



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

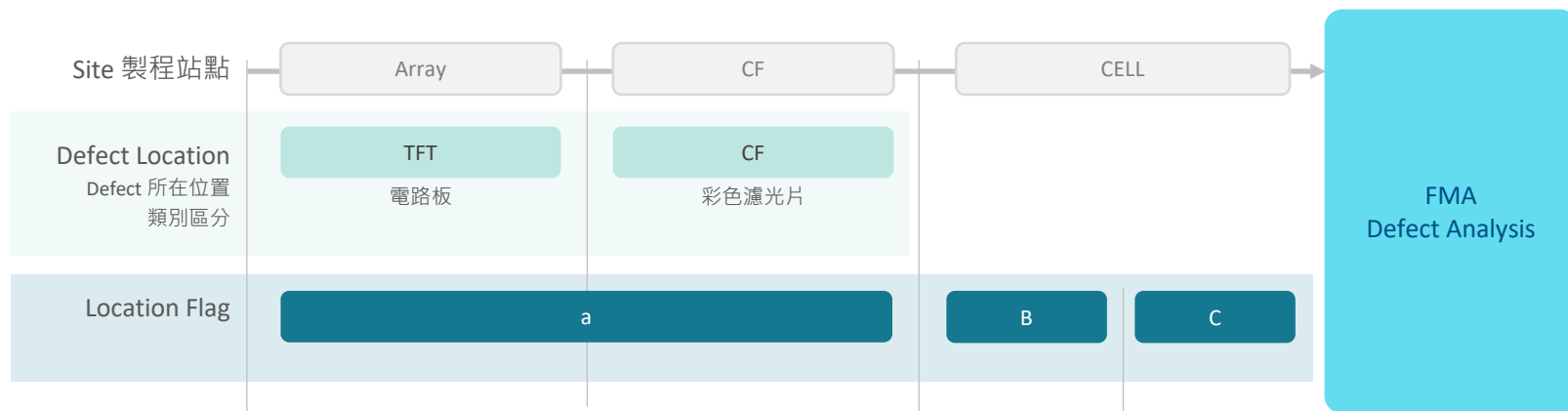
FMA Defect 分析



Failure Mode Analysis 失誤模型分析

Defect生成站點說明

各階層命名及含意



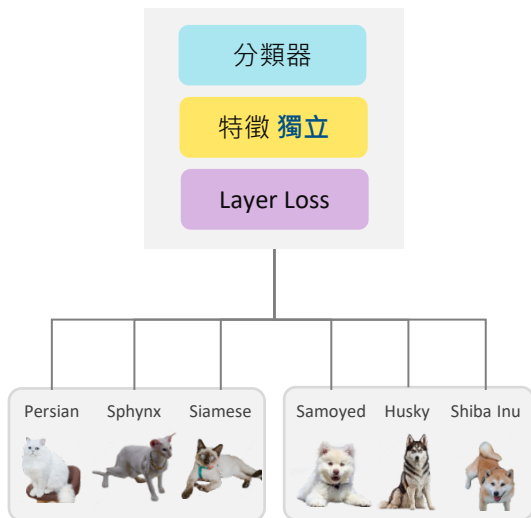
架構細節

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**。

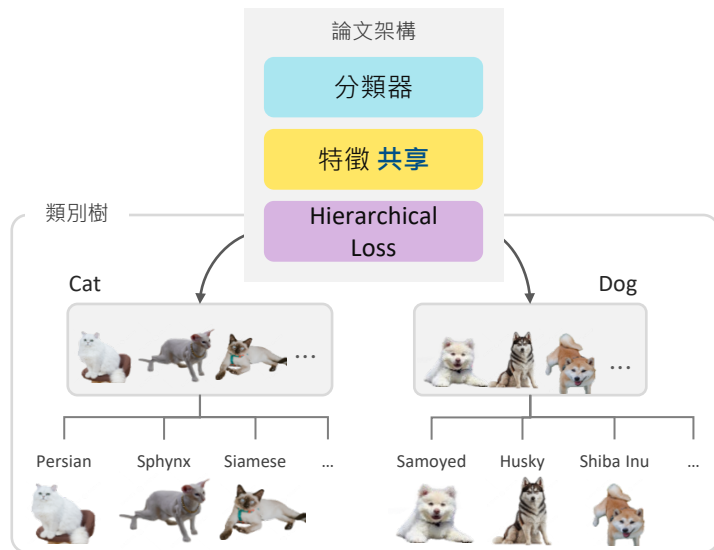


深度階層式分類框架

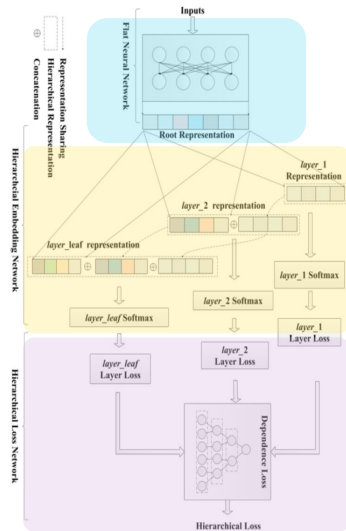
Deep Hierarchical Classification framework

