

【全人類がわかるE資格コース PyTorch版】 プロダクト開発演習

5.フレームワーク事前調査

今回使用するMMDetectionフレームワークのチュートリアルを再現する。

これによりフレームワークの使い方を身に付けるとともに、正常にインスタンス・セグメンテーションが実行できる環境を構築する。

再現実行するチュートリアルは、google Colab環境でGPUを使用することを前提とし、Mask R-CNNを使ったものを選定する。

```
In [1]: # Google Driveのマウント
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

5.1 mmdetection v2.23.0 のインストール (google Colab版)

https://github.com/open-mmlab/mmdetection/blob/master/demo/MMDet_Tutorial.ipynb

```
In [2]: # pritrainモデルの動作確認
# MMDetectionのインストール
# Check nvcc version
!nvcc -V
# Check GCC version
!gcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Oct_12_20:09:46_PDT_2020
Cuda compilation tools, release 11.1, V11.1.105
Build cuda_11.1.TC455_06.29190527_0
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
In [3]: # ***** こちらは旧バージョンをロード v2.24.0～最新版v2.25.0 では学習(tools/t
# 関係ファイルのインストール(旧バージョン用)
# install dependencies: (use cu111 because colab has CUDA 11.1)
!pip install torch==1.9.0+cu111 torchvision==0.10.0+cu111 -f https://download.pytorch.org
# install mmcv-full thus we could use CUDA operators
### !pip install mmcv-full -f https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.
### バージョンv2.25.0ではAttributeError: 'ConfigDict' object has no attribute 'device'
### mmdetection v.2.23.0を使用するため、それに合わせてmmcvも1.5.xから1.3.17にダウンロード
!pip install mmcv-full==1.3.17 -f https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.0+cu111
# Install mmdetection
!rm -rf mmdetection
```

```
### !git clone https://github.com/open-mmlab/mmdetection.git
### mmdetection 最新バージョンv2.25.0ではtools/train.pyに問題があり、
### AttributeError: 'ConfigDict' object has no attribute 'device' が発生するので、回避策
!git clone https://github.com/open-mmlab/mmdetection.git -b v2.23.0 --depth 1    ### +
%cd mmdetection

!pip install -e .
```

that are installed. This behaviour is the source of the following dependency conflict s.

torchtext 0.12.0 requires torch==1.11.0, but you have torch 1.9.0+cu111 which is incompatible.

torchaudio 0.11.0+cu113 requires torch==1.11.0, but you have torch 1.9.0+cu111 which i s incompatible.

Successfully installed torch-1.9.0+cu111 torchvision-0.10.0+cu111

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Looking in links: <https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.0/index.html>

Collecting mmcv-full==1.3.17

 Downloading https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.0/mmcv_full-1.3.17-cp37-cp37m-manylinux1_x86_64.whl (50.4 MB)

 |██████████| 50.4 MB 95 kB/s

Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from mmcv-full==1.3.17) (1.21.6)

Requirement already satisfied: opencv-python>=3 in /usr/local/lib/python3.7/dist-packages (from mmcv-full==1.3.17) (4.1.2.30)

Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from mmcv-full==1.3.17) (7.1.2)

Collecting addict

 Downloading addict-2.4.0-py3-none-any.whl (3.8 kB)

Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from mmcv-full==1.3.17) (3.13)

Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from mmcv-full==1.3.17) (21.3)

Collecting yapf

 Downloading yapf-0.32.0-py2.py3-none-any.whl (190 kB)

 |██████████| 190 kB 13.6 MB/s

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->mmcv-full==1.3.17) (3.0.9)

Installing collected packages: yapf, addict, mmcv-full

Successfully installed addict-2.4.0 mmcv-full-1.3.17 yapf-0.32.0

Cloning into 'mmdetection'...

remote: Enumerating objects: 1575, done.

remote: Counting objects: 100% (1575/1575), done.

remote: Compressing objects: 100% (1094/1094), done.

remote: Total 1575 (delta 612), reused 781 (delta 466), pack-reused 0

Receiving objects: 100% (1575/1575), 16.33 MiB | 13.50 MiB/s, done.

Resolving deltas: 100% (612/612), done.

Note: checking out '3e2693151add9b5d6db99b944da020cba837266b'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

/content/mmdetection

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Obtaining file:///content/mmdetection

Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from mmdet==2.23.0) (3.2.2)

Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from mmdet==2.23.0) (1.21.6)

Requirement already satisfied: pycocotools in /usr/local/lib/python3.7/dist-packages (from mmdet==2.23.0) (2.0.4)

Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from mmdet==2.23.0) (1.15.0)

Collecting terminaltables

