題目：HTC Hand Detection

隊名：異顏難進

組員：陳信豪、曾千蕙、郭士庭

1. 題目描述：
   1. 給第一人稱視角的圖，判斷左右手及用 bounding box 標出手的位置
2. 方法流程
   1. 流程架構圖

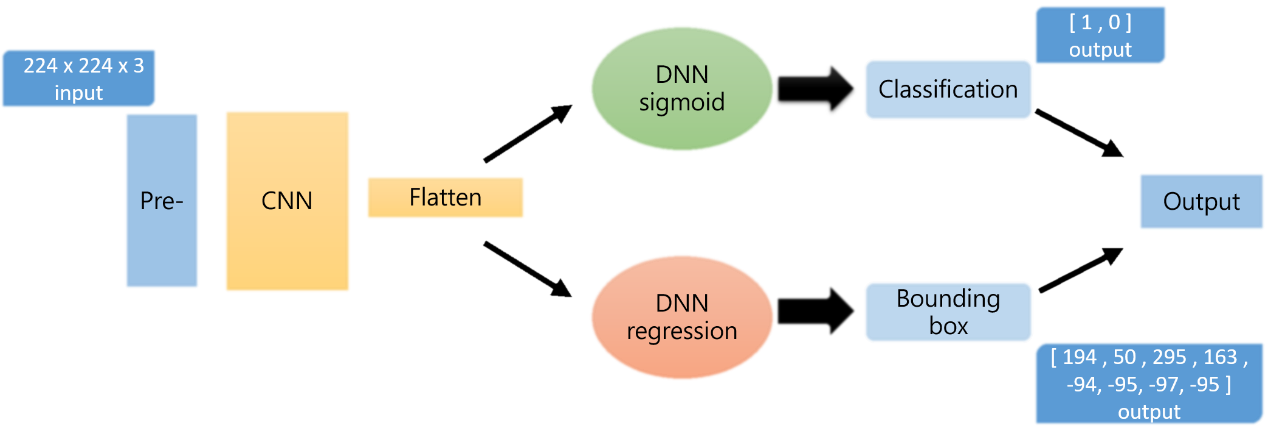
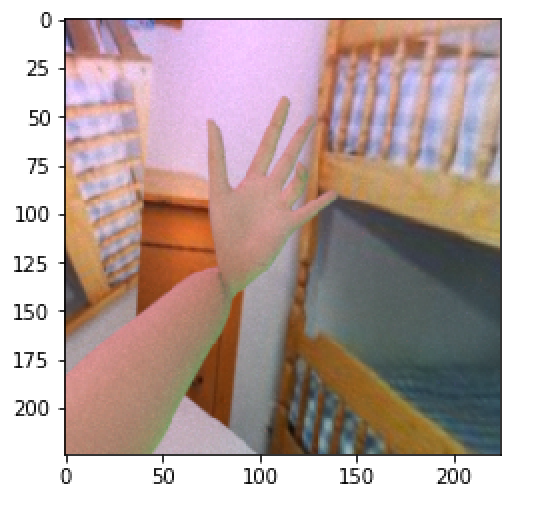
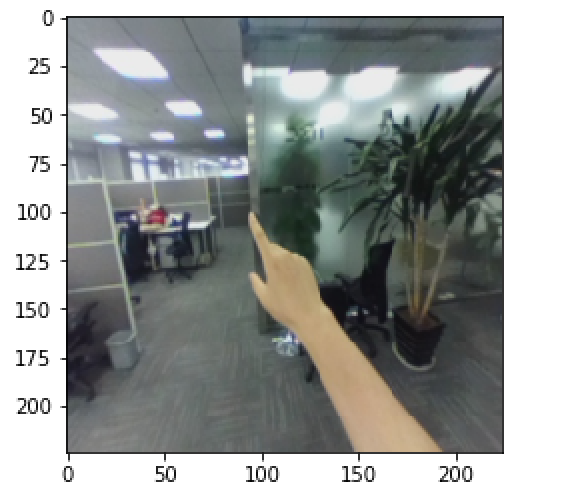
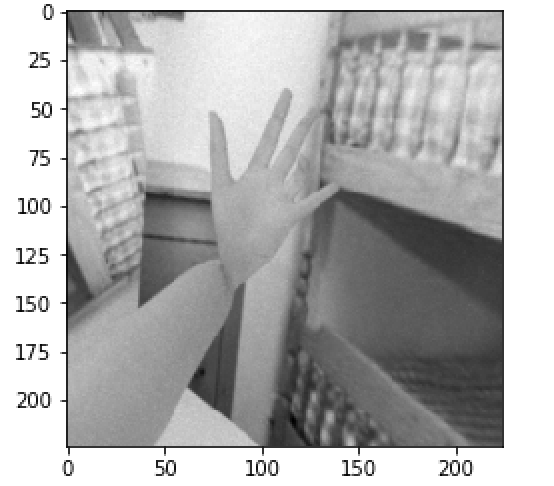
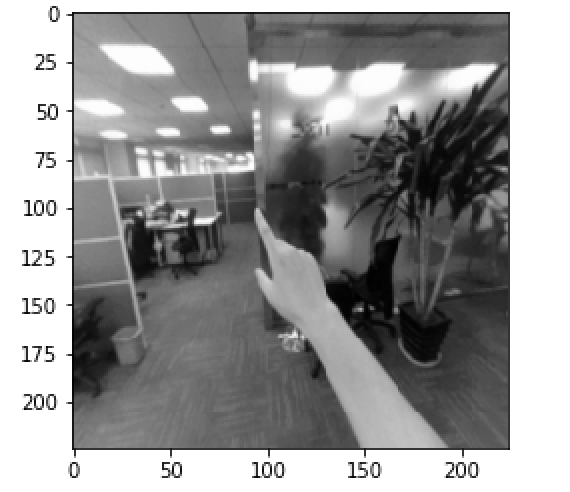


