

HW6

612415013 蕭宥羽

1. How to execute codes.

Original

```
data transforms = {  
    'train': transforms.Compose([  
        transforms.Resize((224,224) ),  
        #####在此區塊填入圖像轉換方法#####  
        #####  
        transforms.ToTensor(),  
    ]),  
    'val': transforms.Compose([  
        transforms.Resize((224,224) ),  
        transforms.ToTensor(),  
    ]),  
}
```

Weak Augmentation

使用三種不同的 Weak Augmentation

水平翻轉

```
'train': transforms.Compose([  
    transforms.Resize((224,224) ),  
    #####在此區塊填入圖像轉換方法#####  
    transforms.RandomHorizontalFlip(),  
    #####  
    transforms.ToTensor(),  
]),
```

垂直翻轉

```
'train': transforms.Compose([  
    transforms.Resize((224,224) ),  
    #####在此區塊填入圖像轉換方法#####  
    transforms.RandomVerticalFlip(),  
    #####  
    transforms.ToTensor(),  
]),
```

高斯模糊

```
'train': transforms.Compose([  
    transforms.Resize((224,224) ),  
    #####在此區塊填入圖像轉換方法#####  
    transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0)),  
    #####  
    transforms.ToTensor(),  
]),
```

Strong Augmentation

Strong Augmentation 使用以下五種方式

```
'train': transforms.Compose([
    transforms.Resize((224,224) ),
    #####在此區塊填入圖像轉換方法#####
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.ColorJitter(brightness=0.1, contrast=0, saturation=0, hue=0),
    transforms.RandomRotation(degrees=5),
    transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0)),
    #####
    transforms.ToTensor(),
]),
```

Code

image_datasets 字典使用了這些資料轉換方法來建立訓練集和驗證集的資料集對象。對於每個資料集，都使用 datasets.ImageFolder 類來加載資料，並將對應的資料轉換方法應用於該資料集的圖像。image_datasets 中的每個元素都代表了一個資料集，它包含了圖像數據和對應的標籤，而這些圖像數據已經根據指定的轉換方法進行了預處理或增強。

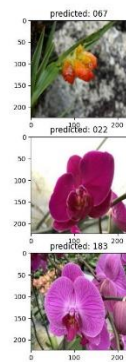
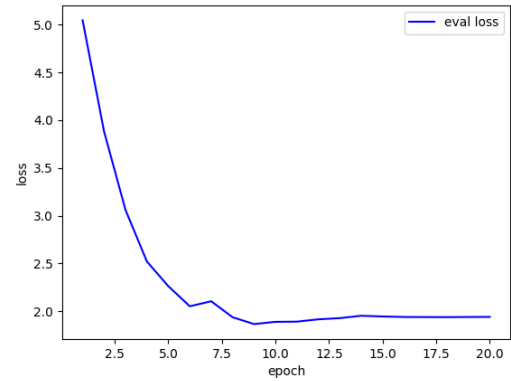
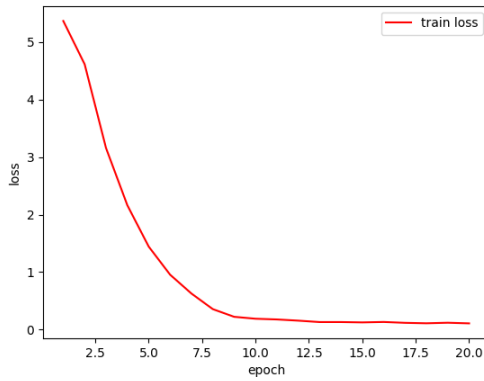
接著透過 torch.utils.data.DataLoader 加載成批次的資料。

```
# 資料集載入 =====
data_dir = 'ML_CNN/dataset/training'
image_datasets = {
    x: datasets.ImageFolder(
        os.path.join(data_dir, x),
        data_transforms[x]
    )
    for x in ['train', 'val']
}
dataloaders = {
    x: torch.utils.data.DataLoader(
        image_datasets[x],
        batch_size=batch_size,
        shuffle=True,
        num_workers=num_workers
    )
    for x in ['train', 'val']
}
dataset_sizes = {x: len(image_datasets[x]) for x in ['train', 'val']}
class_names = image_datasets['train'].classes
# 資料集載入 =====
```

