玉山人工智慧

公開挑戰賽

隊伍：我們的興趣好分散

成員：廖家鴻，游智鈞，俞曉璟，

彭淮南，呂政彥，高承翰

★摘要

在本次比賽，我們主要使用 EfficientNet 來擷取特徵與分類。雖然無法得知具體使用的特徵為何，但藉由觀察與實驗，我們猜測 EfficientNet 使用的特徵為整張影像邊特徵的組合。在權衡訓練時間與模型成效後，我們選用 EfficientNet-B3 做為主要的網路架構。同樣地，為了減少訓練時間，我們將輸入影像的大小設置為 224×224。在實驗過程中，我們觀察到模型存在過擬合的情況，究其原因，是由於訓練資料與測試資料的差異性。我們使用了兩種方法解決差異性的問題，其一為直接增加資料集，其二為使用資料擴增（Data Augmentation）的方法來產生略為不同的影像。使用這兩個方法之後，我們有效的控制過擬合的問題。此外，我們發現 EfficientNet 分類 isnull 類別尤其困難。我們使用兩種方法進行解決，其一是細分 isnull 類別，其二為設定合適的閾值（threshold），用以區分影像是否為 isnull 類別。在 isnull 類別，模型的性能有顯著的提升。

★環境

我們使用的系統平台及程式語言，如下列的表格所示：

Operating System	Ubuntu 20.04
Programming Language	Python 3.6.10

我們使用的所有 Python 函式庫，如下列的表格所示：

Package	**Version**
torch	1.9.0 + cu111
torchvision	0.10.0 + cu111
torchaudio	0.9.0
efficientnet-pytorch	0.7.1
tqdm	4.61.1
scikit-learn	0.24.2
pandas	1.1.5

★特徵

主要由 EfficientNet 來尋找特徵，無法得知確切的特徵。雖然如此，但藉由實驗不同資料擴增方法與增加資料集後，可以稍微了解到 EfficientNet 所關注的特徵為何。

1. 由資料前處理觀察 EfficientNet 所關注的特徵

原以為與其他分類任務相同，隨機裁切對部分影像中不完整的中文字辨識有幫助，但實驗結果顯示，裁切對於整體成效不利。推測影像進行裁切後，可能造成裁切前的中文字與裁切後的中文字變成不同的兩個字，就連人眼都無法正確判斷。這種特性讓我們得知 EfficientNet 擷取之特徵為一整張影像中所有資訊，而非部分資訊。

與其他分類任務相同，在訓練時，對訓練影像做小幅度的旋轉、些微的平移及小尺度的拉伸能些微提升模型的精準度。這個現象符合中文字的特性，無論是旋轉、平移或是拉伸，只要不是太大幅度的改變影像，還是可以將辨識出影像中的中文字。因為人眼在辨識中文字所使用的特徵為邊（edge），故我們推測 EfficientNet 所找的特徵應為邊特徵。