圖 1

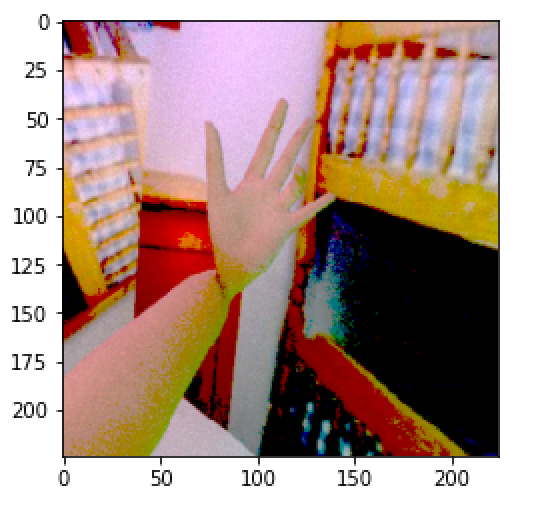
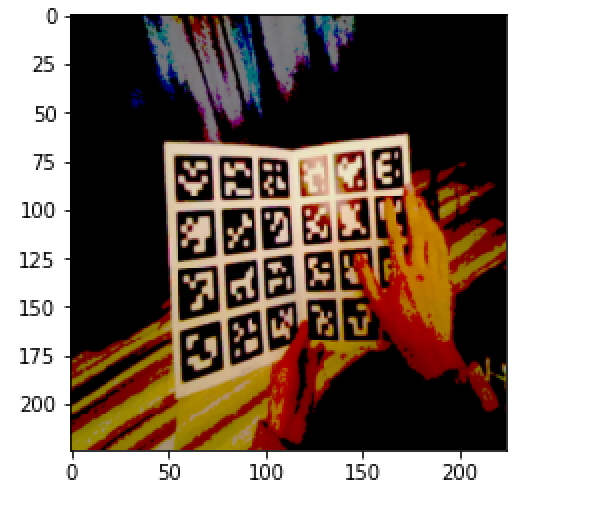
1. 前處理
   1. 一開始由於synth跟vive的圖片大小不一樣，所以會resize成同樣大小的224x224x3。
   2. 有嘗試過rgb2gray變灰階 與 rgb-d把圖片加深處理，但效果都沒有很好，所以最後跑資料的時候，都沒有做色彩上的處理。
   3. 範例：
      1. 原圖

