深度學習 作業五

學號：312831002　姓名：廖健棚

UNet

Double Convolution base

使用 2 層 Convolution2D+BatchNorm2d+ReLU

```
class DoubleConv(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(DoubleConv, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True)
        )
    def forward(self, x):
        return self.conv(x)
```

UNet Structure

Encoder channel 從 64 到 1024, decoder 則是從 1024 到 64

```
# Encoder
self.conv_down1 = DoubleConv(in_channels, 64)
self.conv_down2 = DoubleConv(64, 128)
self.conv_down3 = DoubleConv(128, 256)
self.conv_down4 = DoubleConv(256, 512)
self.conv_down5 = DoubleConv(512, 1024)  # Additional layer
self.maxpool = nn.MaxPool2d(kernel_size=2)
# Decoder
self.upsample = nn.Upsample(scale_factor=2, mode='bilinear', align_corners=True)
self.conv_up4 = DoubleConv(1024+512, 512)  # Additional layer
self.conv_up3 = DoubleConv(512+256, 256)
self.conv_up2 = DoubleConv(256+128, 128)
self.conv_up1 = DoubleConv(128+64, 64)
```

實驗設定

使用 UNet 架構訓練圖片的 Mask,有鑑於圖片大小太大，無法直接訓練，需要對圖片做索小，目前方式是先將圖片縮小至 800x800 的 tensor 進行訓練，並在要驗證 metrics 時將圖片還原，與 ground true 做比較，以下是將預測的結果進行縮放的函式。

```
mask = F.interpolate(mask, size=size, mode="nearest").squeeze(0)
```

訓練條件:

Input channel:3(RGB), output channel:8(mask0-7)

Loss: CrossEntropy()

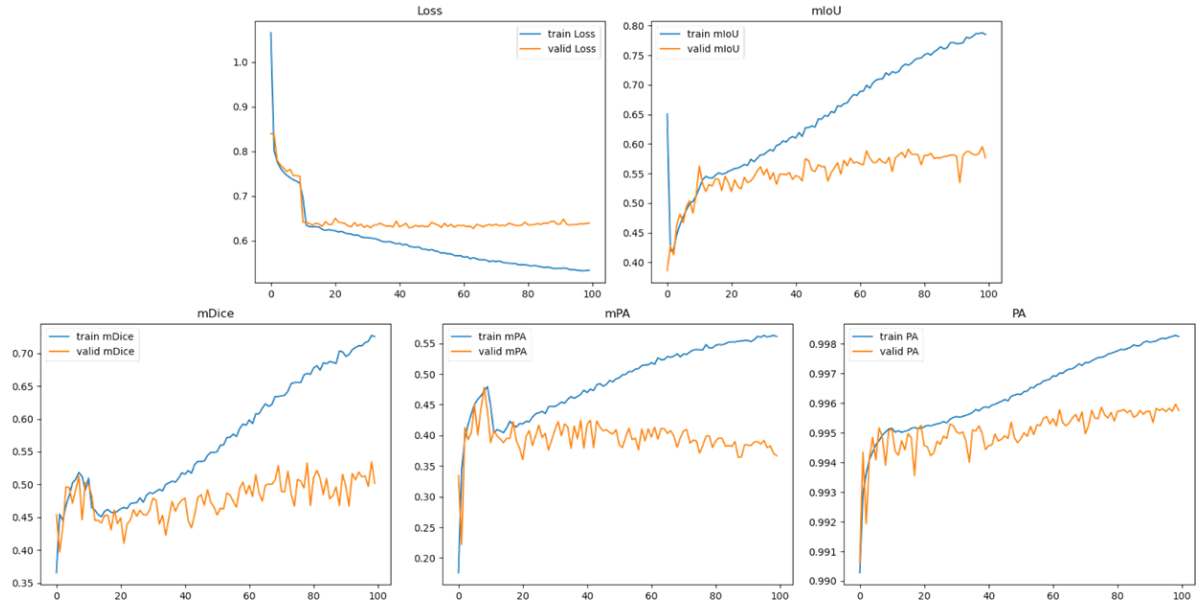
lr:1e-5, 100 epochs

optimizer = RMSprop, L2 penalty:1e-8, momentum:0.9

訓練過程

在訓練過程中前 20 代有明顯下滑，但是後期便有 overfitting 的趨勢

Metrics 在訓練過程中 mdice(mean dice of class)、miou 及 mPA 都可以大略當作評估預測結果的指標，但是像是 PA(Pixel Accuracy)就不較具有參考性，因為在 mask 當中 0 為 mask 中的背景佔據大範圍且要較容易猜測，所以數值會一直趨近於 99。



訓練結果

資料切分

Train set size: 6538, Valid set size: 934, Test set size: 1867

驗證 function

miou 計算各類別的聯集分之交集

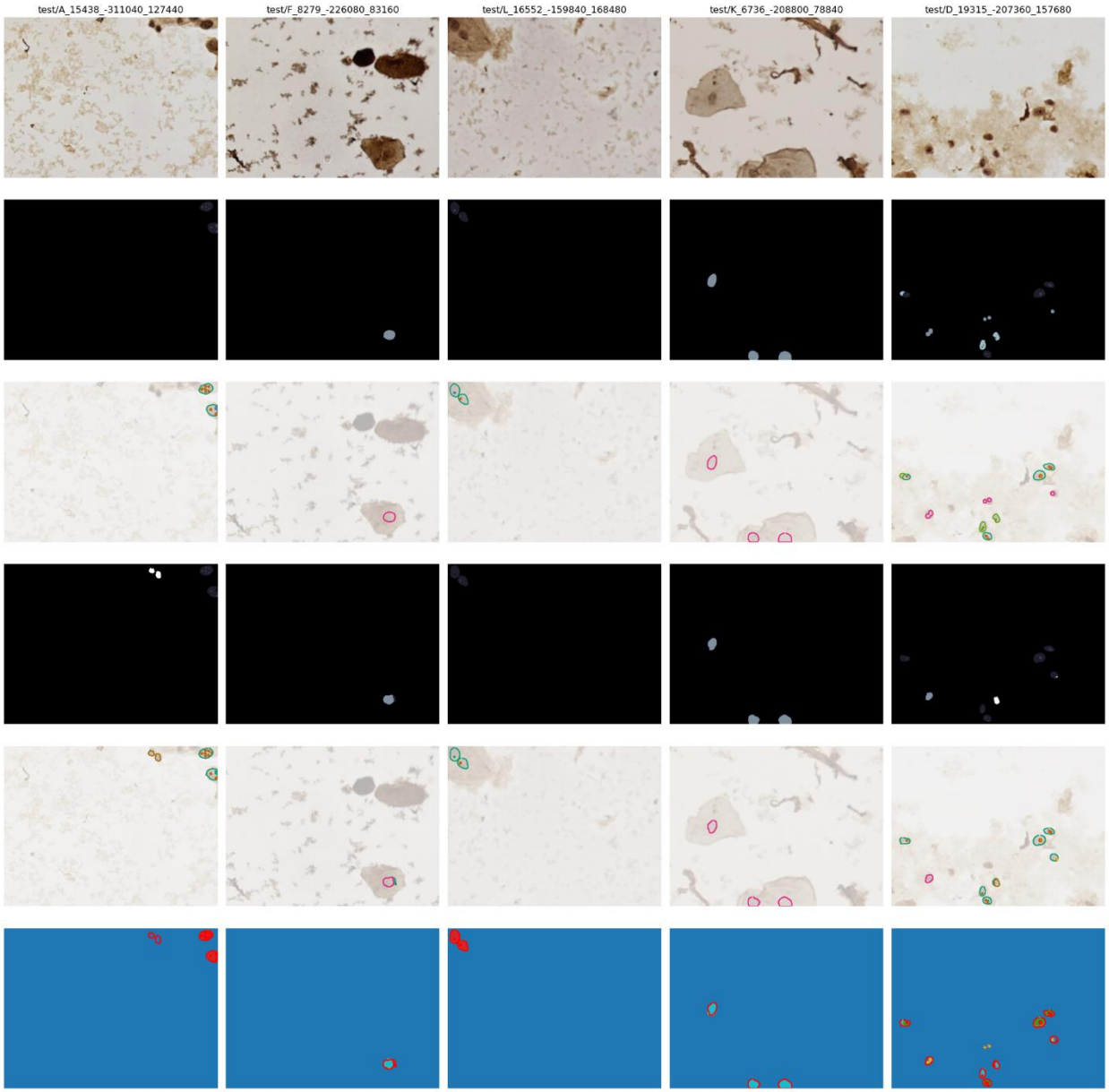
mice 計算各類別兩集合的交集大小相對於兩集合大小總和的比例

```
def calculate_mean_iou(predicted_masks, ground_truth_masks, num_classes):
    class_iou = torch.zeros(num_classes)
    for class_idx in range(num_classes):
        class_mask_pred = (predicted_masks == class_idx)
        class_mask_gt = (ground_truth_masks == class_idx)
        class_iou[class_idx] = calculate_iou(class_mask_pred, class_mask_gt)
    mean_iou = class_iou.sum() / (class_iou != 0).sum()
    return mean_iou.item()

def calculate_dice_per_class(predicted_masks, ground_truth_masks, num_classes):
    dice_scores = []
    smooth = 1e-6  # Smoothing factor to avoid division by zero

    for class_idx in range(num_classes):
        pred = predicted_masks == class_idx
        target = ground_truth_masks == class_idx
        intersection = (pred & target).sum().item()
        union = pred.sum().item() + target.sum().item()
        dice = (2. * intersection + smooth) / (union + smooth)
        dice_scores.append(dice)

    return dice_scores

def calculate_mean_dice(dice_scores):
    return sum(dice_scores) / len(dice_scores)
```

驗證結果

Test metric: Loss: 0.5784,mIoU: 0.6733,mDice: 0.5811,PA: 0.9967,mPA: 0.4952

我的模型預測可以辨識到 ground true 有 mask 的地方，但是在分類上，在很多小細節或是比較分散的，都會有辨識錯的類別。圖片在下方。



參考資料

i. Milesial. (n.d.). Milesial/Pytorch-UNet: Pytorch implementation of the U-Net for image semantic segmentation with high quality images. GitHub. https://github.com/milesial/Pytorch-UNet

ii. OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat