

National Sun Yat-sen University
Department of Electrical Engineering

112-2 機器學習系統設計實務與應用

期末報告

組別：你說的隊

組長：B103012034 黃麗穎

組員：B103012041 黃詣辰

B103015006 胡庭翊

B103015001 林佳明

組員	黃麗穎	黃詣辰	胡庭翊	林佳明
工作比例	25%	25%	25%	25%
簽名				
工作分配	硬體部署 資料蒐集&標籤 書面報告	硬體部署 資料蒐集&標籤 書面報告	訓練模型 資料蒐集&標籤 書面報告	訓練模型 資料蒐集&標籤 書面報告

一、專題主題

- 實作主題：電工實驗小助手

上了四學期的電工實驗，我們時常遇到麵包板上元件數值接錯的情況以及將器材分類收回時常有放錯袋子的情形，因此我們希望藉由本專案加速檢查麵包板上元件的正確性以及在整理材料的時候幫助快速分類。除了辨識元件以外我們還增加了預輸入元件個數和警告出現非預期元件的功能。

- 國內外作品

1. AOI 視覺影像技術於電路板元件檢測之應用：

- (1)技術差異：

文章	<u>影像處理技術</u> ：灰階、二值化、邊緣偵測、細線化、最大區塊等外型處理 <u>定位</u> ：計算傾斜角度自動定位 <u>辨識</u> ：平均值和變異特徵配合歐幾里得距離，手動選擇與提取特徵
本專案	專用於目標檢測的深度學習模型 YOLOv8 學習大量影像數據，能夠直接從影像中辨識及框出目標物

- (2)應用差異：

文章	PCB 板上電阻、電容、二極體缺件、錯件和急性相反的檢測
本專案	麵包板上的電阻、電容辨識，能夠設定目標電路個元件個數並對偵測出不應該存在的元件提出警告。可以處理元件隨意擺放的情況並同步顯示偵測到各元件的數量

因為利用深度學習模型能夠達到更高的準確率並且可以識別不同類型的元件、應用範圍更廣，所以本專題採用 YOLOv8 模型的方法。

2. Detection of circuit components on hand-drawn circuit images by using faster R-CNN method

- (1)技術差異：

文章	Faster R-CNN：兩階段物件偵測，RPN 網路、分類層、交替訓練、Anchor 框、Anchor 框計算
本專案	YOLOv8:單階段檢測，物體分類及定位一起完成

(2)應用差異:

文章	針對手繪電路圖的元件檢測與分類
本專案	針對麵包板上電阻及電容的檢測，允許元件任意擺放並且將檢測出的元件數量同步到螢幕上，如果檢測到不應存在的電阻或電容則會跳出警告。

本專案針對實際的電路進行檢測相較此篇文章對手繪電路元件進行檢測較具實用性，並且因為模型結構簡單、計算效率高，所以較適合嵌入式系統。

二、功能與使用流程

● 系統應用場景

1.電工實驗(電路實作)時：接完麵包板之後，我們會先計算講義上電路圖上的各元件個數並且將其輸入，接著就可以開始偵測元件並讓他判斷是否有異常出現的電阻電容值。當我們看到跳出警告，便可以從那裏開始偵錯，減少從頭開始 debug 電路的時間。

2.元件分類時：做完電工實驗後，我們會把所有元件從麵包板上拆除放置在桌上，此時便可以利用此系統一次性辨識所有元件，幫助我們將相同元件分類並歸位。

● 系統功能

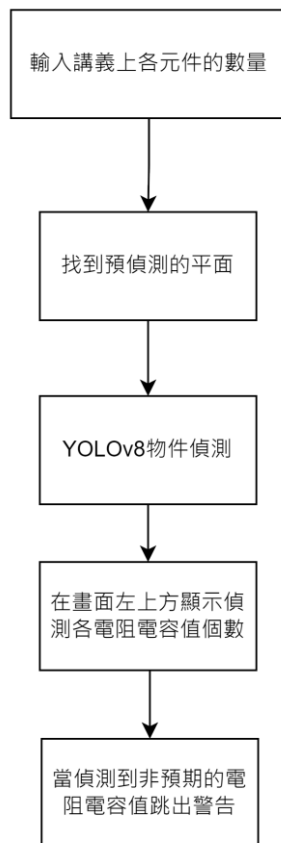
1.用於電工實驗時，或者自行在家裡練習接電路時的 debug，幫助我們快速發現電路接錯的部分，並且即時做出修正。

