智慧化企業整合

Project 3

運用 CNN 辨識胸腔 X 光照片

110034566 黃采琳

指導教授：邱銘傳 教授

目錄

一、緒論.....1

1.1 研究背景.....1
1.2 5W1H.....1

二、資料來源與前處理.....1

2.1. 資料來源.....1
2.2 資料前處理.....2

三、模型建立與訓練.....4

3.1 Densenet 介紹.....4
3.2 模型訓練.....5

四、超參數優化.....6

五、結論.....9

5.1 研究貢獻.....9
5.2 未來展望.....9

六、參考文獻.....10

一、緒論

1.1 研究背景

全球疫情肆虐打亂了人們的生活節奏，疫情初期更是掀起人們的恐慌，只要一出現咳嗽、流鼻涕等上呼吸道感染與症狀，就大量湧向醫院要求主動篩檢，這對醫院來說是極大的負擔，為了讓大型醫院能專注治療嚴重的個案，會將篩檢任務分配給其他第二、三級區域的醫院篩檢站，放射科醫師一天約需看數百張至數千張的 X 光片，而在判讀胸腔 X 光片時，必須不斷放大縮小局部畫面，確保不會出現任何細小變化，在數量與時間的壓力下，消耗大量的眼力與腦力，難免會有疏漏，且每位醫師所判讀的結果可能也會不同。本研究希望以 CNN 建立一判讀胸腔 X 光片的模型，來協助醫師判讀 X 光片更有效率，使醫師的負擔將得以減輕，並可優先將資源放在嚴重的患者上。

1.2 5W1H

What	醫師判讀的胸腔 X 光片過多，可能會有判讀錯誤的情形
Who	臨床醫師、放射科醫師
When	欲檢測是否罹患 COVID19
Where	各大醫院
Why	協助降低醫院人力負擔，能將資源放在嚴重患者身上
How	運用深度學習方法

二、資料來源與前處理

2.1. 資料來源

資料來源是從 kaggle 網站抓取下來的 Chest X-Ray (Pneumonia,Covid-19,Tuberculosis)資料集，包含 Normal、Covid19、Pneumonia(肺炎)、Tuberculosis(結核病)四種類別：

1. COVID19 : 共有 558 張
2. Normal : 共有 1530 張
3. Pneumonia : 共有 1257 張
4. Tuberculosis : 共有 586 張

以上這幾種病徵，大多都會以胸腔 X 光片進行醫學上的判定，特別針對 COVID19 的患者，需要進行隔離的醫治，因此如何在最快的時間內確定患者的罹患的疾病，把握黃金治療期間，同時避免因等待期間過久而增加了本土傳染的機會，運用深度學習的方法，協助醫生加快工作效率，也解決在醫療資源較匱乏的區域，醫師能更加專注在治療病患，挽救更多的生命。

2.2 資料前處理

(1) 因原始資料集切分的數量不太對，因此本研究重新整理資料集後以 8:1:1 切分訓練、驗證、測試集

```
#訓練、驗證、測試: 8:1:1
#train
def moveFile(fileDir):
    pathDir = os.listdir(fileDir)        #取圖片的原始路徑
    filenumber=len(pathDir)
    rate=0.8        #自定義抽取圖片的比例，比方說100張抽10張，那就是0.1
    picknumber=int(filenumber*rate)  #按照rate比例從資料夾中取一定數量圖片
    sample = random.sample(pathDir, picknumber)    #隨機選取picknumber數量的樣本圖片
    print (sample)
    for name in sample:
        shutil.move(fileDir+name, tarDir+name)
    return
NAME = ["COVID19","NORMAL","PNEUMONIA","TURBERCULOSIS"]
name = ["covid19","normal","pneumonia","turberculosis"]
if __name__ == '__main__':
    for i in range(4):
         fileDir = "/content/drive/MyDrive/chest_x_ray/{}/" .format(NAME[i])        #源圖片資料夾路徑
         tarDir = '/content/drive/MyDrive/chest_x_ray/train/{}/'.format(name[i])        #移動到新的資料夾路徑
         moveFile(fileDir)
```

以下為資料切分的訓練集數量：

```
#各類別訓練集數量
name = ["covid19","normal","pneumonia","turberculosis"]
for i in range(4):
    train_path = "/content/drive/MyDrive/chest_x_ray/train/{}" .format(name[i])
    read = os.listdir(train_path)
    num = len(read)
    print(num)
```

```
552
1512
1200
560
```

