

Task A. Model Selection

在本次深度學習作業中，我選擇了 ResNet-50 與 DenseNet-121 兩個模型作為遷移學習的基礎架構。這兩個模型皆為 `torchvision.models` 中常見的預訓練模型，並已在 ImageNet 上完成訓練。選擇它們的原因在於，它們分別代表兩種不同的設計思路：ResNet 採用殘差連接(Residual Connections)，而 DenseNet 採用密集連接(Dense Connections)。這兩種結構都能有效解決深層網路的梯度消失與特徵流失問題，但在特徵重用、參數效率與訓練穩定性上各有優勢，適合用於觀察在小型資料集(如醫學影像)上的遷移學習效果差異。

ResNet-50 是一個具有 50 層深度的卷積神經網路，核心概念在於透過「殘差學習」讓模型能學習輸入與輸出之間的差異，而非完整映射。這樣的設計能讓深層網路更容易訓練，並且能夠在不顯著增加計算量的情況下獲得更好的表現。ResNet 的架構相對規則，參數量中等，且在多數視覺任務中都有穩定的遷移效果。由於醫學影像的資料量通常較少、類別差異細微，ResNet-50 的穩定性和良好的收斂特性，使其成為微調(Fine-tuning)階段的理想選擇。

DenseNet-121 則是一個強調特徵重用的架構。它的每一層都會接收前面所有層的輸出，形成密集的特徵流通，使模型能夠充分利用前期所學到的低層資訊，並減少特徵重複學習的情況。這種連接方式能提高特徵效率，使 DenseNet 在相對較少參數的情況下仍能保持高準確度。同時，它的梯度傳遞更順暢，因此在資料較少或需凍結部分層進行特徵擷取(Feature Extraction)時，往往能展現穩定的表現。

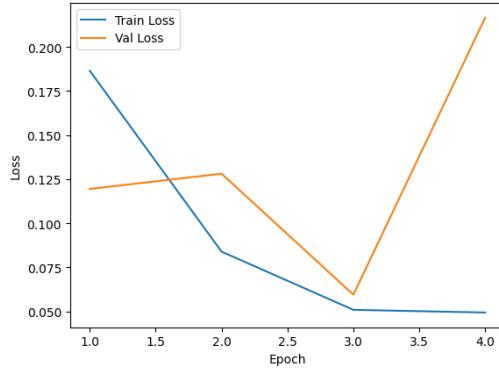
綜合考量，ResNet-50 的結構穩定且收斂快速，適合作為「全網微調」模型，用以驗證在完整訓練下的性能上限；DenseNet-121 則更適合在「特徵擷取模式」下使用，能以較低的訓練成本獲得良好的泛化能力。兩者的組合能提供在模型複雜度、計算成本與泛化表現之間的對比觀察。

Task B. Fine-tuning the ConvNet**(1) ResNet-50**

我用 ResNet-50 載入 ImageNet 的預訓練權重後，將最後一層改成二分類，整個網路都開放微調。訓練曲線相當穩定，前幾個 epoch 損失下降很快，之後逐漸收斂。測試結果為 loss 0.246、accuracy 0.92、F1 0.91。從混淆矩陣來看，模型在 PNEUMONIA 類的召回率特別高(0.96)，而 NORMAL 類相對略低(0.85)。換句話說，它寧可「誤判一點 NORMAL」也不想漏掉肺炎樣本，這對醫學任務而言其實是一種合理的偏向。

ResNet-50 的優勢在於殘差結構能穩定地傳遞梯度，因此即使網路很深，微調時也不容易陷入梯度消失。這讓模型能有效地把在自然影像學到的高階語義特徵轉移到 X-ray 資料上。我觀察到在全網更新的情況下，ResNet 能學

到更具語義性的形狀差異，例如肺部陰影區塊的對稱性、胸腔邊緣的亮度落差等。雖然訓練時間較長、顯存占用大，但換來的是更高的上限表現。若要改善 NORMAL 的誤判問題，我會考慮加強資料增強或引入類別權重，讓模型在決策邊界更平衡。整體而言，ResNet-50 的 fine-tuning 非常有效，能充分展現深層架構在新任務上的遷移能力。



