



自動農作物分類與疾病檢測系統建置： 應用於蘋果與番茄疾病檢測

蘇明祥^{1,*} 李承彥²

¹ 東吳大學資料科學系助理教授

² 東吳大學資料科學系學生

摘要

臺灣於農業上面臨人口高齡化、勞動力不足等問題，因而經常無法於農作物種植過程中給予植株有效的管理，容易使農作物遭受疾病的威脅，進而導致重大損失。本研究擬建置一農作物疾病偵測系統並應用於偵測蘋果的黑星病、番茄早疫病、晚疫病和葉黴病。本論文首先擬收集蘋果與番茄農作物語料集和 PlantVillage 農作物疾病資料集，其次根據農作物語料集訓練葉片偵測模型與農作物分類模型用以偵測植株葉片及識別農作物種類，最後將偵測出之植株葉片輸入疾病檢測模型進行疾病的偵測，找出感染疾病的植株。此農作物分類與疾病檢測系統擬透過空拍機協助農民提升監控農作物疾病時的效率，使農民能對植株早期發現及進行早期治療，避免農作物疾病所造成的農業影響。

關鍵詞：疾病偵測、農作物疾病偵測系統、葉片偵測、YOLO、Vision Transformer

* 通訊作者：蘇明祥

電子郵件：huntfox.su@gmail.com

（收件日期：2024 年 1 月 15 日；修正日期：2024 年 3 月 5 日；接受日期：2024 年 3 月 5 日）



Establishment of an Automatic Crop Classification and Disease Detection System: Applied to Apple and Tomato Disease Detection

Ming-Hsiang Su^{1,*}, Cheng-Yen Lee²

¹Assistant Professor, Department of Data Science, Soochow University

²Student, Department of Data Science, Soochow University

Abstract

Agriculture in Taiwan is facing problems such as population aging and labor shortage. Therefore, effectively managing the crops in the planting process is often impossible, easily exposing them to disease threats and leading to heavy losses. This study builds a crop disease detection system and applies it to apple scab, tomato early blight, late blight, and leaf mold. The system first collects apple and tomato crop photos and crop disease data from the Plantvillage dataset on the Kaggle platform. Secondly, according to the crop dataset, the crop type recognition and leaf detection model is trained to identify crop types and detect plant leaves. Finally, the detected leaves are inputted into the disease recognition model for disease detection, and the plants infected with the disease are found. The crop detection system intends to help farmers improve the efficiency of monitoring crop disease through aerial cameras so that farmers can treat plants early and avoid agricultural impacts caused by crop disease.

Keywords: disease detection model, crop disease detection system, leaf detection model, YOLO, Vision Transformer

* Corresponding Author: Ming-Hsiang Su
E-mail: huntfox.su@gmail.com

壹、研究動機與研究問題

臺灣農業人口高齡化問題嚴重，根據行政院農業委員會農糧署（2020）統計，以農牧為本業的人口當中有 45.6% 為高齡戶，此情形造成在農作物種植的過程中經常缺乏足夠勞動力進行管理，而農民在無法有效監控農作物的狀況下，容易因農作物遭受疾病肆虐而造成大規模的損失。此外，目前臺灣相關研究多為針對特定一種農作物進行偵測，但有許多農民同時種植多項農作物，因此系統若能夠應用在多種農作物將可帶來益處。本研究將藉由深度學習技術建立影像分類和農作物疾病偵測模型，並進行訓練及驗證，使模型擁有足夠良好的效能以應用於農作物種植實務中。由於蘋果和番茄為臺灣重要的農作物之一，本論文將農作物自動分類與疾病檢測模型應用於偵測蘋果和番茄作物之疾病，將可透過空拍機拍攝農作物影像進行疾病偵測，以協助農民提升監控農作物疾病的效率，並降低被疾病影響而受到的損失。

貳、文獻回顧與探討

一、農作物與疾病葉片病徵

蘋果於臺灣主要產地於臺中市、新竹縣與南投縣，採收期約為每年 9 ~ 12 月，於 2021 年收成量達 1,458 公噸（農業部，n.d.）。種植過程中易遭受黑星病的威脅，氣溫於攝氏 15 ~ 20 度連續降雨時發病嚴重，發病初期於葉片出現橄欖色的不規則圓形絨毛斑點，後逐漸轉灰黑色，且葉片變小、畸形、捲曲，如圖 1 所示（農業部農業藥物試驗所，n.d.）。而番茄於臺灣主要產地為新竹縣、花蓮縣、宜蘭縣、臺東縣、嘉義縣、彰化縣、屏東縣和南投縣，採收期約為每年 1 ~ 4 月（農業部，n.d.）。番茄種植過程中易受到早疫病、晚疫病和葉黴病，如圖 2，而影響收成。早疫病易於攝氏 28 ~ 30 度且高濕環境下發生，受害植株初期葉片呈暗褐色至水浸狀黑色小斑點，逐漸擴大為輪紋狀斑點，且周圍有黃色暈環，最終則葉片脫落；晚疫病易發生於天氣多濕轉涼下，被害葉初成水浸狀，不久轉為暗褐色向周圍擴大，並時產生白色黴狀物；葉黴病易發生於攝氏 18 ~ 26 度高濕環境，葉片背面初期呈現不明顯灰白圓形斑點，正面則呈淡黃色斑點，隨時間產生灰黴且轉黃褐色至灰紫色並捲曲枯死（農業部農業藥物試驗所，n.d.）。

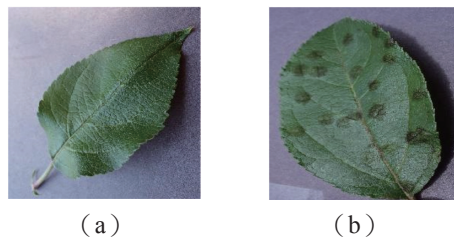


圖 1 (a) 健康蘋果葉片、(b) 蘋果黑星病葉片

資料來源：Hughes and Salathe（2015）。

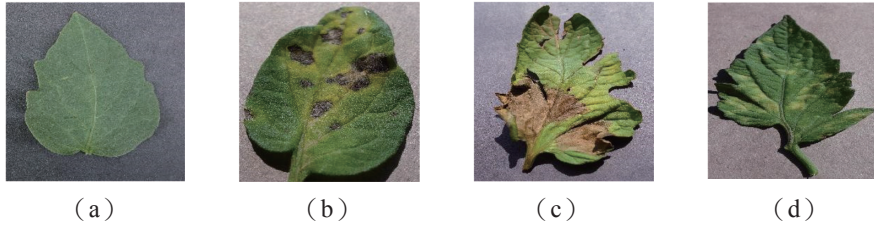


圖 2 (a) 番茄健康葉片、(b) 早疫病葉片、(c) 晚疫病葉片、(d) 葉微病葉片

資料來源：Hughes and Salathe (2015)。

二、Vision Transformer (ViT)