2. Experimental results

Original

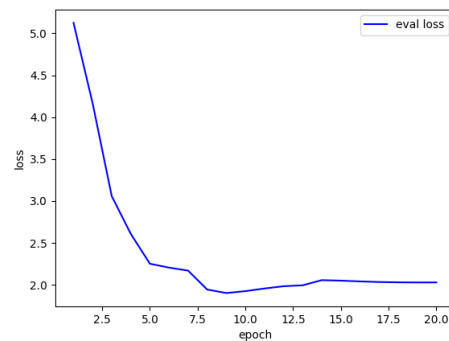
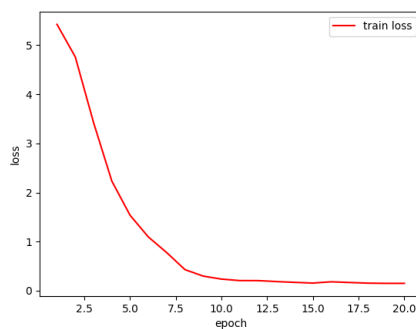
```
Training complete in 3m 27s  
Best val Acc: 0.549467
```

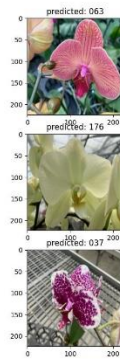
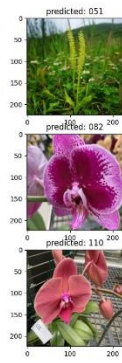


Weak Augmentation

I. `transforms.RandomHorizontalFlip()`,

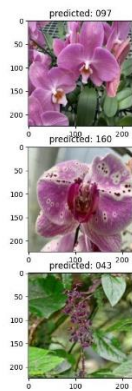
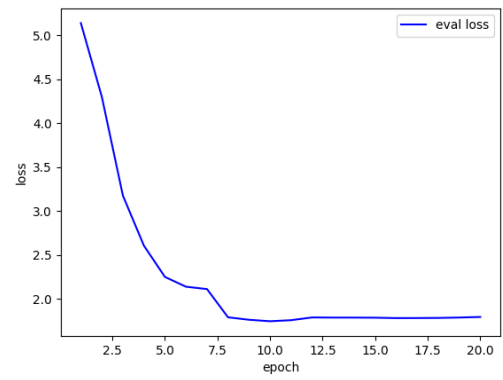
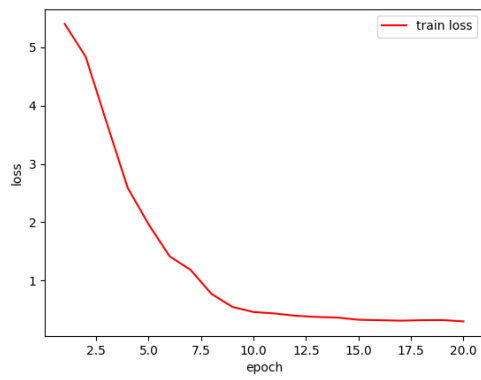
```
Training complete in 2m 42s  
Best val Acc: 0.560122
```





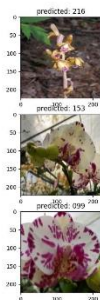
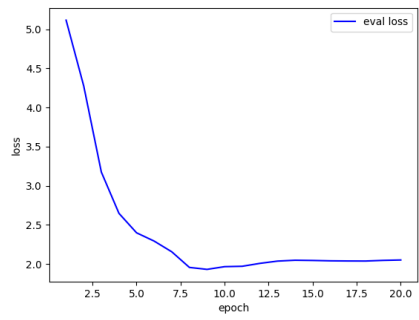
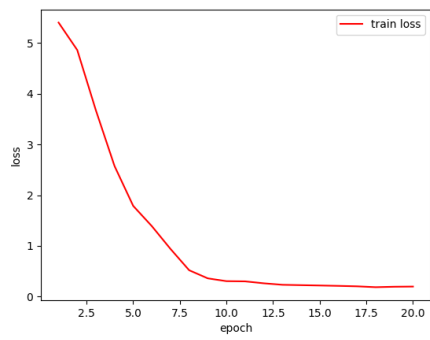
II. `transforms.RandomVerticalFlip()`,

Training complete in 3m 2s
Best val Acc: 0.570776



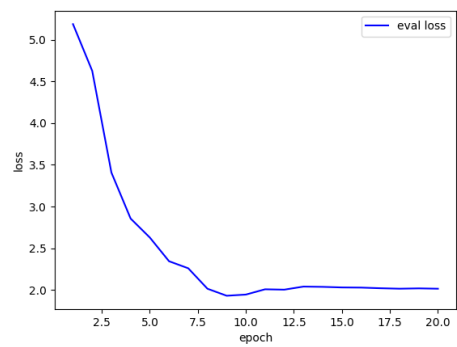
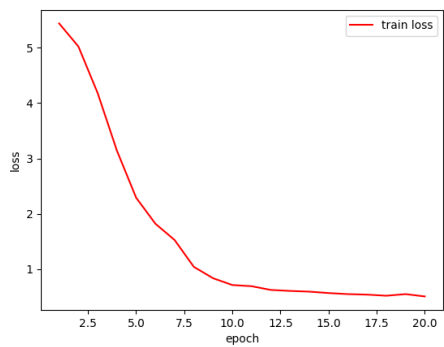
III. `transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0))`,

