# VRDL Homework 3

0716072 吳季嘉

## GitHub link of my code

https://github.com/072jiajia/VRDL_HW3

## Reference

https://github.com/facebookresearch/detectron2 (Library)
https://arxiv.org/pdf/1712.00726v1.pdf (Cascade R-CNN)
https://arxiv.org/pdf/1803.08494.pdf (Group Normalization)
https://arxiv.org/pdf/1703.06211.pdf (Deformable Convolutional Networks)

## Brief Introduction

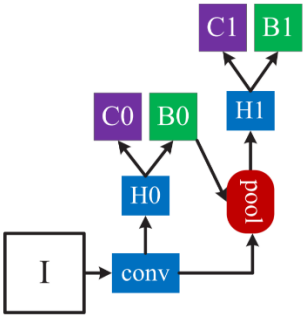
在這份作業，我套用 Facebook Research 開源的 detectron2 library。模型使用 Misc/cascade_mask_rcnn_X_152_32x8d_FPN_IN5k_gn_dconv.yaml。此模型的架構使用 Cascade Mask R-CNN，並以 pretrain 在 ImageNet-5k 上的 ResNeXt-152 32x8d 作為 backbone。使用 Group Normalization 做標準化，在 ResNeXt 的 Res3, Res4, Res5 這三個 stage 使用 Deformable Convolution。

在 inference 的時候，我將圖片 resize 成 2 種不同大小做預測，以增加較大及較小物件偵測的精度，最後將兩個預測融合並做 nms 去除重複項目。

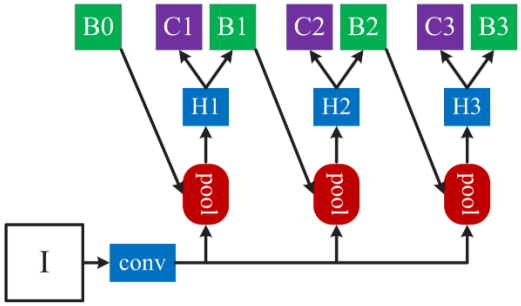
## Related Works

### Cascade Mask R-CNN

此概念是由 Cascade R-CNN: Delving into High Quality Object Detection 這篇論文提出。原先的 Faster R-CNN 會先提出 proposal 以及其對應的類別（圖中的 B0 和 C0）然後對 proposal 再進一步做 regression 和 prediction。此篇論文則提出，可以對 regression 後的 bounding box 再多做幾次 regression，因此也可以得到更精準的 bounding box。




Faster R-CNN




Cascade R-CNN

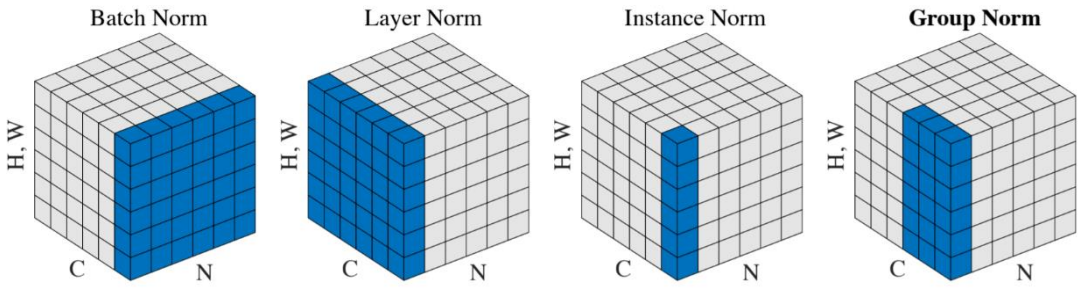
以這份作業我使用的方法為例，模型會先從 RPN 輸出 prososal，其中與物件 IoU 大於 0.5 的為正樣本，其餘為負樣本。接著 detection stage 1 會對他進行 classification 和 regression，然後以這些 regression 完的 Bounding Box 作為 detection stage 2 的 proposal，其中 IoU 大於 0.6 的為正樣本，其餘為負樣本。detection stage 3 亦同，但 IoU 的 threshold 設為 0.7。

Loss Function 則和 Mask R-CNN 相同，只是最後的 cls loss 和 reg loss 要計算 3 個 stage，並給 stage 1 較小的權重，stage 3 較大的權重。

