112-2 機器視覺期末專題報告

主題: SOP檢測

參與者:1092B0026 楊孜弘

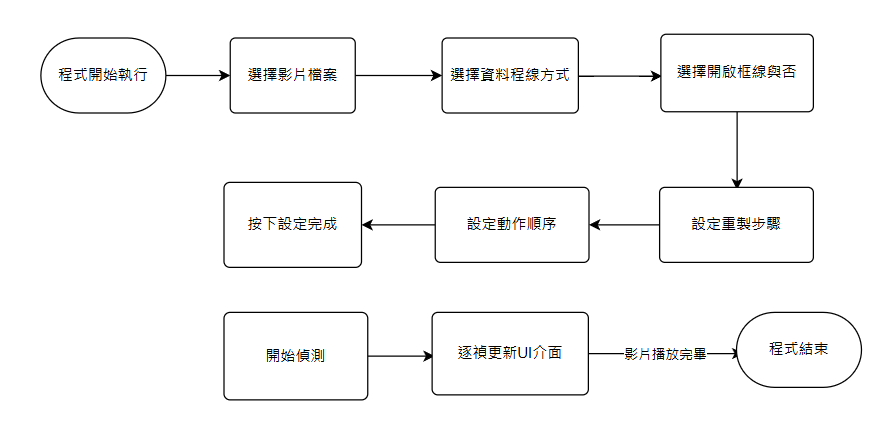
       1092B0030 李沐風

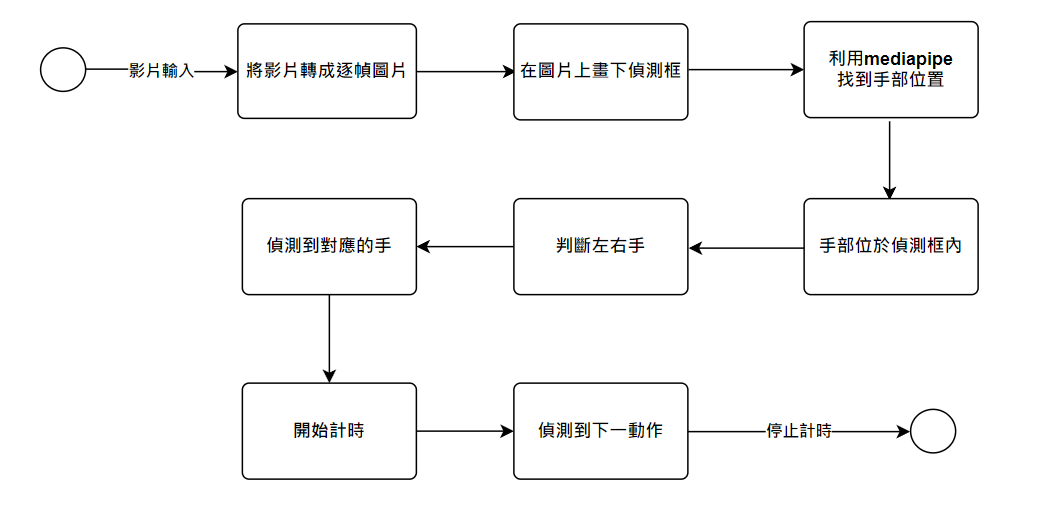
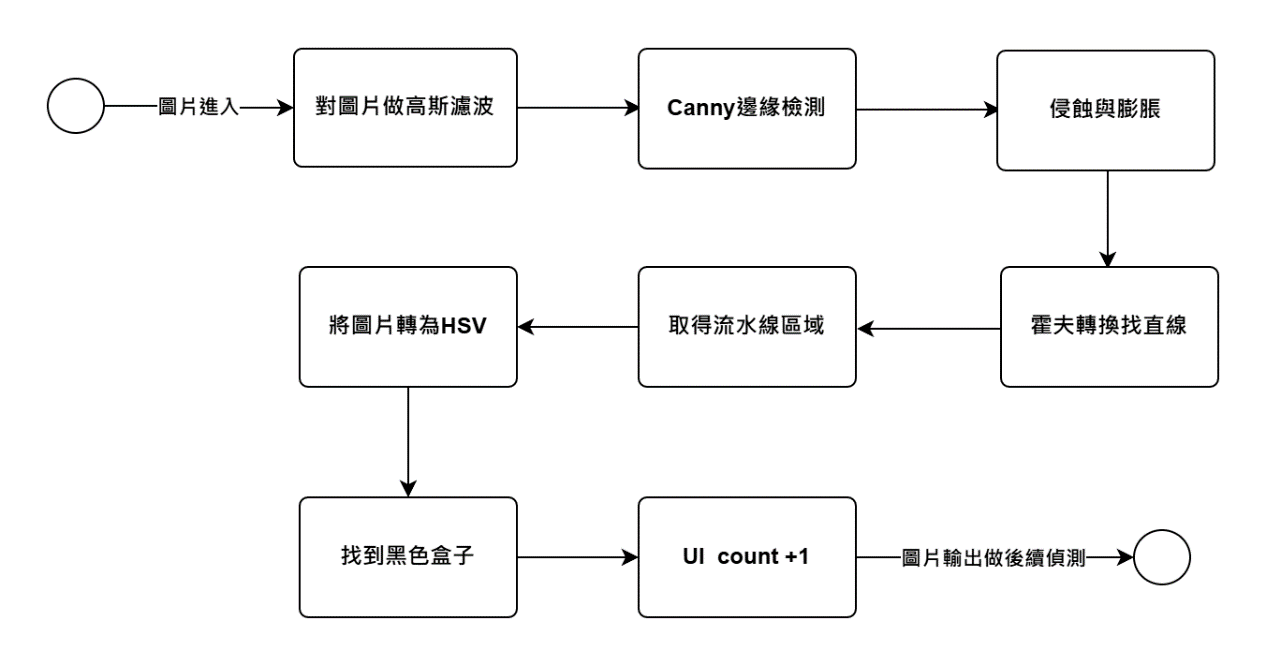
時間:2024/06/23