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

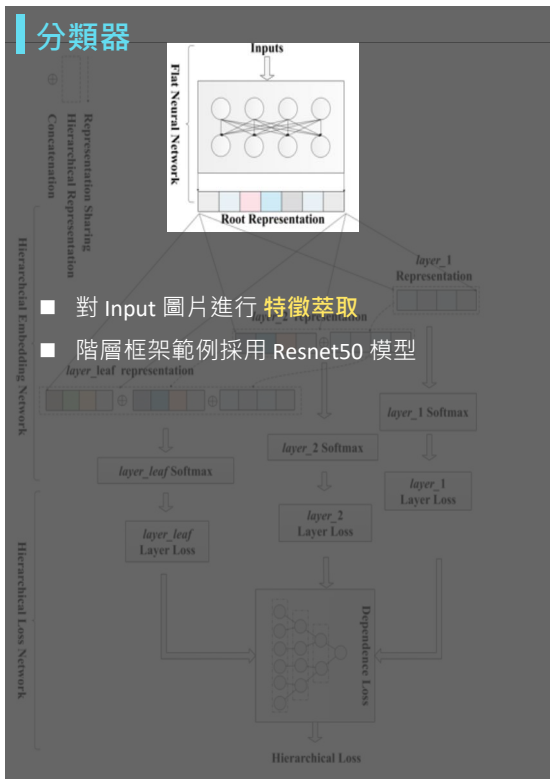
- 將自定義的類別樹 (Category tree) · 「階層結構」結合「神經網路」。
- 階層式框架損失計算 (Hierarchical Loss)。



論文架構原圖

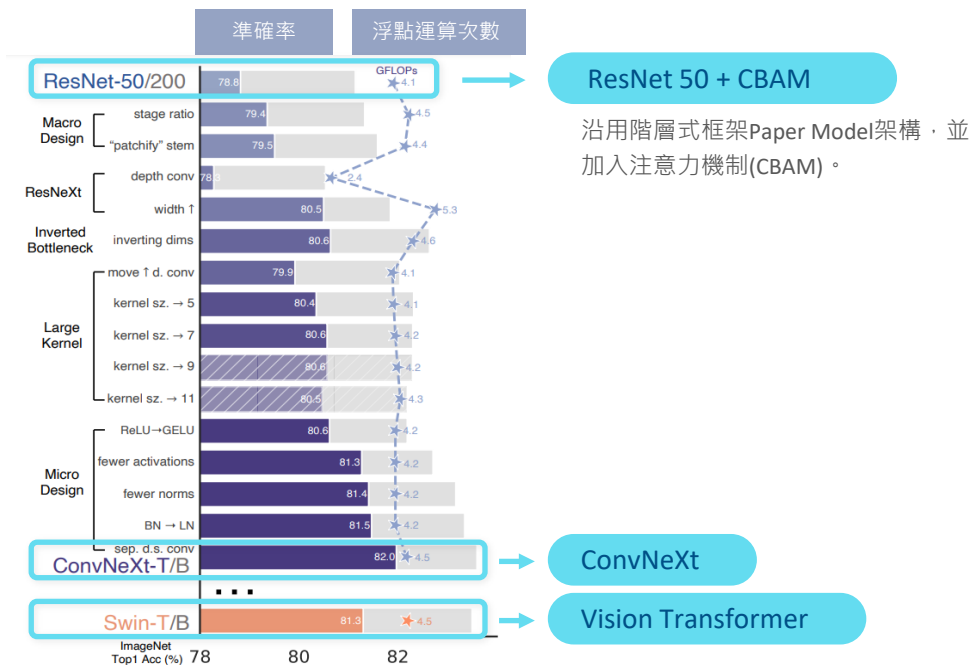


深度階層式分類框架_分類器選擇



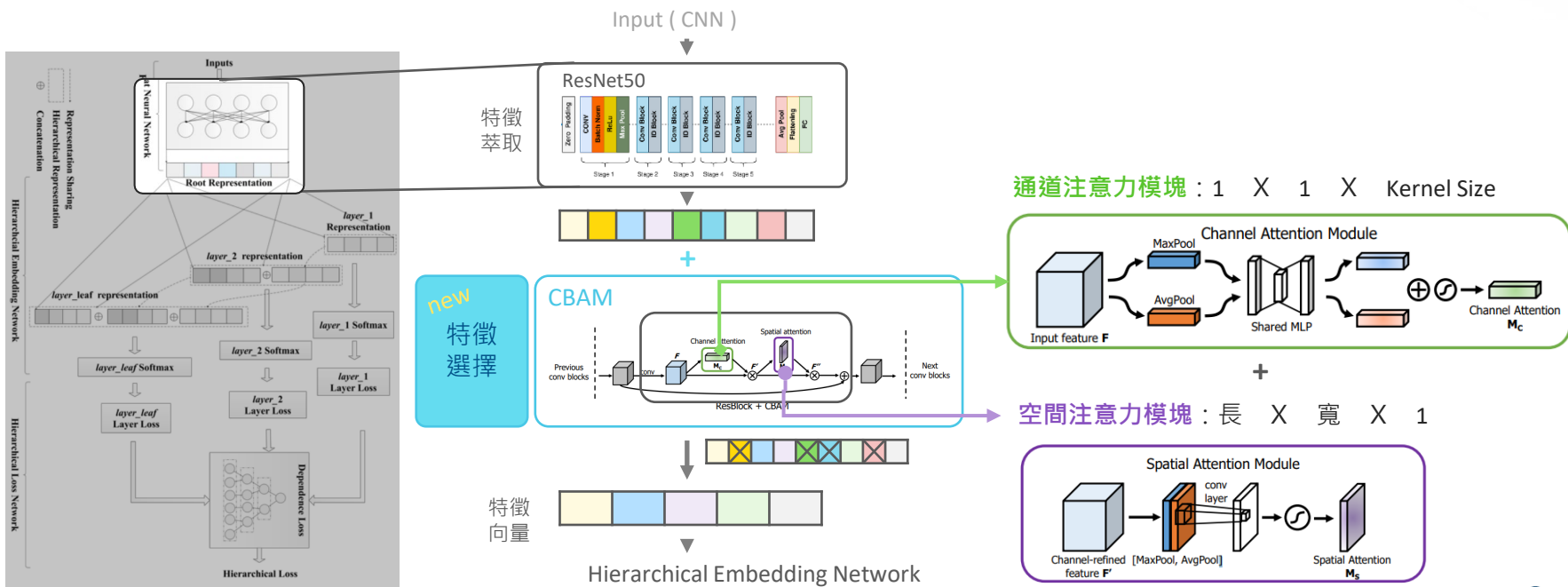
根據 Paper 選擇在 ImageNet 上表現較佳的模型進行後續實現。

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

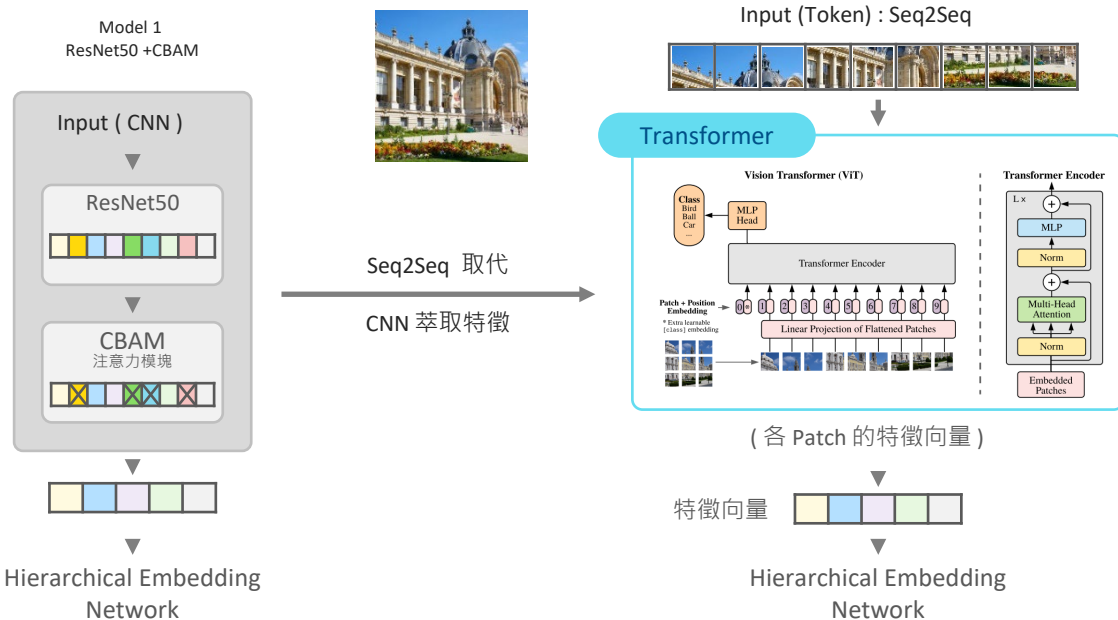


Vision Transformer

→ 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)

