

2019MLSPRING FINAL PROPOSAL

成員

B05902128 鄭百凱(隊長)
B05902117 陳柏文
B05902125 葛洧為
B05902033 高晟瑋

題目

RSNA Pneumonia Detection Challenge

Problem Study

YOLO

YOLO會將原圖切成 $S \times S$ 個grid，對每一個grid會產生一組output，表示這個grid應該是什麼東西和box的大小。這個output包含x,y的offset還有box的長、寬、信心分數，以及屬於各個class的機率，共有 $5+N$ 個class這麼多維。

Training時則是用CNN的架構，flatten之後接DNN，產生 $SS(5+N)$ 個output。Loss function則是由很多部分組合在一起，其中包含用IOU計算box的正確面積比率和計算class之間的機率分布差異。

Tips

可以先決定好box的形狀，比如說如果要偵測人，那box的形狀應該要比較細長。

Reference： https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088 (https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

R-CNN

先把圖片分成2000個candidate region proposals，這些region來自於selective search algorithm，方法是先產生sub-segmentation，接著遞迴地利用greedy algorithm合併這些區域，最後生成candidate region，然後丟進CNN裡面出來4096維特徵向量，最後使用SVM進行分類，除此之外還會預測四個偏移值使邊界更加準確，這方法壞處便是需要為每張圖片準備2000個candidate region，而且每張圖片需跑47秒，以及greedy沒有辦法learning可能導致bad candidate region。

Fast R-CNN

方法與上面不同的地方在於把整張圖片丟進去CNN，再從得到的convolutional feature map中的候選區域包成正方形，接著利用RoI pooling重新reshape成固定大小餵進fully connective layer，其中RoI feature vector會用softmax來計算偏移值。比較快的原因便是不需要每次都丟2000個region proposals進入CNN。

Faster R-CNN

跟上述的方法大致相同，但是把selective search的過程拿掉，換成另外的network進行學習。

reference

<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> (<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>)

retinanet

retinanet以resnet為基底，在其上方建置一個feature pyramid network，anchor box在P3與P7的feature pyramid上分別表示 32×32 與 512×512 的大小，每個pyramid level，設置aspectratio=1:2,1:1,2:1，scalratio=0， $2^{1/3}$ ， $2^{2/3}$ 。相當於每個level上會產生9個anchor。對於每個anchor box，預測其類別，並進行box regression，根據100k個anchor的focal loss總和來計算出total focal loss。

reference

<https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4>
(<https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4>)

Proposed Method

- 目前是使用Retinanet的架構，backbone model為resnet50

training

- 把training labels轉成要求的csv格式丟進去train
- epoch: 5
- learning rate: 1e-5
- optimizer: adam

prediction

- 把score >= 0.3的box都輸出成training_label的格式
- 再用助教的code轉成submission.csv 上傳

接下來目標

- 嘗試retinanet但用不同的backbone model
- 試試看不同的框架，ex: YOLO...
- 嘗試將病人資料也加入一起train。

Result

- best score: 0.18930
- reference: <https://github.com/fizyr/keras-retinanet> (<https://github.com/fizyr/keras-retinanet>)