**農地作物現況調查影像辨識競賽 – 秋季賽:**

**AI作物影像判釋**

**參賽組別名稱: TEAM\_2035**

1. **環境**

本次競賽中，我們在Windows環境上建置訓練環境，深度學習框架大部分是使用Pytorch [1]、Torchvision [2] 以及Scikit-learn [3]套件來訓練深度學習模型。我們會使用Pytorch 是因為它比起Tensorflow [4]來的更有彈性，且在操作上更接近於Python的語法。 Pytorch 被用於當今的許多深度學習項目，它在深度學習領域中的受歡迎程度正在增加，是一個非常適合拿來做為機器視覺模型訓練的工具。

另外，因為我們有使用到半監督式學習，Public-Test與Private-Test作為我們額外的訓練資料，可以用來提升模型的訓練成果。更多詳細的規格如表1所示:

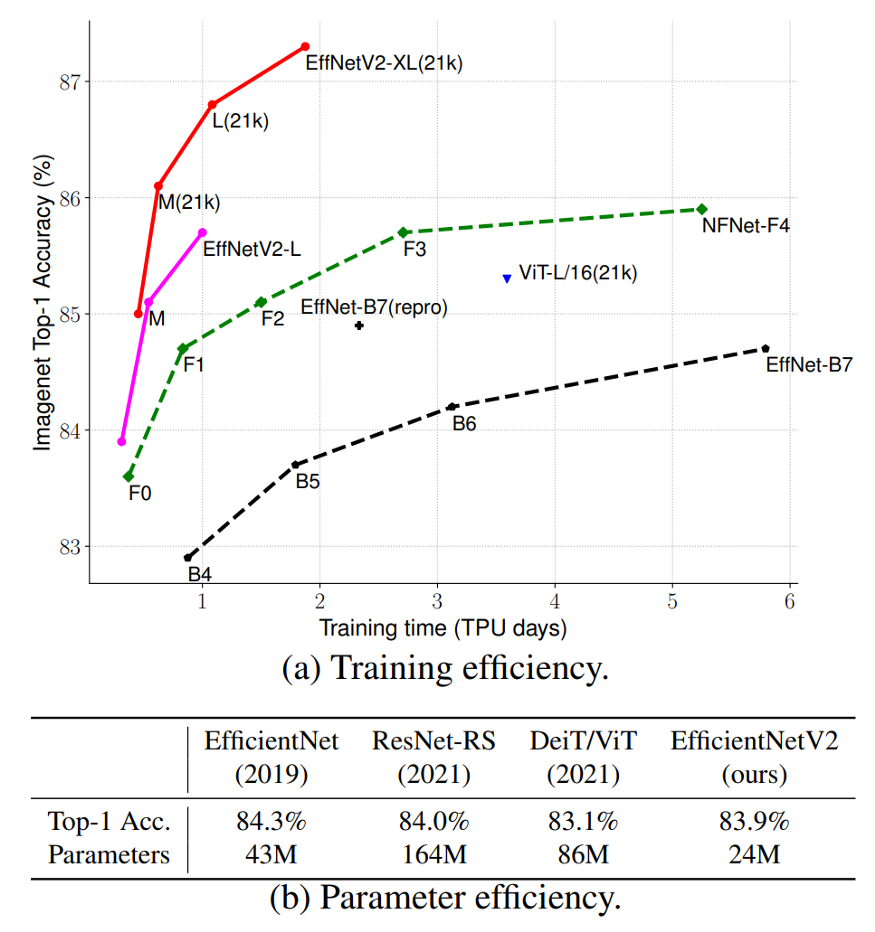
表1、訓練環境規格

|  |  |
| --- | --- |
| **作業系統** | Windows |
| **顯示卡** | 2張NVIDIA TITAN RTX |
| **處理器** | Intel(R) Core (TM) i9-9820X CPU |
| **記憶體** | 128GB |
| **程式語言** | Python |
| **主要函式庫 (詳見Github)** | Pytorch |
| **預訓練模型** | ImageNet-1K [15] |
| **額外資料集** | Public-Test 和 Private-Test |

1. **演算方法與模型架構**

近年來，深度學習的網路架構被廣泛的應用在計算機視覺(Computer Vision)領域上，相關的研究被循序的提出來，其中在圖像分類(Image Classification)的任務上獲得了很大的進展。

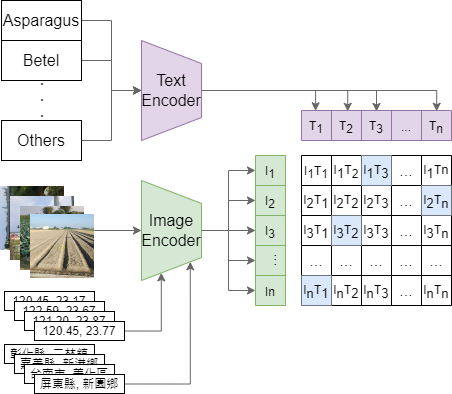
EfficientNet [4]作為現今最強大的卷積神經網路(Convolutional Neural Network)骨幹網路(Backbone)之一，它在圖像分類任務中作為特徵萃取器(Feature Extractor)一直佔有一席之地，而隨著今年推出的EfficientNetV2 [5]，該網路結構更進一步地提高了準確性、減少了訓練時間和預測延遲。透過圖 1我們可以觀察到EfficientNetV2僅含有2400萬個參數，相比於Vision Transformer [6]的8600萬個參數少上許多，且它比起EfficientNet少了近一半的參數量，但在準確度上卻保有相似或更高的準確度。(EfficientNetV2-S參數量為21.5M，EfficientNetV2-M參數量為54.1M)