```
Downloading terminaltables-3.1.10-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->mmdet==2.23.0) (1.4.3)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->mmdet==2.23.0) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->mmdet==2.23.0) (0.11.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->mmdet==2.23.0) (3.0.9)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1->matplotlib->mmdet==2.23.0) (4.1.1)
Installing collected packages: terminaltables, mmdet
    Running setup.py develop for mmdet
Successfully installed mmdet-2.23.0 terminaltables-3.1.10
```

In [4]:

```
# インストール環境の確認
from mmcv import collect_env
collect_env()
```

Out[4]:

```
{'CUDA available': True,
'CUDA_HOME': '/usr/local/cuda',
'GCC': 'gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0',
'GPU 0': 'Tesla P100-PCIE-16GB',
'MMCV': '1.3.17',
'MMCV CUDA Compiler': '11.1',
'MMCV Compiler': 'GCC 7.3',
'NVCC': 'Build cuda_11.1.TC455_06.29190527_0',
'OpenCV': '4.1.2',
'PyTorch': '1.9.0+cu111',
'PyTorch compiling details': 'PyTorch built with:¥n - GCC 7.3¥n - C++ Version: 2014
02¥n - Intel(R) Math Kernel Library Version 2020.0.0 Product Build 20191122 for Intel
(R) 64 architecture applications¥n - Intel(R) MKL-DNN v2.1.2 (Git Hash 98be7e8afa711d
c9b66c8ff3504129cb82013cdb)¥n - OpenMP 201511 (a.k.a. OpenMP 4.5)¥n - NNPACK is enab
led¥n - CPU capability usage: AVX2¥n - CUDA Runtime 11.1¥n - NVCC architecture flag
s: -gencode;arch=compute_37,code=sm_37;-gencode;arch=compute_50,code=sm_50;-gencode;ar
ch=compute_60,code=sm_60;-gencode;arch=compute_70,code=sm_70;-gencode;arch=compute_75,
code=sm_75;-gencode;arch=compute_80,code=sm_80;-gencode;arch=compute_86,code=sm_86¥n
- CuDNN 8.0.5¥n - Magma 2.5.2¥n - Build settings: BLAS_INFO=mkl, BUILD_TYPE=Release,
CUDA_VERSION=11.1, CUDNN_VERSION=8.0.5, CXX_COMPILER=/opt/rh/devtoolset-7/root/usr/bi
n/c++, CXX_FLAGS= -Wno-deprecated -fvisibility-inlines-hidden -DUSE_PTHREADPOOL -fopen
mp -DNDEBUG -DUSE_KINET0 -DUSE_FBGEMM -DUSE_QNNPACK -DUSE_PYTORCH_QNNPACK -DUSE_XNNPAC
K -DSYMBOLICATE_MOBILE_DEBUG_HANDLE -O2 -fPIC -Wno-narrowing -Wall -Wextra -Werror=ret
urn-type -Wno-missing-field-initializers -Wno-type-limits -Wno-array-bounds -Wno-unkno
wn-pragmas -Wno-sign-compare -Wno-unused-parameter -Wno-unused-variable -Wno-unused-fu
nction -Wno-unused-result -Wno-unused-local-typedefs -Wno-strict-overflow -Wno-strict-
aliasing -Wno-error=deprecated-declarations -Wno-stringop-overflow -Wno-psabi -Wno-err
or=pedantic -Wno-error=redundant-decls -Wno-error=old-style-cast -fdiagnostics-color=a
lways -faligned-new -Wno-unused-but-set-variable -Wno-maybe-uninitialized -fno-math-er
rno -fno-trapping-math -Werror=format -Wno-stringop-overflow, LAPACK_INFO=mkl, PERF_WI
TH_AVX=1, PERF_WITH_AVX2=1, PERF_WITH_AVX512=1, TORCH_VERSION=1.9.0, USE_CUDA=ON, USE_
CUDNN=ON, USE_EXCEPTION_PTR=1, USE_GFLAGS=OFF, USE_GLOG=OFF, USE_MKL=ON, USE_MKLDNN=0
N, USE_MPI=OFF, USE_NCCL=ON, USE_NNPACK=ON, USE_OPENMP=ON, ¥n',
'Python': '3.7.13 (default, Apr 24 2022, 01:04:09) [GCC 7.5.0]',
'TorchVision': '0.10.0+cu111',
'sys.platform': 'linux'}
```

In [5]:

