

# VRDL HW1

學號:0716314

姓名:陳鎧勳

## 0. GitHub link of my code:

[https://github.com/KaivinC/CV\\_HW1](https://github.com/KaivinC/CV_HW1)

## 1. Reference:

<https://github.com/JDAI-CV/LIO>

[https://openaccess.thecvf.com/content\\_CVPR\\_2020/papers/Zhou\\_Look-Into-Object\\_Self-Supervised\\_Structure\\_Modeling\\_for\\_Object\\_Recognition\\_CVPR\\_2020\\_paper.pdf](https://openaccess.thecvf.com/content_CVPR_2020/papers/Zhou_Look-Into-Object_Self-Supervised_Structure_Modeling_for_Object_Recognition_CVPR_2020_paper.pdf)

## 2. Brief introduction:

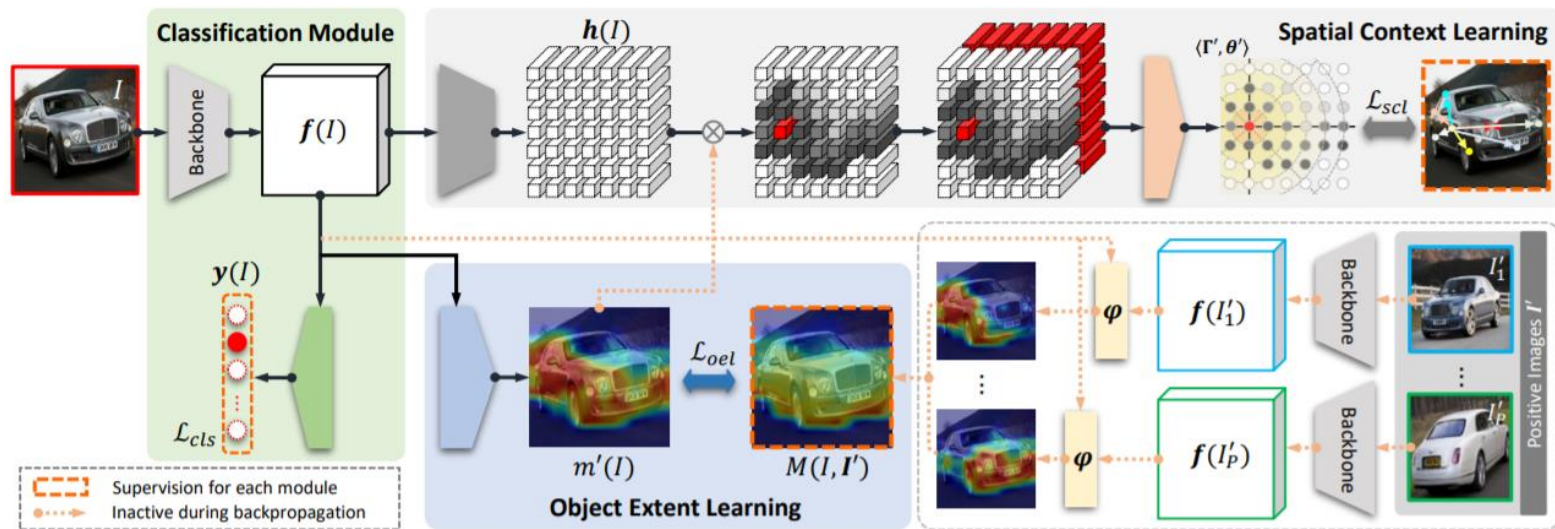
這個作業我所使用的技術是參考了 Look-into-Object: Self-supervised Structure Modeling for Object Recognition 這篇論文裡的方法，裡面所提到的 OEL 和 SCL 技術可以簡單快速地移植到任何的 backbone 上。

## 3. Methodology:

### 1. Data pre-process:

一開始拿到的 raw data csv 是 "id,label"，為了符合模型，所以我把 csv 格式改成 "id.jpg,category(0~195)"，而我有一組 dict 紀錄每個 category 映射到哪個 label 方便之後轉回來。其他 data augmentation 會在下方 Hyper-parameters 介紹

## 2. Model architecture:



Backbone 都是利用 resnet-50，pipeline 主要分為上面灰色的 SCL 和下面藍色的 OEL 以及右邊的綠色區塊。

OEL 主要目的是分離前景和背景，方法是利用  $r$  個同種類的正樣本做類似 attention 的方法算出其相似度，並把結果當作 ground truth 和把  $f(I)$  經過  $1 \times 1$  卷積所得的  $m'(i)$  算  $L_{oel}$ ，最後把  $m'(I)$  和灰色區塊的  $h(I)$  做 fuse。

SCL 則是因為物體的結構對於分類有重要意義，而找出物體空間上的相對關係有助於找出物體的結構，而 SCL 這邊的方法就是利用 OEL 所得到最有可能是物體的點來當作參考點，根據像素座標來預測像素與參考點的相對關係(用極座標來表示)，而 ground truth 就是實際算出來的極座標，這邊的 loss 為  $L_{scl}$ 。

最後 back propagation 時的  $L_{total} = L_{cls} + L_{scl} + L_{oel}$ ，而在 infer 的時候，黑色和藍色區塊會被 disable，只有綠色區塊有用。

## 3. Hyperparameter:

本次作業我的 optimizer 是選用 SGD，而 learning rate 一開始是 0.001 而每 40 epoch 會變成原本的 0.1 倍，而 momentum 則是 0.9，總共 epoch 數為 200。

data augmentation 則是會把每張圖片 resize 成 512 在 random crop 成 448 接著經過標準化(mean、std 是整個 dataset 的)，之後隨機旋轉  $\pm 10$  度，

最經過後 random horizontal flip

最後，我根據不同的 train/val 切割法訓練了 6 個不同的模型，最終結果是由 6 個模型投票出來的

#### 4. Summary:

這次作業學到很多，包括如何自己刻一個可以用的 model 還有設置 hyperparameter 像 learning rate 隨著 epoch 下降還有 data augmentation，但只有單純的 resnet50 連過 baseline 都是一定的挑戰，所以去參閱了 SOTA 的文章，才順利的把準確度提高到超過 baseline。