2.拿實驗器材時常常發現有人將元件放錯袋子，有了這個系統，可以幫助我們快速地進行電阻、電容的分類。

● 系統使用流程

1.說明：訓練完模型後，找到欲偵測的電路板或者電阻、電容的袋子，開啟鏡頭，開始 AI 物件偵測，顯示出電阻、電容值後，便可以針對錯誤的地方即時做出修正。

2.簡易流程圖

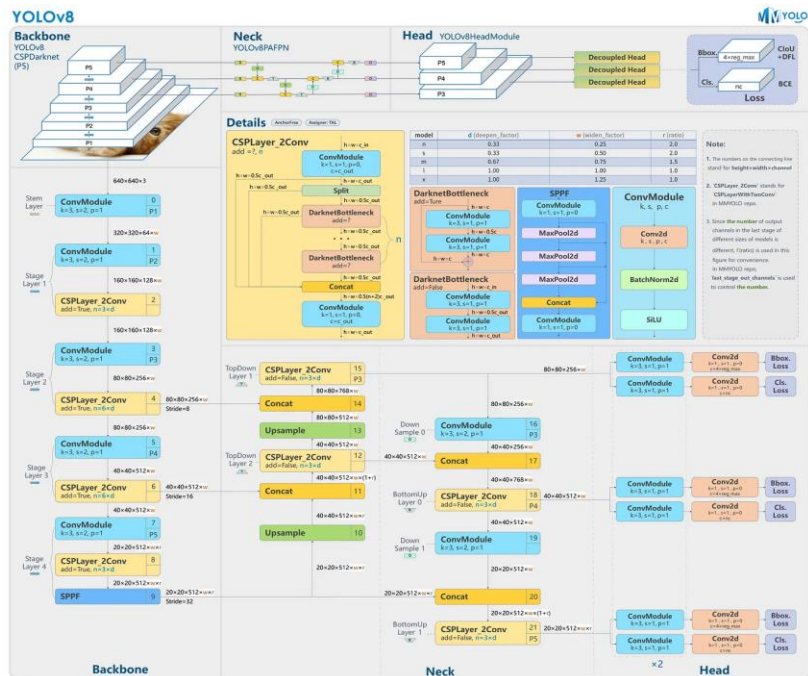


三、模型介紹

- 使用模型：YOLOv8(n)

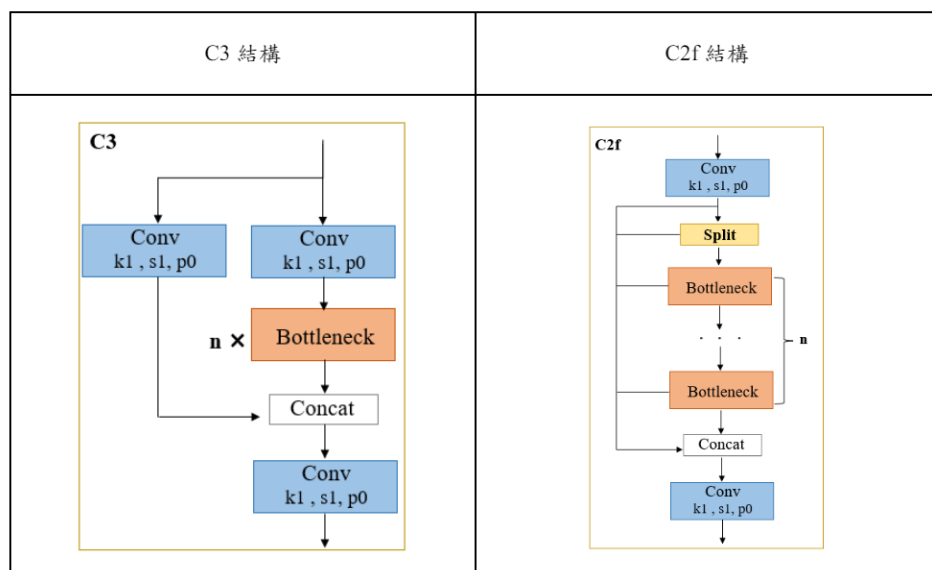
介紹：YOLOv8 是一種物件偵測模型，針對前幾代 YOLO 模型的優點進行保留，缺點進行改進。“n” 在 YOLOv8(n) 中代表 "nano"，表示這是一個較小型、輕量級的版本，用意是為了提升速度並在計算資源有限的情況下也能準確地預測。我們會使用此模型就是因為考量到要在 jetson-nano 上實現專題結果，所以選了這個較輕量級的 YOLO 模型。

