

Give reasons to adopt the specific deep learning model. (10%)

原本我選擇自行設計 CNN 模型，架構如下圖，包含前面的卷積層、激活函數、池化層與為了避免 overfitting 而增加的 Dropout，以及後面拉平所有神經元後好幾層全連接層。

```
# CNN model
model = Sequential()

model.add(Conv2D(16, (3,3), input_shape=(150, 150, 3)))
model.add(BatchNormalization(axis=3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(Dropout(0.5))

model.add(Conv2D(32, (3,3)))
model.add(BatchNormalization(axis=3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(Dropout(0.5))

model.add(Conv2D(64, (3,3)))
model.add(BatchNormalization(axis=3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))
model.add(Dense(512, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(120, activation='softmax'))
```

然而在經過多次調整，像是增加卷積層或全連接層、調整訓練的 epoch 等方式，在 kaggle 上的 loss 始終在 4.多，最後我開始嘗試使用其他經典的 CNN 模型架構。

[keras application](#) 有提供數種經典的 CNN 模型架構可以直接使用，包含 VGG16、ResNet-50、InceptionResNet V2 等，同時這個網站還有列出每種模型的大小、Top-5 Accuracy、參數數量、CPU執行時間等資訊可以參考。

由於我的電腦運算能力普通，所以需要在效能與速度間做取捨，最終選擇 InceptionV3 這個模型，因為相對來說這個模型的不大，運算所需的時間較少，而且正確率也沒有低太多。

Please list and describe your adopted methods for image preprocessing. (25%)

- 調整圖片大小：由於資料集中的圖片大小不一，統一調整為 (150, 150, 3)，一方面這樣才能作為模型的輸入，另一方面也可以減少後續模型所需的計算量。藉由 cv2 預設的 bilinear interpolation 方法進行圖片縮放。
- 數值正規化：將代表圖片的數字從 0~255 縮放到 0~1 之間。不過最後沒有使用這個方法，因為浮點數所需的空間比整數大，會導致 MemoryError。

Please describe the definitions and meaning of selected model parameters. (25%)

InceptionV3 的參數有：

- include_top: 是否包含最後的全連接層
- weights: 是否載入預訓練權重
- input_tensor: 把 Keras tensor 作為輸入層
- input_shape: 輸入圖片的維度
- pooling: 要不要 pooling，以及選擇 avg 或 max

- `classes`: 有幾種預測類別
- `classifier_activation`: 輸出層選用的激活函數

我採用預訓練權重，基於預訓練過的模型用在現有資料集上可以減少訓練時間。並且調整模型的輸入大小。`include_top = False` 是因為有自定義輸入的大小。

模型使用的架構有：

- `convolution layer`: `Conv2D`; 提取特徵，因為是二維圖片所以是 2D
- `pooling layer`: `max pooling`; 選出範圍內最大的數值
- `batch normalization`: 對當前數值做正規化
- `Activation function`: `hidden layer` 是 `relu`, `output layer` 是 `softmax`

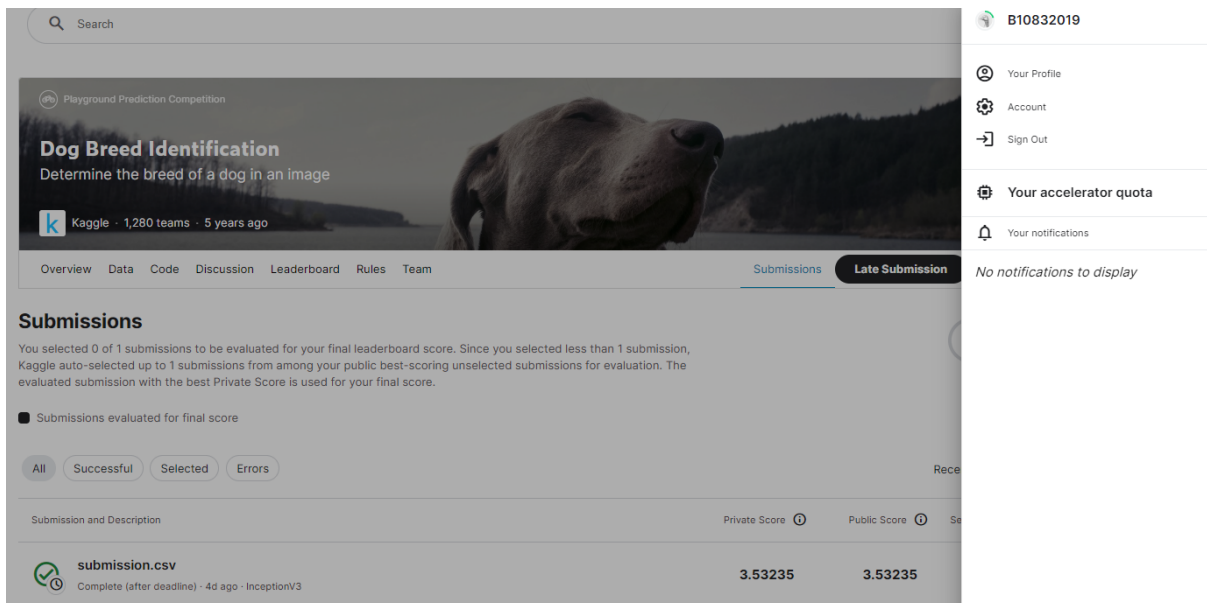
2 screenshots

a. Print out Accuracy and Loss of training results. (10%)

```
model.evaluate(X_train, Y_train)
```

```
224/224 [=====] - 358s 2s/step - loss: 0.8972 - accuracy: 0.7346
[0.8972253203392029, 0.7345911860466003]
```

b. Final submission score on Kaggle with your account name. (10%)



The screenshot shows the Kaggle Playground Prediction Competition page for 'Dog Breed Identification'. The page includes a search bar, a sidebar with user profile information (B10832019), and a main content area with tabs for Overview, Data, Code, Discussion, Leaderboard, Rules, and Team. The Submissions tab is selected, showing a list of submissions. The user's submission, 'submission.csv', is highlighted with a Private Score of 3.53235 and a Public Score of 3.53235. The page also includes a sidebar with user profile information and a search bar.