一、專題摘要

本次專題我們組所做的是「流水線SOP檢測」，其目的是希望以影像處理的方式達成自動化偵測的功能，檢測員工的組裝流程有無錯誤，查看員工的效率，以及檢測在長時間下員工的產品完成率是否達到一定標準。而我們寫了一個UI介面給使用者，讓使用者可以查看工人做的動作、秒數、以及完成的次數。

1. 流程圖

* 1. UI 介面使用

1. SOP動作偵測
2. 流水線完成品偵測
3. 程式解說

匯入必要的程式庫：

**MediaPipe：**我們使用此套件的手部追踪功能，即時識別視頻中的手部位置。

**OpenCV：**我們使用此套件來讀取和處理影像，實施顏色分割和邊緣檢測等影像處理技術，以識別和追踪流水線上的物品。

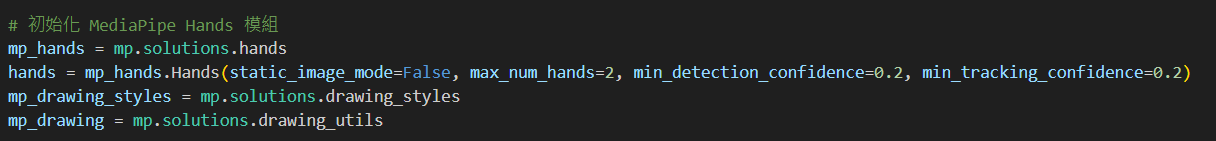
**CustomTkinter**：這是一個用於建立GUI的套件，我們使用此套件來建造UI給使用者客製化偵測過程並查看結果

主要功能實現：

**初始化影片：**使用者一開始設定好影片，並轉成逐幀圖片，嵌入至UI，與其結合

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

**手部偵測與分析：**我們使用MediaPipe追踪視頻中的手部，並分析其動作(放螺絲、鎖螺絲、丟到流水線)是否符合預設的標準操作程序(SOP)。而我們在此處初始化mediapipe模組，並將追踪器同時追踪的最大手數為2。而為了讓手能更好的被追蹤，我們將信心閥值調整成0.2