影像分類為深度學習應用於電腦視覺中的技術，透過演算法將影像進行分類，應用於自動駕駛、醫學和運動科學等多種影像相關任務。目前影像分類演算法為基於卷積神經網路 (Convolutional Neural Network, CNN) 所建構之分類模型，如：GoogLeNet (Szegedy et al., 2015)、Residual Network (ResNet) (He, Zhang, Ren, and Sun, 2016) 和 Dense Convolutional Network (DenseNet) (Huang, Liu, Van Der Maaten, and Weinberger, 2017)。除 CNN 模型為基礎之影像分類模型外，Dosovitskiy et al. (2020) 提出 ViT 模型，ViT 是首度將自然語言領域中的轉換器 (Transformer) 架構應用於電腦視覺任務之影像處理模型，架構如圖 3 所示，藉此萃取影像資訊後進行分類。

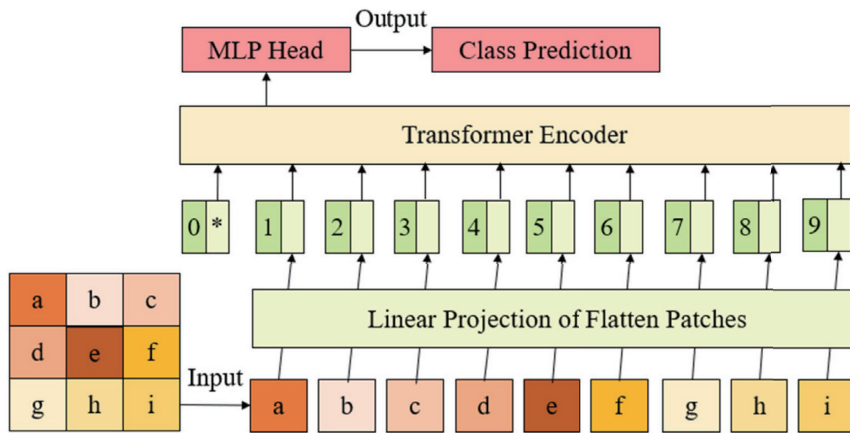


圖 3 ViT 架構

資料來源：Dosovitskiy et al. (2020)。

ViT 為降低運算成本，首先將影像分割成 N 個 ($P \times P \times C$) 的補丁 (Patch)，每個補丁以 X_p^N 表示，其中 C 是通道數 (Number of Channels)，(P, P) 是每個圖像補丁的分辨率 (The Resolution of Each Image Patch)。其次為補丁嵌入 (Patches Embedding)，所有 X_p^N 需透過 $(P \times P \times C) \times D$ 的權重矩陣 $\mathbf{E}$ 進行線性映射 (Linear Projection)，並和類似 Bidirectional

Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, and Toutanova, 2018) 中的類別標記 (Class Token) (X_{class}) 結合為 $(N+1) \times D$ 的二維向量，再加上位置嵌入 (Position Embedding) ($\mathbf{E}_{pos}$) 為 Z_0 輸入轉換器編碼器 (Transformer Encoder)。 Z_0 公式如方程式 (1) 所示。

$$Z_0 = [X_{class}; X_p^1 \mathbf{E}; X_p^2 \mathbf{E}; \dots; X_p^N \mathbf{E}] + \mathbf{E}_{pos} \quad (1)$$

where $\mathbf{E} \in \mathbb{R}^{(P \times P \times C) \times D}$, $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$

轉換器編碼器由 L 層的 Multi-Head Self-Attention (MSA) 與多層感知器 (Multilayer Perceptron, MLP) 組成，如圖 4。每一層 Z 會先經過層級正規化 (Layer Normalization, LN) 再進入 MSA，輸出後進行殘差連結 (Residual Connection) 與一開始輸入的 Z 相加，通過 LN 和 MLP 後再次進行殘差連結後輸出， Z^0 則會再經過一次 LN 為 Y ，而 Y 代表輸入影像的資訊。經過多層運算後，最後一層輸出的 Y 便會透過 MLP 預測影像類別以輸出結果。

$$Z'_\ell = MSA(LN(Z_{\ell-1})) + Z_{\ell-1}, \text{ where } \ell = 1 \dots L \quad (2)$$

$$Z_\ell = MLP(LN(Z'_\ell)) + Z'_\ell, \text{ where } \ell = 1 \dots L \quad (3)$$

$$Y = LN(Z_L^0) \quad (4)$$

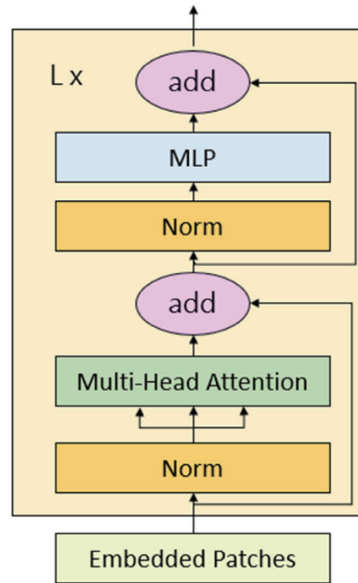


圖 4 轉換器編碼器

資料來源：Dosovitskiy et al. (2020)。

三、物件偵測

物件偵測 (Object Detection) 為影像分類技術延伸應用的一項深度學習方法，此技術的概念為透過演算法偵測出圖片或影片中的目標物件並以邊界框 (Bounding Box) 框住，同時將邊界框內物件進行分類。物件偵測模型可大致分為二階段偵測 (Two-Stage Detector) 和一階段偵測 (One-Stage Detector)，二階段偵測相比一階段偵測通常速度較慢，但有更高的準確率 (Accuracy) (Du, Zhang, and Wang, 2020)。二階段偵測藉由演算法事先於圖片進行區域提議 (Region Proposal) 找出潛在區域，再針對這些區域分類並於原始圖片框出物件位置，以 Region-Based Convolutional Neural Networks (R-CNN) (Girshick, Donahue, Darrell, and Malik, 2014) 為範例：偵測流程第一步使用選擇性搜索 (Selective Search) 演算法產生多個區域提議，再將每個區域提議壓縮後輸入 CNN 提取影像資訊，最後藉由支持向量機 (Support Vector Machine, SVM) 將影像分類並經由線性回歸模型校正邊界框。一階段偵測則是將兩步驟合而為一，物件位置偵測和分類同時進行，以 YOLOv1 (Redmon, Divvala, Girshick, and Farhadi, 2016) 為例：首先將輸入圖片調整為 448×448 並切割成 7×7 個網格，再將其輸入 CNN 中並針對每個網格皆負責的邊界框進行分類，最後藉由非最大抑制 (Non-Maximum Suppression, NMS) 篩選邊界框後輸出結果。

