



FMA Defect 分析

部門 : ML5C01

指導者 : 唐竹宣

實習生 : 湯琦恩

Agenda

01 專案簡介

02 架構細節

03 結果分析

04 訓練環境與設備

05 結論&未來展望

06 實習心得

個人簡介



湯琦恩
Chi-En Tang

ML5C01

學歷

國立暨南國際大學 電機工程學系 大三

專長

電腦視覺、影像處理、深度學習

競賽經歷

2023 科技論文競賽 人工智慧組 第三名

AI CUP 2022 教育部全國大專校院人工智慧競賽

Coding 101 大學程式設計競賽

國科會大專生計畫

相關經歷

AI 人工智慧導論 助教

專案簡介

FMA Defect 分析



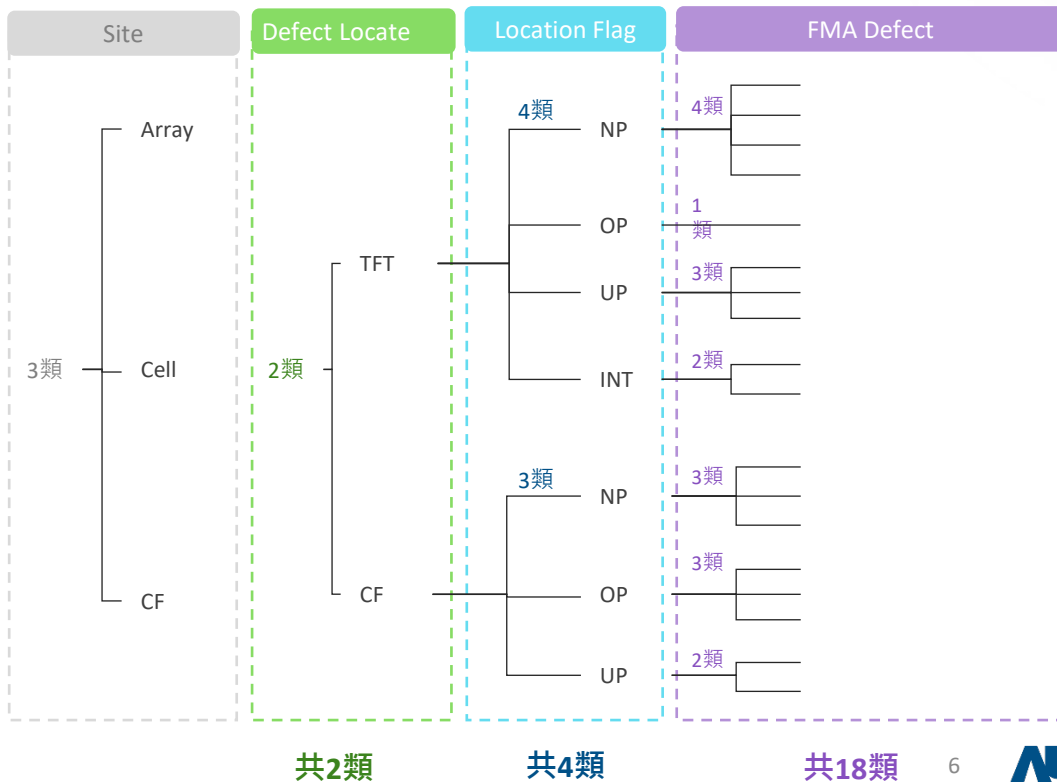
自訂義分類樹架構

訓練集照片篩選準則



為了保持各類別照片量平衡並避免過擬和，
選取數量 > 80張的類別作為訓練資料集

自訂義 Category Tree 類別階層結構



