

Vehicular Vision System

Final Project Report

即時人物追蹤



王彥儒 吳承翰 朱恩達

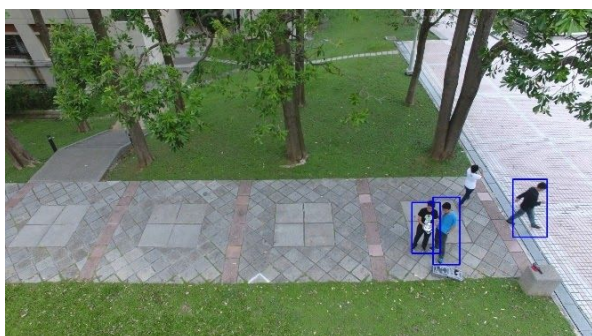
01.12.2020
18TH GROUP

Introduction

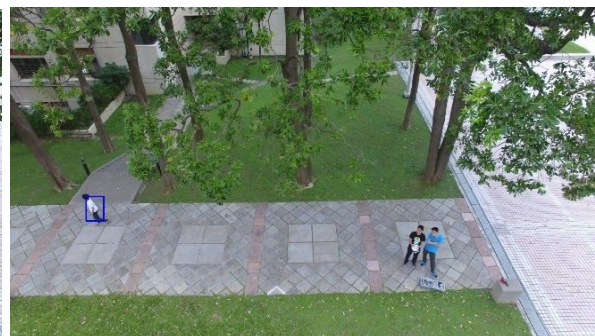
傳統固定式攝影機常受到地理環境因素限制，導致監控效果差。而無人機擁有高機動性、靈活性，可以主動變換監控位置與角度持續追蹤，克服物體遮蔽、視野死角、監控邊界。所以我們想利用空拍機進行人物偵測，進而追蹤人物，讓無人機自行移動追蹤，不需要額外人員操控。

Question & Discussion

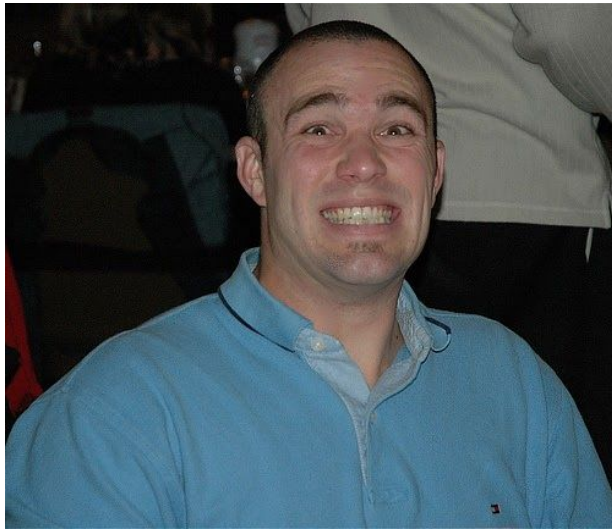
我們假設的第一個問題是空拍偵測與平面偵測的影像不同，無人機飛行高度介於8公尺到18公尺，在高空中的影像除了透視攝影造成人物歪斜現象外，其人物所占影像比例也會隨高度上升而降低。所以當我們使用YOLOv3進行偵測時，部分行人將無法準確偵測；此外，YOLOv3的資料集來自於Pascal VOC dataset 與 COCO dataset，這兩個資料影像的人物總比例較高，與空拍影像差異大。所以我們向製作自己的空拍資料集進行訓練完成偵測追蹤。但是後來實際操作Tello無人機，發現飛行高度上升至2公尺後，無人機會開始不穩定的飄浮移動，最後只能讓無人機高度設置在1.8公尺，所以並沒有額外訓練自行拍攝的空拍影像資料集。