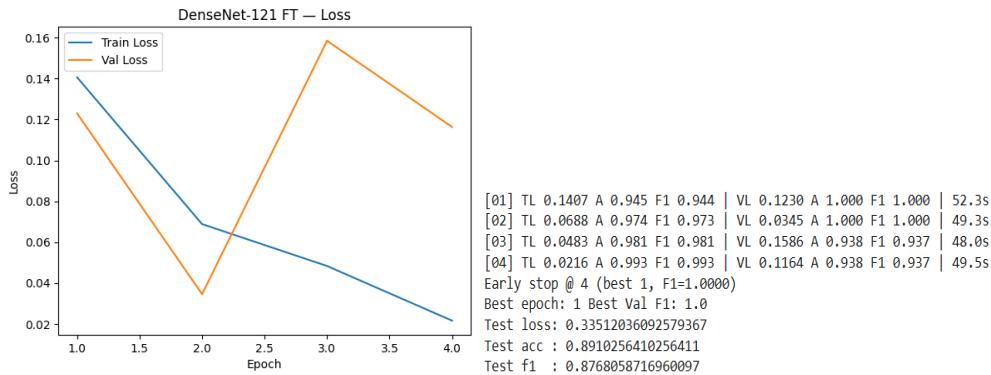
```
Classes: ['NORMAL', 'PNEUMONIA']
[01] train loss 0.1864 acc 0.927 f1 0.927 | val loss 0.1195 acc 1.000 f1 1.000 | 54.6s/epoch
[02] train loss 0.0839 acc 0.968 f1 0.967 | val loss 0.1281 acc 1.000 f1 1.000 | 47.1s/epoch
[03] train loss 0.0509 acc 0.984 f1 0.984 | val loss 0.0595 acc 1.000 f1 1.000 | 47.0s/epoch
[04] train loss 0.0494 acc 0.984 f1 0.983 | val loss 0.2165 acc 0.938 f1 0.937 | 48.2s/epoch
Early stopping at epoch 4 (best 1, F1=1.0000)
Best epoch: 1 Best Val F1: 1.0
Saved: artifacts/resnet50_finetune_best.pth
```

(2) DenseNet-121

DenseNet-121 同樣載入 ImageNet 權重後進行全網微調。不同的是，它的密集連接設計讓特徵能被層層重用，因此學得很快，第一個 epoch 就達到最佳驗證 $F1=1.000$ 。雖然之後略有震盪，但整體仍維持高分。最終測試結果是 loss0.335、accuracy0.891、F10.877。和 ResNet 相比，分數稍低，但訓練速度明顯快，每個 epoch 只要約 50 秒，資源效率很好。

DenseNet 在這個任務上的表現反映出它的結構特性：密集連接讓低階的邊緣與紋理特徵能被反覆利用，對資料量有限的 X-ray 任務特別友善。不過我也注意到它對資料分佈變化稍微敏感，例如驗證集表現滿分但測試略低，可能是因為早期就過擬合在驗證樣本的對比特徵上。若要進一步提升，我會延後早停條件、或加入亮度與對比度的微量增強，讓模型更泛化。

整體來說，DenseNet-121 的微調效果依然不錯，雖然上限略低於 ResNet-50，但訓練快速、穩定度高、顯存需求小，對計算資源有限或時間緊的情況特別實用。它在效率與表現之間取得不錯的平衡，能證明 fine-tuning 確實讓預訓練模型快速適應 X-ray 二分類任務。



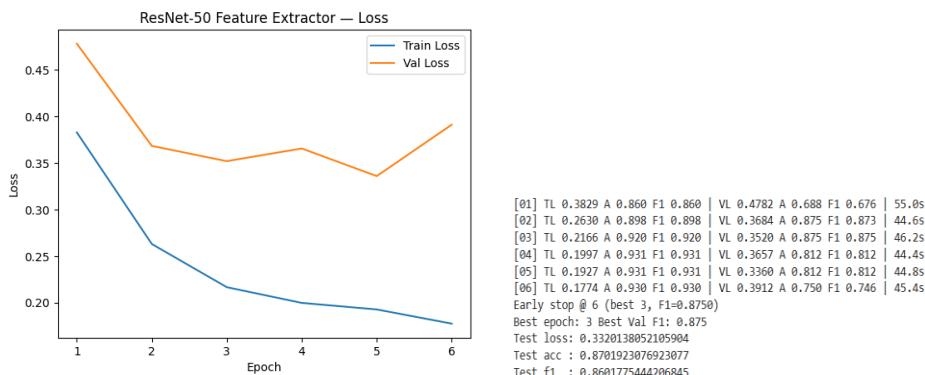
Task C. ConvNet as Fixed Feature Extractor

(1) ResNet-50

這部分我把 ResNet-50 改成固定特徵提取器，只保留最後一層可以更新。訓練時，整個骨幹都凍結，只有分類頭在學習。整體收斂速度比微調慢一點，但曲線仍穩定。最佳驗證 F1 出現在第 3 個 epoch (ValF1=0.875)，之後逐漸趨平並在第 6 個 epoch 早停。最終測試結果是 loss0.332、accuracy0.870、F10.860。

可以明顯看出，只訓練最後一層後，表現比 fine-tuning 版本低一些（約 F1 下降 0.05）。原因很直接：ResNet 的高層特徵原本在 ImageNet 上學的是自然影像的語義分佈，與 X-ray 這種灰階結構圖還是有落差。雖然底層的邊緣、紋理特徵仍有用，但由於沒有重新調整高層語義權重，模型在決策邊界的適應能力受限。從訓練曲線來看，驗證 F1 在 3 個 epoch 後就達平臺，表示分類頭已經把可用的特徵榨乾。

不過這個設定仍有它的優勢。ResNet 凍結後運算量大幅減少，訓練時間短、顯存壓力低，而且在沒有微調骨幹的情況下還能維持 F1 約 0.86，說明預訓練特徵仍相當有辨識力。若想進一步提升，我會考慮在分類頭前多加一層 dropout 或 batchnorm 作為輕量調整，讓模型對輸入分佈的細微差異更有彈性。整體來說，ResNet 作為固定特徵提取器的表現穩定但受限，上限比微調版本低一個層級，適合用在訓練資料少或訓練時間受限的情況。

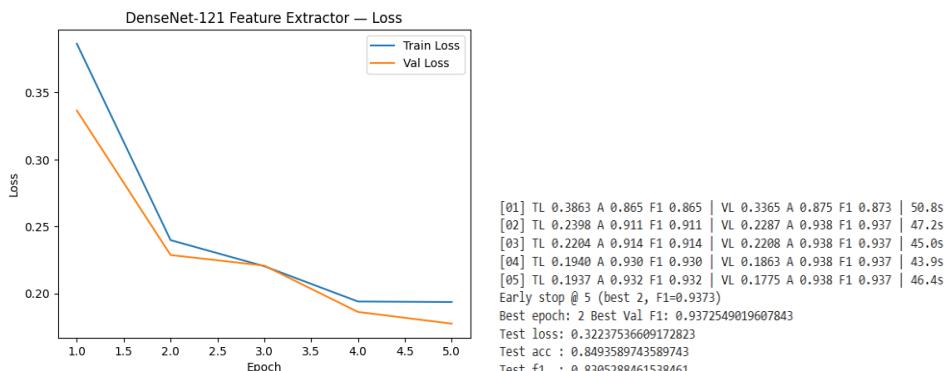


(2) DenseNet-121

DenseNet-121 在凍結骨幹後的表現比我預期的好。由於它的密集連接結構，低階與中階特徵被大量重用，即使不更新權重，也能提供相當豐富的表徵資訊。這讓分類頭在少量更新下就能學出不錯的區分能力。訓練過程非常平穩，第 2 個 epoch 就達到最佳驗證 F1=0.937，之後幾個 epoch 幾乎持平。最終測試結果是 loss0.322、accuracy0.849、F10.831。

雖然測試 F1 比 ResNet 略低（差不多 3 個百分點），但從驗證表現看，DenseNet 在固定特徵的狀態下對資料的適應性明顯更好。這主要是因為密集連接的設計能保留更多層次的特徵表現，尤其是紋理與局部對比的細節。從曲線來看，它幾乎沒有過擬合跡象，代表特徵本身相當穩定，也說明它在新任務上的遷移效率高。每個 epoch 約 45 – 50 秒，比 ResNet 略快，訓練成本低。

比較兩個模型，ResNet 在 fine-tuning 時的上限比較高，但一旦凍結權重，DenseNet 的設計優勢就開始顯現：它的特徵共享讓「固定特徵提取」的效果更好。換句話說，ResNet 需要讓權重動起來才能真正發揮，而 DenseNet 即使不更新也能保留相當多的可辨識性。這種差異說明不同架構在遷移學習中對「可重用特徵」的依賴程度不同。整體來看，兩者在這次任務中都能穩定運作，但 DenseNet 在 featureextractor 模式下的效率與穩定性稍勝一籌。



Task D. Compare and Analysis

整體來說，fine-tuning 的效果明顯比固定特徵提取好。ResNet-50 在微調後的 F1 有 0.913，DenseNet-121 也到 0.877；但當我只訓練最後一層時，分別降到 0.86 和 0.83。這代表把整個網路開放訓練，模型能更好地重新學習 X-ray 的特徵，而只靠固定特徵的版本會有點吃虧。

ResNet 的差距最明顯，因為它的高層特徵比較抽象，需要重新調整才能抓到醫學影像的重點；一旦凍結，適應力就下降。DenseNet 的差距則比較小，它的密集連接讓低層資訊能被後續重複使用，即使不微調也能維持穩定表現。

整體來看，fine-tuning 比較適合想追求高表現、資料量夠的情況；而 featureextractor 版本則省時間、省資源，效果雖略低但很穩。兩種方法對比下來，就是「ResNet 靠深度換性能、DenseNet 靠結構保穩定」的典型例子。

Task E. Tset Data Analysis

我覺得測試集的表現很難再往上拉，主要是訓練階段的設定已經讓模型到達它能學到的極限。像是學習率、batchsize、正則化強度這些參數，其實都會影響模型的穩定度。當我調太高的時候，loss 會在低點附近一直震盪；但如果太低，模型又容易卡在某個次佳解裡面，準確率上不去。

另外，像 ResNet 這種參數比較多的模型，在 fine-tuning 時很容易出現「看起來訓練很好、但測試掉分」的情況。我覺得那不是資料問題，而是模型真的學得太細，把訓練集記住了。DenseNet 雖然穩一點，但它在固定特徵提取的時候又太保守，變成沒辦法再學到更多有用的模式。

整體來說，我覺得測試表現難提升，不是模型不夠好，而是這組訓練參數已經跑到一個平衡點。要再進步，可能要動學習率排程、增加 earlystopping 的彈性，或試試不同層級的凍結策略，讓模型能在穩定和泛化之間再多一點空間。