# Generating Visual Information in Deep Reinforcement Learning

Alan, CHUNG Ho Lun (MAEG, Yr4)  Supervisor: Dr. Juan Rojas

## Introduction

Robotics manipulation of diverse object is one of the greatest challenges in robotics development nowadays. In order to achieve this, we use graph based soft-actor critic reinforcement learning with visual input. We utilize GNN for graph-based computation since it can handle dynamic number of input and can recognize relation between objects and robot arm. We use Mask RCNN to detect object inside the environment.

## GNN(Graph Neural Network)

Graph Neural Network(GNN), is using the idea of convolutional Network in graph. With can allow us the find pattern and connection inside the graph. It is very useful since many data in the real world is not a fixed size matrix or image, e.g., social connection, road etc.
A general GNN, will have three stage of action to perform: message, aggregation, and update.

***General GNN Algorithm***

***Input:***

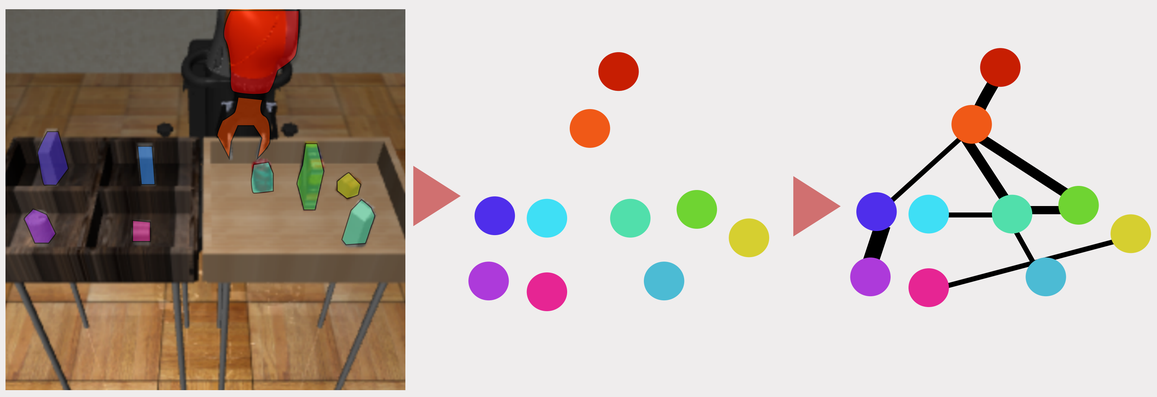
- *a graph (V,E) with V has shape (m,no. node), and E has shape (2, no. edge)*
- *a message function MESSAGE*
- *a aggregation function AGGR*
- *a update function UPDATE*

***Output:***

- *a isomorphic graph(V,E), with same or different embedded dimension.*

```
messages_buffer = {}

For every edge e in the graph E
    messages_buffer <- (e, MESSAGE(source, destination))

New_V = {}

For every node v in the graph V
    message_to_v <- filter(message_buffer, destination of e == v)
    aggr_message = AGGR( message_to_v )
    New_V = UPDATE( v, aggr_message )

return (New_V, E)
```

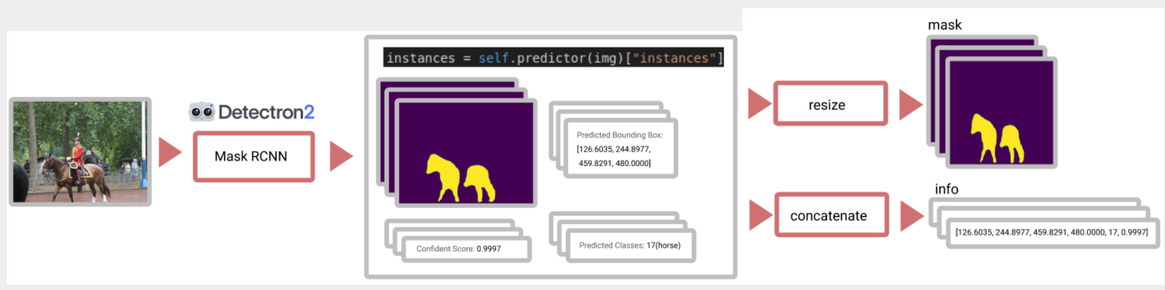
Thus, in this project, we try to use a graph to represent the state of a bin Picking environment, in which every object in the environment will be a node in the graph.