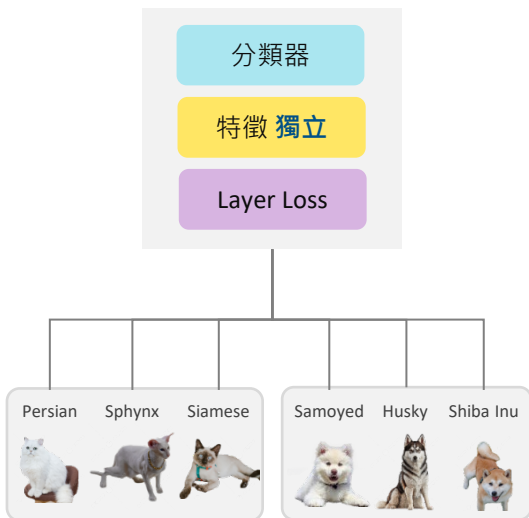
架構細節

分類框架說明

常見分類框架

Classification Framework

- 模型學習全部類別(classes)的特徵。
- 類別之間互相 **獨立**。

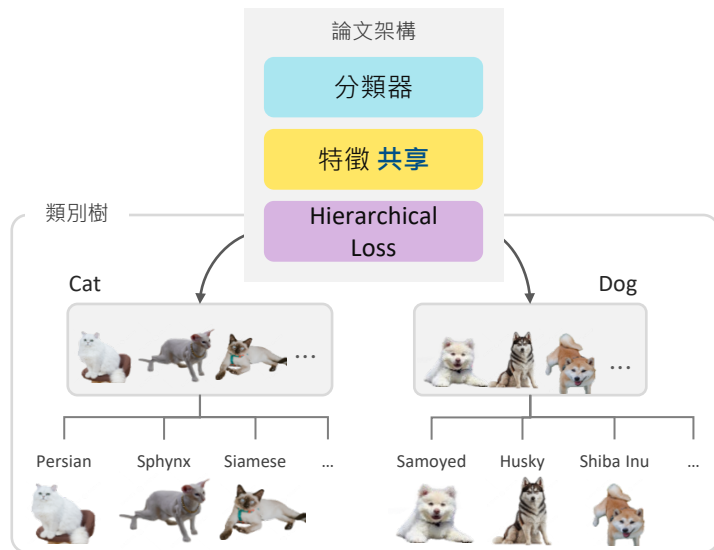


深度階層式分類框架

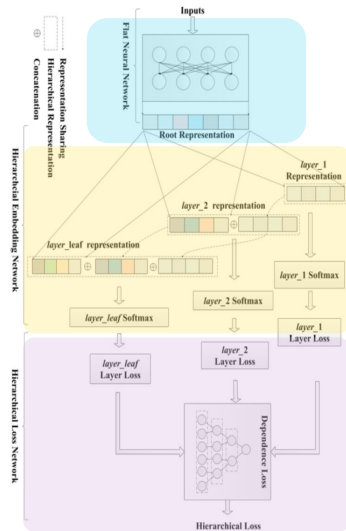
Deep Hierarchical Classification framework

參考論文：Deep Hierarchical Classification for Category Prediction - 2020.05

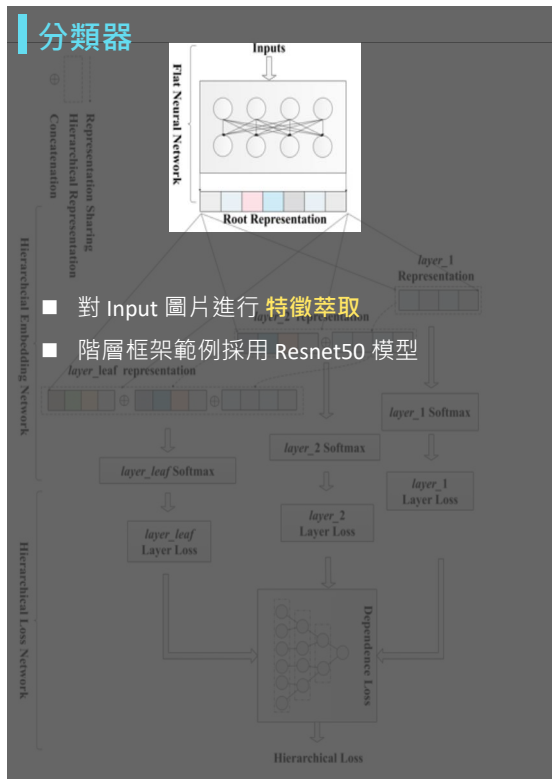
- 將自定義的類別樹 (Category tree) 結合「階層結構」於「神經網路」上進行訓練。
- 階層式框架損失計算 (Hierarchical Loss)。



論文架構原圖

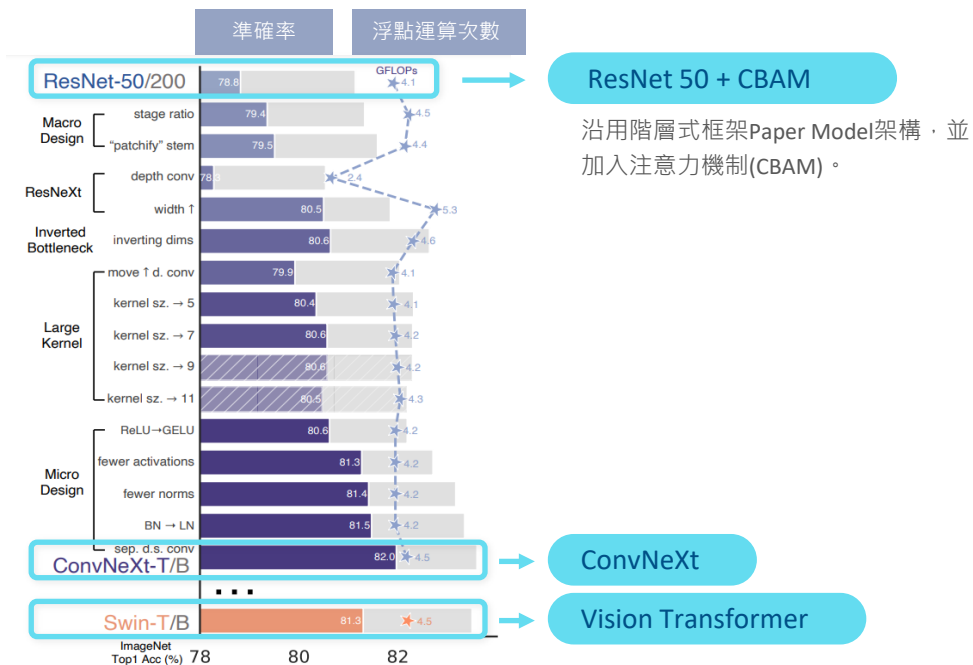


深度階層式分類框架_分類器選擇



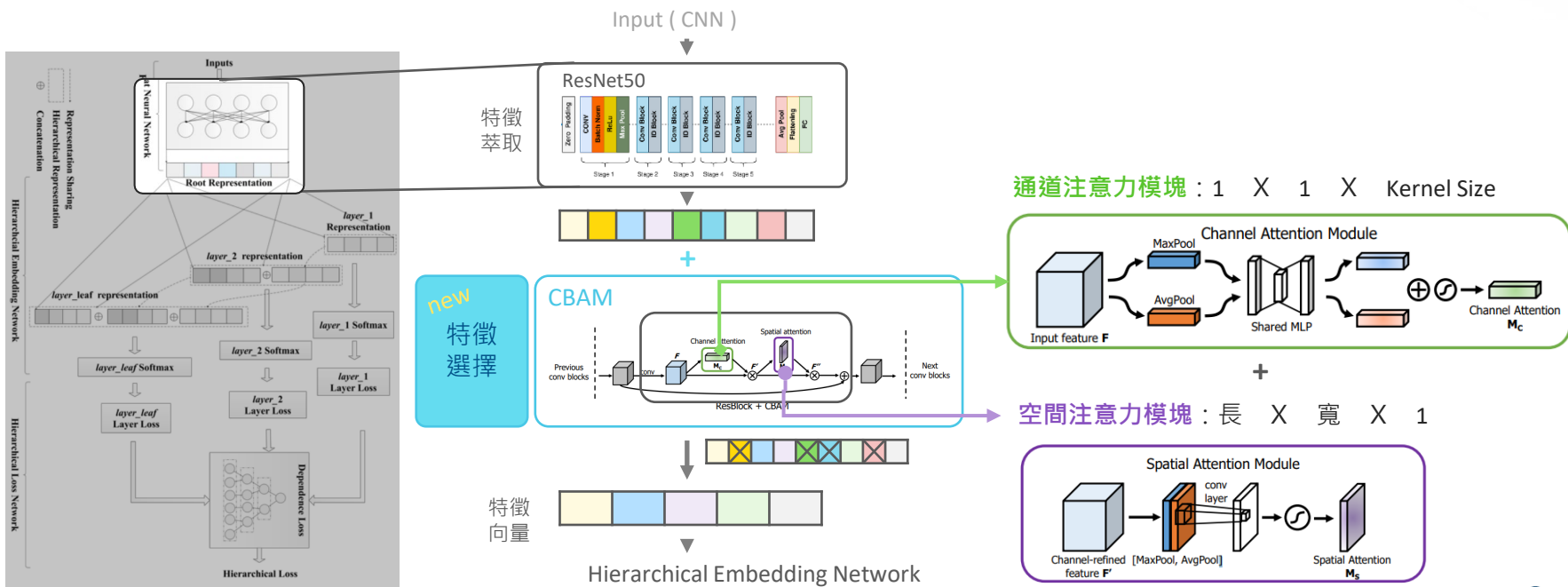
根據 Paper 選擇在 ImageNet 上表現較佳的模型進行後續實現。

參考論文：A ConvNet for the 2020s - 2022.03



ResNet50 +CBAM

- 基於論文範例模型加入延伸模塊(CBAM)
- CBAM 學習哪一部分為要重點關注的特徵，提高該部分特徵的權重，並抑制不必要的特徵
- 應用注意力模塊於通道與空間兩個維度中，學習特徵圖 (feature map) 在通道與空間上該注意的部分