將每個動作與執行秒數變成一個相對應的字典，用於後續UI查看動作秒數使用

再初始化steps陣列，用於SOP邏輯順序判斷一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

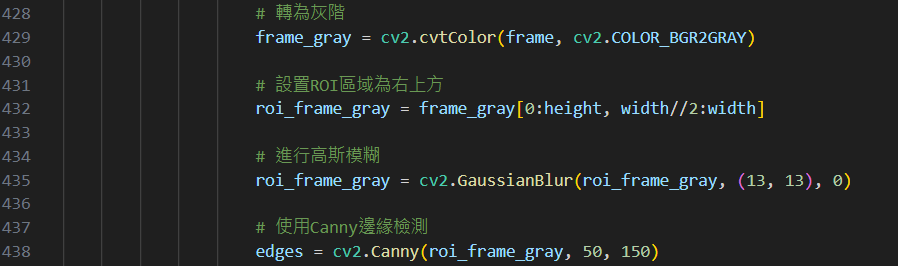
一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述在圖片上畫多個偵測框，對應不同動作

當左手的節點在放螺絲偵測框內，並且上一個動作不為鎖螺絲(丟到流水線)，便更改當前動作、開始計時，再將資訊傳送到UI

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

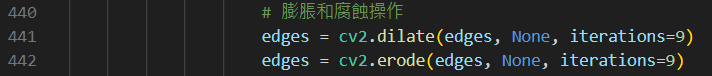
自動產生的描述

**流水線計數：**圖片進入後會先將流水線的位置定義在畫面的右上方，之後對其做高斯濾波，因為在每一幀的圖片中，中間會有一盒螺絲。然而因為光的關係，不做濾波的話，之後的Canny邊緣檢測會在螺絲盒中檢測出大量的邊緣，這會導致後續在做霍夫轉換時因為有大量的雜訊造成效果不理想，而高斯濾波可以大幅度地降低Canny偵測出這些雜訊的機率。

