HTC & ADLxMLDS 2017 Competition Report

Egocentric RGB Hand Detection

台大工科所 葉峻孝 r04525061
台大工科所 郭漢遜 r06525087

1. Introduction

1.1. Intro of Competition

此次競賽實作是需要利用深度學習的方法，將一群大量的手部圖片做 training，能夠分離背景，偵測到手部的區域。並且需要分辨被偵測到的是左手還是右手。HTC 給我們 training 的圖片來自 DeepQ-Synth-Hand Dataset 的 100000 張利用工具生成的圖片，並且每張圖片都有標上手部區域的 label, mask。並且有少數的 DeepQ-VivePaper Dataset 的真實手部照片位於 Fig .1.

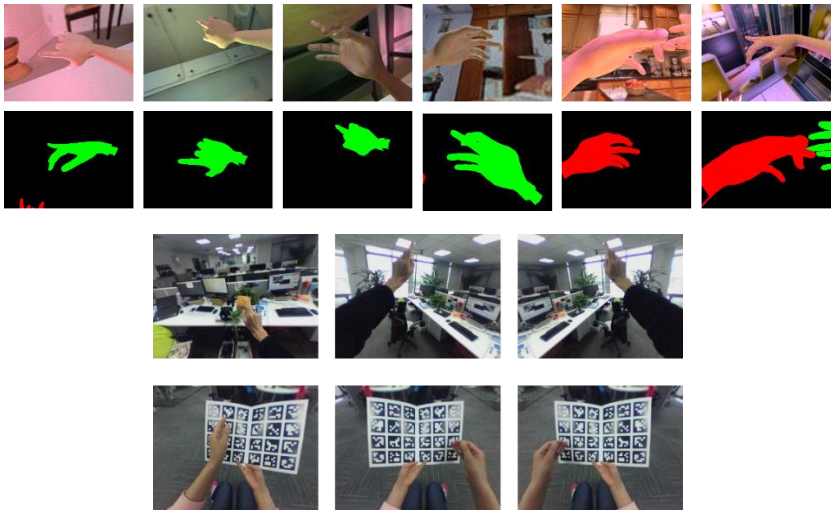

Figure 1. HTC training data

此實作手部辨認切割的方法有許多種，最多人使用是 Faster R-CNN, Mask R-CNN, YOLO-2, DarkFlow…。這些方法都是用來做 Object Detection 的經典方法。但因為可以看到 HTC 給的 Training data 跟 testing data 的圖片其實有很大的差別是 training data 是運用影像生成的圖片，但 testing data 是真實的圖片。所以 HTC 有提供一些方法能夠把 training data 利用 image style transfer 來轉換成比較像真實圖片的 data。像是利用 Conditional GAN 來用 style transfer 讓生成圖片看起來像真實圖片，並且就能夠實際運用在 Training 上面。

1.2. Proposed Procedures

我們實作分割手部主要是先利用 FCN 訓練所有的 Training data。然後之後會得出許多被分割後的手部影像圖片，我們利用影像處理的方法，來偵測手部區塊的 Bounding Box。因為想要讓圈出來的 Bounding Box 區域能夠越像真實 label 的區域，我們運用一個 Regression model 來修正預測的 Bounding Box，判斷使他學習可能的位移量，讓我們預測出來的 Bounding Box 區域能夠移動到更像真實的 Bounding Box 區域。

