**1132 Digital Image Processing Assignment #3 書面報告**

學號：1110927 姓名：陳柏翰

**主題：影像除霧處理 Image Dehazing**

利用在影像處理概論Chapter 6 Image Restoration 所學習的知識與技術，撰寫一個程式去除所附6張圖片中的霧霾。

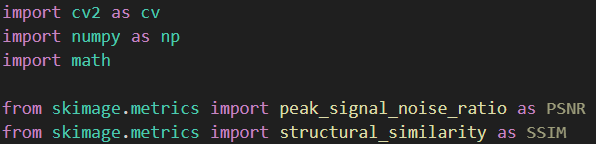
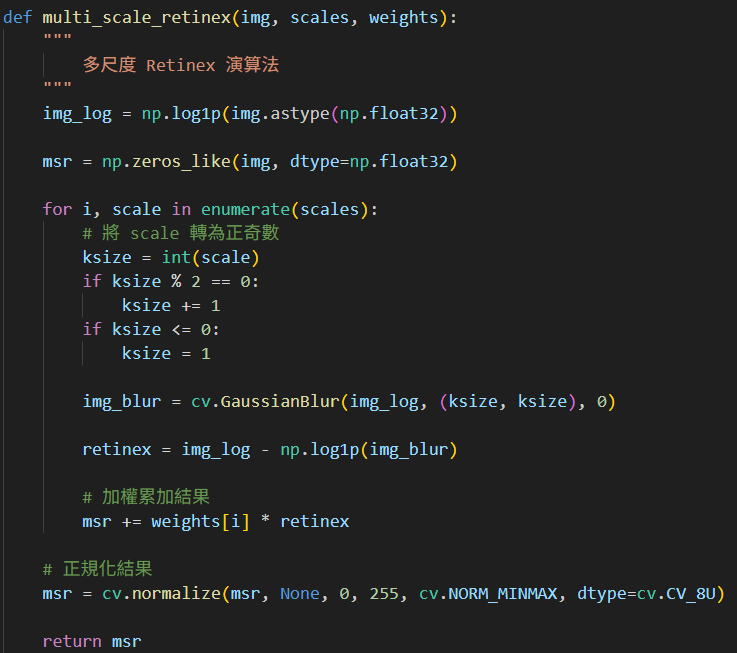
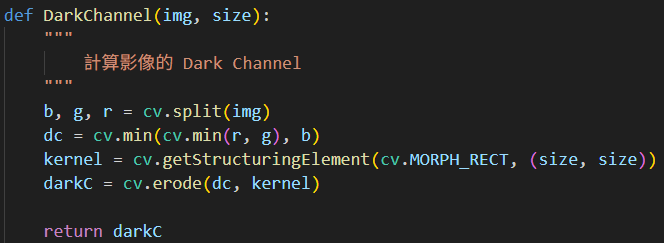
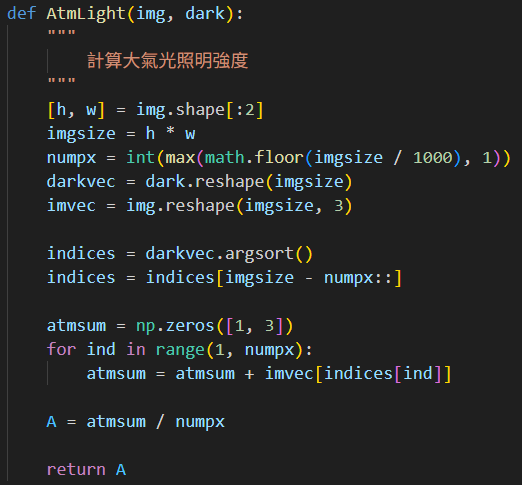