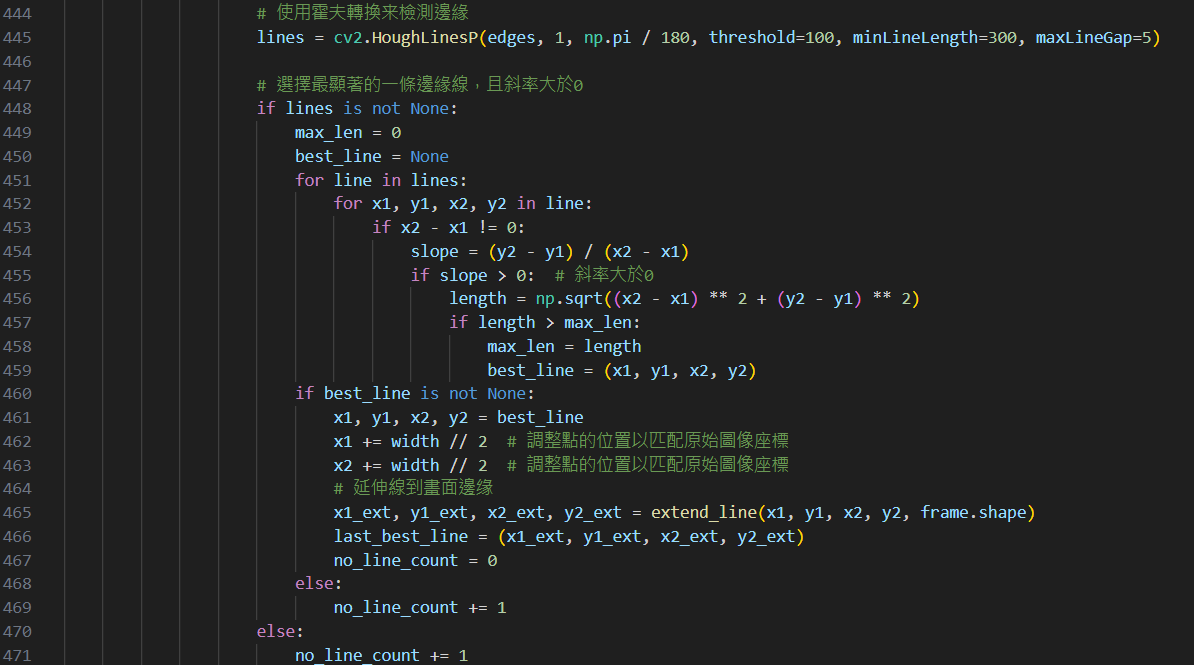
做完高斯濾波後我們就做邊緣檢測。然後再對圖片進行侵蝕與膨脹，原因包含以下幾點。

(1)邊緣檢測後，圖片中可能存在許多孤立的噪聲點。這些噪聲點會干擾後續的邊緣分析。通過侵蝕操作，可以有效地去除這些噪聲點，保留主要的邊緣結構。

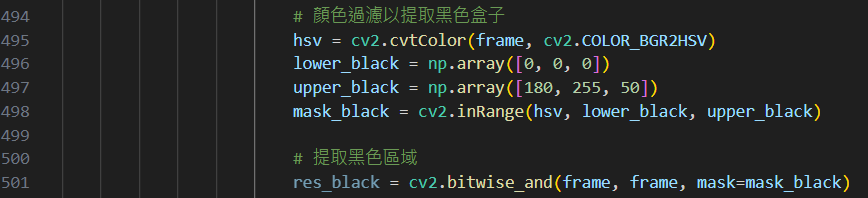
(2)侵蝕操作可能會導致邊緣變得更細甚至斷裂，這時候使用膨脹操作可以幫助填補這些斷裂的邊緣，使邊緣恢復連續性。



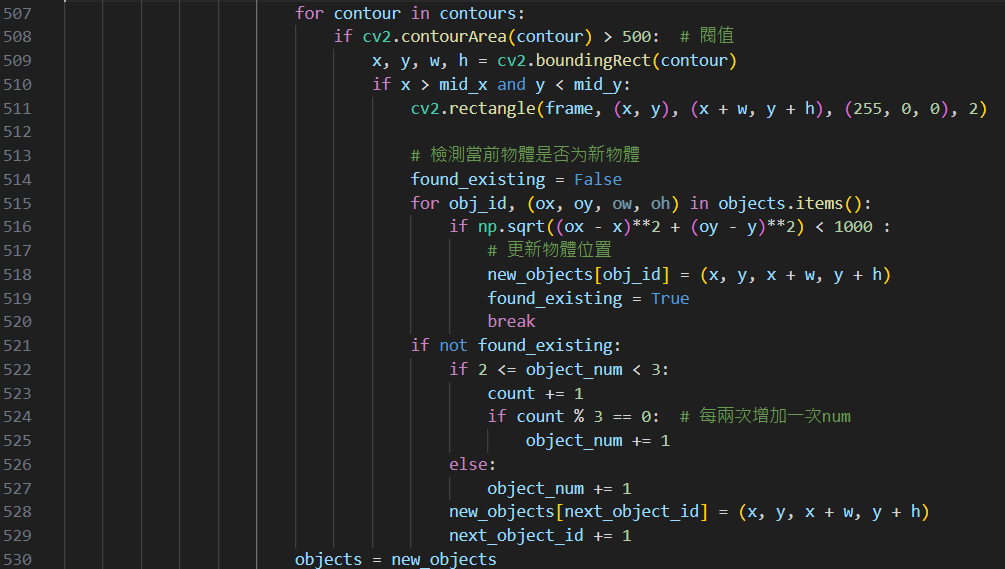
侵蝕與膨脹後，會用先前提到的邊緣來做霍夫轉換找到流水線邊緣的直線，而此直線必須要斜率小於0，有了這條直線，就能夠取得履帶的上半部分作為我們感興趣的區域。



最後將圖片轉成HSV，並針對產品的顏色用遮罩提取出來。



提取黑色物體後，判斷是否為新的物體，若是則UI介面計數加一，若否則不做更動。



**3. UI介面：**

一張含有 文字, 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

自動產生的描述計算與呈現：UI介面中包含四個視窗，第一個視窗用於讓使用者上傳影片，並根據需求開啟獲關閉偵測框線。而左側的部分會顯示使用者輸入的影片，可以查看影片進度，並且包含暫停功能。

