國立清華大學 電機工程學系

實作專題研究成果報告

A Study on Active Learning for Improving Object Detection Model in Home Care System Using Limited Amount of Data

利用主動學習在有限資料下優化居家照護系統的目標檢測模型之研究

專題領域：系統組

組　　別：A85

指導教授：孫民

組員姓名：劉亦傑、謝霖泳

研究期間：109年7月1日至110年5月底止，計10個月

Abstract

In recent years, with the development of technology and medicine, the population of the elderly keeps increasing. The problem of taking care of the elderly living alone gradually arises. Therefore, our lab founded CarePLUS Group with the aim to develop a smart home care system, monitoring the action of the elderly and detect whether they are in danger using computer vision. Besides, the home care system will remind them to take medicine and do exercise regularly. By doing so, we can accompany the elderly living alone using the power of AI instead of labor.

In CarePLUS Group, we are responsible for preprocessing training data, and our goal is to improve the performance of object detection model by finding out frames that are more valuable. With a limited amount of data, we construct "Active Sample" algorithm by Active Learning to achieve this goal. By the videos recorded by the camera and the prediction result from the detection model, it is possible to strengthen some parts that the model tends to misjudge.

The ultimate goal of Active Sample is to enhance the ability of our model to distinguish humans. In this study, we observe the effect of our algorithm by building Detectron2[11] developed by Facebook AI.

The result of our experiments reveals that sampling the frames merely with the change of light is not sufficient. Thus, we take the area of bounding boxes into consideration. The accuracy of revised method increases by 47% in comparison of the baseline model. From the perspective of efficiency, the number of frames needed for our revised algorithm is only 2.5% of the original one, which greatly reduce the cost of data labeling.

In conclusion, we can improve our model efficiently and successfully by finding out the frames using Active Sample. In the future, we expect that our algorithm can be not only applied in offline use but in real time detection.

摘要

近年來，在科技和醫療的進步之下，社會中老年人口比例漸增，如何妥善照護「獨居老人」的問題也逐漸顯現，因此我們的實驗室以智慧居家照護為宗旨成立了"CarePLUS"團隊，決定以計算機視覺(Computer Vision)領域內的技術來嘗試著即時監測老人的行動，並進一步判斷他們是否有危險、有狀況；抑或是在正常情況下，去陪伴老人、提醒其準時吃藥、適時起來動一動身體等等，在不使用勞力的情況下，倚靠人工智慧的力量使其能隨時隨地陪伴著獨居老人們。

在 CarePLUS 團隊中，我們是在整個大型專案中擔任對訓練資料前處理的工作，也就是偏向如何藉由挑出更有價值的資料，來讓目標檢測模型在有限資料量下去適應新環境，或是達到更好的效能。因此，我們想利用「主動學習」的概念去建構出自定義的 Active Sample 演算法，原理上利用攝影機錄下的影片以及模型預測的結果根據此演算法去做處理，就能挑出理論上模型比較弱的地方，以針對不足去做補強。

Active Sample 演算法最終目的就是要能夠讓模型更有偵測人類的能力，因此實作上我們設計許多方法去篩選，來嘗試找出所謂重要的影格，將這些影格放回模型去訓練，則可以觀察演算法的效果。模型設計上，我們是利用 Facebook AI 所設計的 Detectron2[11]框架來設計、實作用來評估效能的模型。

我們的實驗結果說明了，藉由單純的光影變化去做篩選可能會有不足，但若加入 bounding box 面積進一步篩選，則最終在準確度上相對於 baseline 模型能有 47%左右的大幅提升；而若從效率的角度來看，利用我們的演算法可以只需要原始影片的 2.5%的影格就能達到效果更好且大大節省了「標記資料」的花費。

經過我們的努力研究，可以讓我們利用自行設計的 Active Sample 演算法去更有效地找出目標檢測模型較疲弱之地方，來盡可能高效率地去提升模型，且所需要的輸入資料也更少，我們認為未來若能進一步到實時(Real time)的程度，也就是一邊收集資料的同時就去分析資料的重要程度，將那些對模型更有提升效果的資料放回模型去訓練，使得目標檢測模型學習的效率提升。

目錄

一、前言

在科技日益發達的今天，人們的平均壽命越來越長，老年人口的比例也越來越多，因此「獨居老人」的照護問題逐漸浮上檯面。衛福部統計每六個年長者就有一個年長者有跌倒的經驗，但因為許多年輕人平常都在外地工作，無法照顧家中的爺爺奶奶，以致於長者們發生緊急意外時，無法提供及時的協助而造成遺憾。

但不管是請看護照顧老人，抑或是把他們送到養老院，這些成本都很高，每個月都需要付出一筆龐大的費用。如果單純用監視器觀察老人的行為，可能也要有人 24 小時不間斷地盯著看，否則意外發生時仍難以提供及時協助。於是我們的實驗室以「智慧居家照護」為宗旨成立了"CarePLUS"團隊，在使用者家中裝上鏡頭，配合深度學習的技術即時監測老人是否有危險，並以聲音互動的方式向老人詢問是否有需要幫助。主要的特點有每日活動量與環境偵測、日常作息偵測、居家意外偵測以及長期資料分析。舉例來說，當偵測到老人跌倒或滑倒時，趕快詢問他是否需要幫助；當偵測到老人久坐一整天時，也會發出聲音提醒他起來動一動，並傳送通知給老人的子女讓他們充分了解長輩的健康狀況，諸如此類的提醒目的都是想辦法維持老人的健康。

