

基於深度學習技術提升 PCB 線路瑕疵檢測在 AOI 應用下的錯誤預警能力

陳昱奇、蔡沐霖、吳偉民、冷堂愷、劉晏誠、林鼎鈞、吳志宏

國立高雄大學電機工程學系

{a1105155|b1105129|b1115157|b1115112|a1095161|a1115162}@mail.nuk.edu.tw

、johnw@nuk.edu.tw

摘要

本研究旨在基於深度學習技術，改善印刷電路板 (Printed Circuit Board, 以下簡稱 PCB) 中線路的瑕疵檢測問題，以及提升在自動光學檢測 (Automated Optical Inspection, 以下簡稱 AOI) 應用中的錯誤預警能力。本次研究中使用黑白 PCB 資料集，包含六種常見瑕疵：copper、good、mousebite、open、pin-hole、short 及 spur。另外為了解決實際應用中可能發生的瑕疵資料集數量不足問題，我們團隊提出透過生成對抗網絡 (Generative Adversarial Network, 以下簡稱 GAN) 擴充資料集，並設置多種評估指標以確保生成影像的可靠性與實用性。

模型訓練部分，本研究比較多個常見的神經網絡模型，包括 AlexNet、ConvNet、EfficientNet、GoogLeNet、MNASNet、MobileNet、ResNet、SqueezeNet、VGG、ViT 及 Xception，並調整學習率 (learning rate)、訓練輪數 (epoch)、批量大小 (batch size) 等參數，尋找最佳配置。對實驗中各模型的訓練結果進行觀察，結合過程中記錄的各種指標，進行綜觀的比較與分析，以進一步評估其在 AOI 應用中的實際效果。

關鍵字：深度學習、印刷電路板、自動光學檢測、瑕疵檢測、生成對抗網絡。

1. 緒論

PCB 是電子產業的關鍵元件，其品質直接影響產品性能。根據台灣電路板協會 (TPCA) 最新統計，2024 年上半年台商 PCB 全球產值達到新台幣 3722 億元，較 2023 年同期成長 6%。預計 2024 年全年產值將達 8337 億元，年增幅約 8.3%。

PCB 製造過程會遇到的問題十分繁雜，且製程中每個步驟都可能產生瑕疵，若未及時檢測，將導致產品效能下降甚至重大損失。因此，高效且精準的瑕疵檢測是重要的研究方向之一。隨著深度學習技術的快速發展，卷積神經網絡 (CNN) 已成為影像分類的主要技術。從 ResNet 到 MobileNet，再到新架構的 ViT (Vision Transformer) 提出，多種模型在結構設計與計算效率上都各具特色，且面對不同應用場景與數據特性時，表現可能存在不同的擅長區間，因此有必要對多種模型進行系統比較，以辨識其適用性與優勢。

本研究結合深度學習技術，針對 PCB 外觀瑕疵檢測的問題，訓練不同模型進行比較，再加上生成對抗網絡 (GAN) 技術的引入，可以通過數據庫增列解決資料分布不均的問題，以求進一步提高模型效能。

2. 研究背景

卷積神經網絡 (Convolutional Neural Network, 以下

簡稱 CNN) 在電腦視覺中佔有非常重要的地位，在影像辨識上的精準度可以達到超過人類的水準，是目前深度學習發展的一大主力。AlexNet [1] 為 2012 年被提出的卷積神經網絡，是同年的 ImageNet LSVRC 競賽的冠軍，造成很大的轟動，至此 CNN 廣泛地被研究。AlexNet 問世之前，深度學習冷門了一段時間。因為 LeNet 在早期的小數據集可以取得好的表現，但在大的數據集上表現很差，因此許多人改鑽研其他的機器學習方法。而 AlexNet 的出現，可以說是時代的分水嶺，正式開啟了 CNN 的時代。

AlexNet[1] 採用 Kernel Size 3x3、stride 為 2 的 Maxpooling。比 LeNet 的 meanpooling，更能保留重要的特徵。且 stride < size (2<3)，因此 pooling 可以重疊，能重複檢視特徵，避免重要的特徵被捨棄。由於輸入層較大，可以輸入 224x224 尺寸的彩色圖片，可學習較大的局部特徵。Activation function 由 LeNet 使用的 Sigmoid 改為 ReLU，可以避免因為神經網絡層數過深或是梯度過小，而導致梯度消失並且相較於早期使用的 Sigmoid/Tanh 收斂速度較快。並採用了 dropout 及資料擴增以降低 overfitting。

VGG[2] 比 AlexNet 採用更深的網路，特點是重複使用一組 module，並改用小 kernel 替代 AlexNet 裡的大 kernel。VGG 使用多個 VGG Block(由不同數量的 3x3 Convolutional kernel 以及 2x2 的 Maxpooling 組成)，代替 AlexNet 的 5*5 卷積核，可達到相同的 receptive field，同時減少參數量。

GoogLeNet [3] 使用 1x1 卷積作為 dimension reduction 來減少計算量。藉由減少計算 bottleneck，可以增加網絡深度和寬度。在淺層的神經網絡，分類更加準確，所以網絡中間所產生的特徵圖有其分辨性，因此使用這些特徵圖做分類，增加額外的分支。好處為加強 gradient 的回傳以及額外的 regularization。

ResNet [4] 為了減輕越深的神經網絡訓練難度，作者想到恆等映射 (Identity Mapping) 來解決。輸入等於輸出，所以不管網路再怎麼深，都不應該比什麼權重都沒乘來的差。然而實驗發現，嘗試用恆等映射的方式來訓練深層神經網絡，幾乎無法在合理時間內找到足夠好的準確率。接著作者提出一種「殘差映射」(Residual Mapping) 訓練深層神經網絡，作者優化「殘差」這件事情，會比直接學習恆等映射要來的容易。在大於 50 層的網路，作者提出 Bottleneck 的設計，來節省大量的參數。透過 1x1 的 conv 降維後再升維的技巧，架構可以兼顧更高維度與效能。

SqueezeNet[5] 透過將 3x3 filters 取代為 1x1 filters 並

減少 3×3 filter 的 input channel 數，最後在網絡的後期進行 Downsample，使卷積層有較大的 activation map。藉此減少 CNN 中的參數量，同時保持 accuracy。與 AlexNet 相比，使用 SqueezeNet 尺寸減少了 50 倍，同時達到 AlexNet 的 top-1 和 top-5 accuracy。

Xception[6]Depthwise Separable Convolutions：傳統的 CNN 由於 Filter 都是使用所有通道（深度），運算量會較多，透過先進行 Depth-wise convolutions（通道分離）再進行 Point-wise Convolutions（維度調整），可以達成降低運算量的目的，且與傳統的 CNN 相比，運算量大幅減少，並且 Performance 下降的幅度相當小，使其在 Edge device 有相當大的優勢。作者基於通道及空間可分離的特性提出假設，透過觀察後發現 Inception 介於兩種極端的中間（CNN & Depthwise Separable），從而將 Inception Module 改為 Depthwise Separable Convolution 並在 ImageNet 上有了小幅的提升，也證明其效用較之 InceptionV3 在相同的參數量下，能更有效的利用參數，同時架構的架設也更為簡易。

MobileNet[7] 使用 Depthwise Separable Convolution 建構神經網路以及引入兩個超參數 Width Multiplier、Resolution Multiplier 讓開發人員可以依據自己的應用與資源選擇合適的模型。Depthwise Separable Convolution 包含了深度卷積（depthwise convolution）和逐點卷積（pointwise convolution），與一般的卷積操作不同，核心思想就是把卷積過程的計算量減少。一般卷積的 filter 是用在所有的輸入通道上（input channels），而 depthwise convolution 則對每個輸入通道採用不同的 filter，也就是說一個 filter 對應一個輸入通道。每個輸入通道做完 depthwise convolution 後，再採用 pointwise convolution 將輸出融合，其實就是使用 1×1 的卷積核做計算。如此的操作，輸出和一般的卷積層差不多，但是大大的降低了計算量及模型參數量。

EfficientNet[8] 作者認為應該要同時調整這三種縮放方法，因複合縮放（Compound Scaling）：網路深度、寬度、解析度是相互影響的。在不同網路深度、解析度下寬度縮放的準確度比較，作者得到了一個結論：為了追求更高的準確率，平衡網路深度、寬度、解析度是很重要的關鍵。作者使用一組固定的縮放係數（就是複合係數 ϕ ）將網路模型均勻地縮放，此超參數須自己定義，是用於控制能使用多少資源在模型縮放上。該研究發現平衡這三種縮放能夠獲得更好的性能，因為他們之間會相互影響：對於解析度高的圖像，使用更深的網路、獲取更大的感受野，能夠有更好的效果，同時增加網路寬度可以得到更細度的特徵。受 MnasNet 的啟發，作者利用多目標神經結構搜索產生高效能的模型 EfficientNet-B0，使用的搜索空間與 MnasNet 相同，但優化目標稍有不同：MnasNet 是優化延遲，而 EfficientNet 則是優化 FLOPS。

Transformer[9]Transformer 如今已成為熱門的神經網路架構，並且大量的應用在自然語言（NLP）任務。它的成功追朔於 2017 年 Google 所提出的 Attention Is All You Need。這樣的重大突破使得 Google 團隊將這一套 Transformer 架構中的 Encoder 抽離出來變成 Vision Transformer（ViT）應用在影像分類技術上。此外它拋棄了

CNN 層並以 self-attention 計算做取代，並在分類問題上取得不錯的成績。

GAN[10] 是由兩個神經網絡所構成的，一個叫做 Generator（生成器）另一個叫做 Discriminator（判別器）。Generator 做的事情就是想辦法把一堆雜訊變成看起來很逼真的照片，而 Discriminator 做的事情就是想辦法去判別說哪個是真的照片哪個是從 Generator 生成的假照片。所以在訓練的過程中，Generator 生成的照片會越來越逼真，想辦法騙過 Discriminator，而 Discriminator 也會想盡辦法去學習辨識真的和假的照片。而這樣兩個神經網絡對抗的過程，我們叫做 Adversarial Learning。所以我們的最終目標就是讓 Discriminator 完全無法區別哪個是真的照片哪個是假的照片。