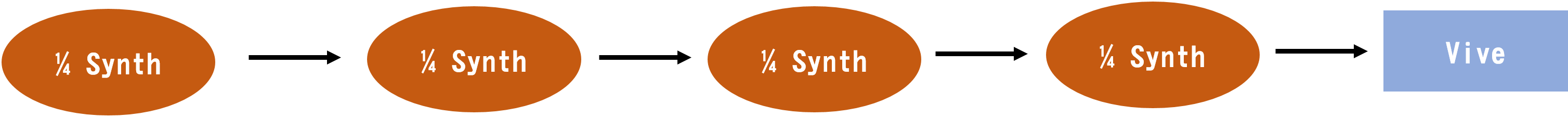


* + 1. 灰階



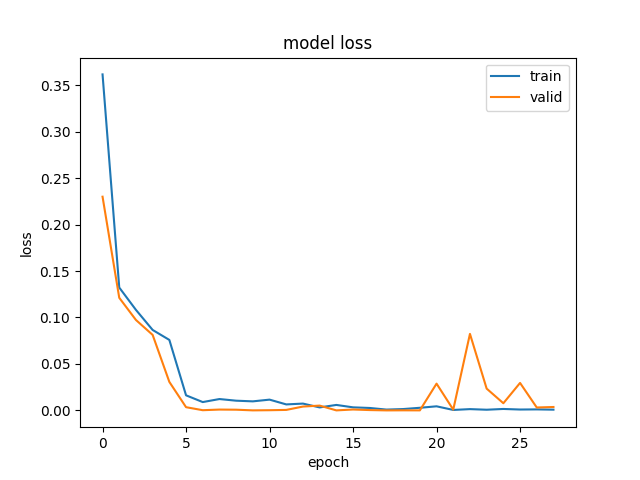
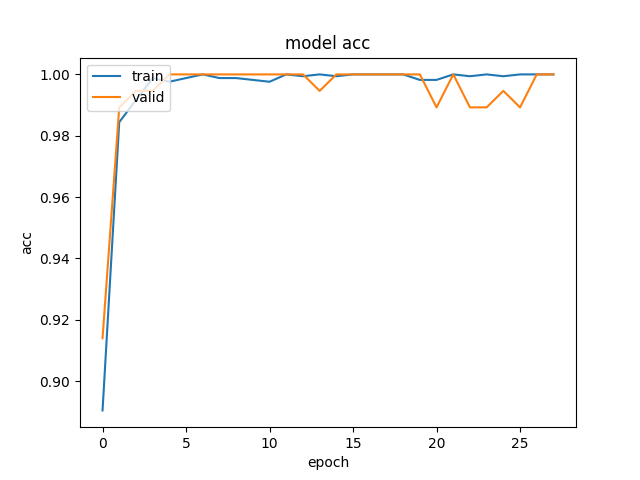
* + 1. 加深



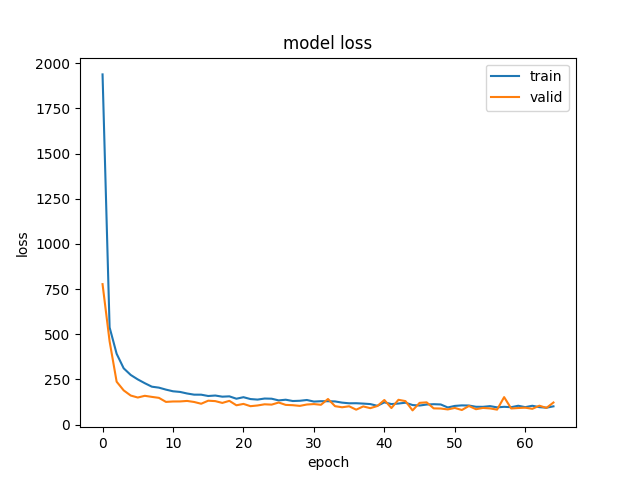
1. 嘗試過的CNN model架構
   1. 最一開始嘗試過CNN + DNN，同時輸出8個維度的output包含判斷左右手，但效果極差，所以改用圖1的架構，將左右手分類及bounding box分開預測。也就是說，用一樣的CNN架構，會先train一個判斷左右手的exist model (準確度可達98%以上)，然後再用此model的參數作為初始值去train另一個 bounding box的model。
   2. 在這之後的整體架構上都沒有修改，只有在CNN這塊做調整及變換
   3. 在CNN這部分嘗試過了傳統 CNN、ResNet50、InceptionV3、Inception ResNet V2。
   4. 最後我們的實驗結果是 InceptionV3 與 Inception ResNet V2 表現最好。
2. Train data的順序
   1. model跑data的流程
   2. 會先將 synth data 切成四份，然後一次只 train 1/4 的 synth 資料，最後跑完synth資料之後，再去fine tune資料很少的vive資料。
3. model 介紹與結果
   1. 用improvement tips 之前的最好成績  
       (後面會再介紹我們 improvement tips)
   2. (offline judger 只有 六張圖、DeepQ online judger 有上千張)