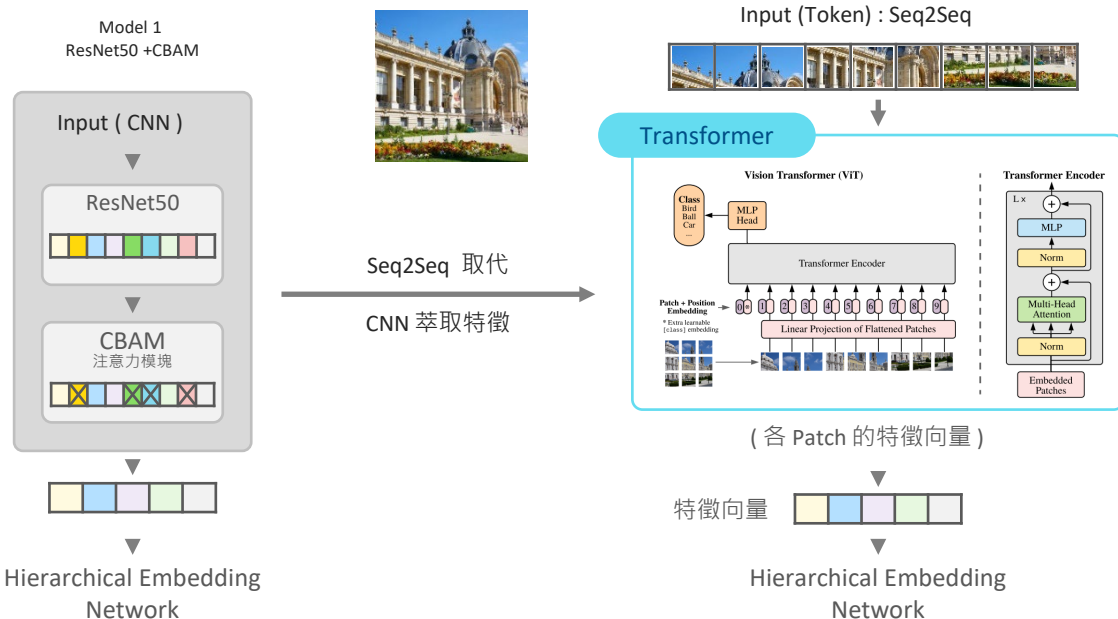


Vision Transformer

→ Transformer 使用圖片切分作為輸入

→ 「Seq2Seq」取代「CNN」萃取特徵

Transformer Encoder機制：使用『自注意力、多頭注意力機制』，結合 Encoder 與 Decoder 輸出預測結果
(參考論文：An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale –(2021.07))



Token 的生成程序

1. 輸入圖片分割成 N 片 (Patch Embedding)
2. 輸入圖片的位置編碼 (Position Embedding)

Transformer Encoder 機制

1. 獲取各 patch 間的關係 (Multi-Head Attention)
2. 獲取單 patch 的局部特徵 (Position Embedding)