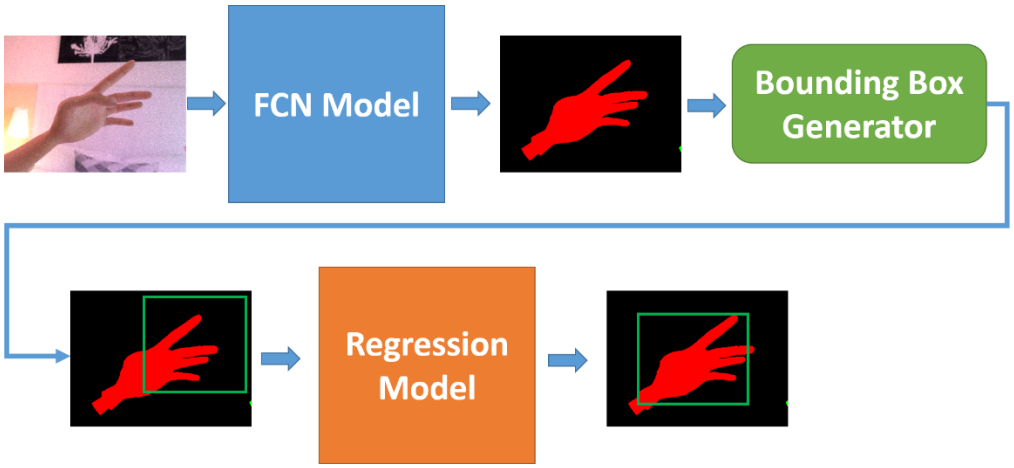

Figure 2. Proposed method procedure

上述 Fig. 2. 為主要的流程圖，最後能夠得出一個準確的座標來預測手部的區域。並能夠判斷偵測出來的手為左手還是右手。

2. Proposed Method

2.1. Modified FCN (Fully Convolutional Networks) Model

我們使用 Ronneberger et al. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2015. 這篇 Paper 的方法來實作 FCN。主要的功能是希望能夠藉由 FCN 來實現圖像分割，但他這篇 paper 是改良 Long, Jonathan et al. "Fully convolutional networks for semantic segmentation."這篇最原始 FCN 的論文，不只是在架構上面有更動，在流程上面也有不一樣的設計，之所以選擇這個改良過的 FCN 是能夠幫助我們在分類手部區域與非手部區域時能夠有比較好的效果。我們的想法是把輸入圖片利用 mask 的檔案，分為左手(紅色)、右手(綠色)、背景(黑色)分為三大類丟進去 model，幫助我們能夠在一張有手與背景的圖片當中成功分割手部區域。以下 Fig. 3. 是我們所改良設計的 FCN model。