(2) 資料擴增

由以上結果可看出各類別的數量呈現不平衡的情況，運用 Augmentor 來進行資料擴增，Augmentor 是一個 Python Package，用於幫助機器學習任務的影像增強與生成，Augmentor 可以使圖片進行透視偏斜、彈性變形、旋轉、裁剪、鏡像，因本次研究的圖片為胸腔 X 光片，為了不影響圖片的完整性，資料擴增只採用旋轉的方式，機率 0.5 設為旋轉 90 度，機率 0.5 左右旋轉，可以設定欲生成圖片的檔案格式、儲存的路徑、數量。

```
#新增covid19
p = Augmentor.Pipeline("/content/drive/MyDrive/chest_x_ray/train/covid19", "/content/drive/MyDrive/chest_x_ray/new", save_format="PNG")
p.rotate90(probability=0.5)
p.rotate(probability=0.5,max_left_rotation=25,max_right_rotation=10)
p.sample(938)
```

```
#新增pneumonia
p = Augmentor.Pipeline("/content/drive/MyDrive/chest_x_ray/train/pneumonia", "/content/drive/MyDrive/chest_x_ray/new", save_format="PNG")
p.rotate90(probability=0.5)
p.rotate(probability=0.5,max_left_rotation=25,max_right_rotation=10)
p.sample(300)
```

```
#新增turberculosis
p = Augmentor.Pipeline("/content/drive/MyDrive/chest_x_ray/train/turberculosis", "/content/drive/MyDrive/chest_x_ray/new", save_format="PNG")
p.rotate90(probability=0.5)
p.rotate(probability=0.5,max_left_rotation=25,max_right_rotation=10)
p.sample(940)
```

因為正常類別的數量最多，因此不額外做擴增。

下圖為資料生成後的各類別訓練集數量，各類別訓練集資料皆為 1500 張左右。

```
#新增後  #各類別訓練集數量
name = ["covid19","normal","pneumonia","turberculosis"]
for i in range(4):
    train_path = "/content/drive/MyDrive/chest_x_ray/train/{}" .format(name[i])
    read = os.listdir(train_path)
    num = len(read)
    print(num)
```

```
1500
1514
1500
1500
```

最後進行圖片歸一化、設定圖片大小為(224,224)、batch_size 為 64：

```
path_train = "/content/drive/MyDrive/chest_x_ray/train/train"
path_val = "/content/drive/MyDrive/chest_x_ray/val/val"
path_test = "/content/drive/MyDrive/chest_x_ray/test/test"
data_gen = ImageDataGenerator(rescale=1./255)
training_set = data_gen.flow_from_directory(path_train,
                                            class_mode='categorical',
                                            shuffle=True,
                                            target_size=(224,224),
                                            batch_size=64,
                                            )
val_set = data_gen.flow_from_directory(path_val,
                                            class_mode='categorical',
                                            shuffle=True,
                                            target_size=(224,224),
                                            batch_size=64,
                                            )
testing_set = data_gen.flow_from_directory(path_test,
                                            class_mode='categorical',
                                            shuffle=True,
                                            target_size=(224,224),
                                            batch_size=64,
                                            )

training_set.class_indices
```

```
Found 1595 images belonging to 4 classes.
Found 752 images belonging to 4 classes.
Found 752 images belonging to 4 classes.
{'covid19': 0, 'normal': 1, 'pneumonia': 2, 'turberculosis': 3}
```

三、模型建立與訓練

3.1 Densenet 介紹

DenseNet 全名為 Densely Connected Convolutional Network。由許多個 Dense Block 組成，每個 Block 皆採用 bottleneck 結構，而 Dense Block 用了許多「層與層的連結」來達到特徵重用性。Densenet 的架構如下圖所示，假如我們有 L 層卷積神經網路，那就有 L 個(層與層之間的)連結，但 DenseNet 設計成有 L(L+1)/2 個連結，因此 Densenet 有以下的優點：

