Vehicular Vision System

HW2 - Autonomous Driving HW2 - Report

Group 16 (309605008黃筱晴 , 0851918嚴麗芯)

Framework :

採用SSD網路(Single Shot MultiBox Detector)，它發表於2016年的ECCV，是YOLO的改良。SSD的做法跟YOLO有兩項不同的改變，第一就是加入 Pyramidal Feature Hierarchy，採用不同大小的特徵圖來檢測: 在大特徵圖上檢測小物體，在小特徵圖上檢測大物體;第二是參考 Faster R-CNN 中的 Anchors，依據每個特徵圖的大小產生一系列的先驗框 (Prior boxes，SSD 稱為 Default boxes)

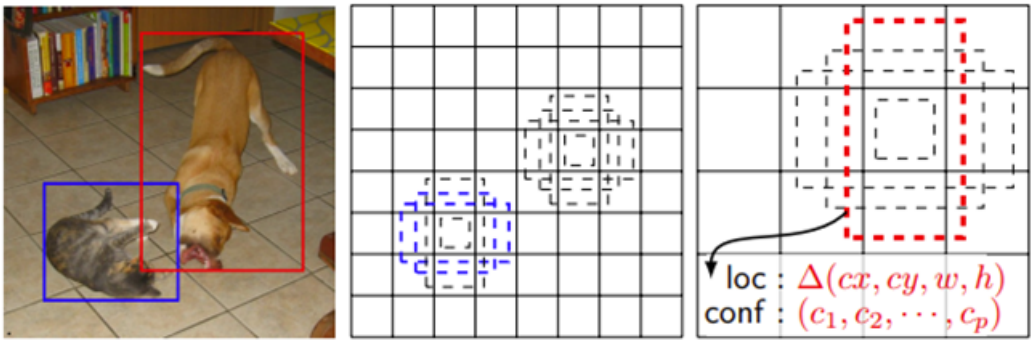
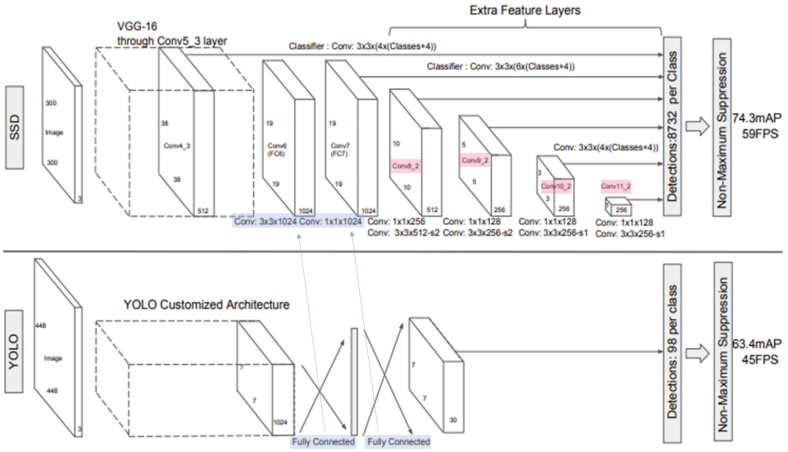
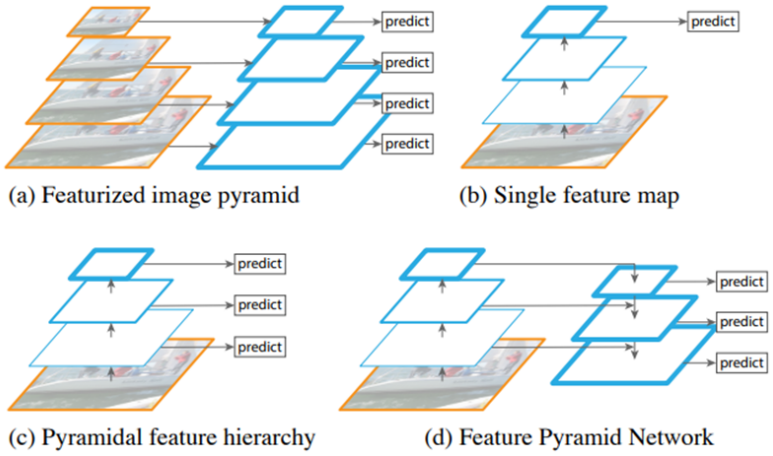


Fig. 1: **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ($(c_1, c_2, \cdots, c_p)$). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

SSD 的 Backbone 採用 VGG16，將 VGG16 的 pool5 層從 size=2x2, stride=2 更改為 size=3x3, stride=1，以及最後兩個全連接層 FC6、FC7 分別改為 3x3、1x1 卷積層，並且在 3x3 卷積層中使用 Atrous Algorithm (就是指 Dilated Convolution)。然後移除所有 dropout 層與 FC8 層，再增加四個卷積層 Conv8_2、Conv9_2、Conv10_2、Conv11_2



