**Lab 07**

|  |  |
| --- | --- |
| Name: | 易峻葦 |
| Student ID: | B08611031 |
| Total Score: |  |

**Note:**

Most of the explanations in this lab is mandatory, However, giving reasonable explanations to your answer or programs will earn you partial credits when your answer is incorrect.

1. **Filters and Convolution (25 points, 5 points each question)**

|  |  |  |
| --- | --- | --- |
| # | Description | Score |
| a | 提取橫線特徵 |  |
| b | 提取右上左下的斜線特徵 |  |
| c | 對圖片的橫向、直向、以及兩個對角都提取一次特徵 |  |
| d | 提取圖片的直線特徵 |  |
| e | 檢測出圖片的輪廓，若一個畫素的相鄰左右的值與相鄰上下的值差異越小，卷積後的數值會越小 |  |

1. **Denoising a Picture (15 points)**

|  |  |  |
| --- | --- | --- |
| # | Description | Score |
| - | Be sure to show all your experiment result (e.g., image processing steps and output images) here.   1. mean filter:   saturn\_mean=mean(saturn,neighborhood)  plt.figure(figsize=(12,12))  plt.subplot(1,4,1)  plt.imshow(saturn,cmap='gray')  plt.title("original")  plt.subplot(1,4,2)  plt.imshow(saturn\_mean+skimage.filters.laplace(saturn\_mean,ksize=3,mask=None),cmap='gray')  plt.title("one")  plt.subplot(1,4,3)  saturn\_mean=mean(saturn\_mean,neighborhood)  plt.imshow(saturn\_mean+skimage.filters.laplace(saturn\_mean,ksize=3,mask=None),cmap='gray')  plt.title("two")  plt.subplot(1,4,4)  saturn\_mean=mean(saturn\_mean,neighborhood)  plt.imshow(saturn\_mean+skimage.filters.laplace(saturn\_mean,ksize=3,mask=None) ,cmap='gray')  plt.title("three")  plt.show()     1. gaussian filter:   saturn\_gaussian=skimage.filters.gaussian(saturn)  plt.figure(figsize=(12,12))  plt.subplot(1,4,1)  plt.imshow(saturn,cmap='gray')  plt.title("original")  plt.subplot(1,4,2)  plt.imshow(saturn\_gaussian+skimage.filters.laplace(saturn\_gaussian,ksize=3,mask=None),cmap='gray')  plt.title("one")  plt.subplot(1,4,3)  saturn\_gaussian=skimage.filters.gaussian(saturn)  plt.imshow(saturn\_gaussian+skimage.filters.laplace(saturn\_gaussian,ksize=3,mask=None),cmap='gray')  plt.title("two")  plt.subplot(1,4,4)  saturn\_gaussian=skimage.filters.gaussian(saturn)  plt.imshow(saturn\_gaussian+skimage.filters.laplace(saturn\_gaussian,ksize=3,mask=None) ,cmap='gray')  plt.title("three")  plt.show()     1. total variation filter:   saturn\_tv=denoise\_tv\_chambolle(saturn)  plt.figure(figsize=(12,12))  plt.subplot(1,4,1)  plt.imshow(saturn,cmap='gray')  plt.title("original")  plt.subplot(1,4,2)  plt.imshow(saturn\_tv+skimage.filters.laplace(saturn\_tv,ksize=3,mask=None),cmap='gray')  plt.title("one")  plt.subplot(1,4,3)  saturn\_tv=denoise\_tv\_chambolle(saturn\_tv)  plt.imshow(saturn\_tv+skimage.filters.laplace(saturn\_tv,ksize=3,mask=None),cmap='gray')  plt.title("two")  plt.subplot(1,4,4)  saturn\_tv=denoise\_tv\_chambolle(saturn\_tv)  plt.imshow(saturn\_tv+skimage.filters.laplace(saturn\_tv,ksize=3,mask=None) ,cmap='gray')  plt.title("three")  plt.show()     1. bilateral filter:   saturn\_bil=denoise\_bilateral(saturn)  plt.figure(figsize=(12,12))  plt.subplot(1,4,1)  plt.imshow(saturn,cmap='gray')  plt.title("original")  plt.subplot(1,4,2)  plt.imshow(saturn\_bil,cmap='gray')  plt.title("one")  plt.subplot(1,4,3)  saturn\_bil=denoise\_bilateral(saturn\_bil)  plt.imshow(saturn\_bil,cmap='gray')  plt.title("two")  plt.subplot(1,4,4)  saturn\_bil=denoise\_bilateral(saturn\_bil)  plt.imshow(saturn\_bil,cmap='gray')  plt.title("three")  plt.show()     1. wavelet denoising filter:   saturn\_wavelet=denoise\_wavelet(saturn)  plt.figure(figsize=(12,12))  plt.subplot(1,4,1)  plt.imshow(saturn,cmap='gray')  plt.title("original")  plt.subplot(1,4,2)  plt.imshow(saturn\_wavelet,cmap='gray')  plt.title("one")  plt.subplot(1,4,3)  saturn\_wavelet=denoise\_wavelet(saturn\_wavelet)  plt.imshow(saturn\_wavelet,cmap='gray')  plt.title("two")  plt.subplot(1,4,4)  saturn\_wavelet=denoise\_wavelet(saturn\_wavelet)  plt.imshow(saturn\_wavelet,cmap='gray')  plt.title("three")  plt.show() |  |