(1)減緩梯度消失的問題

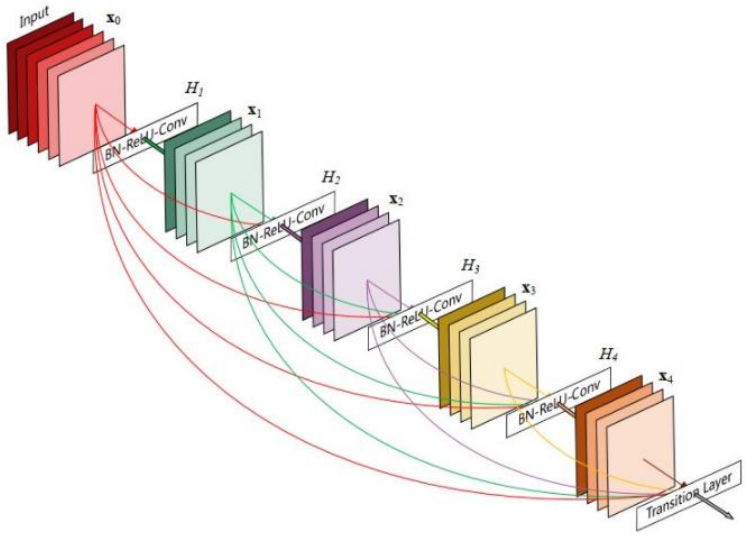
由於 Densely connected，反向梯度傳播十分容易，模型收斂效果佳。

(2)特徵重用性

從低階特徵到高階特徵都會被直連到最後一層卷積層，這讓下一層接受到更全面的圖像資訊。

(3) 減少參數量

對於舊的特徵圖(feature-map)是不需要再去重新學習，能減少許多參數。



3.2 模型訓練

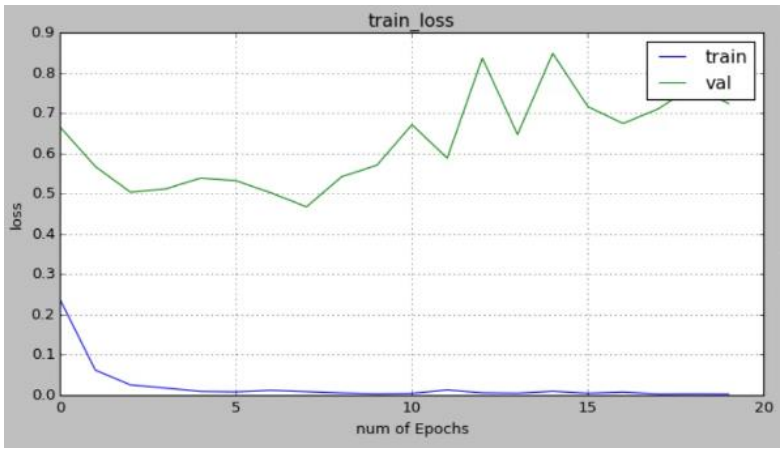
本研究採用 Densenet121 來進行模型訓練，keras 已內建 densenet 架構，模型初始設定優化器為 Adam、學習率 0.001、dropout 為 0.4、loss 為 binary_crossentropy：

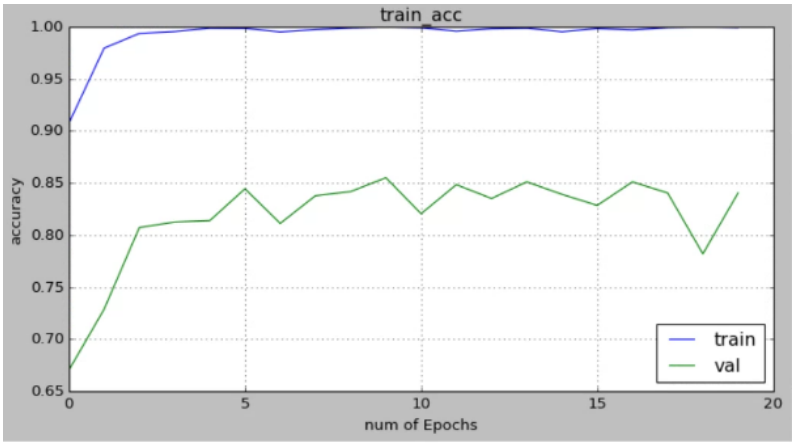
```
#Densenet121
from  tensorflow.keras.applications  import  DenseNet121
def  get_model():
        densenet  =  DenseNet121(weights='imagenet',
                                  include_top=False,
                                  input_shape=(224,224,3)
                                                  )
        model  =  tf.keras.models.Sequential([densenet,
                                              GlobalAveragePooling2D(),
                                              Dense(512,  activation='tanh'),
                                              BatchNormalization(),
                                              Dropout(0.4),
                                              Dense(4,  activation='sigmoid')
                                                                        ])
        model.compile(optimizer=Adam(lr=0.0001),
                                  loss='binary_crossentropy',
                                  metrics=['accuracy']
                                  )

        return  model

model  =  get_model()
model.summary()
```

以下為利用初始設定的參數來進行訓練，得到隨著 epoch 增加，訓練 loss 與訓練精確度的變化圖。





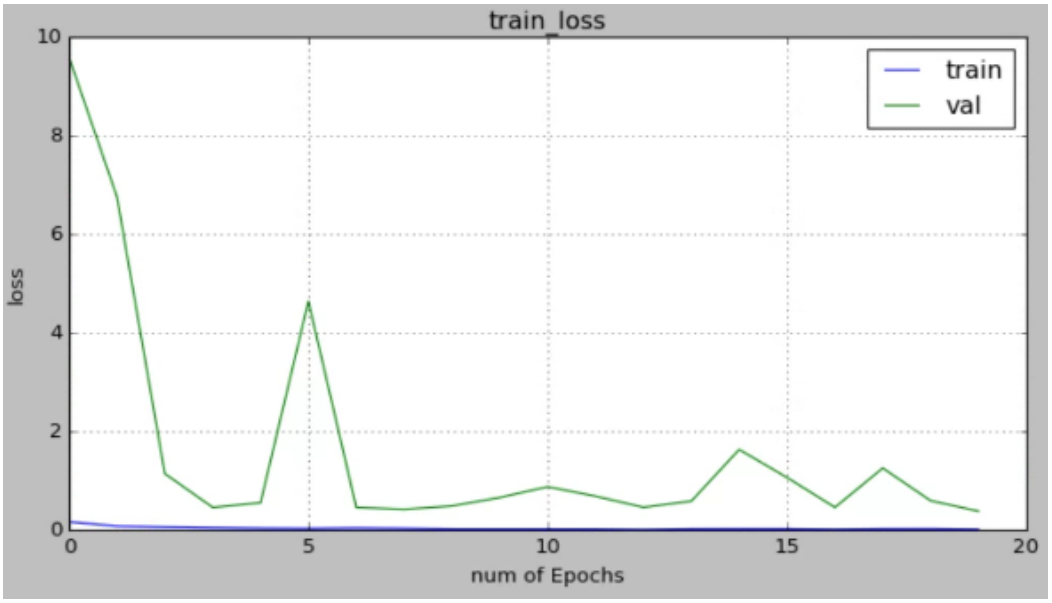
四、超參數優化

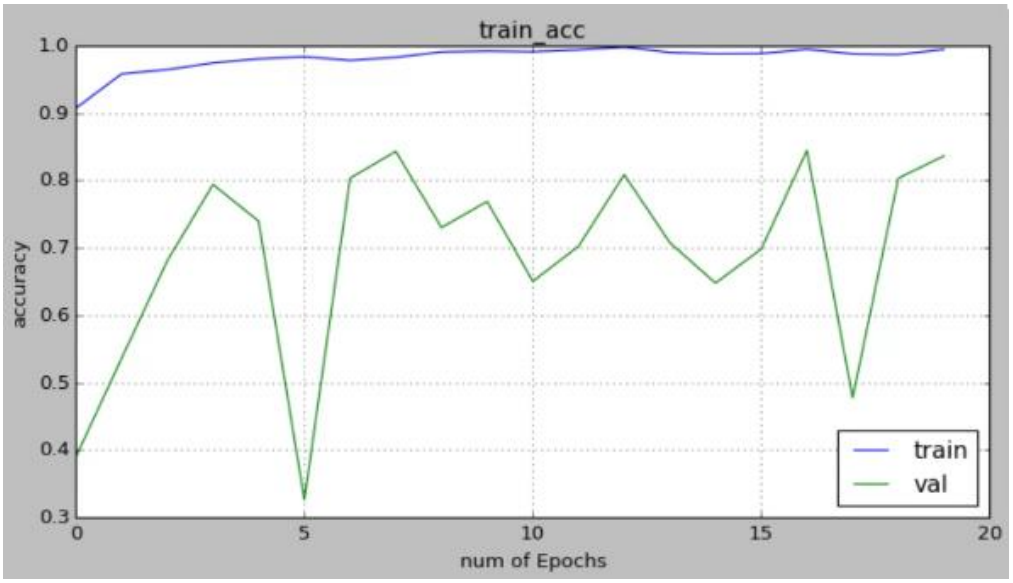
本研究以四因子(optimizer、Learning rate、dropout、Activate function)三水準的直交表(L_9)進行實驗設計，有效降低實驗次數，來找出一參數組合擁有較好的測試精確度。

	optimizer	Learning rate	dropout	Activate function	Train acc	Test acc
1	Adam	0.001	0.3	relu	0.9943	0.8231
2	Adam	0.0001	0.4	tanh	0.9992	0.8564
3	Adam	0.005	0.5	sigmoid	0.9764	0.6609
4	SGD	0.001	0.4	sigmoid	0.9757	0.8457
5	SGD	0.0001	0.5	relu	0.7970	0.7793
6	SGD	0.005	0.3	tanh	0.9980	0.8511
7	Adagrad	0.001	0.5	tanh	0.9940	0.8511
8	Adagrad	0.0001	0.3	sigmoid	0.9214	0.8231
9	Adagrad	0.005	0.4	relu	0.9995	0.8540

從上述實驗設計的結果可以看出，採用 Densenet 進行訓練，普遍的訓練精確度都是都蠻高的，多數有達到 99%，而以測試精確度來看，實驗 2 的測試精確度較高，達到 85.64%。以下呈現實驗 2 的 train

_loss 與 train_acc。





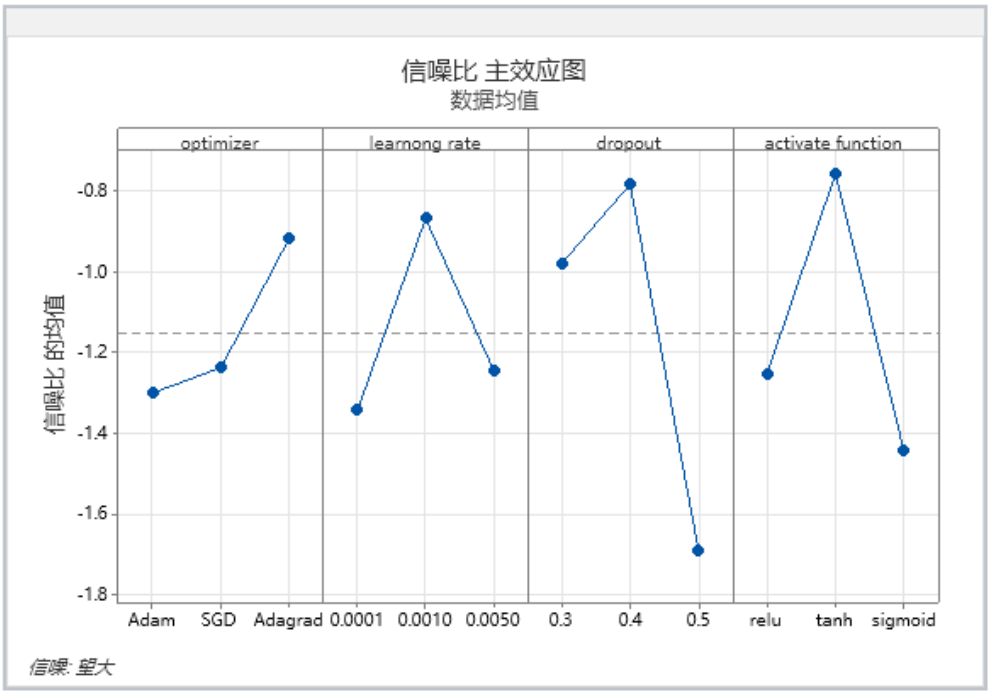
以下為使用 minitab 計算 SN 比，各因子 SN 比最大的水準分別為：

Optimizer: Adagrad

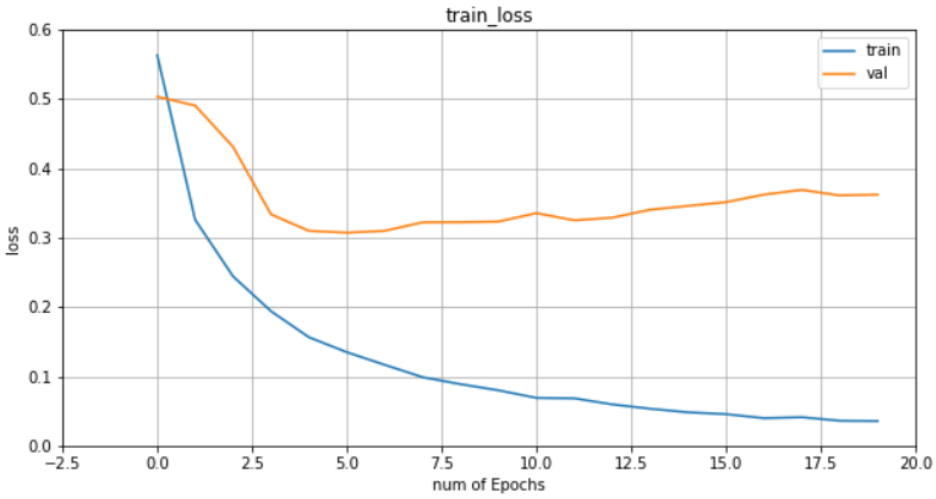
Learning rate: 0.001

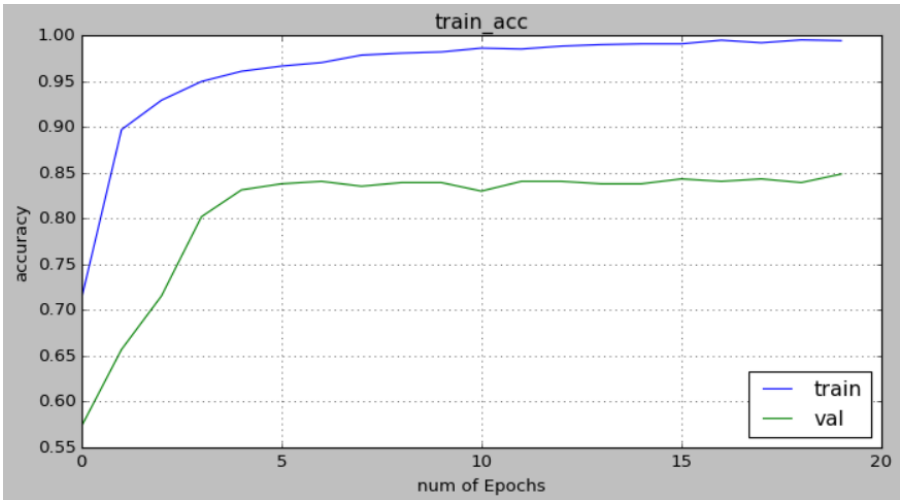
Dropout: 0.4

Activate function: tanh



利用 SN 比所得到的最佳參數組合來進行訓練，得到訓練精確度為 0.9940、測試精確度為 0.8497，比起實驗 2 的精確度略顯低一些。下圖呈現最佳參數組合隨著 epoch 增加的訓練/驗證 loss 與訓練/驗證 acc 的變化。





五、結論

5.1 研究貢獻

本研究透過建立 CNN 模型來協助醫生快速判讀患者的疾病，特別是對新手醫師來說，可能會因為經驗不足而導致對某些變化視而不見，那此模型就可以降低醫師的出錯率。

另外為了讓大型醫院能專注治療病情嚴重的個案，會將篩檢量能分配到第二、三線的醫院、或一些院外的篩檢站，而透過人工智慧的方法能提供醫院具有良好效率的工具，減輕醫院的人力負擔。

5.2 未來展望

可以嘗試增加可辨識疾病的數量，或建立一系統除了辨識疾病之外，還能標記出異常位置，或是有疑點的部位，降低對個人經驗值的依賴，協助醫生進行後續相關醫療處理。

六、參考文獻

1. 胸腔 X 光判讀
https://www.commonhealth.com.tw/article/83517

2. Densenet
https://ithelp.ithome.com.tw/articles/10265328
https://medium.com/%E5%AD%B8%E4%BB%A5%E5%BB%A3%E6%89%8D/dense-cnn-%E5%AD%B8%E7%BF%92%E5%BF%83%E5%BE%97-%E6%8C%81%E7%BA%8C%E6%9B%B4%E6%96%B0-8cd8c65a6f3f