1. 、EfficientNetV2 Benchmark 比較 [5]

在圖像分類模型架構上我們參考了OpenAI推出的神經網路CLIP (Connecting Text and Images) [7]，該架構的核心概念有別於傳統將維度降至類別數量再計算交叉熵損失函數(Cross Entropy Loss)的分類方法，在維度上CLIP會把文字與圖像都映射到一個相對較高維的空間，接著再算相似度求交叉熵損失函數。這樣的作法不僅保留更多的圖像特徵資訊，同時也能夠增加模型的泛化能力(Generalization)。CLIP架構在訓練時會產生兩個encoder，text encoder用來把每一段文字或標籤(label)映射到空間中的某一個點，而另一個image encoder則用來映射每一張圖片。在訓練過程中一次拿一個batch共n組相對應的標籤跟圖片，標籤部分丟給text encoder轉出T1,T2,T3…Tn，而圖片部分丟給image encoder轉成I1,I2,I3,…In。成對的圖片跟標籤內積起來要越大越好，在空間中映射的點也會較相近。

在CLIP的架構之上我們做了些微的調整，在image encoder部分我們使用EfficientNetV2骨幹網路來抓圖像的特徵，該網路所產生的高維度向量(1280維度)再與GPS精準座標(128維度)和地理位置名稱(128維度)合併在一起，向上投影到2048維度的向量空間。最後label會經由text encoder產生高維度向量，同樣投影至向量空間(2048維度)與image encoder的向量倆倆算相似度計算交叉熵損失函數，如圖 2。

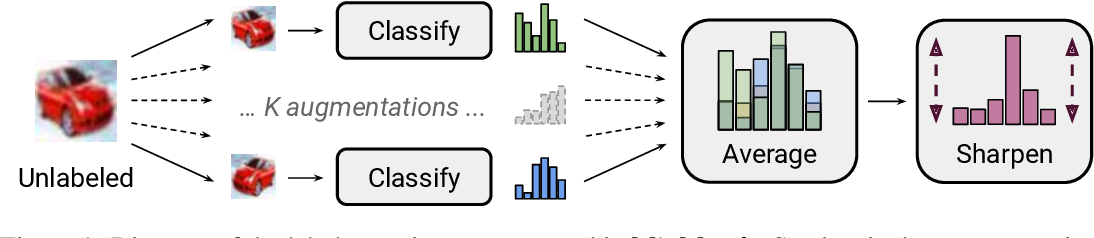


1. 、本研究所提之模型架構

傳統上在訓練圖像分類的模型可以分為監督式學習(Supervised Learning)與非監督式學習(Unsupervised Learning)兩種不同的方式。監督式學習的訓練方法仰賴人工的標記，需要在訓練和測試模型之前由相關人士標記好圖像的標籤，而隨後模型再經由這些標記好的訓練資料來學習並用於預測未來事件；相反地，非監督式學習的訓練方法不需要人工標記，該方法純粹依靠電腦的演算法來探索資料之間彼此的特徵，並從資料中推論出一個可以描述這些未標記資料的函數來分類資料。

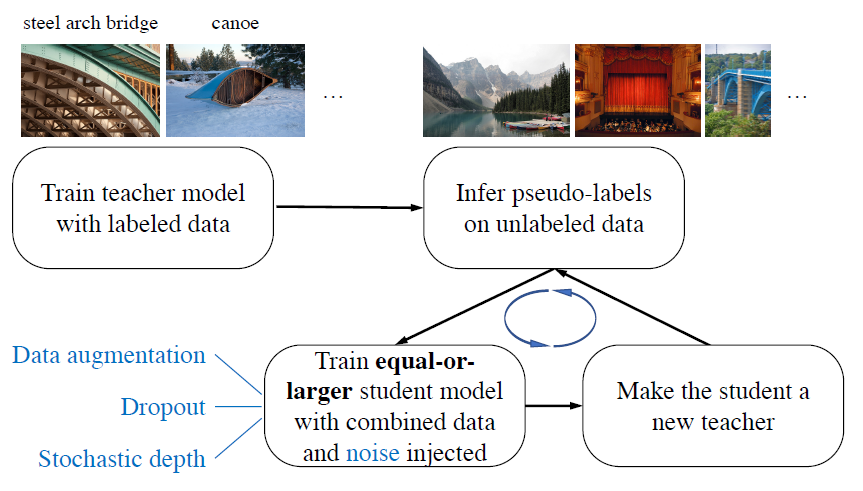
由於現實情況中訓練資料的標記是一項耗費心力且流程繁雜的任務，因此只運用部分有標記與剩餘未標記資料來訓練的半監督式學習(Semi-supervised Learning)開始逐漸盛行。半監督式學習的想法是希望先利用有標記的圖像資料來訓練一個初步的分類模型，接著再使用該模型來標記未標記的圖像，並將這些帶有高置信度(High Confidence)標籤的圖像作為偽標籤(Pseudo Label)來繼續訓練與優化分類模型。

