

Vision Transformer for Small Dataset: Orchid

Species Recognition

1. 環境

Colab

本次競賽中，我們主要使用 Google Colab 作為訓練模型的地方，同時也使用桌機來做一些資料前處理的研究。以下是在 Colab 上的環境：

- 作業系統: Linux
- 語言: Python 3.7.13
- 套件: 這裡列出關鍵的套件以及其版本

torch	1.11.0
torchvision	0.12.0
timm	0.6.2e
cv2	4.1.2
numpy	1.21.6
pandas	1.3.5

以上列出的套件在 Colab 上都可以直接 import 使用，除了 timm。我們可以用以下的指令在來下載(要下載當前 timm 在 github 最新的版本)

```
!pip install git+https://github.com/rwightman/pytorch-image-models.git
```

Local Computer

- 桌機配備
CPU: Intel I7-11700
GPU: RTX 3080 10G
- 作業系統: Windows 10
- 語言: Python 3.9.5
- 套件: 電腦上使用的版本會比較新一點，我們把套件的版本都放在 requirements.txt 裡面，可以用 `$pip install -r requirements.txt` 來下載所有需要的套件。

Pre-trained weights

因為這次競賽中能夠使用的樣本很少，所以需要藉由 pre-trained 來幫助模型學習。我們在 Public 最高的模型是使用 swin transformer v2，我們使用兩種解析度來訓練，以下提供它們的 pre-trained weights 來源：

1. [swinv2_base_window12_192_22k](#)
2. [swinv2_base_window12to24_192to384_22kft1k](#)

這兩個 pre-trained weights 是 Microsoft Research Asia (MSRA) 分別在 ImageNet22k 和 ImageNet1k 上進行訓練所得到的權重。我們分數最高的模型是使用 swinv2_base_window12_192_22k 模型和使用 MSRA 的 pre-trained 權重，在解析度 192*192 訓練，訓練好之後 fine-tune 在解析度 384*384，後面的訓練方法會更詳細地解釋這個操作。

額外資料集

我們一開始有實驗性地使用一些跟花有關的資料集，例如：Oxford102 flowers，想看看能不能先 pre-trained 在這些資料集上，然後再 fine-tune 在本次競賽資料集上。然而根據我們實驗的結果，這個是不理想的，模型訓練速度不但沒有變快，準確度也比較低，因此我們最後選擇只使用 pre-trained。從結果上來看，使用這些額外資料集不見得能幫助預測這次競賽中的蘭花品種，可能的原因有 (1) 本次競賽的蘭花是異類之間太像，但同類差異又很大，此時 pre-trained 在其他資料集上能得到的幫助並不大，比較有明顯幫助的是 pre-trained 在非常大的資料集上，例如：ImageNet22k，這可以幫助 model 事先取得一些較低階的特徵，再透過 fine-tune 取得高階的特徵，這樣的幫助比起 pre-trained 在花的資料集上來說會比較好。(2) 因為這次某些蘭花可能出自廠商的特有種，我們使用額外資料集不一定能學到這些特有種的特徵。綜合以上兩點，我們不考慮使用額外資料集。雖然我們最高分數的模型並沒有使用額外資料集，這裡依然提供我們有找到跟花有關的額外資料集連結：

- [Oxford_102_flowers](#)
- [Harvard orchid flowers dataset](#)

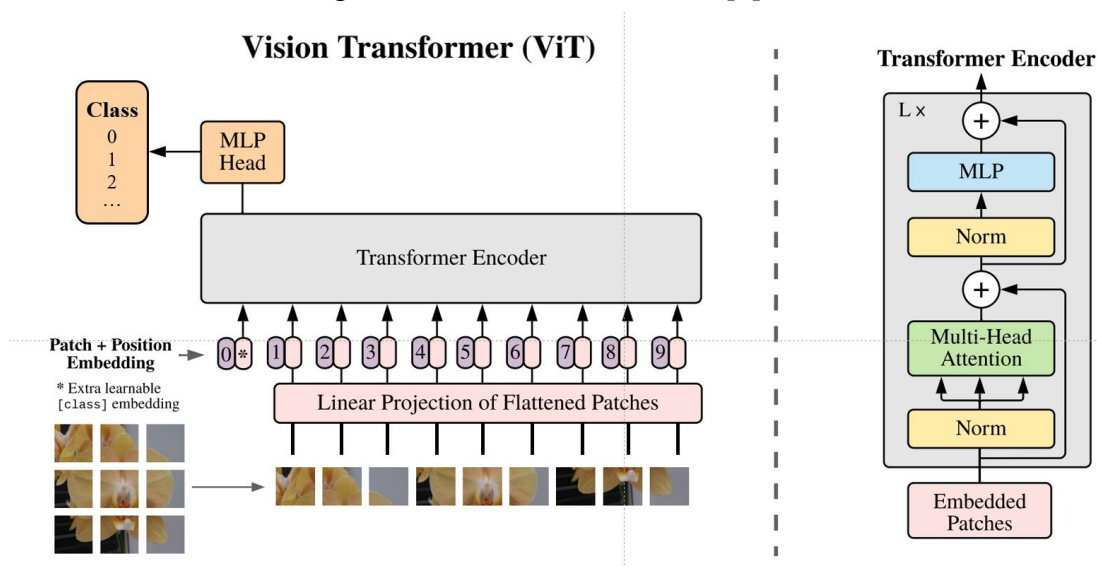
2. 演算法與模型架構

在本次競賽中，我們嘗試過許多模型架構，其中表現較好且有上傳到 submission 之中的模型有三個，分別為 ViT [1], Swin Transformer [2] 和 ConvNeXt [3]，其中 Swin Transformer [2] 表現最好。以下依序介紹模型的架構。

