

資料科學概論 Final Project

生物機電工程學系 大四 林泓佑 賴紫葳

(一)目標

辨識不同作物，資料集包含 10 種不同作物和 1 個土地，每種作物有 300 張圖片 (224 x 224)，共有 3300 張照片。重點方法：訓練和測試資料集的適切分割、利用 PyTorch 建立訓練迴圈、confusion Matrix 的詳細分析、Model 全面評估。

作物類別：

Atemoya



Banana



Cabbage



Carrot



Grapes



Guava



Mango



Papaya



Pineapple



Pumpkin



Bare Land



(二)資料處理

1. 確保圖形輸入尺寸一致

```
data_transform = transforms.Compose([transforms.Resize((256, 256)),
```

2. 標準化

```
transforms.ToTensor(),  
transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]))
```

3. Train、valid、test 資料分割比例

```
train_ratio = 0.7  
valid_ratio = 0.2  
test_ratio = 0.1
```

4. 設定 random seed，確保資料分類的隨機性

```
np.random.seed(24)  
np.random.shuffle(indices)
```

(三)模型參數設定

Information of Train Loader :
batch size of each epoch : 32
number of batches : 73
dataset size : 2310

Optimizer : Adam

Criterion : Cross Entropy Loss

learning rate = 0.0003

decays = 0.0001

(四)模型建置

```
class Classifier(nn.Module):  
    def __init__(self):  
        super().__init__()  
        self.cnn_layers = nn.Sequential(  
            # Input shape[3, 256, 256]  
            # CNN layers  
            nn.Conv2d(3, 64, 3, padding=1), # output shape(64, 256, 256)  
            nn.BatchNorm2d(64),  
            nn.ReLU(),  
            nn.MaxPool2d(2, 2), # output shape(64, 128, 128)  
  
            nn.Conv2d(64, 128, 3, padding=1), # output shape(128, 128, 128)  
            nn.BatchNorm2d(128),  
            nn.ReLU(),  
            nn.MaxPool2d(2, 2), # output shape(128, 64, 64)
```

```

nn.Conv2d(128, 256, 3, padding=1), # output shape(256, 64, 64)
nn.BatchNorm2d(256),
nn.ReLU(),
nn.MaxPool2d(2, 2), # output shape(256, 32, 32)

nn.Conv2d(256, 256, 3, padding=1), # output shape(256, 32, 32)
nn.BatchNorm2d(256),
nn.ReLU(),
nn.MaxPool2d(2, 2), # output shape(256, 16, 16)

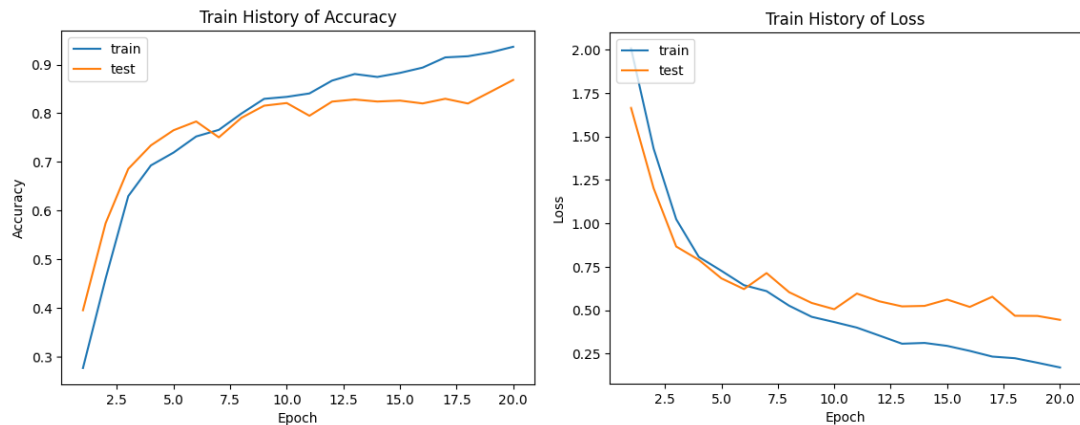
nn.Conv2d(256, 512, 3, padding=1), # output shape(512, 16, 16)
nn.BatchNorm2d(512),
nn.ReLU(),
nn.MaxPool2d(2, 2), # output shape(512, 8, 8)
)
self.fc_layers = nn.Sequential(
# Fully connected layer
nn.Linear(512 * 8 * 8, 256),
nn.Dropout(0.4),
nn.ReLU(),
nn.Linear(256, 128),
nn.ReLU(),
nn.Linear(128, 64),
nn.ReLU(),
nn.Linear(64, 32),
nn.ReLU(),
nn.Linear(32, 11),
)

```

模型特點：

- 進行 5 次卷積層與 5 次池化層
- 使用 ReLU 作為 activation function
- Batch Normalization：穩定模型的訓練過程，提高模型對不同照片的適應性。
- 為了減少過擬合的風險，我們在其中的一些全連接層中引入了 Dropout 操作。