以往半監督式學習製作偽標籤的方法，多半表現不理想。比如使用主成分分析(Principal Component Analysis, PCA)提取特徵，再以此標記未標記資料的作法；又比如使用自編碼機(AutoEncoder)或對抗生成網路(Generative Adversarial Network, GAN [27])生成以假亂真的偽標籤圖像的作法，這些方法不但在訓練時間上很久，且準確度都比不上監督式學習。然而自從2019年發表的MixMatch [8]，半監督式學習終於突破了困境，並刷新了大家的認知。MixMatch首先對沒有標記的資料，做資料增強，得到N筆新的增強影像，接著將這N筆影像資料輸入至同一個分類器，得到不同的預測分布(Softmax Distribution)，將其加總平均，最後再銳化這個分布結果。這樣一來就能夠迫使模型對於未標記的資料作出低熵(Minmize Entropy)的預測，產生偽標籤，如圖 3。這種方法不僅能夠讓模型從極少量的標記訓練資料與大量未標記資料中學習，也增加其對陌生資料的泛化能力，使其表現高於一般監督式學習的方法。



1. 、MixMatch所提之半監督式學習方法 [8]

發表自CVPR 2020的Noisy Student [9]半監督式學習模型更進一步地推進ImageNet分數新高點。Noisy Student的整體思路相當簡單，首先會先對ImageNet訓練一個teacher model，接著應用self-training的技巧不斷大量產生偽標籤，並重新以偽標籤資料與訓練資料訓練一個同等規模的student model，每一輪的student model在下一輪都會變為新的teacher model，如此重複迭代，如圖 4。Student model會在訓練的過程中加入各式各樣的資料增強(Data Augmentation)與雜訊(Noise)，幫助student model變得更加魯棒性(robust)。論文中也以實驗證明在資料量足夠的情況下做self-training效果會比pre-training還要好，另外雜訊的加入也相當重要。



1. 、Noisy Student所提之半監督式學習方法 [9]

綜觀以上，為了在本次農作物影像辨識的任務上充分利用有限的資源，我們最終將那些準心偏移與沒有提供標記的資料一併納入訓練。在模型上採用MixMatch與Noisy Student這兩種半監督式學習的模型輔助判斷，並以集成式學習(Ensemble Learning) 將所有模型產出的結果加權平均。希望藉此結合眾多模型來產生一個更強大、綜合表現更穩定的模型。其中集成式學習部分，包含了一個監督式學習(2個結果)與兩個半監督式學習(6個結果) 一共產出8個結果，最後加權平均輸出。

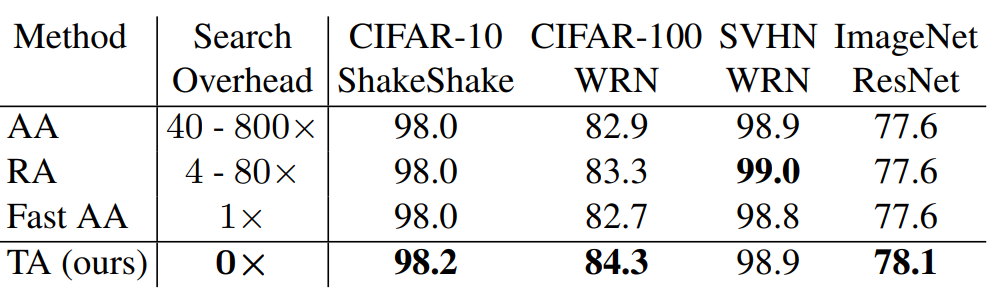
1. **資料處理**

收集和標記資料非常耗時且昂貴，因此對訓練資料做適當的調整與增強，一直是深度學習重要的課題之一；此外，圖像數據增強可用於提高模型性能並減少泛化誤差。由於主辦方所提供的準心，部分資料有偏移，也因此在本次競賽中每張影像在輸入模型訓練之前，會先依照其短邊對齊成一個方框，並向中心點位移，縮放成的大小正方形影像，如圖 5。



1. 、圖片處理示意圖

於訓練階段，影像會進入TrivialAugment [10] 資料增強算法進行增強。該增強算法不同於早期的自動搜索算法如AutoAugment [11]與RandAugment [12]，不需要額外在資料集上學習最佳增強策略，是一種簡單有效的資料增強算法。RandAugment算法簡單但仍然有兩個超參數(Hyperparameter)需要搜索，包含增強方法總數N與形變程度M。TrivialAugment進一步簡化了RandAugment，對於每一張圖片TrivialAugment只會隨機選擇一種增強方法，然後以選定的強度實作它，這意味著我們完全省去掉搜索的時間。從圖 6中我們也可以看到，該增強算法在使用極少運算資源的條件下表現大致優於AutoAugment (AA)與RandAugment (RA)。



1. 、TrivialAugment Benchmark [10]

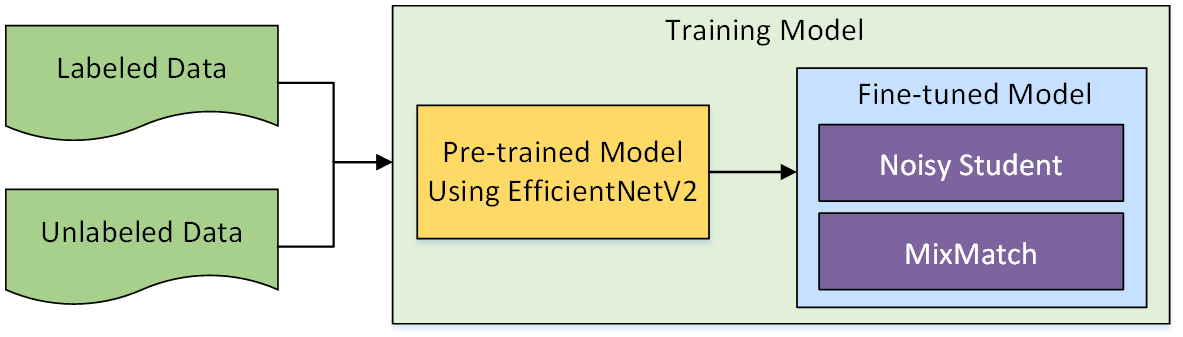
於推論階段，我們使用Test Time Augmentation(TTA)資料增強策略，它會對一張影像作多種選自於訓練階段的增強方法，而模型會對這些增強過的影像進行預測，最後進行加權平均，輸出預測結果。