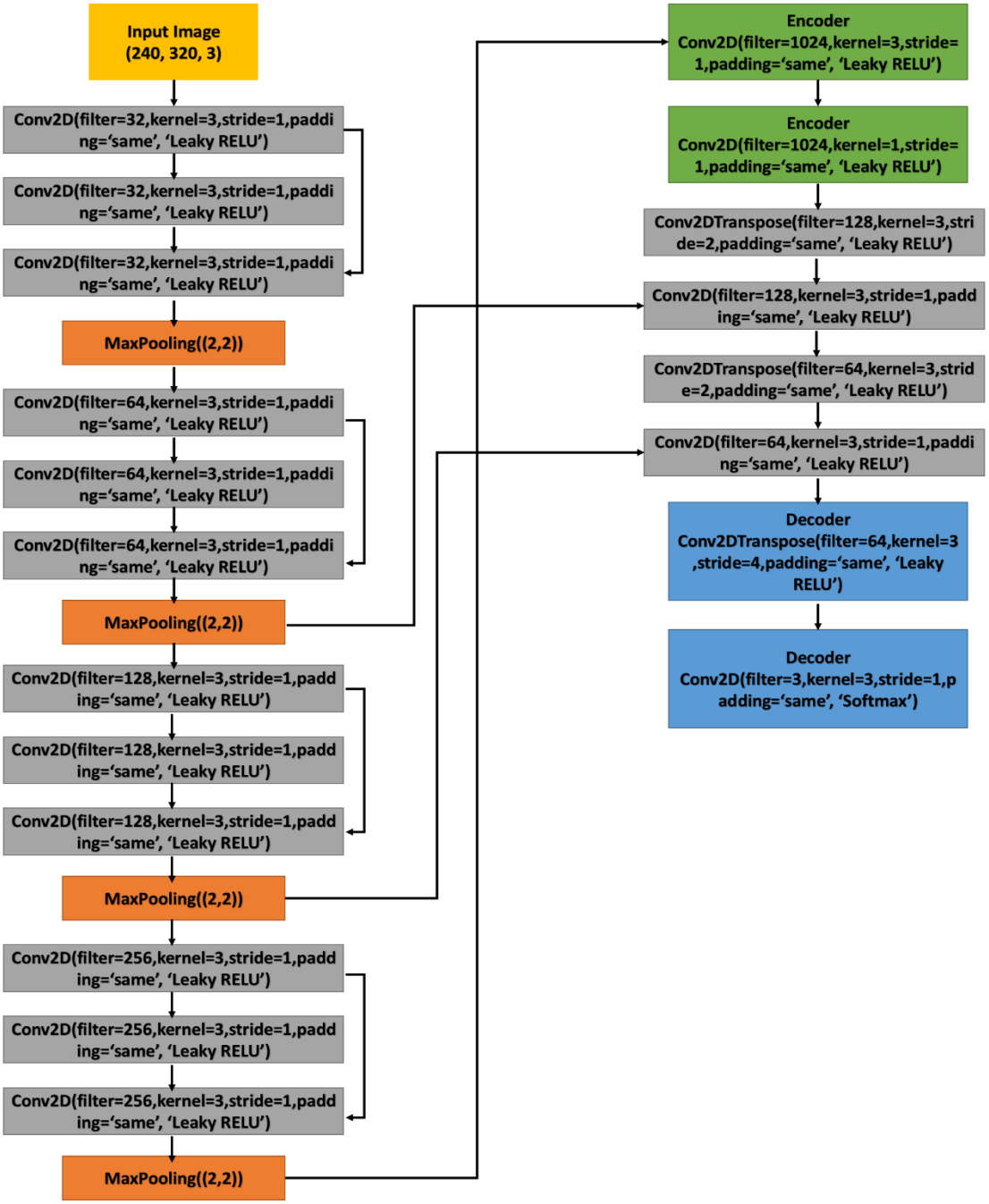

Figure 3. FCN model

我們輸入的訓練圖片大小為(240, 320, 3)的 images。先進入三層 Conv2D 層，filter 為 32, kernel =3, stride=1，我們運用 Leaky_RELU 當作激活函數。通過三層之後的 Output 再經過一個 MaxPooling 來做池化的功能。我們利用同樣的架構對其使用三次的 3 層 Conv2D 層，filter 為 64, 128, 256。所得出的輸出送進一個 filter=1024 的 Encoder。對 Encoder 的輸出再經過兩個 Conv2DTranspose。Kernel 各為 128 與 64。最後經過一個 Decoder。輸出為三類，分為右手、左手、背景(不是手)並且運用 softmax 輸出。選擇一個最大的機率輸出。

2.2. Proposed Bounding Box Generator

從上述 FCN model 我們可以將一張圖片分割成手的區塊與背景(非手的區塊)。我們依照 HTC 提供的 Data 中的 mask，將預測出來是左手的區域標成紅色，預測出來是右手的區域標成綠色。其他非手的區域都標成黑色。這時我們需要產生 bounding box 時，需要得到手部區域的 Top Left (x, y)與 Bottom Right (x, y)座標。我們初步的演算法為偵測整張圖片，若 RGB value 是綠色的話，就先把座標存在一個 axis_list，整張圖辨識完之後會得到一個完整的 axis_list，當中會有被辨認為綠色的座標群。我將此座標 x 部分存在一個新的 list，座標 y 部分存在另一個 list。

接下來我們對兩個 list 作 sorting，將裡面的 x, y 數值個別排序。所以會得到 sorting 由小到大排序的 x, y 座標數值。如 Fig. 4.若我們直接取得綠色區域的座標最小值 (x_min, y_min)當作 Bounding Box 的 Top Left 與綠色區域的座標最大值(x_max, y_max)當作 Bottom Right，這時會發現 Bounding Box 通常都會框不到正確的手部區域。