Vision Transformer (ViT)

ViT [1] 借鏡了 Attention is all you need [4]，將原本使用在 NLP 領域中的 Transformer 改成使用在視覺領域上。ViT [1] 把一張圖片分成數個 16×16 的 patches，接著透過一層的 Linear Projection 將這些 patches 轉成 token，這個步驟是 ViT [1] 最大的特點，在這之後的操作就跟 Transformer 一樣。在程式中，其實 Linear Projection 就是利用一層 convolutional layer 把 patch 轉成向量，這個目的是為了滿足 Transformer block 的 input 必須是一個 sequence，同時將圖片的 patch 轉成 token。下面是模型的架構圖

Figure 1 ViT model architecture [1].

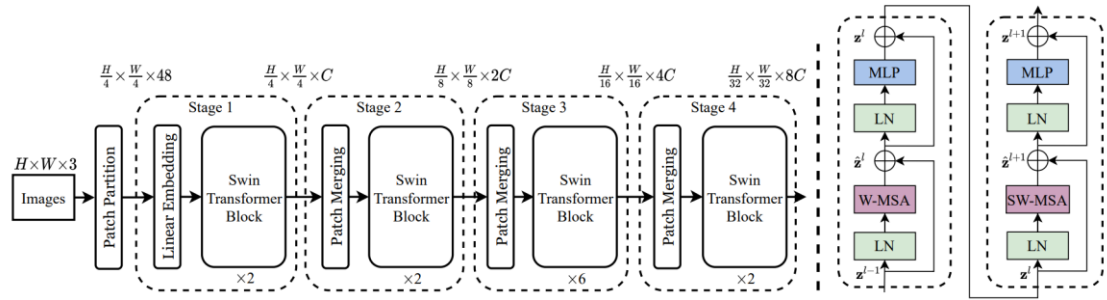


Swin Transformer

Swin Transformer [2] 是基於 ViT [1] 做改良的模型。其中 Swin 的 S 和 w 分別表示為 shift 和 window，也就是移動窗口的 Transformer。Swin Transformer [2] 最大的特色有兩個：(1) 使用層級式(hierarchical)的方式建構 Transformer block 這個特點讓 Swin transformer 有了像 CNN 一樣的特性，隨著網路的加深，越深層的 patch 所包含的訊息會逐進擴增，讓模型擁有更多尺度的特徵訊息。(2) 引入 locality 的想法，利用 shift window 達到 cross window 計算 self-attention。這個操作可以引入 CNN 在局部做捲機的操作，能更關注在局部性，同時也能夠降低運算量。Swin Transformer 把整個 Transformer encoder 分成四個 stage，每個 stage 裡面會有數個重複的 block。同時每個 stage 都會降低 input 的解析度，而 output channel 的維度會增倍，這個設計就是希望 Transformer 能夠像 CNN 一樣。

下面是 Swin Transformer 的整體架構

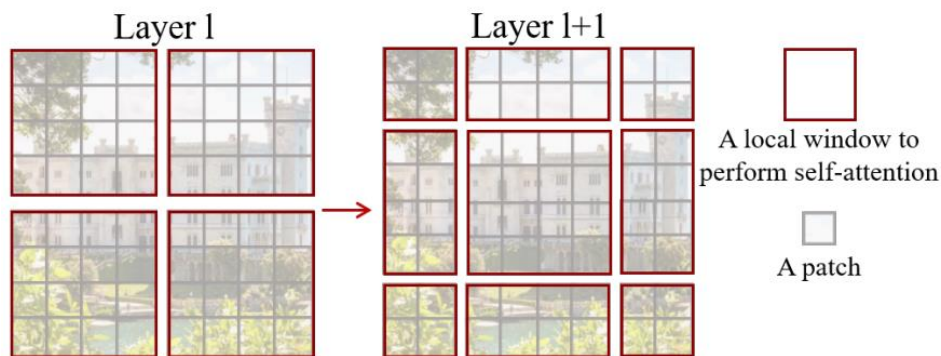
Figure 2 Swin Transformer model architecture [2].



一開始會將圖片做 patch embedding，將圖片轉成數個小的 patch，接著做 Linear embedding，這步驟跟 ViT [1] 的第一個步驟一樣。比較特別的是每個 stage 進入下一個 stage 之前都會做 patch merging，patch merging 是用來做 downsampling，降低 input 的解析度和調整 channel 的數量，為了就是像 CNN 一樣有層次的設計，好處是這樣子也可以減少一些運算量。傳統的 Transformer 都是用全局來計算 self-attention，而 Swin Transformer [2] 使用的是 window attention。window attention 只在一個比較小的窗口進行 attention 的計算，這邊使用到人類對於影像上的 domain knowledge，在影像中相鄰的 patch 之間關連程度通常比較大，而太遠的 patch 比較沒有關連性。因此利用 Swin Transformer 利用 window attention 更專注在局部上，同時也能降低計算量，因為能節省計算不必要的相關性。這個做法雖然有點違背 Transformer 當初的設計動機，但是在視覺領域上這個想法是成立的，因此 Swin Transformer [2] 在視覺領域的各項下游任務中都能表現很好。

為了達到 cross window 計算 attention，Swin Transformer [2] 使用 shifted window attention，如下圖

Figure 3



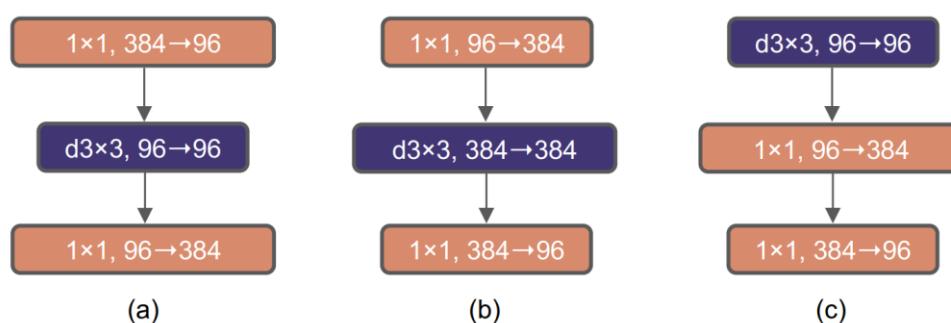
左邊是沒有 overlapping 的 window attention，而右邊是 shifted 過後 window attention。我們可以發現原本沒有交集的 patch 之間，因為 shifted window 之後就在同個 window 之內，這樣就能把資訊從其他 window 分享過來。雖然我們介紹的是 Swin Transformer [2]，實作上我們是用 Swin Transformer 的優化版本，Swin Transformer v2 [6]。Swin Transformer 和 Swin Transformer v2 的模型架構都相同，只差在 Swin Transformer v2 做了一些數值計算的優化。Swin Transformer v2 在計

算 self-attention 將原本用 dot-product 計算相似性的方法改成為 cosine 來算相似性，這可以改善某些特徵內積過大而 dominate 整體的 attention，這使得訓練上可以訓練得更好。

ConvNeXt

我們將 CNN 架構模型當作 baseline model，在許多 CNN 架構的模型中，ConvNeXt [3] 是我們實驗中表現最好的 CNN 模型。這個模型將傳統的 CNN 架構進行多處的修改，作者只使用 CNN 架構就可以超越 Swin Transformer，可見這個 CNN 架構是非常有料。ConvNeXt 的核心是把 Transformer 融合到 CNN 架構中，下面就來介紹 ConvNeXt 做了哪些改動

- **Inverted Bottleneck:** 作者把 Depthwise Convolution 當成 Self-attention Layer 看待，用來模仿 ViT [1] 的架構。Inverted Bottleneck 如下圖



- **增大 kernel size:** 自從 VGG [5] 出世，kernel size 大部分都使用 3x3，然而 ConvNeXt 設計不同大小的 kernel size 3x3, 5x5, 7x7, 11x11。作者發現 7x7 的效果比較 3x3 好。
- **使用 GELU 而不是 ReLU:** 使用 NLP 經常使用的 activation function GELU。
- **用更少的 activation function 和 normalization layer:** 只在 Depthwise Convolution 之後才加上 LayerNorm，在 invert bottleneck 中加入 GELU。如下圖

模型總結

我們介紹了 ViT [1], Swin Transformer [2], ConvNeXt [3]，在本次競賽資料集上的排名是 Swin Transformer v2 > ConvNeXt > ViT。我們也特別說明 Swin Transformer 使用的是 v2 版本[6]。以下我們列出三個模型在 Public 上的分數

	參數數量	Public Score
ViT-B	86,258,907	0.773985
Swin Trransformer v2-B	87,118,291	0.893669
ConvNeXt-S	49,623,099	0.817151

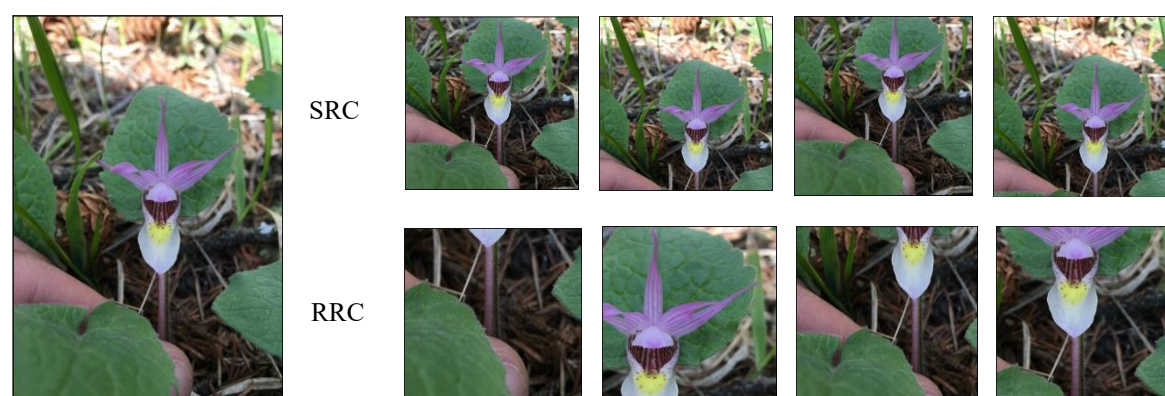
以上是模型架構的介紹，主要說明模型架構以及模型設計的想法，如果有需要補充的話我們都非常樂意再做詳細的說明。

