AIMAS Homework 2

F94081076 郭立晨 資訊 113

題目: 利用資料集訓練模型區別健康與不健康的分類問題

實作過程:

1. 使用 Keras、sklearn、pandas 等套件作為工具。
2. 讀入 Data_Entry_2017.csv 檔案並利用 dataframe 儲存，內容包含病患的去識別化資料、疾病分類等重要標記。
3. 將每張 X 光影像的路徑也新增到 dataframe 新欄位中，方便後面做資料前處理。
4. **前處理:**
   - ✓ 將原本有疾病的照片標記成 Unhealthy，若是原本標記是”No Finding”則標記成 healthy。
   - ✓ dataframe 新增欄位存放數字標記，若該圖片是 Unhealthy 則為 1；healthy 則為 0。
   - ✓ 去除不需要的欄位，以及去除部分照片來平衡健康、不健康的照片數量。

     ```
     Healthy      8753
     Unhealthy    6246
     ```

     ```
     Healthy      6253
     Unhealthy    6246
     ```
   - ✓ 分割 dataframe 文字資料集為訓練集(0.85)及測試集(0.15)
     train 10624 Validation 1875
5. 使用 Keras ImageDataGenerator 產生存放照片的「容器」，再用 flow_from_dataframe 函式，將照片與 dataframe 的文字標記組合成完整的訓練集、驗證集和測試集，作為模型的輸入。ImageDataGenerator 也可以設定參數使圖片做 Augmentation；我建立了有 Augmentation 以及沒有 Augmentation 的 ImageDataGenerator 作為對照。
6. **建立模型**，使用 Keras 已經建立好的 VGG16 模型作為基本架構，再加上全連接層、拋棄層，部分層設定為凍結層無法被訓練，只訓練高層的全連結層分類器。最後一層一個輸出使用 sigmoid 作為激活函式。
7. 為了加快訓練速度，使用 Transfer Learning 載入”imagenet”預訓練的權重；設定輸入層的維度為 (150,150,3)。
8. 使用” binary_crossentropy”作為損失函式處理離散的分類問題，”Adam” 作為 optimizer，Learning Rate=0.00001；用準確率作為模型評斷標準。(Learning rate 不能太大否則會破壞 pre-trained weights)

9. 設定 10 個 epoch，開始訓練模型。

實作結果:

Model Structure

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14714688
dense (Dense)	(None, 4, 4, 512)	262656
dropout (Dropout)	(None, 4, 4, 512)	0
dense_1 (Dense)	(None, 4, 4, 512)	262656
dropout_1 (Dropout)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense_2 (Dense)	(None, 1)	8193

Total params: 15,248,193
Trainable params: 13,512,705
Non-trainable params: 1,735,488

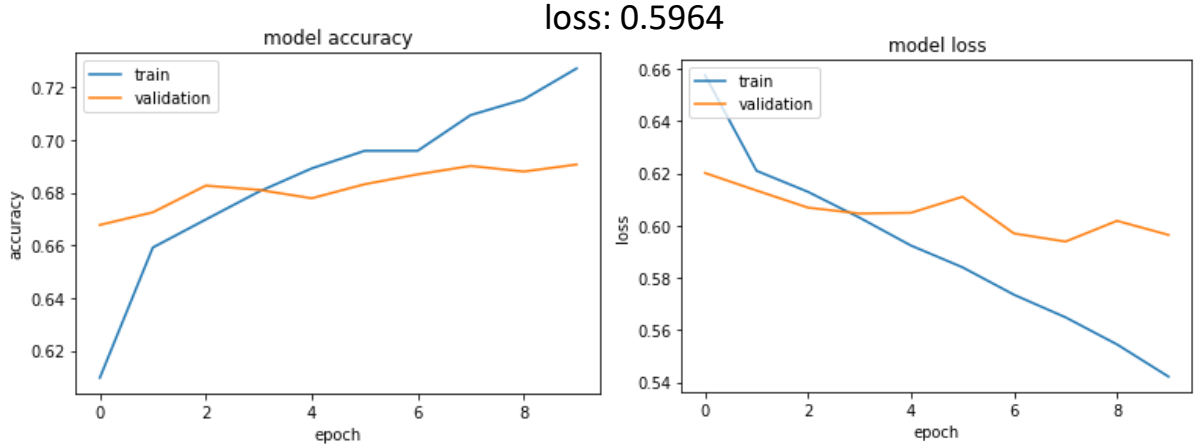
Augmentation

rescale=1./255, horizontal_flip = True, vertical_flip = False, height_shift_range= 0.05, width_shift_range=0.1, rotation_range=5, shear_range = 0.1, fill_mode = 'reflect', zoom_range=0.15