GAN 的優點也是他的缺點，它繞過了學 Representation 這個概念，GAN 的學習目標就是直接想辦法生出一堆看起來很像真實照片的假照片，所以 GAN 的學習過程屬於隱式的，也就是說他沒有學到真正的 Representation，所以很難和文字、語音做整合，但是也因為這樣 GAN 的訓練、推論速度比其他生成模型，如 Diffusion model 來的快。

3. 研究方法

A. 資料前處理

1. 原始資料集裁切

本研究使用 3000 張黑白一整塊的 PCB 影像作為資料集 [11]，每張影像均包含瑕疵（defect）與無瑕疵（good）兩種版本。根據資料集的特性，對同一塊 PCB 影像中帶有瑕疵與無瑕疵的相同區域進行裁切，提取出多個小區塊，標註為瑕疵類別或無瑕疵類別。一張 PCB 影像可以裁切出多個小區塊，經過資料處理後，最終生成約 20000 張裁切的 PCB 區塊影像，作為訓練分類模型的資料來源。

針對數量較少的瑕疵類別，透過資料增強技術進行平衡處理，包括旋轉、翻轉、添加高斯雜訊以及模擬光源變化等操作，以使各類別的樣本數量達到均衡。

最終，模型的訓練任務為區塊分類，分類模型需辨識出 6 種類別的瑕疵（copper、mousebite、open、pin-hole、short、spur）以及 1 類無瑕疵的區塊（good），以實現高準確率的瑕疵檢測目標。

2. GAN 擴充資料集

本實驗另外利用 GAN 生成六種瑕疵的資料集，目的是為了提高瑕疵的多樣性。GAN 所使用的超參數如下：

Latent dim = 100
Image size = 28
Batch size = 64
Num epochs = 160
Learning rate = 0.0002
Betal = 0.5
Num images = 700

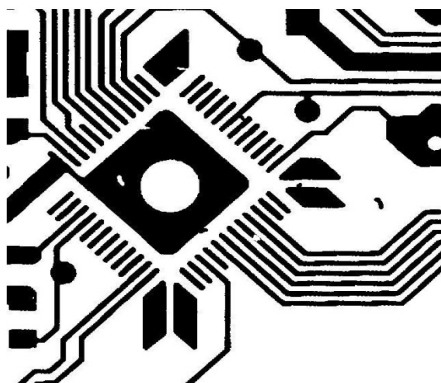


圖 1: 帶有瑕疵的黑白 PCB 影像

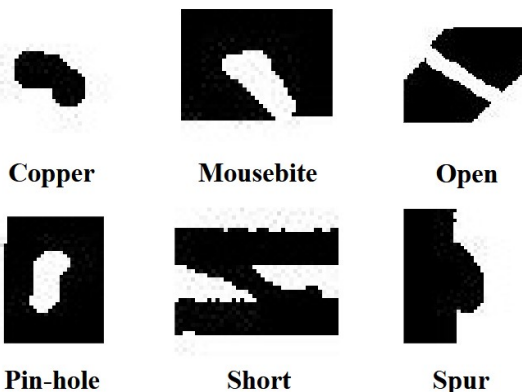


圖 2: 切割出來的 6 種 PCB 瑕疵區塊

其中 Num epochs 設為 160 是因為訓練過程中，Discriminator loss 與 Generator loss 約會在此時達到收斂，因此不再運算過多的 epoch。Num images 是自己定義的參數，用於指定要為各個瑕疵分別生成幾張圖，這裡設 700 就是為六種瑕疵各生成 700 張圖，共生成 4200 張圖。接著會人工剔除無法分辨其瑕疵的圖片（圖1，無法分辨瑕疵為何），或是不符合瑕疵原有特性的圖片（圖2，可以看出是 mouse bite，但不符合 mouse bite 要有一個缺口，此圖缺口是封閉的）。本資料集中，copper 共剔除 36 張圖片；mouse bite 共剔除 198 張圖片；open 共剔除 106 張圖片；pin-hole 共剔除 42 張圖片，short 共剔除 85 張圖片；Spur 共剔除 109 張圖片。

3. K-fold

為 10 個 Fold，以避免特定資料對模型造成偏誤的影響。實驗中選取其中的 3 個 Fold 作為訓練集驗證測試集，目標提升模型評估的穩定性與準確性，以及降低實驗的誤差。

B. 資料集多樣性與 GAN 生成圖像檢測

1. K-means

在剔除完之後，我們利用 K-means 來可視化資料分佈，其運作方法為將高維度的圖像降維後做聚類，目的是為了得知各特徵的分佈情況，可以用於比較原資料集與加入生成圖片後的資料夾是否有很大的差距。由各特徵分佈



圖 3: GAN 無法分辨瑕疵為何



圖 4: 可以看出是 mouse bite，但不符合 mouse bite 要有一個缺口，此圖缺口是封閉的

位置可以看出，增加 GAN 生成圖片後的資料集並不會破壞應有的分佈狀況，且生成圖片的分佈位置皆與原資料集的圖片相近。

2. FID

利用 FID 來評估計算真實圖片與生成圖片的特徵向量之間的距離，其公式如下：

$$FID = \|\mu_1 - \mu_2\|^2 + \text{Tr} \left(\Sigma_1 + \Sigma_2 - 2(\Sigma_1 \Sigma_2)^{\frac{1}{2}} \right) \quad (1)$$

1 為真實圖像，2 為生成圖像， μ 為特徵平均值， Σ 為特徵共變異數，Tr 為矩陣對角線元素的總和。FID 值越接近 0 越好，表示生成圖像與真實圖像分佈完全一致。本實驗中使用 inception-v3 網路來提取圖像特徵，計算出的結果如表一。由此可以得知，生成出來的圖片可能與原始圖片有細小的差異，但整體而言都很接近原始資料集的特徵。

瑕疵特徵	FID 值
spur	23.30
short	36.35
pin-hole	28.85
open	24.82
mousebite	15.36
copper	43.80

表 I: 瑕疵特徵及對應的 FID 值

C. 分類模型訓練

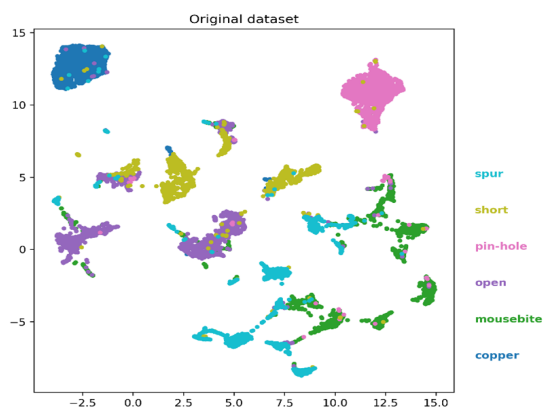


圖 5: 原資料集

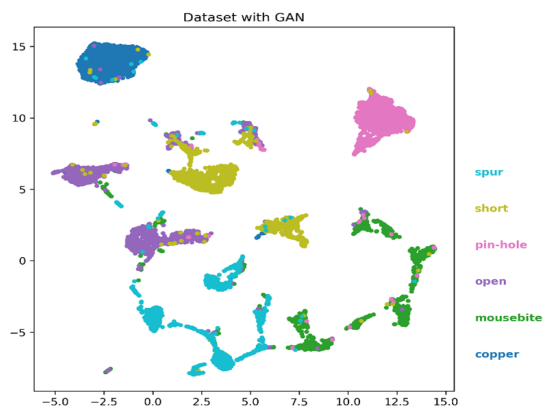


圖 6: 增加 GAN 生成圖片後的資料集

分別以六組參數：batch size 為 8、32、64，以及 learning rate 為 $1e-3$ 、 $1e-4$ 、 $2e-5$ ，對使用及未使用 GAN 的資料集進行分類模型的訓練，在訓練過程中以 50 個 epoch 為基準，逐一紀錄 epoch 為 10、30、50 的 confusion matrix、train/valid loss curve、train/valid accuracy curve，最後也紀錄 precision、recall、F1 score、mPA、test loss、test accuracy 於 result.csv 中。

D. 訓練結果評估

以模型訓練得到之 confusion matrix、train/valid loss curve、train/valid accuracy curve 以及 result.csv，分析個別模型對於此應用場景下的訓練與辨識效果，並觀察使用 GAN 擴充資料集是否會對模型結果產生顯著的影響。

4. 實驗結果

A. AlexNet

原始資料訓練特點

- 1) Batch Size 值較小時，Epoch 的增加 Loss 無明顯下降，表現差不多。
- 2) Batch Size 值較大時，Epoch 的增加 Training Loss 明顯下降。

加入 GAN 的訓練結果

- 1) Epoch 值較小時，Loss 表現與原始資料差不多。

- 2) Epoch 值較大時，Loss 表現能看出比原始資料好一點。

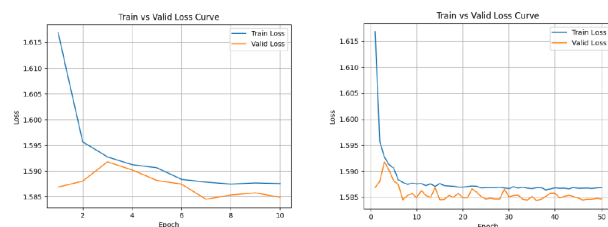


圖 7: AlexNet B8 Lr1e-3 Loss 曲線
(左:epoch10, 右:epoch50)

