# Mask R-CNN

## Abstract

### Introduction

conceptually simple, flexible, and general framework for object instance segmentation.

Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance.

### Methodology

A branch to extends Faster R-CNN

A branch to bounding box recognition.

simple to train and adds small overhead to Faster R-CNN

We show top results in all three tracks of the COCO suite

Including instance segmentation, bounding box object detection, and person keypoint detection. Without bells and whistles,

### Conclusion

Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition.

available at: https://github.com/facebookresearch/Detectron

## Introduction



**cocoapi**  = coco datasets

**detectron** = code and demo, including datasets

**pytorch** = caffe2 Library implement by C++ and support python

### Pre-trained Network

Useful

* Using weight from pre-trained network for reducing training time
* Good for small datasets

Pre-trained sources

* 2 small networks (Mem3G) -> Caffenet , VGG\_CNN\_M\_1024
* 1 huge networks (Mem11G) -> VGG16

### Demo

* ./tools/demo.py

Using VGG16 network trained for detection PASCAL VOC2007, the object proposals are precomputed inorder to reduce installation

* ./tools/demo.py --net cafenet ,or with --net vgg\_cnn\_m\_1024 for small networks

## Methodology

This example demonstrated from this command

|  |  |
| --- | --- |
| python tools/train\_net.py \ | |
| --cfg configs/wattanapong/train\_e2e\_mask\_rcnn\_R-101-FPN\_4x\_coco2014\_train\_valminusmini\_2gpu.yaml \ |  |
| OUTPUT\_DIR /tmp/detectron-output > train\_test\_e2e\_mask\_rcnn\_R-101-FPN\_4x\_coco2014\_train\_valminusmini\_lr0\_02.txt | |

### Resnet

## Results

## Reference

## 