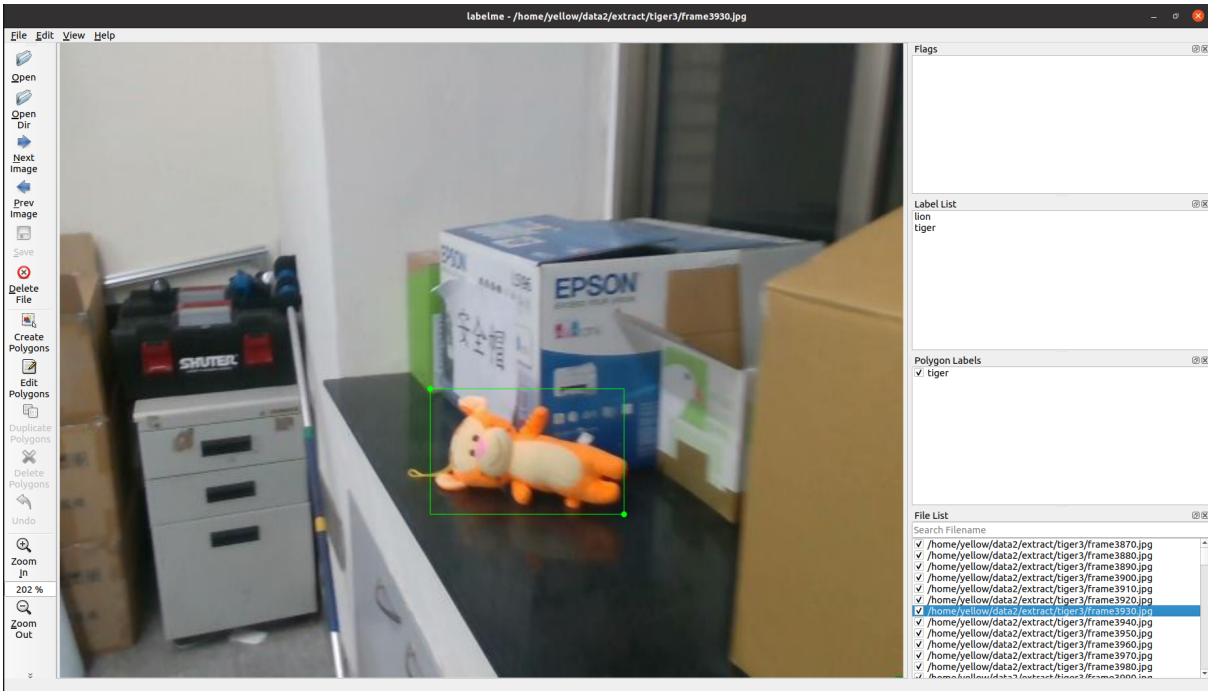
與 YOLO 採用全連接層預測不同，SSD 並在模型裡新增了輔助結構Pyramidal Feature Hierarchy，即是在不同大小的特徵圖 (不同的感受野) 中檢測，SSD是首次使用這種架構的網路。



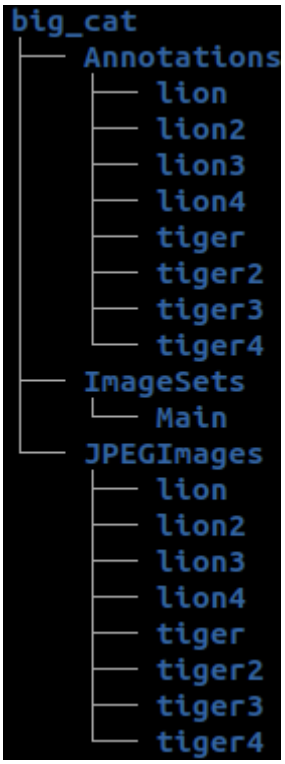
Dataset :

我們目標物有兩個類別，分別是Lion和Tiger。首先，我們將目標物放在多個任意的位置上，然後使用RealSense D435 camera 並放置在一張可移動式椅子上朝着目標物移動與旋轉，其間會多次改變物件的姿態錄製影片。最後將錄得的影片分割照片並以1/10 frames比例取樣，最後兩個物件各有約1500張影像作為訓練集，每張照片中只有單一物件，影像的大小是640*480。

在Label上，我們使用MIT提供的開源軟體LabelMe進行資料標記，在該軟體上手動label出bounding box後，該軟體會生成.json檔案。



dataset下載:
https://drive.google.com/file/d/1oiHwSYmIWDrEhec0NNmJFxVQ4LXmc3hE/view?usp=sharing
dataset檔案結構:



Training Process

SSD網路的實作和traning code是參考至https://github.com/amdegroot/ssd.pytorch。修改Dataloader的部份來配合我們自己的資料集，並嘗試許多訓練參數的調整。

- hyperparameters
  前面VGG網路的部份有load pretrained weight
  Optimizer：SGD
  BatchSize = 4
  Workers = 4
  LearningRate = 1e-4
  Momentum = 0.4
  WeightDecay = 5e-3
  Gamma = 0.1

- loss changing

Discussion and conclusion :

1. 影像資料前處理
   讀進image時，為了配合pretrain的VGG weight，先縮放影像大小為300*300。然後因為OpenCV讀進影像的格式為BGR，再轉換至RGB色彩空間。接著對RGB三個顏色分別做normalization，再進行Augmentation增加資料的多樣性。
   可能因為有做資料前處理的關係，尤其是resize的影響，使用同樣weight跑inference，用D435（像素640*480）可以有不錯的表現，但嘗試使用其他相機（羅技C270，像素1280*720）表現和使用D435有蠻大落差。

2. 資料集性質
   一開始只收了各700張左右的資料，雖然以張數來看應該已經足夠，但訓練時loss雖然有收斂但稍微偏高，跑inference效果也不好。分析原因可能是資料的變異度過低。後來增加了幾個差異較大的背景以及不同光源下的資料，並調整訓練參數，可以讓loss在<3收斂，效果好很多。