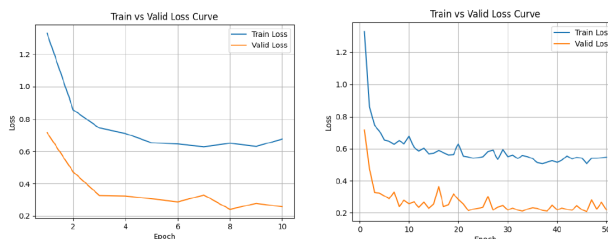


圖 8: AlexNet B32 Lr1e-3 Loss 曲線
(左:epoch10, 右:epoch50)

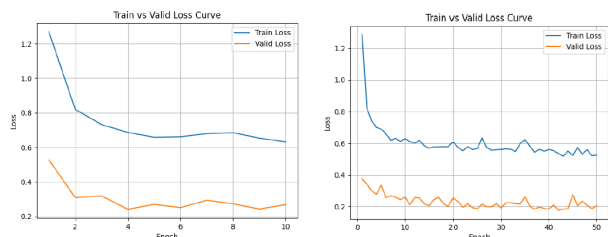


圖 9: AlexNet B32 Lr1e-3 GAN Loss 曲線
(左:epoch10, 右:epoch50)

B. VGG

此模型在 learning rate 為 $2e-5$ 時訓練成效最佳，而在 learning rate 為 $1e-3$ 及 $1e-4$ 時無法正確檢測各瑕疵。learning rate 為 $1e-3$ 及 $1e-4$ 的訓練結果：

- 1) train loss 與 valid loss 約會停留在 1.58~1.595 之間，如圖10 (learning rate = $1e-3$, batch size = 8)、圖11 (learning rate = $1e-4$, batch size = 8)。
- 2) 兩者並無法正確檢測出瑕疵，由 confusion matrix (圖12) 可以看出，其檢測結果皆集中於 good 上。
- 3) 增加了 GAN 生成的資料集後，其 train loss 與 valid loss 約會停留在 1.65 1.665 之間，如圖13 (learning rate = $1e-3$, batch size = 8)、圖14 (learning rate = $1e-4$, batch size = 8)，且依然無法正確檢測出瑕疵，confusion matrix 與可視化結果如圖15、圖16所示。
- 4) 原資料集的 Precision 固定為 0.0715，增加 GAN 生成圖片的資料集 Precision 固定為 0.0662，Recall 皆固定為 0.1429。

learning rate 為 $2e-5$ 的訓練結果：

- 1) train loss 會收斂至 0.01 以下，valid loss 收斂的位置，batch size 為 8 時，會比 32 與 64 還小，但皆在 0.01 附近，如圖17 (batch size = 8)
- 2) 資料集與增加 GAN 生成的資料集的準確率皆能達到 97.5% 以上。
- 3) 增加 GAN 生成圖片的資料集不一定會較準確，有時會比原資料集大，有時較小，但若將所有 fold 中的各個 batch size 的 test accuracy 加總做平均計算，會發現增加 GAN 生成圖片的資料集平均 (98.267%) 會比原始資料集 (98.483%) 低。此 learning rate 的 confusion matrix 與可視化結果如圖18、圖19所示。
- 4) Precision 與 Recall 皆能達到 0.95 以上。

不同 learning rate 下表現懸殊的情況，推測原因有二：

- 1) 資料集的分布較不平均，good 資料集的總數明顯比其他資料集大上許多，因此訓練的模型可能較無法分辨各種瑕疵，這也能解釋為何 confusion matrix 上的辨識結果都會集中在 good 上。
- 2) VGG 模型訓練的 learning rate 不能太大，由設定的三個參數中可以得知，最小的 2e-5 表現明顯比 1e-3、1e-4 優異。

