1. Implementation Details 313551176 王駿睿
   1. Unet 架構模型

本次實作先建構了一個元件CC (2次conv)，有別於原始Unet的捲積單位，本次實作對於2次的捲積中，將channel的數量做漸進式遞增或遞減，避免突然大量流失資訊，另外本次實作了參數feature=[64,128,256,512]，其中[64,128,256,512]代表Unet整體架構中每一層的通道數量，而Unet最底部bottleneck的通道數量為feature[-1] \* 2，此例子中就是1024，並可以透過增加feature的數量，來增加Unet模型的深度。

* 1. Resnet34\_Unet

本次實作延續CC元件，額外增加Resnet34架構中的殘差捲積元件BasicBlock，在encoder中所有架構和channel數量皆和Resnet34相同，decoder和skip的操作參考Lab2敘述中的圖片架構。

* 1. 訓練過程

loss函數採用dice loss，其值為 1 – dice score，此設計希望training和validation所要達成的目標一致。optimizer使用Adam，並使用scheduler調控learning rate，在一定epoch過後lr下降一定比例，並加上 early stop機制，當模型的validation loss沒有明顯下降，會自動停止訓練。

* 1. utils
     1. EarlyStopping:

參數patience，當validation loss超過固定epochs沒有改善，就中斷訓練。

參數min\_delta，作為validation loss最小需改善的量。

參數path，模型暫存路徑。

* + 1. dice\_score:

epsilon為0.1，避免除以0，可支援多筆資料，返還dice score的平均值。

* + 1. dice\_loss:

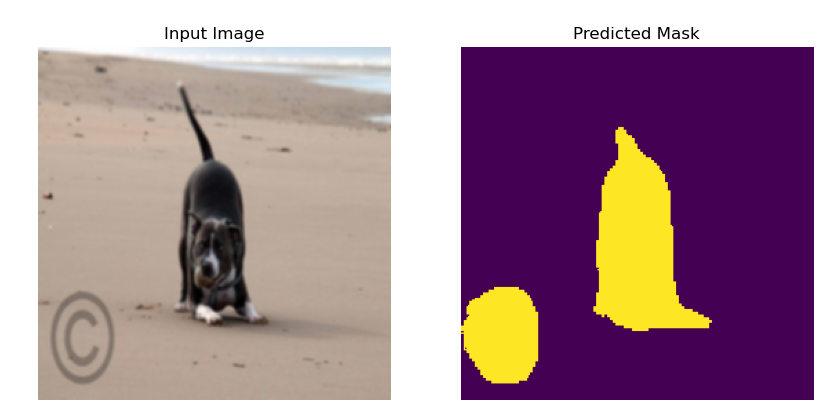
和dice\_score對應，回傳1-dice\_score。

* + 1. check\_loss\_line:

給訓練過程中的train loss和validation loss，顯示訓練loss曲線圖。

1. Data Preprocessing

在觀察資料當中，有發現有些照片有浮水印的標籤，以及些許的簽名，甚至有些圖片還有一些噪點，但是這些照片的邊界都非常清晰明確。對於這些和模型預測無關或不穩定的因素，本次有實作將資料先經過標準化、低通濾波器、去除陰影，抹平雜訊的干擾和將浮水印與背景融合減少下圖發生機率



除此之外也有把training data做資料擴增，每次取data時，會有50%的機率將照片左右翻轉，以增加資料的多樣性。

(沒有做上下翻轉是因為資料分布不太會有照片上下顛倒的情況)

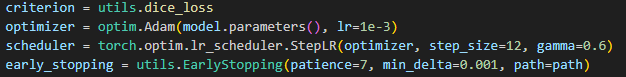
1. Analyze the experiment results

總體來說Unet和Resnet34\_unet收斂後最佳表現皆落在dice loss < 0.1中，也就是dice score > 0.9

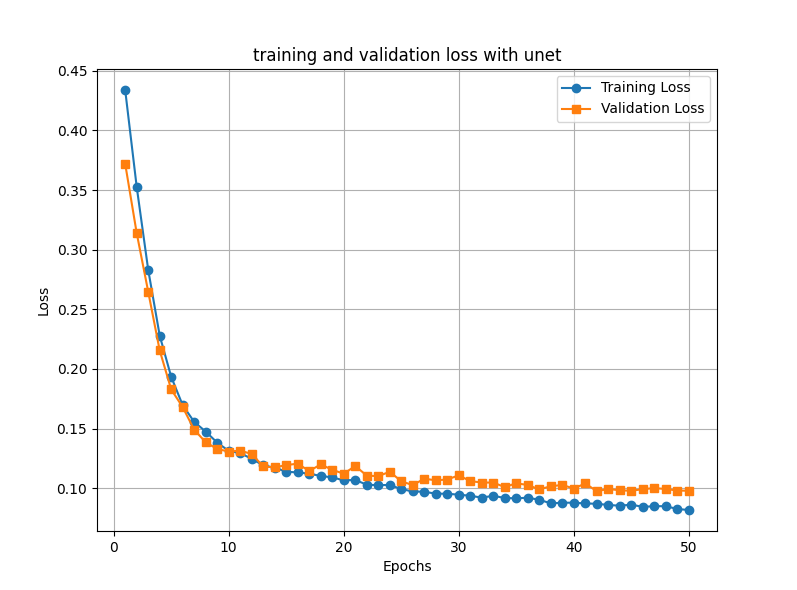
所有的實驗圖片數據皆在./src/result當中

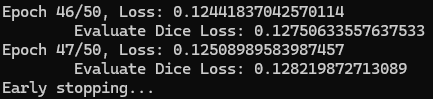
* 1. Unet
     1. feature layer

測試Unet整體架構的深度影響

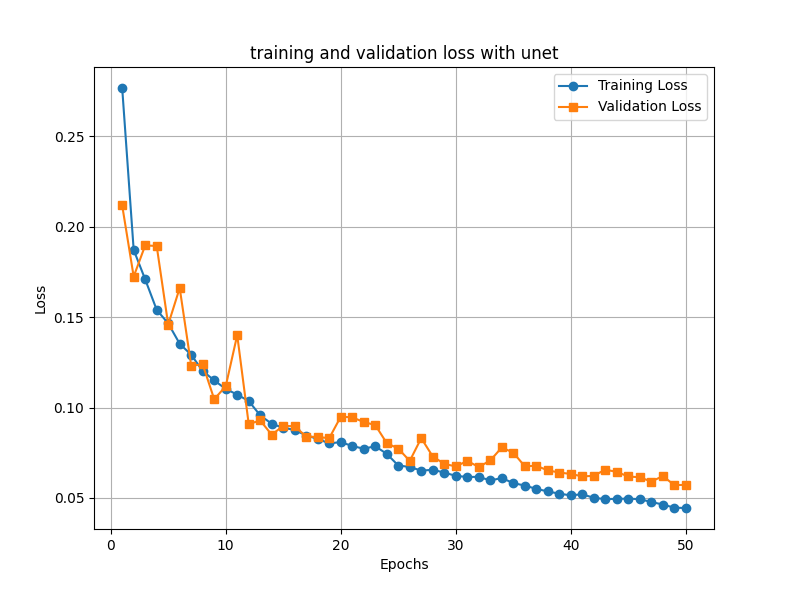


* + - 1. feature = [4, 8, 16, 32]





* + - 1. feature = [64, 128, 256, 512]

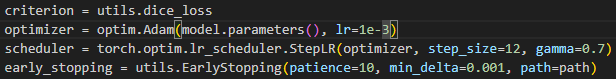


* + - 1. little conclusion

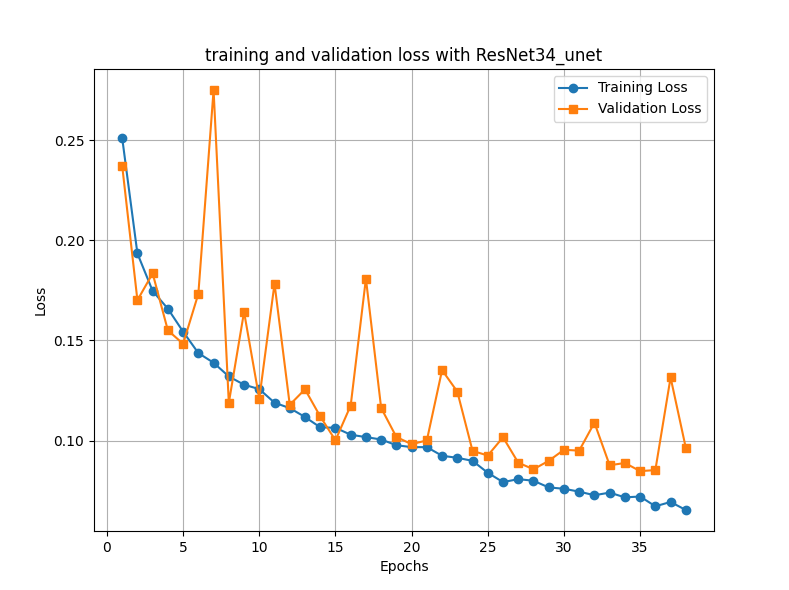
架構越深代表模型能夠掌握更多數據，又因為Unet的特性，不會因為架構變深而遺忘先前資訊。

* 1. Resnet34\_unet
     1. learning rate

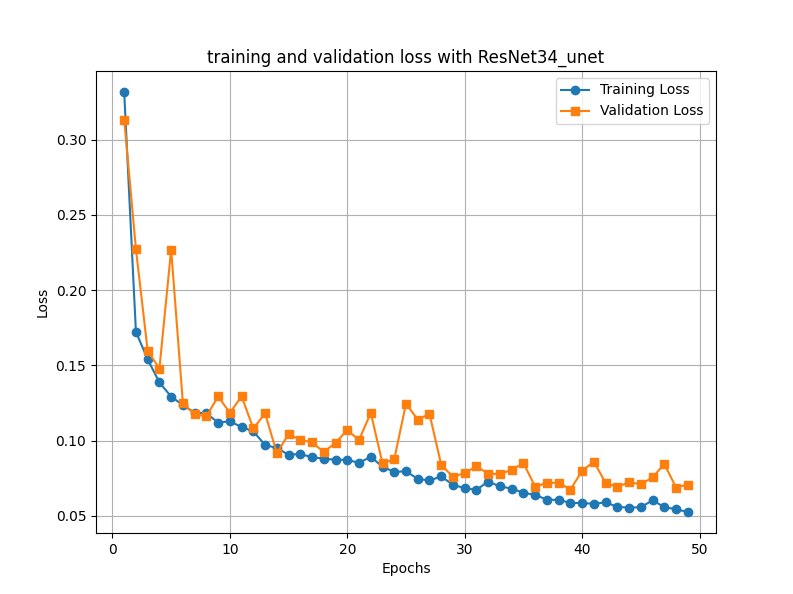
測試lr對模型收斂的影響



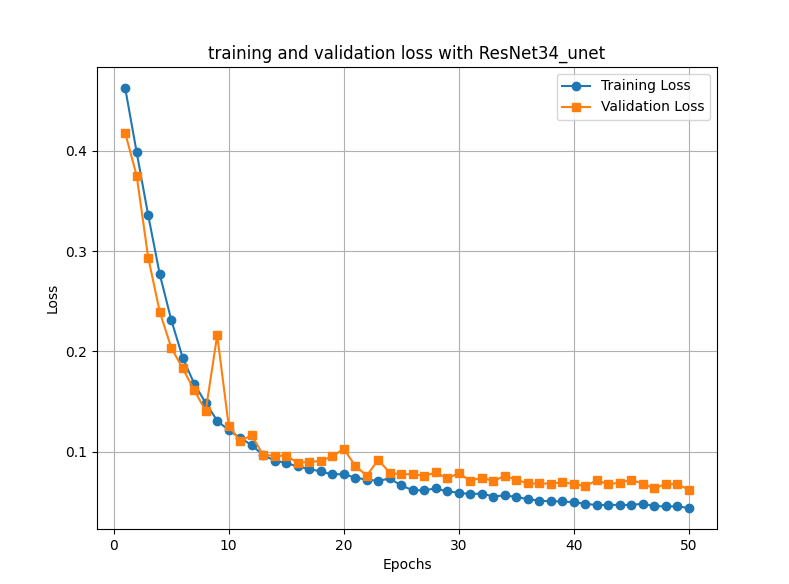
* + - 1. lr=1e-2



* + - 1. lr=1e-3



* + - 1. lr=1e-4



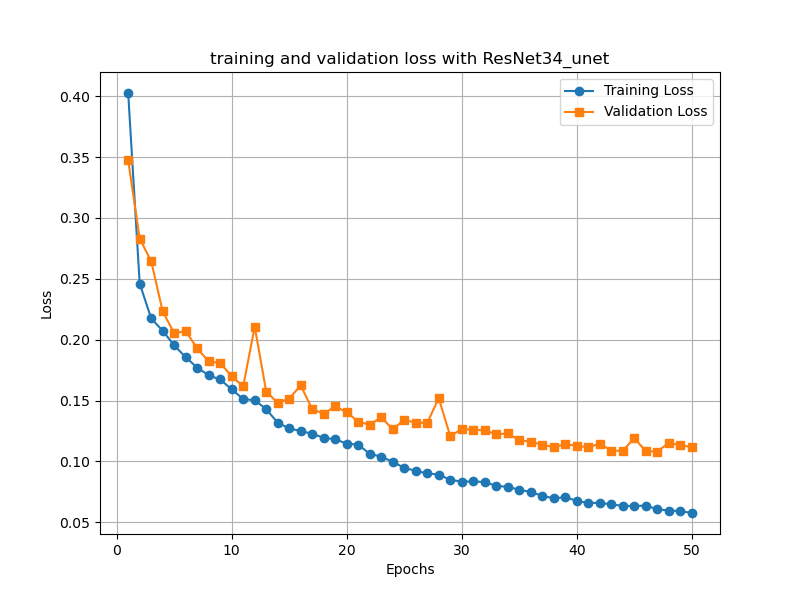
* + - 1. little conclusion

可以發現越小的lr模型越趨於穩定，validation loss的震盪也越小。

* 1. 其餘的單一實驗
     1. 去掉陰影

在觀察模型的inference，發現有些有陰影的情況模型不好預測，因此在做資料前處理的時候多加上了一層remove\_shadow避免干擾，結果dice score沒有明顯提升，反而造成了寫許overfitting。





1. Execution steps
   1. 環境: 以下皆在linux上運行

python -m venv env

source env/bin/activate

python -m pip install --upgrade pip

pip install -r requirements.txt --extra-index-url <https://download.pytorch.org/whl/cu124>

(cu124為cuda版本，根據硬體替換)

* 1. Training

訓練Unet: python ./src/train.py -m 0

訓練ResNet34\_unet: python ./src/train.py -m 1

* 1. Inference

Unet: python ./src/inference.py -m 0 --model ./saved\_models/unet.pth

ResNet34\_unet: python ./src/inference.py -m 1 --model ./saved\_models/ResNet34\_unet.pth

1. Discussion

當模型進行計算時，U-Net 的核心設計就是在於保留先前圖片的資訊。如果是需要在圖片上標記區塊的任務，若僅使用一般 CNN，特徵經過多層CNN處理後可能已經和原始圖片相差甚遠，導致無法很好標記。而 U-Net 透過Skip Connections，將encoder中對應層的特徵直接傳遞到decoder，這樣可以補充該層先前計算的細節，確保模型在慢慢恢復原始圖片大小的過程中仍能保有完整的空間資訊，而提升標記能力。而論文中也有佐證說U-Net架構下對於像是醫學方面，不僅只需要少量數據即可訓練，推理速度也相較為快速(512x512可以在1秒內完成)。

reference: <https://arxiv.org/abs/1505.04597>