以這樣的方法因為有使用到 IoU 較大的 proposal，所以相對於只有一個 stage 的 Mask R-CNN 可以減少 False Positive 的預測，且此方法不同於直接設定高的 IoU threshold，以漸進的方式可以確保正樣本的數量。

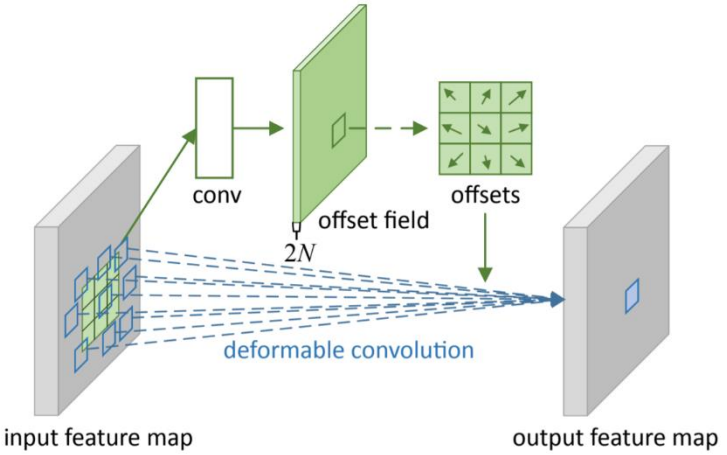
## Group Normalization

因為 Batch Normalization 在 Batch Size 小的時候效果會變差，所以在 Group Normalization (GN)這篇論文中提出了這個概念來避免 Batch Size 的影響。如下圖所示，GN 其實就是 Layer Normalization 和 Instance Normalization 的折衷版，但調整好 Group Size，可以得到比前兩者更好的 performance 。




## Deformable Convolutional Networks

如下圖所示，可變形卷積會針對不同的特徵去學習出如何將原先固定形狀的卷積核調整成其他的形狀，以提高模型的 performance。經過實驗，deformable conv 的設計可以提高模型對圖片大小，旋轉等映射的泛化能力。



# Methodology

## Data pre-process

在 training 階段，所有的照片有 50%的機率會做 random crop，但 crop 後的照片必須確保包含至少兩種物件以增加 training 時的 robustness。接著我隨機翻轉並調整所有照片的亮度、對比、飽和度，再將所有的照片做等比例的縮放，將短邊縮放成 300 到 500 之間的一個隨機值，最後將照片做隨機旋轉。

## Model

模型使用 Cascade Mask R-CNN，以 ResNeXt-152 32x8d 作為 backbone。在 Normalization 的 Layer 使用 Group Normalization，在 ResNeXt 的 Res3, Res4, Res5 這三個 stage 使用 Deformable Convolution。

## Hyperparameters

**optimizer**

這個模型共訓練的 30000 個 iteration，
使用 SGD 作為 optimizer，
learning rate 定為 0.001，
前 100 個 iteration 做 linear warm up，接著做 cosine annealing 下降 lr
為了避免梯度過大將梯度 clip 到 -1 到 1 之間。

**Data Augmentation**

50%會做 Random Crop，Crop Size 是原圖大小的 0.5 ~ 1 之間的隨機值。
50%會做水平翻轉
20%會做 Random Rotate，隨機旋轉 -30 ~ 30 度
亮度、對比、飽和度各有 20%會做，調整為 0.75~1.25 倍之間的隨機值

**Testing**

先產生兩張 resize 後的照片。一張短邊為 400，另一張的短邊為 800。
先分別對兩張照片做 prediction，然後將兩份結果 concatenate 起來並使用 nms 將重複的部份移除。
其中 400 為 training data 大小的平均值 (300 + 500) / 2
800 為 training data 在 crop 後的整張圖的應該被 resize 的大小
nms 的 threshold = 0.7 和 RPN 所使用的相同。

# Findings and Summary

在這份作業中因為我使用了的模型相對較大，所以不僅要使用 Group Normalization 來取代 Batch Normalization，我還不能讓 training 的 image size 太大避免 CUDA out of memory。因此我讓 training data 有 50%是原圖，另外 50%是被 crop 後 resize 的放大圖。讓 model 可以對大物件和小物件都有偵測能力。並在 inference 的時候對 resize 成不同大小的同一張照片做預測。結果

發現這樣有更好的 performance。