針對不同的模型，我們也有單獨實作幾個在深度學習領域中經典的、易於實現的圖像數據增強方法，例如雜訊 (Noise: *gaussian, erasing*)、標籤平滑 (Label Smoothing)、樣本混和(Mixup [13]、Cutmix [14])等用以增強數據。這些增強方法已在ImageNet [15] 視覺識別挑戰賽(ILSVRC)上得名的多篇論文研究中諸如AlexNet [16]、GoogLeNet [17]、VGG [18]、ResNet [19]、EfficientNet [4]以及這次競賽我們使用的幾個非監督式學習模型(Mixmatch [8]與Noisy Student [9])上被證明可用來提高圖像數據的性能。

最後，在非作物類的資料部分，由於非作物類的圖像數量比一般作物類的圖像少非常多，我們特別將非作物類的圖像隨機拆分成兩張圖片，藉此增加資料量。

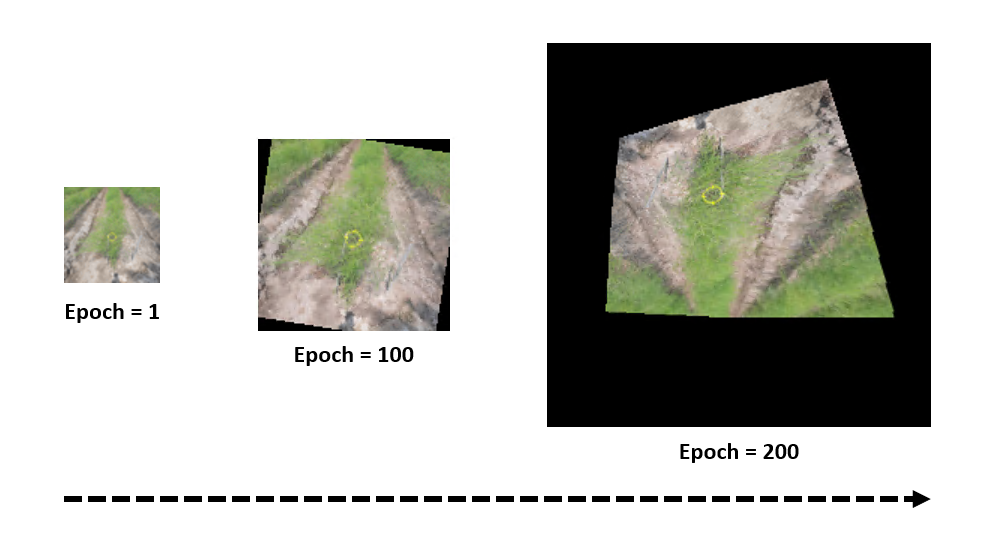
1. **訓練與推論方式**

本研究的訓練與推論流程大致可以分為三個階段如圖 7，我們將依序說明每一個步驟，並簡述相關訓練參數的設定。



1. 、訓練流程

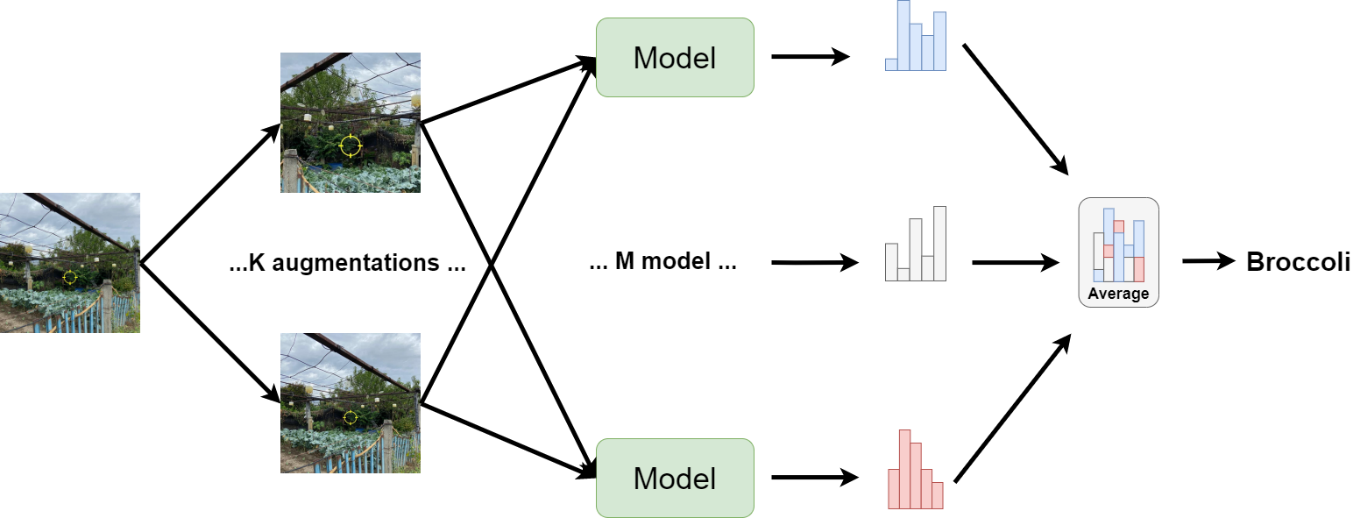
第一階段，我們會先以監督式學習的方式訓練我們的Teacher model，其中Teacher model的骨幹網路為EfficientNetV2-S，且採漸進式的影像大小來訓練(Progressive Learning [5])，如圖 8。因為當圖像尺寸較小時，增強較弱訓練出來的準確度較好；但相反地較大的圖像，必須在更強的增強下表現才會更好，也因此在訓練過程中順應地調整資料增強的強度和圖像大小有助於模型的表現。(此階段會產生1個模型)