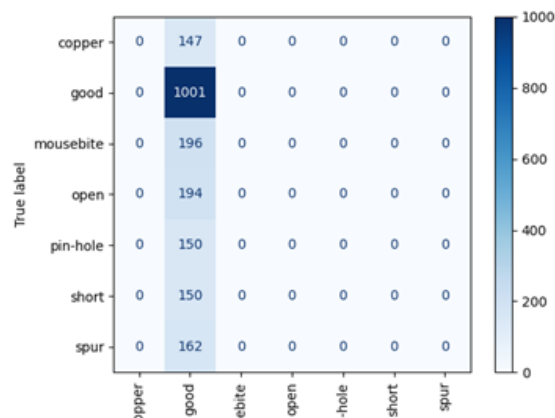


圖 12: VGG confusion matrix

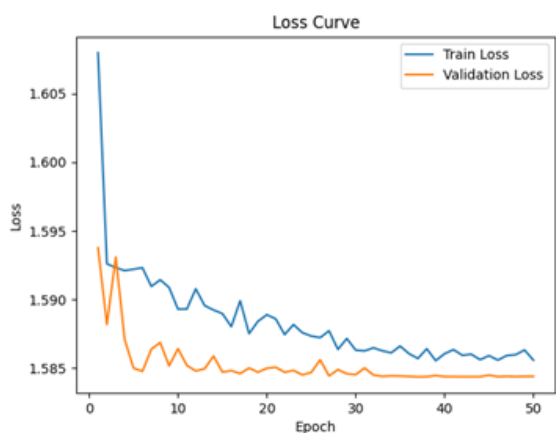


圖 10: VGG B8 Lr1e-3 Loss 曲線

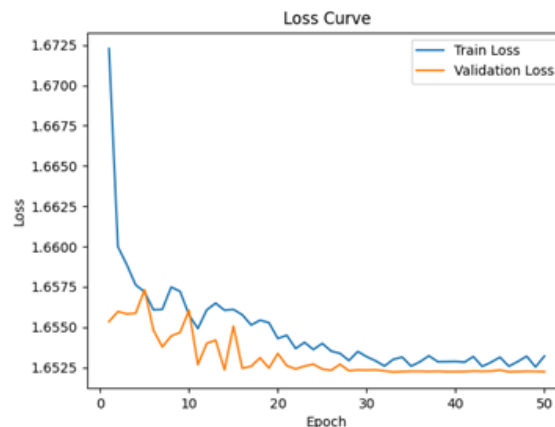


圖 13: VGG B8 Lr1e-3 GAN Loss 曲線

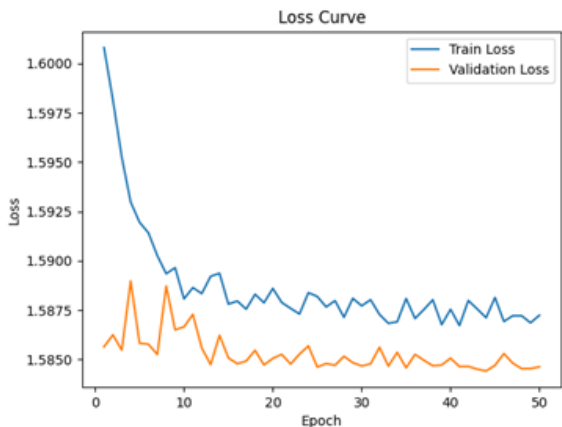


圖 11: VGG B8 Lr1e-4 Loss 曲線

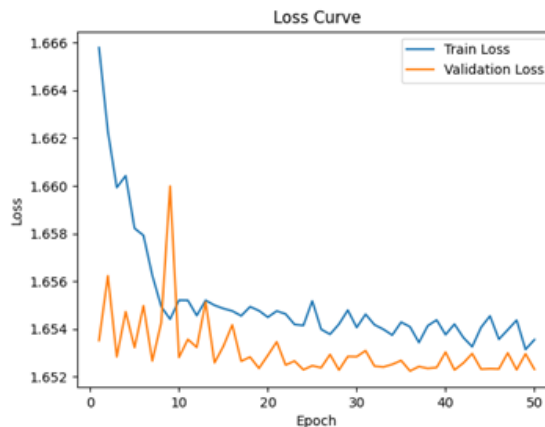


圖 14: VGG B8 Lr1e-4 GAN Loss 曲線

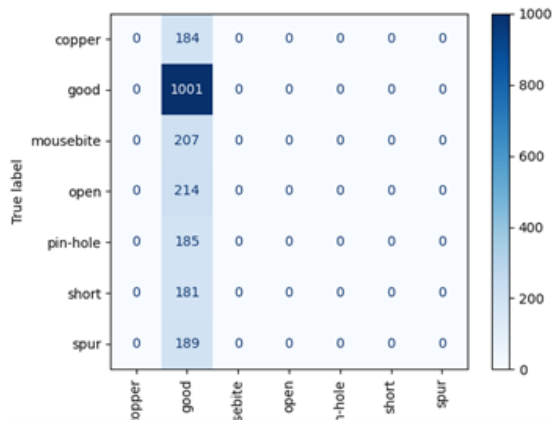


圖 15: VGG GAN confusion matrix

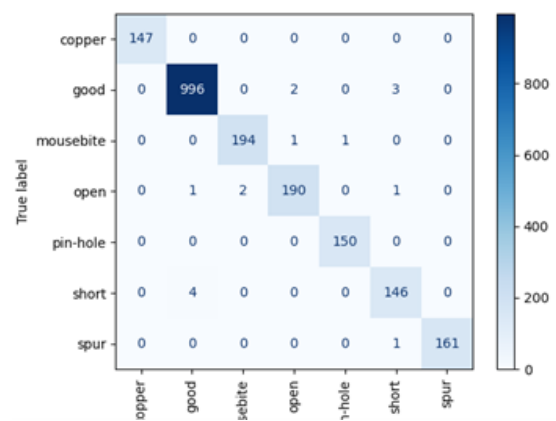


圖 18: VGG B8 Lr2e-5 GAN confusion matrix

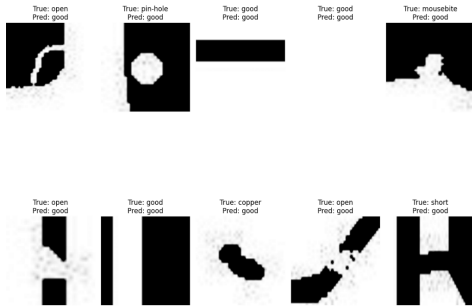


圖 16: VGG GAN 可視化結果

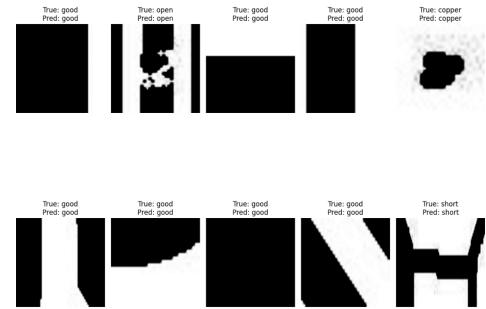


圖 19: VGG B8 Lr2e-5 GAN 可視化結果

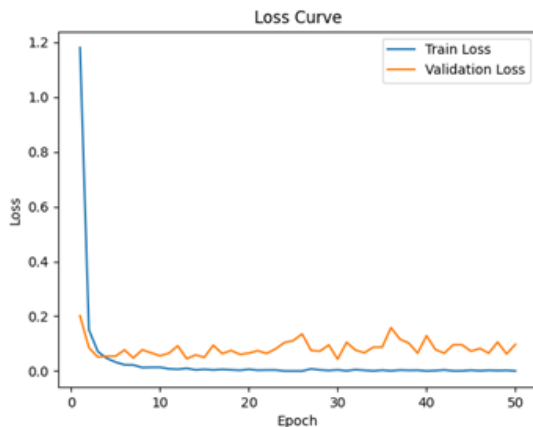


圖 17: VGG B8 Lr2e-5 GAN Loss 曲線

C. GoogLeNet

在 GoogLeNet 的模型訓練當中，觀察 accuracy、loss 曲線：

- 1) 隨著 Batch size 的減少，在 Batch size 64 時 loss 會有很大的震盪，在 Batch size 32 時，震盪會收斂變小，而在 Batch size 8 會呈現平滑。
- 2) 準確率也隨 Batch size 減少而震盪減少，在 Batch size 64 會有很大的震盪。
- 3) Train loss 在 epoch 10 20 左右就會收斂。

在 GoogLeNet 的模型訓練當中，觀察 confusion matrix 與其指標：

- 1) 在 Batch size 8 和 Batch size 32 時，多數特徵會辨識為 good。
- 2) 承上點，若這種狀況發生時，瑕疵的 precision 會較高，而 recall 會較低，和瑕疵檢測所需要的高 recall 矛盾，所以應該避免此類參數。

加入 GAN 之後的變化：

- 1) 加入 GAN 之後，還是會有震盪，只是變成集中在 Epochs 30 以前，過 Epoch 30 之後震盪明顯變小。

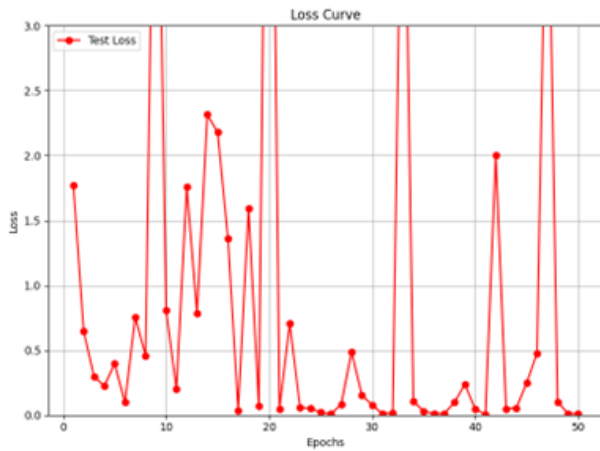


圖 20: GoogLeNet loss 曲線

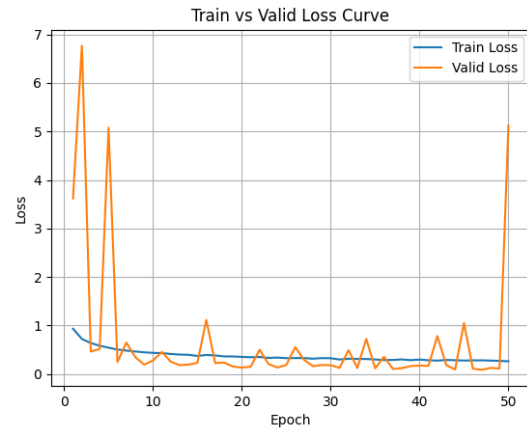


圖 22: ResNet 未收斂 loss 曲線

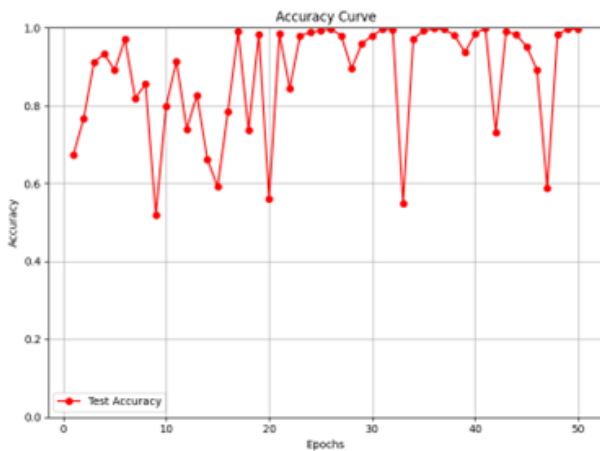


圖 21: GoogLeNet accuracy 曲線

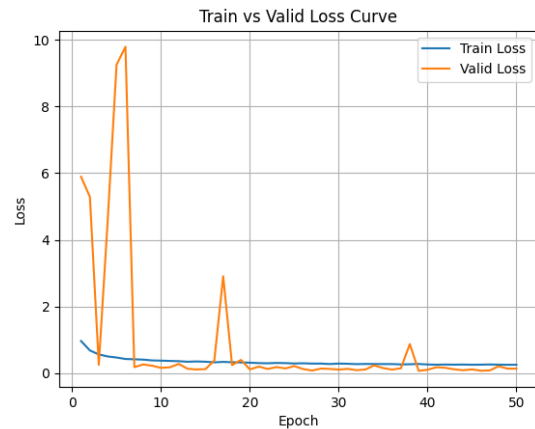


圖 23: ResNet 已收斂 loss 曲線

2) Train Loss 收斂速度差不多。

D. ResNet

在 ResNet 的模型訓練當中，觀察 accuracy、loss 曲線：

- 1) ResNet 在訓練中不論是有無使用 GAN 擴充的資料集，近乎都是呈現無法收斂的情況22，除了設定 batch size 為 32，learning rate 為 $1e-4$ 或 $2e-5$ 的情況有收斂完成。
- 2) 在準確度上也對應了 loss 曲線的情況，因未成功收斂而導致準確度較差，在特定數據下才有準確率達到 0.97 的情況24。
- 3) 針對有使用 GAN 的資料集，在 loss 曲線有收斂完成的情況下，準確率較無使用 GAN 的資料集略低 0.01 左右。

在 ResNet 的模型訓練當中，觀察 confusion matrix 與其指標：

- 1) 對於同一個 class，在 loss 曲線有收斂完成的情況下，precision 與 recall 都達到 0.96 左右，而未完成收斂的情況則是 precision 明顯高於 recall。
- 2) 以 confusion matrix 綜觀全部的 classes，最差的表現通常落在 open、good，而以 open 的情況又較差25。

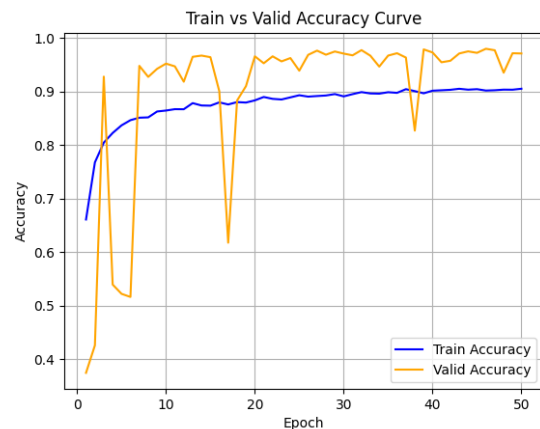


圖 24: ResNet Accuracy 曲線

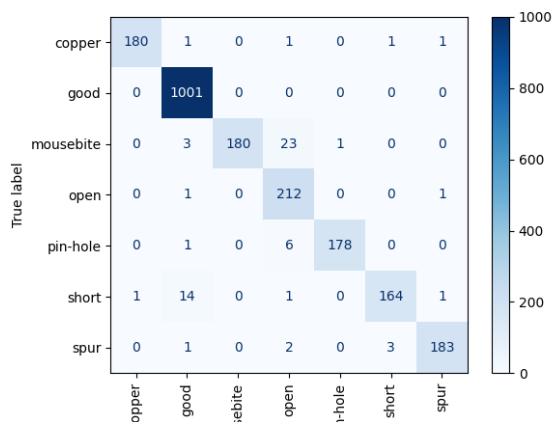


圖 25: ResNet Confusion Matrix

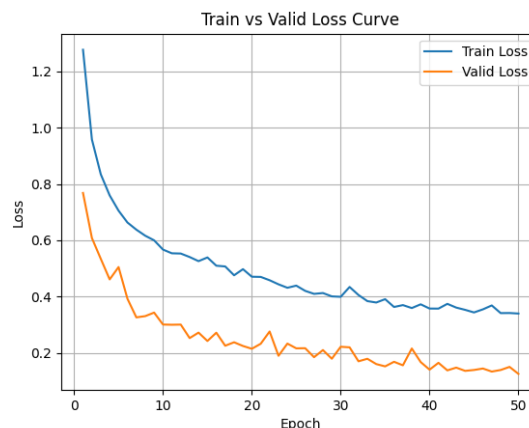


圖 26: SqueezeNet 已收斂 loss 曲線

E. SqueezeNet

在 SqueezeNet 的模型訓練當中，觀察 accuracy、loss 曲線：

- 1) SqueezeNet 在不論是有無使用 GAN 進行擴充的資料集下，loss 曲線幾乎都有收斂²⁶，除了在 batch size 為 8 與 learning rate 為 $1e-3$ ，以本次實驗來說 batch size 極小 learning rate 極大的情況下，loss 曲線完全沒有收斂²⁷，甚至是模型完全沒有學習到特徵。
- 2) 而在 learning rate 為 $1e-3$ 的情況下，多數的 loss 曲線會在 epoch 為 10 左右出現較大的起伏²⁸，進而影響到後面的模型收斂與最後的準確率。
- 3) 準確度上因為多數情況下 loss 曲線都有收斂，所以準確度普遍落在 0.96 左右²⁹，在 learning rate 為 $1e-3$ ，並且最後有收斂的情況下，準確率在 0.69 左右³⁰。
- 4) 針對有使用 GAN 的資料集，在 loss 曲線有收斂完成的情況下，準確率與無使用 GAN 的資料集無明顯差異。