目前大部分人類辨識的神經網路模型都是建立在 front view 或 side view 之下，現行的許多老人監控系統亦然。和一般老人監控系統的不同點在於，我們使用的鏡頭安裝於使用者家中的天花板上，也就是用「魚眼鏡頭」(fish eye)的視角來偵測老人的行動。

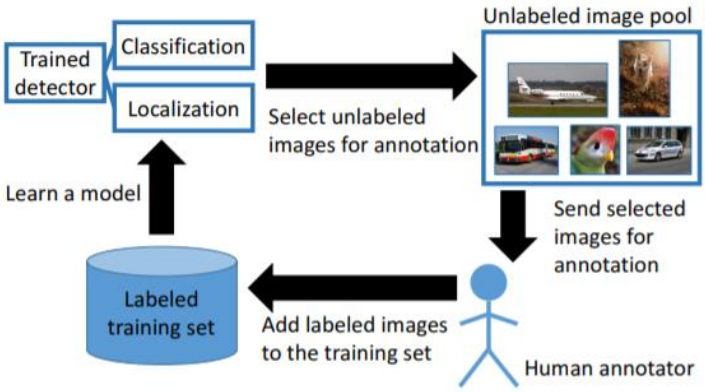
然而，魚眼視角雖然可以比較沒有死角地觀察老人的行為，但也增加了目標檢測模型(object detection model)訓練上的難度，因為，魚眼視角的資料並不普及，沒有足量資料提供我們訓練模型，因此實作上有許多需要克服的問題。

除了角度的問題之外，另一個問題就是當我們把相機換到另一個新的環境時，我們希望它可以在短時間內迅速的提升效能或準確度，並能一邊收集資料一邊進行學習，於是我們參考了[1][2][3][4][5]的作法，並融合這幾篇論文中的概念來發想。利用主動學習(Active Learning)讓模型進入新環境後，可以在更短時間內或更少的資料量下有顯著的進步。而主動學習就需要從現有的資料中，挑出模型比較不會的地方，讓模型能加強學習效率，因此我們定義了 Active Sample 的演算法，去挑出現有資料中的的偽陽性(false positive)和偽陰性(false negative)的影格，並重新將這些相對重要的影格丟回模型中，因為對模型而言連續的影片中並不是每張影格都能對模型有顯著幫助，特意挑出影格就能在較少的資料量下，以較高的效率去增強模型尚未學好的部分。

二、原理分析與系統設計

2.1 原理和演算法分析

在許多機器學習的領域範疇中，最珍貴的往往就是標記資料所消耗的人力。為了節省這些曠日廢時的人力成本，這次專題中，我們主要用到的方法為主動學習(Active Learning)，它的精神就是挑出模型比較學不會的地方，經過人工 label 之後，讓模型去學習，圖一是從[3]中擷取出來的，說明了主動學習的過程。藉由主動學習，我們就不用將所有資料都進行 label，也就節省了許多人力成本。至於挑出模型比較學不會的部分，我們使用的方法稱為"Active Sample"，也就是「主動抽樣」，挑出影片中模型容易誤認的 FP 和 FN，讓模型加強學習。



圖一：主動學習流程圖

False Positive(偽陽性，以下簡稱 FP)和 False Negative(偽陰性，以下簡稱 FN)是兩個我們主要會探討的問題，因為這些 FP 和 FN 就是模型比較弱、比較容易犯錯的地方，也就是在 Active Learning 的過程中，需要被 sample 出來給模型學習的地方，因此先定義 FP 和 FN。

FP 就是模型原先預測結果認為有人(positive)，但實際上卻沒有人的情況。首先我們的演算法假設：「人會動，所以人才會造成動作(motion)，而物品都是靜止的。」因此「動作」在演算法判斷有沒有人的過程中扮演著關鍵的角色，至於實際上運用的方法主要是看這個影格的 motion 和前面連續 500 個影格的平均去比較，利用每兩張影格間 pixel 的差異，只要某影格前面連續很多個影格都是幾乎靜止的，但此時突然有很大的 motion，演算法就會判定此影格有人(positive)。

但這樣便會有很多問題產生，比如說當一陣風吹來，窗簾飛起來，這時候因為在先前的影格之中，窗簾都是靜止的，而此時突然產生了很大的動作，或像是電視播放、電暖器或電風扇在運轉等情形，演算法都會將這些情況誤判為有人。此外，我們觀察到光影突然的變化所造成的 motion 也很容易產生誤判，

像是房間的燈突然亮起，演算法就會判斷成有人。諸如此類實際上沒有人卻被演算法誤判成有人的情況，稱為偽陽性(FP)。

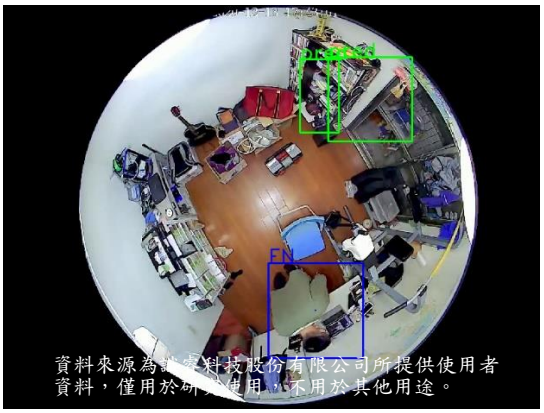
相較於 FP，FN 就是模型認為沒有人，但實際上卻有人的情況。會造成 FN 的主要原因就是人的 motion 太低，導致模型把人和背景融為一體了。因為 CarePLUS 的宗旨就是照顧老人，而老人常常會在電視機前面久坐一整天，這個期間因為幾乎都沒有 motion，因此模型就會認為這是沒有人的情況，稱為 FN。