訓練時，透過將影像轉成黑白影像，人工去掉一些雜訊。使用下列步驟去除雜訊：

i. 調整影像顏色與亮度

ii. 影像銳化

iii. 去除影像中紅色雜訊，例如印章、格線

iv. 將彩色影像轉成黑白影像

v. 影像降噪

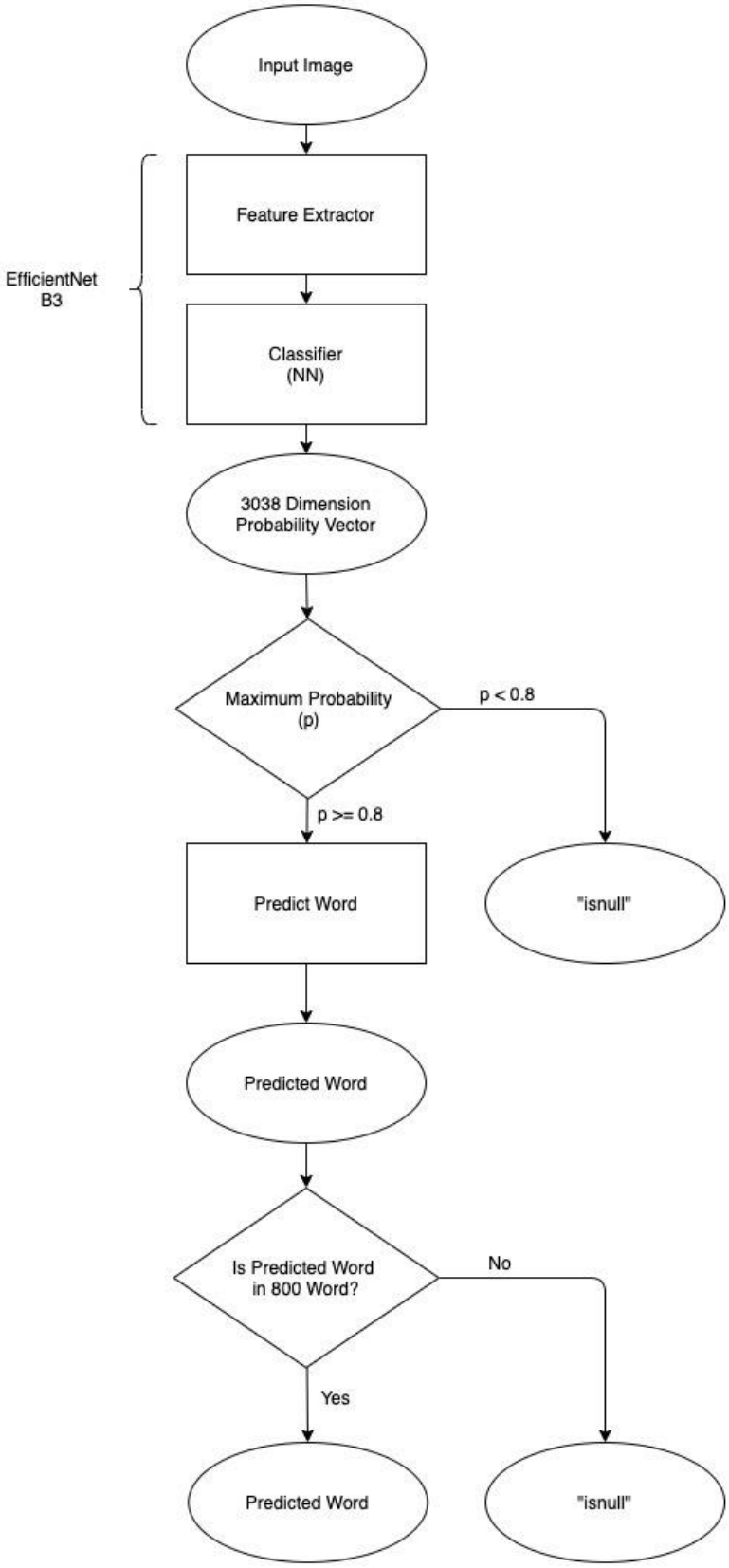
實驗結果顯示，原影像與人工進行前處理的影像再經過神經網路的分類判斷後，得到差不多的成效。故我們推測 EfficientNet 可以自行去除一些顏色的雜訊，且顏色並不是 EfficientNet 主要的特徵。

2. 由增加資料集觀察 EfficientNet 所關注的特徵

除了官方給的 800 字影像外，我們也從 Traditional Chinese Handwriting Dataset（此資料集為開源資料集，附在文末的參考資料）挑選一些日常生活比較常用的字，還使用 Python 開源套件 Pillow 的 ImageDraw 函式生成一些不同字體的印刷字，加入資料集進行訓練。測試資料集的 macro f1 score 從原本的 0.9211 提升到 0.9313。雖然只有微幅的進步，但由此推論，給予模型額外的資料，即使這些資料，從肉眼看起來不太一樣（新增的影像顏色只有黑色和白色，而且沒有什麼雜訊），但模型還是能透過這些資料找到一些不錯的特徵來達成任務。進而推論，模型找到的特徵應與中文字的邊有關，與顏色較沒有關係。

無論是在資料前處理或是在增加資料集時，我們都可以觀察到相同的現象：EfficientNet 找到的特徵為邊特徵，甚至是整張影像的所有邊特徵的組合。顏色特徵不在我們的考量。

★訓練模型


Input Image
Feature Extractor
EfficientNet B3
Classifier (NN)
3038 Dimension Probability Vector
Maximum Probability (p)
p < 0.8
p >= 0.8
"isnull"
Predict Word
Predicted Word
Is Predicted Word in 800 Word?
No
Yes
Predicted Word
"isnull"

Hyperparameter:

- Feature Extraction and Classifier
  - Model: EfficientNet B3
  - Input Image Size: 224x224
  - Number of Classes: 3038
  - Batch Size: 32
  - Number of Epoch: 50
  - Train Data Augmentation:
    - RandomAffine
    - Resize: 224 x 224
    - Normalize
      - Red Channel Mean: 0.726
      - Green Channel Mean: 0.686
      - Blue Channel Mean: 0.695
      - Red Channel Standard: 0.205
      - Green Channel Standard: 0.210
      - Blue Channel Standard: 0.183
  - Validation Data Augmentation:
    - Resize: 224 x 224
    - Normalize
      - Red Channel Mean: 0.726
      - Green Channel Mean: 0.686
      - Blue Channel Mean: 0.695
      - Red Channel Standard: 0.205
      - Green Channel Standard: 0.210
      - Blue Channel Standard: 0.183
  - Loss Function: Categorical Cross Entropy
  - Optimizer: Adam
  - Initial Learning Rate: 0.001
  - Learning rate scheduler: Cosine Annealing Schedule with Warm Restarts
  - Number of iterations for the first restart(T_0): 1
  - Factor increases after a restart(T_mult): 2
- Post Processing
  - Threshold(區分是否為 isnull 之閾值): 0.8