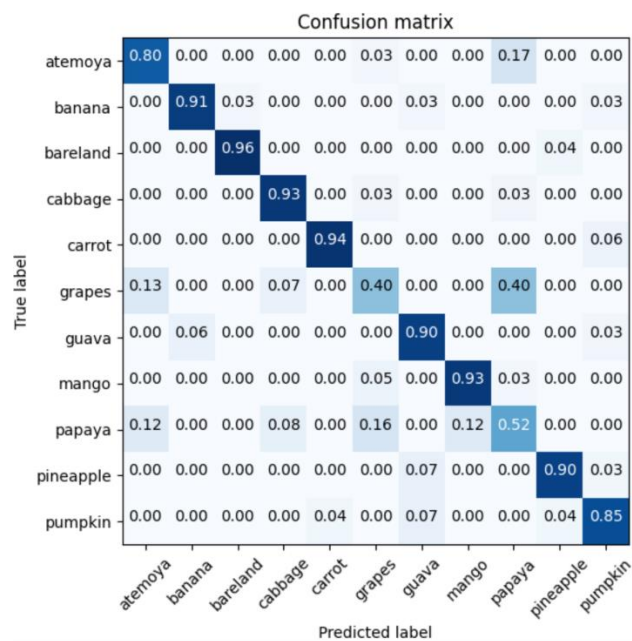
(五)結果



Test Loss: 0.067

Test Accuracy of atemoya : 80.00% (28/35)
 Test Accuracy of banana : 91.18% (31/34)
 Test Accuracy of bareland : 96.43% (27/28)
 Test Accuracy of cabbage : 93.33% (28/30)
 Test Accuracy of carrot : 94.44% (34/36)
 Test Accuracy of grapes : 40.00% (6/15)
 Test Accuracy of guava : 90.32% (28/31)
 Test Accuracy of mango : 92.50% (37/40)
 Test Accuracy of papaya : 52.00% (13/25)
 Test Accuracy of pineapple : 89.66% (26/29)
 Test Accuracy of pumpkin : 85.19% (23/27)

Test Accuracy (Overall): 85.15% (281/330)



可以看到模型在分辨 grapes 和 papaya 的準確度特別低。

(六)過程中嘗試提高準確度的方法

(使 Accuracy 從 69%上升到 85%)

1. 每次訓練的 batch size 增大 (25→32): 較大的 batch size 提高了模型的訓練速度, 並使模型更好地利用硬體資源 (如 GPU)。此外, 更大的 batch size 有助於提高模型的泛化能力, 因為模型在每批次的訓練中看到的樣本更具代表性。
2. 模型內的卷積層&池化層數量增加 (3 層→5 層): 較深的模型能夠捕捉到更複雜的特徵和模式, 進而提高模型的表徵能力。
3. 模型內 dropout 的數量增加 (0.2→0.4): 有助於減少過度擬合。
4. Learning rate 減小 (0.001→0.0003): 較小的 learning rate 使得模型在訓練過程中更加穩定, 有助於收斂到更好的解。
5. 訓練 epoch 增加 (10 次→20 次): 增加訓練 epoch 可以使模型更充分地學習數據的特徵, 進而提高模型的性能。
6. 增加 weight decay: 有助於防止模型的權重過大, 降低過度擬合的風險。
7. 增加 scheduler: 使用學習率 scheduler 可以根據訓練的進展動態調整學習率。這有助於更有效地訓練模型, 使得模型更容易收斂到最優解。

(七)結果分析與討論

- Grapes 和 Papaya 誤差較大: 如果 Grapes 和 Papaya 在特徵表示上可能與其他作物相比較難區分, 模型可能難以從圖片中提取區分性特徵, 導致辨識的難度增加。我們在嘗試調整參數的過程中, 發現這兩者的準確度始終都是最低的, 此外, 我們從 Confusion Matrix 裡面發現 Grapes 或 Papaya 較容易被誤認成其他作物, 但是其他作物認成 grape 和 papaya 的機率並不高, 可以推測這兩者本身的特徵不太好辨認, 或許這兩個作物需要比其他作物有更高的模型複雜度才能準確辨識, 但是模型複雜度無法在同一個模型中做調整。此外, 可以發現它時常把 Grapes 和 Papaya 誤認成彼此, 推測這兩者對於模型來說可能很相似, 導致大大拉低了整體的精準度。未來期望了解哪些特徵對於 Grapes 和 Papaya 的區分起著關鍵作用, 這樣就可以了解此模型在某個特徵區辨的侷限性。

- 過程中嘗試的一些方法，但是對準確度沒甚麼影響(和原本差不多)，例如將 Optimizer 改為 SGD：optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum=0.95)。推測若要使用此 Optimizer 獲取更好的準確率則模型其他部分也要調整。
- 可以看出 Train 和 Test 之間的 Accuracy 和 Loss 差距不大，反映了模型在應對未見過的數據時的良好泛化能力，對於辨識大部分作物都是有效的，這也表示出我們在模型複雜度選擇、資料處理、模型參數調整都是合適的。