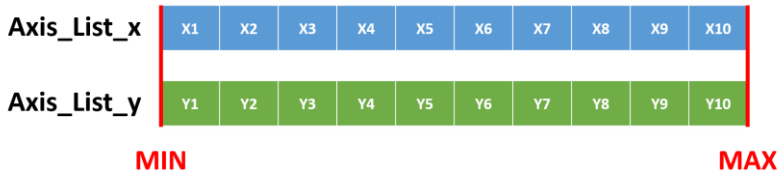

Figure 4. Bounding box method 1

在這個步驟我們觀察了許多 FCN 預測過後的分割圖片，其實大部分的圖片都會有許多的雜訊，例如: 有一些背景被辨認為是手部的區域，或是有些手的區域被辨認為背景。如果我們直接取最大最小的座標值去框的話，通常都會被雜訊所影響。而得到不正確的數值。

後來我們想了一個方法，如 Fig. 5.我們設定了兩個閥值 Thread_1 跟 Thread_2，假設在 sorting 過後的 list 不要取最大值，而是我們取排在 sorting list 中座標百分位數是 10 與百分位數是 90 的座標，這時 Thread_1=0.1 Thread_2=0.9，我們利用這兩個閥值取到 list 當中的座標，這時框出來的 Bounding Box 的效果就會比直接取最大最小值的座標的效果好很多。

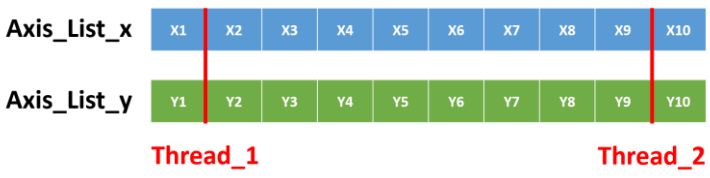

Figure 5. Bounding box method 2

為了要解決 FCN 切割手部區塊時，有時候會將左手辨認成右手，此時如果做 Bounding Box 時就會有許多的錯誤發生。但我們觀察到，通常 Training 時的圖片，若是左手的話，通常都會在整張圖片中的右半邊，右手亦然。這代表幾乎不

會有左手是整支出現在圖片的右半部區塊，這時我們使用一個演算法。我們把圖片分為左中右三個區塊，各占 30%、40%、30%。這時我們判斷圖片，若有右手切割塊(綠色)出現在左邊 30%的區塊，我們認定改變這些右手區塊成為左手區塊(紅色)。運用此演算法能夠有效提升我們辨認左右手區域的正確度。

2.3. Bounding Box Regression Model

我們設計另外一個演算法來改進 Bounding Box 座標的準確度，因為我們發現有時我們框到的 Bounding Box 其實跟實際上的 Label 座標相差不大，但是不夠精準。所以所以將找到的 bbox 的 image 送入 model 產生一個四維向量, 代表這張 bbox 應該往甚麼方向移動或縮放, 所以 input = 一張在手附近的截圖, output = 這塊圖的位移及縮放參數[x,y,w,h]。以下是 Regression 的 model。

