

AI作物影像判釋

組員: 何彥南 110753202、林政委111753138、李尚霖111753165

Problem statement

我國農地範圍廣泛，但分佈破碎，造成收集資源的人力和時間成本極高，且農業領域相對缺乏AI技術，因此本實作將會把大量已收集並標住過的農作物進行分類和預測。

Dataset

資料為33種類農作物，每種作物包含2000 ~ 3000 張照片，共有 89513 張照片。並含有拍攝作業用之準心資訊，照片由行政院農業委員會提供，影像皆由專家進行分類。下面針對資料本身的問題做分析與整理：

1. 準心可用性

準心為協助專家判斷作物之依據，但準心也可能產生偏移。其中非中心的準心標記有 22 %。其中只有 0.5 % 資料準心偏移中心超過 100(差不多準心大小)。且幾乎全部偏移都是在 Y 軸。

下圖可以看到準心標記錯誤的問題，因此種照片佔極少數，所以先忽略。我們初步作法是以準心為基準往外取夠大的範圍，只要照片中有包含到準心和周遭一定範圍的作物就好。



圖1. 準心的干擾範例

2. 解析度大小

照片本身的解析度大小會影響整個模型訓練的難度，解析度越高需要越多的記憶體、訓練時間、模型參數量，此外，在資料移動上也會很費力。

然而在高解析度的照片中有許多像素都

是重複值，當大到一定程度後會對模型預測幫助不大，太高甚至會干擾結果。此外，模型也需要配合去增加隱藏層和模型複雜度才能有效去學習高解析度的圖片。

因此，在初始解析度大小上，我們統一將原始最高的 2400 萬像素壓縮 200 萬畫素左右。原始資料也從 127 G 減少到 32 G，方便後續資料的處理與搬移。

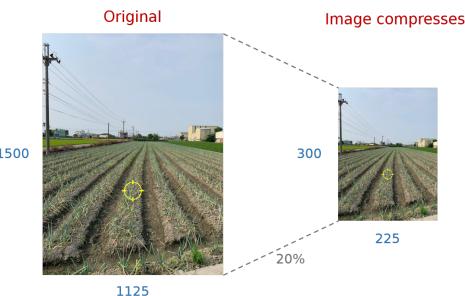


圖2. 圖片解析度壓縮範例

3. 照片長寬不統一

照片需要統一固定的大才能輸入到模型中進行訓練，主要的做法有剪裁(crop)和縮放(resize)。如下圖就是以統一大小的正方形對照片做 Crop，以此作為輸入特徵。

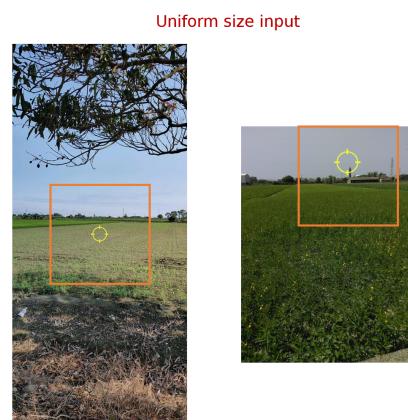


圖3. 用 crop 方式統一輸入大小範例

4. 照片轉向問題

在資料中有些圖片可能會有翻轉的現象，當某些方向的照片太少會導致模型無法有效辨識有轉向的照片。處理這問題有兩種方向，一是偵測非正向的照片並將其轉回正常方向，二是讓模型學習不同方向的照片，也就是資料增強的技術。在資料增強上一般的作法是將一張照片複製成四張分別轉向不同方向。使用 `torchvision.transforms.RandomRotation` 方法讓模型在訓練過程中，`dataloader` 會隨機對圖片進行翻轉，讓模型可以學習到不同方向的資訊。

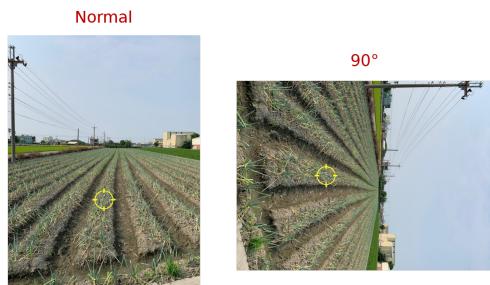


圖4. 照片轉向範例

5. 照片亮度 & 色彩差異

在蒐集照片時可能會因為天氣、光源或是設備上的成像差異導致模型難以對圖片進行學習。所以圖片做標準化是很重要的一個步驟，在 2017 年發在 *Biomedical Texture Analysis* 一篇關於生物醫學影像的文章 [1] 上提到：

Image normalization ensures optimal comparisons across data acquisition methods and texture instances. The normalization of pixel values (intensity)

is recommended for imaging modalities that do not correspond to absolute physical quantities.

