

- **GitHub link of my code:**

[https://github.com/oning0517/VRDL\\_HW2.git](https://github.com/oning0517/VRDL_HW2.git)

- **Brief Introduction**

Before starting training, I should convert the data into COCO format or middle format in order to use MMDetection. I choose to convert the data into COCO format. Because converting the data into middle format is easier, I choose the middle format. Below is the example of COCO format.

```
{
  "images": [image],
  "annotations": [annotation],
  "categories": [category]
}

image = {
  "id": int,
  "width": int,
  "height": int,
  "file_name": str,
}

annotation = {
  "id": int,
  "image_id": int,
  "category_id": int,
  "segmentation": RLE or [polygon],
  "area": float,
  "bbox": [x,y,width,height],
  "iscrowd": 0 or 1,
}

categories = [{
  "id": int,
  "name": str,
  "supercategory": str,
}]
```

Ref: [https://mmdetection.readthedocs.io/en/latest/2\\_new\\_data\\_model.html](https://mmdetection.readthedocs.io/en/latest/2_new_data_model.html)

In training, I use the model, Faster RCNN.

- **Faster R-CNN**

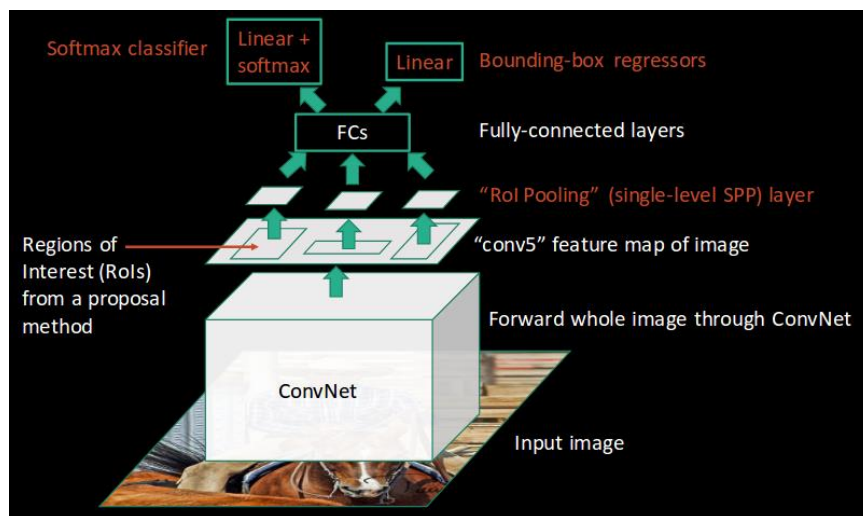
R-CNN is slow in training and inference. For example, we have 2,000 proposals which each of them needed to be processed by a CNN to extract features, R-CNN will repeat it 2,000 times to extract features.

As a result, the author of Faster R-CNN modified the order of operations, structural problems, and simplified the design of the model to improve the speed of operations.

Basically, the significant changes are:

1. Remove the SVM structure and use softmax as the NN for Classification.
2. Forward whole image through ConvNet to get the feature map first. And then map the ROI one the map to get the actual ROI. Because the image only needs to undergo Convolution once, which saves lots of computing and hardware resources.
3. Use the ROI pooling layer to unify the dimensions before the ROI is sent to the Fully-connected layer (FC) layer.

The picture below is the structure of Faster R-CNN.



Ref: <https://ivan-eng-murmur.medium.com/object-detection-s2-fast-rcnn-%E7%B0%A1%E4%BB%8B-40cfe7b5f605>

## ● Summary

I am so sad that I face an error and I cannot deal with it.

Below is the screenshots of my error.

```
# Import your package and check the version
%cd mmdetection

import torch, torchvision
print(torch.__version__, torch.cuda.is_available())

import mmdet
print(mmdet.__version__)

import mmcv
print(mmcv.__version__)

from mmcv.ops import get_compiling_cuda_version, get_compiler_version
print(get_compiling_cuda_version())
print(get_compiler_version())

from mmdet.apis import inference_detector, init_detector, show_result_pyplot

# You must import the below 5 packages
import os
import cv2
import time
from tqdm import tqdm
from google_drive_downloader import GoogleDriveDownloader as gdd
```

```

[Errno 2] No such file or directory: 'mmdetection'
/content/mmdetection/checkpoints/mmdetection
1.10.0+cu111 False
2.18.1
1.3.18
-----
ImportError                                Traceback (most recent call last)
<ipython-input-59-07ad5e0e8c7c> in <module>()
    12 print(mmcv.__version__)
    13
--> 14 from mmcv.ops import get_compiling_cuda_version, get_compiler_version
    15 print(get_compiling_cuda_version())
    16 print(get_compiler_version())

-----
      3 frames
/usr/lib/python3.7/importlib/_init_.py in import_module(name, package)
    125         break
    126         level += 1
--> 127     return _bootstrap._gcd_import(name[level:], package, level)
    128
    129

ImportError: /usr/local/lib/python3.7/dist-packages/mmcv/_ext.cpython-37m-x86_64-linux-gnu.so: undefined symbol: _Z27points_in_boxes_cpu_forwardH2at6TensorE50_S0_

NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
-----

```

I could run this code one week ago. I cannot understand why it appeared error since Wednesday. As a result, I cannot train my model after I try to edit my code on Wednesday, and I cannot have the screenshot of my result. However, I still convert the data into COCO format successfully.

## ● Other References

### Load data:

<https://stackoverflow.com/questions/41176258/h5py-access-data-in-datasets-in-svhn/41264930>

### Convert data format:

[https://colab.research.google.com/github/ZwwWayne/mmdetection/blob/update-colab/demo/MMDet\\_Tutorial.ipynb#scrollTo=UsJU5D-QPX8L](https://colab.research.google.com/github/ZwwWayne/mmdetection/blob/update-colab/demo/MMDet_Tutorial.ipynb#scrollTo=UsJU5D-QPX8L)  
[https://mmdetection.readthedocs.io/en/latest/2\\_new\\_data\\_model.html](https://mmdetection.readthedocs.io/en/latest/2_new_data_model.html)

### Introduction of Faster R-CNN and R-CNN:

<https://jhui.github.io/2017/03/15/Fast-R-CNN-and-Faster-R-CNN/>

### Model and Training:

[https://colab.research.google.com/github/ZwwWayne/mmdetection/blob/update-colab/demo/MMDet\\_Tutorial.ipynb#scrollTo=UsJU5D-QPX8L](https://colab.research.google.com/github/ZwwWayne/mmdetection/blob/update-colab/demo/MMDet_Tutorial.ipynb#scrollTo=UsJU5D-QPX8L)  
[https://mmdetection.readthedocs.io/en/v2.18.1/2\\_new\\_data\\_model.html](https://mmdetection.readthedocs.io/en/v2.18.1/2_new_data_model.html)