四、You Only Look Once (YOLO)

YOLO 於 2015 年首次提出，是近年來相當熱門的物件偵測模型，於應用上擁有優秀的偵測速度與準確率，系列至 2022 年已有多種經過改善的版本，包括 YOLOv1 (Redmon et al., 2016)、YOLOv2 (Redmon and Farhadi, 2017)、YOLOv3 (Redmon and Farhadi, 2018)、YOLOv4 (Bochkovskiy, Wang, and Liao, 2020)、YOLOv7 (Wang, Bochkovskiy, and Liao, 2023) 與許多其他版本，而各版本之間藉由模型架構和訓練過程優化以在減少運算成本的同時提升預測準確度。YOLOv4 於 2020 年發表，模型架構由主幹 (Backbone)、頸部 (Neck)、頭部 (Head) 組成，負責萃取影像特徵、融合不同尺寸特徵圖、預測輸出，分別為 CSPDarknet-53、Spatial Pyramid Pooling (SPP) + Path Aggregation Network (PAN)、YOLOv3 Head。CSPDarknet-53 為 YOLO (Redmon and Farhadi, 2018) 所使用的 Darknet-53 加上 Cross-Stage Partial connections (CSP) 結構，CSP 可增加 CNN 學習能力，使網路在獲取更加豐富的影像特徵情況下減少運算成本以至於模型輕量化；SPP + PAN 則能夠透過有效融合不同尺度的特徵圖以增加最終輸出特徵圖的資訊。YOLOv7 發表於 2022 年，與 YOLOv4 相比，網路架構於主幹、頸部、頭部皆進行更改。主幹使用 Extended Efficient Layer Aggregation Networks (ELAN)，結構如圖 5，使網路可藉由控制最長與最短梯度的路徑以使更深層的網路能夠有效的學習和收斂，此外在卷積層和批次正規化 (Batch Normalization) 後的激活函數由 Mish 更改為 Sigmoid Linear Unit (SiLu)。

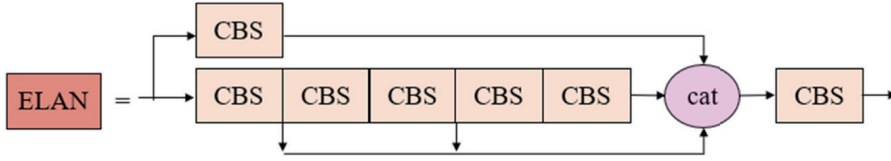


圖 5 ELAN 架構圖

資料來源：Wang et al. (2023)。

$$Mish = x \times \tanh(\ln(1 + e^x)) \quad (5)$$

$$SiLu = x \times \frac{1}{1 + e^{-x}} \quad (6)$$

頸部部分 YOLOv7 在 SPP 架構上再加上 CSP，如圖 6，使模型可透過 CSP 增加融合不同尺寸特徵圖的能力，此外頸部也使用類似於 ELAN 架構的網路，本研究將其以 ELAN-N 代稱，如圖 7。另外在模型中主幹和頸部皆使用一個網路區塊於多數 ELAN 和 ELAN-N 之後，其將兩個通道的輸出進行連接（Concatenate），本研究將此網路區塊稱為 MP，如圖 8。

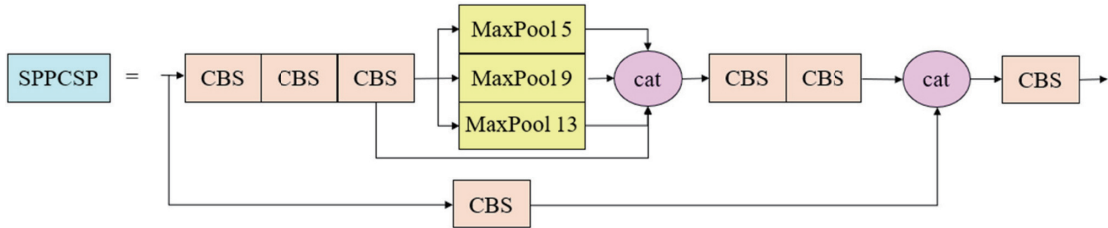


圖 6 SPPCSP 架構圖

資料來源：Wang et al. (2023)。

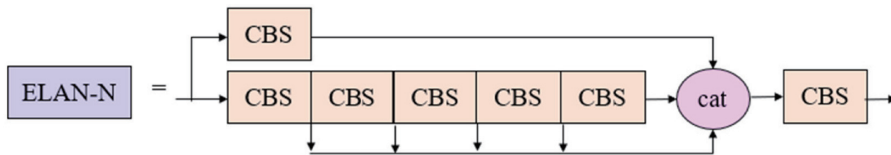


圖 7 ELAN-N 架構圖

資料來源：Wang et al. (2023)。

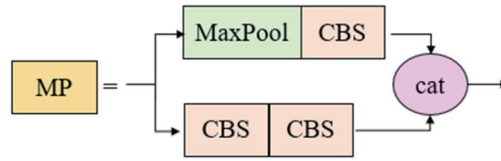


圖 8 MP 架構圖

資料來源：Wang et al. (2023)。

頭部由使用 Model Re-Parameterized 集成技術的 RepConv，如圖 9，和 CBS 組成，此技術使 RepConv 在訓練過程中分為三個分支，並在應用時將分支聚合為一個，以至模型於應用時可減少運算成本並提升效能。YOLOv7 整體模型，如圖 10 所示。