3. 資料處理

在這次競賽中，我們沒有增加或刪去原始的訓練資料，只有做數據增強(data augmentation)，透過數據增強可以大幅度地增加模型的穩定性以及準確度。我們使用的數據增強方法大多取自於 Touvron et al. [7], [8]，裡面介紹一些不同於過去的數據增強方法，以及訓練 Trasnformer 的一些技巧，以下會分別介紹使用了哪些數據增強的方法。

Simple Random Crop (SRC)

Simple Random Crop(SRC)是用來改善 Random Resize Cropping(RRC)的方法，出自於 [7]。一般在訓練模型時經常會使用 RRC 來增加模型的穩地性，這相當於加上了正規化(regularization)來限制模型，可以降低 overfitting 的情況。隨機性是 RRC 的優點，但其優點本身也存在著缺點，因為有時可能裁切到不合適的位置就會給予 model 錯誤的 labeling errors。而 SRC 的想法就是希望改善在隨機 Cropping 下同時又不失去重要訊息，主要是用水平移動來裁切。以下放幾張圖來對比(取自類別 122 的樣本 qk8u6asrhl.jpg):



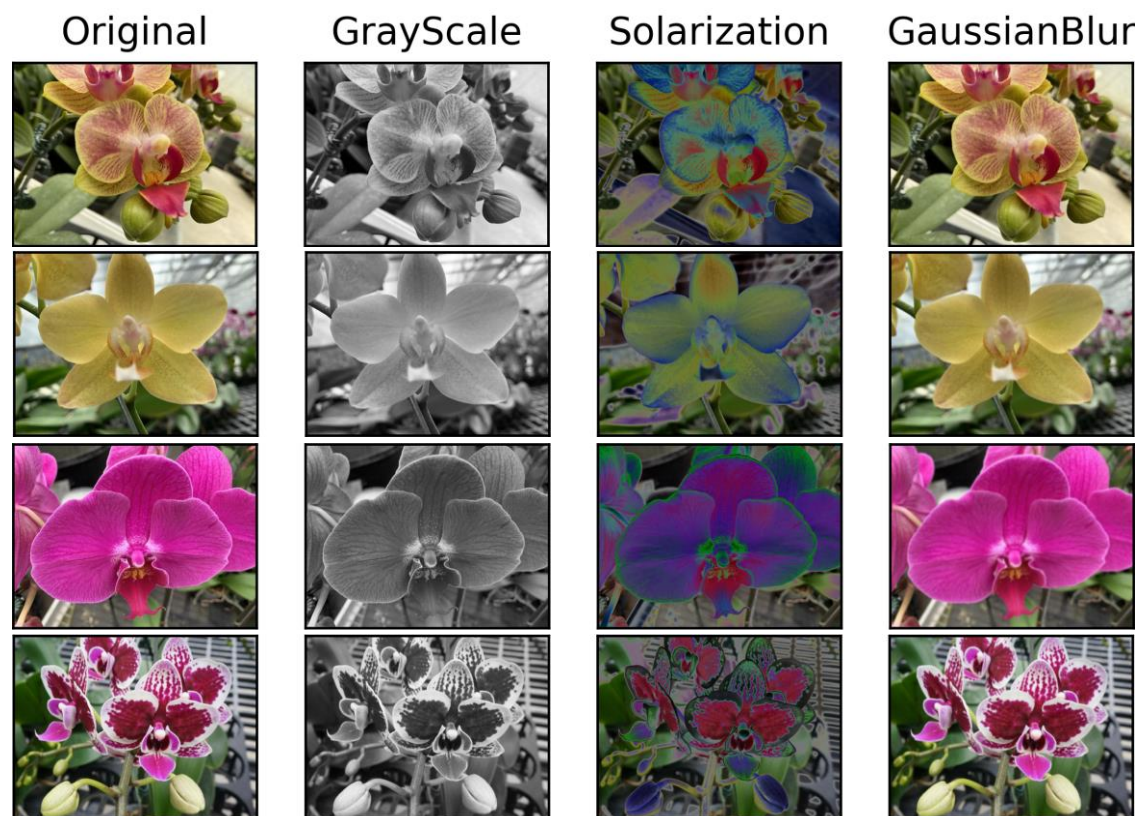
明顯地，可以發現 SRC 相較於 RRC 更穩定，同時也保留了 RRC 的隨機性。

3-Augment

這個方法同樣出自於 [7]。3-Augment 只用了三種數據增強的方法，相較於其他 data augmentation 的方法更為簡單，其他的常見的數據增強方法有 RandAugment [9] 和 AutoAugment [10]，這兩種都使用了 14 種不同的數據增強方法，3-Augment 更簡單更輕便，同時效果也很好。3-Augment 包含以下三種數據增強方法

- Grayscale: 把圖片轉成灰階。這有利於模型學習到顏色不變性，能夠讓模型關注在形狀上的差異。
- Solarization: 過度曝光。這個操作會增強照片在顏色上的噪音，模型不能只依靠單一顏色來做判斷，因此模型會更加關注照片的形狀。
- Gaussian Blur: 高斯模糊，通常用來降低細節層次。可以增強模型學習物件邊緣的能力，讓模型更專注在形狀上。

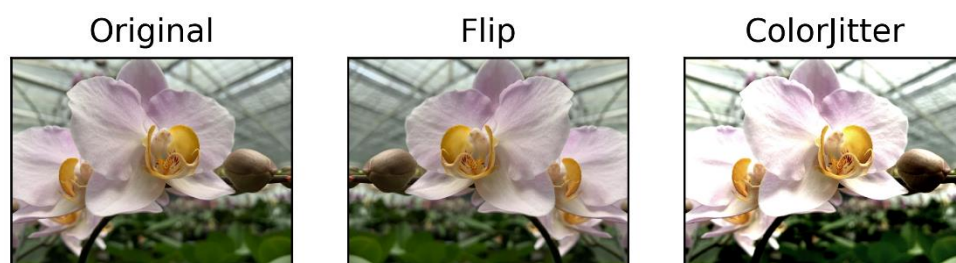
我們可以發現上述的三種方法的目的都是讓模型能更學習的更有一般性 (general)，並且三者的共同點都強調**形狀**的特徵，透過這三種數據增強的方法，可以幫助模型區分出不同類別之間細微的差異。我們從樣本中取一些圖片來示範使用 3-Augment 大概的情況：



在實際訓練中，我們會隨機選擇 3-Augment 之中的其中一個來使用，也就是說，模型會平均地看到三種不同處理的結果，擴大訓練樣本集合來增加模型的穩定性。

Color-Jitter and Horizontal Flip

除了上述的數據增強方法之外，我們也使用 color-jitter 和 horizontal flip 來增加樣本分佈的廣度，使模型具有不變性。以下是範例：



其中，我們是用 torchvision.transforms 來實作，而 ColorJitter 的 brightness、contrast 和 saturation 都設為 0.3。Horizontal flip 的機率設為 0.5。

Normalize

雖然將資料標準化算是訓練的基本常識，但因為對訓練影響很大所以在報告中還是特別提一下。我們根據 training.zip 提供所有資料計算 RGB 三個 Channels 的平均值和標準差，接著所有樣本在做完所有前處理之後，都會進行標準化。以下是三個維度的平均值和標準差

mean	(0.4909, 0.4216, 0.3703)
std	(0.2459, 0.2420, 0.2489)

以下再透過範例來突顯有標準化和沒有標準化的差異

Original



Normalized



Original



Normalized



資料處理總結

因為我們主要的模型是 Vision Transformer，因此參考一些訓練 Vision Transformer 的方法，這其中包含了 SRC 和 3-Augment。我們實驗後發現這些數據增強的方法確實可以幫助模型學習，在測試集上的表現有變好。我們更進一步發現，這些數據增強的技巧也對 CNN 架構的模型有幫助。例如，有使用 SRC 和沒有使用 SRC 對 ConvNeXt 是有顯著的影響，在我們的例子中，test size 設為 0.2 的情況下，使用 SRC 的話，準確度可以在 validation set 上升大約 5 點，而 3-Augment 就相對沒有顯著地提升模型的準確度。其中影響最大的數據增強是 Normalize，如果沒有做 Normalize 模型甚至會訓練不好，在 validation set 的表現也會非常差。最後，在測試的資料上，我們有做的處理是 CenterCrop 和 Normalize。

4. 訓練方法

前面介紹了我們主要使用的模型是 Swin Transformer v2，這邊會詳細地說明訓練的方法。在訓練上我們參考了 [2], [7], [8] 來訓練 Swin Transformer v2。訓練流程是將每一筆蘭經過 transform 轉成 192*192 解析度，接著才會餵給模型進行訓練。當訓練完整個模型後，會將再重新 fine-tune 在解析度 384*384 上訓練。這個作法是參考 [11]，根據此論文，我們在低解析度訓練，接著在高解析度進行微調會有助於增進模型的準確度。這個想法來自於希望訓練和測試的分佈越接近越好，因為實務上往往訓練和測試的 distribution 其實並不一致，而導致模型可能出現 bias；然而透過先再低解析度進行訓練，接著在高解析度進行微調可以幫助模型更好的 fit 測試資料。大致上的訓練流程如上，下面會仔細地介紹每個部份的方法以及參數設定：

Data settings

- Input image size: training on 192/fine-tune on 384
- Interpolation to resize image(random, bilinear, bicubic): bicubic
- Batch size: 32

Model settings

- Number of classes: 219
- Dropout rate: 0.1
- Dropout path rate: 0.1
- Attention dropout rate: 0.1

Swin Transformer

- Model architecture: swinv2_base_window12to24_192_to384_22kft1k
- window size: 24
- embed_dim: 128
- depths: (2, 2, 18, 2)
- num_heads: (4, 8, 16, 32)
- pretrained_window_sizes: (12, 12, 12, 6)

Loss Function

- LabelSmoothingCrossEntropy: 0.1(smoothing)

Training settings

- Epochs: 200
- Clip gradient norm: 5.0

Optimizer: AdamW

- Learning rate (base): $3e-3$
- Weight decay: 0.02

Scheduler: CosineLRScheduler

- Warmup epochs: 5
- Warmup learning rate: $1e-5$
- Min learning rate: $1e-5$
- K_decay: 0.75
- T_initail: 10

訓練方法總結

我們在實作中，也有直接使用 384 解析度搭配 swin transformer v2 作者訓練好的 pre-trained 權重來訓練，我們發現先訓練在 192 然後再 fine-tune 在 384 的模型，會比直接用 384 搭配 pre-trained weights 訓練稍微好一些，loss 有些微地下降。同時，我們在 pre-trained 和 fine-tune 所使用的參數都一樣，包含 epochs。唯一不同的地方就是模型的輸入 image size 和 patch 相關的層數。還有一個比較重要的是，我們在訓練中，只會訓練 Multi-Head Self-Attention(MHSA)，這是從 [8] 得到的想法。只訓練 MHSA 不僅訓練參數變少許多，同時也可以加快訓練速度，更重要的是，在小資料集上做 fine-tune 的表現會更好!

Fine-tune 時，我們試過只訓練 head 分類器，並沒有特別提升模型的表現，因此最後我們在 fine-tune 時也會訓練整個 MHSA 和 head，這樣做的話模型的表現有些微的提升。

5. 分析與討論

本次的競賽雖然是分類問題，但由於是在蘭花的子集合中進行分類，同類之間有大的差異，異類之間的差異很小，這類的任務屬於 Fine-Grained Image Classification。在這種任務中，我們希望模型能夠萃取到**細微**的特徵，以區分該品種和其他品種之間的差異。因此即便模型的準確度已經有一定水準，但是能否學習到關鍵的特徵才是我們真正在乎的，因此我們將透過 attention map 和 saliency mapping 觀察模型學習的情況。第二節的模型架構已經說明我們訓練三種模型，並且表現最好的模型是 Swin Transforemer v2，在分析與討論中，我們將使用三種訓練好的模型來評估模型學習的情況。在訓練階段，我們會透過 Metric(loss, accuracy, f_1 score)來評估模型的表現，而在訓練完模型之後，我們會透過模型可


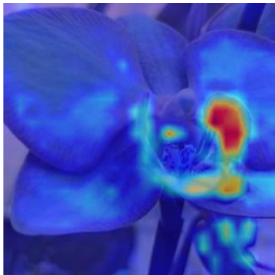
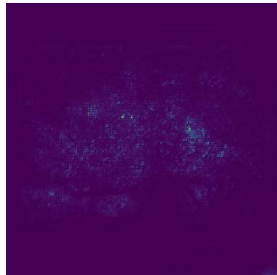
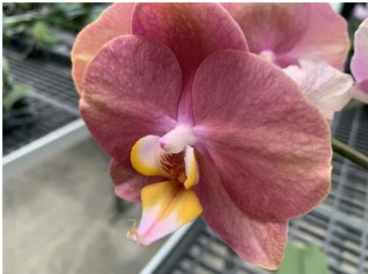

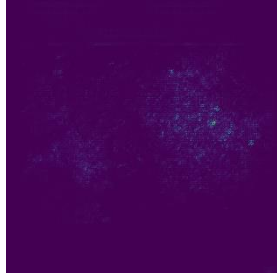

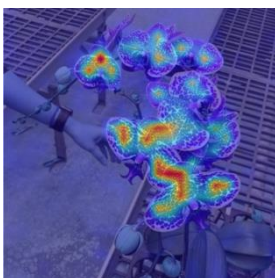
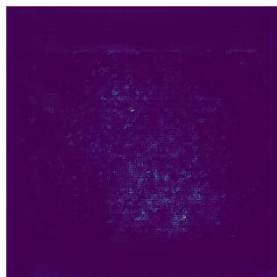

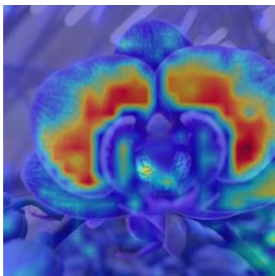

解釋的方法來判斷模型是否學習的良好，並且根據模型預測各個類別的機率來分析模型錯誤分類的原因。以下是我們評估模型學習情況的方法

- Prediction Error: 在模型訓練好之後，我們把模型用來預測 test dataset 上面，透過模型提供的機率，觀察模型預測錯誤時的原因。例如：可能某些類別的蘭花太過於相似導致模型錯誤預測，或著是某些角度很像另外一個品種的蘭花而導致模型錯誤預測...等等。
- Visualization attention mapping or Saliency Mapping: Transformer 的模型可以透過將 attention layer 的 attention scores 取出畫 attention map，透過視覺化 attention 之後可以了解模型關注的地方在哪裡，是否真的關注在花的本體上，還是有其他的雜訊沒處理掉，而模型關注在雜訊上，這些可能的原因都可以透過將 attention mapping 發現，讓我們去做更進一步地改善。CNN 架構的模型雖然沒辦法畫 attention map，但是透過 gradient 也可以畫 Saliency Mapping，因此我們也有將 ConvNeXt 的 saliency mapping 拿來跟 ViT 和 Swin Transformer 做比較。在實做中，因為 Swin Transformer 的 attention mapping 不太好實現，因此我們實作 ViT 的 attention mapping，雖然兩個模型架構和計算上不太一樣，但 ViT 的 attention mapping 仍然能提供有用的資訊，例如在資料處理上的修正。