在 SqueezeNet 的模型訓練當中，觀察 confusion matrix 與其指標：

- 1) 對於同一個 class，在 loss 曲線有收斂完成的情況下，precision 與 recall 都達到 0.96 左右，而未完成收斂的情況則是 precision 與 recall 數據持平，表示模型並沒有完整學習到特徵。
- 2) 以 confusion matrix 綜觀全部的 classes，最差的表现通常落在 open、good，又以 open 的情況又較差³¹。

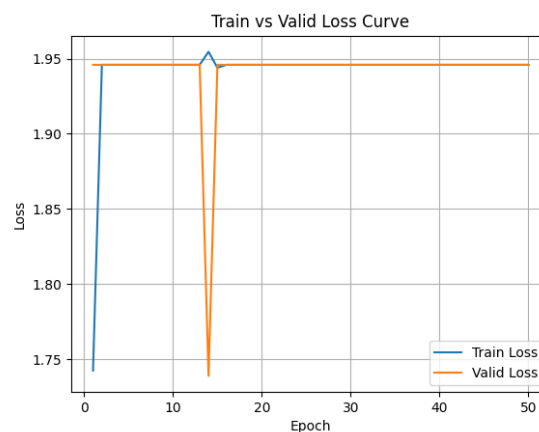


圖 27: SqueezeNet 之 loss 曲線表示模型無學習到特徵

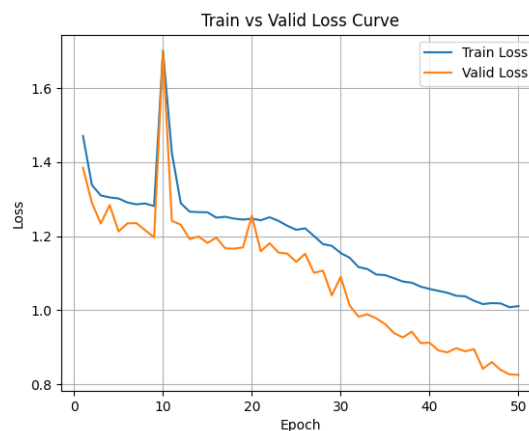


圖 28: SqueezeNet 之 loss 曲線收斂但較不穩

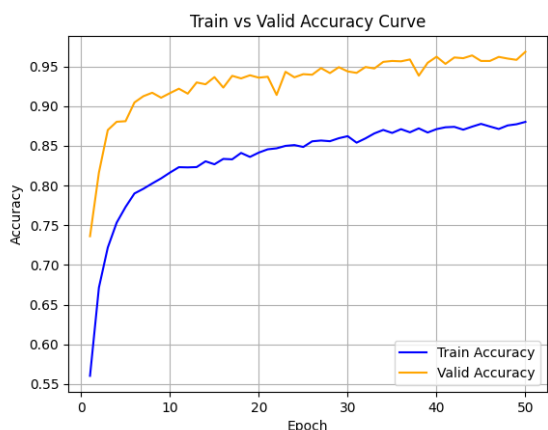


圖 29: ResNet Accuracy 曲線

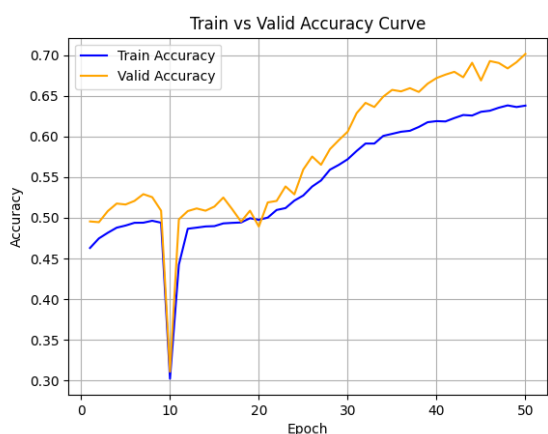


圖 30: ResNet Accuracy 曲線

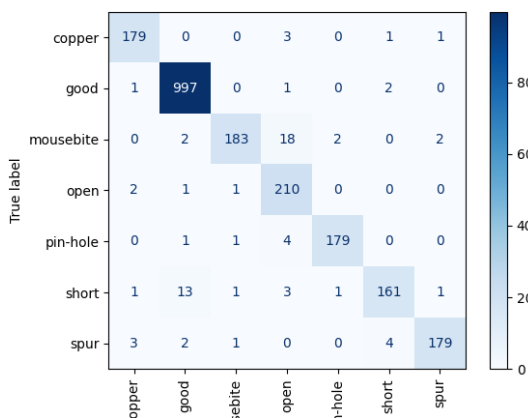


圖 31: ResNet Confusion Matrix

F. Xception

Learning rate 觀察：

當 batch size = 64，且 learning rate 為 1e-3(圖32)、1e-

4(圖33) 時，valid loss 會產生大幅波動，波動的頻率隨著 learning rate 的降低而趨緩；而 valid loss 在 learning rate 為 2e-5(圖34) 時，valid loss 曲線恢復正常收斂。

對於本資料集，較大的 learning rate(1e-3、1e-4) 可能導致模型參數修正過於劇烈，造成 accuracy 與 loss 曲線劇烈波動，而較小的 learning rate(2e-5) 有著最穩定 accuracy 與 loss 曲線。

Batch size 觀察：

當 learning rate = 2e-5，且 batch size 為 8(圖35)、32(圖36) 時，valid loss 會產生大幅波動，波動頻率隨著 batch size 的上升而趨緩；當 batch size 為 64(圖34) 時，valid loss 曲線恢復正常收斂。

對於本資料集，較小的 batch size(8、32) 可能導致模型參數修正過於劇烈，造成 accuracy 與 loss 曲線的劇烈波動，而較大的 batch size(64) 有著最穩定 accuracy 與 loss 曲線。

模型總結：

較小的 Learning rate 與較大的 Batch size 更適合本資料集，可以得到更加穩定的訓練過程。

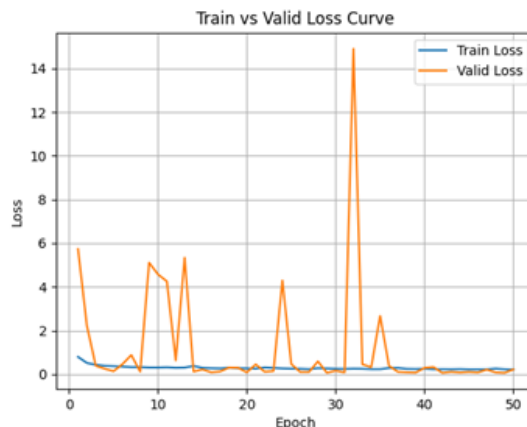


圖 32: Xception B64 Lr1e-3 Loss 曲線

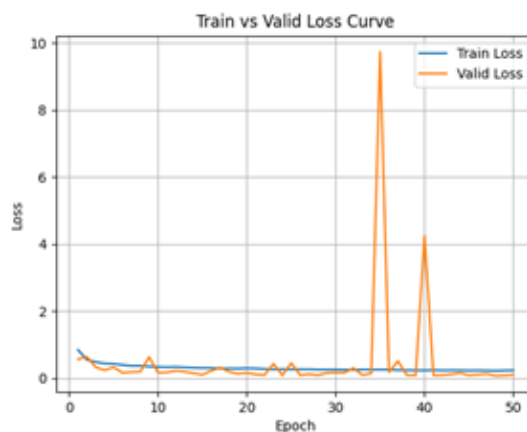


圖 33: Xception B64 Lr1e-4 Loss 曲線

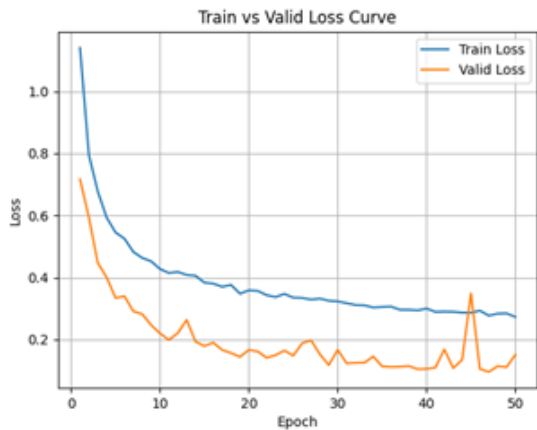


圖 34: Xception B64 Lr2e-5 Loss 曲線

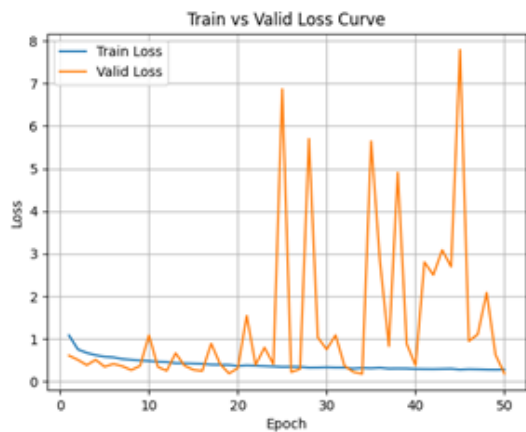


圖 35: Xception B8 Lr2e-5 Loss 曲線

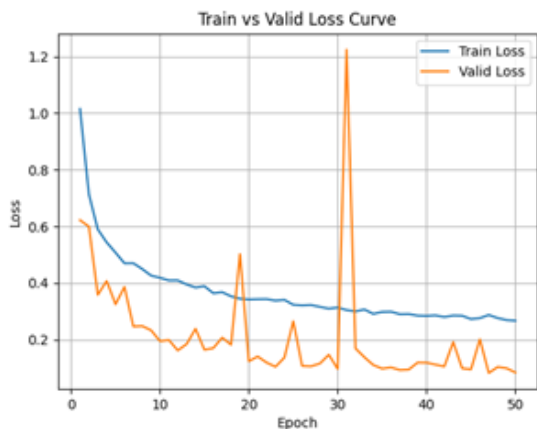


圖 36: Xception B32 Lr2e-5 Loss 曲線

G. MobileNetV1

在 MobileNetV1 的模型訓練當中，觀察 accuracy、loss 曲線：

- 1) Loss 約在 epoch30 左右收斂。
- 2) 但當 learning rate 增加 (約 $1e-4, 1e-3$)，會震盪且無發收斂；Batch size 減少也會有一樣情況，但主要依 learning 為主。
- 3) 此模在此次實驗數據中，以較高的 batch size 和較小的 learning rate 表現較好。

在 MoblieNet 的模型訓練當中，觀察 confusion matrix 與其指標：

- 1) 在 learning rate $1e-3$ 時，部分模型將 good 預測為其他特徵，如 mousebite、spur 等等。

加入 GAN 之後的變化：

- 1) 在加入 GAN 後，平均精度 (mAP) 有些微上升約 12%。

H. MNASNet

在 MNASNet 的訓練結果中，我們可以發現，訓練的 valid loss 基本上都沒有收斂，形狀為跳動 (圖37) 或呈現V形 (圖38)，且結果十分不穩定，接下來透過分析 confusion matrix 中的結果來推導原因。batch size 為 8 的情況：

1. learning rate 為 $1e-3$ 時，原始資料只有在 fold 1 的 epoch 30 有呈現正常辨識結果，剩餘的皆集中判定成一到三個特徵，其中最容易將各特徵判定為 good，(圖39)。增加 GAN 生成的資料集，在 epoch 50 時，都有較為準確的辨識，但仍可看出部分 good 被辨識為瑕疵或是瑕疵被辨識為 good (圖40為部分 good 被辨識為瑕疵，圖41為部分瑕疵被辨識為 good)，而 epoch 10 及 30 時，有時會將大部分特徵辨識為 good，有時能正常辨識各特徵，且辨識效果不一定比 epoch 50 還要差。
2. Learning rate 為 $1e-4$ 與 $2e-5$ 時，epoch 10 的訓練結果全都會辨識成 good (圖42)，而在 epoch 30 時，大都有辨識正確，部份 good 被辨識為瑕疵或是瑕疵被辨識為 good，epoch 50 與 epoch 30 的結果相似，不一定較為準確，但會出現極端狀況，像是 good 全部都辨識為 good，但瑕疵部份辨識正確，部份被辨識為 good (圖43)。

batch size 為 32 的情況：

1. 且 learning rate 為 $1e-3$ 時，原資料集的 epoch 10 會將所有特徵辨識為 good，epoch 30 及 50 時，原資料集與增加 GAN 生成的資料集，會集中將各特徵辨識成某一特徵，此特徵為 good 的頻率最高 (7/12)，open 次之 (3/12)，最後是 spur (1/12)，唯一一次正常辨識的是增加 GAN 生成的資料集的 fold 3 epoch 50 (1/12)，另外，雖然集中辨識為某一特徵，仍可發現極小部分的特徵被正確辨識。
2. learning rate 為 $1e-4$ 時，原資料集各 epoch 結果，幾乎是將所有特徵辨識為 good，但增加 GAN 生成的資料集，在 epoch 50 時，可以較明顯的看出正確辨識結果，但仍有大量 good 被辨識為瑕疵或是瑕疵被辨識為 good。
3. learning rate 為 $2e-5$ 時，不論有無增加 GAN 生成圖片，各特徵皆被辨識為 good。

Batch size 為 64 的情況：

1. learning rate 為 $1e-3$ 時，在原資料集與增加 GAN 生成的資料集的 epoch 10 與 epoch 30 中，特徵皆被全數辨

識為 good。而 epoch 50 時，原資料集在 fold 1 中較能辨識各特徵但仍有大量 good 被辨識為瑕疵或是瑕疵被辨識為 good 的情況，剩餘兩個 fold 中，各特徵幾乎被辨識為 good，增加 GAN 生成的資料集，會各特徵特定辨識成某幾個固定特徵，或是呈現不規則的分布。

2. learning rate 為 $1e-4$ 與 $2e-5$ 時，各特徵皆被辨識為 good。

分析 Precision 與 Recall 的數據：

1. 若是全數被辨識為 good，原資料集的 Precision 會是 0.0715，增加 GAN 生成圖片的資料集的 Precision 會是 0.0662，而兩者 Recall 皆為 0.1429。
2. 第一點的情況，在 batch size 越大及 learning rate 越小的時候，越容易發生。
3. 其餘的數據非常不穩定，例如在 batch size 為 8，learning rate 為 $1e-3$ 的時候，原資料集三個 fold 的 Precision 分別為 0.9105、0.256、0.6202。三者相差甚遠，可以看出相同的參數下，結果的優劣各有不同。同樣的參數，在 fold2 中 epoch 10 到 50 的 precision 分別為 0.014、0.9341、0.256，同樣非常不穩定，另外同時觀測 loss curve（跳動形狀）可以發現，0.9341 恰好是出現在 valid loss 中相對較低的點。

結論推導：

1. 各參數的 valid loss 皆無法收斂，而跳動的 valid loss，其低點有可能產生出較好的實驗結果。
2. 訓練結果中很容易出現全數被辨識為特定特徵的情況，其中又以全數被辨識為 good 最為頻繁出現。
3. 某些參數是能較為正常辨識特徵的，但仍有 good 被辨識為瑕疵或是瑕疵被辨識為 good。
4. 在 Batch size 較大的情況下，越小的 learning rate，越容易將各特徵全數辨識為 good。
5. epoch 為 10 時，時常會出現將所有特徵全數辨識為 good 的情況。

模型總結：

1. 在此實驗中，時常出現全數辨識為 good，以及 good 被辨識為瑕疵或是瑕疵被辨識為 good 的情況。而 good 這個特徵又恰好是兩個資料集中圖片總數最多的一個特徵，因而造成了模型較易將各特徵辨識為 good，或是較明顯的將 good 誤檢的的情況。
2. 在此資料集中，MNASNet 不適合較大的 batch size 以及較小的 learning rate 同時出現。

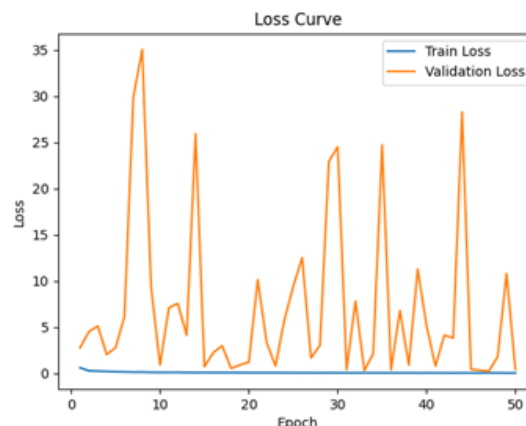


圖 37: MNASNet 跳動 Loss 曲線

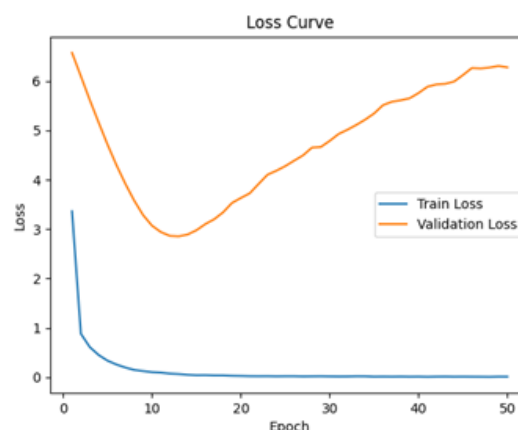


圖 38: MNASNet V 型 Loss 曲線

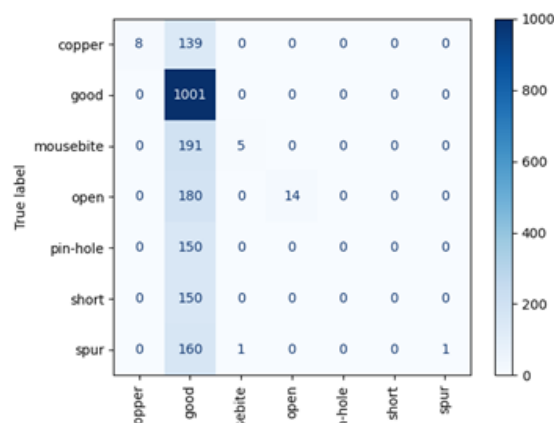


圖 39: MNASNet 錯誤辨識為 good confusion matrix 範例

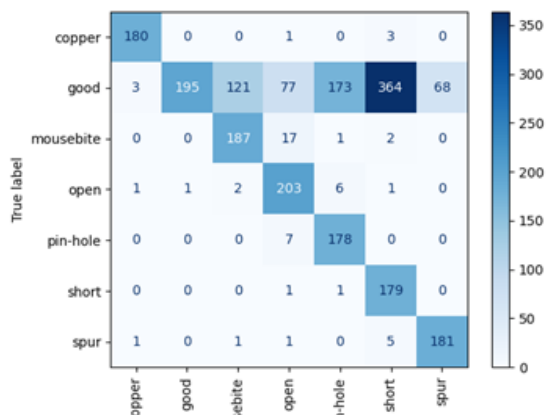


圖 40: MNASNet 部分 good 辨識為瑕疵 confusion matrix

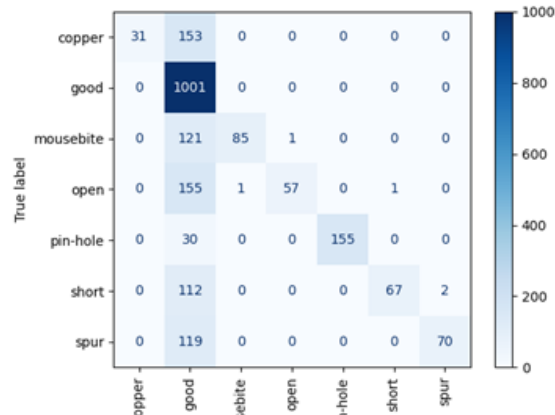


圖 43: MNASNet good 正確辨識、部分瑕疵辨識為 good confusion matrix

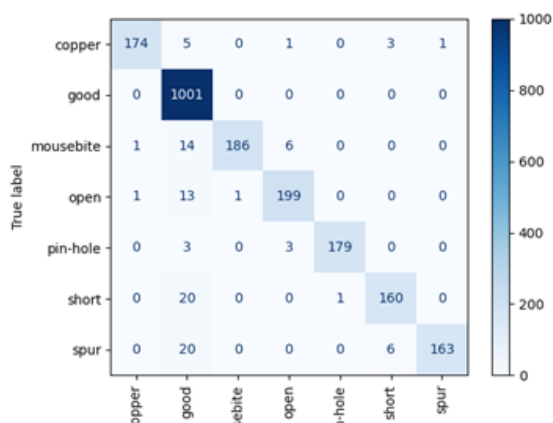


圖 41: MNASNet 部分瑕疵辨識為 good confusion matrix

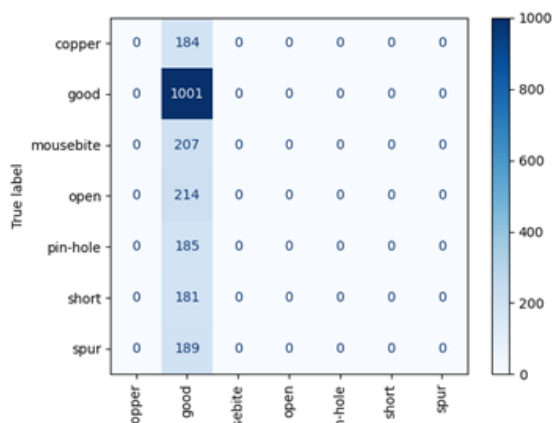


圖 42: MNASNet 全瑕疵辨識為 good confusion matrix

I. EfficientNet

EfficientNet 原始資料集訓練結果中

- 1) Epoch 值較大時，模型訓練表現 Loss 確實略微下降。注意到隨著 Batch size 值較大時，Loss 下降較為明顯。
- 2) 觀察到隨著 Batch size 值越大，訓練表現的 Loss 略為增加。

GAN 生成資料集的表現特點

- 1) 表現與原始資料集相比，並無明顯差別。
- 2) 觀察相同 Epoch= 50 與 Learning Rate= 2e-5 的條件下，Batch Size 的增加 Loss 的表現無明顯改善。

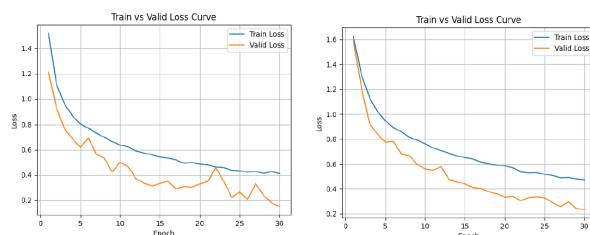


圖 44: EfficientNet epoch30 Lr2e-5 Loss 曲線 (左:Batch size32，右:Batch size64)

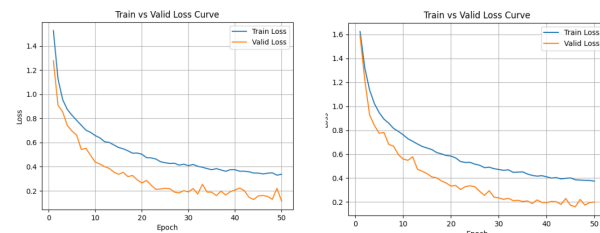


圖 45: EfficientNet epoch50 Lr2e-5 Loss 曲線 (左:Batch size32，右:Batch size64)

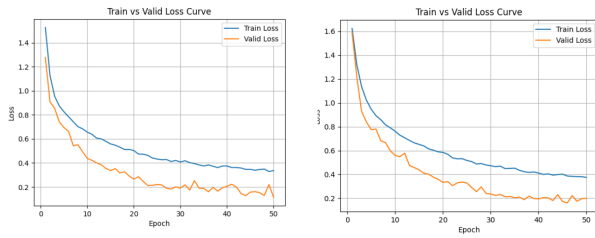


圖 46: EfficientNet epoch50 Lr2e-5 GAN Loss 曲線
(左:Batch size32, 右:Batch size64)

K. ViT

在 ViT 的模型訓練當中，觀察 accuracy、loss 曲線：

- 1) 隨著 learning rate 增加，模型收斂的速度也隨之增快，若 learning rate 過低 (如 $lr = 1e-3$)，會有可能遇到模型收斂困難的問題。
- 2) 模型訓練的準確度在 epoch10~20 的區間時就已經接近 100%，訓練集準確度接近 100%，但驗證集的準確率在收斂至約 95% 時就不再收斂。
- 3) 更改 batch size 參數，batch size 提升時整體 loss 皆有所下降的趨勢，準確率也有相對提升，不過隨著 learning rate 提升，改變 batch size 所造成的影響也隨之變小。

在 ViT 的模型訓練當中，觀察 confusion matrix 與其指標：

- 1) 對於同一個 class，有時候 Precision 會低於 Recall，有時 Precision 高於 Recall，表示模型漏檢或是誤報的現象並不固定。
- 2) 綜觀全部的 class，表現最差 class 大多都落在 open、short、spur，而 open 的表現最差，約落在 90%。

加入 GAN 之後的變化：

- 1) 對於整個訓練來說，隨著 epoch 增加的模型收斂速度有略為提升。
- 2) 在較低 learning rate 當中的表現有得到不錯的提升，到了 $lr=2e-5$ 原本就很高的準確率也上升了 1~2%。

L. ConvNeXt

Learning rate 觀察：

此模型於相同 batch Size 時，在 learning rate 為 $1e-3$ 時，20 個 epoch 內便能使 valid loss 低於 0.2(圖48)，30 個 epoch 的 valid accuracy 已達 0.95(圖49)；在 learning rate 為 $2e-5$ 時，50 個 epoch 的 valid loss 卻依然高於 0.3(圖50)，且 valid accuracy 只到達 0.9(圖51)。

對於 PCB 瑕疵資料集，較大的 learning rate($1e-3$) 有著最快的收斂速度及最高的 test accuracy。

Batch size 觀察：

此模型於相同 learning rate 時，batch size 為 8 與 64 的 loss curve(圖52) 與 accuracy curve(圖53) 極其相似，且 test result(表III) 的各項數值差距均於 10% 以內。

當 batch size 由 8 提升至 64 時，recall 與 F1 score 皆略微提升，precision 略微降低。Batch size 為 32 時，mPA、TestLoss 和 TestAccuracy 有著最佳表現，有著最低的 loss 與最高的 accuracy。