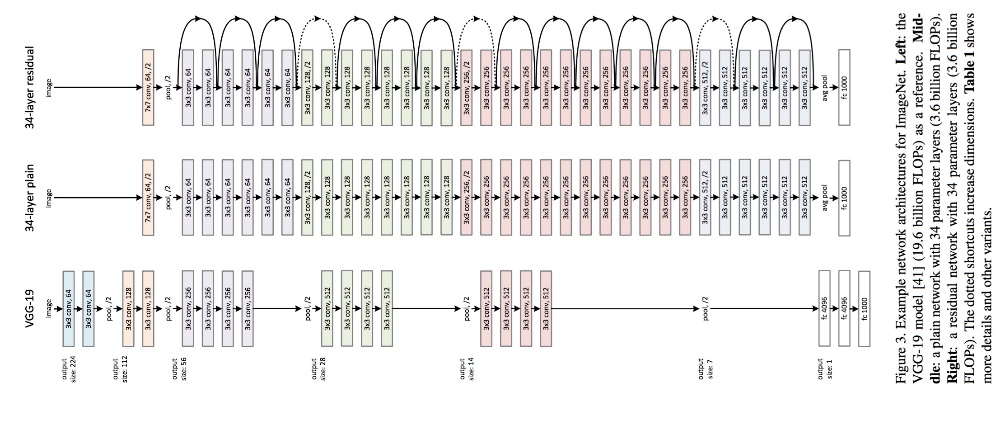
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Best off-line score | Best on-line score | Parameters | Depth |
| CNN + DNN | 0.361 | 0.09 | 38,984,072 | 21 |
| ResNet50 | 0.18 | 0.11 | 25,636,712 | 168 |
| InceptionV3 | 0.75 | 0.3 | 23,851,784 | 159 |
| InceptionResNetV2 | 0.75 | 0.33 | 55,873,736 | 572 |

* 1. CNN + DNN (一個DNN負責分類，另一個負責bounding box):
     1. 這部分的 CNN 用 4~5 層嘗試過，但效果都卡在 0.1 左右
     2. Exist model (判斷左右手)

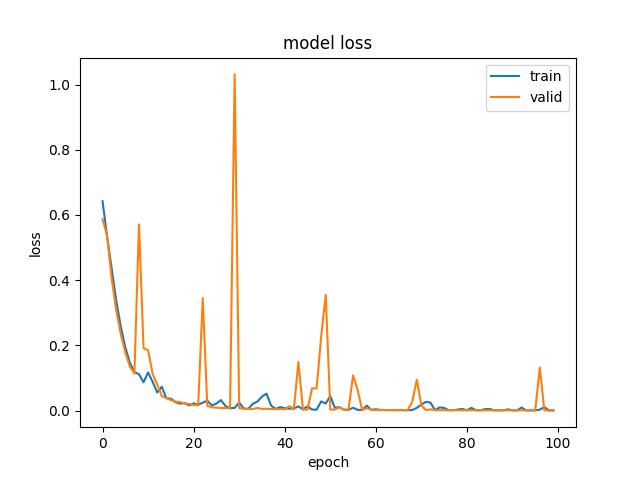
* + 1. Bounding box model(regression 畫出八個位置點)

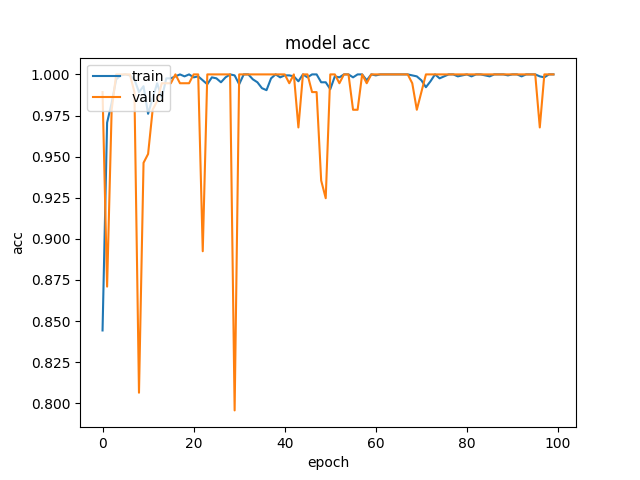


* 1. ResNet50 + DNN:
     1. 這部分使用 Keras 提供的pre-trained weights for Classification on ImageNet 的 ResNet model，特色是每一區塊會把原本的資料跳過 convolution，並再將之加到 convolution 出來的結果，這樣可以多保留一些原始的資料特徵。
     2. 架構:

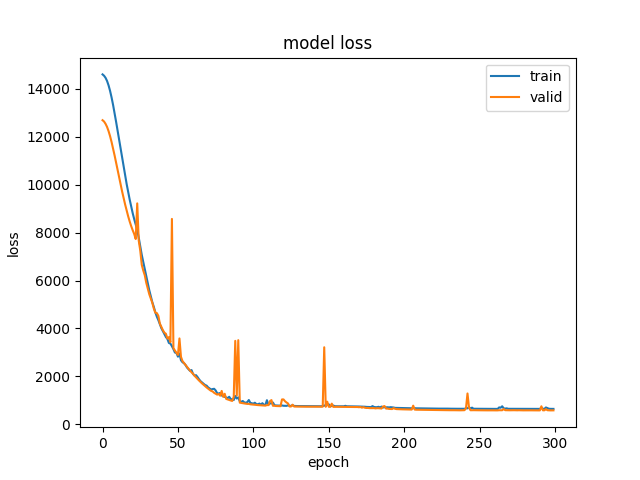
(圖片來自：http://euler.stat.yale.edu/~tba3/stat665/lectures/lec18/img/?C=M;O=D)