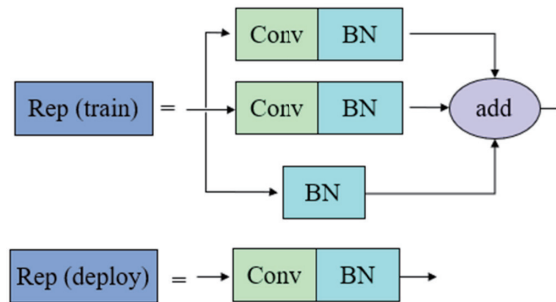


圖 9 RepConv 架構圖

資料來源：Wang et al. (2023)。

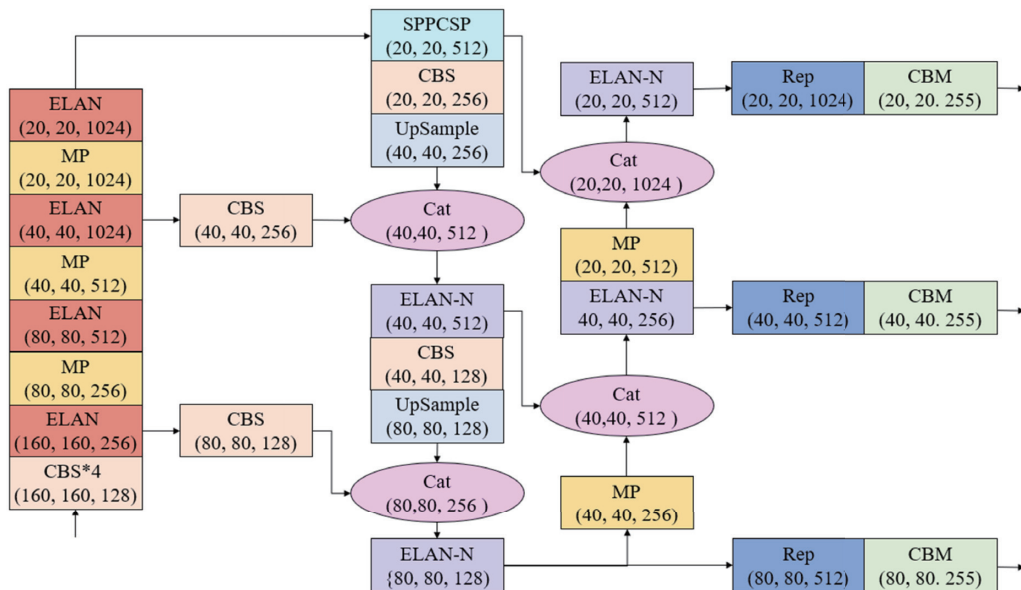


圖 10 YOLOv7 架構圖

資料來源：Wang et al. (2023)。

參、研究方法與步驟

一、農作物疾病偵測系統架構

本研究擬針對蘋果黑星病、番茄早疫病、番茄晚疫病和番茄葉黴病進行疾病偵測系統的建立，使用者將可透過空拍機利用此系統監控農作物疾病，系統架構如圖 11 所示。首先空拍機將拍攝農作物影像傳送至系統，藉由農作物種類及葉片偵測模型偵測影像中植株的葉片，且將該葉片進行農作物種類的辨識，接下來根據定位擷取葉片的影像並去除葉片主體外的背景後輸入所屬農作物種類的疾病偵測模型，最後輸出對於葉片的疾病偵測結果以反映該葉片所屬農作物植株是否生病。

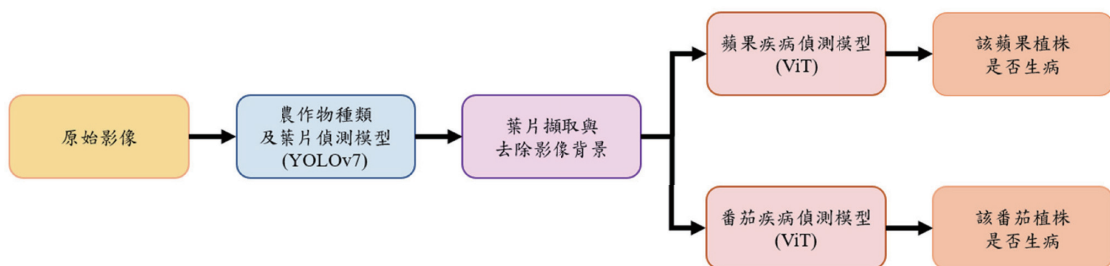


圖 11 農作物疾病偵測系統架構

資料來源：作者自製。

二、資料收集

於農作物種類及葉片偵測模型所使用到的資料預計將透過 Google 圖片蒐集，預計蒐集距離植株約三公呎以內可觀測到葉片的蘋果和番茄至少各 500 張植株影像。蘋果和番茄農作物疾病偵測模型則預計使用 Kaggle 平臺 PlantVillage 資料集中的健康蘋果葉片 1,645 張、蘋果黑星病葉片 630 張、健康番茄葉片 1,591 張、番茄早疫病葉片 1,000 張、番茄晚疫病葉片 1,909 張、番茄葉黴病葉片 952 張。模型的訓練和驗證部分，本研究預計將資料集拆分為 80% 訓練集和 20% 測試集，以此評估系統於農作物種類集葉片偵測模型、蘋果和番茄疾病偵測模型的檢測能力。

三、資料前處理

本研究使用 Tzutalin (2015) 於 Github 提供的 Labellmg 軟體，對蘋果和番茄植株影像進行葉片物件框座標與類別的標籤，並於標籤後輸出能夠用於 YOLOv7 模型訓練中的資料格式檔案。另外，為避免模型對於訓練資料的過度擬合而造成驗證時因未見過之資料類型偵測的錯誤判斷，本研究欲對訓練資料進行資料增強 (Data Augmentation)，藉由對於影像進行色相—飽和度—明度增強 (Hue-Saturation-Value Augmentation)、隨機平移 (Random Translation)、隨機縮放 (Random Scale)、隨機翻轉 (Random Flip)、馬賽克 (Mosaic)，

以增加資料量並使影像資料擁有更多元的變化，加強模型於實際應用時的辨識能力。於蘋果和番茄疾病偵測模型中，藉由對於影像進行隨機縮放裁剪（Random Resized Crop）、隨機翻轉、隨機仿射變換（Random Affine）、顏色抖動（Color Jitter）以進行資料擴增。

四、農作物種類及葉片偵測

本研究擬選擇 YOLOv7 作為農作物疾病偵測系統中的農作物種類及葉片偵測模型，因 YOLOv7 為當今強大且擁有優秀性能的物件偵測模型，與其他傳統偵測模型相比，YOLOv7 於準確率和速度上皆有相當優異的表現。藉由 YOLOv7，此農作物疾病偵測系統可於輸入的農作物影像之中偵測葉片位置並區分該植株是屬於哪一類農作物。接著該葉片圖像進而被擷取並輸入至該類農作物之疾病偵測模型。為評估 YOLOv7 應用於農作物種類及葉片偵測模型，本研究使用平均精確率（Mean Average Precision, mAP）作為評估指標。為得出 mAP，首先必須以預測目標邊界框與真實目標計算 Intersection Over Union（IOU），並藉此計算模型的混淆矩陣。若 IOU 大於本研究設定的閾值 0.5，則判定這個目標是 True Positive（TP），反之小於閾值則判定是 False Positive（FP）。