		Predicted Class: P	Predicted Class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

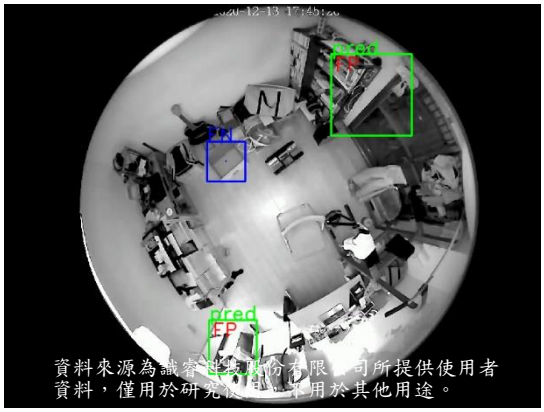
圖二：FP 與 FN 的標準定義

然而，在觀察 sample 出來的 FP、FN 資料之後，我們發現 FP 的資料中，仍有一些是真的有人的，同理在 FN 的資料中，有一些是真的沒有人的。然而，因為這些 sample 出來的資料都需要經過 label 之後重新給模型學習，當 FP/FN 本身的精確度就不高時，就無法彰顯 Active Sample 的成效，也會徒增 label 資料的人力成本。

綜上所述，我們認為 FN 的問題比較需要先被解決。FN 是實際上有人，但模型的 prediction 卻認為沒有人的情況。對我們而言，畫面中有老人但卻沒有被偵測到的這個情況比較嚴重，因為這樣就無法在他發生意外時提供即時援助，這也是模型比較需要學習的地方，因此相較於將物品誤判為人的 FP，我們想先著手解決 FN 的問題。因此，本專題的主要目的就是從原始資料中找出更有價值的 FN。



圖三：符合 FN 條件的照片



圖四：不符合 FN 條件的照片

圖五：不符合 FN 條件的照片

根據我們對 FN 的定義，理論上 FN 的影格中應該都要有人（如圖三），而沒有人（如圖四、圖五）的影格就不應該被認為是 FN。觀察資料後發現，不應該出現在 FN 資料中的那些 frame，普遍都是框到一個很小的物品，可能因為箱子側邊產生了一瞬間的光影變化（如圖四）或手機的螢幕突然亮起來（如圖五）等情況，讓模型誤判成 FN，因此我們的目的就是將這些沒有框到人的 FN 給去掉。下面三張範例的藍色框處即為 FN 的位置。

根據這個性質，我們希望做進一步的改進，改善原始 Active Sample 演算法，多使用「motion 的 response 在圖片上造成的大小」來判斷這張 FN 是否 sample 正確的標準，也就是上面三張圖中「藍色框的面積大小」。理由是如果這個框真的有框到人，在圖上所占的面積應該會遠比箱子或手機等物品占的面積還要大許多，所以如果這個藍框的面積太小，我們就認為此 FN 可能是一個錯誤的 FN。因此，我們嘗試去篩選這些原始 FN 的資料，在改良版本的演算法設計上多判斷一項面積閾值，判斷藍色框面積符合前述條件的圖片記下來，並在後續的討論中與原始版本的演算法比較其效果差異。

Algorithm 1: Active Sample Algorithm

Input: A video sequence S, prediction result produced by model P
Output: Valuable frames V
Parameter:
ϕ_h, θ_h for FP, FN human thresholds respectively;
ϕ_{opt}, θ_{iou} optical threshold, iou threhold respectively;

$F_P, F_N \leftarrow$ empty list; `// Store FP, FN results`
while S *is not over* **do**
  $F_i \leftarrow$`TakeFrame(`S`)`; `// Take a frame among S`
  $P_i \leftarrow$`FindPred(`P, F_i`)`; `// Find prediction box from` F_i
  $B_i \leftarrow$Apply background subtraction on F_i;
  $O_i \leftarrow$Analyze optical flow from F_i; `// for False Negative use`
  foreach *element* p *of* P_i **do**
    $M_{box} \leftarrow$ Find motion box by B_i;
    Find motion region ratio R_{motion} through B_i;
    if $R_{motion} > \phi_h$ **then**
      Append P_i to F_P;
      Draw bounding box information on V;
    end
    Save V as a figure and put V back to training set;
  end
  foreach *element* q *of* O_i **do**
    $M_{box} \leftarrow$ Find motion box by B_i;
    Find motion region ratio R_{motion} through B_i;
    Find optical flow region ratio $R_{optical}$ through q;
    if $R_{motion} > \theta_h$ *and* $R_{optical} > \phi_{opt}$ **then**
      $R_{iou} \leftarrow$ Find iou result between motion box and P_i;
      if $R_{iou} > \theta_{iou}$ **then**
        Append M_{box} to F_N;
        Draw bounding box information on V;
      end
    end
    Save V as a figure and put V back to training set;
  end
end

Algorithm 1 是我們 Active Sample 最初的算法分析。首先，我們會先定義整個系統，輸入為一段影片和原始模型在這段影片上的每張影格所產生出的預測結果；輸出則是有價值、模型所需要的影格。另外還需要再去定義出幾項可調的參數，包含了 FP, FN 判斷為人的閥值、光影與 motion 分析與 Intersection of Union 的閥值等等，供演算法使用。

接著便是演算法的主體，對於影片中的每一個影格，都去進行以下的分析，先取出一張影格後，利用[6]提出的背景分離技術(background subtraction)來去除雜訊，以及利用 OpenCV 分析這張影格的「光流(Optical flow)」，光流是用來描述相對於觀察者的運動所造成的觀測目標、表面或邊緣的運動。最後再由此影格去找到其對應的模型輸出結果，如此一來我們演算法分析上所需要的資訊都準備完畢。