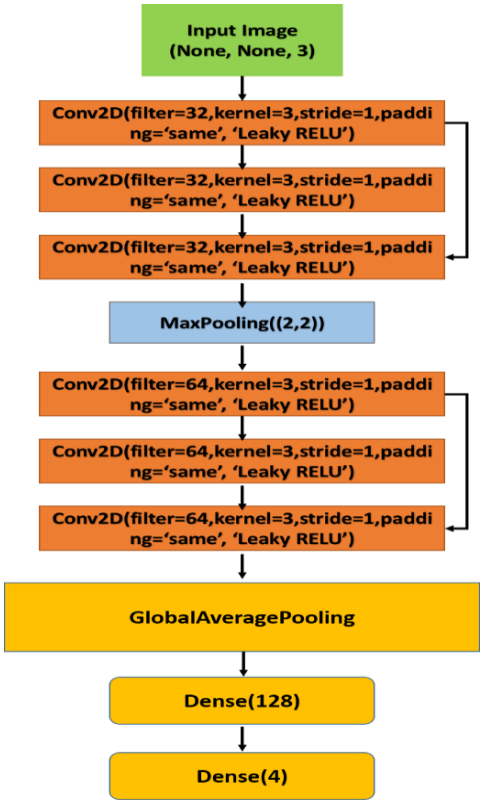

Figure 6. Regression model

可以從上面的圖 Fig. 6 看到，我們運用輸入是從正確的 label 座標加入一些雜訊位移所產生出來 Bounding Box 割出的圖檔，經過 3 層 Conv2D 層，kernel=32, strides=1。而激活函數為 Leaky_RELU。再來把輸出放入一個 MaxPooling，接下來再往下加 3 層 Conv2D，kernel=64。最後接上一個 GlobalAveragePooling，並且接上 Dense(128)、Dense(4)。最後輸出是希望能夠輸出 4 個 value。各代表 Top_left、Bottom_right 的位移量。

3. Experiment

3.1. Results of FCN Training

我們運用 FCN 來做 Training，總共我們 train 了 HTC 所給的 dataset 從 s000-s008。每個 dataset 都有 10000 張的圖片。我們的 FCN model 的參數: batch_size 為 10、Learning_rate = 0.005、Optimizor = 'Adam'，使用的 loss = Categorical_Cross_Entropy、所 train 的 Epoch 為 s000-s004 datasets 總共每個各 train 了 5 個 epoch，而 s005-s008 各個各訓練了 3 個 epoch。以下 Fig. 7.是實際上的輸入跟 model 輸出。我們拿 s009 dataset 來做 testing。

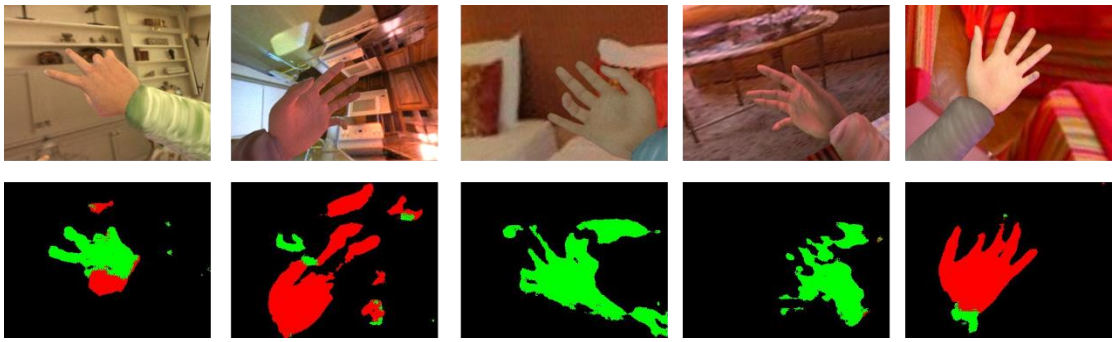

Figure 7. Result on HTC training data

以上是 model 預測後所產生的圖片，可以看到第二張、第三張圖片就會受到背景影響，所以會看起來 mask 沒有那麼準確，有些區域都會包含到背景。第一張圖片會受到一些雜訊所影響，所以會有些右手(綠色)區域會包到一些左手(紅色)的區域。

3.2. Bounding Box Generator

我們運用上述所說明要抓 Bounding Box 的做法。運用雙重 Thread 的改良之後能夠避免掉雜訊的影響造成 Bounding box 的預測錯誤。以下 Fig. 8.是我們的一些實測結果。