也就是說對圖片標準化可以降低在實際成像上的差異與干擾，讓模型可以更容易去比較照片中的資訊。

在圖片標準化的部分，我們使用 `torchvision.transforms.Normalize` 函式對圖片進行 z-score normalization。



圖5. 照片明亮度不同範例

Method

CoATNet

此模型架構由CNN結合Transformer改進而來，傳統CNN架構藉由提取方形範圍中的特徵和前饋來進行模型建構，充分利用二維圖形單一範圍內像素相互關係，但對於圖形全域關聯敏感對較低；Transformer模型與CNN特性恰好相反，對於全域關聯敏感性高，小範圍中關聯敏感性低，此模型結合兩者的優點，對圖片先利用CNN提取出局部特徵，再前饋進入Transformer模型。

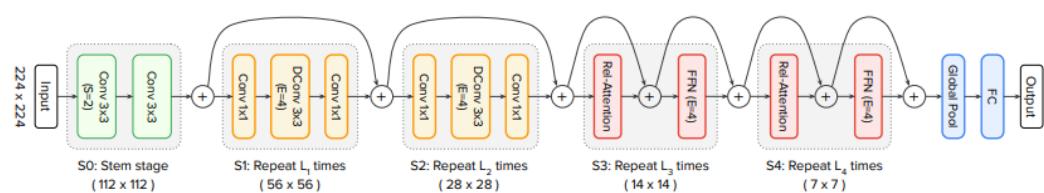


圖6. CoATNet 模型架構

Experiment

在模型參數上因時間與效能限制使用原作者預設的模型。在訓練環境下，因為沒顯卡運算資源，所以 kaggle 提供的 GPU(Tesla P100-PCIE-16GB)。然而在 kaggle 上有一個 session 有12 小時的限制，所以我們將模型輸入限制在照片大小限制在 224x244，這樣就可以在時間內跑完 25 epoch。

在資料集上將資料切分成 train(0.7)、val(0.1)、test(0.2)訓練時會使用 train 資料進入模型進行訓練，每個 epoch 會使用 val 資料進行驗證，並保留 val 結果最好的模型

1. 各類樣本數量

樣本數量會直接影響訓練時間和效果，因為我們運算資源有限，所以想測試一下大概需要多少樣本量才能讓模型具有一定的預測能力。藉此可以在減少每次訓練時間同時，獲取具參考性的實驗結果。

本實驗分成全部資料、每類別 2000 筆、每類別 1500 筆三組進行測試。輸入大小皆為 224x224，統一用 40 batch size 和 0.01 learning rate (每 5 epoch 減半) 跑 25 epoch。

結果如表 1，可以看到抽部分資料訓練結果差一大截，所以後續實驗還是會以全部資料進行比較。比較意外的是每類別抽 1500 的結果居然比抽 2000 的還要好，這邊判斷可能是抽樣的樣本差距

導致的，換句話說就是 SP2000 剛好抽到比較多難預測的樣本，所以更嚴謹的作法是對每組實驗重複抽樣多次，分別訓練後，將結果平均，以減少這方面的誤差。

表 1. 樣本數實驗結果

Exp.	Weighted Precision	Top1	Top 3	Top 5
all	0.647	0.628	0.817	0.883
SP2000	0.293	0.282	0.528	0.653
SP1500	0.374	0.328	0.555	0.669

2. Crop & Resize

這部分要測試不同方式將照片統一規格到 244x244 時效果，如圖 7 可以看到這部分將實驗分成三組，準心 Crop 224x224、Resize 到 224x224、Crop 690x690(最小長寬)再 Resize 到 224x224。

結果可以看到(表 2)，整體來看是直接扭曲原始圖片 Resize 到 224 效果最好，但是以 Weighted Precision 來看，先用 crop 切到一定大小再用 resize 壓縮到 224 的效果最佳。

表 2. crop 與 resize 方法實驗結果

Exp.	Weighted Precision	Top1	Top 3	Top 5
C224	0.575	0.545	0.751	0.824
C690 + R224	0.646	0.628	0.817	0.883
R224	0.644	0.640	0.826	0.890

3. 地理資訊

本資料集有詳細提供每張照片拍攝之地點(鄉鎮)，我們首先利用測試集中作物和鄉鎮關係建立二維矩陣 A，矩陣中記錄對應鄉鎮中種植過的作物，再來照片進入模型預測後得到一組矩陣 B (紀錄為某作物的值)，讀取此作物之鄉鎮，透過 A 取得此鄉鎮出現過之作物，將這些作物從 B 中提出數值並選取其中最大值作為預測作物；在預測階段中加入此步驟後精準度提升 2%