演算法前半部分所做的是重要 FP 影格的探尋，是較為單純的架構，因為前面所提到說，對於老人照護而言我們將物品寧可多誤判人，也不希望說有人卻沒被偵測到，故 FP 影格的部分並非我們著重的點，此段演算法架構上也就

相對簡單，針對這張影格上所有的預測結果，利用 motion box 這項閥值以及 background subtraction 的資訊來做分析，就能達到不錯的效果。

接著來到演算法後半部分，則是我們主要想優化的地方。方法上我們與 FP 有些許不同，不是針對每一個影格的預測結果，而是如 Algorithm 1 中的第 17~19 行，去建立光流法和 motion box 間的關係。接著判斷有無大於預先調定的參數，即 FN 是否為人和光影變化是否充足這兩項指標。如果有達到前述兩條件，則這組 FN 就可被稱為「FN candidate」，下一步就是去計算它們的 motion box 和模型原始預測結果之間的 IoU(Intersection of Union，即預測位置和實際位置的重疊程度)，並藉由此 IoU 值去檢查說有無大於我們演算法所設定的 IoU 閥值，此條件須也滿足才能將其選為真正有價值的 FN。

Algorithm 2: Active Sample Algorithm (including area)

Input: A video sequence S, prediction result produced by model P
Output: Valuable frames V
Parameter:
ϕ_h, θ_h for FP, FN human thresholds respectively;
$\theta_{opt}, \theta_{iou}, \theta_{area}$ (optical, IoU, area) thresholds respectively;

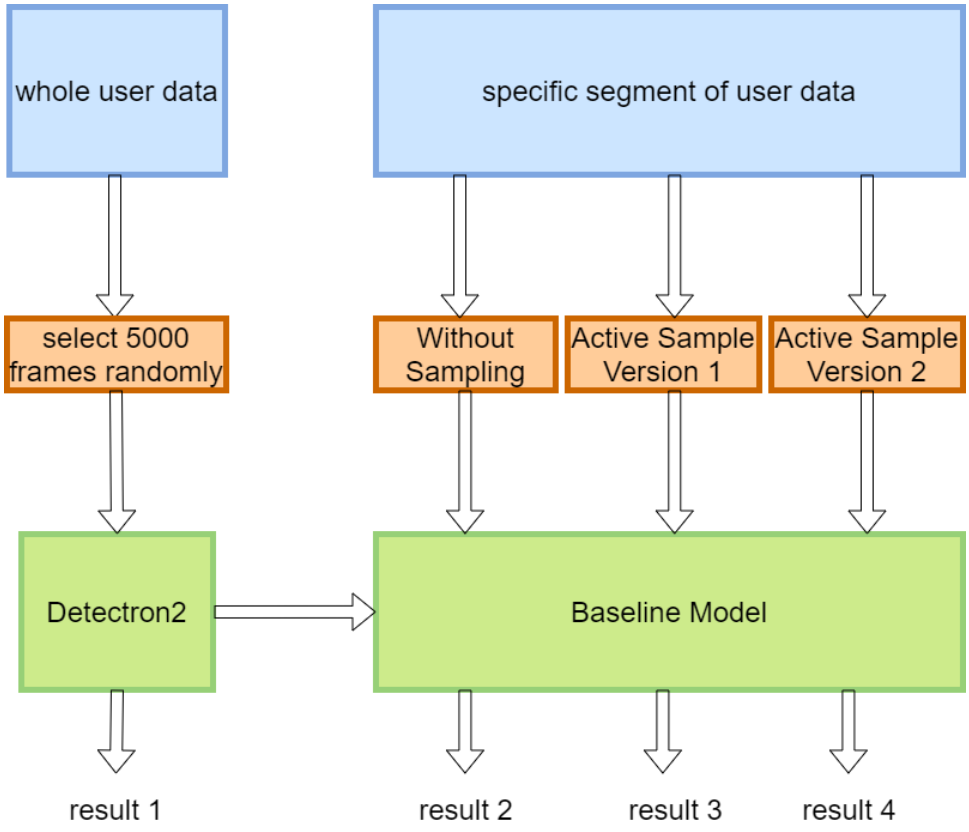
```
1  F_P, F_N ← empty list; // Store FP, FN results
2  while S is not over do
3      F_i ←TakeFrame(S); // Take a frame among S
4      P_i ←FindPred(P, F_i); // Find prediction box from F_i
5      B_i ←Apply background subtraction on F_i;
6      O_i ←Analyze optical flow from F_i; // for False Negative use
7      foreach element p of P_i do
8          M_box ← Find motion box by B_i;
9          Find motion region ratio R_motion through B_i;
10         if R_motion > φ_h then
11             Append P_i to F_P;
12             Draw bounding box information on V;
13         end
14         Save V as a figure and put V back to training set;
15     end
16     foreach element q of O_i do
17         M_box ← Find motion box by B_i;
18         Find motion region ratio R_motion through B_i;
19         Find optical flow region ratio R_optical through q;
20         A ←Calculate area of M_box;
21         if R_motion > θ_h and R_optical > θ_opt and A > θ_area then
22             R_iou ← Find iou result between motion box and P_i;
23             if R_iou > θ_iou then
24                 Append M_box to F_N;
25                 Draw bounding box information on V;
26             end
27         end
28         Save V as a figure and put V back to training set;
29     end
30 end
```

Algorithm 2 是我們根據 Algorithm 1 去做進一步的改良。整體而言與 Algorithm 1 相當類似，輸入輸出也相同，包含了 FP, FN 判斷為人的閥值、光影與 motion 分析與 IoU 的閥值等等，供演算法使用。

至於演算法的主體上，與 Algorithm 1 不同的部分在於 FN 的優化，根據我們對於原始 FN 輸出結果之觀察，可以發現到如果僅使用光影變化、motion box、IoU 等指標進行分析，則會取樣出許多畫面內甚至沒有人存在的影格，

因此，除了針對光流法產生的結果來去和 background subtraction 的資訊去建立光流法和 motion box 間的關係，我們此處還會再去計算 FN box 的面積當作篩選之參考。接著，判斷有無大於預先調定的參數閾值時，額外多加入一段 FN box 的面積的閾值來進行篩選，如此一來代表篩選條件更為嚴格，有機會能篩選出更關鍵之影格，也意味著能運用更少張數的影格來讓模型學到相仿的效果。

2.2 系統設計與目標檢測模型



圖六：系統流程圖

圖六是我們整個系統的流程。有了前述 Active sample 演算法後，我們就需要自行訓練出目標檢測模型來進行評估。在這次的專題中，我們所挑選的是 Faster RCNN[7]，背後是基於 ResNet-101[8]的架構，並加上[9]FPN(Feature Pyramid Network)來比較優秀地提取出不同 scale 下的特徵。雖然原始的 Faster RCNN 論文中並無使用 FPN 這項技巧，但在速度和準確度 trade off 之觀點下，FPN 是能有相當不錯的表現，因此選用 FPN 的架構來進行實作，本文將稱此為 Faster RCNN R101-FPN 之模型。

資料部分，我們另外向 CarePLUS 團隊取得在使用者家的內部資料進行訓練，並非網路上公開抓取的 dataset。好處是資料就是真正使用者家的實際環境，能夠更貼近應用面；但缺點即資料量相對較不充足，且實際攝影機錄製的狀況也不穩定，有時候影片會幀數過低導致影片模糊。

我們的模型是根據前述所提之 Faster RCNN R101 架構，去實作在 Facebook AI 利用 Caffe2 所開發之「Detectron 2」框架上，因為目的是要測試看看 Active Sample 演算法的效果如何。

實作上會先載入已經在 COCO dataset 上進行過一輪預先訓練(pre-train)的 Faster RCNN R101-FPN，但經過 COCO 訓練並不會有對於魚眼視角有足夠的辨識能力，因此會先利用部分魚眼資料進行 pre-train，來讓模型先能夠對於魚眼視角的狀況能有基本的辨識能力，也就是建立出一個屬於我們的 baseline 模型。

資料集的建構上依照正常情況下去建立「訓練集(training set)」、「驗證集(validation set)」、「測試集(test set)」。而 pre-train 的部分，我們的影片格式每秒有 6 張影格，因此經過對於資料集的總數來評估，認為實作上能設計 training set 有 5000 個影格(也就是約 15 分鐘的影片)，validation set 則是 1000 個影格，test set 也是 1000 個影格，這樣的數量足以讓我們進行研究、討論。我們用來訓練的資料為 30 分鐘的影片，共分成以下三種情形。

1. 直接將 30 分鐘影片當做新的訓練集放回 baseline 訓練
2. 訓練在本影片經過原始之「Active Sample」演算法所挑選的影格
3. 訓練在本影片經過改良後之「Active Sample」演算法所挑選的影格

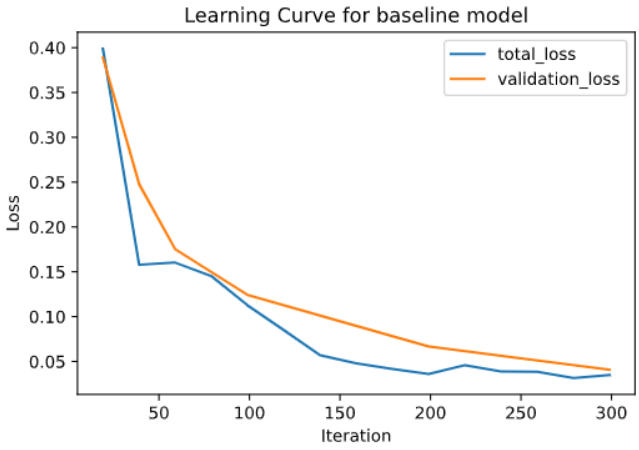
以上這三部分，都將會在 baseline 模型上額外去訓練 200 個 iterations，並將於後續實驗結果的部分來和 baseline 模型（也就是總共四種情況）去比較、探討。

三、實驗結果

我們的實驗結果主要將會分三階段進行分析：

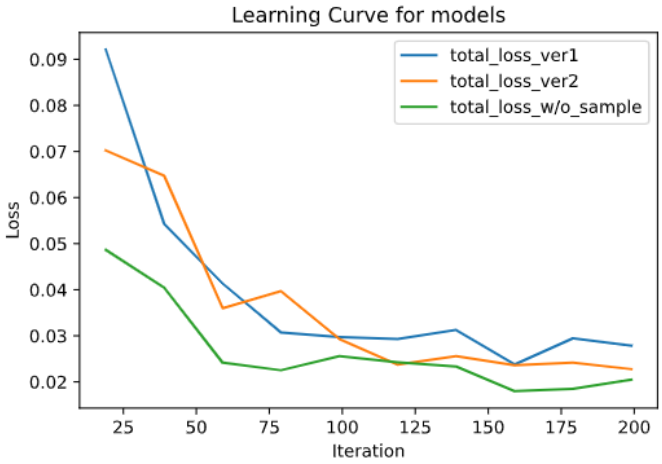
1. 針對訓練時 loss 的收斂情形，以判斷模型訓練的夠不夠好
2. 整理 AP(average precision)的表格來去定量描述我們的成果
3. 視覺化模型的預測結果，來呈現定性上的成果