* + 1. 討論：這個 model 跑出來的結果雖然 off-line 的結果變差(0.361 -> 0.18)，但on-line的結果有略微的進步(0.09 -> 0.11)。而進步的幅度不明顯有可能是因為我們在嘗試 ResNet50 時，沒有將 synth 的資料 train 的很齊。
    2. Exist model (判斷左右手):

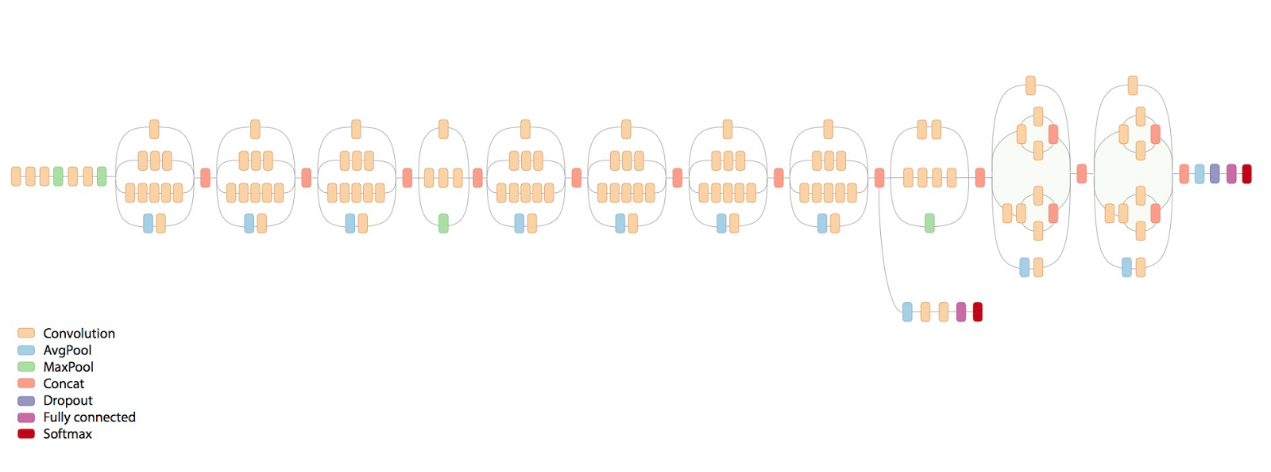




* + 1. Bounding box model:

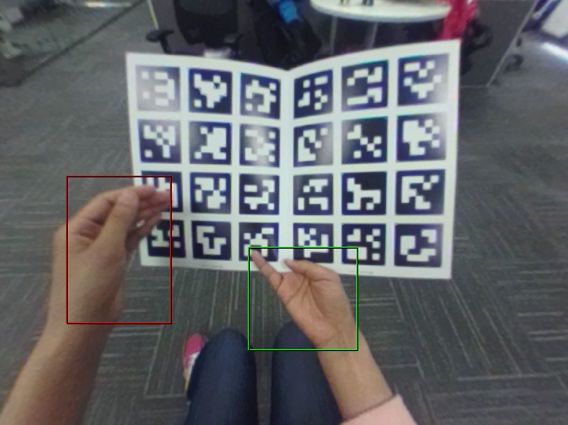


* 1. InceptionV3 + DNN:
     1. 介紹：這部分也是使用 Keras 提供的ImageNet pre-trained weights分類的 model，這 model 特色是每個區塊會平行做 convolution，最後再將結果合併。
     2. 架構：

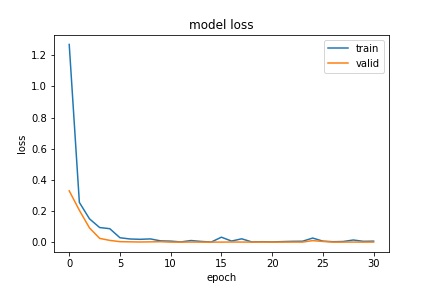


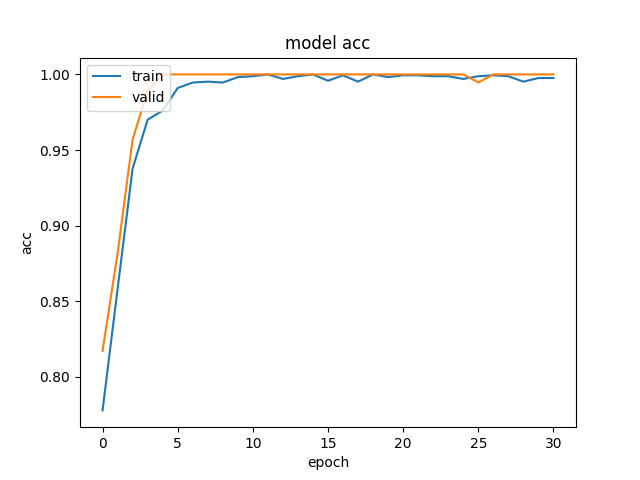
(原圖來自: https://goo.gl/CS5cqC)