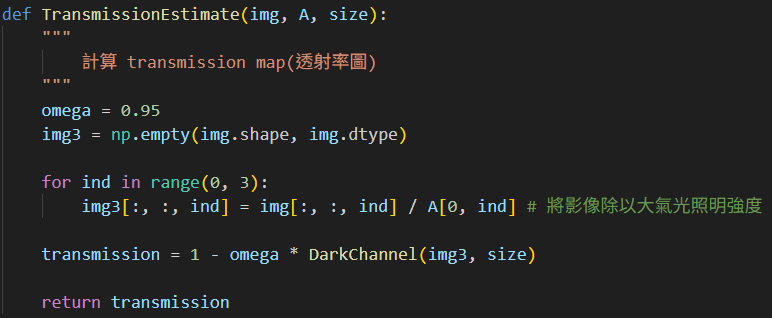
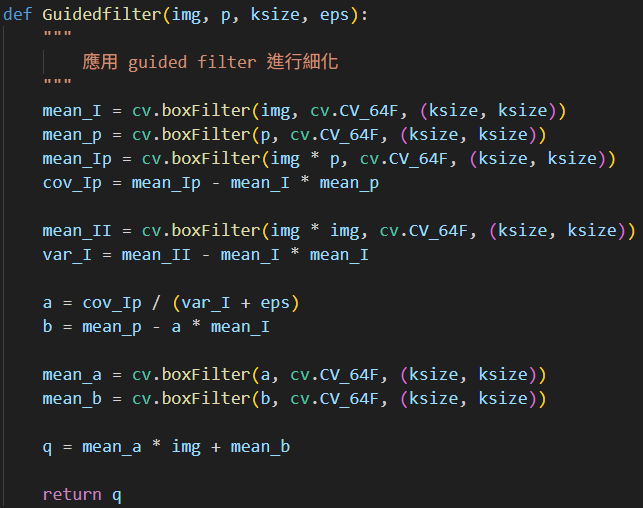
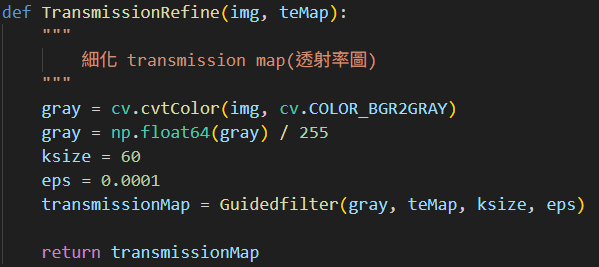
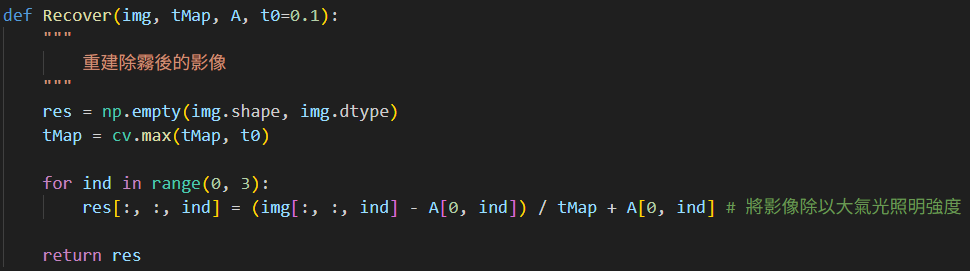
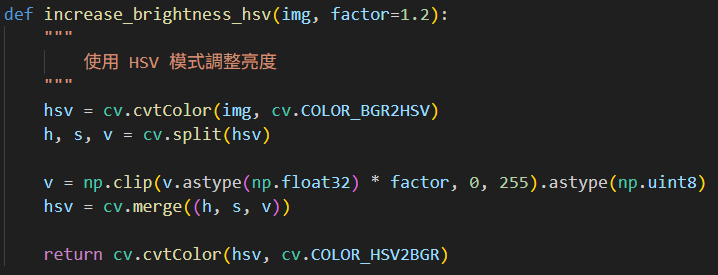
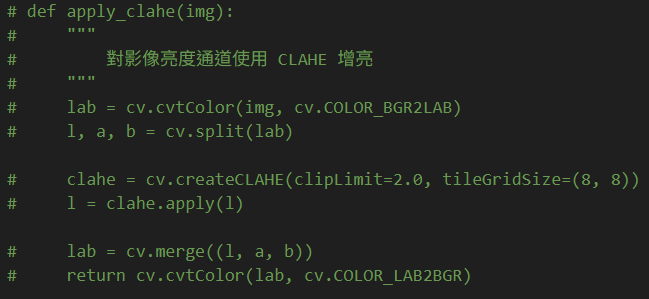
使用附件中的6張照片及標準答案圖片(Ground Truth)做測試，顯示改善後較清晰圖片並計算處理過後的圖片與標準答案圖片(Ground Truth)的 PSNR 與SSIM值。

以作業所附之hazy01.jpg與其執行改善結果為例：

開發環境：Microsoft Windows 11, Visual Studio Code, OpenCV 4.11.0, NumPy 2.2.5, Python 3.10.6, Scikit-image 0.25.2

**演算法與程式碼說明：**

1. 導入函式庫  
   
2. multi\_scale\_retinex  
   實作多尺度Retinex演算法，提升影像動態範圍與細節對比。  
   
3. DarkChannel  
   計算影像的暗通道，用於後續大氣光與透射率估計。  
   
4. AtmLight  
   根據暗通道選出最暗像素，估算場景中的大氣光強度。  
   
5. TransmissionEstimate  
   根據大氣光與暗通道計算影像的透射率圖。  
   
6. Guidedfilter  
   以原始灰階影像為引導，平滑並細化透射率圖。  
   
7. TransmissionRefine  
   將引導濾波應用於透射率圖，提升邊緣保留效果。  
   
8. Recover  
   根據透射率與大氣光重建無霧影像內容。  
   
9. white\_balance  
   對影像進行白平衡處理，提升色彩準確性。  
   
10. increase\_brightness\_hsv  
    轉換至 HSV 模式後調整亮度值，提昇整體亮度。  
    
11. 遺珠之憾  
    在最終過程中，針對色域偏差以及亮度調整有嘗試一些操作，下述函式可以在某些圖片的處理中可以保持紋理並可以讓模糊的部分產生更清楚的辨識度。  
    

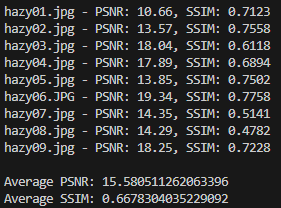
**執行成果：**

** **

** **

** **

** **

**   **

**心得與討論：**

本次作業非常具有挑戰性，作業目標不只需要專注在除霧本身，還原後的影像色彩是否失真以及確保不同程度的霧化還原亦是至重要的一環，在思考如何更好的加強既有的方法以及替換既有方法之間面對了許多複雜的技術細節。

我接觸到了許多極具價值的改進方向，包括自適應窗口大小的 Dark Channel、結合梯度資訊的透射率估計、使用雙邊濾波器細化透射率、白平衡、更精準的大氣光估計方法 (例如區域分割、迭代優化和 DWARP)，以及色彩標準化等等。

然而，大多數的方法並沒有取得更好的結果，亦或是其計算成本與其改進程度並不成正比。

最後，在參考A Real-time Optimization System for Image De-fogging Based on the Retinex Algorithm Using Fuzzy Theory(Li, 2013)論文中，基於多尺度Retinex(MSR) 演算法和模糊理論的影像除霧方法，在應用原有的方法之前，增強影像的細節和色彩，並減少霧霾的影響，可以改善Dark Channel 的估計，從而影響後續的透射率估計和最終的除霧結果。

基於這個方法的成果，我也初步整理了後續的改進方向：

* 整體程式架構的調整。
* 參考論文中進一步的方法，使用影像積分來加速均值濾波的計算。作者提出，這項工作可以顯著提高 MSR 的效能。
* 進一步研究更有效率的濾波器實作，或者嘗試使用其他近似濾波方法。
* 如果時間允許，可以進一步實作論文 中提出的模糊理論方法，以自動調整 MSR 的參數。

Google Site 網址：