Fig. 1

深度階層式分類框架

特徵共享

- 每一層用 concat 特徵對應 該層類別 Softmax
- 下面層數需包含上層特徵資訊
- 達到『父層』與『子層』之間特徵共享



損失計算 Hierarchical Loss

總損失 = 層內損失 lloss + 層對層之間損失 dloss

$$J(\theta) = \sum_{i=1}^L \alpha_i lloss_i + \sum_{i=2}^L \beta_i dloss_i$$

(1) 層內損失：各層的 Layer Loss

→ CrossEntropyLoss

(2) 層對層損失：判斷是否為父層與子層關係

$$dloss_i = \sum_{\text{batch}} (ploss) D_i \pi_{i-1} (ploss) D_i \pi_i - 1$$

前一個 Level 預測對不對

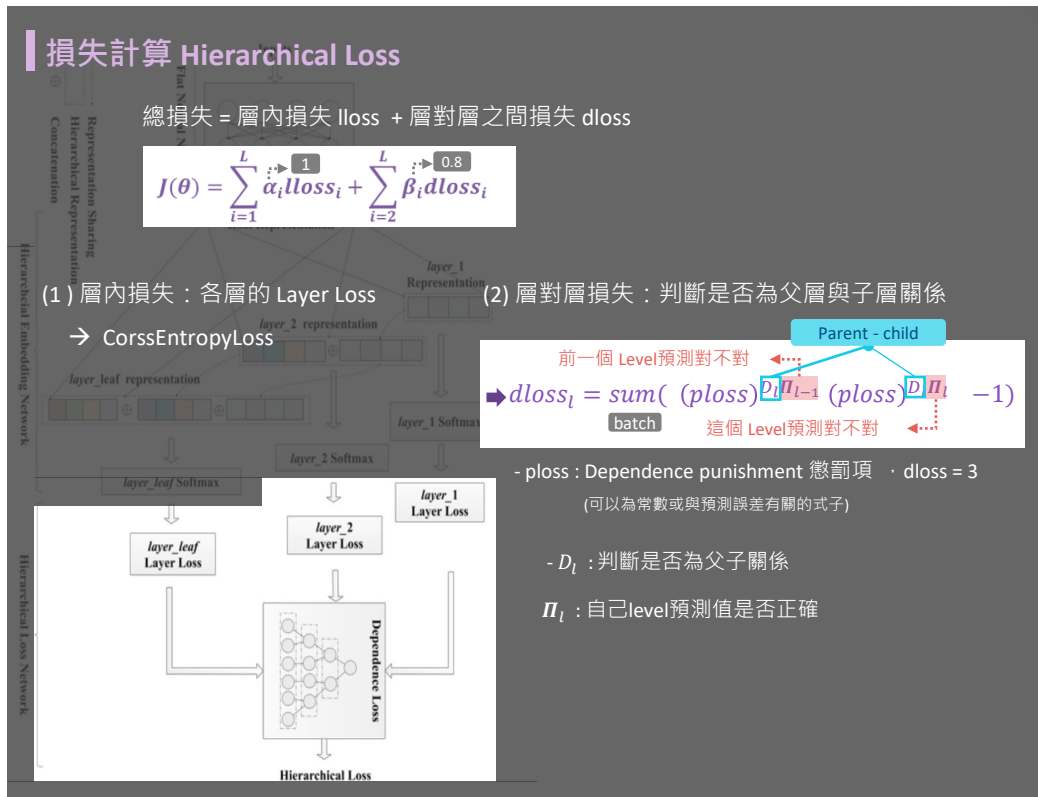
Parent - child

這個 Level 預測對不對

- ploss : Dependence punishment 懲罰項 · dloss = 3
(可以為常數或與預測誤差有關的式子)

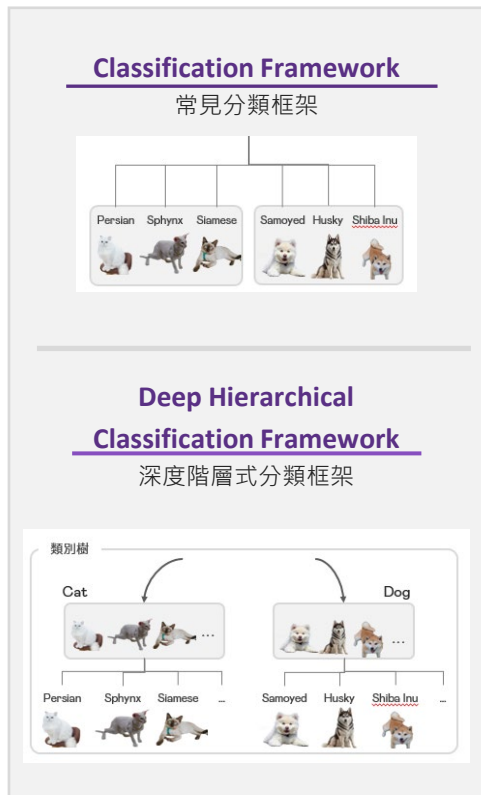
- D_i : 判斷是否為父子關係

π_i : 自己 level 預測值是否正確

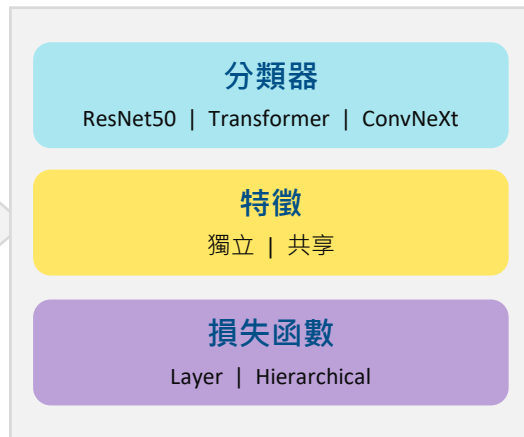


實驗架構

分類框架



論文架構



實驗組合

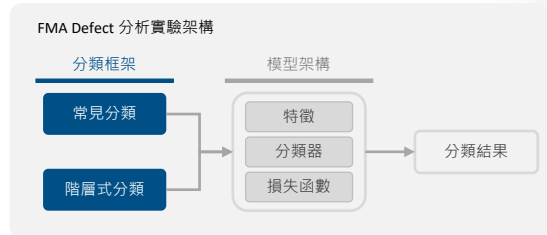
分類器	Type	特徵	損失函數
ResNet50	一般框架	獨立	Layer
Transformer	階層式框架	獨立	Hierarchical
ConvNeXt		共享	Hierarchical