* + 1. 討論：這個 model 效果就好非常多，一開始 off-line 的 judge 測試就可以到0.5，上傳 DeepQ 的 on-line 成績也有0.28，後來就持續用這個model做測試。
    2. 輸出結果:

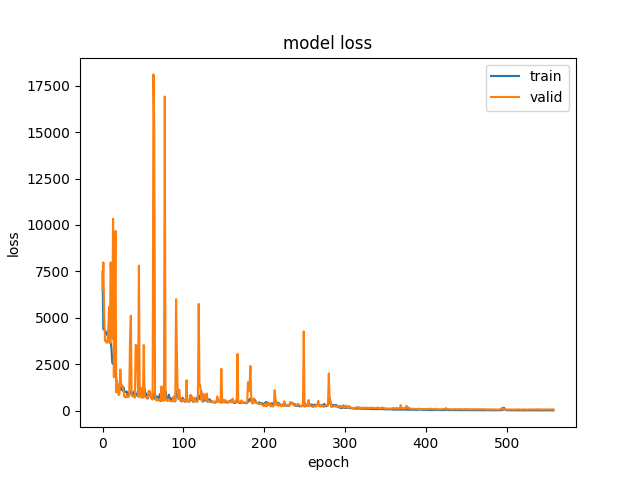


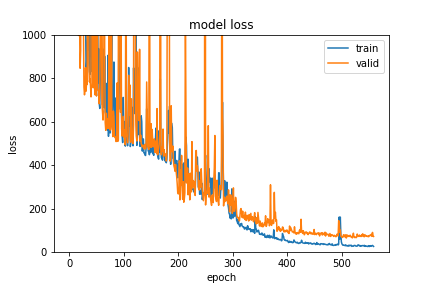
* + 1. Exist model (判斷左右手):