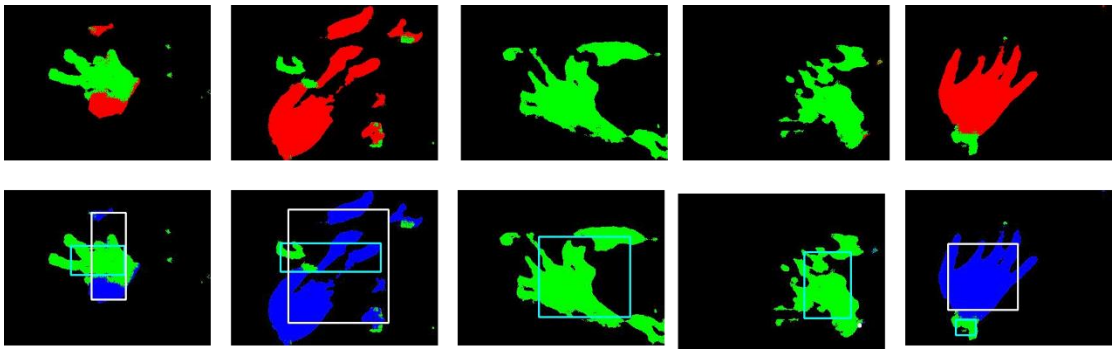

Figure 8. Result on HTC testing data

可以看到我們抓到的藍色框就是用來抓右手(綠色)區域，白色框就是用來抓左手(紅色)區域。以上的結果，可以得知其實還是會受一些雜訊影響，所以導致有些圖片會被認為有兩隻手，但其實只有一隻手。例如第一張圖片其實應該是只有右手，但因為偵測出有左手的區塊，所以辨認有兩隻手。這部分還需要作改良。接下來在 Fig. 9. 我們實作在 HTC 所給定在本機的 testing data 總共有六張圖片。

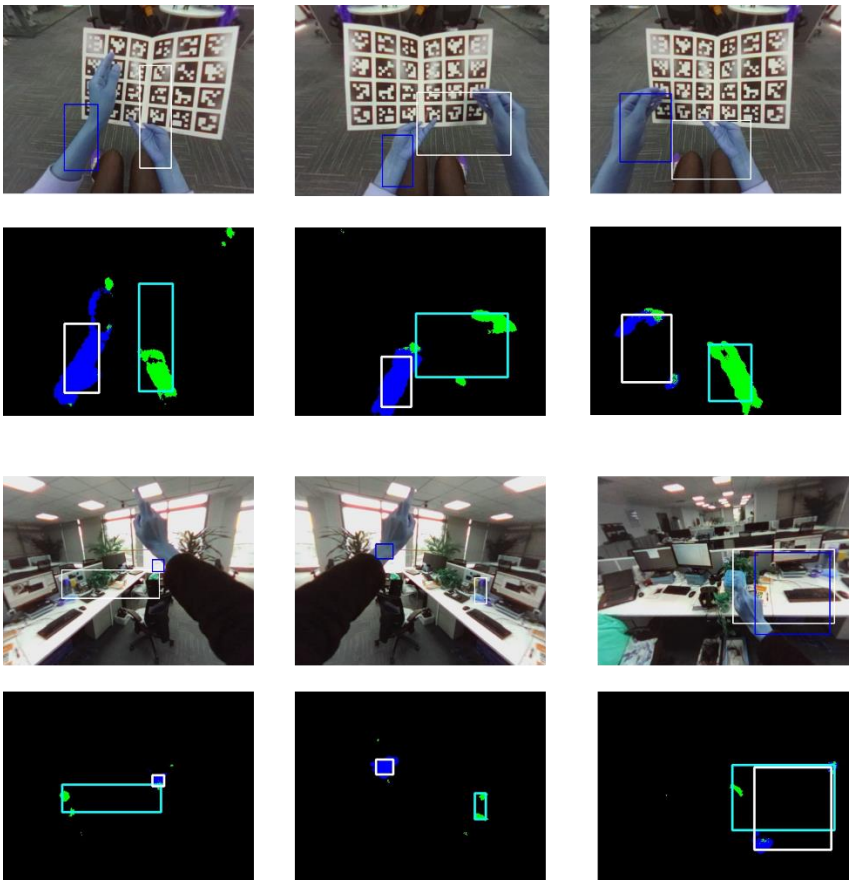

Figure 9. Result on HTC testing data

藍色 Bounding Box 為框住右手區塊部分，白色 Bounding Box 為框住左手區塊的部分。由上面六張圖的結果可以看出，我們在下面三張圖得效果比較不好，有時框框會辨認到雜訊並把它是為是手的部分。但在上面三張圖都能大概抓到手的部分。但還需要改良更好。

3.3. Bounding Box Sampling

我們利用 FCN model 切割好出來的手部區域，對其每個 pixel 都算成一個點，然後我們去 sample 這些點，展開變成 proposal box。然後我們存下這個 proposed box 裡面框到的區域，把這些區域丟入 CNN 三分類的分類器，運用機率找出這個區域是左手還是右手。以下是我們的實作。我們運用 HTC 本機的六張圖片做測試。

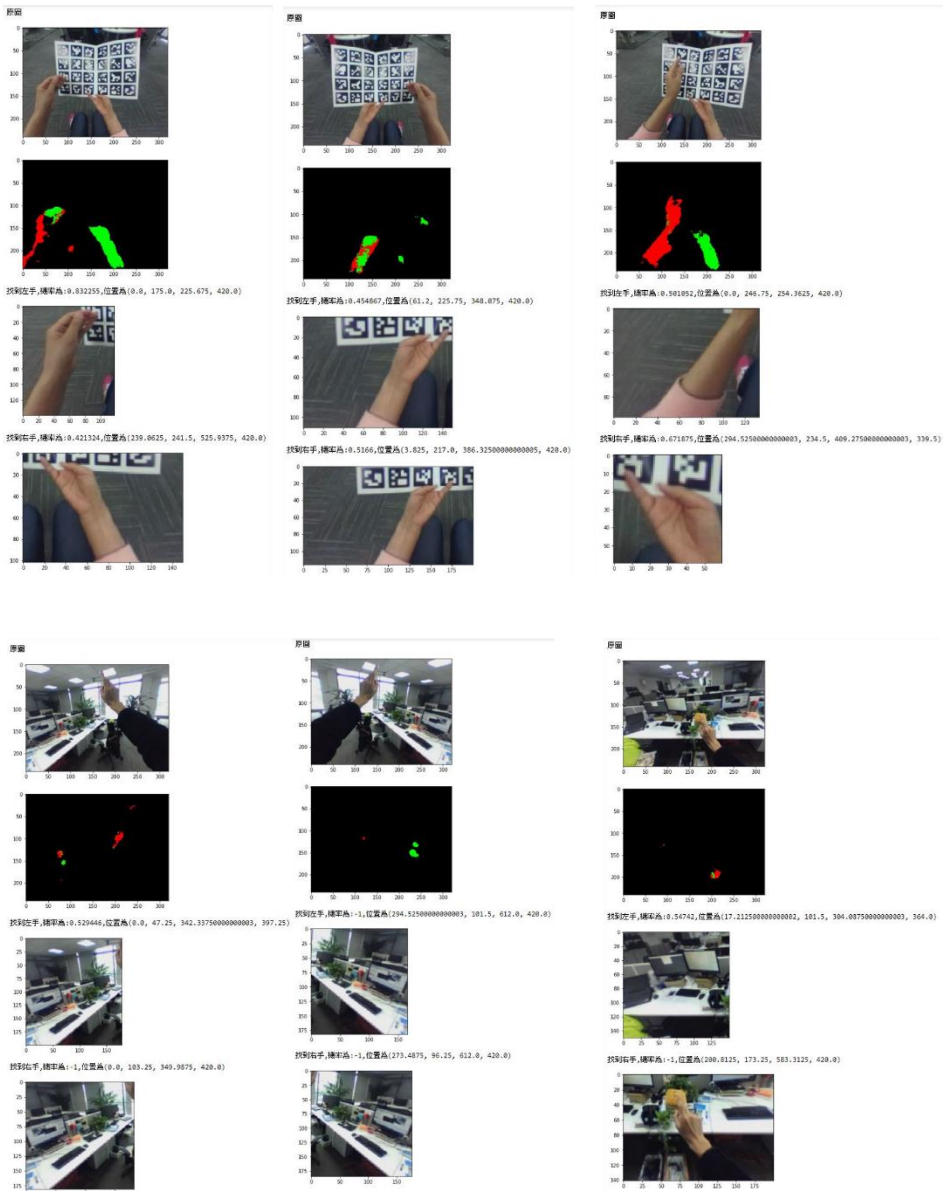

Figure 10. Result on HTC testing data

以上 Fig. 10. 做法利用這個簡單的三分類器，我們可以切出左右手的 Bounding Box。並且可以得知 Model 認為這是左手或右手的機率。所以如果機率太低，還可以將其剃除。

3.4. Regression Model

我們運用上述所說明到的演算法，在計算出 Bounding Box 之後，希望能夠讓我們所預測出的 Bounding Box 再精準一點。所以我們先利用 HTC 的 Training data 當中的 label 坐標先產生出一堆尺寸大小不一的切割圖，這些切割圖都是真實 label 座標對其做一些位移所產生出來的圖。到時希望能夠把這些圖丟進 model 然後產出逼近正確的座標。

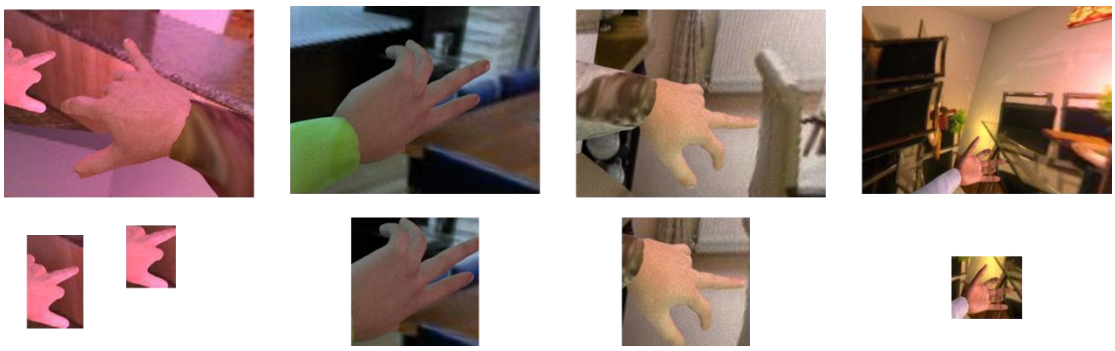

Figure 11. Segmented images on training data

以上 Fig. 11.圖檔為我們要送進 Regression Model 的圖片。送進 model 最後得出一個四維向量，就是 Bounding box 偏移的座標。