根據模型最後預測的結果搭噴 attention mapping，我們能夠推測模型錯誤預測的原因，進而在前處理上做一些修改來增進模型的表現。以上是我們在訓練完模型後可以分析的地方，那麼後面會從訓練資料中找一些模型錯誤預測和成功預測的例子。

模型正確預測之理想情況

由於所有訓練資料都是蘭花，並且某些品種的蘭花彼此之間極為相似，因此模型不能單靠著簡單的特徵來分類，必須抓到更細微的特徵才能不失一般性，因此我們認為理想的情況是模型能夠準確地關注在蘭花細微的特徵上，例如**花瓣的形狀**、**花蕊的形狀**、**葉片的皺褶**...等等。下面我們從成功預測的類別中選一些理想的情況作為範例來畫模型可解釋圖

Class	Original	Attention Map	Saliency Map
35			
45			
50			
52			

其中 attention map 紅色的部位表示模型在該部位計算的 attention score dominate 於其他地方，即該位置對於模型分類上是重要的**關鍵**。從上面的 attention map 可以發現模型學習的方向大致上如我們所預期，將注意力關注在應該關注的地方。其中模型在類別 35 和 45 上都關注在花蕊的形狀上，這說明了我們使用的數據增強 3-Augment [7] 的方法是有效的，因為 3-Augment 的三個數據增強都是為了讓模型在**形狀**上加強學習。同時，在類別 50 中也可以發現顏色的深淺也會影響模型的


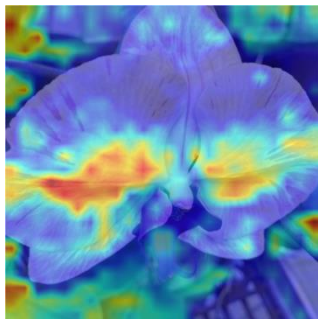

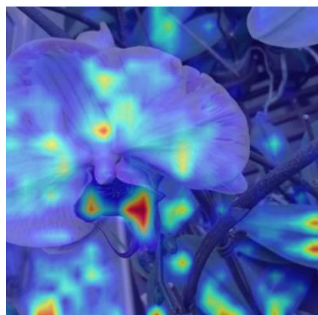
判斷，顏色比較深的蘭花分數就越高。至於 saliency map 的呈現度跟 attention map 相比就不是那麼好，但還是能隱約看出模型有抓出花的位置。透過這幾張模型的解釋圖，我們發現對於目前模型來說花蕊形狀、花蕊的顏色、葉片紋路和葉片的顏色都是關鍵特徵。

模型不理想之情況

模型會錯誤預測有很多種可能的原因，以下列出我們認為可能導致模型出現預測錯誤的原因，同時也搭配 attention map 來說明

- 模型學習方向錯誤

某些類別之間的蘭花十分相似，甚至人也很難直接分辨出這些類別的差異性，然而訓練資料中存在不少這種情況，雖然模型能夠準確預測出正確類別，但模型不一定學習到**關鍵特徵**。如以下

Class			Probability
55			→ 85.02%
121			→ 99.98%

我們發現類別 55 和 121 這兩類別的準確率都已經高達 80% 以上，但是在 attention map 上卻看到模型學習一些**不是**關鍵的地方，例如蘭花旁邊的綠色葉子、果實以及入鏡人的衣物，這些都不是我們期望模型學習的地方。不過還是可以看到模型也有學習到像是花瓣紋路和花蕊的特徵，但沒有發現模型有學習到人類難以觀察到的重要特徵。導致這個問題的原因可能出自於資料不足，也可能是訓練資料的分佈不一致，這些原因都可能導致模型出現 bias。解決此問題最直接的想法，可能是個別地訓練模型來訓練這些過於相似的類別，一個模型專門分類某幾類相似的品種，或許這樣能夠萃取關鍵的特徵。但在實做中我們都是訓練一個模型來預測全部類別的蘭花，因此未來可以考慮試試看用多個模型來預測，或是更加強 **local** 上特徵，以此增加類別之間的辨識度。

