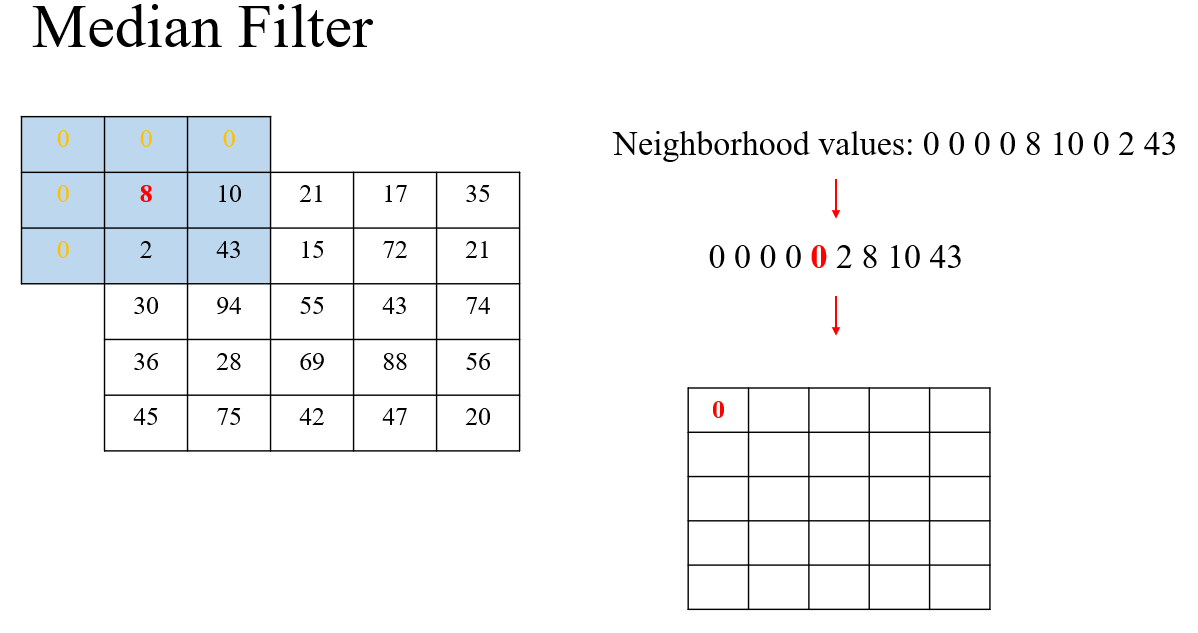


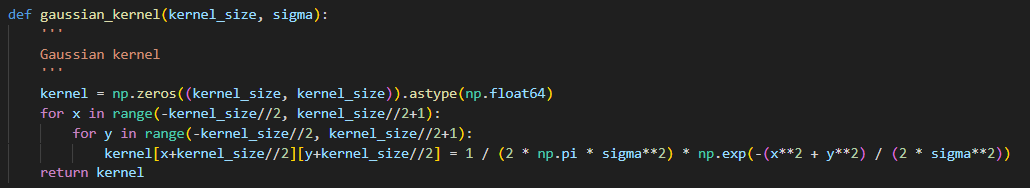


Function median\_filter:

利用shape找出image的height和width，並利用astype設定image的type為np.float64，然後利用np.zeros創建median\_filtered\_image的array。

接著利用for迴圈將計算每個pixel的value，其中會先用get\_n\_kernel抓取當前pixel的kernel，並利用np.median(kernel)找出kernel中的中值(median value)。

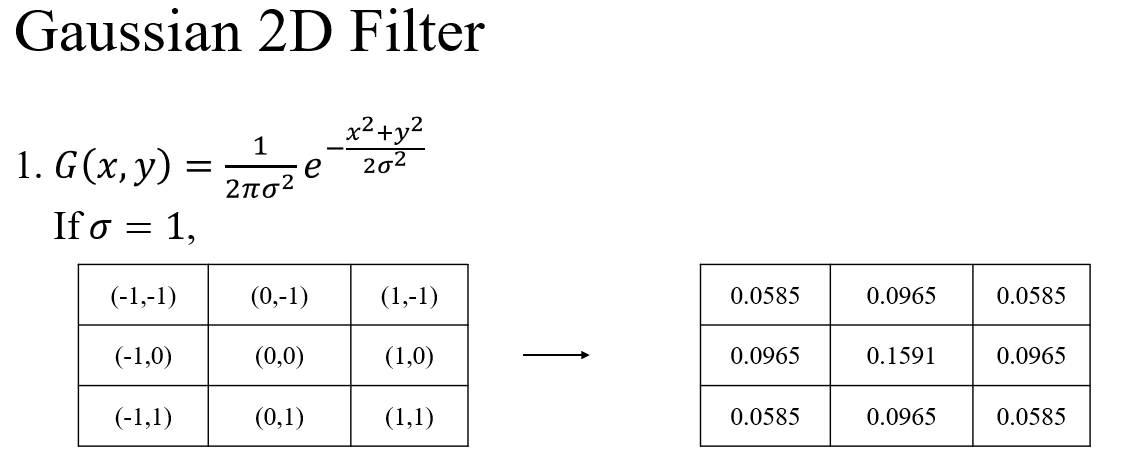


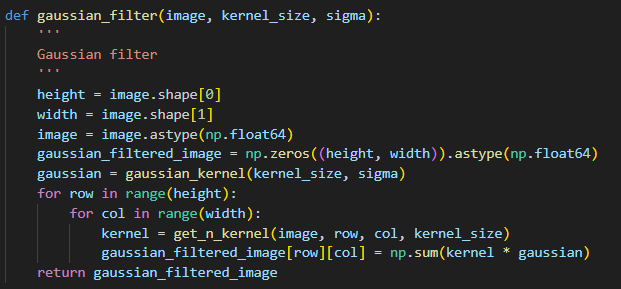


Function gaussian\_kernel:

