

說明：請各位使用此 template 進行 Report 撰寫，如果想要用其他排版模式也請註明題號以及題目內容（請勿擅自更改題號），最後上傳前，請務必轉成 PDF 檔，並且命名為 report.pdf，否則將不予計分。

學號：R12945060 系級：生醫電資所碩二 姓名：羅佳蓉

1. (0.5%) CNN model

- a. 貼上 private submission 所使用的 CNN model 程式碼。

```
class FaceExpressionNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.base_model = models.resnet18(pretrained=True)
        self.base_model.fc = nn.Linear(self.base_model.fc.in_features, 128)
        self.fc = nn.Sequential(
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(128, 7)
        )
    def forward(self, x):
        x = self.base_model(x)
        x = self.fc(x)
        return x
```

2. (1%) Data Augmentation (ref: <https://pytorch.org/vision/stable/transforms.html>)

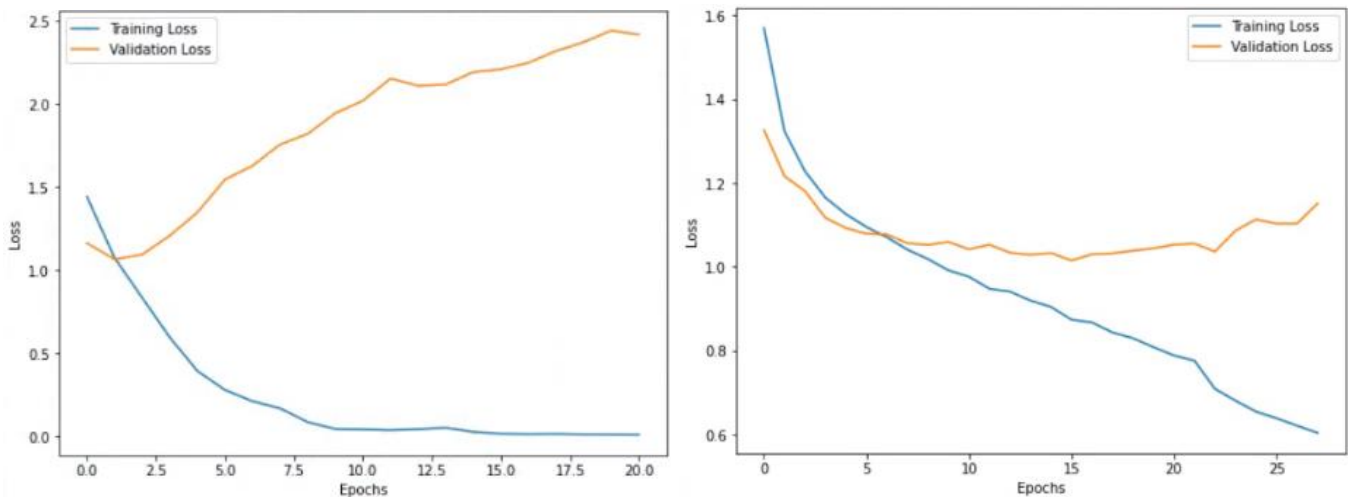
- a. 貼上這部分的程式碼，並說明為何使用這些 augmentation。

```
train_tfm = T.Compose([
    T.RandomHorizontalFlip(),
    T.RandomRotation(15),
    T.ColorJitter(brightness=0.4, contrast=0.4, saturation=0.4),
    T.RandomAffine(degrees=0, translate=(0.1, 0.1)),
    T.RandomResizedCrop(64, scale=(0.8, 1.0)),
    T.ToTensor(),
    T.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
])
```

隨機水平翻轉：有助於使模型對左右翻轉的不敏感，因為人臉可能在不同情境下會有所變化。

- 隨機旋轉：適當的旋轉可以模擬拍攝角度的變化，提高模型對圖像方向的穩健性。
- 亮度、對比度、飽和度調整：模擬不同光線條件下的圖片，有助於提高模型對光線變化的適應性。
- 隨機平移：可以模擬人物位置的小幅度變化，增強模型對不同拍攝位置的辨識能力。
- 隨機裁剪並調整大小：避免過度依賴圖像某些局部特徵，提高模型的泛化能力

