HW02 Classification

1. 前處理
   * Train/Valid [改變的機率值為0.1]

|  |  |
| --- | --- |
| 尺寸調整 | Resize((224, 224)) |
| 將圖像轉為 Tensor | ToTensor() |
| 空間操作 | RandomHorizontalFlip() |
| RandomVerticalFlip() |
| RandomRotation(10) |
| RandomPerspective(distortion\_scale=0.6, p=1.0) |
| RandomAffine(degrees=(30, 70), translate=(0.1, 0.3), scale=(0.5, 0.75)) |
| ElasticTransform(alpha=250.0) |
| 顏色操作 | ColorJitter(brightness=0.4, contrast=0.4, saturation=0.4, hue=0.1) |
| RandomGrayscale(p=0.1) |
| RandomInvert(p=0.1) |
| RandomPosterize(bits=2, p=0.1) |
| RandomSolarize(threshold=1.0) |
| 噪聲操作 | AddGaussianNoise(mean=0.0, std=0.05) |
| AddPoissonNoise(lam=0.1) |
| AddSpeckleNoise(noise\_level=0.1) |
| AddSaltPepperNoise(salt\_prob=0.05, pepper\_prob=0.05) |
| 高斯模糊 | GaussianBlur(kernel\_size=(5, 9), sigma=(0.1, 5.)) |
| 標準化 Normalize [-1,1] | Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) |

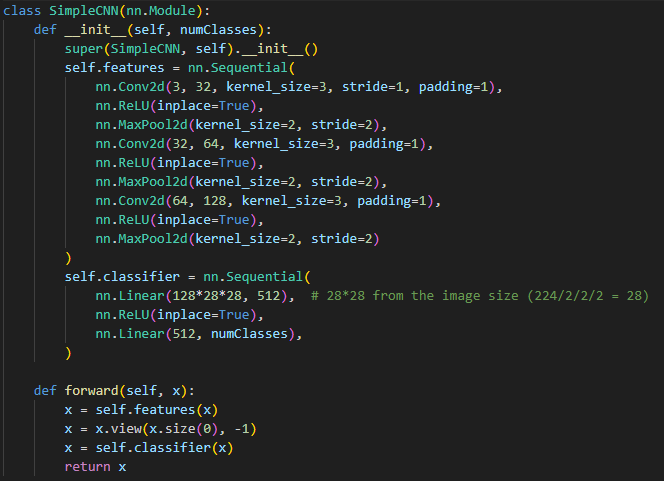
* Test

|  |  |
| --- | --- |
| 尺寸調整 | Resize((224, 224)) |
| 將圖像轉為 Tensor | ToTensor() |
| 標準化 Normalize [-1,1] | Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) |

補充: 由於測試圖片加入了翻轉變形、改變顏色跟雜訊干擾等等變化，以此在前處理時訓練及驗證圖片也要加入這些圖片變化操作，確保後續訓練能獲的更好的模型效果

1. 訓練模型選擇

* 先前: 自己寫的CNN模型
  + 使用自己寫的訓練模型，效果很差，精準度在0.02~0.03徘徊，比baseline要求的0.05還要低，距離最終目標還有很多路要走。

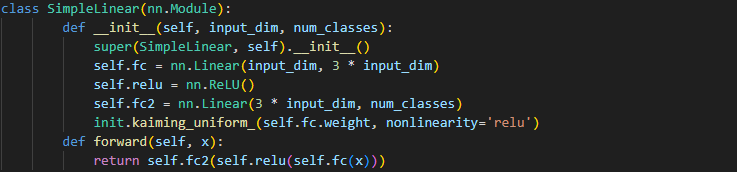


* 最終: 套用CNN pre-trained模型 (特徵提取)
  + 使用老師提供的[TRANSFER LEARNING FOR COMPUTER VISION TUTORIAL](https://pytorch.org/vision/stable/models.html)，裡面有一個部分關於pre-trained image classification models，有許多pre-trained可供使用，我使用幾個CP值較高的模型，以及舉例上有使用的 “resnet50” 以下表個的模型作為訓練模型使用。

|  |  |
| --- | --- |
| Pre-trained model | |
| convnext\_base | swin\_v2\_b |
| efficientnet\_v2\_m | maxvit\_t |
| resnet50 |  |

* 做完Pre-trained model後，再套一層自己的**linear model**，將 CNN 模型的**特徵映射到最後的分類空間**，題目設計是要辨識出50隻角色，所以這裡的”分類空間=50”

|  |
| --- |
| Pre-trained model |
| meta\_simpleLinear |



* 訓練與驗證的損失率 Loss

|  |  |
| --- | --- |
|  |  |
| convnext\_base | swin\_v2\_b |
|  |  |
| efficientnet\_v2\_m | maxvit\_t |
|  |  |
| resnet50 | meta\_simpleLinear |