1.架構說明

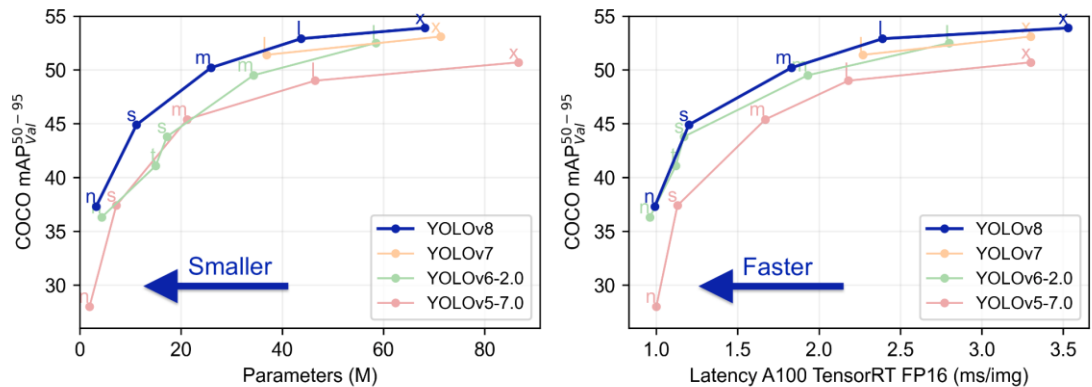


(1)Head 的部分改成了 Decoupled-Head，接分類與檢測頭分離，並解將 Anchor-Based 換成了 Anchor-Free(無錨檢測)，Anchor-Free 的優點是不需要預設 anchor，只需要對不同尺度的特征圖的目標中心點和寬高進行回歸，減少了耗時和算力。同時，Anchor-Free 的方法也可以避免一些由於 anchor 設置不合理導致的漏檢或重覆檢測問題。

(2)Backbone 和 Neck 參考了 YOLOv7 的設計，使用新的卷積層架構 C2f，並對於不同尺度的模型調整了不同的通道數，大幅提升了模型的效能。C2f 進一步增加跳層連結。C2f 模塊的設計受到 C3 模塊和 ELAN 模塊的啟發，以確保 YOLOv8 在保持輕量化的同時，獲取更加豐富的梯度流信息，C2f 具有更少的參數和更好的特徵提取。



2. 與其他 yolo 版本模型比較-優缺點

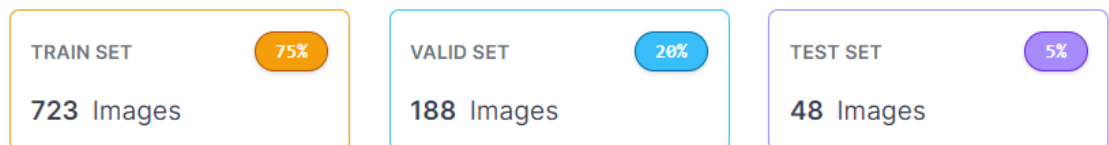


可以發現 YOLOv8 在減少參數的同時，進一步提高了準確度以及速度，而且 YOLOv8 對於偵測小物件，有較優異的準確度，適合偵測我們的主題小物件的電阻。這主要是我們選擇此模型的原因，但使用此模型最主要的困難處是部署在 jetson nano 上，由於 YOLOv8 為 2023 年的發表的，所需的 python 版本較高，與 jetson nano 環境不符合，這部份我們使用虛擬環境以及安裝許多套件後，才成功實現在 jetson nano，但偵測速度與電腦比大幅下降。

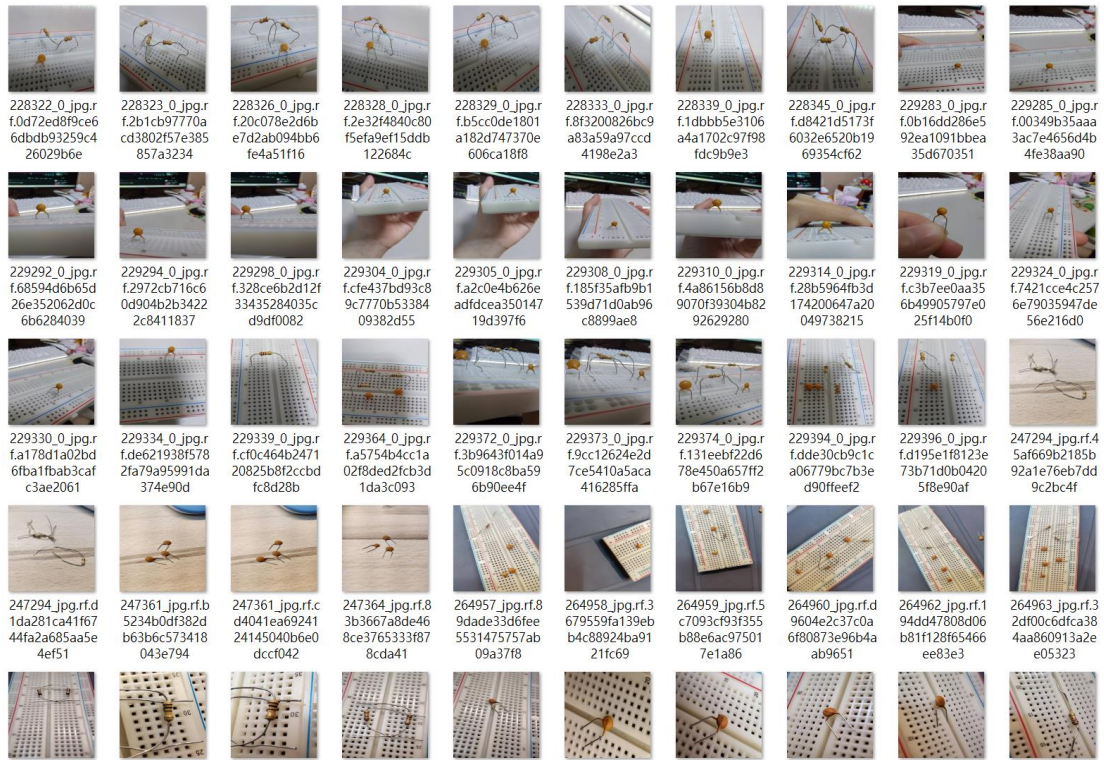
四、訓練方法

- 訓練資料的選擇與來源：

1. 來源：自行拍攝並在 ROBOFLOW 網站上進行資料擴增，最後生成總計 959 筆資料。為了訓練，我們將 75% 的照片作為 train set，20% 的照片作為 valid set，5% 的照片作為 test set。