```
# Check Pytorch installation
import torch, torchvision
print(torch.__version__, torch.cuda.is_available())

# Check MMDetection installation
import mmdet
print(mmdet.__version__)
```

```
# Check mmcv installation
from mmcv.ops import get_compiling_cuda_version, get_compiler_version
print(get_compiling_cuda_version())
print(get_compiler_version())
```

```
1. 9. 0+cu111 True
2. 23. 0
11. 1
GCC 7. 3
```

5.2 MMDetectionチュートリアルの試行

https://github.com/open-mmlab/mmdetection/blob/master/demo/MMDet_InstanceSeg_Tutorial.ipynb

```
In [6]: # download the pre-trained checkpoints for inference and finetuning.
!mkdir checkpoints
!wget -c https://download.openmmlab.com/mmdetection/v2.0/mask_rcnn/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth
--2022-06-25 10:24:21-- https://download.openmmlab.com/mmdetection/v2.0/mask_rcnn/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth
Resolving download.openmmlab.com (download.openmmlab.com) ... 47.254.186.233
Connecting to download.openmmlab.com (download.openmmlab.com)|47.254.186.233|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177867103 (170M) [application/octet-stream]
Saving to: 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth'

checkpoints/mask_rc 100%[=====] 169.63M 10.1MB/s in 16s

2022-06-25 10:24:38 (10.6 MB/s) - 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth' saved [177867103/177867103]
```

```
In [8]: # ネットワークモデルのコンフィグレーション
import mmcv
from mmcv.runner import load_checkpoint

from mmdet.apis import inference_detector, show_result_pyplot
from mmdet.models import build_detector

# Choose to use a config and initialize the detector
config = 'configs/mask_rcnn/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco.py'
# Setup a checkpoint file to load
checkpoint = 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain_poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth'

# Set the device to be used for evaluation
#### device='cuda:0'
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu') #### for CPU

# Load the config
config = mmcv.Config.fromfile(config)
# Set pretrained to be None since we do not need pretrained model here
config.model.pretrained = None

# Initialize the detector
```

```

model = build_detector(config.model)

# Load checkpoint
checkpoint = load_checkpoint(model, checkpoint, map_location=device)

# Set the classes of models for inference
model.CLASSES = checkpoint['meta']['CLASSES']

# We need to set the model's cfg for inference
model.cfg = config

# Convert the model to GPU
model.to(device)
# Convert the model into evaluation mode
model.eval()

```

```

load checkpoint from local path: checkpoints/mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth
MaskRCNN(
    backbone: ResNet(
        (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace=True)
        (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    )
    (layer1): ResLayer(
        (0): Bottleneck(
            (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (relu): ReLU(inplace=True)
            (downsample): Sequential(
                (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
                (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            )
        )
        (1): Bottleneck(
            (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (relu): ReLU(inplace=True)
        )
        (2): Bottleneck(
            (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        )
    )
)

```