3.1 模型 loss 收斂情形



圖七：baseline model 的 loss 收斂情形

圖七是我們把 baseline 模型訓練和驗證時的 loss 記錄下來，並描繪成上圖的學習曲線。可以觀察到兩項 loss 都是持續下降且收斂的，讓模型持續能降低整體的 loss。除此之外，可以觀察到在訓練末期，validation_loss 是很貼近於 total_loss 的，這也就意味著模型並沒有遇到 overfitting 的狀況，也就是 validation_loss 持續降低。300 個 Iterations 是可讓模型收斂到相對低的 loss。但可以觀察到縱軸在最後收斂時的 loss 約略落於 0.05 左右，此值將會於後續的訓練中降低到更理想的值。



圖八：根據 baseline model 進一步訓練的各方法 loss 收斂情形

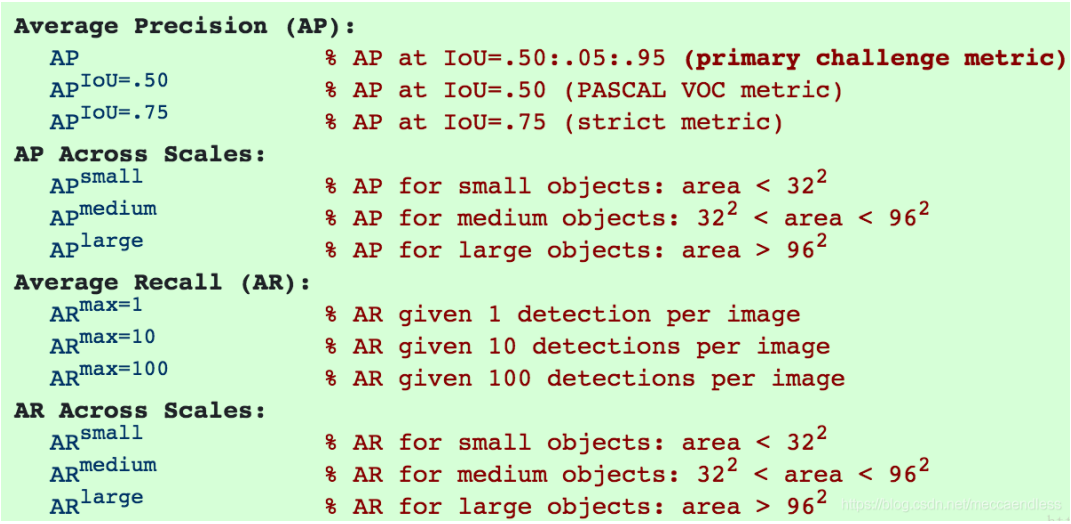
圖八是在 baseline 底下利用我們前述所整理之三種方法多訓練 200 個 Iterations 的學習曲線。其中藍線代表經過原始 Active Sample 處理資料進行訓練的模型，可以看到 loss 收斂至 0.03~0.04 之後就趨於平緩；接著，橘線是我們經由改良過後的 Active Sample 去處理資料進行訓練的模型，可以看到模型收斂情形比起藍線來得更好，loss 收斂的效果更好了。

最後，綠線則是完全沒有 sample 將整份資料丟進去訓練的效果，可以發現儘管模型 loss 似乎收斂到三者之中最低，但根據表一（附於 3.2 定量分析中）可以發現效果並非最好。我們推論可能是將整份資料不經篩選直接丟進去訓練帶有太多模型不需要的資訊，導致誤判發生的機率提高，以至於最後的 Average Precision 表現較差。

3.2 定量分析(Quantitative Analysis)

實驗結果將會討論以下四種情況：

1. Baseline: Baseline 模型的 performance
2. Without sampling：將 30 分鐘影片當做新的 training set 放回 baseline 訓練
3. Version 1：訓練在經過原始之「Active Sample」演算法所挑選的影格
4. Version 2：訓練在經過改良後之「Active Sample」演算法所挑選的影格



圖九：AP、AR 之定義（資料來源[10]）

首先，我們先針對幾項 AP、AR 指標做簡單介紹，這幾項指標是與著名的 COCO dataset 有相同的定義，如圖九所示。

由於我們改進的 Active Sample 演算法中，已經將 bounding box 面積過小的情況去除了，所以 small object 的指標無法被定義。除此之外，AR 的部分，因為 COCO 預設下設定 AP 就是 max detections = 100 的 AP，所以為了方便討論，主要就選擇 AR(max=100)當作我們的 AR 指標，AR 的部分同樣地也不考慮 small object 的影響。故我們最後挑選了 AP, AP50, AP75, APm, APl, AR, ARm, ARl 做比較。

	Baseline	Without Sampling	Active Sample (ver.1)	Active Sample (ver.2)
AP	0.493	0.699(+41.8%)	0.661(+34.1%)	**0.727(+47.5%)**
AP_{50}	0.740	0.950(+28.4%)	0.915(+23.6%)	**0.974(+31.6%)**
AP_{75}	0.676	0.914(+35.2%)	0.886(+31.1%)	**0.936(+38.5%)**
APm	0.483	0.685(+41.8%)	0.649(+34.4%)	**0.711(+47.2%)**
APl	0.681	0.810(+18.9%)	0.781(+14.7%)	**0.853(+25.3%)**
AR	0.709	0.754(+6.3%)	0.732(+3.2%)	**0.779(+9.9%)**
ARm	0.704	0.746(+6.0%)	0.719(+2.1%)	**0.769(+9.2%)**
ARl	0.748	0.811(+8.4%)	0.824(+10.2%)	**0.855(+14.3%)**
Extra Frames for Training	0	≒10,000	≒500	≒250