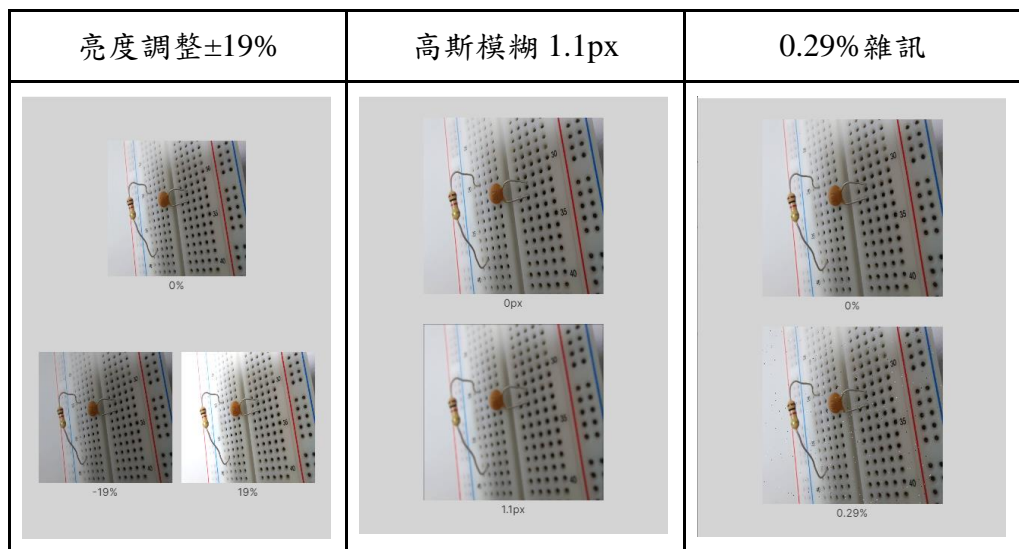
2. 選擇：在資料的選擇上，我們依照單一物件、複數物件、距離遠近、背景複雜度、光線不同、角度與擺放位姿的不同進行資料的收集。



● 參數配置：

在 data augmentation 上，我們將圖片進行水平與垂直鏡像、 90° - 180° - 270° 翻轉、 $\pm 10^{\circ}$ 旋轉($(10^{\circ}, 10^{\circ})$, $(10^{\circ}, -10^{\circ})$, $(-10^{\circ}, -10^{\circ})$, $(-10^{\circ}, -10^{\circ})$)、亮度調整 $\pm 19\%$ 、高斯模糊 1.1px 、 0.29% 雜訊的調整。其中，圖片的旋轉、鏡像、翻轉是因為元件的擺放沒有方向性，因此在圖片角度調整能增加實際使用的泛用性；亮度調整是因為實際操作時的光線亮度不盡相同，需要保證在強光與弱光下都能正確辨識物件；模糊與雜訊則是因為鏡頭的效能與解析度不同，如此能增加因為設備問題而降低的正確度。

水平鏡像、垂直鏡像	90° - 180° - 270° 翻轉	$\pm 10^{\circ}$ 旋轉
<p>preprocessed</p> <p>vertical</p> <p>horizontal</p>	<p>preprocessed</p> <p>clockwise</p> <p>counter-clockwise</p> <p>upside-down</p>	<p>$0^{\circ}, 0^{\circ}$</p> <p>$10^{\circ}, 10^{\circ}$</p> <p>$10^{\circ}, -10^{\circ}$</p> <p>$-10^{\circ}, 10^{\circ}$</p> <p>$-10^{\circ}, -10^{\circ}$</p>



至於在 training 的超參數上，我們使用的是 yolov8 的 default 配置：

epochs	100	Total number of training epochs. Each epoch represents a full pass over the entire dataset. Adjusting this value can affect training duration and model performance.
time	None	Maximum training time in hours. If set, this overrides the <code>epochs</code> argument, allowing training to automatically stop after the specified duration. Useful for time-constrained training scenarios.
patience	100	Number of epochs to wait without improvement in validation metrics before early stopping the training. Helps prevent overfitting by stopping training when performance plateaus.
batch	16	Batch size, with three modes: set as an integer (e.g., <code>batch=16</code>), auto mode for 60% GPU memory utilization (<code>batch=-1</code>), or auto mode with specified utilization fraction (<code>batch=0.70</code>).
imgsz	640	Target image size for training. All images are resized to this dimension before being fed into the model. Affects model accuracy and computational complexity.
optimizer	'auto'	Choice of optimizer for training. Options include <code>SGD</code> , <code>Adam</code> , <code>AdamW</code> , <code>NAdam</code> , <code>RAdam</code> , <code>RMSProp</code> etc., or <code>auto</code> for automatic selection based on model configuration. Affects convergence speed and stability.