★訓練方式及原始碼

針對這個任務，我們實驗非常多不同的超參數，最後總結一些重要超參數，調整這些超參數能有效的提升模型的效果。

EfficientNet 的論文中提到，EfficientNet 從 B0 到 B7 共有 8 種不同的版本，數字越大，代表整個網路架構越大。雖然在調整好所有超參數的情況下，較大的神經網路通常有著較佳的分類能力，但訓練一個龐大的網路是十分耗時的，而且超參數的調整直接影響了模型效果的好壞。鑑於我們的運算資源有限，我們必須在模型的效果與訓練時間做權衡，在做了一系列實驗下，發現 EfficientNet-B3 的訓練時間尚可接受，且效果不差。故最終選擇 EfficientNet-B3 做為我們的主要的網路架構。

除了網路架構外，輸入影像之大小也同樣會影響模型的泛化能力與訓練時間。雖然 EfficientNet 的作者在論文中建議，在訓練 EfficientNet-B3 時，輸入影像尺寸最好為 300 x 300，但為了減少每次調整超參數所需要的訓練時間，我們將輸入影像尺寸改為 224 x 224。

在實驗中，我們觀察到模型的泛化能力差，再做了一些分析後發現，模型會有過擬合的情況，而且發現其原因來自於訓練資料與測試資料的差異性。為了減少訓練資料與測試資料之間的差異性，

我們應用了兩個方法來解決這個問題。其一，直接增加資料集。我們從網路上獲取一些開源的資料集，雖然與官方的資料集略有差距，但因中文手寫字的資料集取得不易，故還是先拿來使用。另外，在官方提供的資料集中，我們發現有些影像上是沒有任何手寫字。我們就在這些影像上印上一些不同字體的印刷字做成資料集。其二，使用資料擴增（Data Augmentation）的方法來產生一些略為不同的影像，這些方法中，以隨機小角度的旋轉、隨機位移效果最為顯著。

我們在後來的實驗中發現 EfficientNet 分類不在 800 字的類別（isnull 類別）時，效果非常糟糕。我們認為有兩個原因造成分類 isnull 類別困難。其一，isnull 的字與其他相比，特徵不明顯且多元；其二，EfficientNet 很難透過分類 801 個類別（即 800 字與 isnull），就得知「只要判定不為官方提供的 800 字，就分類為 isnull」。我們透過下列兩個技巧來解決這個問題。

有鑒於 isnull 類別的影像數太過龐大，造成 isnull 類別的特徵模糊，我們將 isnull 類別加以細分，只要超過某一定影像數量，我們就將該字從 isnull 類別獨立出來，成為一個新的類別。如此一來，每張從 isnull 類別獨立出來的新類別的特徵會比較明顯。等到 EfficientNet 分類完畢，再把原本屬於 isnull 類別的字歸類為

isnull。我們從 isnull 類別，額外獨立出 2237 個字，也就是 EfficientNet 的需要分類類別有 3038 類（官方 800 字、從 isnull 獨立出來的字以及 isnull）。

雖然用了上述的方法能夠解決一部份的 isnull，但若輸入的影像不是 3037 字之一，則 EfficientNet 就很容易判斷錯誤，導致整個模型的成效受到影響。我們猜測當輸入影像的字不是 3037 字的其中一個時，EfficientNet 最後輸出的機率向量之機率值應該都會偏低，所以我們對 EfficientNet 輸出的機率向量進行分析。我們取出機率向量中的最大值，稱之為最大機率值，發現一個現象：當 EfficientNet 判斷正確時，最大機率值通常會非常趨近於 1；當 EfficientNet 判斷錯誤或是類別為 isnull 時，最大機率值通常會偏低。我們採用的方法為設定閾值（threshold），來區分是否為 isnull。若大於等於閾值就維持 EfficientNet 的判定；若小於閾值就判定為 isnull。經過反覆實驗，我們最後決定將閾值設為 0.8。

GitHub：
https://github.com/strupfrmnth/2021-ESUN-AI-Competition

★結論

- 經過許多影像分類論文的調查和比較，我們認為 EfficientNet 為最適合用來辨識中文手寫。其中 EfficientNet B3 在模型訓練時間及預測表現上最符合我們的需求。
- EfficientNet 關注的特徵為邊特徵，因此影像顏色上的處理不會影響預測結果。由於中文字的特性，過度裁切可能會使得原有影像中的文字變成其他的字，進而造成誤判。
- 擴充資料集對於提升準確度有顯著的效果。對於原始 800 類而言，藉由隨機小角度的旋轉、隨機位移產生新的資料，可以讓模型在訓練時更加明確地辨認特徵；對於不屬於原始 800 類的文字而言，讓模型額外學習更多種類的文字在「判斷是否屬於原有 800 類」的任務中，相較於只訓練 801 類有更好的成效。
- 藉由設定最大機率值的閾值可以排除模凌兩可的答案，提升整體的準確度。

★參考資料

- Traditional Chinese Handwriting Dataset

The dataset is AI. FREE Team development from [STUST EECS_Chinese MNIST(總集)]. If used, modified, or shared, please cite the source and the message.
(Source: https://github.com/AI-FREE-Team/Traditional-Chinese-Handwriting-Dataset)

- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks