

DL Lab5: Semantic Segmentation

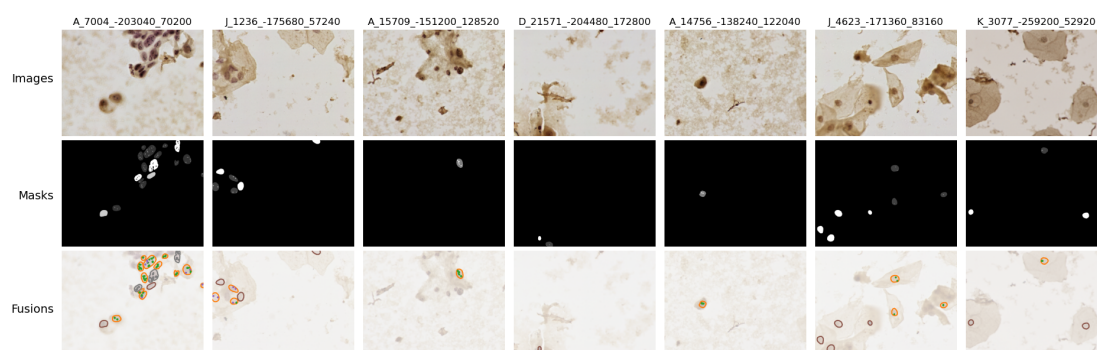
Data preprocessing

- CCAgT dataset: <https://data.mendeley.com/datasets/wg4bpm33hj/2>

讀取資料：透過路徑將每個子資料夾的影像存進 `images` 和 `masks` 變數中。

資料處理：每開啟一組 `image` 和 `mask` 影像時，就使用自定義的 `RandomRotateFlipTransform` 函數進行隨機 180 度旋轉及隨機垂直翻轉，用來增加資料多樣性。接著將資料透過原始比例 `resize` 成 `624*832` 的大小，`interpolation` 分別使用 `BILINEAR` 和 `NEAREST`，並且經過 `ToTensor` 轉換，其中 `mask` 還要乘以 255 再轉成 `long` 型態。此外，`image` 要進行 `Normalize`，使用的是 `ImageNet` 資料集的全局平均值和標準差。最後透過 `random_split` 將整個資料集分為訓練集(7)、驗證集(1)和測試集(2)，訓練集會進行 `shuffle`，確保訓練資料的隨機性，`batch size` 均設為 10。

首先在原始影像上覆蓋一層半透明的白色層，接著透過 `Sobel` 運算檢測 `Mask` 的邊緣，再將不同類別的邊緣進行著色，形成 `fusion` 的影像。



U-Net neural network

參考PyTorch官方的Sample code實作兩種4層的U-Net神經網路模型：

- (1) 在兩個卷積層中間加入`dropout(0.5)`，並使用空間注意力機制 (`Spatial Attention`)，此機制注重於影像中的空間位置，有助於模型集中在影像中的某些區域。

為了提升模型的泛化能力，在`Optimizer`的部分有嘗試加上 `weight_decay`，即L2正規化。






由於上述模型能正確的分割出目標，但卻多分割了許多不重要的區域，因此新的模型將對注意力機制進行改善。

(2) 使用通道注意力機制 (Channel Attention)，此機制適用於需要模型理解和區分不同類別特徵（如顏色、紋理等）的任務。它能為不同通道分配不同的權重，以突出當前任務中更重要的通道。

*CrossEntropy：使用class_weights，藉此讓模型更好的專注在特定的區域上。

*Early stop：當validation loss提升0.01五次，就停止訓練。

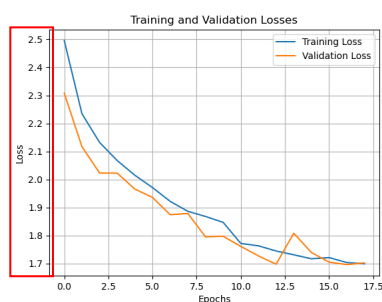
表格為其中五次訓練資訊[1]~[5]，橘色字體為針對上次訓練進行修改的部分。

	[1]	[2]	[3]	[4]	[5]
Transform	None	Rotate/Flip	Rotate/Flip	Rotate/Flip	Rotate/Flip
Resize	624*832	624*832	624*832	576*768	576*768
Attention	Spatial	Spatial	Spatial	Spatial	Channel
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001	0.001	0.001
Weight decay	None	None	0.01	None	None
Scheduler (step / gamma)	10 / 0.5	10 / 0.5	7 / 0.5	10 / 0.5	10 / 0.5
Epoch (Early stop)	24 best: 19	18 best: 17	50 (None) best: 44	50 (None) best: 30	50 (None) best: 48
Dice loss	All data mean	All data mean	All data mean	Per batch mean	Per batch mean
Loss					