lr0	0.01	Initial learning rate (i.e. <code>SGD=1E-2</code> , <code>Adam=1E-3</code>). Adjusting this value is crucial for the optimization process, influencing how rapidly model weights are updated.
lrf	0.01	Final learning rate as a fraction of the initial rate (<code>lr0 * lrf</code>), used in conjunction with schedulers to adjust the learning rate over time.
momentum	0.937	Momentum factor for SGD or beta1 for Adam optimizers, influencing the incorporation of past gradients in the current update.
weight_decay	0.0005	L2 regularization term, penalizing large weights to prevent overfitting.
warmup_epochs	3.0	Number of epochs for learning rate warmup, gradually increasing the learning rate from a low value to the initial learning rate to stabilize training early on.
warmup_momentum	0.8	Initial momentum for warmup phase, gradually adjusting to the set momentum over the warmup period.
warmup_bias_lr	0.1	Learning rate for bias parameters during the warmup phase, helping stabilize model training in the initial epochs.
box	7.5	Weight of the box loss component in the loss function, influencing how much emphasis is placed on accurately predicting bounding box coordinates.

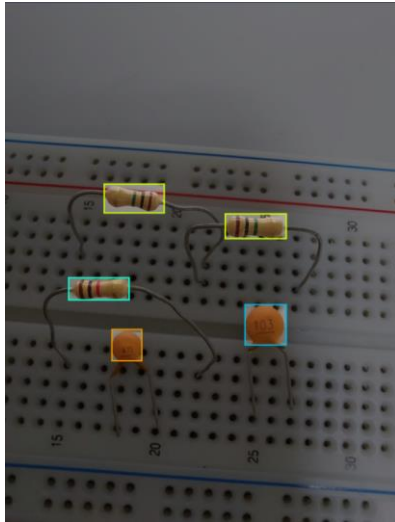
進行 batch=16、optimizer 使用 auto 模式調整參數、learning rate 從 0.01 到 0.0001 等等的模型訓練。不過在 epoch 數上，我們為求訓練快速並避免 over fitting 發生因此將其設定為 64。

而在 predict 的參數配置上，我們將信賴閾值設為 0.4:

```
results = model(img, stream=True, conf=0.4)
```


- 如何 label：

在 Roboflow 上進行協力的分類與標記，這裡我們使用電工四類比助教發的材料作為分類種類(電阻：1k、10k、100k、1M。電容：470pF、0.01uF)。標記時以元件主體不含接腳為主，bounding box 盡量避免背景。



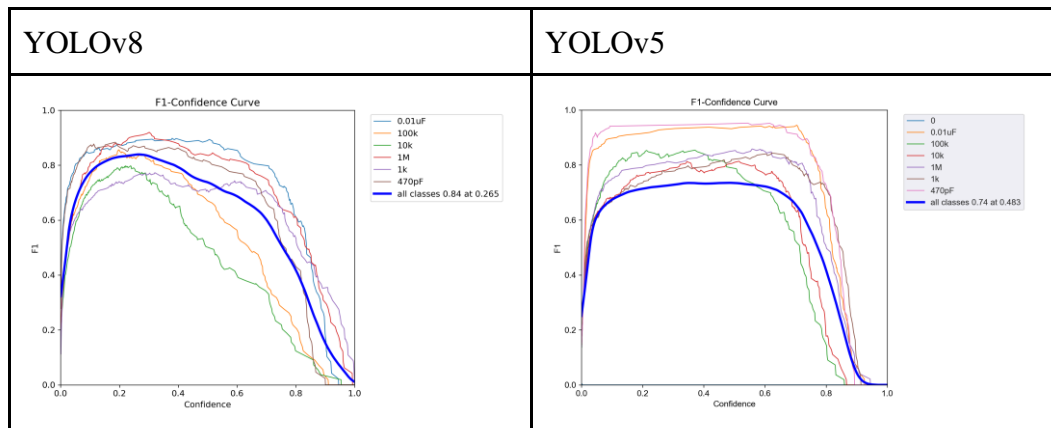
並確保各類別的資料數相同，避免 under present 的發生：



五、效能評估

主要以準確度為選擇依據，因為我們認為預測電路元件時，電路不會一直變化，所以偵測速度(FPS)不在我們評估的項目中。通常電工實驗在接電路時，正常只會有少數電阻不小心接錯，所以我們的目標為模型的預測要十分準確，希望準確度達 0.8 以上，可以正確找出錯誤電阻，並且不要把正確電阻預測成錯誤。我們藉由圖表分析和主要以實際操作模型預測元件的正確性作為評估標準。藉由嘗試不同模型架構，尋找最合適的模型，來達成可辨識出小物件電阻電容元件。