1. 環境變數

* .env:

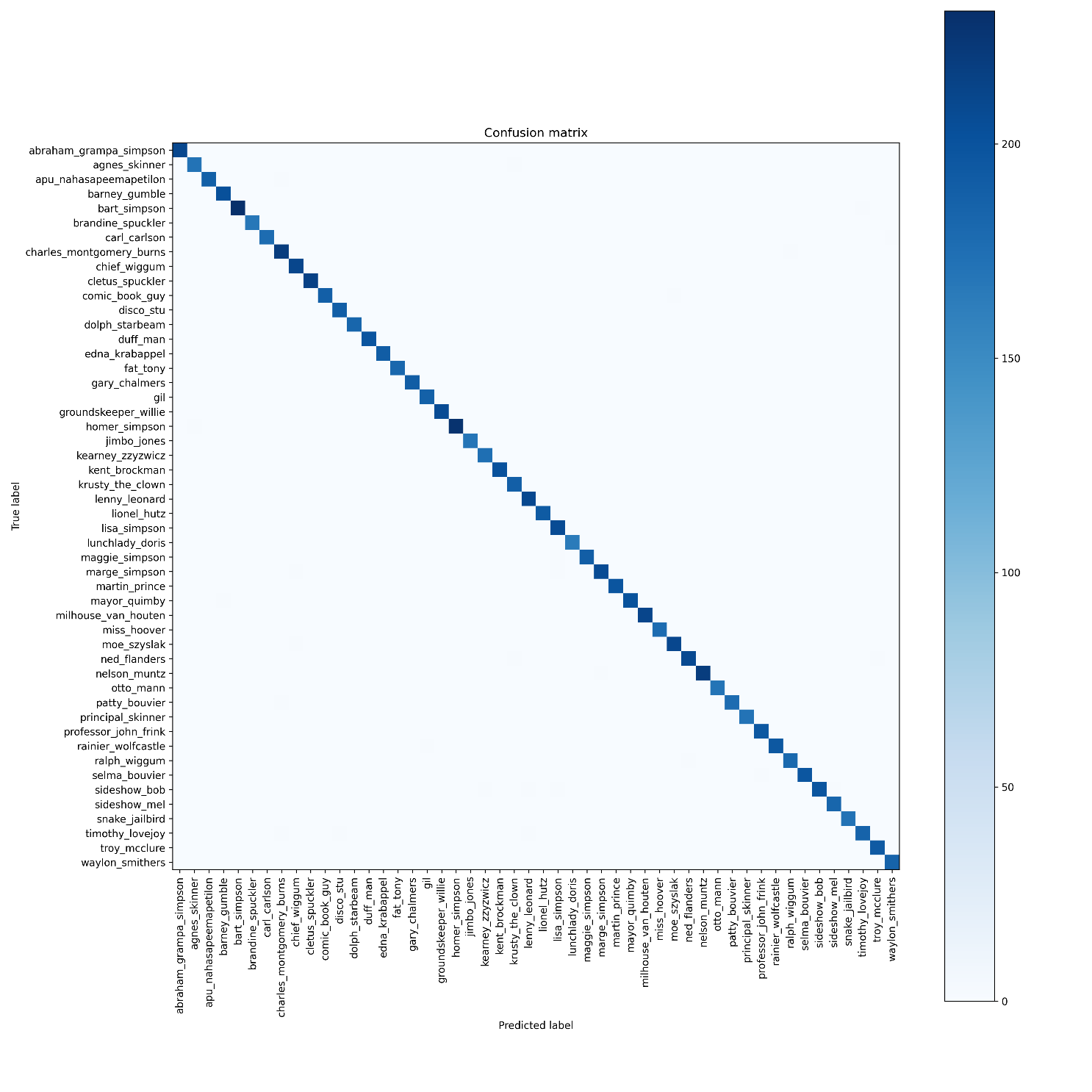
第一次這樣寫，好處是將模型名稱，資料夾路徑等等，定義再一起，方便管理，想換成其他pre-tarined可以直接從這裡進行修改，不用每個檔案有用的的地方都改。

* 使用方法:

下載: pip install python-dotenv



1. 混淆矩陣(Confusion Matrix)



1. 每層權重(Layer filter)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| …… (省略) …… ….. (省略) …… ….. (省略) …… ….. (省略) …… ….. (省略) ……  …… (省略) …… ….. (省略) …… ….. (省略) …… ….. (省略) …… | | | |
|  |  |  |  |