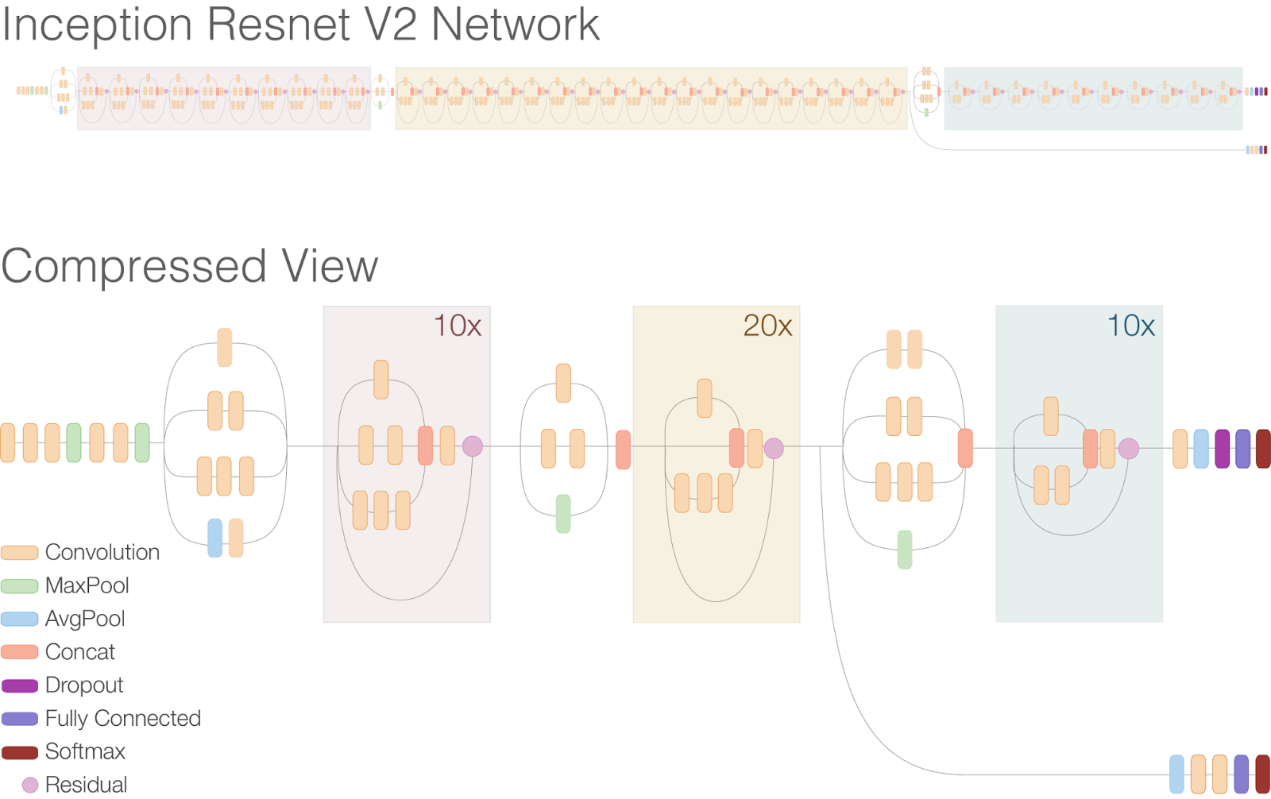




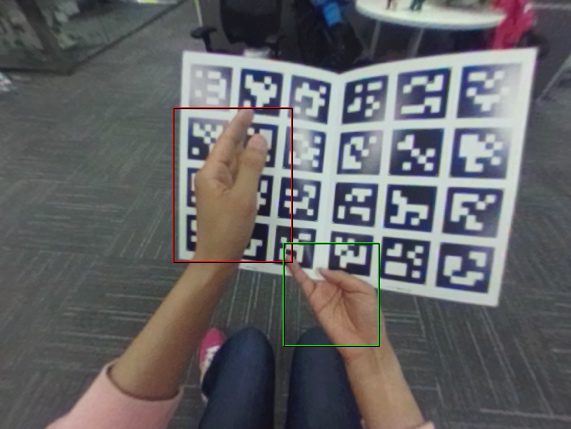
* + 1. Bounding box model:  
       （這部分的train loss 浮動很大，不過長久之後還是會收斂。）



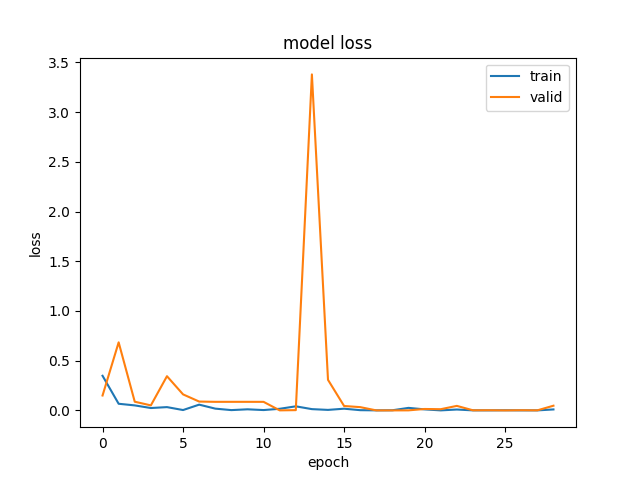


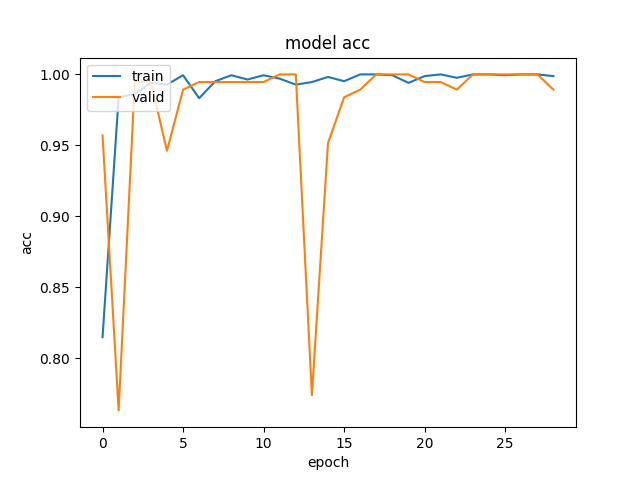
* 1. InceptionResNet V2 + DNN:
     1. 介紹：這也是 Keras 提供的 model，它的特色就是結合 Inception 與 ResNet 的架構
     2. 架構：

(原圖來自: https://goo.gl/PNjtV0)