1. 、漸進式學習 [5]

第二階段，我們會用第一階段所訓練出的Teacher model對有標籤的資料產生軟標籤(Soft Label)、對沒有標籤的資料產生偽標籤(Pseudo Label)，藉此訓練Student model，經過重複跌代過後我們最終可以得到Student model。於此階段我們也會一同訓練MixMatch模型，MixMatch會使用到第一階段Teacher model的預訓練模型來調權重，為了縮短訓練時間我們限定模型一次只產生2張增強影像()，並對他們的分布(Softmax Distribution)取平均再做銳化處理(Sharpening)，藉此產生偽標籤以便我們重新訓練新模型，如圖7。另外，我們會為每一個模型套用指數移動平均(Exponential Moving Average, EMA [20])，這個方法能夠有效地防止訓練的過度擬合(Overfitting)，它會參考前幾個epoch階段的模型權重做移動平均，以此提高測試指標並增加模型魯棒性。最終我們會得到有套用EMA與沒有套用EMA一共8個模型。

第三階段，我們會將8個模型所集成的結果作平均。每個模型在預測時會進行Test Time Augmentaion，對兩張圖像作預測。一張影像會直接做縮放()，而另一張會先center crop成 再縮放到 (目的是為了讓訓練與推論階段的目標大小一致，詳見FixRes [21])，兩張影像會作加權平均，如圖 9。



1. 、推論流程

在訓練時，我們將資料切分成訓練資料與驗證資料比例為9:1。監督式學習與半監督式學習的訓練參數，優化器是 SGD [22]；權重衰減(weight decay)為0.0001；訓練計劃總共為200個epoch，前100個epoch影像大小 batch是512張圖像，後100個epoch影像大小 batch是256張圖像；所有模型都使用2個GPU進行訓練；初始學習率(learning rate)設置為 ；在學習率調度器(learning rate scheduler)上，我們使用餘弦衰減(cosine decay)，並在前5個iteration進行warm-up。模型骨幹使用在ImageNet-1K [15] 上預訓練的EfficientNetV2-S和EfficientNetV2-M模型權重作為pretrain model進一步初始化，該預訓練權重由Pytorch官方提供。

1. **分析與結論**

在本節當中，我們將從模型架構以及訓練方式兩個角度切入來分析模型的成效，提供預測的結果，並簡述未來可能改進的方向。

1. **不同模型架構**

我們一共有嘗試4種模型架構，每個模型皆以EfficientNetV2作為骨幹網路。Base Model為EfficientNetV2接上簡易的分類器，CLIP架構則將EfficientNetV2套入Image Encoder與Text Encoder算相似度。另外，我們也有分別測試GPS座標位置與Location地理位置對於模型成效的影像。從表4中，我們可以發現有加入GPS座標位置與Location地理位置一同訓練的模型擁有較優異的表現，也代表說相同類別的作物會比較大機率出現在鄰近的地理位置。

表4、不同模型架構的效能比較

|  |  |  |
| --- | --- | --- |
| Model | *Acc*. (%) | *Loss* |
| Base model | 85.23 | 1.07 |
| CLIP framework | 86.72 | 1.03 |
| CLIP framework + GPS | 87.49 | **1.00** |
| CLIP framework + GPS + Location | **87.71** | 1.01 |

註: 評估方式採用驗證集的數據

1. **不同訓練方式**

包含監督式學習、半監督式學習與集成式學習，我們一共有3種訓練方式。其中每種訓練方式按照不同的參數設定、資料增強以及訓練方式又可以延伸出更多的變形，其中我們有五種模型組合。在表5中，我們可以發現集成式學習的模型表現最好，在集成學習之前Noisy Student中的Student Model表現最好。