4. Conclusion

此次競賽實作我們運用改良過的 FCN model 去達成切割圖片取得手部區域。並且使用多種演算法來取得手部區域的座標。在實作當中一開始遇到 FCN model 訓練出來的模型太過 overfitting，導致抓不到手部的區域。所以我們後來改變訓練的 data 並且改變 learning_rate。重要的是訓練時的 epoch 也不需要太多，因為其實 100000 的 data 數量夠多。後來使 model 的效能比較好後，就能夠比較生成出好的手部切割圖片，接下來我們運用座標演算法取得手部區域的 Bounding Box。

因為有時取到的 Bounding Box 會受到雜訊影響，所以我們先把雜訊去除，在進行座標提取，效果明顯比之前好很多。再來我們運用一個 Regression model 去使我們預測出來的 Bounding Box 能夠更接近真實 label 座標。並且我們再使用另一個簡單的 model 訓練切割下來的 Bounding Box 圖片。分類這些圖片，預測 Bounding box 當中是左手還是右手，或者我們視為背景。此次競賽我們認為自己的實作還有很多地方可以改良，像是其實可以先運用 GAN 做 style transfer。把生成圖片轉變成真實圖片的風格。而此次競賽，平台上的操作如果可以寫得更仔細，還有平台能夠讓我們測試時跑更快的話，我們應該還有時間能夠表現更好。

Reference

1. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, Cham, 2015.
2. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.