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (7)$$

接下來可透過混淆矩陣計算召回率（Recall）和精確率（Precision），並以 X 軸召回率和 Y 軸精確率繪製出 PR 曲線（Precision-Recall Curve），而 AP 則為 PR 曲線下的面積（Area Under Curve, AUC），最後 AP 除以類別的數量得出 mAP。mAP 高便代表此農作物品種及葉片偵測模型擁有好的物件偵測能力。mAP@.5 表示當 IOU 閾值為 0.5 時的平均精確率，mAP@.5:.95 則表示 IOU 閾值從 0.5 到 0.95 之間的平均精確率。

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$mAP = \frac{1}{Number\ of\ Class} \sum AP(Class) \quad (10)$$

五、葉片擷取與去除影像背景

由於蘋果和番茄疾病偵測模型的訓練資料影像背景幾乎無雜訊干擾，因此為避免於實務

中農作物種類及葉片偵測模型擷取葉片影像後，疾病檢測模型遭受葉片影像的背景雜訊影響，本研究擬藉由 Daniel Gatis 於 Github 平臺所提供的 Rembg 方法 (Gatis, 2020) 將擷取葉片影像背景雜訊去除並保留葉片主體，如圖 12。



圖 12 葉片影像去背

資料來源：作者自製。

六、蘋果和番茄農作物疾病偵測

本研究擬選擇 ViT 建立分別用於蘋果和番茄的農作物疾病偵測模型。ViT 跳脫以往電腦視覺經常使用的 CNN 架構，改以自然語言領域中的轉換器建立影像分類模型，且於訓練資料充足的情況下，ViT 能夠獲得優秀的準確率並同時擁有優良計算效率。ViT 於輸入影像分割中，可將影像分割為 N 個 $16 \times 16 \times C$ 或 $32 \times 32 \times C$ 的補丁，且根據模型的層數 (Layers)、隱藏層 (Hidden) 大小、MLP 大小、注意力頭數 (Heads) 可分成 ViT-Base 和 ViT-Large 兩種預模型。本研究擬實驗 ViT-Base-16、ViT-Base-32、ViT-Large-16、ViT-Large-32 找出最優模型使其可進行正確的疾病的偵測，並且將閾值設定為 0.5 用以反映該植株是否正遭受疾病威脅，若大於 0.5 判定為生病，反之判定沒生病。為評估各 ViT 架構分別於蘋果和番茄的疾病偵測模型，本研究擬計算模型預測準確率、召回率、精確率作為評估指標。

肆、研究結果

一、農作物種類及葉片偵測

於蘋果的葉片資料蒐集中，研究與尚品 (蘋) 果園合作，針對蘋果各品種包含富士蜜蘋果、青龍蘋果、金冠蘋果、世界第一號、秋香蘋果、陸奧蘋果、五爪蘋果、津輕蘋果、海棠蘋果建立模型，共收集 599 張影像。番茄部分與金勇番茄農場進行合作，品種包含：日本桃太郎番茄、金棗番茄、日本紅櫻桃番茄、以色列串收番茄、南美洲芝麻紅番茄、義大利黃金番茄，共收集 743 張影像。收集範例如圖 13 (a) 和圖 13 (b)，皆為距離植株 3 公尺內之影像。

於此階段中，研究首先利用 LabelImg 對葉片影像進行標註，標註結果如圖 14，每一行代表一個標註物件，由左至右則分別是類別、邊界框的中心點 X 座標、中心點 Y 座標、寬度、高度的標準化結果。其中，類別 0 代表蘋果葉片，類別 1 則代表番茄葉片。

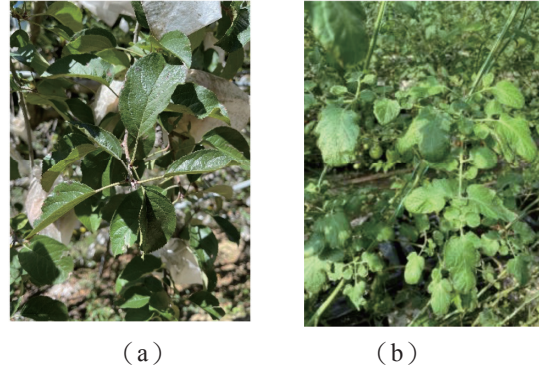


圖 13 (a) 蘋果植株、(b) 番茄植株

資料來源：Hughes and Salathe (2015)。

```
1 0.527282 0.523065 0.665675 0.357143
1 0.811839 0.100322 0.231481 0.184276
```

圖 14 資料標註檔

資料來源：作者自製。

接下來，研究使用了 YOLOv7 建立農作物種類及葉片偵測模型，YOLOv7 透過 COCO 資料集進行預訓練，因此已擁有基礎物件偵測能力。此階段目的為將 YOLOv7 進行下游訓練，使模型能夠捕捉出影像中的葉片並將其進行農作物的分類。訓練環境使用 Google Colab Pro，訓練超參數優化器 (Optimizer) 使用 Adam (Kingma and Ba, 2014)，時期 (Epoch) 為 10，輸入影像尺寸為 640×640 ，其他則使用 YOLOv7 預設數值。另外，實驗中所使用到的資料擴增技術於描述如表 1 所示。

表 1 YOLOv7 資料增強設定

資料增強技術	數值	數值意義
HSV- 色相增強	0.015	色調會在原始值隨機增加或減少 1.5%
HSV- 飽和度增強	0.700	飽和度會在原始值隨機增加或減少 70%
HSV- 明度增強	0.400	亮度會在原始值隨機增加或減少 40%
隨機平移	0.200	沿著水平和垂直方向隨機平移 20%
隨機縮放	0.500	隨機縮放 50%
隨機翻轉	0.500	50% 的機率水平翻轉
馬賽克	1.000	與其他三張圖像拼接成一張

註：HSV：色相 (Hue)、飽和度 (Saturation)、明度 (Value)。

資料來源：作者自製。

第一階段結果如表 2 所示，研究針對模型 Batch Size，包含 8、16、32、64 的設置進行實驗，並以精確率、召回率、 $mAP@.5$ 、 $mAP@.5:.95$ 評估模型的性能。在表 2 中，我

們可以看到當 Batch Size 設定為 32 時可以獲得最佳結果，其精確率、召回率、mAP@.5、mAP@.5:.95 分別為 74.57%、77.92%、84.70%、53.12%。

表 2 農作物及葉片偵測模型 Batch Size 實驗結果

模型	精確率 (%)	召回率 (%)	mAP@.5 (%)	mAP@.5:.95 (%)
Model_1 (Batch Size = 8)	63.60	65.17	71.00	40.03
Model_2 (Batch Size = 16)	73.83	75.26	82.74	50.29
Model_3 (Batch Size = 32)	74.57	77.92	84.70	53.12
Model_4 (Batch Size = 64)	72.16	75.80	81.32	49.47