高度8公尺空拍影像以YOLOv3偵測



高度8公尺空拍影像以YOLOv3偵測



Pascal VOC dataset 範例圖



COCO dataset 範例圖

Methodology

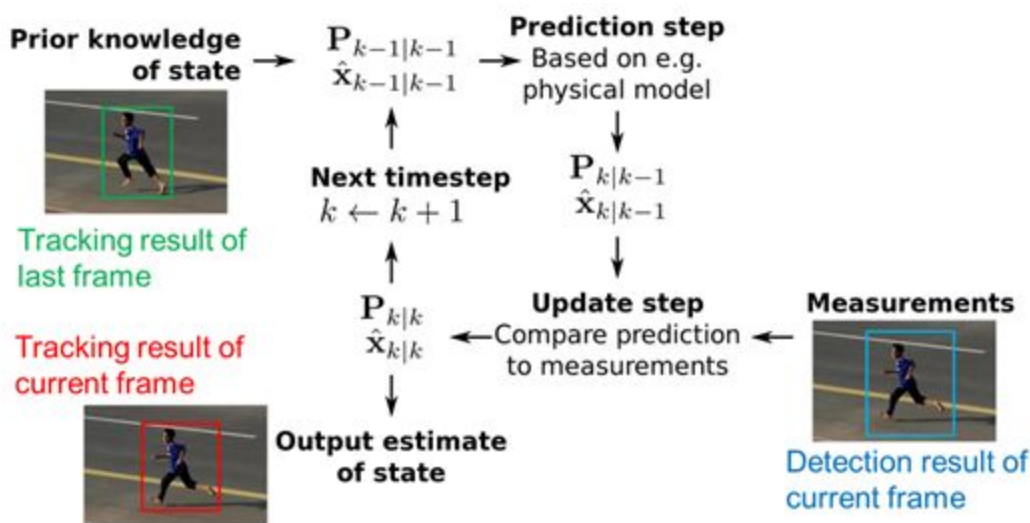
1. YOLOv3 人物偵測

我們所使用的模型為YOLOv3，並且利用作者Joseph Redmon在官網提供的權重進行偵測使用。該權重是由Pascal VOC與 COCO dataset訓練而成，我們所使用的608x608最大解析度版本，在COCO dataset上可以達到57.9mAP，以及20FPS。由於這次的計畫我們只針對人物追蹤，所以在偵測上只會單純偵測人物位置，其他類別的物品一概忽略。

2. Deep SORT 人物偵測

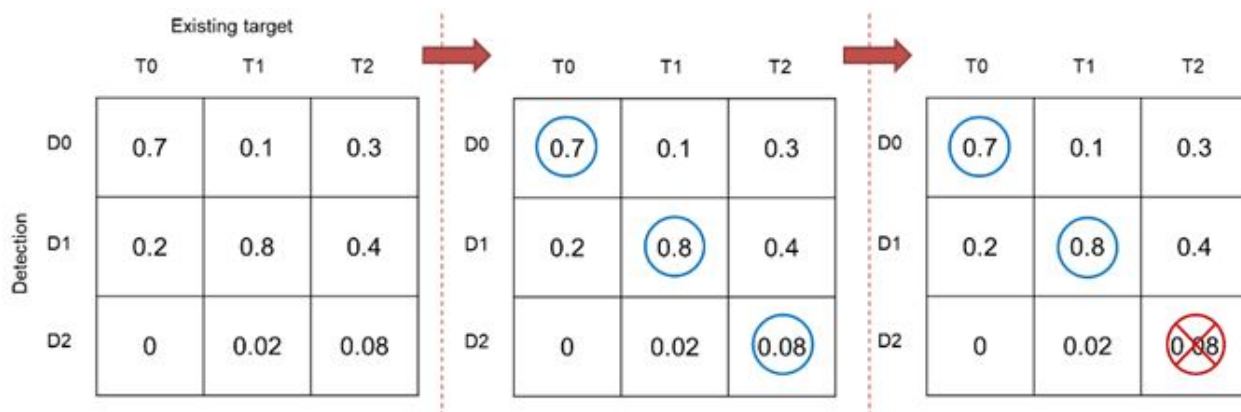
SORT全名為(Simple Online and Realtime Tracking)

- A. 第一步驟是進行人物運動估計，使用的方法為卡爾曼濾波器 (Kalman Filter)。簡單來說，我們會利用卡爾曼濾波從上一幀來預測目前狀態，再將預測結果與當前幀的資訊進行比對計算，最後再將總結果輸出。



卡爾曼濾波器追蹤流程圖

B. 第二步驟是多目標追蹤，當濾波器輸出結果後，我們會針對每個追蹤人物產生一個成本矩陣，這個成本矩陣是計算偵測邊界框與已存在目標預測邊界框之間的IoU所得。產生矩陣後，使用匈牙利演算法得到最佳任務分派解。最後去除掉IoU過低的指派。



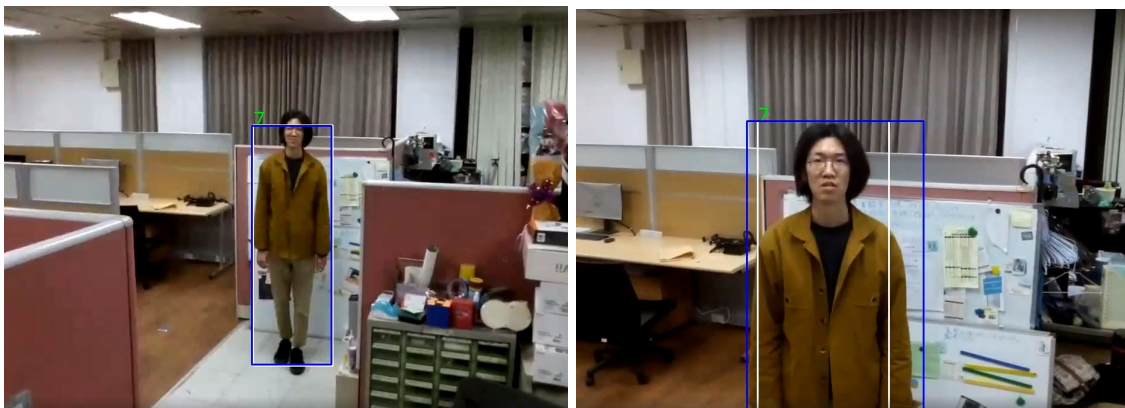
成本矩陣示意圖

C. 最後一步驟是關於追蹤身分刪除，我們設定的追蹤目標對大容忍度為150frames(當FPS為30秒時，可以保留5秒)，也就是說，當追蹤目標在五秒內被遮蔽、消失、誤判，在五秒內可以被重新辨

識為同一身分，但超過5秒後，追蹤目標將改變身分(ID Change)

3. Tello API控制

我們使用助教提供的Tello-Python API與 Tello進行連線操作。執行程式後無人機會在5秒後垂直起飛至離地面1.8公尺。前後左右追蹤是藉由YOLOv3所產生的BoundingBox框進行操作，Tello相機的影像解析度為960x720，所以當目標框高度小於500時，代表距離目標過遠，Tello會自動前行0.6公尺；相反的，當目標框高度大於600時，代表距離目標過近，Tello會自動後退0.6公尺。左右旋轉的方式是將畫面切成5等分，當目標偏離中心時，Tello會自動每次旋轉15度微調。如果15秒內不對Tello有操作或是追蹤目標時，Tello會自動降落。



Tello無人機鏡頭畫面

在操控方面其實是我們面臨最大的困難，當我們在資訊大樓1樓試飛時會發生無故飄移以及無法操控的問題，不排除是因為光滑地板反射還是網路連線干擾不穩等問題。但當地點換到3樓或是實驗室內又能正常操控。

Experiment

我們也針對YOLO+SORT的方法進行追蹤準確度與執行速度的相關實驗。我們使用UAV123 資料集 UAV是無人機的簡寫，資料內容都是用空拍機低空拍攝的影片，總共123部影片，但我們的實驗只針對人的追蹤，所以只使用了其中34部影片。因為目前沒有多目標追蹤的空拍資料集，所以只使用 UAV123，但其只給予單目標追蹤的GroundTruth BoundingBox。