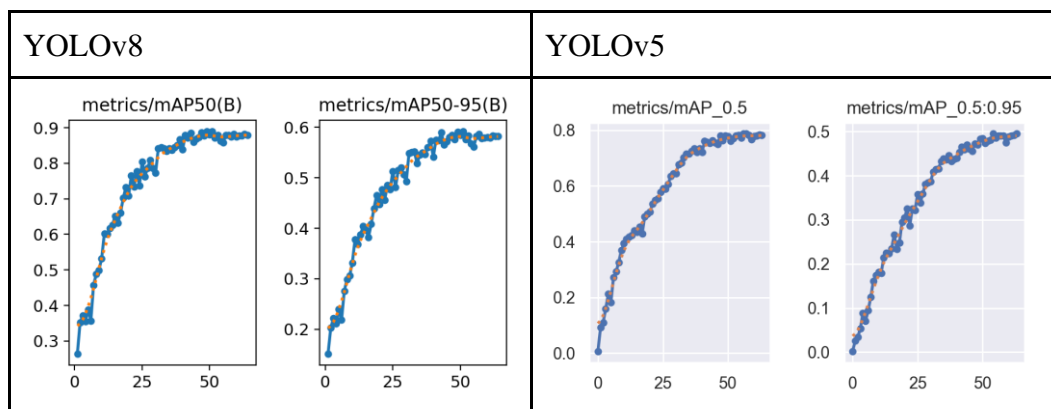
1.F1-confidence curve



F1 score 可以綜合評估模型，因為當精確度較高時，預測錯的機率低，但可能沒預測出來的機率高；當召回率較高時，預測出某類別的機率高，但預測錯的機率也可能很高。而 F1 score 為兩者的調和平均數，可以綜合評估模型。若 F1-confidence curve 的曲線峰值較高，可判斷為較好的模型。

YOLOv5 的最大 F1 分數為 0.74，對應的 confidence 值為 0.483，YOLOv8 的最大 F1 分數為 0.84，對應的 confidence 值為 0.265，雖然 YOLOv5 的曲線波動較小，表示模型的表現較穩，但 F1 curve 的高峰值卻只有 0.74，YOLOv8 為 0.84，有更高的準確性，所以我們選擇 yolov8 模型。

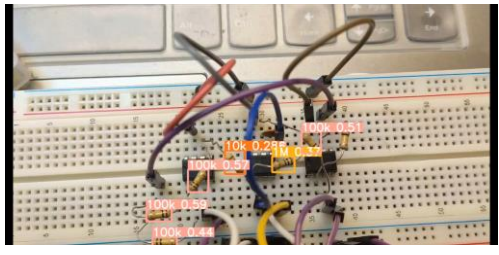
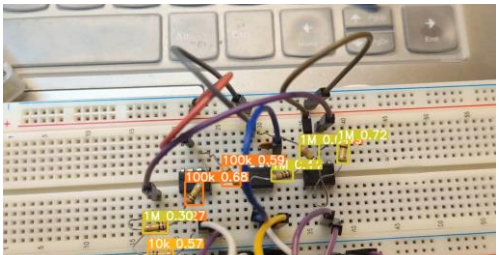
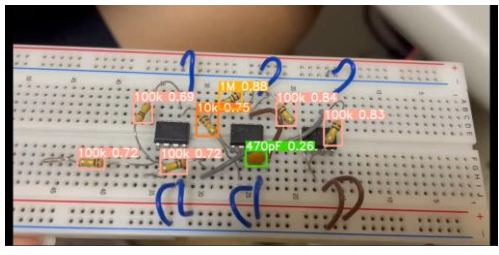
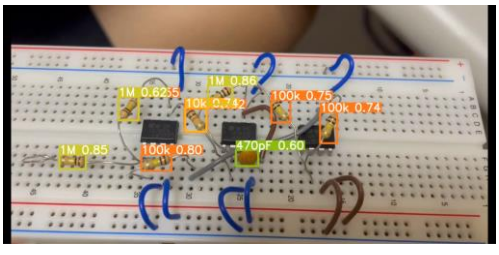
2.在不同 epoch 下的 mAP 圖



YOLOv8 的 mAP50 和 mAP50-95 在不同 epoch，都比 YOLOv5 高。YOLOv8 的 mAP50 為約 0.9，而 YOLOv5 的 mAP50 為約 0.8，高出 10%。從上面 mAP 兩張圖也可以看出選擇 YOLOv8 的效果較好。

由驗證集所跑出的效能評估，可以得知 YOLOv8 效果較好，其實 YOLOv5 在驗證集的效果也可以接受。然而當我們放上實際用接好電工實

驗四電路的影片時，YOLOv5 預測準確度十分差，可以幾乎是全預測錯誤。下面是預測的影片截圖。

YOLOv8	YOLOv5
	
電路上 7 個電阻在 YOLOv5 上只有一個 1M 的電阻預測正錯，其他六個都錯誤。而 YOLOv8 只有一個沒預測出來，剩下 6 個電阻值皆正確。	
YOLOv8	YOLOv5
	
下個影片也是 YOLOv8 預測十分準確，只有電容值預測錯誤，而 YOLOv5 的預測效果很差，且非常不穩定，10 秒影片幾乎是每 0.1 秒就改變預測結果，YOLOv8 可以很明確的顯示認為的電阻值，不太會改變。	
YOLOv8	YOLOv5
Speed: 2.7ms preprocess, 136.5ms inference	Speed: 0.5ms pre-process, 90.2ms
預測執行速度，YOLOv8 =136.5ms/每偵，YOLOv5 =90.2ms/每偵，YOLOv5 快很多	

六、方法實現成果與 DEMO

<https://youtu.be/3nA230rSUYy>

七、參考文獻

[1]YOLOv8 詳解 <https://henry870603.medium.com/object-detection-yolov8%E8%A9%B3%E8%A7%A3-fdf8874e5e99>

[2]基於動物辨識的 YOLOv5 與 YOLOv8 之性能比較

[https://acade.must.edu.tw/admin/files/47%E5%8D%B72-%E6%98%8E%E6%96%B0%E5%AD%B8%E5%A0%B1\(%E5%9F%BA%E6%96%BC%E5%8B%95%E7%89%A9%E8%BE%A8%E8%AD%98%E7%9A%84YOLOv5%E8%88%87YOLOv8%E4%B9%8B%E6%80%A7%E8%83%BD%E6%AF%94%E8%BC%83\)e024002-%E8%A8%B1%E7%80%9F%E5%82%91_%E5%90%B3%E4%BA%9E%E8%8A%AC_%E5%AD%AB%E5%AD%90%E6%81%A9_%E6%9E%97%E7%BE%A9%E6%A5%A0_20240416083826.pdf](https://acade.must.edu.tw/admin/files/47%E5%8D%B72-%E6%98%8E%E6%96%B0%E5%AD%B8%E5%A0%B1(%E5%9F%BA%E6%96%BC%E5%8B%95%E7%89%A9%E8%BE%A8%E8%AD%98%E7%9A%84YOLOv5%E8%88%87YOLOv8%E4%B9%8B%E6%80%A7%E8%83%BD%E6%AF%94%E8%BC%83)e024002-%E8%A8%B1%E7%80%9F%E5%82%91_%E5%90%B3%E4%BA%9E%E8%8A%AC_%E5%AD%AB%E5%AD%90%E6%81%A9_%E6%9E%97%E7%BE%A9%E6%A5%A0_20240416083826.pdf)

[3]yolov8 configuration

<https://docs.ultralytics.com/usage/cfg/>

[4]吳明芳,陳添智,陳俊榮,翁若綿 - AOI 視覺影像技術於電路板元件檢測之應用

AOIEA 自動光學檢測設備聯盟>知識分享>技術專欄

[5]Mihriban Günay, Murat Köseoğlu, ” Detection of circuit components on hand-drawn circuit images by using faster R-CNN method”,
INTERNATIONAL ADVANCED RESEARCHES and ENGINEERING
JOURNAL, Volume 05, Issue 03 December 2021

[6]Faster-RCNN 及 YOLO 比較

<https://blog.csdn.net/JianqiuChen/article/details/104018322>

八、心得與討論

- B103012041 黃詣辰

我們這次專題的主題是結合了我們自己的科系(電機系)而發想的，因為每次電工實驗總是被種類繁多的電阻電容深深的困擾著，debug 的時候也常常因為記不起來所有電阻的色碼而感到困擾，而且常常也會因為上一堂課的同學將電阻分類錯誤，導致我們拿到錯誤的電阻而不自知，

我們想要透過自己設計的系統來幫助大家減少課堂中的困擾，所以就有了想要做這個專題的想法。

當然，訓練過程中我們也遇到了不少的困難，起初訓練完成，在電腦上測試的時候就發現 yolov4、yolov5 的效果都沒有我們原先想像的好，另外一個問題是，當我們接上助教所提供的 webcam 時，因為解析度太低，而我們的電阻又很小，導致根本無法清楚的看到整個電阻元件，當初甚至有了想要更換主題的想法，但我們還是抱持著最後一絲希望去測試 yolov8，在電腦上的測試結果很好，但因為 v8 是比較新的模型，所以網路上比較少人教學該如何把模型載入至 nano 上，經過了一番嘗試之後，我們成功的載入模型，且準確度也像在電腦上測試的一樣準確。

還好我們當初沒有更換主題，最後才能夠做出成功的版本，也讓我們更加熟悉機器學習實作的流程，同時培養我們解決問題、小組合作分工的能力，經過一學期之後也確實學到了不少的東西！

- B103012034 黃麗穎

期末專題我們組經歷許多波折才完成最終成品，我們這次主題選擇是以電工實驗四的電阻電容當作分類的類別，但剛好電工四的電阻值都是倍數的關係(1k 10k 100k 1M)，所以其實每個電阻只有一條色碼不同，其餘三條色碼都是相同的，在作業六的時候就發現非常難分辨電阻，但我們選擇繼續做的原因是我們覺得資料集優化後可能可以改善預測不準，作業六是我們拿 kaggle 公開資料集，每張照片本身就很不清楚，所以訓練效果不好，期末專題就都改成自己拍攝電阻，第二個原因是覺得這個主題對電機系學生很實用可以幫忙偵錯電路。只是在我們用自己的資料集訓練完第一個模型 yolov4 後，發現成效還是非常差，準確度大概只有 10%，後來換了 yolov5 也只提升一點點，與我們預期的效果差距太大。