註：mAP：Mean Average Precision。

資料來源：作者自製。

於此之後，研究在 Model_3 的基礎上進行初始學習率 (Learning Rate, LR) 調整的實驗，範圍包含 0.1、0.01、0.001，結果如表 3 所呈現，當初始 LR 設置於預設的 0.01 時，可以獲得最佳精確率、召回率、mAP@.5、mAP@.5:.95，分別為 74.57%、77.92%、84.70%、53.12%。

表 3 農作物及葉片偵測模型 LR 實驗結果

模型	精確率 (%)	召回率 (%)	mAP@.5 (%)	mAP@.5:.95 (%)
Model_3-1 (LR = 0.1)	30.00	48.64	33.69	15.90
Model_3-2 (LR = 0.01)	74.57	77.92	84.70	53.12
Model_3-3 (LR = 0.001)	39.50	52.66	41.78	18.01

註：LR：學習率 (Learning Rate)；mAP：Mean Average Precision。

資料來源：作者自製。

最終，此部分研究的該最佳模型 Model_3-2 作為建議 (Proposed) 模型，其於各類別的結果則如表 4，蘋果的 mAP@.5 為 83.2%，番茄的 mAP@.5 則為 86.2%。偵測混淆矩陣則如圖 15 所示，蘋果葉片檢測準確率為 83%，番茄為 94%。因此，此建議模型將作為這次研究於農作物疾病偵測系統中的農作物及葉片偵測模型，其於影像中的偵測範例如圖 16，模型能夠透過空拍的角度，有效偵測出影像中的葉片，並分類於所屬農作物品種。

表 4 建議模型 mAP@.5

模型	蘋果 (%)	番茄 (%)	All Classes (%)
建議	83.2	86.2	84.7

註：mAP：Mean Average Precision。

資料來源：作者自製。

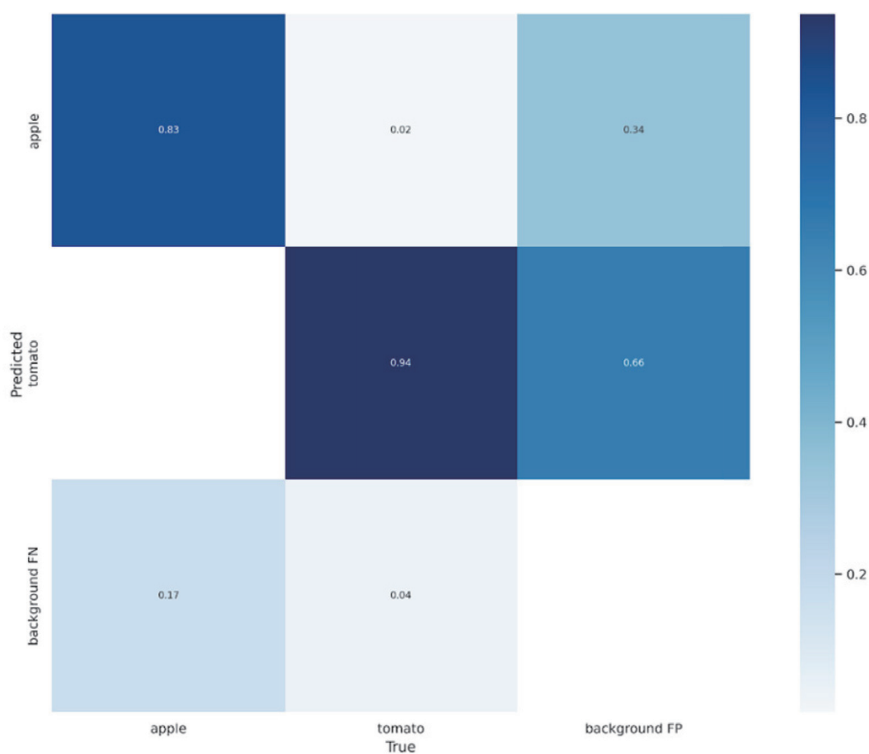


圖 15 建議模型檢測混淆矩陣

資料來源：作者自製。

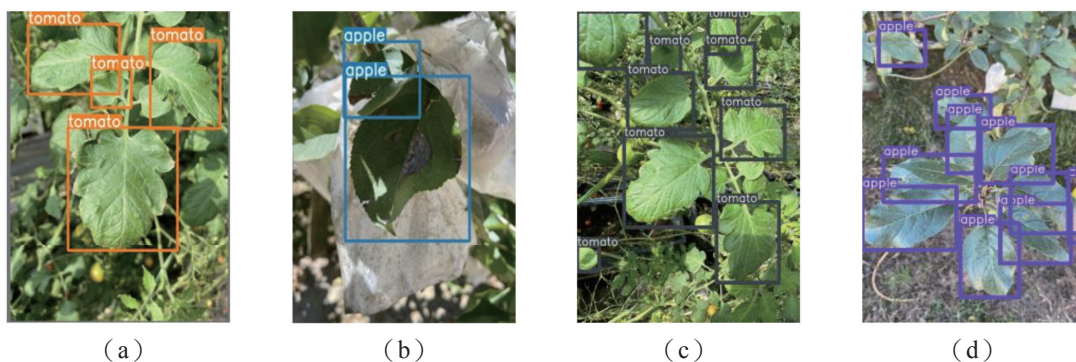


圖 16 農作物及葉片偵測模型偵測結果

資料來源：作者自製。

二、蘋果和番茄農作物疾病偵測

此部分實驗分別針對蘋果黑星病、番茄早疫病、晚疫病和葉黴病建立疾病檢測模型，並且比較 Vit-Base-16、Vit-Base-32、Vit-Large-16、Vit-Large-32 以及 Meta 所提出 ConvNext v2 (Woo et al., 2023) 的疾病檢測能力與速度。研究使用健康蘋果葉片 1,645 張、蘋果黑星病

葉片 630 張、健康番茄葉片 1,591 張、番茄早疫病葉片 1,000 張、番茄晚疫病葉片 1,909 張、番茄葉黴病葉片 952 張進行模型訓練與測試。

模型皆透過 ImageNet-22k 進行預訓練，因此已擁有基礎影像辨識能力。訓練環境使用 Google Colab，超參數的設定如表 5 所示，其他則使用模型預設數值。損失函數使用 Cross Entropy，如方程式 (11)，其中 t_i 為真實標籤， p_i 為模型輸出的結果。實驗中所使用到的資料擴增技術於描述如表 6 所示。