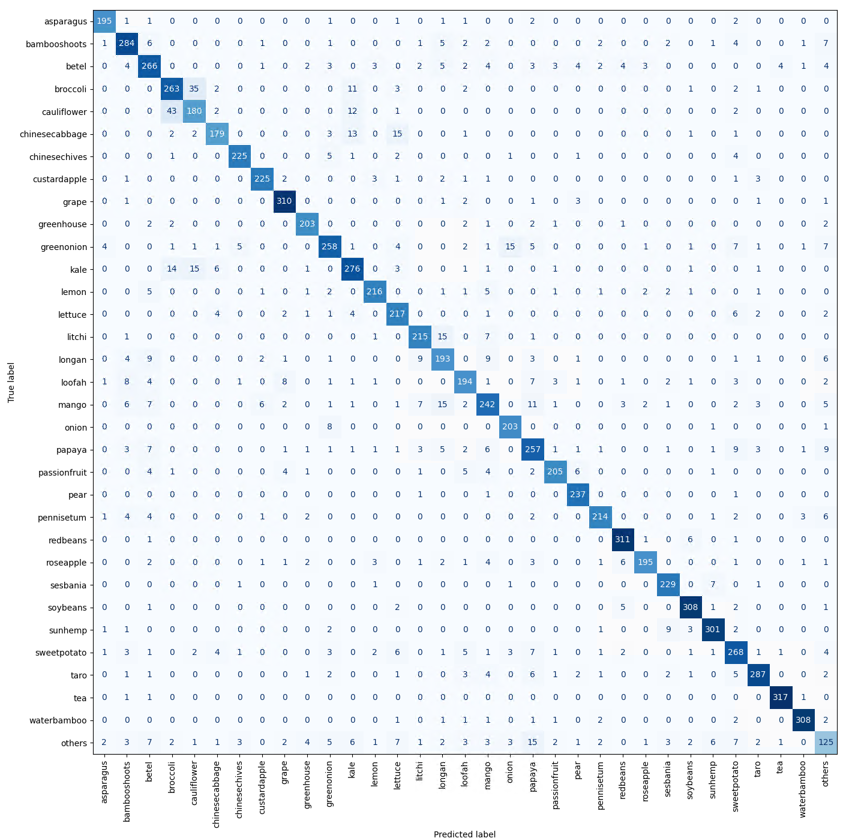
表5、不同訓練方式的效能比較

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Epoch | (%) | (%) | Extra training Data |
| Teacher Model (V2S) | 100 | 88.68 | 88.08 | - |
| Student Model (V2S) | 100 | **88.80** | **88.44** | Public-test |
| Student Model (V2M) | 80 | 88.17 | **87.16** | Public-test & Private-test |
| MixMatch (V2S) | 100 | 85.90 | 86.81 | Public-test |
| Ensemble Learning | - | **89.14** | - | - |

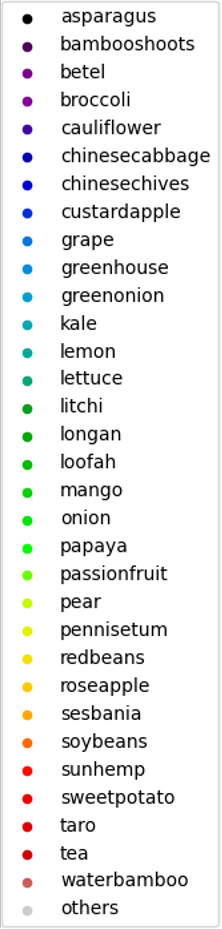
註: 評估方式採用驗證集的數據，EfficientNetV2-S (V2S)，EfficientNetV2-M (V2M)

1. **結果分析**
2. **分類結果**

從混淆矩陣(Confusion Matrix)與特徵分布圖(TSNE feature distribution)中如圖 10、圖 11，我們可以觀察到broccoli類與cauliflower類的影像特徵非常相似，造成兩個類別的Recall與Precision相對較低。另外，tea、redbean與grape三類的作物影像特徵差異很大，因此在分類結果上比較好。



1. 、混淆矩陣





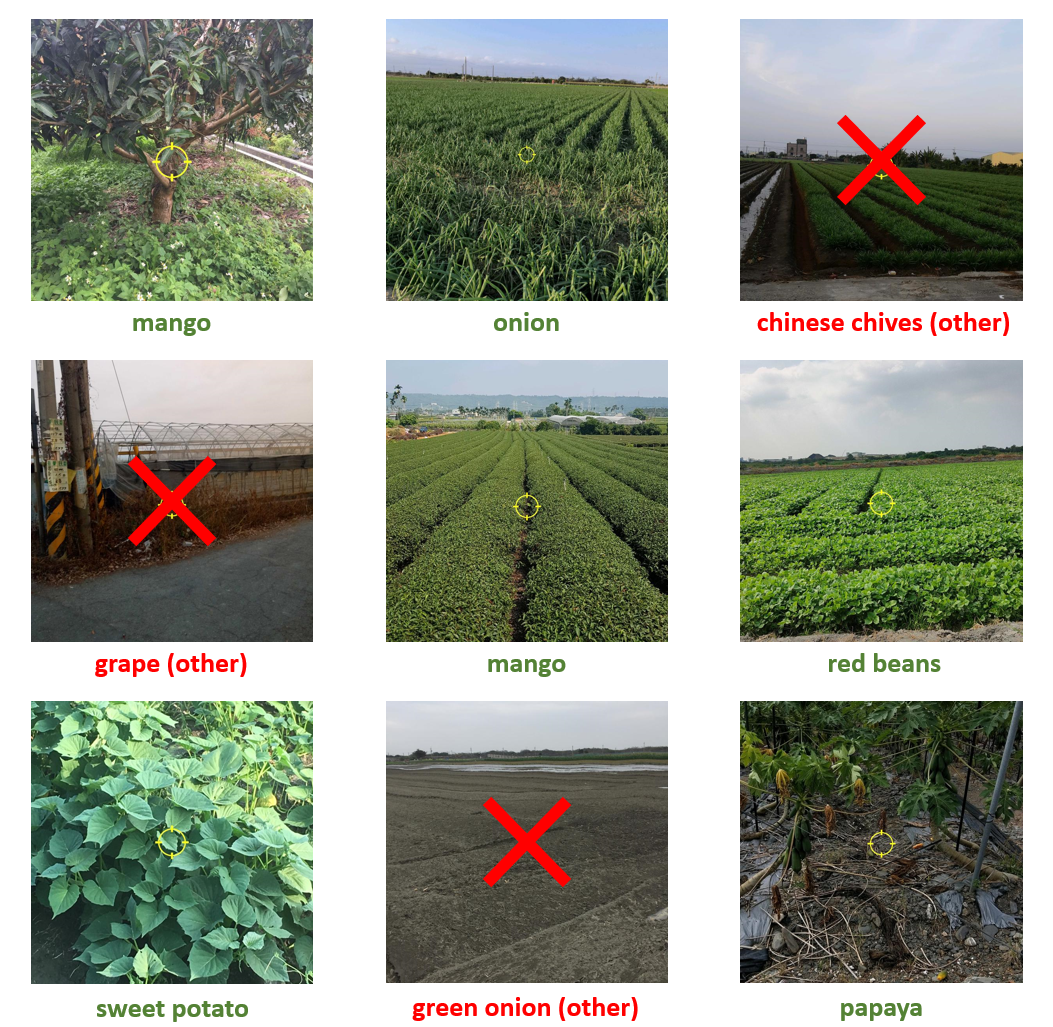
1. 、TSNE特徵分布圖 [23]