最後我們選擇 yolov8，其實我一開始就想使用 yolov8，但上網查詢發現部屬在 jetson nano，有些困難，但最後真的是其他模型準確度太差，就算增加資料集數量也是沒有改善，所以還是換了 yolov8 試試看，結果非常準確，差了兩年提出的模型，效能差距好大。在電腦上成功預測後，就只剩下部署環境的方式，幸好最後真的成功，當然中間也有發生發現 webcam 的畫質太差，導致電阻色碼完全是模糊或消失，最後只好用影片方式呈現。

期末專題或這堂課我覺得我學到的主要是思考該怎麼使用機器學習在解決現有問題上，我覺得訓練模型很好上手網路資源很多，但要如何有效

利用才是重點。期末其他組別報告他們的創意，尤其是猴子偵測，非常有創意，我學習到很多。

- B103015001 林佳明

由於電工實驗是電機系的必修，因此我們每個組員都非常理解在接完複雜的電路之後發現實驗結果有誤需要偵錯是一件費時又費工的事，拿實驗器材時也常常拿到分類錯誤的元件，因此在期末專題構想的時候我們很快便決定好了我們的主題:電工實驗小助手，期望成果可以減輕大家做電工實驗的負擔並讓器材歸位更加容易以減少錯誤分類的情形發生。

為了提升效能，我們重新翻閱課堂講義使用各種上課教的方法來做系統改善(例如使用 TensorRT)，因此我覺得期末專題是一個複習所學的良好機會。我們也嘗試了多個 YOLO 版本並分析他們的優缺點，由於我們認為電路偵錯對準確度的要求比速度還要重要很多，而 YOLOv8 雖然辨識速度非常緩慢，但好在辨識準確度很高，因此我們最終選擇採用 YOLOv8。另外一個遇到的困難是即時影像偵測的部分只能使用電腦的鏡頭，因為 webcam 照出來的畫面太模糊了，連看電阻顏色線條都有困難，我想若是之後可以換一個解析度更高的 webcam 應該可以大大的增加這個期末專題使用的方便性。我覺得評量尺規中設置期末專題與國內外作品比較也是很好的學習，這讓我們可以查找資料來了解其他人已經做過什麼，用了什麼方法，並且能夠了解我們自己的專題的重要性與優點。

我很幸運能和我積極認真的組員們一起完成這門課大大小小的作業及任務，大家窩在一起想辦法解決問題的感覺很棒，合作的過程中除了完成自己的部份還會互相提醒哪裡可以更好，是很愉快的合作經驗。

- B103015006 胡庭翊

我們這次的期末專題是結合電機系每個人都必然會碰到的元件以及電路，結合 yolo 物件偵測技術發想的。最初我們認為電阻與電容規格固定不會有甚麼變化性，且元件作為拍攝素材具有易取得性，預期這會是一個很好的主題，然而，從作業六的訓練結果我們便發現實際上並沒有我們想的那麼容易，除了最初使用的公開資料集本身拍攝的不是很好之外，常用的阻值(0.1k、1k、10k、100k)是由簡單倍數構成，換言之電阻色碼僅相差一碼，在實務上模型其實不見得會正確抓取特徵。

在期末專題上我們最終選擇自行拍攝資料集，針對不易辨識色碼的問題額外調整了光線、亮度以及焦距，然而我們最初使用 yolov5 的成效依舊不好，電容我們預期是以數字作為辨識依據，這部分的成效尚且還可以，然而電阻塞碼中的紅、橙、黃應該是因為顏色太接近，所以總是判讀成同個類別。另外，在實務上 webcam 的解析度也有限，元件上的色碼、容值都看不出來，這也是個問題。

我們一度有想過更換主題，但時間有限進度實在迫在眉睫，好加在我們在嘗試過調整 batch 數、從 Darknet 轉換到 TensorRT 等方法後，最終發現使用 YOLOv8 的正確率十分的高！

透過這次的期末專題我們更加熟悉如何在實務上使用機器學習模型進行應用，也發現透過 python 呼叫及控制程式能將模型做到的各種可能性，在這門課上我除了對如何使用機器學習模型有了實務上的經驗外，亦增加了對各種參數的敏銳性與對各種狀況排除的經驗，例如，在課堂上實際將模型放置在樹梅派或 jetson nano 上時往往會出現一些意料之外的錯誤，有時需要針對該錯誤指令進行排查，有時甚至需要重燒 SD 卡，而在構思模型的應用功能上時我們也在種種邏輯錯誤中討論、偵錯，完善程式碼，這些過程雖然有時痛苦，但也因如此才能造就出做出成品的感動，並拓展我們對於機器學習的經驗與興趣。