表一：實驗數據

而我們挑選的這幾項指標中，根據 COCO dataset 所定義，最值得參考的就是 AP 指標，因為是相對最為嚴格的指標，其定義為把 IoU=0.5 至 IoU=0.95 以 0.05 為間隔去計算 AP，並將這些值平均作為表一中的 AP。AP50, AP75 則是比 AP 本身稍微寬鬆的指標，若模型還在初始訓練階段，表現不佳時可以觀察這兩個指標的變化。

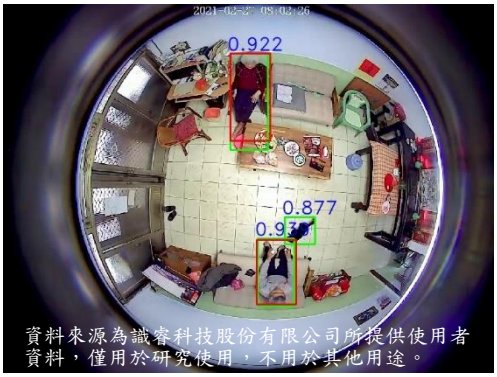
除此之外，我們還會嘗試去對 medium object 和 large object 相關的指標去評估，即 APm, APl, ARm, ARl。由於物體在魚眼鏡頭下，不同位置造成的變形相較於普通鏡頭來的更加嚴重，意味著位於魚眼中央的人所佔之面積會比較大，所以我們也將這些數據一併整理於表一。最後，由於我們想知道究竟影格的數量是否會大大影響模型，因此會整理這幾種情況下的影格數做比較、討論。

根據表一，可以先針對 Baseline 模型分析。可以看到其 AP 僅有非常不理想的 0.493，換言之其先前所 pre-train 的資料，由於是隨意挑選的，因此對於其辨識能力的提升相當有限。就算是比較寬鬆的指標，如 AP50, AP75，分別只有 0.74 和 0.676。

表一之中，第二行數據(without sampling)是不經過任何演算法，直接將 30 分鐘的影片放入模型訓練後的效果，可以看到此時模型已經比起 Baseline 模型，在各項指標都有飛躍提升，但我們整個模型額外多使用了約略 10,000 多張的影格（圖片）數，若考慮每張進行訓練前都需要進行標記(label)，則是一件非常浪費資源的事情。

第三行數據，是利用原始的 Active Sample 所產生的輸出進行訓練，並從原本 10,000 多張的影格，在我們設定之閥值下產生出約 500 張左右的影格，但根據結果我們會發現到其效果整體都不如前一種方法。但我們所改良的 Active Sample 演算法，效果則為所有方法中最好的，以 AP 為例就足足比 Baseline 進步了 47.5%，AP75 也有將近 40%的長進。訓練所需的影格量卻只需要 250 張左右，比原始的 Active Sample 減少了一半，而這就是我們本專題實驗結果最佳的項目，也是本專題的創新貢獻。

3.3 定性分析(Qualitative Analysis)

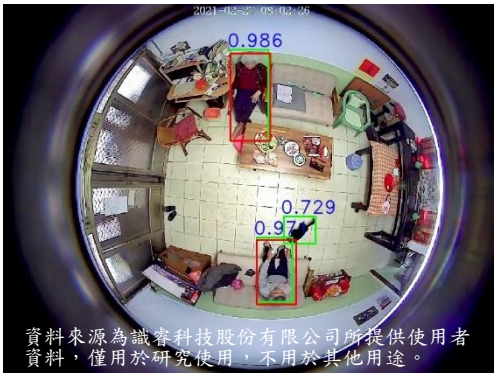


圖十：

Baseline 之判別結果

圖十一：

Without Sampling 之判別結果



圖十二：

Active Sample (ver.1)之判別結果

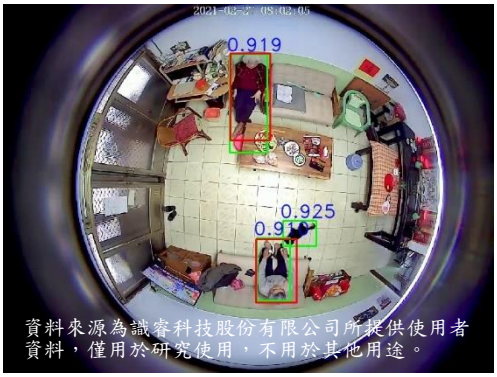
圖十三：

Active Sample (ver.2)之判別結果

在圖十至圖十三中，紅色框是 ground truth，也就是人的真實位置，而綠框是模型的 prediction，至於綠框旁邊的藍色數字代表 confidence score，觀察這幾張圖我們可以發現以下幾件事情。

首先，畫面中央的黑狗在 Baseline（圖十）當中被框上綠框了，也就是模型認為這是一個人，分數為 0.877，但是在 without sampling 的版本（圖十一）中，狗的綠框不見了，而在原始的 Active Sample 方法（圖十二）中，狗的綠框又出現了，又再次被誤判為人，雖然這不是我們樂見的結果，但這無悖於表一的數據，以 AP 為例，從沒有進行 sample 時有 0.699，但經過原始的 Active Sample 方法之後卻降到現在的 0.661，所以模型又再次把狗誤判成人是合理的。但是，在經過我們改良的演算法之後，狗的綠框又再次消失了，代表說模型學習的更好，對應到 AP 也從 ver1 的 0.661 到 ver2 的 0.727，足足有 13%的長進。