模型總結：

- 1) 對本資料集較大的 Learning rate($1e-3$) 更適合，有著更快的收斂速度及更高的準確率。

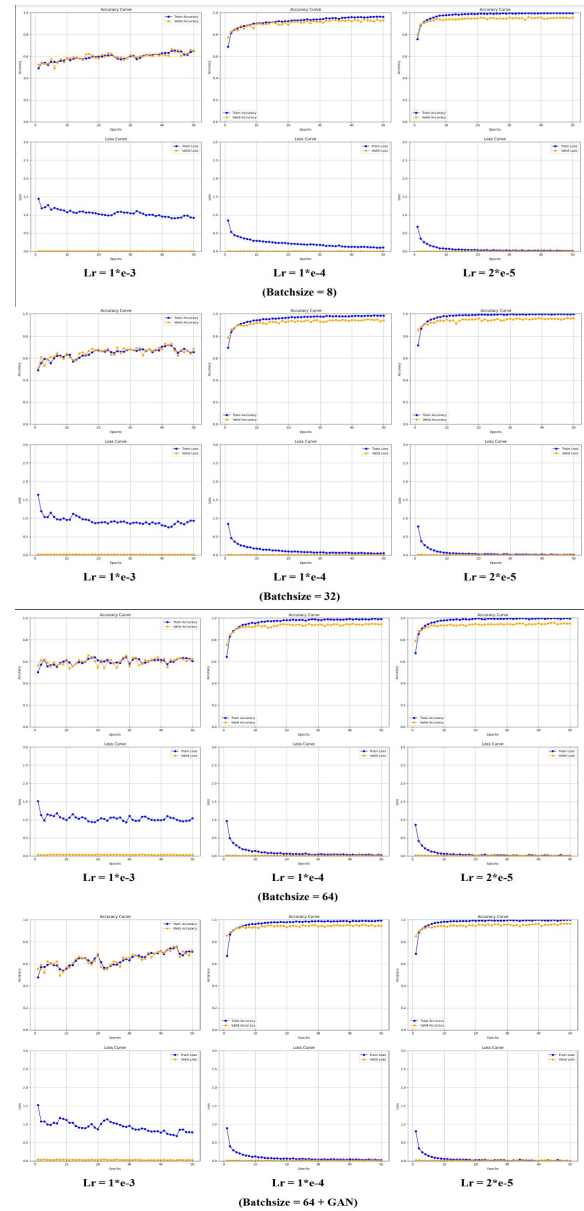


圖 47: ViT 訓練結果

- 2) 對本資料集 Batch size 為 32 最適合，可於最多的指標獲得最佳效果。

表 II: ConvNeXt Batch size64 Test Result

	Precision	Recall	F1 Score	mAP	Loss	Accuracy
Lr1e-3	0.9512	0.946	0.9481	0.946	0.1335	0.9598
Lr1e-4	0.9515	0.9247	0.9371	0.9247	0.1581	0.9571
Lr1e-5	0.9002	0.876	0.8858	0.9876	0.2961	0.919

表 III: ConvNeXt Learning rate1e-3 Test Result

	Precision	Recall	F1 Score	mAP	Loss	Accuracy
B8	0.9609	0.936	0.9474	0.936	0.1398	0.9598
B32	0.9617	0.944	0.9523	0.944	0.1278	0.9633
B64	0.9512	0.946	0.9481	0.946	0.1335	0.9598

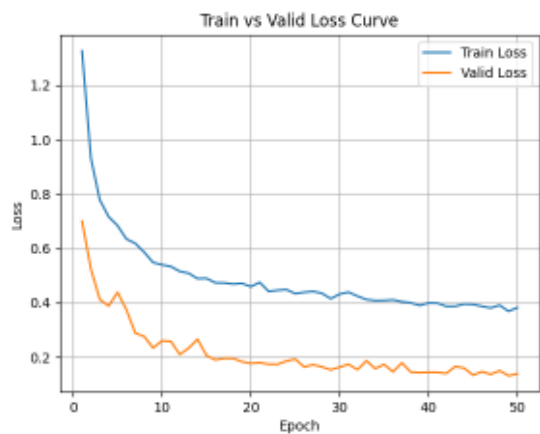


圖 48: ConvNeXt B64 Lr1e-3 Loss 曲線

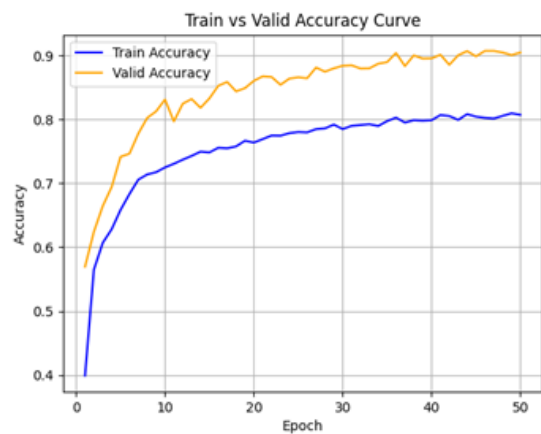


圖 51: ConvNeXt B64 Lr2e-5 Accuracy 曲線

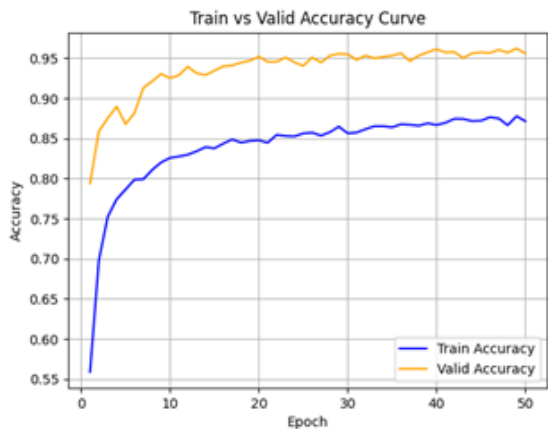


圖 49: ConvNeXt B64 Lr1e-3 Accuracy 曲線

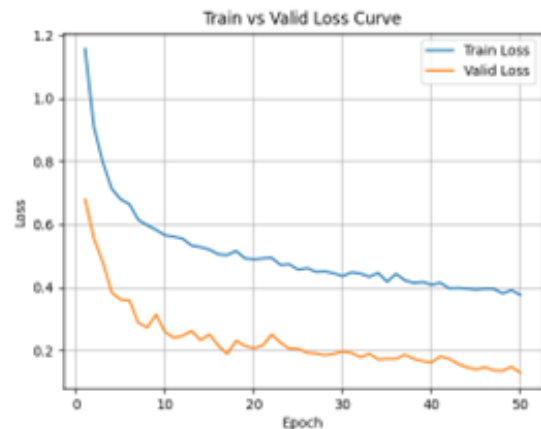


圖 52: ConvNeXt B8 Lr1e-3 Loss 曲線

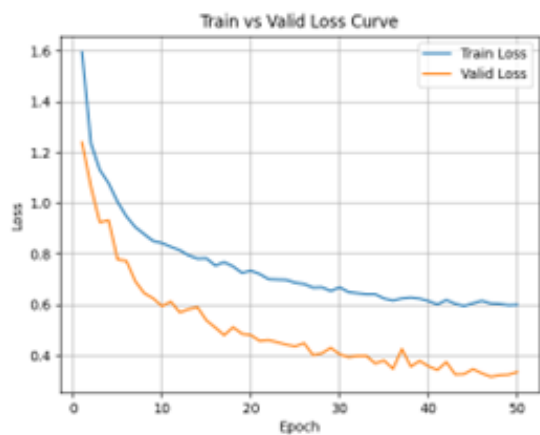


圖 50: ConvNeXt B64 Lr2e-5 Loss 曲線

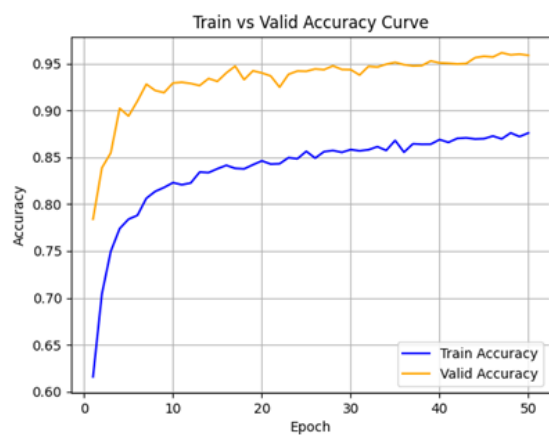


圖 53: ConvNeXt B8 Lr1e-3 Accuracy 曲線

5. 結論

透過實驗數據的綜合分析，得到以下結論：資料擴增的影響：引入 GAN 生成的瑕疵資料後，資料多樣性有明顯提升，尤其在初期訓練階段，有助於模型 Loss 的穩定收斂。我們也觀察到 GAN 擴增資料對於不同模型成效也不一，其中 ViT、MobileNetV1、GoogLeNet 等模型訓練成效都不錯，mAP 分別提升約 1% 至 2%。

模型參數對表現的影響：

較小的 Batch size 有助於穩定訓練，但可能限制模型對細微瑕疵的 recall 表現。大批次大小則提升 recall 能力，但可能導致收斂速度變慢。Learning rate 對模型穩定性影響十分顯著，尤其是 VGG 模型，在使用極低 Learning rate（如 $2e-5$ ）時，表現精度達到最佳，而過高的 Learning rate 則可能導致震盪以致無法收斂。

特殊情況的觀察：MNASNet 表現出較明顯的不穩定性，其 valid loss 波動顯著，且即使引入 GAN 資料，精度提升幅度有限。

整體模型表現：本實驗大中多數的模型表現較為穩定，且能適應 GAN 擴增資料的模型。而 VGG 模型則可能比較適用於極高精度的任務場景，MNASNet 則需進一步調整以提升穩定性。

未來，本研究可在以下方向展開更深入的探索：

1. 提升 GAN 資料的生成方法，提升其與實際資料特性的匹配程度，進一步增強模型對瑕疵的辨識能力。
2. 受限於訓練時長，本次模型訓練的參數選擇並不能做到全面涵蓋，可能需要探索更高效的訓練策略，以加速 Loss 收斂與提升模型精度的目的。
3. 再引入更多新的深度學習模型或技術，驗證這些模型在 PCB 瑕疵檢測領域的適用性。

參考文獻

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Computational and Biological Learning Society*. 3rd International Conference on Learning Representations, 2015, pp. 1–14. [Online]. Available: <https://arxiv.org/pdf/1409.1556>
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [5] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *ArXiv*, vol. abs/1602.07360, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:14136028>
- [6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2375110>
- [7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:12670695>

- [8] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:167217261>
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:225039882>
- [10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014. [Online]. Available: <https://doi.org/10.48550/arXiv.1406.2661>
- [11] Papers with Code, "Deeppcb dataset," 2024, accessed: 2025-01-24. [Online]. Available: <https://paperswithcode.com/dataset/deep-pcb>