下圖為第一個視窗展示效果

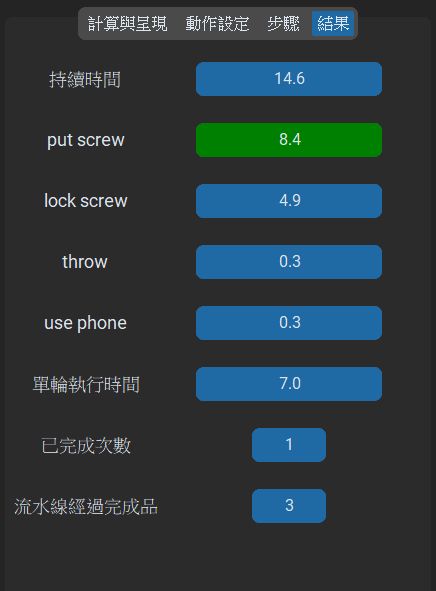
**動作設定(左圖)**：

這裡主要在設定哪一個步驟要設為重製步驟，也就是做完整組動作的判斷依據。

**步驟(中間圖)**：使用者可以在此頁調整執行的步驟，用於客製化偵測過程。

**結果(右圖)**：

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體, 設計 的圖片

自動產生的描述我們會用綠色來標示正在執行的步驟，並顯示整組動作花費的時間(持續時間)，各個步驟單獨花費的時間(單輪執行時間)，工人已完成幾組動作(已完成次數)，最後顯示流水線上經過的完成品數量。

1. 實際展示和主要功能
2. 顯示正在執行的步驟
3. 顯示各個步驟**單獨花費的時間**
4. 顯示**整組動作**花費的時間
5. 顯示已完成**幾組動作**
6. 顯示流水線經過的**完成品數量**

成果影片路徑: <https://youtu.be/95WuU860fBs>

成果介紹簡報: [SOP偵測 - 簡報 (canva.com)](https://www.canva.com/design/DAGI3KgHrCQ/aJAL5D_JytPPj2p-CZSxFA/edit)

1. 應用層面

本次專題針對目標為相較於傳統的代工廠、製造廠，或是因為技術、資金等原因還未完成高度自動化的流水線，而仍需大量人力在作業台前「手做」；因為是人為就代表著會受個人狀態影響，可能是狀態不佳導致效率變差，可能是分神而導致流程出錯，而我們所設計的這個簡易的程式正是為了阻止這個情況的發生；對業主而言可以了解各個工作員一段時間內的表現差異，對受雇者而言可以為程式賦予警報系統，當有長時間流程錯誤情況時，可以即時的提醒受雇者，減少不必要的風險發生。

1. 專題貢獻
2. 完整的UI架構

此UI介面可適用於各個產線檢測，僅需改變輸入影片檔以及修改動作偵測演算法即可適用於各種產線檢測。

1. 一套可行的檢測步驟

透過實際從網路上找到的現實產線影片，並運用本學期所學的影像處理技巧以及網路上的資料，完成一整套作業流程檢測系統，以證明此想法的可行性。

1. 總結

這個專題應用MediaPipe、OpenCV、以及customtkinter等工具來自動化偵測流水線上的SOP和產品，透過手部動作追踪及圖像分割技術，精確檢測員工操作是否符合標準。我們也創建了直觀的UI介面，方便用戶即時查看工作進度和效率。針對環境噪聲，我們也透過高斯濾波等影像預處理技術有效提升了檢測的準確性。成功達成流水線計數的功能。

1. 未來展望
   1. 結合深度學習

現在的版本實作過程中我們有發現三個問題分別是:

(1) 鏡頭角度的輕微改變都會造成判斷準確率降低

(2) 純影像辨識程式過於冗長不易閱讀

以上三點問題，我們覺得以深度學習來辨識可以改善；針對第(1)點問題，只要訓練模型使用正確便不會有用因為鏡頭位置改變，造成偵測框位置偏移而誤判的問題；針對第(2)點只要我們使用yolo模型便可以便可以簡單的辨識出動作，後續只要做邏輯判斷即可。

* 1. 擴增應用功能

目前的功能主要只集中在偵測動作順序、時間及完成品的計數，而這個專題可擴展性很大可以進一步提升，例如:

功能擴展:

* 更多動作檢測：增加對更多類型動作的檢測，擴大系統的應用範圍
* 多攝像頭支持：整合多攝像頭，提高整體系統的穩定性及檢測範圍。

應用推廣:

* 工業4.0：將系統整合至智能工廠，實現生產全過程的智能自動化。
* 跨領域應用：探索在其他領域（如醫療、物流等）的應用可能性，推 廣技術應用範圍。

九、參考資料

1. 【Bilibili. (2021, December 19). 我在流水线打螺丝. Bilibili. 】<https://www.bilibili.com/video/BV1S44y1E7HN/?spm_id_from=333.337.search-card.all.click&vd_source=11f336e7c7c8284bc6fe491c1e39c4e6>

1. 【Schimansky, T. (n.d.). CustomTkinter documentation. Retrieved June 24, 2024, from 】<https://customtkinter.tomschimansky.com/documentation/>