除了狗的框之外，我們也可以觀察老奶奶（圖十至圖十三的中上處）與老爺爺（圖十至圖十三的中下處），不管在哪個方法之下，他們兩個都有成功被模型預測到，但分數卻有些許差異。在圖十二與圖十三當中，老奶奶的分數幾乎一模一樣，約為 0.98，而老爺爺的分數則從 0.971 上升到 0.990，證明我們所用的 Active Sample 方法的確有讓模型學習的比原始的方法更好，代表模型對於人的辨識能力有提升。



圖十四：

Baseline 之判別結果

圖十五：

Without Sampling 之判別結果

圖十六：

Active Sample (ver.1)之判別結果

圖十七：

Active Sample (ver.2)之判別結果

接著，我們另外選取四張另一段不同時間的圖片（圖十四至圖十七），是我們認為模型表現比較不好的狀況。我們可以看到狗雖然在四個方法中都有被模型預測出來，而不是我們想要的，但綠框的分數其實是不一樣的，且有優劣差異。在圖十六中，狗的分數仍居高不下，足足有 0.870，但經過我們的方法之後，狗的分數直接降到 0.717（圖十七）。此外，爺爺的分數更從 0.78（圖十六）增加至 0.93（圖十七），這個黑狗分數顯著的降低和爺爺分數的顯著提升都可以說明我們改良的演算法(Active Sample ver.2)的確能成功挑出更有價值的影格提升模型效能。

四、結論

經過我們的實驗，從結果可以知道若採用單純的光影變化去設計演算法來篩選可能會有不足，但若經過我們對演算法進行的改良，也就是多考慮一項「bounding box 面積」的因素去篩選，則能在準確度有大幅提升、所需要的影格數量則是大大減少，我們認為這樣的結果可以非常有效地節省了標記數據(data labeling)的成本。

未來我們也希望可以將模型發展到可以做到 real time，一邊收集資料的同時就去分析資料的重要程度，將那些對模型更有提升效果的資料放回模型去訓練，使得目標檢測模型學習的效率提升。

而從各面相來看我們的方法都比原始的作法來的有效，因此在本次專題內已經達成目前最優(State-Of-The-Art)。儘管如此我們認為方法上還是有地方需要加強，像是在本專題中，我們只有針對 FN 進行處理，如果也可以對 FP 進行優化，或許讓模型的表現還可以更上層樓。

五、參考文獻

[1] Burr Settles. Active Learning Literature Survey, *2010*

[2] Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. arXiv preprint arXiv:1905.03677, 2019

[3] Localization-Aware Active Learning for Object Detection, arXiv preprint arXiv:1801.05124, 2018

[4] Towards Human-Machine Cooperation: Self-supervised Sample Mining for Object Detection, arXiv preprint arXiv:1803.09867, 2018

[5] Scalable Active Learning for Object Detection, arXiv preprint arXiv:2004.04699, 2020

[6] Efficient adaptive density estimation per image pixel for the task of background subtraction, 2006

[7] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv preprint arXiv:1506.01497, 2015
[8] Deep Residual Learning for Image Recognition, arXiv preprint arXiv:1512.03385, 2015
[9] Feature Pyramid Networks for Object Detection, arXiv preprint arXiv:1612.03144, 2016
[10] COCO Detection Evaluation, https://cocodataset.org/#detection-eval
[11] Detectron2, https://github.com/facebookresearch/detectron2

六、計畫管理與團隊合作方式

6.1 計畫管理

我們一開始加入孫民老師的實驗室做專題，就是希望能夠以實作為主，藉由這次專題去學到修課所學不到的。於是我們時常與博士班的學長請益，去討論我們該做什麼樣的題目，而學長提到說實驗室的 CarePLUS 正在進行一項大型專案，是關於老人照護的題目，希望我們能從這整個大型專案挑選一小部份去實作，一方面能一定程度地協助大型專案的進度，另一方面則是能有更多實作機會而非純理論研究。

由於我們先前對於機器學習、深度學習領域涉略還不夠深，因此上學期前半段主要的進度是讓我們自行去學習相關的線上課程，並於每周向學長們報告課程內所學，並記錄學長給予的方向、建議及回饋。到了後半段，則是開始每周自己去選取深度學習領域的 paper，並於每周開會的時候一樣花費 2 小時到實驗室報告。並於上學期期末的時候，去把想做的題目跟整個系統的流程設計定案。

真正實作的進行主要是到了下學期才啟動，一開始是先熟悉 Active Sample 演算法，去發想能改進的方向和可行性，以及訂下具體的目標去思考解法。然後，我們的專題會需要一個目標檢測模型來讓我們評估效能，而在與學長討論過後，我們決定自行利用 Detectron2 的框架去架設模型，認為這樣自由度可以更高，比如說可以快速地選擇所要使用的 network，也可以學到更多架設模型的細節，像是：資料集處理、API 的使用等等。

6.2 團隊合作

我們在前期分工上是兩人一起進行，一同發想我們的專題概念，及記錄下每次與學長的開會內容。而到了實作部份，我們的分工則主要由謝霖泳對 Active Sample 演算法進行改良，而劉亦傑則處理目標檢測模型的架設和數據的統整。最終報告主體、摘要、海報等內容則是由兩人共同合力，一步步完成本專題報告中的每一個小部份，最後再進行統整。