```
s=False)
    (bn2) : BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
    (conv3) : Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
    (relu) : ReLU(inplace=True)
)
)
(layer2) : ResLayer(
    (0) : Bottleneck(
        (conv1) : Conv2d(256, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (bn1) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2) : Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3) : Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3) : BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (relu) : ReLU(inplace=True)
        (downsample) : Sequential(
            (0) : Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
            (1) : BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        )
    )
    (1) : Bottleneck(
        (conv1) : Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2) : Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3) : Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3) : BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (relu) : ReLU(inplace=True)
    )
    (2) : Bottleneck(
        (conv1) : Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2) : Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3) : Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3) : BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (relu) : ReLU(inplace=True)
    )
    (3) : Bottleneck(
        (conv1) : Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2) : Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2) : BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3) : Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3) : BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
```

```

ats=True)
    (relu) : ReLU(inplace=True)
)
)
(layer3) : ResLayer(
    (0) : Bottleneck(
        (conv1) : Conv2d(512, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (bn1) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv2) : Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
            (bn2) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv3) : Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3) : BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
                (relu) : ReLU(inplace=True)
                (downsample) : Sequential(
                    (0) : Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
                    (1) : BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
                )
            )
        (1) : Bottleneck(
            (conv1) : Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv2) : Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
            (bn2) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv3) : Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3) : BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
                (relu) : ReLU(inplace=True)
            )
        (2) : Bottleneck(
            (conv1) : Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv2) : Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
            (bn2) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv3) : Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3) : BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
                (relu) : ReLU(inplace=True)
            )
        (3) : Bottleneck(
            (conv1) : Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn1) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv2) : Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
            (bn2) : BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
            (conv3) : Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn3) : BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
                (relu) : ReLU(inplace=True)
            )
        (4) : Bottleneck(
            (conv1) : Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
    )
    (5): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
    )
)
(layer4): ResLayer(
    (0): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
            (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
            (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        )
    )
    (1): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_st
ats=True)
    )
)

```

```

ats=True)
    (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
    )
)
)
init_cfg={'type': 'Pretrained', 'checkpoint': 'open-mmlab://detectron2/resnet50_caff
e'}
(neck): FPN(
    (lateral_convs): ModuleList(
        (0): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ConvModule(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): ConvModule(
            (conv): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (3): ConvModule(
            (conv): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
        )
    )
    (fpn_convs): ModuleList(
        (0): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        )
        (1): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        )
        (2): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        )
        (3): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        )
    )
)
init_cfg={'type': 'Xavier', 'layer': 'Conv2d', 'distribution': 'uniform'}
(rpn_head): RPNHead(
    (loss_cls): CrossEntropyLoss(avg_non_ignore=False)
    (loss_bbox): L1Loss()
    (rpn_conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (rpn_cls): Conv2d(256, 3, kernel_size=(1, 1), stride=(1, 1))
    (rpn_reg): Conv2d(256, 12, kernel_size=(1, 1), stride=(1, 1))
)
init_cfg={'type': 'Normal', 'layer': 'Conv2d', 'std': 0.01}
(roi_head): StandardRoIHead(
    (bbox_roi_extractor): SingleRoIExtractor(
        (roi_layers): ModuleList(
            (0): RoIAlign(output_size=(7, 7), spatial_scale=0.25, sampling_ratio=0, pool_m
ode=avg, aligned=True, use_torchvision=False)
            (1): RoIAlign(output_size=(7, 7), spatial_scale=0.125, sampling_ratio=0, pool_
mode=avg, aligned=True, use_torchvision=False)
            (2): RoIAlign(output_size=(7, 7), spatial_scale=0.0625, sampling_ratio=0, pool
_mode=avg, aligned=True, use_torchvision=False)
            (3): RoIAlign(output_size=(7, 7), spatial_scale=0.03125, sampling_ratio=0, poo
l_mode=avg, aligned=True, use_torchvision=False)
        )
    )
    (bbox_head): Shared2FCBBoxHead(
        (loss_cls): CrossEntropyLoss(avg_non_ignore=False)

```

