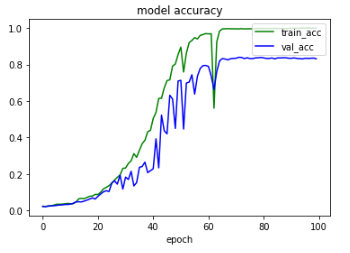
一、HW2

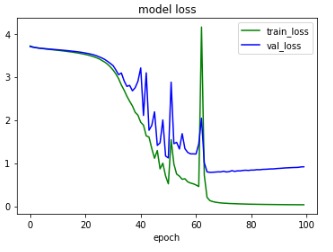
* Method description

1. VGG16
2. Random weight
3. 不使用VGG16最後兩層Fully-connect，降低需學習特徵數量
4. 資料集隨機分為35(train)、30(test)兩個set

* Experimental results – accuracy



accuracy : 0.8315789469501429



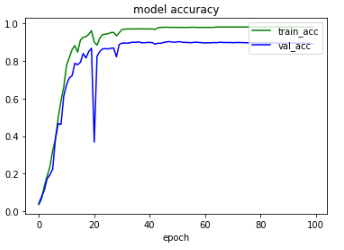
loss : 0.9151371487399989

二、HW2\_3block

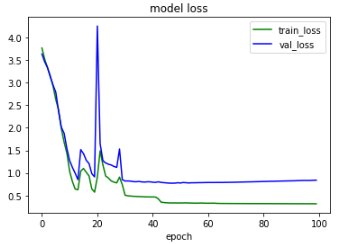
* Method description

1. VGG16
2. Random weight
3. 只保留前三個block，把後面兩個block移除，降低需學習特徵數量
4. 不使用VGG16最後兩層Fully-connect，降低需學習特徵數量
5. 資料集隨機分為35(train)、30(test)兩個set

* Experimental results – accuracy



accuracy : 0.8938596487045288



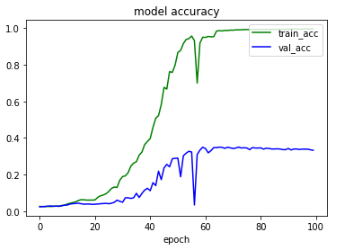
loss : 0.8390824134804701

三、HW2\_3channel\_0

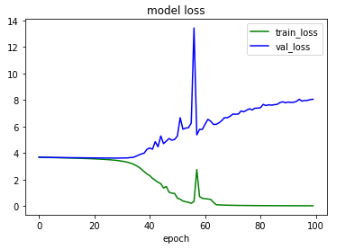
* Method description

與HW2\_without\_random相似，但把原本的1 channel改成3 channel，其他兩個channel以0填滿。

* Experimental results – accuracy



accuracy : 0.3324561403639484



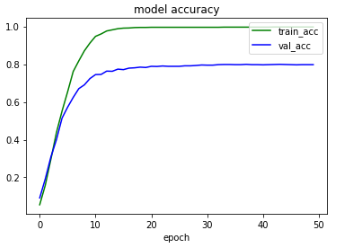
loss : 8.053018941377339

四、HW2\_imgnet

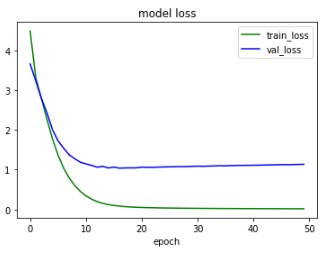
* Method description

1. VGG16
2. 使用imagenet pre-train weight
3. 不使用VGG16最後兩層Fully-connect，降低需學習特徵數量
4. 資料集隨機分為35(train)、30(test)兩個set
5. 需將資料的channel改為3，方法為將其他兩個設為0