結果分析 & 討論

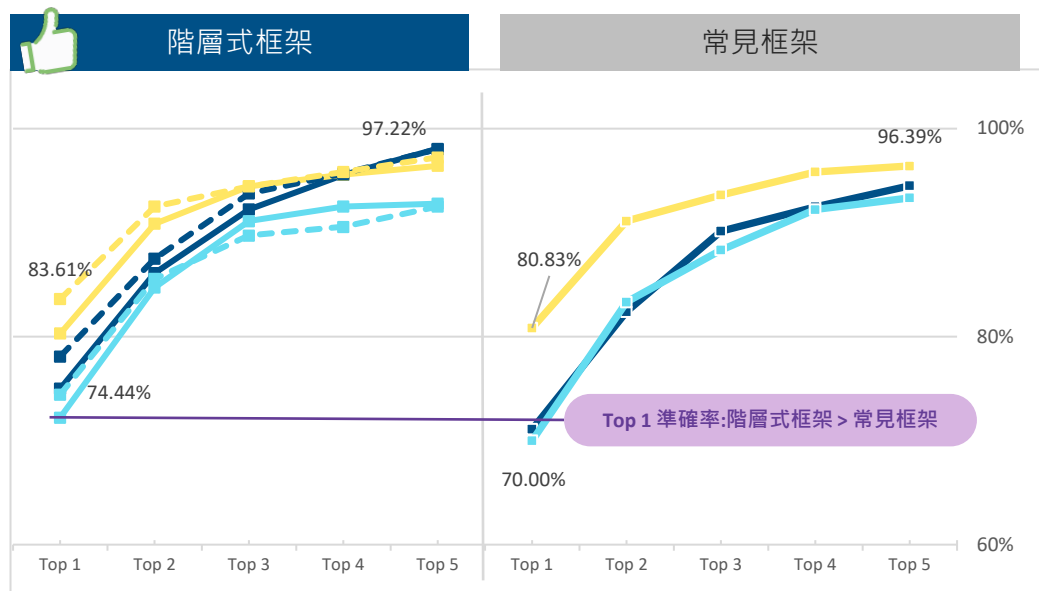
框架分析

常見分類 vs. 階層式分類

- Top 1 準確率：階層式分類框架 > 常見分類框架
- 階層式框架模型訓練，使得模型學習到類別樹父子階層關係
- 常見框架無類別樹概念，無法透過父層輔助判別較相似的種類



分類器			
<div> <div></div> ResNet50 <div></div> Transformer <div></div> ConvNeXt </div>			
Type	特徵	損失函數	圖例
一般	獨立	Layer	——
階層式	獨立	Hierarchical	——
	共享	Hierarchical	- - -



特徵分析

共享特徵 vs. 獨立特徵

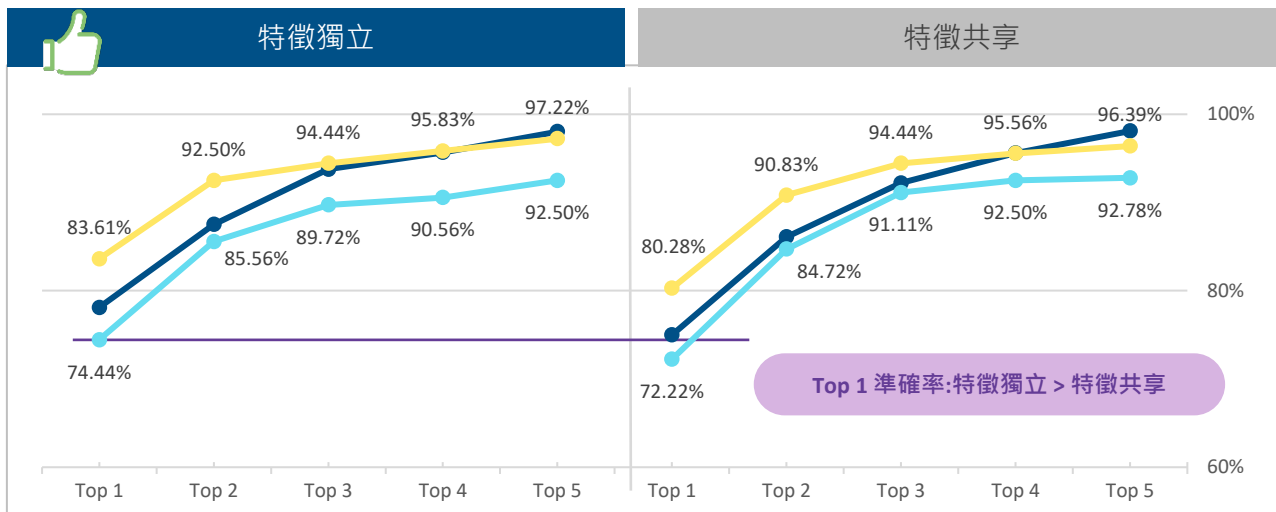
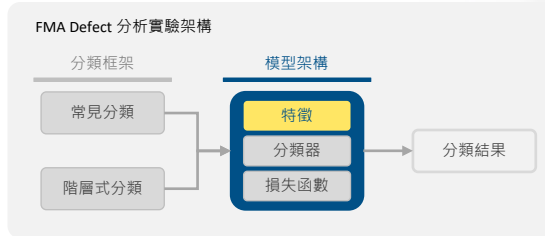
→ 在「階層式分類架構」中，Top1 準確率 **特徵獨立** > 特徵共享

→ **特徵獨立特色**：每一層特徵各自獨立

優點：可以著重學習子層之特徵判別

→ **特徵共享特色**：父子層間特徵繼承

缺點：父階層特徵可能佔比較大，影響子層的判別結果

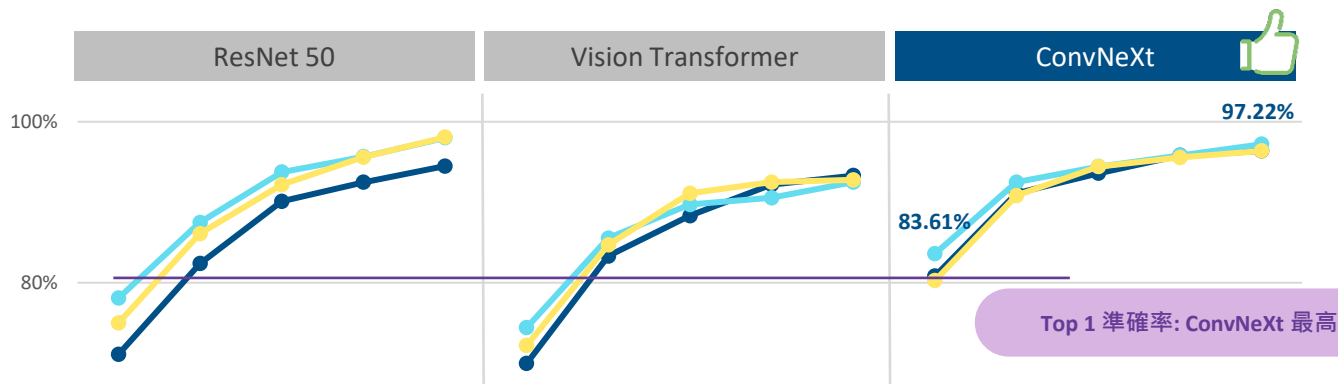
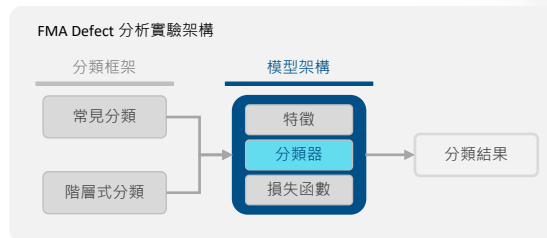


模型分析

ResNet50 / Vision Transformer / ConvNeXt

→ Top 1 準確率：ConvNeXt > ResNet50 (+CBAM) > Vision Transformer

→ ConvNext 保留 CNN 的優點並加入 Transformer 架構，
在提升模型準確率的同時也降低模型大小

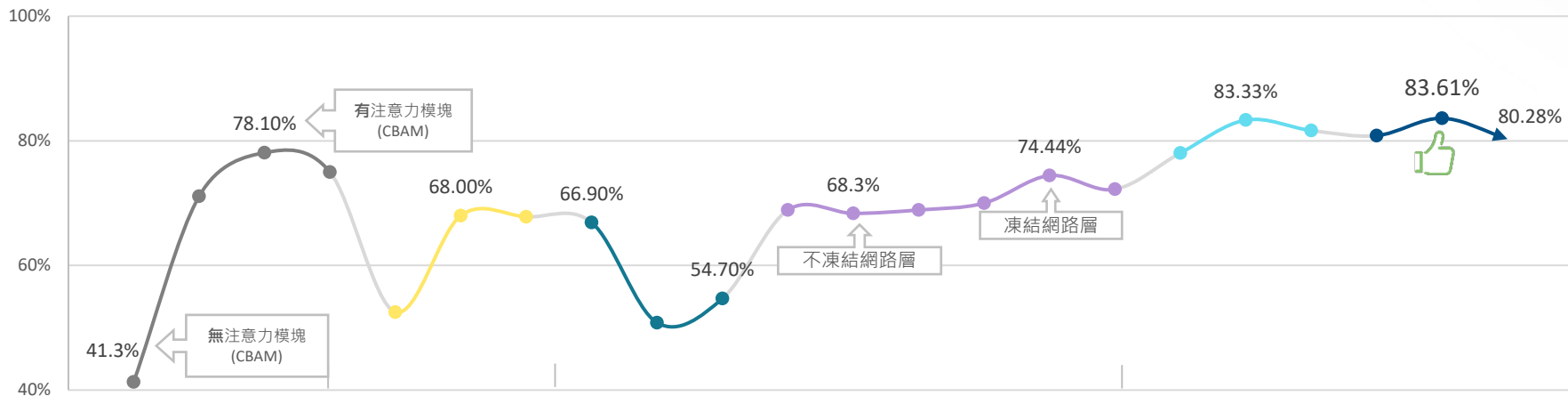


Top 1 準確率: ConvNeXt 最高

	Top 1	Top 2	Top 3	Top 4	Top 5	Top 1	Top 2	Top 3	Top 4	Top 5	Top 1	Top 2	Top 3	Top 4	Top 5
常見分類	71.10%	82.40%	90.13%	92.48%	94.50%	70.00%	83.33%	88.33%	92.22%	93.33%	80.83%	91.11%	93.61%	95.83%	96.39%
階層式分類_特徵獨立	78.10%	87.50%	93.76%	95.66%	98.02%	74.44%	85.56%	89.72%	90.56%	92.50%	83.61%	92.50%	94.44%	95.83%	97.22%
階層式分類_特徵共享	75.00%	86.10%	92.20%	95.60%	98.10%	72.22%	84.72%	91.11%	92.50%	92.78%	80.28%	90.83%	94.44%	95.56%	96.39%

模型架構歷程

→ 各分類器延伸種類 & 模型大小進行實驗 · 判別 FMA Defect 的準確率



ResNet50

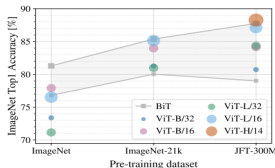
訓練花費時間長
(約2~3天)

ResNet101

訓練花費時間比
Resnet50還久
(約3~4天)

Small Tiny
Vision Transformer

表現較CNN差:



此模型在訓練集較少的表現會比較差

(註: 原論文中提及)

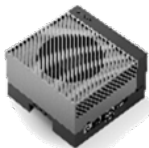
Small Tiny
ConvNeXt

1. 訓練花費時間短 (約4小時)
2. 模型大小 (Tiny > Small): 目前收集資料量Tiny 即可
3. 準確率表現最佳 → 由此模型繼續往下分析

訓練環境與設備

→ 模型訓練時間：ConvNeXt < Transformer < Resnet50

各模型訓練效能比較

使用設備	訓練參數		Model Size					訓練時間 (Epoch 100)	開發環境
	Input Size	Batch size	Model	權重檔	Forward/backward pass size	Params size	訓練佔Memory		
NVIDIA JETSON AGX ORIN DEVELOPER KIT ■ 12-core Arm Cortex-A78AE 64 bit CPU ■ Memory 36GB 256-bit 	224 * 224	6	Resnet50	100 kB	11083.21 MB	109.12 MB	30 G	2~3 天	■ Pytorch 1.11.0 ■ Python 3.7.9
		32	Transformer	22.2 MB	105.91 MB	20.94 MB	14.8 G	5 小時	
		32	ConvNeXt	111.5MB	1337.26 MB	917.3 MB	11.2 G	4 小時	

ConvNext 佔用的Memory少・時間較快

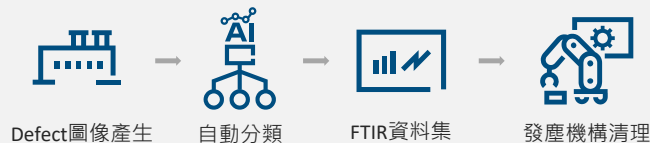
結論&未來展望

未來展望

場域應用

1. 持續收集訓練集與更多 Defect 類別的照片增加實用性
2. 模型訓練帶入更多 Defect 種類增加可辨識之種類 (18→147)
3. 將影像分類結果結合，成分分析「FTIR 資料數據」快速找到 Defect 發生機構及位置

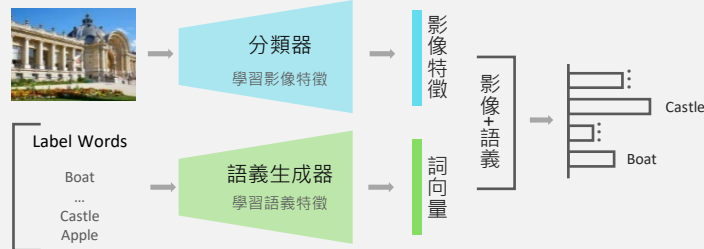
場域應用流程圖



技術層面 (提升模型準確率)

1. 持續調整訓練中父子層特徵共享之 **權重比例**
參考論文：B-CNN: Branch Convolutional Neural Network for Hierarchical Classification
2. 嘗試 **多標籤分類框架**，導入語義特徵，輔助影像分類
參考論文：Deep Semantic Dictionary Learning for Multi-label Image Classification

多標籤分類框架



實習心得

專案內容

參與專案開發 實作經驗增加
接觸工廠運作 報告能力提升

實習氛圍

部門前輩指導
辦公室氛圍輕鬆愉快
食堂菜單豐富

宿舍生活

與夥伴交流實習期間的點滴





Tap Into The Possibilities

AUO

Color List

RGB 0, 80, 135

RGB 100, 220, 240

RGB 255, 230, 100

RGB 180, 145, 215

RGB 135, 220, 100

RGB 190, 230, 225

RGB 255, 240, 180

RGB 215, 180, 225

RGB 210, 240, 155

RGB 255, 200, 170

RGB 210, 180, 150

RGB 20, 120, 145

RGB 230, 165, 15

RGB 100, 60, 150

RGB 5, 135, 50

RGB 230, 95, 90

RGB 120, 90, 70

RGB 0, 0, 0

RGB 65, 65, 65

RGB 100, 100, 100

RGB 150, 150, 150

RGB 200, 200, 200

RGB 240, 240, 240