

#資料集製作

1. create_iris.py
 - 一次生成多組資料集的 iris 數據集，搭配 rgb_try2.py 使用
2. rgb_try2.py
 - 自製每組影像序列的 iris 資料集
3. predict_rgb.py
 - 製作結果偵測影像的 iris 數據及格式，並利用訓練好的分類權重檔，計算各成熟度類別顆數
4. Ripness_data.py
 - 番茄成熟度標記資料
 - 19th_2 tomato 資料集要依照「一週一次」或「一週兩次」決定要用那些 ripness 矩陣
5. Sequoia_mot.py
 - 將追蹤資料整理成 mot 資料集，並生成 gt.txt
 - 需先將 rgb 照片檔和 labels_with_ids 檔放入目標資料夾
6. Excel_merge.py
 - 合併同一組資料集的所有訓練集(或測試集)的 csv 檔，形成 merge.csv 檔，供各分類訓練程式使用
7. File.py
 - 將拍攝好的資料集整理成如論文所提的格式

#成熟度分類

主要修改變數變數

train_data:訓練集的 csv 檔

test_data:測試集的 csv 檔

versus_component:分類指標

1. Svm.py
 - Svm 分類
 - 可生成分類模糊矩陣和各分類指標貢獻度
2. Knn.py
 - Knn 分類
 - 可生成分類模糊矩陣和訓練集的主成分分析(pca)結果
3. Ann.py
 - 人工神經網路分類

#物件追蹤

1. Mot_evaluate.py
 - 評估物件追蹤的效果

2. Yolo_tracking_master 資料夾

- 使用當中的 track.py 進行物件追蹤，有 strongsort, ocsort 和 bytetrack 三種
- 本論文訓練的追蹤權重檔放在 weights 資料夾內
- 路徑轉至 ultralytics-15b3b0365ab2f12993a58985f3cb7f2137409a0c 資料夾內，可訓練 yolo 權重檔

#函式檔(不用更改的)

1. Excel_clear.py

- 清除 iris 數據集中有空白無法計算(例如某番茄框全被 mask 掉)的部分

2. warp.py

- Image_warpping，但如果之後拍攝距離有變，要改變裡面座標的位置

3. Otsu.py

- 利用 OTSU's threshold 去除 yuv 影像中非番茄部分

4. Ellipse.py

- 以番茄框的長寬為橢圓形的長短軸，將影像切成橢圓形

5. Voting.py

- Voting 程式碼，讓同一個番茄 id 的成熟度一致

#新增資料夾

- 裡面有沒用到的程式碼