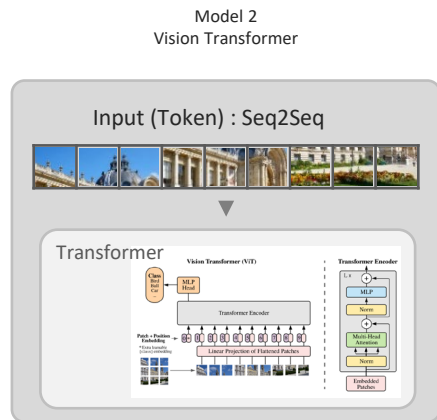
Transformer Encoder 機制

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

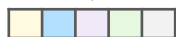
ConvNeXt

→ 基於ResNeXt架構保留CNN卷積優點，並加入 Swin Transformer 的模型注意力機制，達到縮小模型計算量

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

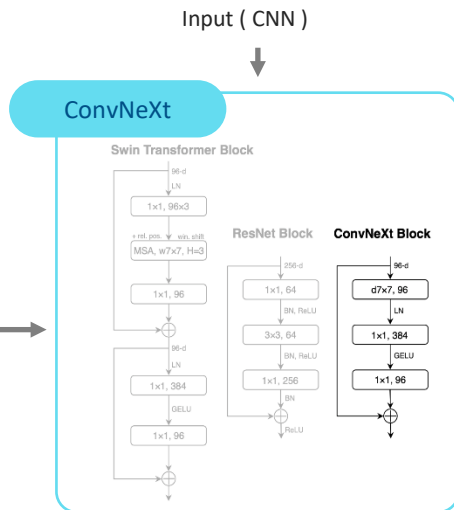


(各 Patch 的特徵向量)



Hierarchical Embedding
Network

基於ResNeXt CNN架構



特徵向量

Hierarchical Embedding Network

CNN 平移不變性

從 ResNeXt 模型修改，保留滑動窗口機制

Transformer 注意力機制

ResNeXt Block 採用depth-wise conv和 1*1 conv 的組合，相當於做空間與通道的注意力權重機制。

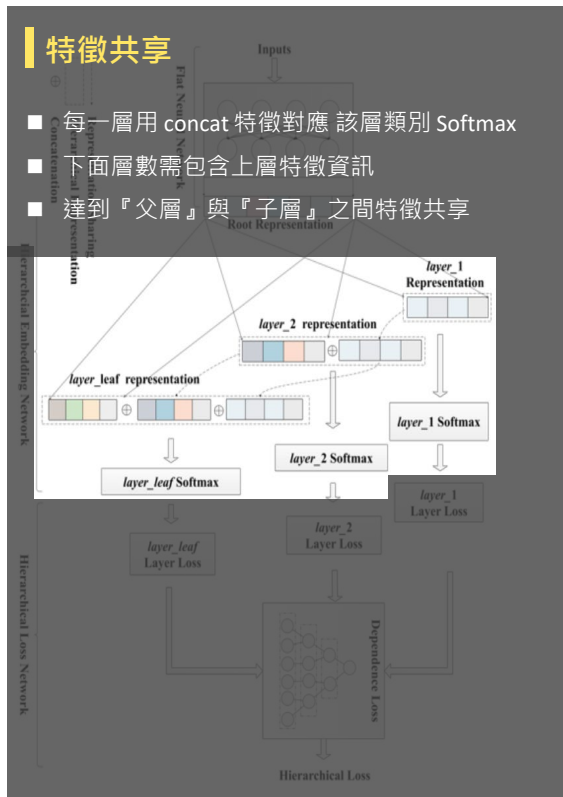
縮小模型計算量

反轉 Bottleneck 使隱藏層尺寸為輸入層尺寸之4倍。

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

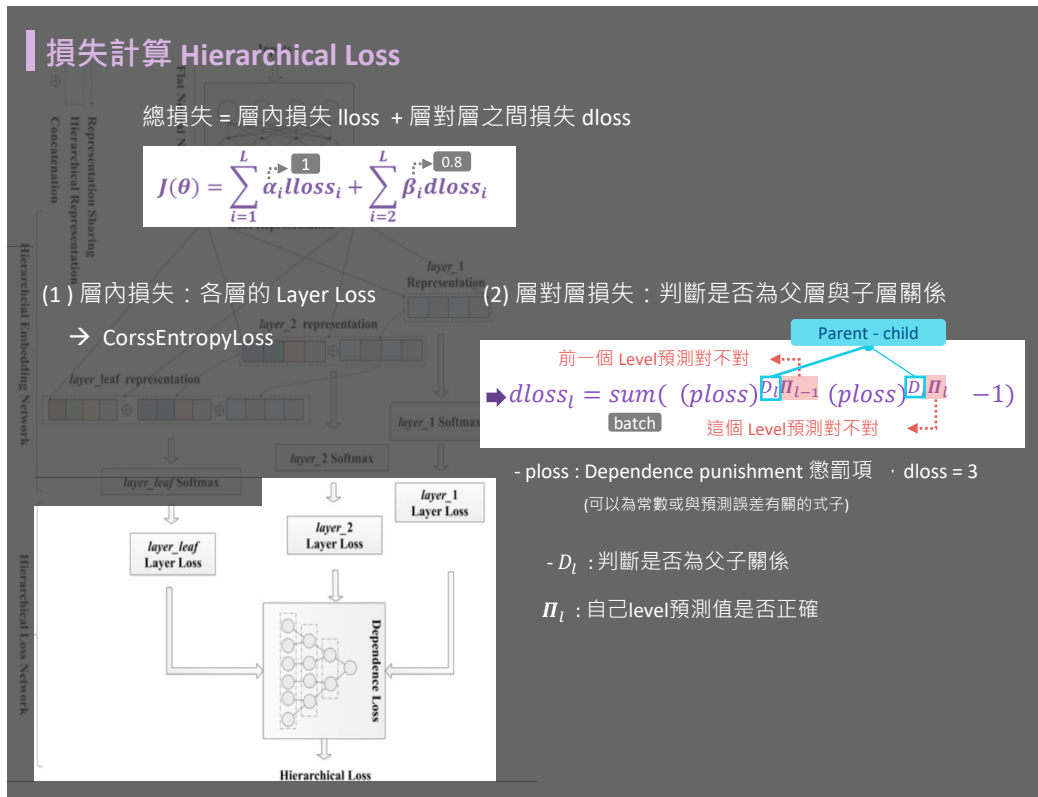
$$dloss_i = \sum_{\text{batch}} (ploss) D_i \pi_{i-1} (ploss) D_i \pi_i - 1$$