1. **Properties of Convolution (20 points)**

|  |  |  |
| --- | --- | --- |
| # | Description | Score |
| 1 |  |  |
| 2 |  |  |

1. **Image Segmentation and Color Space (20 points)**

|  |  |  |
| --- | --- | --- |
| # | Description | Score |
| - | Paste your result here and briefly describe your image processing procedure and approach. How do you think your results are?  Result:    Procedure:   1. 先大概抓取圖片中要處理的範圍，x大概為(200,520)、y的範圍大概為(100,430)，範圍外的值都給黑色。 2. 將原圖轉換成hsv後，依序對上述的範圍進行h、v、s的過濾處理   for h in range(100,430):      for w in range(200,520):          if fan\_hsv[h,w,0]>0.15 and fan\_hsv[h,w,0]<0.18:              if fan\_hsv[h,w,2]>0.6 and fan\_hsv[h,w,2]<=1:                   if fan\_hsv[h,w,1]>0.65 and fan\_hsv[h,w,1]<=1:                      img[h,w]=1    (hsv 處理後的圖)   1. 最後再用median 的方式處理   neighborhood=square(width=5)  img=median(img,neighborhood)  最後的效果還不錯，但一些尖尖角角的部分並不是處理的那麼好 |  |

1. **Document Scanner (20 points)**

|  |  |  |
| --- | --- | --- |
| # | Description | Score |
| - | Paste your result here and briefly describe your image processing procedure and approach. How do you think your results are?  Result:    Procedure:   1. 現在原圖抓取大概的範圍候用transform.ProjectiveTransform()的方式將掃描後的尺寸決定好。   src=np.array([[0,0],[0,680],[350,680],[350,0]])  dst=np.array([[125,185], [65,525], [300,530],[267,180] ])  tform3 = transform.ProjectiveTransform()  tform3.estimate(src, dst)  warped = transform.warp(invoice, tform3, output\_shape=(680, 350))   1. 我想把背景中的陰影去掉，因此想利用上課教的方式去實作，但由於背景是白色，與一般的背景黑色不同，因此我先把圖的黑白對調   warped=rgb2gray(warped)  warped\_reverse=1-warped     1. 再利用課堂上的方式去除背景雜訊   bg=morphology.erosion(warped\_reverse,morphology.square(3))  bg=filters.gaussian(bg, sigma=7)  warped\_reverse=warped\_reverse-bg   1. 最後再用local thresh 取抓值，但因為此時的圖與最後的索求是黑白顛倒的，因此最後一步是要用小於而不是大於   local\_thresh=filters.threshold\_local(warped\_reverse,block\_size=255,offset=-10/255)  warped\_final=warped\_reverse<local\_thresh  我的結果可以使字大略看得懂，但與原圖相比，字會有點變形，但至少背景沒有黑黑的雜訊出現。 |  |