計算gaussian的value，以Gaussian 2D filter的公式進行計算，如作業簡報的公式算法 G(x, y) = (1 / 2 \* pi \* sigma \*\* 2) \* exp \*\* (-(x \*\* 2 + y \*\* 2) / (2 \* sigma \*\* 2))。

其中會以kernel\_size的大小去做整個kernel的處理如圖所示(圖片是3x3的kernel大小)。





Function gaussian\_filter:

利用shape找出image的height和width，並利用astype設定image的type為np.float64，然後利用np.zeros創建gaussian\_filtered\_image的array。

利用gaussian\_kernel找出kernel的gaussian value並存在gaussian中，利用for迴圈計算每個pixel的value，其中會先用get\_n\_kernel抓取當前pixel的kernel，並利用np.sum(kernel \* gaussian)得出kernel \* gaussian的總和，再存入當前pixel的value中。



Function image:

先用cv2的imread讀取圖片到origin\_image。

接著使用mean\_filter(origin\_image, 3)得出mean\_filter且kernel為3x3大小的圖片並存入mean\_filtered\_image\_3中，再來使用cv2.imwrite存檔，最後再使用cv2.imread讀取圖片到mean\_3。

接著使用mean\_filter(origin\_image, 7)得出mean\_filter且kernel為7x7大小的圖片並存入mean\_filtered\_image\_7中，再來使用cv2.imwrite存檔，最後再使用cv2.imread讀取圖片到mean\_7。

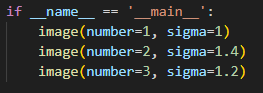
接著使用median\_filter(origin\_image, 3)得出median\_filter且kernel為3x3大小的圖片並存入median\_filtered\_image\_3中，再來使用cv2.imwrite存檔，最後再使用cv2.imread讀取圖片到median\_3。

接著使用median\_filter(origin\_image, 7)得出median\_filter且kernel為7x7大小的圖片並存入median\_filtered\_image\_7中，再來使用cv2.imwrite存檔，最後再使用cv2.imread讀取圖片到median\_7。

接著使用gaussian\_filter(origin\_image, 5, sigma)得出gaussian\_filter且kernel為5x5大小和傳入的sigma做運算後的圖片，並存入gaussian\_filtered\_image中，再來使用cv2.imwrite存檔，最後再使用cv2.imread讀取圖片到gaussian。

利用cv2的imshow查看original image、mean\_3(Mean Filtered Image (3x3))、mean\_7(Mean Filtered Image (7x7))、median\_3(Median Filtered Image (3x3))、median\_7(Median Filtered Image (7x7))和gaussian(Gaussian Filtered Image)。

利用cv2的waitKey(0)和destroyAllWindows()將imshow顯示出來的圖片關閉。



執行image，產生所有助教給的圖片的Mean Filtered Image (3x3 kernel size)、Mean Filtered Image (7x7 kernel size)、Median Filtered Image (3x3 kernel size)、Median Filtered Image (7x7 kernel size)和Gaussian Filtered Image。

Result images

img1\_q1\_3.jpg



img1\_q1\_7.jpg



img1\_q2\_3.jpg



img1\_q2\_7.jpg



img1\_q3.jpg



img2\_q1\_3.jpg



img2\_q1\_7.jpg



img2\_q2\_3.jpg



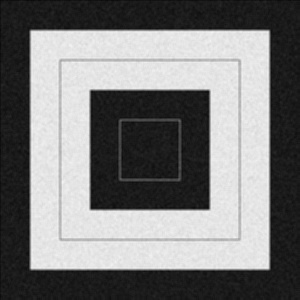
img2\_q2\_7.jpg



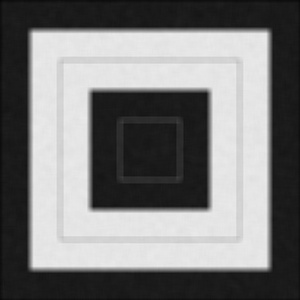
img2\_q3.jpg



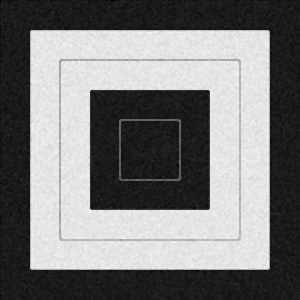
img3\_q1\_3.jpg



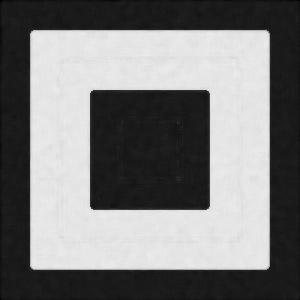
img3\_q1\_7.jpg



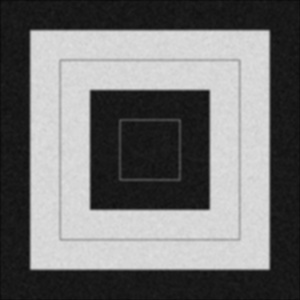
img3\_q2\_3.jpg



img3\_q2\_7.jpg



img3\_q3.jpg



最後的結果總結：

這次hw5