

## 1.1 Task A: Model Selection

Model Choices: 1. DenseNet-121 2. ResNet-50

### 1. DenseNet-121

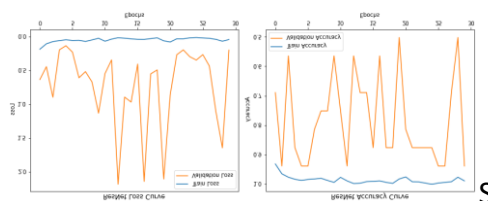
- 架構複雜度：DenseNet-121 透過「密集連接」的方式讓每一層可以接收所有前面層的輸出，使其在使用較少參數的情況下仍具備高效能。模型約有 800 萬 個參數，相對緊湊，並包含數個密集區塊（Dense Blocks）及降維的過渡層（Transition Layers），降低計算成本。
- 預訓練效能：在 ImageNet 資料集上，DenseNet-121 擁有 74% 以上的準確率，適合小樣本數據的遷移學習。其架構改善了梯度消失問題，有助於提升模型在遷移學習任務上的適應性。
- 計算效率：DenseNet-121 因參數效率高，訓練和推論速度較快，特別適合需要節省資源的任務，如醫學影像分類等。

### 2. ResNet-50

- 架構複雜度：ResNet-50 具有 50 層 且採用殘差連接（Residual Connections）設計，解決梯度消失問題並降低深層網路的訓練難度。模型約有 2300 萬 個參數，雖比 DenseNet-121 複雜，但兼具效能和效率。
- 預訓練效能：ResNet-50 在 ImageNet 資料集上達到 76% 以上的準確率，能有效應對多樣化、複雜的任務，適合進行深入的特徵提取。
- 計算時間與效率：ResNet-50 在深層模型中相對高效，但需要較高的計算資源。殘差結構加速了收斂速度，對多樣性或大數據量的遷移學習任務尤其適合。

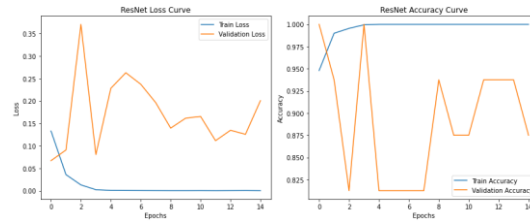
## 1.2 Task B: Fine-tuning the ConvNet

ResNet-50:

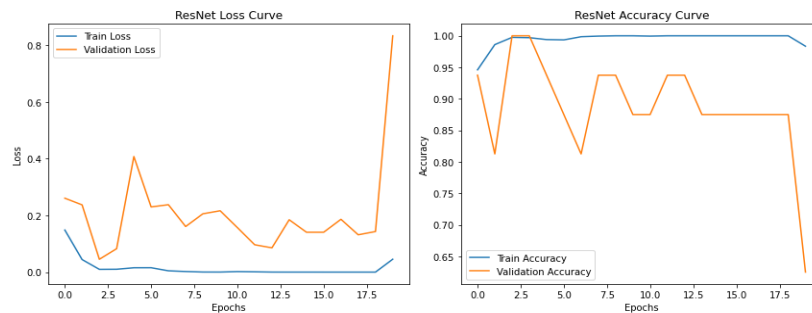


learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-3	30	1e-4	32	1612.61s	0.4301	89.42%