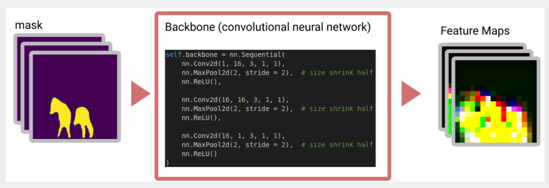
We will first transform each object in a image to a group feature vectors. But instead of using basic GNN, We used a transformer, a tool that perform great on NLP problem, to work out what is the connection between the nodes.
The Transformer allow the agent to pay more attention on the object that is more important and ingore those which is not important to the current task. For example, when the task is to put the milk box in a certain position, the milk box and the destination shoud have larger weight and other object should be paid less attention.

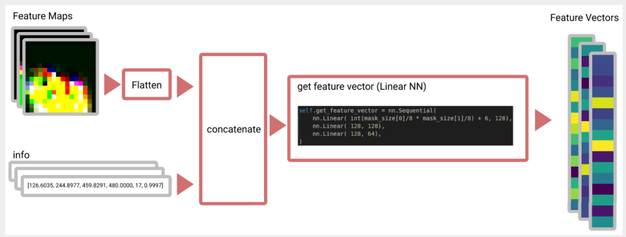
## Mask RCNN

To achieve that, we use a cutting edge object detecton algorithm, mask RCNN, which allow use to locate and classify object in an image.
We use Detecton2, a computer vision library built on to of pytorch, to train and build our mask RCNN model. After getting the instances, we preproces them further action, i.e. resize the mask to a certain size, and merge all other information into a *info* vector.




First, we pass the resized mask to a “backbone”(a convolutional network), to shrink the size of the mask for ease of computation and extract feature informations.



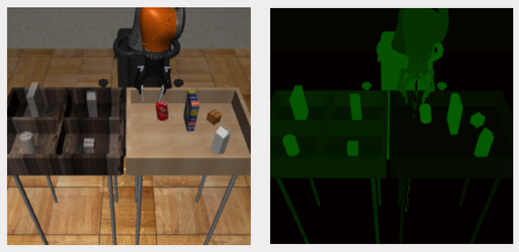



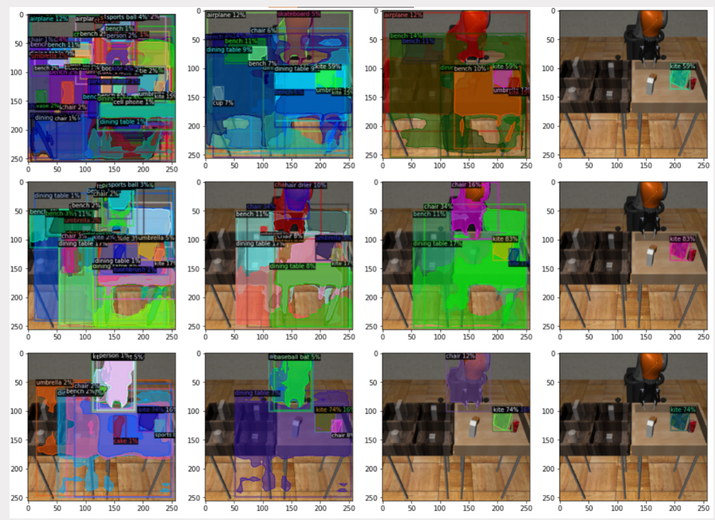

Finally, we flatten the feature map, then concatenate it with other information, then pass it through a linear layer to get a set of Feature Vectors, as a graph for the representation of the enviornment.

## Dataset

We initally decided to use COCO dataset pre-trained Mask RCNN model to be the be the object detection model. COCO stand for Common Object in Context, which is a large scale dataset for object detection, segmentation and other computer vision task. It contain over 220K label image and 80 object catagories with over 1.5M instance labled.
However, in testing in the mujoco envorinment, we find out that the COCO pre-train dataset it is not a good representation for the binPicking environment.

Therefore, we are trying to the envoriment’s rendering engine to make our own dataset on the fly, and after a certain epoch, we will tune the model again with the dataset created





model by row: R50_FPN1x, R50_FPN3x, X101_FPN3x
confidence level by column: 1%, 5%, 10%, 50%

## Reference

Semi-Supervised Classification with Graph Convolutional Networks. T.N.Kipf et al.
Mask R-CNN. K.He, G. Gkioxari et al.
Microsoft COCO: Common Objects in Context. T.Y. Lin, M. Maire et al.
Pytorch Geometric. https://github.com/rusty1s/pytorch_geometric, Fey, Matthias and Lenssen, Jan E.
Pytorch Detectron2: https://github.com/facebookresearch/detectron2, Y. Wu, A. Kirillov et al.