Diagram illustrating the calculation of the layer-to-layer loss (dloss). It shows a 'Parent - child' relationship between two levels. The loss is calculated based on the 'ploss' (Dependence punishment) and the 'D' (judgment of parent-child relationship) and 'π' (self-level prediction value) for both levels. The formula is: $dloss_i = \sum_{\text{batch}} (ploss) D_i \pi_{i-1} (ploss) D_i \pi_i - 1$.

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

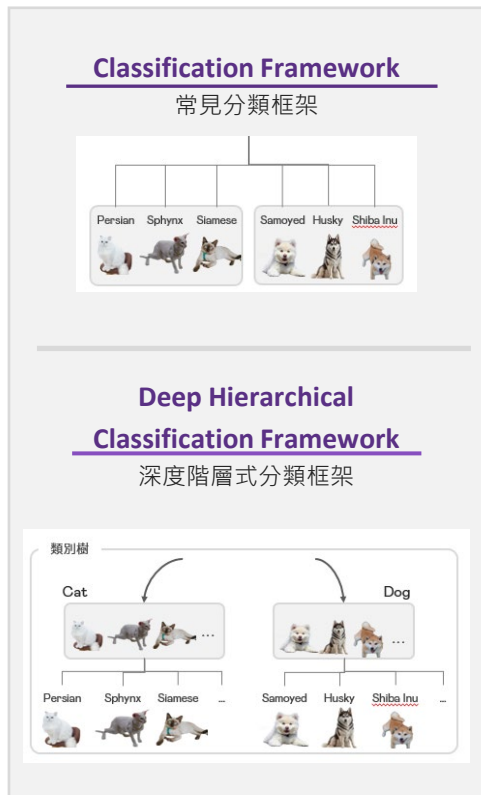
- D_i : 判斷是否為父子關係

π_i : 自己level預測值是否正確

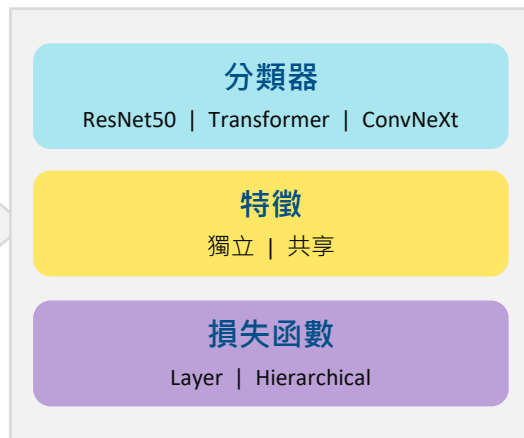


實驗架構

分類框架



論文架構



實驗組合

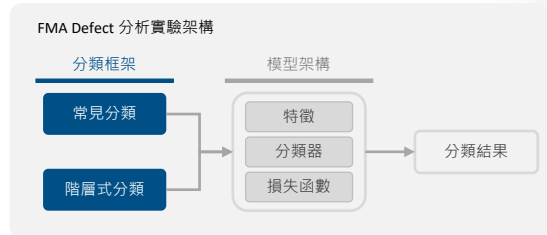
分類器	Type	特徵	損失函數
ResNet50	一般框架	獨立	Layer
Transformer	階層式框架	獨立	Hierarchical
ConvNeXt		共享	Hierarchical

