

tags: Machine Learning

HW6

408410042 林靖紳

1. Weak Augmentation

- 只使用原資料集：

- Best accuracy: 0.554

```
-----  
100%|██████████| 24/24 [00:13<00:00, 1.81it/s]  
train Loss: 0.1224 Acc: 0.9811  
100%|██████████| 11/11 [00:06<00:00, 1.82it/s]  
val Loss: 1.9921 Acc: 0.5510  
Training complete in 6m 29s  
Best val Acc: 0.554033
```

- 使用

transforms.RandomAdjustSharpness(sharpness_factor=2.5)

- 隨機地調整圖像的銳化程度
- sharpness_factor 是一個參數，它控制著調整銳化程度的強度。
- sharpness_factor 的值越高，圖像的銳化程度越高
- Best Accuracy: 0.557

```
-----  
100%|██████████| 24/24 [00:14<00:00, 1.67it/s]  
train Loss: 0.1141 Acc: 0.9759  
100%|██████████| 11/11 [00:05<00:00, 1.87it/s]  
val Loss: 1.9721 Acc: 0.5449  
Training complete in 6m 48s  
Best val Acc: 0.557078  
<Figure size 640x480 with 0 Axes>  
<Figure size 640x480 with 0 Axes>  
<Figure size 1800x900 with 0 Axes>
```

- 使用 **transforms.RandomCrop(size=(224,224), padding=64, padding_mode='edge')**

- 隨機裁剪圖像
- 會將圖像隨機裁剪為指定的大小 (224, 224)，並根據 padding 參數在圖像周圍添加填充
- padding 的值為 64，表示在每個邊界添加 64 個像素的填充
- padding_mode 參數則指定了填充的模式，這裡使用 'edge' 模式，表示在邊界填充像素的值將使用邊緣像素的值。
- Best accuracy: 0.559

```

-----
100%|██████████| 24/24 [00:14<00:00, 1.63it/s]
train Loss: 0.6763 Acc: 0.7965
100%|██████████| 11/11 [00:05<00:00, 2.19it/s]
val Loss: 1.7287 Acc: 0.5556
Training complete in 6m 52s
Best val Acc: 0.558600
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 1800x900 with 0 Axes>

```

• 使用 `transforms.RandomHorizontalFlip()`

- 隨機地將圖像水平翻轉
- 通過隨機地將圖像水平翻轉，可以在不改變圖像內容的情況下，產生更多不同的訓練樣本

◦ **Best accuracy: 0.574**

```

100%|██████████| 24/24 [00:13<00:00, 1.84it/s]
train Loss: 0.1656 Acc: 0.9615
100%|██████████| 11/11 [00:04<00:00, 2.36it/s]
val Loss: 1.9613 Acc: 0.5723
Training complete in 11m 34s
Best val Acc: 0.573820
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 1800x900 with 0 Axes>

```

2. Strong Augmentation

ORIGINAL:

• 只使用原資料集：

◦ **Best accuracy: 0.554**

```

-----
100%|██████████| 24/24 [00:13<00:00, 1.81it/s]
train Loss: 0.1224 Acc: 0.9811
100%|██████████| 11/11 [00:06<00:00, 1.82it/s]
val Loss: 1.9921 Acc: 0.5510
Training complete in 6m 29s
Best val Acc: 0.554033

```

TEST1:

```

1 | transforms.RandomHorizontalFlip(),
2 | transforms.RandomAdjustSharpness(sharpness_factor=2.5), # 銳化
3 | transforms.RandomCrop(size=(224, 224), padding=64, padding_mode='edge'), # 裁
4 | transforms.RandomRotation(50), # 旋轉

```

1. `transforms.RandomHorizontalFlip()`：隨機水平翻轉圖像。
2. `transforms.RandomAdjustSharpness(sharpness_factor=2.5)`：對圖像進行銳化操作，增加圖像的清晰度。
3. `transforms.RandomCrop(size=(224, 224), padding=64, padding_mode='edge')`：隨機裁剪圖像到指定的大小 (224, 224)，並使用 “edge” 填充模式進行填充。

4. `transforms.RandomRotation(50)`：隨機旋轉圖像，旋轉角度範圍為 -50 度到 50 度之間。

Best accuracy: 0.557

```
-----
100%|██████████| 24/24 [00:15<00:00, 1.53it/s]
train Loss: 1.0150 Acc: 0.6960
100%|██████████| 11/11 [00:06<00:00, 1.71it/s]
val Loss: 1.7700 Acc: 0.5540
Training complete in 7m 13s
Best val Acc: 0.557078
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 1800x900 with 0 Axes>
```

3. 比較一、二題的結果

從實驗結果可以得出以下結論：

1. Weak Augmentation 的效果：

- 只使用原始資料集時的準確度為 0.554
- 使用 `transforms.RandomAdjustSharpness` 時的最佳準確度為 0.557，稍微提升了一點
- 使用 `transforms.RandomCrop` 時的最佳準確度為 0.559，進一步提升了一些
- 使用 `transforms.RandomHorizontalFlip` 時的最佳準確度為 0.574，顯著提升了準確度

2. Strong Augmentation 的效果：

- 只使用原始資料集時的準確度為 0.554
- 使用 `transforms.RandomHorizontalFlip`、`transforms.RandomAdjustSharpness`、`transforms.RandomCrop` 和 `transforms.RandomRotation` 組合的最佳準確度為 0.557
- 與 Weak Augmentation 相比沒有顯著的提升

綜合來看，在這個實驗中，Strong Augmentation 並沒有明顯優於 Weak Augmentation。雖然 Strong Augmentation 使用了更多的圖像轉換方法，但在這個實驗中，這些額外的轉換方法並沒有帶來顯著的增益。相反，只使用

`transforms.RandomHorizontalFlip` 的 Weak Augmentation 在準確度上取得了更好的結果。

推測原因：

- 因為在 strong augmentation 的實驗中，使用到隨機裁切，可能切到的地方不是所要的蘭花位置，造成判斷的準確率下降

- 增強過程中引入了過度複雜性或噪聲，使模型難以從增強後的圖像中學習到有用的特徵
- 資料集本身已具有足夠的變化和多樣性，進一步的增強並不能提供額外的好處