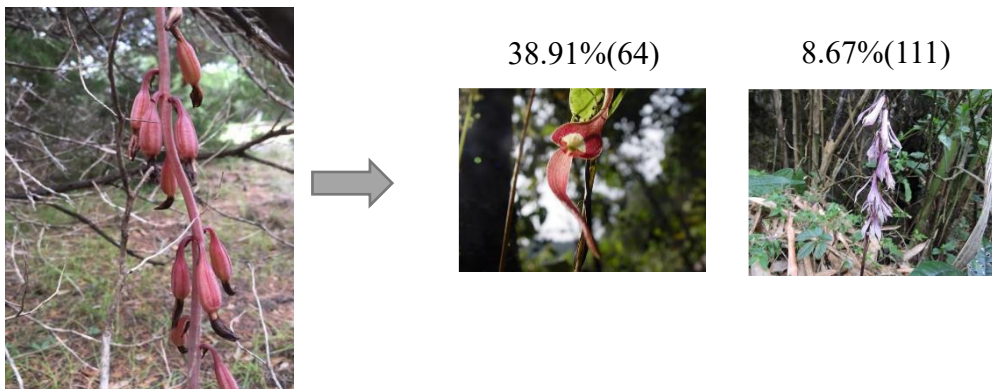
- **訓練資料分佈不一致**

我們在評估模型時會將模型預測錯誤的圖片存起來，發現模型有時預測錯誤可能是資料集本身的問題，有些錯誤預測的圖片跟其他正確分類的圖片看起來明顯不太一樣。比如有些花通通都是開花的樣子，但有一張沒開花，那麼模型很可能就對那張沒開花的圖片做錯誤預測。下面幾張照片都是類別 216 蘭花的品種，有些樣本偏紅色，有些樣本偏黃色，但大致上可以看出它們有些地方是相似，例如花的紋路或花葉內側的顏色。



但是上面的樣本都是有開花的樣本，然而當模型看到沒開花樣本就可能做出錯誤預測，比如下面的例子

錯誤分類樣本



因為這個錯誤分類的樣本與其它同類的樣本差異過大，導致模型出現錯誤判斷，判斷其為類別 64 和 111 的機率分別有 38.91%和 8.67%。要解決此類問題或許可以考慮把過於極端的樣本排除掉，例如上面的例子。一般可能覺得如果模型可以看過開花和沒開花的樣本，那麼模型在推論上穩定性會更好，但是在我們的例子中，由於樣本非常稀少，這會使得模型產生 bias，因此必

須考慮到樣本分佈不一致或不均衡的問題。

分析錯誤原因

我們透過畫模型可解釋圖來試著了解模型學習到什麼，以及透過模型錯誤預測的樣本來了解模型可能出現甚麼樣的問題。在最後的分析結論中，我們條列一些可能導致模型產生錯誤的原因

- **樣本分佈不一致:** 前面提到樣本是否維持一致性對模型很重要，比如照片拍攝的角度就很重要。如果十張照片中，九張都是開花，只有其中一張是沒開花，那麼模型對這張沒開花的照片就會出現 **bias**，原因來自於分佈不均導致模型出現偏差。可能的解決辦法是**增加樣本集**，或是盡量讓資料的分佈更一致。如果要讓模型能夠只看一兩張特殊情況的樣本就能學習到該特殊的情況，或許未來可以考慮 **few-shot learning** 來增加模型學習的能力。
- **無法萃取出關鍵特徵:** 導致模型無法區分相似類別的根本原因可能是無法萃取出關鍵特徵，或是模型學習方向不對，因此未來可以試試看在 Transformer 的架構中增加計算 local 的部分，將增加局部性。

未來改善

- **Self-Supervised Learning:** 我們使用的模型是 Swin Transformer v2。在該論文中有提到，可以考慮使用 Self-Supervised Learning 來 pre-train Transformer 的 ecoder。[2] 的團隊提出 SimMIM [12] 方法來做 pre-trained wieghts，這個方法的好處是不需要 label，因此可以簡單地擴增訓練資料。實作上可以考慮收集各式各樣的蘭花，先用 SimMIM 的方法 pre-train 在這些沒有標註的蘭花和本次競賽的蘭花上，接著再 fine-tune 到本次競賽的訓練資料上，或許能夠提升模型的表現。
- **Meta Learning:** 因為本次競賽各類的樣本數很少，已經滿接近 few-shot learning 的樣本數，因此可以考慮看看用 few-shot learning 的方法來提升模型的表現。
- **資料集加入文字敘述:** Transformer 的優勢在於可以結合文字來做訓練，如果能夠結合文字和影像去做分類，模型的一般性程度會更高。

分析總結

在樣本數集少的資料中要訓練 Transformer 是一大挑戰，如果未來應用上的資料可以更多的話，或許就能真正體現 Transformer 的強大。同時 Swin Transformer 不僅僅能夠用在 image classsification 的任務上，在下游任務的執行上也是非常強大，或許可以拿來偵測蘭花特徵的任務。最後，報告中有些地方可能我們理解不那麼正確，也可能是觀念上理解錯誤，但還是希望能帶給讀者一些啟發，在 Fine-Grained Image Classification 任務中能夠得更好的結果。

參考文獻

- [1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M. Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
- [2] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
- [3] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022)
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
- [5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR (2015)
- [6] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv:2111.09883 (2021)
- [7] Touvron, H., Cord, M., & Jégou, H. Deit iii: Revenge of the vit. arXiv preprint arXiv:2204.07118 (2022)
- [8] Touvron, H., Cord, M., El-Nouby, A., Verbeek, J., Jégou, H.: Three things everyone should know about vision transformers. arXiv preprint arXiv:2203.09795 (2022)
- [9] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: RandAugment: Practical automated data augmentation with a reduced search space. arXiv preprint arXiv:1909.13719 (2019)
- [10] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)
- [11] Touvron, H., Vedaldi, A., Douze, M., Jegou, H.: Fixing the train-test resolution discrepancy. Neurips (2019)
- [12] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In Tech report (2022)