實驗設備環境為：

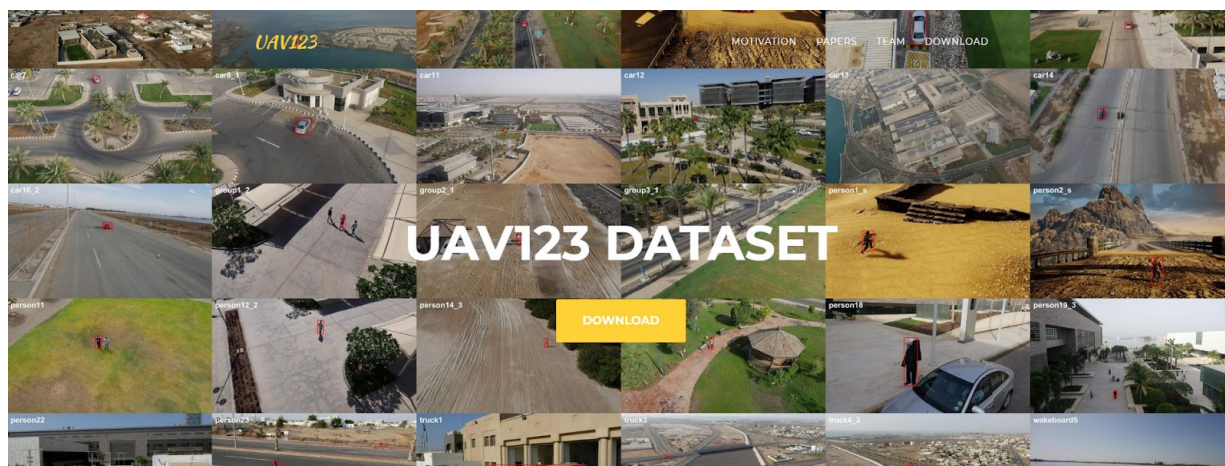
CPU: Intel i7- 6700

GPU: NVIDIA TITAN Xp x1

RAM: 16G

OS: Ubuntu 16.04

Language: Python 2.7

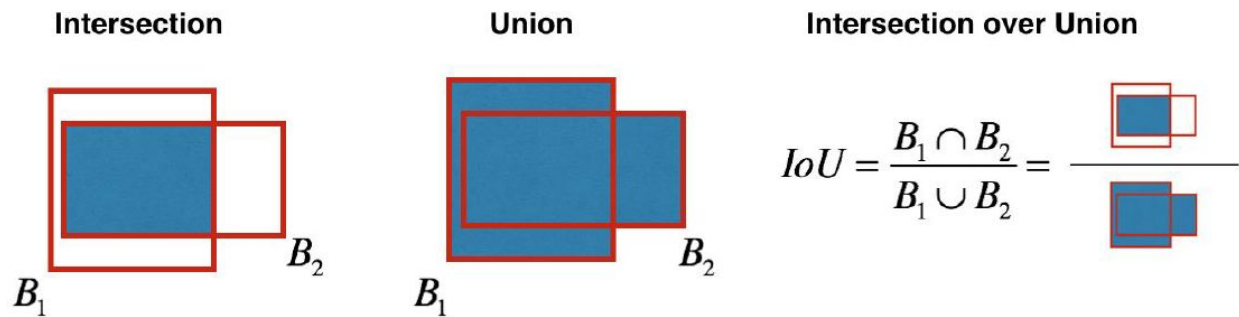


UAV123 資料集

Evaluation Metrics

追蹤評估的方式我們使用AOS(Average Overlap Score) 所有幀的平均重疊率

$$AOS = \frac{1}{N} \sum_{k=1}^N IoU(bb_{tr}^k, bb_{gt}^k)$$



AOS 與 IOU 計算方式

N是影片的總幀數

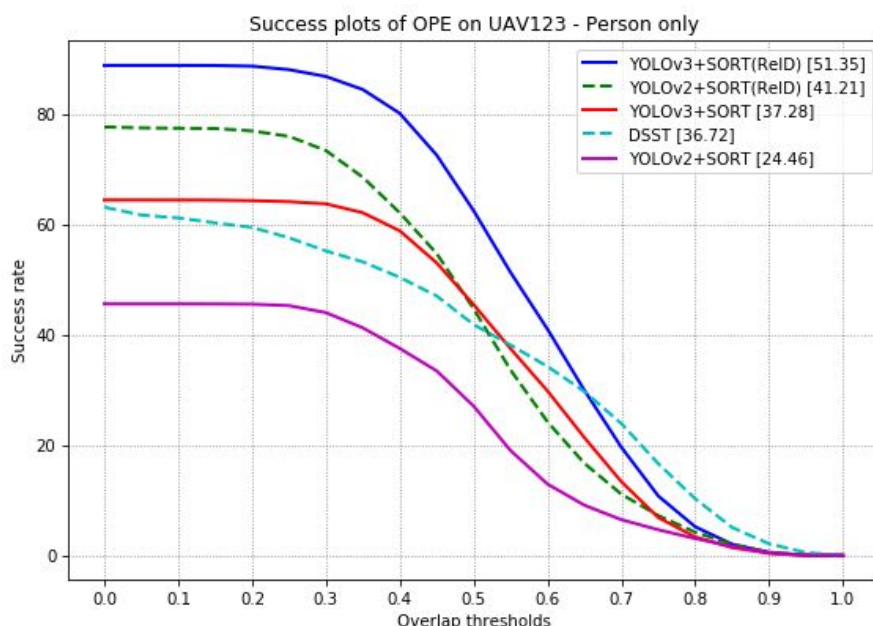
bb代表bounding box

k代表第k幀

tr是我們追蹤的結果

gt為GroundTruth

Results (Comparison)



UAV123 人物追蹤圖表

x軸代表IOU的門檻值，當偵測框與GT框重疊超過一定比率時才會算偵測成功，y軸為Success plots。線下的總面積為AUC (Area under curve)。我們可以發現YOLOv2 到 YOLOv3 AUC從24.46 提升到37.28，可見偵測效果還是對追蹤有很大的影響。

ReID是為了避免短時間遮蔽或是身分變換等問題所採用的方法，我們這裡採用的是顏色外觀特徵，比起HOG (Histogram of Oriented Gradients)計算更有效率且容易實作。加入ReID後，YOLOv3的AUC還可以再提升(37.28 -> 51.35)，而YOLOv2加入後能超越原本單純YOLOv3+SORT的表現。

這裡加入比較DSST方法，來自於"Accurate scale estimation for robust visual tracking" BMVC 2014，主要使用傳統影像處理的多個濾波器進行追蹤。DSST在門檻值為0.8時能超越其他方法的原因在於它能在第一幀目標出現就給予偵測框，比其他深度學習方法要迅速。

