

1. 建立一個 Conda 環境
  - conda 環境名稱為 1132\_PY\_Pillow
  - 偶數學號同學 Python 版本 3.12.x
  - 奇數學號同學 Python 版本 3.11.x ``
2. 安裝 pillow 套件 (查看套件版本並將版本資訊加入 environment.yml 檔中)
3. 建一個專案目錄，在目錄下要包含下列子目錄及檔案
  - imgs 目錄 (裡面放網路上找到 jpeg 檔案)
  - results 目錄 (裡面存放程式處理影像後的影像成果)
  - src 目錄 (裡面存放所有本練習題用到的程式 py 檔)
  - environment.yml 檔案
  - setup\_env.ps1 檔案
  - setup\_env.sh 檔案
4. 任務 1: 使用批次功能在執行程式批次讀入在 imgs 目錄下的全部 jpeg 檔轉換 png 檔格式後放到 results 目錄下
  - 在終端機下指令 `python test.py -jpg2png` 完成 jpeg 轉 png 任務
  - 在終端機下指令 `python test.py -png2jpg` 完成 png 轉 jpeg 任務
5. 任務 2: 建立一個 imtool.py 程式完成下列功能 (請利用函數呼叫形式撰寫)
  - 在終端機下指令 `python imtool.py testing.jpg -resize 0.5` 將放在 imgs 目錄下的 testing.jpg 檔放大 0.5 倍後放到 results 目錄下變成是 testing\_resized.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -VFlip` 將放在 imgs 目錄下的 testing.jpg 檔垂直方向翻轉後放到 results 目錄下變成是 testing\_VFlip.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -HFlip` 將放在 imgs 目錄下的 testing.jpg 檔水平方向翻轉後放到 results 目錄下變成是 testing\_HFlip.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -R90` 將放在 imgs 目錄下的 testing.jpg 檔旋轉90度後放到 results 目錄下變成是 testing\_R90.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -R180` 將放在 imgs 目錄下的 testing.jpg 檔旋轉180度後放到 results 目錄下變成是 testing\_R180.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -R270` 將放在 imgs 目錄下的 testing.jpg 檔旋轉270度後放到 results 目錄下變成是 testing\_R270.jpg 檔
  - 在終端機下指令 `python imtool.py testing.jpg -thumbnail 200 150` 將放在 imgs 目錄下的 testing.jpg 檔產出寬 200 高 150 的 testing\_thumbnail.jpg 檔 並放到 results 目錄下
  - 在終端機下指令 `python imtool.py testing.jpg -BLUR` 將放在 imgs 目錄下的 testing.jpg 檔套用 模糊 (BLUR) 濾鏡後產出 testing\_BLUR.jpg 檔 並放到 results 目錄下
  - 在終端機下指令 `python imtool.py testing.jpg -CONTOUR` 將放在 imgs 目錄下的 testing.jpg 檔套用 輪廓 (CONTOUR) 濾鏡後產出 testing\_CONTOUR.jpg 檔 並放到 results 目錄下
  - 在終端機下指令 `python imtool.py testing.jpg -DETAIL` 將放在 imgs 目錄下的 testing.jpg 檔套用 細節增強 (DETAIL) 濾鏡後產出 testing\_DETAIL.jpg 檔 並放到 results 目錄下
  - 在終端機下指令 `python imtool.py testing.jpg -EDGE_ENHANCE` 將放在 imgs 目錄下的 testing.jpg 檔套用 邊緣增強 (EDGE\_ENHANCE) 濾鏡後產出 testing\_EDGE\_ENHANCE.jpg 檔 並放到 results 目錄下
  - 在終端機下指令 `python imtool.py testing.jpg -EDGE_ENHANCE_MORE` 將放在 imgs 目錄下的 testing.jpg 檔套用 深度邊緣增強 (EDGE\_ENHANCE\_MORE) 濾鏡後產出

`testing_EDGE_ENHANCE_MORE.jpg` 檔並放到 `results` 目錄下

- 在終端機下指令 `python imtool.py testing.jpg -EMBOSS` 將放在 `imgs` 目錄下的 `testing.jpg` 檔套用 浮雕效果 (EMBOSS) 濾鏡後產出 `testing_EMBOSS.jpg` 檔並放到 `results` 目錄下
- 在終端機下指令 `python imtool.py testing.jpg -SHARPEN` 將放在 `imgs` 目錄下的 `testing.jpg` 檔套用 銳利化效果 (SHARPEN) 濾鏡後產出 `testing_SHARPEN.jpg` 檔並放到 `results` 目錄下
- 在終端機下指令 `python imtool.py testing.jpg -SMOOTH` 將放在 `imgs` 目錄下的 `testing.jpg` 檔套用 平滑效果 (SMOOTH) 濾鏡後產出 `testing_SMOOTH.jpg` 檔並放到 `results` 目錄下
- 在終端機下指令 `python imtool.py testing.jpg -SMOOTH_MORE` 將放在 `imgs` 目錄下的 `testing.jpg` 檔套用 深度平滑效果 (SMOOTH\_MORE) 濾鏡後產出 `testing_SMOOTH_MORE.jpg` 檔並放到 `results` 目錄下
- 在終端機下指令 `python imtool.py testing.jpg -FIND_EDGES` 將放在 `imgs` 目錄下的 `testing.jpg` 檔套用 邊緣訊息 (FIND\_EDGES) 濾鏡後產出 `testing_FIND_EDGES.jpg` 檔並放到 `results` 目錄下