**Ch5 影像增強**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **影像負片**     |  |  | | --- | --- | | 0，以灰度模式讀取圖像 |  | | 1，以彩色模式讀取圖像 |  | |
| 伽瑪矯正     |  |  | | --- | --- | | Gamma=0.1 | Gamma=0.2 | | Gamma=0.5 | Gamma=1.0 | | Gamma=5.0 | Gamma=10.0 | |
| **Beta矯正**     |  |  | | --- | --- | | Beta(a=b=0.5) | Beta(a=b=2.0) | |
| **直方圖**     |  |  | | --- | --- | |  |  | |
| **直方圖等化**     |  |  | | --- | --- | |  |  | |
| **卷積** |
| **卷積2D** |
| **平均濾波**      原圖    Average Filtering(2,2)    Average Filtering(5,5)    Average Filtering(10,10) 參數愈大濾波效果越明顯 |
| **高斯濾波**      **原圖**    **Gaussian Filtering(5,5)**    **Gaussian Filtering(9,9) 與平均濾波結果一樣，當參數愈大濾波效果越明顯。**  但高斯濾波的參數必須是基數，否則會觸發斷言失敗的錯誤。 因為高斯濾波的核是一個平均分佈的權重，中心點表示最高權重，其他權重按照高斯分佈函數遞減。為了保證濾波的效果，這個核的大小必須是奇數，因為這樣才能確保中心點在圖像上，而且大小必須大於0，否則就沒有濾波效果 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **影像梯度**   |  |  | | --- | --- | | 原圖 | Gradient in x | | Gradient in y | Gradient | |
| **拉普拉斯運算子**    <-原圖   |  |  | | --- | --- | | cv2.CV\_32F | cv2.CV\_16S | |
| **混合拉普拉斯運算子**    <-原圖   |  |  | | --- | --- | | composite\_laplacian\_32F | composite\_laplacian\_16S |   將圖片放大後仔細對比，發現湖中的水圈變得更明顯，白色線條有被加強 |

|  |
| --- |
| **非銳化遮罩**    <-原圖  <-unsharp\_masking(1.5)  <-unsharp\_masking(10.0)在執行程式時發現若  #g\_mask = int(f[x, y]) - int(f\_avg[x, y])會發生以下報錯  #TypeError: only size-1 arrays can be converted to Python scalars  因此將程式碼改為:g\_mask = np.uint8(f[x, y]) - np.uint8(f\_avg[x, y])  避免因將圖像數組中的像素值轉換為int型，而導致精度損失和意外的行為。所以使用np.uint8將其轉換為8位無符號整數。 |
| **雙邊濾波**     |  |  | | --- | --- | | 原圖 | bilateralFilter(5,50,50) |     bilateralFilter(10,50,50).jpg  特地找了一張只有臉部且有雀斑的圖片做為代碼測試對象， 可以發現當d為5時，臉部的雀斑有淡化但並不明顯  當d為10時，臉部的雀斑就有明顯的差異了。  為了測試sigmaColor與sigmaSpace的值增強是否有區別 因此生成出左:bilateralFilter(5,70,70)  與右: bilateralFilter(5,100,100) 發現並無明顯區別。 |