* Experimental results – accuracy



accuracy : 0.7973684214709098



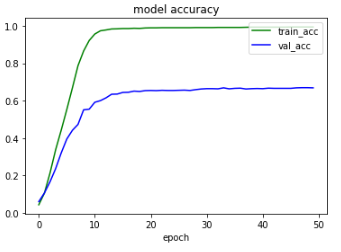
loss : 1.1360951570042392

五、HW2\_imgnet\_deep

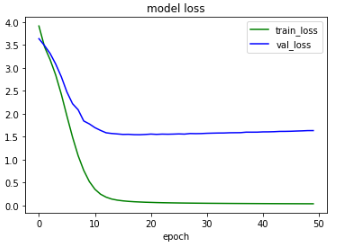
* Method description

與HW2\_imgnet大致相同，但把原本的vgg model的再加一個block，然後pre-train使用作業檔案中的model.h5。

* Experimental results – accuracy



accuracy : 0.6684210530498571



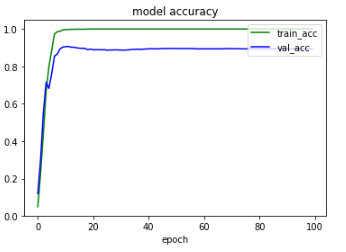
loss : 1.6330566192928113

六、HW2\_imgnet\_vgg19

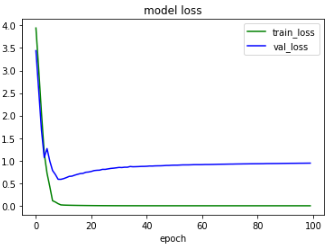
* Method description

與HW2\_imgnet大致相同，但把架構改為VGG19。

* Experimental results – accuracy



accuracy : 0.8947368421052632



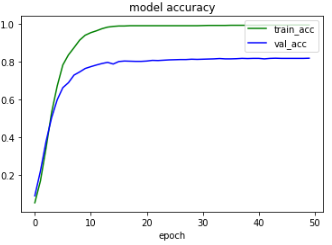
loss : 0.9456656854002831

七、HW2\_imgnet\_x3

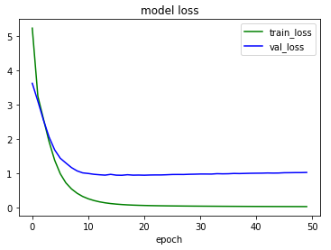
* Method description

與HW2\_imgnet相同，但把其餘兩個channel以原封不動複製的方式使用。

* Experimental results – accuracy



accuracy : 0.820175438178213



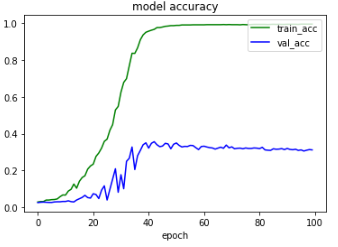
loss : 1.0287599108721082

八、HW2\_without\_random

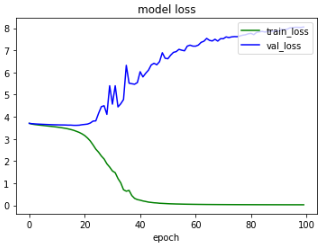
* Method description

與HW2類似，但在分類資料時，不採用random的方式，而是直接將前35張視為train，後30張視為test。

* Experimental results – accuracy



accuracy : 0.311403508785001



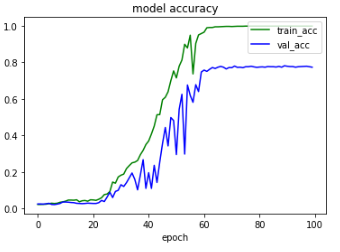
loss : 8.055809844167609

九、HW2\_withtop

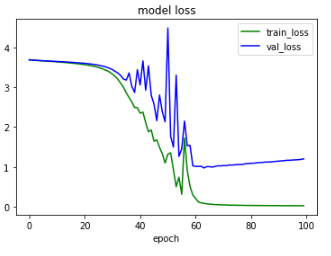
* Method description

與HW2類似，但使用了最後兩個Fully-connect。

* Experimental results – accuracy



accuracy : 0.7736842103171767



loss : 1.2041900910829242

* Discussion of difficulty or problem encountered

這次做了許多不同方式的實驗，為的是想發現各種不同的方法、參數、架構間有何差異。而平均而言，這次以深度學習的方法實作人臉辨識，相較於上次的SAD、SSD都是有顯著的提高。

但這當中可以發現Deep Learning也不是萬能的，以同樣跟上次作業一樣的分法區分train/test set，可以發現準確率是低於上次的，原因大概就是兩個set中的人差異性其實很大(光的照射、角度…等)，所以在train中學到的特徵反而在test中完全不堪使用。因此在這次的作業中，主要的作法都是將資料隨機排列後才進行切割資料集。光是這個小小的一個動作，準確率就可以提高將近50%了，由此可見，再進行深度學習前很重要的一個步驟是確認資料集有沒有任何問題。

接著在上面的結果圖可以發現，有的曲線在收斂前會有很明顯的震盪，但有些卻不會，我發現這之間的差別應該是在於有沒有使用pre-train或是使用random weights，使用pre-train的人在收斂前比較少發生震盪，然而random weights還蠻常的。兩者在收斂的速度上也會有所差別，有使用pre-train的模型在訓練的開始時就開始慢慢地在上升其準確度了。但如果沒使用pre-train，大部分的狀況都要等到20 epochs之後準確度才會開始有所改變。在這部分可以見到pre-train在訓練模型上可以為我們帶來很大的好處。

最後是關於模型架構，原本以為加深是可以增加準確度的，但然而實際加深後卻發現準確率反而降低了。後來問了一下學長才知道，我們這次的訓練資料很少，所以加深模型只是會增加他要被訓練的數量。但然而訓練集並不一定是足夠讓模型學到良好的權重，在這種少資料的模型，試看看降低層數幫助反而可能會更大。於是嘗試了一下將VGG16改為剩下3個block，沒想到效果真的有很顯著的提升，達到跟VGG19差不多的效果。而這應該也是為什麼我們把最後的fully connect layer拿掉，但模型卻能提升的原因，因為fully connect形成的parameters實在太多了。

整體而言，這次的作業目的應該是為了讓我們熟悉Keras的使用，以及深度學習的架構調整方式。我覺得雖然在這次的作業中，剛開始時遇到了許多不同類型的問題，光是處理input檔時就花了滿多時間的。但透過Google，一步一步地慢慢解決，的確也學習到很多的東西。對深度學習模型的操作也更加熟悉了！