Without Augmentation

accuracy: 0.6907

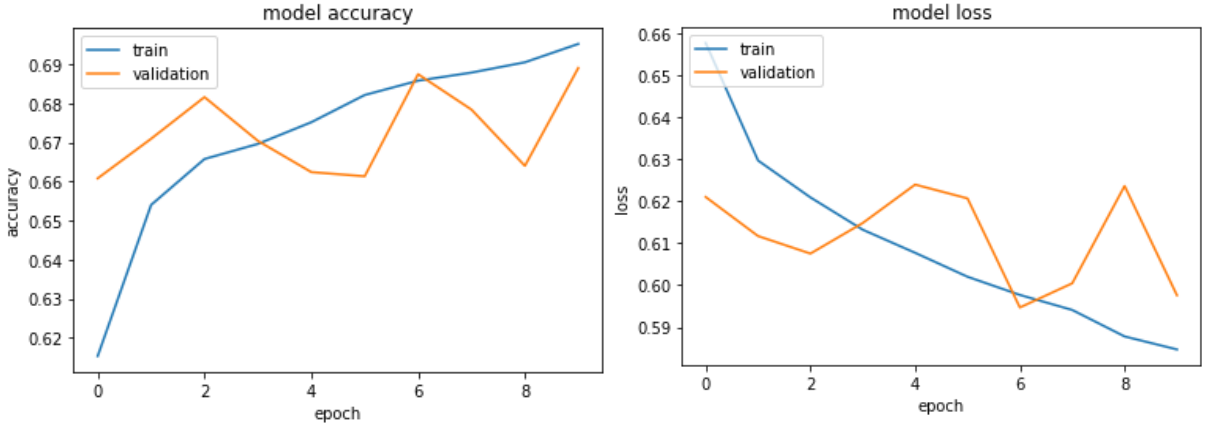
loss: 0.5964



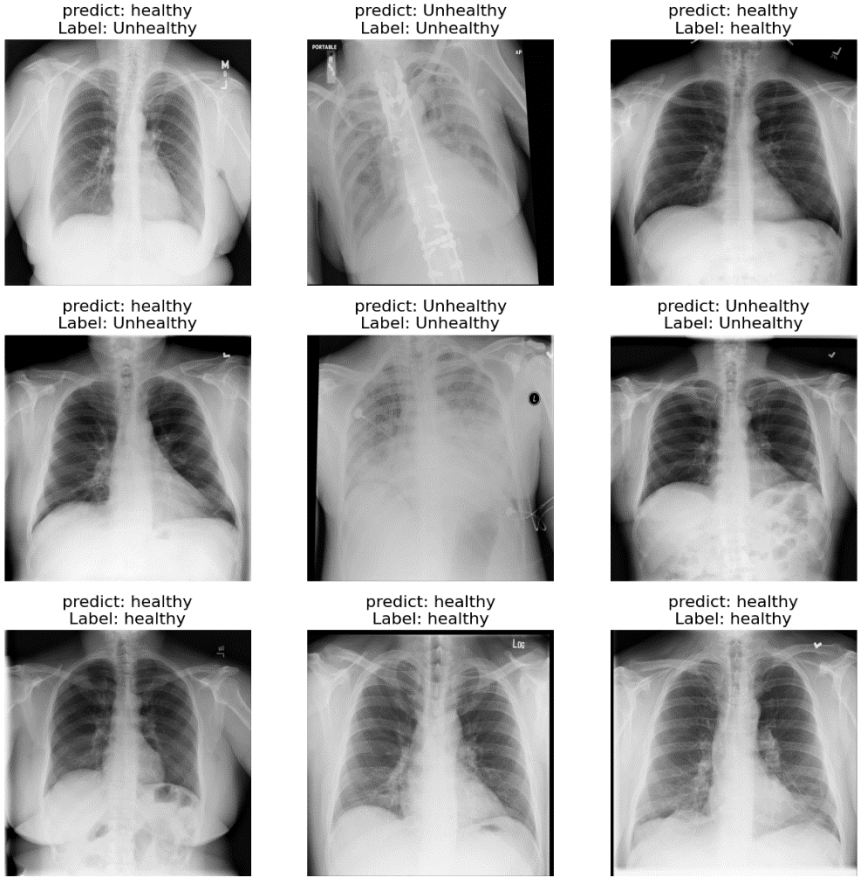
Augmentation

accuracy: 0.6837

loss: 0.5970



預測結果



有無 Augmentation 的結果差異不大因此只印出其中一種的結果

心得感想:

第一次實作 Kaggle 提供的題目才知道原來資料集都是這麼龐大，因為硬碟大小以及計算資源的限制所以只下載其中一部分來訓練。光是處理這一小部分資料集就花了我不少時間，也不能像對待其他較小的資料集一樣將照片一次全部讀入並存起來使用，不然會造成暫存空間不足的情況。因此我也看了 Kaggle 上其他網友的 Notebook，最後使用了 ImageDataGenerator 作為引入照片資料的容器，也算是多學到了一個資料前處理的工具。

在訓練的過程中，我特別嘗試了 Augmentation 是否會影響訓練的準確度，經過實驗發現好像效果並不明顯，但也有可能是我使用的 Augmentation 不適合用來處理 X 光影像，這就衍生出了另一個課題，究竟如何找出最適合資料集的 Augmentation? 希望在未來持續學習的過程裡可以學到這方面的知識。

另外，我發現在訓練的過程中準確率雖然有上升但是都上升的很緩慢，從

訓練集的準確率以及測試集的準確率來看應該是有成功的訓練，但是，可能需要更多的 epoch 以及更多張訓練圖片才能得到更好的訓練結果。而當我調整輸入層圖片的大小時，所需的運算時間也大幅提升，礙於 GPU 使用時間的限制只好維持(150,150,3)的輸入大小，我認為準確率不夠高也有可能是我將圖片調整成太小造成關鍵的特徵被犧牲。總而言之，深度學習真的是一個需要運算資源的一項工具，只要資源夠豐富，深度學習相對的一定也能有更好的成效。

模型下載連結:

https://drive.google.com/drive/folders/14CQrsqHpCwzsxOdfeFj4DBPCjyGFialo?usp=sharing