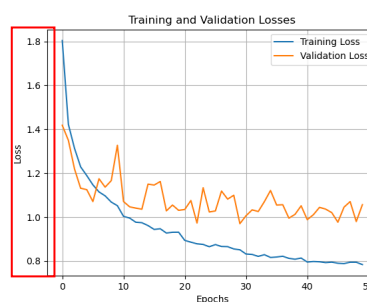
The training loss and validation loss during training

此項任務使用Cross entropy loss和Dice loss的加總來計算總體損失，在Dice loss的部分有使用了兩種計算方式：

- (1) 將所有類別的預測結果視為一體計算平均，不區分不同樣本或batch。[2]
- (2) 在batch內對每個樣本單獨處理再取平均，考慮了每個樣本的獨特性。[4]



[2]

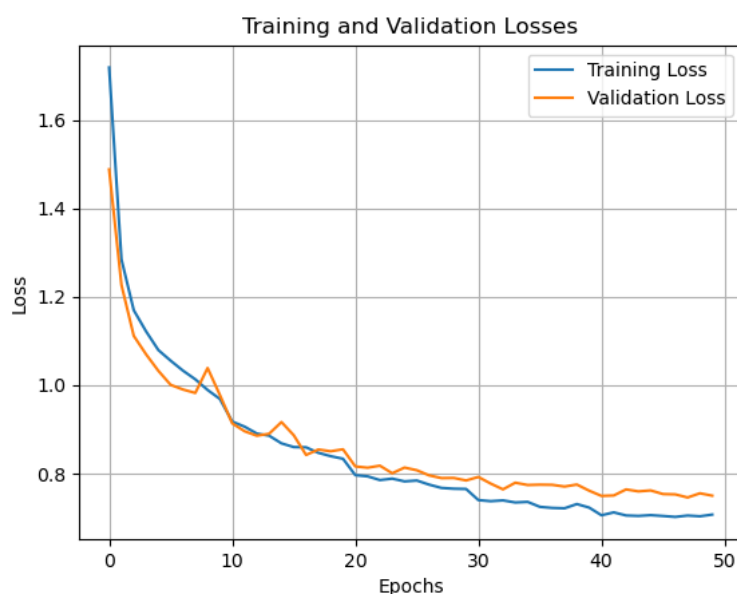


[4]

可以看到兩種計算方式的loss範圍差距很大。深入了解後，我發現第一種方法專注於僅使用Dice loss的情況，而第二種方法較適用於需要同時考慮Cross entropy loss和Dice loss的情況。

最終使用第二種Dice loss計算方式，訓練過程的Loss如下：

由於Training loss穩定下降並趨於穩定，加上時間及設備硬體的限制，這次作業有試了許多方法進行調整，但沒有嘗試訓練更多個epoch。



[5]

Metrics Method

以下四種評估指標的計算方式參考：https://github.com/tumeng0302/lab5_utilities

- ⊙ **Pixel Accuracy (PA): 0.6300**，表示模型正確分類像素的比例。
- ⊙ **Dice: 0.8901**，衡量預測分割與真實分割之間的重疊程度。
- ⊙ **Intersection over Union (IoU): 0.5488, 0.4124, 0.1352, 0.4980, 0.1385, 0.0966, 0.4069, 0.0187**，衡量每個類別的預測和真實標籤之間的重疊程度。
- ⊙ **mean Intersection over Union (mIoU): 0.2795**，所有類別 IoU 的平均值。

以下三種評估指標為自己計算的：

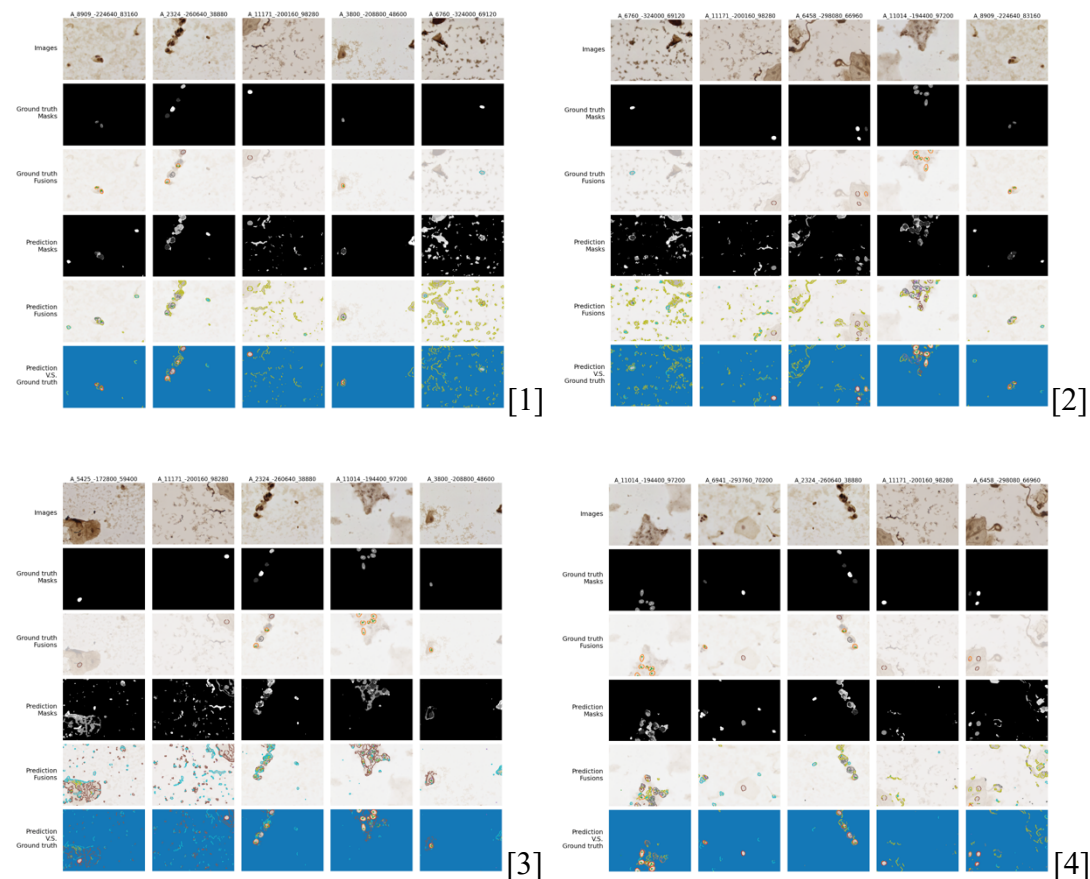
- ⊙ **class Pixel Accuracy (cPA): 0.9798, 0.3483, 0.2609, 0.0772, 0.2060, 0.0654, 0.0110, 0.0620**，分別為每個類別計算像素精度。
- ⊙ **mean Pixel Accuracy (mPA): 25.1378**，所有類別像素精度的平均值。
- ⊙ **mean Average Precision (mAP@[.5:.95:.05])**: 所有類別AP值的平均值，是一個綜合考慮模型在所有類別上性能的指標。

*多次嘗試計算mAP都會突然清除kernel，詳細計算方式請看ipynb檔。

Output graphical presentation

最後一系列的影像透過將預測邊緣疊加在真實的fusion影像上做比較，若預測邊緣和真實邊緣重疊且為相同類別，則將顏色改為紅色；若兩者不重疊，則分別顯示原始顏色。此外，真實Mask區域會保留真實fusion影像的背景顏色，其餘背景則設定為藍色。

可以看到[1]~[4]的分割效果較不明確，且[1]~[3]幾乎未分割為正確的類別。



最終的訓練結果圖如下：

有些樣本有準確的預測，有些樣本則預測為錯誤的類別，且預測出部分不重要的區域。若有更多的時間或更好的資源，應該能嘗試更多的調整。