資料來源：作者自製

表 5 蘋果和番茄農作物疾病偵測模型超參數設置

優化器	LR	Batch Size	時期	Input Shape
Adam (Kingma and Ba, 2014)	0.00002	16	5	224×224

註：LR：學習率 (Learning Rate)。

資料來源：作者自製。

$$Loss = -\sum_{i=1}^n t_i \log(p_i), \text{ for } n \text{ classes} \quad (11)$$

表 6 農作物疾病偵測模型資料增強設定

資料增強技術	數值	數值意義
隨機縮放裁剪	Size = 224	隨機裁切並調整為 224×224
隨機翻轉	N/A	隨機水平或垂直翻轉
隨機仿射變換	Degrees = 15, Translate = (0.1, 0.1)	隨機進行仿射變換，包括旋轉最多 15 度， 平移最多 10%
顏色抖動	Brightness = 0.1, Contrast = 0.1, Saturation = 0.1, Hue = 0.1	隨機調整的亮度、對比度、飽和度和色調， 最多 10%

資料來源：作者自製。

首先為蘋果黑星病檢測的模型，實驗結果透過表 7 可以發現我們選用的影像分類模型於此任務中擁有相當好的檢測能力，能夠看到 Vit-Base-16、Vit-Large-16、Vit-Large-32、ConvNext v2 的檢測結果於準確率、召回率、精確率皆可達到 100%。平均單一影像處理速度表現上，Vit-Base-16、Vit-Base-32、ConvNext v2 皆於 0.0200 秒以內接下來觀察各模型於

檢測時間上的表現，得知 Vit-Base-16 平均檢測一張圖像僅花費 0.0137 秒。最終我們選擇 Vit-Base-16 作為農作物疾病偵測系統中的蘋果疾病檢測模型，因其在擁有最好檢測能力的同時保有相當快的檢測速度。

表 7 蘋果疾病偵測模型實驗結果

模型	準確率 (%)	召回率 (%)	精確率 (%)	時間 (s)
Vit-Base-16	100.00	100.00	100.00	0.0137
Vit-Base-32	99.78	99.19	100.00	0.0146
Vit-Large-16	100.00	100.00	100.00	0.0357
Vit-Large-32	100.00	100.00	100.00	0.0407
ConvNext v2	100.00	100.00	100.00	0.0181

資料來源：作者自製。

接下來針對番茄早疫病、晚疫病和葉黴病建立疾病檢測模型。最終的實驗結果如表 8 所呈現。在此實驗中可以發現所有模型都擁有相當好的檢測能力，於準確率、召回率、精確率皆能夠到達 98.00 ~ 99.00% 以上。於平均單一影像處理速度表現上，Vit-Base-16、Vit-Base-32、ConvNext v2 皆於 0.0200 秒以內。於是在最終模型的選擇上，同樣選擇了 Vit-Base-16 作為系統中所使用到的檢測模型，因其不僅擁有好的檢測能力，同時有相當好的檢測速度使系統可以獲得好的性能。

表 8 番茄疾病偵測模型實驗結果

模型	準確率 (%)	召回率 (%)	精確率 (%)	時間 (s)
Vit-Base-16	99.73	99.68	99.75	0.0139
Vit-Base-32	98.99	98.80	99.32	0.0142
Vit-Large-16	99.54	99.35	99.69	0.0348
Vit-Large-32	99.73	99.80	99.67	0.0384
ConvNext v2	99.54	99.54	99.41	0.0183

資料來源：作者自製。

綜上所述，實驗結果中，我們可以發現 Vit-Base-16、Vit-Base-32、Vit-Large-16、Vit-Large-32、ConvNext v2 的預訓練模型於蘋果和番茄的葉片疾病檢測上皆有相當優異的表現。在同時考慮了檢測準確度和檢測速度後，我們皆選擇了 Vit-Base-16 作為整體系統中疾病檢測的模組。

接下來，透過觀察圖 17，首先，我們可以發現研究提出的模型於番茄疾病偵測中，正確預測的模型輸出普遍為 0.30 ~ 0.40 間。另外，蘋果疾病預測中，正確預測的模型輸出則普遍於 0.55 ~ 0.75 之間。透過以上分析，可將其作為實際應用時間閾值設定的依據，用於區別病徵不清以無法判斷之植株。例如，可以設定蘋果疾病偵測的閾值為 0.55，即當模型輸出大

於 0.55 時，判定該植株健康狀態為模型預測結果。同時，設定輸出 0.50 ~ 0.55 的葉片為無法有效判斷，即需要人工檢測。這樣的設定可以減少模型的假陰性和假陽性，提高模型的準確性和可靠性。

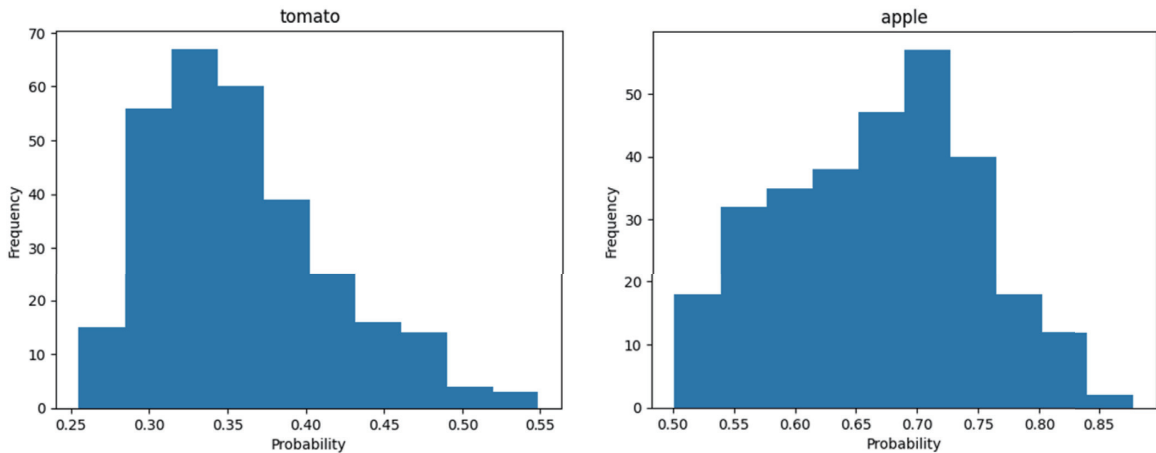


圖 17 農作物疾病偵測模型測試集正確輸出之機率分布

資料來源：作者自製。

三、系統整合

最終，我們將農作物及葉片偵測模型、農作物疾病偵測模型進行整合，建立完整系統，提出農作物疾病偵測系統。透過此農作物疾病偵測系統，能夠利用植株影像，檢測葉片上的病癥，並判斷植株是否感染疾病。