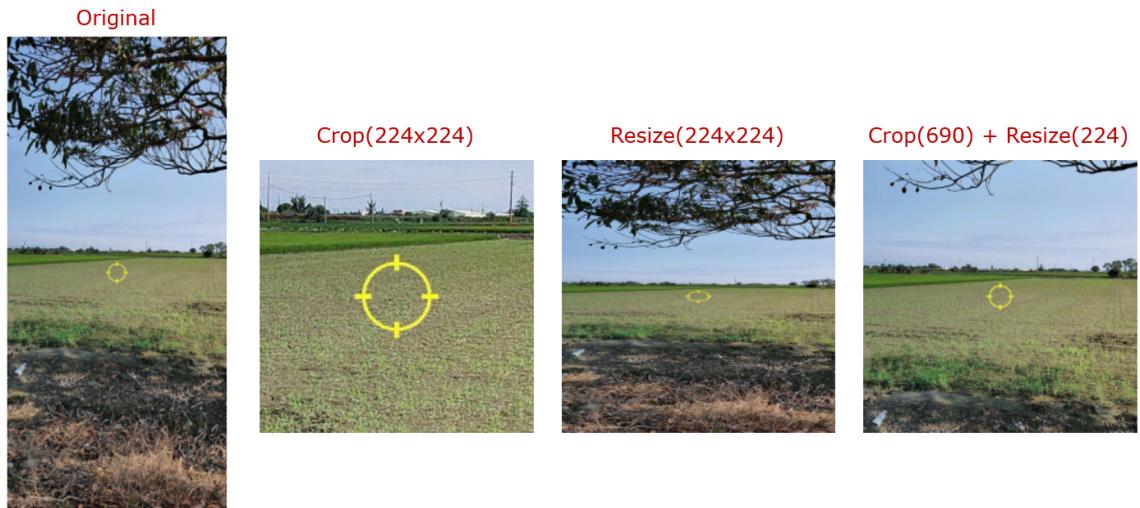


圖7. Crop & Resize 實驗示意圖

72 ▲	85	TEAM_2314	0.7668202	2022/12/15 21:47:14	22
73 ▲	91	TEAM_2405	0.7625553	2022/12/15 00:04:18	5
74 ▲	92	TEAM_2426	0.7553168	2022/12/15 10:16:02	7
75 ▲	96	TEAM_2025	0.7429153	2022/12/15 19:14:41	5
76 ▲	97	TEAM_2730	0.7409843	2022/12/13 18:31:49	2

圖8. 最終比賽上傳結果

天空是蠻重要的一個資訊。

4. Mask 掉天空



圖8. mask 掉天空示意圖

這部分使用 openCV 將圖片轉成 HSV 格式後 再使用 inRange 篩選出對應顏色範圍(10 ~ 80) 抽取出來，其他的 mask 掉，如圖8可以看到天空的部分。

由表 3 的結果是原圖比較好，由此可知

表3. mask sky 實驗結果

Exp.	Weighted Precision	Top 1	Top 3	Top 5
mask-sky	0.630	0.602	0.804	0.872
no-mask	0.644	0.640	0.826	0.890

Result

圖 8 為最後 AI cup 上傳結果，分數為 0.75，位於第92 名。這是我們第一次接觸影像辨識相關的任務，還有許多需要學習和加強的地方。因比較晚參賽加上運算資源不足，所以最後有一些實驗結果來不及套用在最終模型上。這也是鰻

可以習的地方。

過程中我們學習到：

1. 對於圖片資料的一些基本前處理方式和經驗。
2. 理解 CoATNet 並使用公開在 github 的模型。將其套用在本次的任務上。
3. 設計實驗流程和程式
4. 免費運算資源的使用

Code available

程式碼整理在 github 上：

<https://github.com/aaron1aaron2/Crop-Image-Classification>

Reference(s)

- [1] Depeursinge, A., Fageot, J., & Al-Kadi, O. S. (2017). Fundamentals of texture processing for biomedical image analysis: a general definition and problem formulation. In Biomedical Texture Analysis (pp. 1-27). Academic Press.
- [2] PEÑA, José M., et al. Object-based image classification of summer crops with machine learning methods. *Remote sensing*, 2014, 6(6): 5019-5041.
- [3] YANG, Guofeng, et al. Fine-grained image classification for crop disease based on attention mechanism. *Frontiers in Plant Science*, 2020, 11: 600854.
- [4] BHOJANAPALLI, Srinadh, et al. Understanding robustness of transformers for image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021. p. 10231-10241.
- [5] HAN, Kai, et al. A survey on visual transformer. arXiv preprint arXiv:2012.12556, 2020, 2(4).
- [6] [Crop Image Classification with Transfer Learning | Kaggle](#)
- [7] [Crop analysis and prediction | Kaggle](#)
- [8] [Agriculture crop images | Kaggle](#)
- [9] [Learn the Basics — PyTorch Tutorials 1.12.1+cu102 documentation](#)
- [10] [PyTorch深度學習工具箱 - Yanwei Liu - Medium](#)

[11] [GitHub - Atcold/pytorch-Deep-Learning: Deep Learning \(with PyTorch\)](#)

[12] [Image classification \(huggingface.co\)](#)

[13] [Image classification with Vision Transformer \(keras.io\)](#)

[14] [\[Train + Infer\] CoAtNet + EfficientNet | Kaggle](#)