表6、各類別的預測結果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-score** | **Samples** |
| asparagus | 0.97 | 0.95 | 0.96 | 205 |
| bambooshoots | 0.86 | 0.88 | 0.87 | 320 |
| betel | 0.79 | 0.86 | 0.82 | 320 |
| broccoli | 0.81 | 0.83 | 0.82 | 320 |
| cauliflower | 0.77 | 0.77 | 0.77 | 240 |
| chinesecabbage | 0.89 | 0.84 | 0.86 | 217 |
| chinesechives | 0.97 | 0.93 | 0.95 | 240 |
| custardapple | 0.92 | 0.95 | 0.93 | 240 |
| grape | 0.95 | 0.96 | 0.96 | 320 |
| greenhouse | 0.95 | 0.93 | 0.94 | 216 |
| greenonion | 0.87 | 0.84 | 0.85 | 316 |
| kale | 0.89 | 0.86 | 0.87 | 320 |
| lemon | 0.95 | 0.89 | 0.92 | 240 |
| lettuce | 0.87 | 0.89 | 0.88 | 240 |
| litchi | 0.9 | 0.9 | 0.9 | 240 |
| longan | 0.81 | 0.82 | 0.82 | 240 |
| loofah | 0.82 | 0.81 | 0.82 | 240 |
| mango | 0.82 | 0.81 | 0.82 | 318 |
| onion | 0.89 | 0.96 | 0.92 | 213 |
| papaya | 0.81 | 0.82 | 0.81 | 316 |
| passionfruit | 0.94 | 0.88 | 0.91 | 234 |
| pear | 0.91 | 0.98 | 0.95 | 240 |
| pennisetum | 0.92 | 0.91 | 0.92 | 240 |
| redbeans | 0.95 | 0.98 | 0.96 | 320 |
| roseapple | 0.94 | 0.92 | 0.93 | 225 |
| sesbania | 0.93 | 0.94 | 0.94 | 240 |
| soybeans | 0.95 | 0.97 | 0.96 | 320 |
| sunhemp | 0.93 | 0.96 | 0.94 | 320 |
| sweetpotato | 0.81 | 0.85 | 0.83 | 320 |
| taro | 0.93 | 0.91 | 0.92 | 320 |
| tea | 0.98 | 0.99 | 0.99 | 320 |
| waterbamboo | 0.97 | 0.97 | 0.97 | 320 |
| others | 0.73 | 0.63 | 0.68 | 223 |
| **Accuracy** | - | - | 0.89 | 8963 |
| **Macro Avg** | 0.89 | 0.89 | 0.89 | 8963 |
| **Weighted Avg** | 0.89 | 0.89 | 0.89 | 8963 |

1. **非作物分類**

由於一般作物的影像也經常含有非作物相關的物體，使得非作物類的資料沒有足夠的代表性，導致模型容易將非作物的影像誤判為一般作物，如圖 12可以看出，許多被判成非作物的作物類影像，都與作物影像非常相似。另外，中間第二排最左邊的非作物影像被判成grape，與grape資料本身時常有格網出現的情形有關。



1. 、分類結果
2. **結論與未來改進方向**
3. **綜觀以上我們可以得出以下結論:**

* 對訓練資料做適當的調整與增強，可用於提高模型性能並減少泛化誤差。
* 使用集成式學習能夠結合眾多模型的優缺點，產生一個更強大、綜合表現更穩定的模型。
* 在CLIP的模型架構之上加入GPS座標位置與Location地理位置能夠有效提升準確率。

1. **未來改進方向:**

資料增強策略與超參數調整還有很多可以嘗試的空間，比如2021年發表的TransMix [24] 有望提高模型的準確率與魯棒性。圖像分類的骨幹網路也有很多沒有嘗試過，比如Transformer體系的分類模型Swin-Transformer [25]、CoAtNet [26]..等。另外，假如未來引進異物偵測算法(Out-of-Distribution Detection, OOD)，在模型推論之前將非作物類的影像偵測出來並加以排除，模型效能有望再增長。

1. **程式碼 (詳見** [**https://github.com/jeffzhux/AICUP\_fall**](https://github.com/jeffzhux/AICUP_fall) **)**
2. **Install**

git clone https://github.com/jeffzhux/AICUP\_fall.git

cd AICUP\_FALL

pip install -r requirements.txt

1. **Prepare Data**

└── data

├── train

│ ├── asparagus

│ │ ├── 000b43a3-d331-47ad-99a4-4c0fa9b48298.jpg

│ │ ├── 00a326ba-7e9a-4bc1-8684-4bf404bfd6bb.jpg

│ │ ├── ...