Learning rate=1e-3 太大導致 loss 無法收斂。並且 epoch=30 運算時間過久，我決定使用 epoch=20。

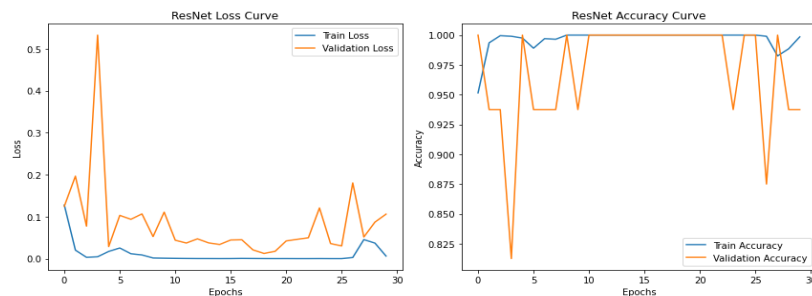


在 epoch=15 中發現還沒有收斂的趨勢，而 epoch=30 就需要過多的計算時間，在時間與收斂趨勢的權衡下，後面都是使用 epoch=20。

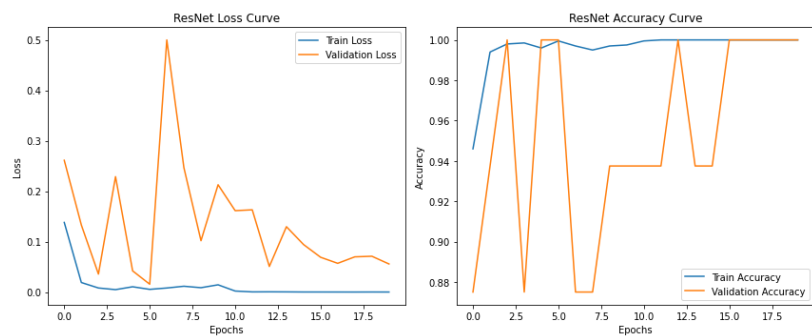


learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-4	20	1e-5	16	1108.82s	0.4493	84.94%

batch size 改到 16 發現明顯不收斂，因此放棄使用 batch size=16



learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-4	30	1e-5	32	1635.44s	0.9531	80.93%



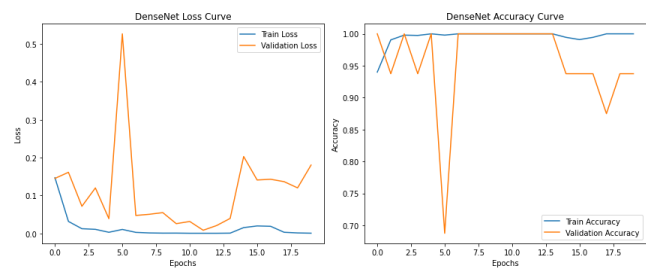
learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-4	20	1e-5	32	1096.16s	0.8148	87.50%

**Batch size**：我們最初測試了 **Batch size** 為 16 的設定，但觀察到收斂不穩定，模型在學習上表現不一致。這促使我們改用 **Batch size** 為 32，這樣能使訓練過程中的收斂更加穩定。

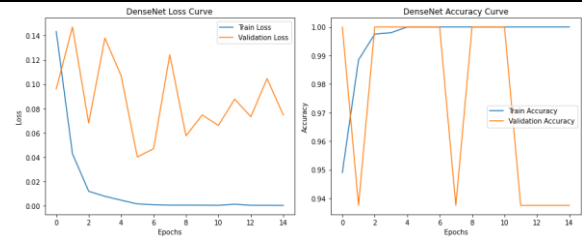
**Learning rate**：我們最初嘗試了  $1e-3$  的 **Learning rate**，但導致了 **loss** 函數的高度波動，使模型難以收斂。將學習率降低至  $1e-4$  後，收斂變得平滑，讓模型能夠更有效地學習，避免頻繁的震盪。

**Epoch**：雖然 30 個回合提供了更多的收斂機會，但計算時間成為一個限制因素。我們發現將回合數減少至 20，能在性能和計算效率之間取得平衡，並且讓模型能夠充分收斂。

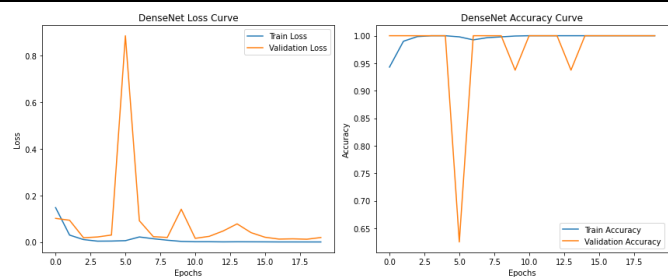
DenseNet-121:



learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
0.8e-4	20	1e-3	32	1134.55s	0.7389	85.58%



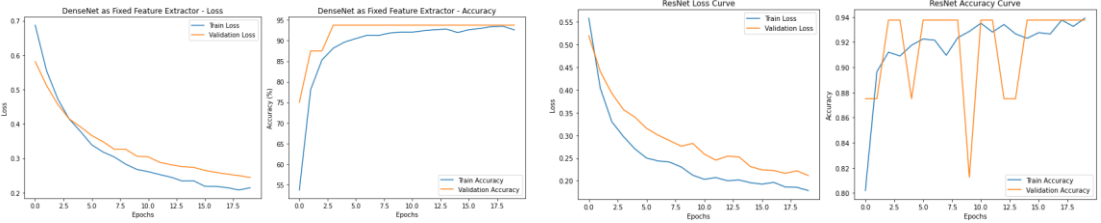
learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-4	15	1e-4	32	820.01s	0.6765	86.54%



learning_rate	epochs	weight_decay	batch_size	Training Time	Test Loss	Test Accuracy
1e-4	20	1e-4	32	1106.29s	0.6302	87.66%

