

a. Reproducibility of the results (6 pts)

```
def seed_everything(seed):
    # Set Python random seed
    random.seed(seed)

    # Set NumPy random seed
    np.random.seed(seed)

    # Set PyTorch random seed for CPU and GPU
    torch.manual_seed(seed)
    if torch.cuda.is_available():
        torch.cuda.manual_seed_all(seed)

    # Set PyTorch deterministic operations for cudnn backend
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
```

b. Number of parameters: Please write the parameter count of your final selected model to the Kaggle competition (4 pts)

```
model = UNet()
total_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print(f"Total number of parameters: {total_params}")
```

Total number of parameters: 31043586

c. The difficulty during training (8 pts)

Ans:我覺得一開始比較難得是這次因為結合 W&B，所以 code 的模式跟之前有一點不一樣，花了一些時間去理解每一段 code，最後有成功連上 W&B 真的很開心，然後花了蠻多時間去理解 Unet 為什麼要這樣設計編碼器與解碼器。

d. Briefly explain the structures of the models you are using: You are required to do analyses using FCN-8s and “U-Net (12 pts)

Ans:

FCN-8s:

**Encoder**：使用基於深層卷積網路（如 VGG-16）的特徵提取模塊，逐步降低特徵圖的空間解析度

**Decoder**：利用上採樣將低解析度特徵逐步還原為原始圖像大小

**Skip Connections**: 結合來自不同層的高解析度與低解析度特徵以提升細節分割精度

U-Net:

**Encoder**：多層卷積+池化操作，提取不同尺度的特徵

**Decoder**：逐層進行上採樣，還原特徵空間解析度

**Skip Connections:** 直接將編碼器的特徵連接到解碼器，保留更多高解析度細節

我覺得 **U-Net** 的跳躍連接處理得更細緻，比 **FCN-8s** 更適合小資料集的醫學影像分割，故使用之