│ │ └── 0a550bc0-9b1f-4fa8-81cc-9b518af29c89.jpg

│ ├── bambooshoots

│ ├── ...

│ └── waterbamboo

├── valid

│ ├── asparagus

│ ├── bambooshoots

│ ├── ...

│ └── waterbamboo

└── tag\_locCoor.csv

1. **Train**

* Train Teacher Model (Supervised Learning)

python train\_noiseStudent.py ./config/noiseStudent\_config.py

* Train Student Model (Semi-Supervised Learning with Noise Student)

python train\_noiseStudent.py ./config/noiseStudent\_config.py

* Train Model (Semi-Supervised Learning with MixMatch)

python train\_mixmatch.py ./config/mixmatch\_config.py

1. **Eval (Evaluating single model performance)**

python test\_sim.py ./config/test\_sim\_config.py

1. **參考文獻**
2. Pytorch. Contributors , from https://pytorch.org/
3. torchvision Contributors: Torchvision main documentation. (n.d.). <https://pytorch.org/vision/stable/index.html>
4. scikit-learn: machine learning in Python — scikit-learn 1.2.0 documentation. (n.d.). <https://scikit-learn.org/stable/>
5. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning, 6105–6114.
6. Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. Cornell University - ArXiv. <https://doi.org/10.48550/arxiv.2104.00298>
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
8. Li, M., Xu, R., Wang, S., Zhou, L., Lin, X., Zhu, C., ... & Chang, S. F. (2022). Clip-event: Connecting text and images with event structures. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16420-16429).
9. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.
10. Xie, Q., Luong, M. T., Hovy, E., & Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10687-10698).
11. Müller, S. G., & Hutter, F. (2021). Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 774-782).
12. Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 113-123).
13. Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 702-703).
14. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.
15. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6023-6032).
16. ImageNet. (n.d.). <https://www.image-net.org/>
17. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.
18. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
19. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
20. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
21. Cai, Z., Ravichandran, A., Maji, S., Fowlkes, C., Tu, Z., & Soatto, S. (2021). Exponential moving average normalization for self-supervised and semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 194-203).
22. Touvron, H., Vedaldi, A., Douze, M., & Jégou, H. (2019). Fixing the train-test resolution discrepancy. Advances in neural information processing systems, 32.
23. SGD Contributor: PyTorch 1.13 documentation. (n.d.). <https://pytorch.org/docs/stable/generated/torch.optim.SGD.html>
24. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).
25. Chen, J. N., Sun, S., He, J., Torr, P. H., Yuille, A., & Bai, S. (2022). Transmix: Attend to mix for vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12135-12144).
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10012-10022).
27. Dai, Z., Liu, H., Le, Q. V., & Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. Advances in Neural Information Processing Systems, 34, 3965-3977.
28. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

**作者聯絡資料表**

* **隊伍**

|  |  |  |
| --- | --- | --- |
| 隊伍名稱 | Private Leaderboard成績 | Private Leaderboard 名次 |
| TEAM\_2035 | 0.9183106 | 25 |

* **隊員(隊長請填第一位)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 (中英皆需填寫) | 學校名稱 (中英文皆需填寫) | 系所 (中英皆需填寫) | 電話 | E-mail |
| 楊家和  (Chia-He Yang) | 國立中興大學  National Chung Hsing University | 資訊工程研究所  Department of Computer Science and Engineering | 0963619100 | [z8765421@gmail.com](mailto:z8765421@gmail.com) |
| 郭哲愷  (Che-Kai Kuo) | 國立中興大學  National Chung Hsing University | 資訊工程研究所  Department of Computer Science and Engineering | 0963697953 | [bjes940125@gmail.com](mailto:bjes940125@gmail.com) |
| 李主翔  (Chu-Hsiang Lee) | 國立中興大學  National Chung Hsing University | 資訊工程研究所  Department of Computer Science and Engineering | 0984360567 | [jeffzhux@gmail.com](mailto:jeffzhux@gmail.com) |

註：E-mail請填寫常用信箱，得獎後將以此信箱作為聯繫窗口。

* **指導教授/指導業師 (選填)**

若為「連結課程」課堂作業或期末專題，請填授課教授，以利依連結課程彙整；若非「連結課程」，但有實際指導之實，則請填此位教授/業師。填寫之指導教授或業師，我們將依信箱資訊，聯絡該指導教授/業師填寫指導證明。指導教授/業師為選填項目，若無則可不填。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 教授/業師姓名 (中英皆需填寫) | 課程名稱 (選填) | 課程代碼  (選填) | 學校名稱  (中英皆需填寫) | 系所  (中英皆需填寫) | 電話 | E-mail |
| 吳俊霖  (Jiunn-Lin Wu) | 影像處理 | 6653 | 國立中興大學  National Chung Hsing University | 資訊工程研究所  Department of Computer Science and Engineering | (04)2284-0497 ext 919 | jlwu@cs.nchu.edu.tw |