- b. 畫出使用/不使用 augmentation 時的 training/validation loss curve，比較並說明其差異。



左圖和右圖分別為不使用及使用 augmentation 時的 training/validation loss curve

- 不使用 augmentation：

模型能夠快速擬合訓練數據，但由於缺乏多樣性，模型過度依賴訓練數據中的特徵，導致 validation loss 在早期過擬合，無法很好地泛化到新的驗證數據。

- 使用 augmentation

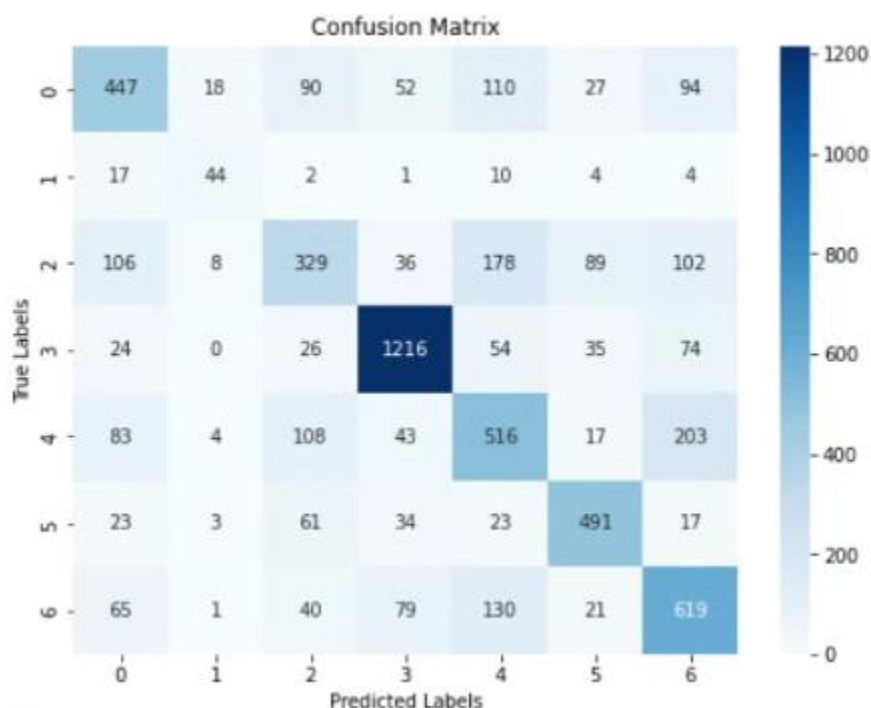
雖然訓練 loss 下降得較慢，但模型對於驗證集的表現更穩定，避免了過早過擬合。這表明資料增強能夠幫助模型學習到更多具有泛化性的特徵。

3. (0.5%) Confusion Matrix (ref: https://en.wikipedia.org/wiki/Confusion_matrix)

- a. 貼上這部分的程式碼。

```
def draw_confusion_matrix(model, valid_loader):
    predictions, labels = [], []
    model.to(device)
    model.eval()
    with torch.no_grad():
        for img, lab in tqdm(valid_loader):
            img = img.to(device)
            output = model(img)
            predictions += torch.argmax(output, dim=-1).tolist()
            labels += lab.tolist()
    cm = confusion_matrix(labels, predictions)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=[str(i) for i in range(7)],
                yticklabels=[str(i) for i in range(7)])
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title('Confusion Matrix')
    plt.show()
```

- b. 分析哪些類別的圖片容易使 model 搞混，並分析可能的原因。



- 106 張恐懼(2)的圖片被錯誤預測為生氣(0)

恐懼和生氣在面部表情上可能都有一些相似的特徵，例如張開的嘴巴或皺起的眉頭。

恐懼可能帶有強烈的面部緊張感，這有時會被模型解讀為生氣的情緒，因為生氣也會顯示出強烈的面部表情特徵。

- 108 張難過(4)的圖片被錯誤預測為恐懼(2)

恐懼和難過的相似性：難過和恐懼的表情可能在某些面部特徵上有重疊，特別是在面部肌肉收縮或眼睛緊張時。這兩種情緒都可能表現出抑鬱或焦慮的面部表情，使得模型難以區分。

背景資訊或光線變化：影像中可能存在光線或角度的變化，讓這兩者看起來更加相似。

- 178 張恐懼(2)的圖片被錯誤預測為難過(4)

恐懼和難過可能都有一種比較壓抑和負面的表情特徵，導致模型將一些帶有恐懼表情的圖像誤判為難過。這兩種情緒都可能表現出緊張感、抑鬱的臉部肌肉，特別是在眉頭和眼睛周圍的區域，使得模型難以區分。

- 130 張難過(4)的圖片被錯誤預測為驚訝(6)

難過和驚訝的表情在某些情況下可能有重疊，尤其是嘴部打開或眼睛張大的情況下（這些表情在驚訝中較為明顯，但某些難過的情緒也可能會有張大嘴巴的特徵）。這可能使模型在某些情境下誤認為驚訝。

- 110 張生氣(0)的圖片被錯誤預測為難過(4)

生氣和難過的面部表情有一些共同點，特別是在眉毛向下的時候，這些表情可能讓模型難以區分這兩個情緒。

- 203 張高興(3)的圖片被錯誤預測為難過(4)

高興和難過的模糊區別：有些高興的表情可能不夠明顯，特別是如果微笑幅度較小，這可能讓模型難以將其與難過區分開。這種情況下，模型可能錯誤地將這些較輕微的微笑判斷為難過。

光線或拍攝角度的影響：如果高興的表情受到了不良光線或角度的影響，可能讓模型無法準確地識別笑容，從而將其誤判為其他情緒。