Conclusion

我們使用影像偵測融合影像追蹤的深度學習演算法，針對Tello無人機所接收的影像，進行處理判斷，進而讓無人機自動操作行駛。另外也比較不同方法之間在空拍追蹤上的準確度與速度。

未來希望能改善的部分有無人機的細部操作移動，以及偵測追蹤的實際廣泛應用。

Demo Video

[NCTU-Vehicular-Vision-1](#)

[NCTU-Vehicular-Vision-2](#)

GitHub

[chiha8888/NCTU-Vehicular-Vision](#)

REFERENCES

1. [nwojke/deep_sort: Simple Online Realtime Tracking with a Deep Association Metric](#)
2. [qqwweee/keras-yolo3: A Keras implementation of YOLOv3 \(Tensorflow backend\)](#)
3. [Qidian213/deep_sort_yolov3: Real-time Multi-person tracker using YOLO v3 and deep_sort with tensorflow](#)
4. [博碩士論文行動網](#)
5. [YOLO: Real-Time Object Detection](#)
6. [Accurate Scale Estimation for Robust Visual Tracking](#)

Team Contribution

Tasks	Contributors(%)
Literature survey	王彥儒 (33%) 吳承翰 (33%) 朱恩達 (33%)
Approach design	
Approach implementation	
Report writing	
Slide making and oral presentation	