結果分析 & 討論

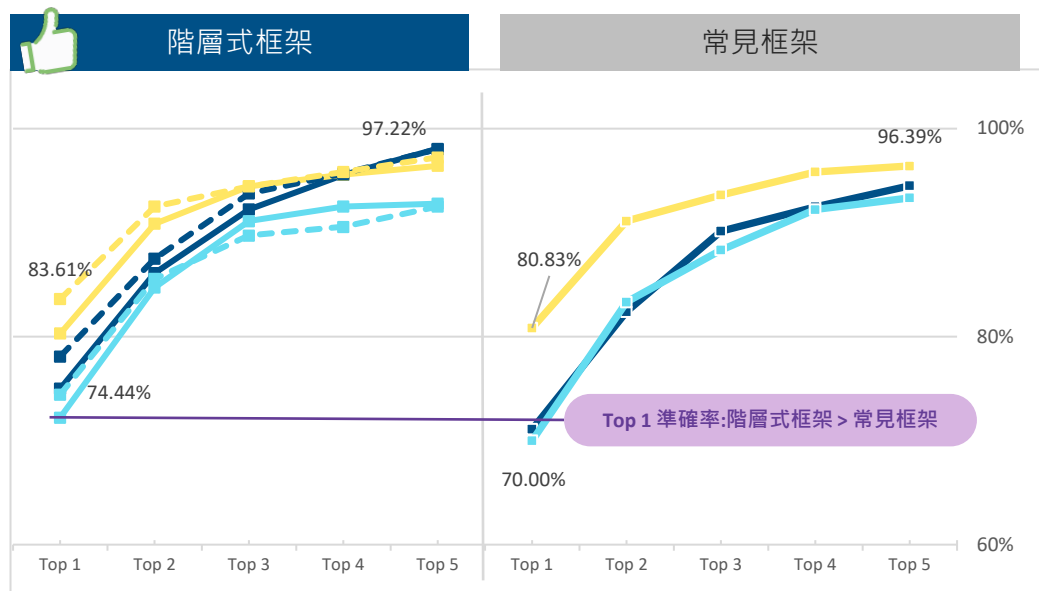
框架分析

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



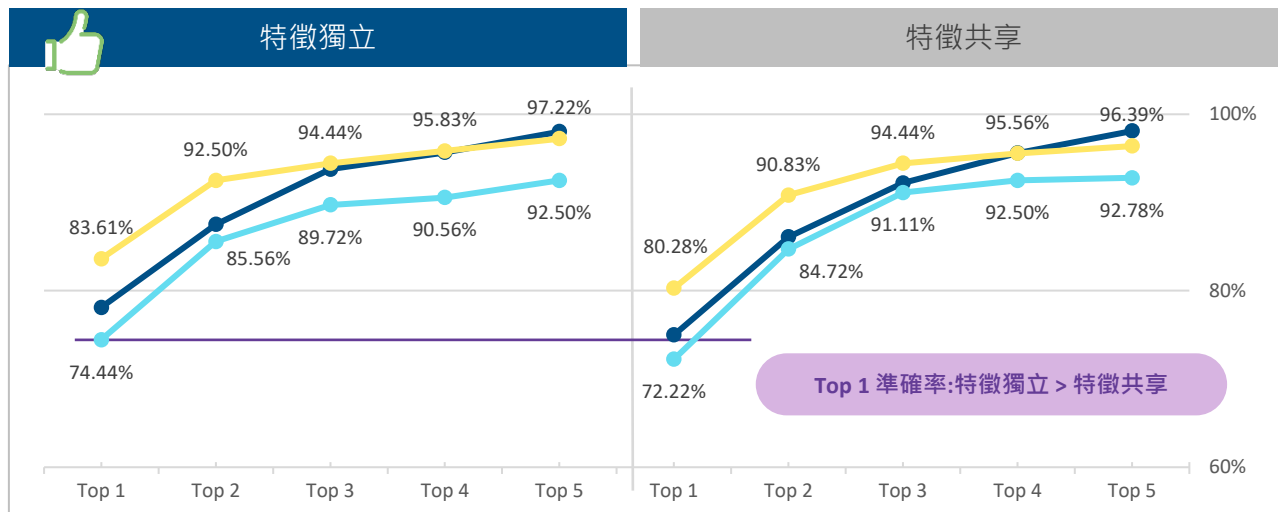
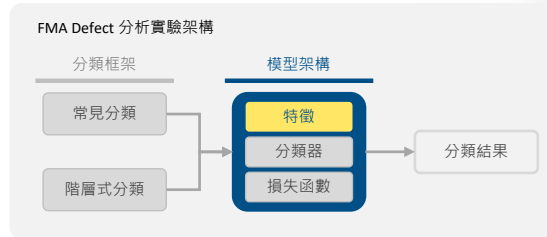
分類器			
	ResNet50	Transformer	ConvNeXt
Type	特徵	損失函數	圖例
一般	獨立	Layer	——
階層式	獨立	Hierarchical	——
	共享	Hierarchical	- - -



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

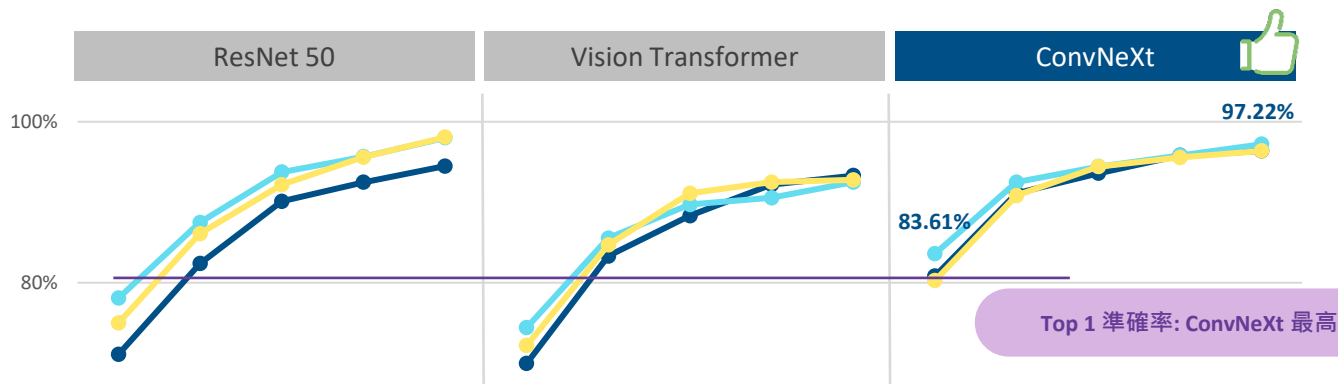
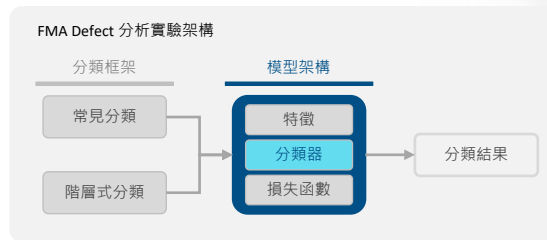


模型分析

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

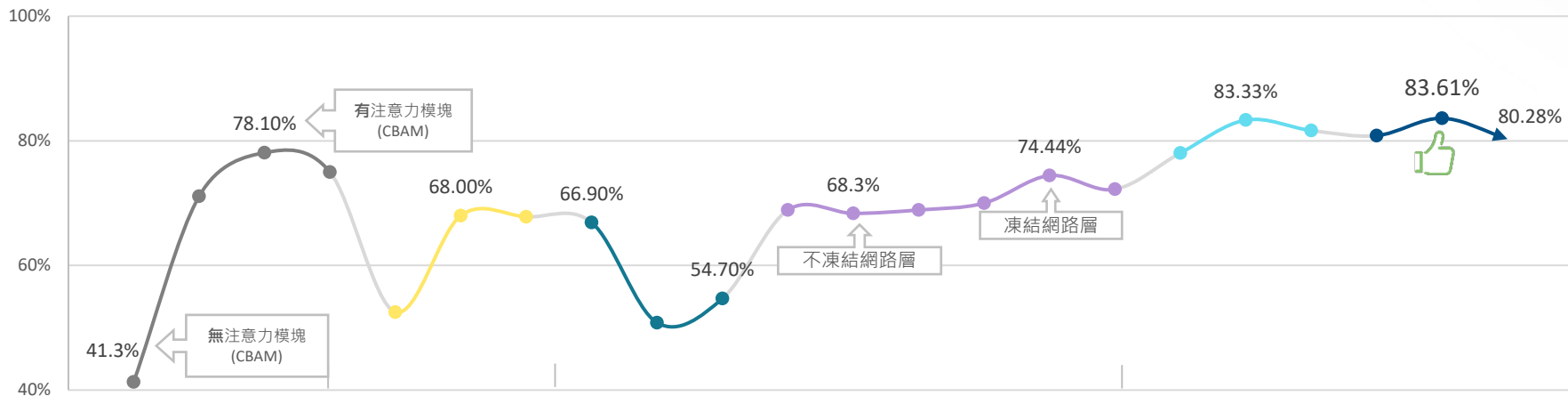


Top 1 準確率: ConvNeXt 最高

	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

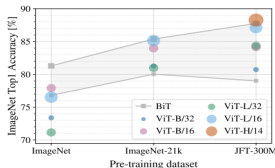
訓練花費時間長
(約2~3天)

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

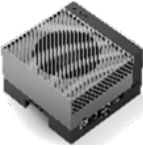
1. 訓練花費時間短 (約4小時)
2. 模型大小 (Tiny > Small): 目前收集資料量Tiny 即可
3. 準確率表現最佳 → 由此模型繼續往下分析

