**P6**

We researched different methodologies for building the object localisation model.

We started from the basic and common image processing model called Convolution Neural Network, also known as CNN. CNN first extracts image features by different filter applications. By building many convolution and pooling layers, CNN enables the machine to recognise objects in images. However, our CNN model could only detect one object with a bounding box at a time. So, we ended up using it as our baseline model.

The second methodology we worked on and is also our final model for our project is Mask R-CNN. Like Faster R-CNN, Mask R-CNN uses a single deep convolution network to extract features for the entire image once. Also, it uses a separate fully convolutional network to predict objects. Besides two original outputs used to classify and localize the object, Mask R-CNN also outputs an additional 3rd thing which is the object mask that supports better performance.

The next methodology is YOLO version 3. It finds all objects in an image grid at one time, and that’s why it is named as YOLO, You Only Look Once. Unlike R-CNN using region-based techniques, YOLO processes full images rather than only a part of images during training, and optimize detection performance directly. This supports YOLO to have a better accuracy rate and be faster.

Xception is based on Depthwise Separable Convolutions which can be understood as an Inception model with a maximally large number of towers. Like Inception, the hypothesis of Xception is that the mapping of cross-channel correlations and spatial correlations in the feature maps of convolutional neural networks can be entirely decoupled. This makes the architecture very easy to define and modify.

Based on our model trial results in term of mAP as shown in the screen, we decided to use Mask R-CNN, which got a highest recorded train mAP of 0.79, as our final model.

**P7**

We implemented the Mask R-CNN model based on the following steps.

First, we obtained the Mask R-CNN project by Matterport from GitHub. In data preprocessing, we created a csv file containing the annotation information including label and bounding box for each image. We utilised the “load data-set function” from Mask R-CNN library to generate an array with the same dimension of the image size to create the masks. Then, we split 75 and 25% for training and testing datasets based on previous steps.

In model training, the Mask R-CNN model we used is based on resnet101 architecture. We defined the model configuration including the number of classes and the learning rate of SGD optimizer. We used pre-trained weights for 5-epochs baseline model training first. We then adjusted the model each time based on the output of class and bounding box losses.

We also conducted different model experimental setting to improve our model performance, more details will be carried out in the next slide.

Finally, the model is evaluated on test dataset and image prediction is generated accordingly. Again, more details regarding this part will be outlined in a later section.