Training complete in 3m 21s
Best val Acc: 0.538813



Strong Augmentation

Training complete in 4m 4s
Best val Acc: 0.537291



3. Conclusion

✚ **Weak Augmentation** - 每次只選擇一到二種 data transforms 的方法，比較只使用原資料集與使用增強後的資料集模型準確率的差異。請測試比較三種不同的 Data transforms 的方法。

I. transforms.RandomHorizontalFlip()

這種方法會隨機水平翻轉圖像，準確率從原始的 0.54 提升到 0.56

水平翻轉通常對於物體識別任務有幫助，因為許多物體在水平翻轉後仍然可識別

II. transforms.RandomVerticalFlip()

這種方法會隨機垂直翻轉圖像，準確率從原始的 0.54 提升到 0.57

垂直翻轉通常對於自然場景影像等任務有幫助，但對於有方向性的物體可能會降低準確率，但在這次的實驗中影響似乎沒有到很大

III. transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0))

這種方法會對圖像施加高斯模糊，模糊處理會降低圖像細節，對於需要高分辨率細節的任務可能會降低準確率準確率從原始的 0.54 下降到 0.53

✚ **Strong Augmentation** - 一次使用 4~6 種 data transforms 的方法，比較只使用原資料集與使用增強後的資料集模型準確率的差異

在 **Strong Augmentation** 中我們同時使用了 5 種不同的資料轉換方法，包括：

```
transforms.RandomHorizontalFlip()  
transforms.RandomVerticalFlip()  
transforms.ColorJitter(brightness=0.1, contrast=0, saturation=0, hue=0)  
transforms.RandomRotation(degrees=5)  
transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0)),
```

原始資料集的模型準確率為 0.54，而在應用了上述 5 種強資料增強方法後，模型準確率下降到 0.53。這個結果顯示，在這次作業中，同時使用過多的資料增強方法反而可能會對模型產生負面影響。我認為可能的原因有以下幾點：

過度的資料扭曲 - 同時應用過多轉換可能會導致圖像失真過度，使得模型難以辨識關鍵特徵。

不相容的轉換組合 - 某些轉換方法的組合可能會相互抵消或引入噪聲，降低資訊量。

資料分佈偏移 - 過多轉換可能會使增強後的資料分佈與原始資料分佈偏離過大，影響模型泛化能力。

✚ 比較一、二題的結果，說明你的實驗中對於影像資料擴增的結論

我觀察上面兩組實驗後，我得到了以下幾點結論到以下幾點：

- ◆ 適度的弱資料擴增是有益的，可提升模型準確率和泛化能力。
- ◆ 但過度強烈的資料擴增，同時使用過多轉換方法，可能會適得其反，降低模型性能。
- ◆ 在設計資料擴增策略時，需要權衡不同轉換方法的效果，尋找最佳組合和強度。
- ◆ 除了準確率，也要關注模型在增強資料上的收斂情況，避免過度擬合。

4. Discussion

在進行這項實驗之前，我一直抱持著強資料增強(Strong Augmentation)會比弱資料增強(Weak Augmentation)取得更好效果的想法。畢竟，強資料增強同時應用了多種資料轉換技術，看似更加強大和全面。然而，當實驗結果呈現在眼前時，我卻陷入了些許疑惑和困惑。

強資料增強的表現不但沒有如我預期那般超越弱資料增強，反而還遜色於後者。這令我開始質疑，是否是我選擇的增強技術不夠恰當？又或者程式碼中存在某些錯誤導致了這樣的結果？我反覆檢視自己的做法，試圖找出可能的原因。經過深思熟慮，我意識到也許當初選擇的強資料增強技術組合並不够優秀，沒能很好地發揮正面作用。這提醒了我，資料增強絕不是盲目堆疊轉換技術，而是需要審慎挑選和調整的過程，否則可能會適得其反，反倒讓模型效能下降。這項實驗經歷讓我重新審視了資料增強這一技術。我領悟到，任何看似強大的技術，如果使用不當都可能會事與願違。因此，在未來的實踐中，我將更加謹慎地挑選和調整資料增強策略，努力尋找最佳組合，避免過度增強導致的噪聲和失真問題。