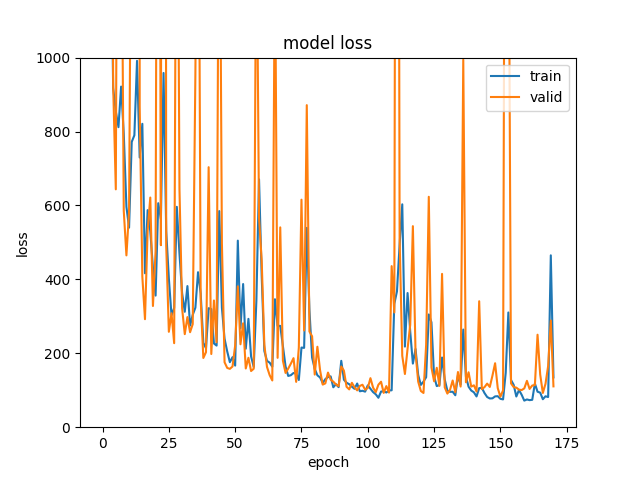
* + 1. 討論：這個方法在文獻測試上的效果是比 Inception 和 ResNet 好的，但是由於我們在跑這個 model 時，時間不夠，所以synth的資料只有拿 1/4 去 train，剛開始跑出來的效果，雖然off-line 的 judge 成績有到 1.0 的高分，但上傳 DeepQ 的成績不如 InceptionV3。**不過用了Improvement Tips，並開始多 train 幾回之後，效果就漸漸超越InceptionV3了**，我們猜測是因為這個model的能力結合了 Inception和 ResNet，所以效果比較好，且不用 train 到那麼多的 synth 資料即可超越 InceptionV3 的效能。
    2. 輸出結果：

1. Exist model(判斷左右手):

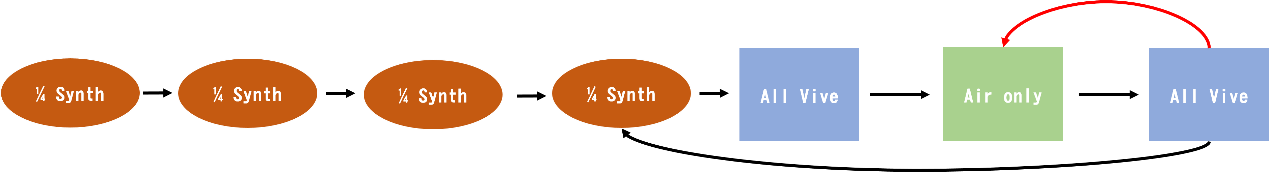




1. Bounding box model



1. Improvement Tips
   1. CNN model的替換，替換上述介紹的CNN model 來提高效果。
   2. 替換 CNN model 後最好的結果仍只有到 0.3，觀察後發現，book 資料的準確度蠻高的，但 air 很低，我們猜測可能是 air 的背景太亮的關係，所以我們嘗試用 opencv 做一些圖片的加深或調暗的處理，然後再丟進 train 過 synth 的model 來 train，但效果反而變差了，所以放棄這部分的嘗試。
   3. Data training order: 由於 air 的辨識率極低，所以我們就嘗試多 train 幾次只有air的vive資料，然後再丟全部的vive資料下去train，這樣的效果有明顯的上升一些，而我們又重複了這個部分，就是1/4 synth -> vive -> air only -> vive (下圖中黑色的路徑) 然後testing。每次跑完一個循環，成績就都會有稍稍提升，但大概在第四次循環左右就沒有提升了，後來就又改成下圖中紅色那條路徑，就是只有train air only跟vive，沒有再回去train synth，結果又開始上升了! 目前InceptionV3 跟 Inception ResNet V2的model都有因為每多 train 一個循環就提升一些分數。



1. 最後的結果與討論：
   1. 成績結果：

|  |  |  |
| --- | --- | --- |
| Model | Best off-line score | Best on-line score |
| InceptionV3 | 1.0 | 0.4749 |
| InceptionResNetV2 | 1.0 | 0.555 |

* 1. 討論：最後最佳的結果是用 Inception ResNet V2 的 model 按照前面所說的 data training order 來重複 train。

1. 改進與檢討：
   1. 如果時間允許應該多跑很多次循環，直到收斂，但由於比賽時間截止，所以目前結果只有到這裡。
   2. 嘗試其他的 model, 例如YOLO [1], faster RCNN [2] 等等
   3. 我們的 model 是以整張圖片作為 input，在 output 時一起預測左手和右手的bounding box。然而其他類型的 bounding box model 會先將圖片切成 grid，對 grid 預測圖片類別以及 bounding box，或是先產生可能的 bounding box，再對 bounding box 做處理以及預測，也就是說，我們可以嘗試將圖片的局部作為 inputs，相較於將整張圖片直接作為 inputs 而言，雜訊可能比較少，model 的負擔可能也比較小，所需的參數量也可以減少，這或許是一個我們可以前進的方向。
   4. 針對synth的圖片再做優化，使用 CycleGAN 的架構，或參考[3] ，生成出更擬真的照片再用於訓練。
2. 參考文獻：

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788).

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

[3] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2016). Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*.