訓練環境與設備

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

■ ConvNeXt 訓練結果：準確率：第一層(DefectLocate) **100%**；第二層(LocationFlag) **92.78%**；第三層(FMA Defect) **83.61%**

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

模型架構平展

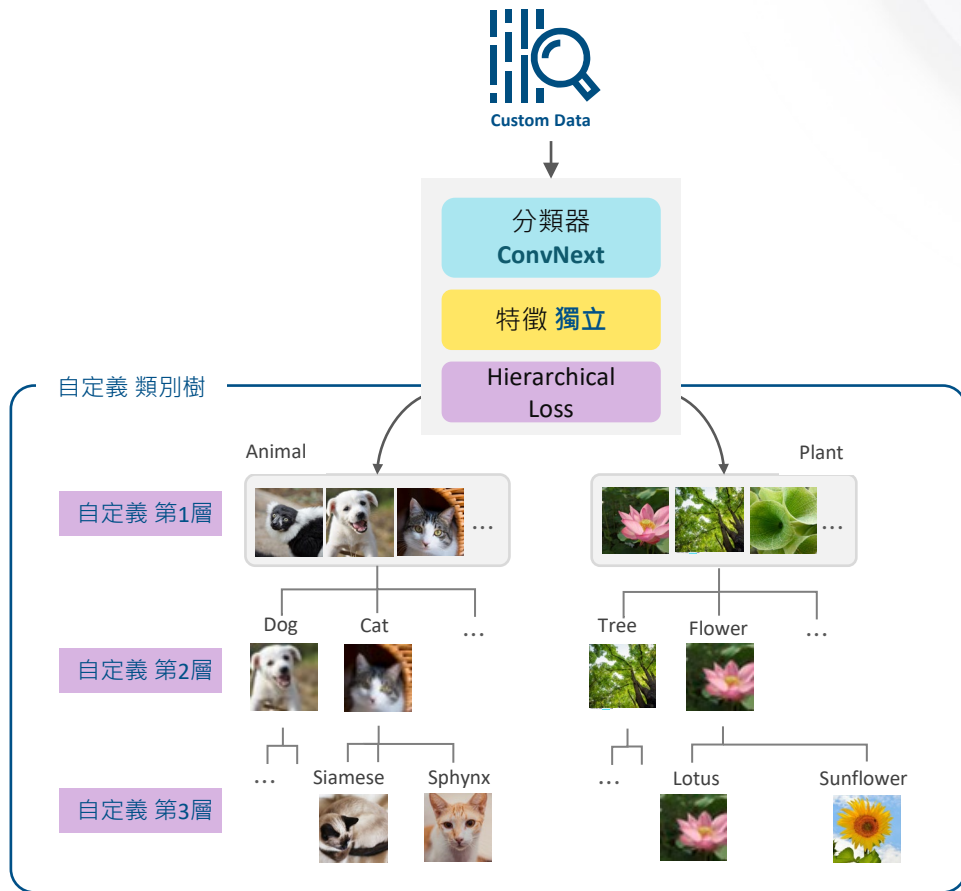
階層式分類框架優點

場域應用

1. 使用者帶入「自定義資料集」，定義類別樹父子層關係
2. 可判別影像的多層分類，模型提供「多輸出」
3. 在信心度未達標準時，可以先提供粗略答案 (1 & 2層)

模型框架

1. 階層框架更符合真實世界「語意關係」，適用於專案場景
2. 一個模型即可預測多層分類結果，訓練效果好，每個階層有對應的損失函數 (Loss Function)



未來展望

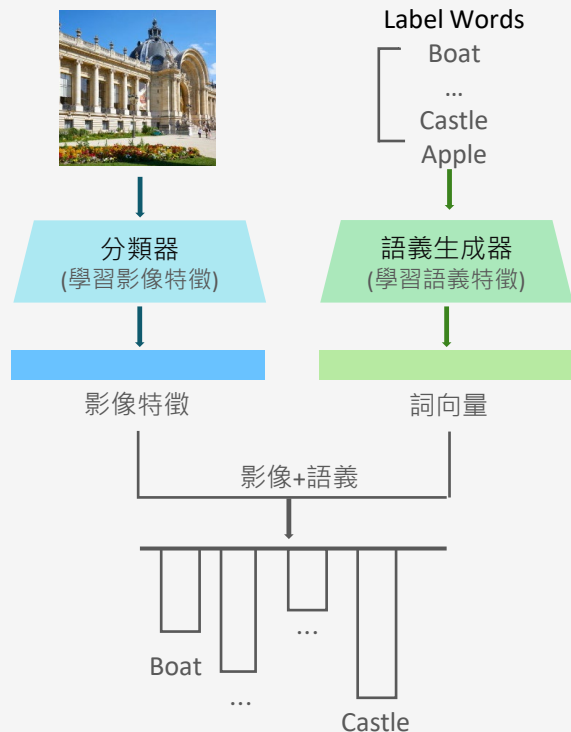
場域應用

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

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