由於訓練時間與報告篇幅關係，沒有把全部的訓練結果放上，與 Resnet 相同，我的 learning rate, epoch 等 hyperparameter，沒有過多的測試，可以發現在與 Resnet 用相同的參數，就能看到 loss 收斂的結果。

### 1.3 Task C: ConvNet as Fixed Feature Extractor



DenseNet-121 as a fixed feature extractor  
Test Loss: 0.3325, Test Accuracy: 87.50%  
Total Training Time: 1010.96s

ResNet-50 as a fixed feature extractor:  
Test Loss: 0.3190, Test Accuracy: 85.26%  
Total Training Time: 1010.73s

在 Fixed Feature Extractor 下，ResNet 和 DenseNet 的表現都優於原始模型，但相較之下，DenseNet 在特徵提取方面表現略勝一籌。具體來說，DenseNet 在訓練時也展現了穩定的收斂性，其 loss 曲線同樣平滑，且波動更少，訓練與驗證的 loss 都快速趨於穩定。

整體來看，DenseNet 的收斂速度與準確率比 ResNet 稍高，且其 loss 曲線更為平穩，這意味著 DenseNet 在固定特徵提取任務中提供了更一致的性能與更少的波動。在這種設定下，兩者都優於原始模型，但 DenseNet 更適合需要高穩定性的任務。

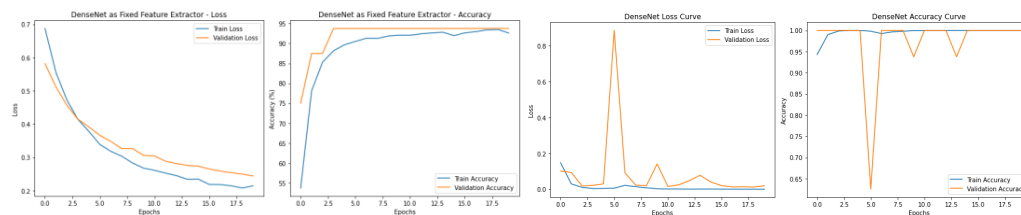
Metric	ResNet-50 (Fixed Feature Extractor)	DenseNet-121 (Fixed Feature Extractor)
Training Accuracy	Steady improvement but fluctuated	More stable and consistent increase
Validation Accuracy	Gradual improvement with fluctuations	Gradual and steady improvement
Training Loss	Gradual decrease with some fluctuations	Smoother and more consistent decrease
Validation Loss	Steady decrease, but less smooth	Smooth and steady decrease

Fixed Feature Extractor 並僅訓練最後的分類層，有效地減少了訓練過程中需要更新的參數數量，這樣做的效果包括： 1.減少 overfitting：通過保持 pre-training 的特徵固定，兩個模型都避免了對較小的 dataset overfitting，由上方的圖可以發現 loss 的波動減少非常多。2.預計減少計算時間：由於只有最後的分類層被微調，因此訓練時間應該比完全微調時短。

Fixed Feature Extractor: Training Time: 1010.96s V.S Fine-tuning: Training Time: 1106.29s  
但是:Fixed Feature Extractor 和原始模型的訓練時間想像中多，可能有以下原因： 分類

層計算量：儘管凍結了 Convolution layer，但最後的分類層仍需更新，這部分計算量不小。 Convolution layer 處理特徵：即使凍結了 Convolution layer，它仍需要處理輸入的特徵，這消耗了計算時間。硬體影響：GPU 計算的平行處理可能讓兩者訓練時間差異不大。模型深度：深層模型即使凍結部分層，前向傳播計算仍需較多時間。總的來說，Fixed 卷積層後，分類層和計算過程中的其他因素可能使訓練時間未顯著減少。

## 1.4 Task D: Comparison and Analysis



1. loss 曲線： fixed Feature Extractor（以 DenseNet-121 為例）：訓練與驗證損失曲線穩定下降，最終測試損失為 0.3325，表現穩定。 Fine-tuning：損失曲線波動較大，最終測試 loss 為 0.6302，顯示微調可能導致過度擬合或不穩定。

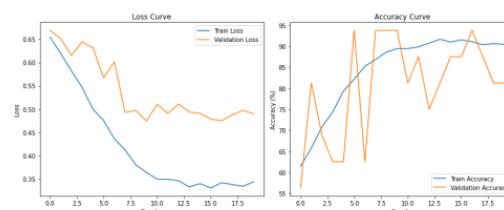
2. accuracy 曲線： fixed Feature Extractor：訓練與驗證準確率快速達到 87.5%，且穩定。 Fine-tuning：準確率波動明顯，最終測試準確率略高於 87.66%，但不穩定，可能過度擬合。

3. Adaptability 與 Generalization：fixed Feature Extractor 穩定且 Generalization 能力強，適合資源有限或需高 Generalization 的情況。 Fine-tuning 具更高 Adaptability，但波動大且可能 overfitting，需更謹慎調整參數。

4. 訓練時間：微調耗時較長（1106.29 秒對比 1010.96 秒）。

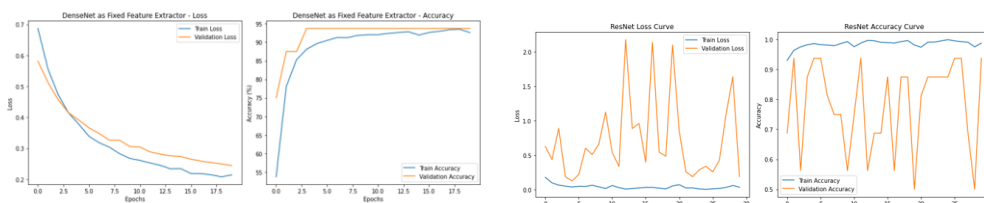
總結： fixed Feature Extractor 穩定且表現佳； Fine-tuning 準確率略高但需控制過度擬合風險。

## 1.5 Task E: Test Dataset Analysis



Test Accuracy	Test Loss	lr	Weight_decay	epochs	batch_size	Drop_out
84.22%	0.4793	0.0008	0.001	20	32	0.6

此為 lab 4's best performance。



DenseNet-121 as a fixed feature extractor

Test Loss: 0.3325, Test Accuracy: 87.50%

learning_rate <sup>1</sup>	epochs <sup>2</sup>	weight_decay <sup>3</sup>	batch_size <sup>4</sup>	Training Time <sup>5</sup>	Test Loss <sup>6</sup>	Test Accuracy <sup>7</sup>
1e-3 <sup>1</sup>	30 <sup>2</sup>	1e-4 <sup>3</sup>	32 <sup>4</sup>	1612.61s <sup>5</sup>	0.4301 <sup>6</sup>	89.42% <sup>7</sup>

而這次的 DenseNet-121 and ResNet-50 不論有沒有使用 Fixed Feature Extractor 兩者 hyperparameter 隨便調的結果 accuracy 都超過 lab 4 最好的 accuracy 84.22%，可以知道 pre-trained models 效果，不管是收斂情況還是 accuracy 都明顯好過 lab 4's result。

我認為的原因為：

1. 缺乏 Pre-Trained 知識：Lab 4 的模型似乎是從頭開始訓練的，或者只接受了有限的 Pre-Trained。在沒有大量資料學習的基礎上，模型難以有效地捕捉高層次特徵，導致收斂速度較慢且準確率較低。相較之下，DenseNet-121 和 ResNet-50 作為 Pre-Trained 模型，已經從大型資料集中學習了許多有用的特徵，使它們能更快適應較小的 x 光資料集，達到更好的準確率和更穩定的損失曲線。

2. 小資料集上的 overfitting：Lab 4 模型的 Val 損失波動較大，準確率也不穩定，這顯示了 overfitting 的情況。由於 x 光資料集相對較小，非 Pre-Trained 模型在 generalized 能力上會遇到挑戰。Pre-Trained 模型則利用了已學到的廣泛特徵提取能力，能更好地避免 overfitting，並提升測試準確率。

3. hyperparameter 調整的不足：Lab 4 中的學習率、權重衰減（weight decay）、以及 dropout 等超參數可能未經過充分優化。相對而言，預訓練模型（DenseNet-121 和 ResNet-50）即使只經過簡單的 hyperparameter 調整，表現也明顯更好，這說明 Lab 4 模型的超參數設定可能不足以達到最佳性能。

4. 收斂與訓練效率：Lab 4 模型的損失曲線較不穩定，且收斂速度較慢；相比之下，DenseNet 和 ResNet 由於有 Pre-Trained 權重，能快速穩定下來。這顯示 Lab 4 模型在收斂上存在困難，導致準確率較低。Pre-Trained 模型由於有更好的初始參數，因此能更快地達到穩定。