系統檢測流程與結果以圖 18 為例，首先，系統輸入植株影像，藉由農作物種類及葉片偵測模型（YOLOv7）偵測模型檢測出葉片物件並將其分類。接下來，系統會透過葉片擷取與去除影像背景模組進行影像處理。最終，葉片所屬的農作物疾病偵測模型會針對各個葉片影像進行疾病的檢測，於結果輸出該葉片是否遭到疾病危害。

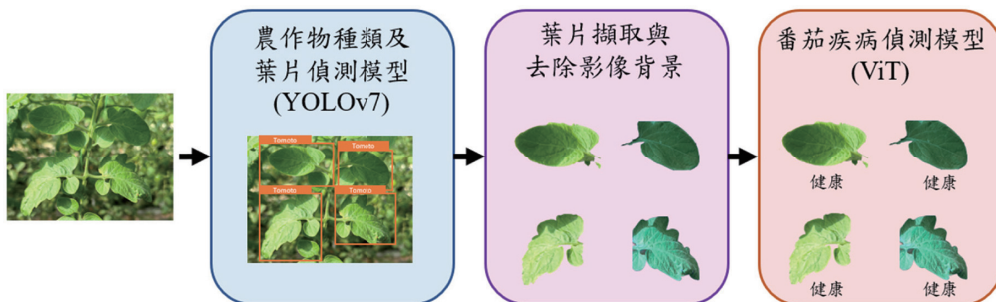


圖 18 農作物疾病偵測系統檢測範例

資料來源：作者自製。

伍、結論與未來研究方向

於本研究中，提出一套結合影像辨識與深度學習技術的農作物疾病偵測系統，研究優勢在於可以應用於多種農作物和疾病的偵測，並且可以透過空拍機提升監控效率和範圍。系統中的重要模組於實驗後選擇 YOLOv7 作為農作物種類與葉片偵測模型，ViT-Base-16 作為蘋果與番茄疾病偵測模型，藉由以上兩階段式的疾病偵測，加強對於病徵的小物件檢測能力。最終，此農作物疾病偵測系統可應用於蘋果黑星病和番茄早疫病、晚疫病、葉黴病的偵測。透過實驗，我們可以得知系統能夠有效檢測農作物疾病，以協助農民提升監控農作物疾病的效率並降低被疾病影響而受到的損失。

於未來研究部分，本研究認為有以下幾點為主要發展方向，將可以加強研究提出的農作物疾病偵測系統性能：

- 一、收集更多農作物影像資料，進一步提升系統性能，同時可擴大可應用的作物種類與疾病類型。
- 二、將系統實際部署到無人機平臺上，在實際環境中進行測試和優化。
- 三、增加植株定位系統，結合地理資訊系統，實現植株精準定位與個體辨識，加強整體系統可應用性。

參考文獻

- 行政院農業委員會農糧署，2020，〈108 年臺灣地區農家戶口抽樣調查報告〉，《農糧署全球資訊網》，<https://www.afa.gov.tw/cht/index.php?code=list&ids=1042>（瀏覽日期：2022 年 8 月 15 日）。
- 農業部，n.d.，〈農漁生產地圖〉，《農業知識入口網》，https://kmweb.moa.gov.tw/theme_list.php?theme=production_map（瀏覽日期：2022 年 8 月 15 日）。
- 農業部農業藥物試驗所，n.d.，〈作物資料〉，《植物保護資訊系統》，<https://otserv2.acri.gov.tw/ppm/PLC02.aspx>（瀏覽日期：2022 年 8 月 15 日）。
- Bochkovskiy A., Wang C.-Y., and Liao H.-Y. M., 2020, “YOLOV4: Optimal Speed and Accuracy of Object Detection,” <https://doi.org/10.48550/arXiv.2004.10934> (accessed January 11, 2024).
- Devlin J., Chang M.-W., Lee K., and Toutanova K., 2018, “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” <https://doi.org/10.48550/arXiv.1810.04805> (accessed January 11, 2024).
- Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., and Houlsby N., 2020, “An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale,” <https://doi.org/10.48550/arXiv.2010.11929> (accessed January 11, 2024).
- Du L., Zhang R., and Wang X., 2020, “Overview of Two-Stage Object Detection Algorithms,”

- Journal of Physics: Conference Series*, 1544, 012033. doi:10.1088/1742-6596/1544/1/012033
- Gatis D., 2020, “Rembg,” *GitHub*, <https://github.com/danielgatis/rembg> (accessed January 11, 2024).
- Girshick R., Donahue J., Darrell T., and Malik J., 2014, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 580-587. doi:10.1109/CVPR.2014.81
- He K., Zhang X., Ren S., and Sun J., 2016, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 770-778. doi:10.1109/CVPR.2016.90
- Huang G., Liu Z., Van Der Maaten L., and Weinberger K. Q., 2017, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 4700-4708. doi:10.1109/CVPR.2017.243
- Hughes D. P., and Salathe M., 2015, “An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics,” <https://doi.org/10.48550/arXiv.1511.08060> (accessed January 11, 2024).
- Kingma D. P., and Ba J., 2014, “Adam: A Method for Stochastic Optimization,” <https://doi.org/10.48550/arXiv.1412.6980> (accessed January 11, 2024).
- Redmon J., Divvala S., Girshick R., and Farhadi A., 2016, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 779-788. doi:10.1109/CVPR.2016.91
- Redmon J., and Farhadi A., 2017, “YOLO9000: Better, Faster, Stronger,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 6517-6525. doi:10.1109/CVPR.2017.690
- Redmon J., and Farhadi A., 2018, “YOLOv3: An Incremental Improvement,” <https://doi.org/10.48550/arXiv.1804.02767> (accessed January 11, 2024).
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A., 2015, “Going Deeper with Convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 1-9. doi:10.1109/CVPR.2015.7298594
- Tzutalin, 2015, “LabelImg,” *GitHub*, <https://github.com/tzutalin/labelImg> (accessed January 11, 2024).
- Wang C.-Y., Bochkovskiy A., and Liao H.-Y. M., 2023, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” in *2023 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA: IEEE Computer Society Press, 7464-7475. doi:10.1109/CVPR52729.2023.00721

Woo S., Debnath S., Hu R., Chen X., Liu Z., Kweon I. S., and Xie S., 2023, “Convnext V2: Co-Designing and Scaling Convnets with Masked Autoencoders,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA: IEEE Computer Society Press, 16133-16142. doi:10.1109/CVPR52729.2023.01548