```

        (loss_bbox): L1Loss()
        (fc_cls): Linear(in_features=1024, out_features=81, bias=True)
        (fc_reg): Linear(in_features=1024, out_features=320, bias=True)
        (shared_convs): ModuleList()
        (shared_fcs): ModuleList(
            (0): Linear(in_features=12544, out_features=1024, bias=True)
            (1): Linear(in_features=1024, out_features=1024, bias=True)
        )
        (cls_convs): ModuleList()
        (cls_fcs): ModuleList()
        (reg_convs): ModuleList()
        (reg_fcs): ModuleList()
        (relu): ReLU(inplace=True)
    )
    init_cfg=[{'type': 'Normal', 'std': 0.01, 'override': {'name': 'fc_cls'}}, {'type': 'Normal', 'std': 0.001, 'override': {'name': 'fc_reg'}}, {'type': 'Xavier', 'distribution': 'uniform', 'override': [{'name': 'shared_fcs'}, {'name': 'cls_fcs'}, {'name': 'reg_fcs'}]}]
    (mask_roi_extractor): SingleRoiExtractor(
        (roi_layers): ModuleList(
            (0): RoIAlign(output_size=(14, 14), spatial_scale=0.25, sampling_ratio=0, pool_mode='avg', aligned=True, use_torchvision=False)
            (1): RoIAlign(output_size=(14, 14), spatial_scale=0.125, sampling_ratio=0, pool_mode='avg', aligned=True, use_torchvision=False)
            (2): RoIAlign(output_size=(14, 14), spatial_scale=0.0625, sampling_ratio=0, pool_mode='avg', aligned=True, use_torchvision=False)
            (3): RoIAlign(output_size=(14, 14), spatial_scale=0.03125, sampling_ratio=0, pool_mode='avg', aligned=True, use_torchvision=False)
        )
    )
    (mask_head): FCNMaskHead(
        (loss_mask): CrossEntropyLoss(avg_non_ignore=False)
        (convs): ModuleList(
            (0): ConvModule(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (activate): ReLU(inplace=True)
            )
            (1): ConvModule(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (activate): ReLU(inplace=True)
            )
            (2): ConvModule(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (activate): ReLU(inplace=True)
            )
            (3): ConvModule(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                (activate): ReLU(inplace=True)
            )
        )
        (upsample): ConvTranspose2d(256, 256, kernel_size=(2, 2), stride=(2, 2))
        (conv_logits): Conv2d(256, 80, kernel_size=(1, 1), stride=(1, 1))
        (relu): ReLU(inplace=True)
    )
)

```

```
In [9]: # 検出器での物体検出推定demo実行  
# Use the detector to do inference  
img = 'demo/demo.jpg'  
result = inference_detector(model)
```

/content/mmdetection/mmdet/datasets/utils.py:70: UserWarning: "ImageToTensor" pipeline

```
is replaced by "DefaultFormatBundle" for batch inference. It is recommended to manually replace it in the test data pipeline in your config file.
'data pipeline in your config file.', UserWarning)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:718: UserWarning: Named tensors and all their associated APIs are an experimental feature and subject to change. Please do not use them for anything important until they are released as stable. (Triggered internally at /pytorch/c10/core/TensorImpl.h:1156.)
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, ceil_mode)
```

In [10]:

```
# Let's plot the result
show_result_pyplot(model, img, result, score_thr=0.3)
```



5.3 新たなユーザーdatasetによるファインチューニング学習のデモ

In [11]:

```
# 新たなユーザーdatasetによる学習のデモ
# 3D object アノテーションされたcocoフォーマットとは異なる（変換要の）balloon データセットをダウンロード
# download and unzip the data
!wget https://github.com/matterport/Mask_RCNN/releases/download/v2.1/balloon_dataset.zip
!unzip balloon_dataset.zip > /dev/null
```

```
--2022-06-25 10:30:06-- https://github.com/matterport/Mask_RCNN/releases/download/v2.1/balloon_dataset.zip
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/107595270/737339e2-2b83-11e8-856a-188034eb3468?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220625%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220625T103006Z&X-Amz-Expires=300&X-Amz-Signature=842d86b91b15f6873315239844a6fc87b7cad53211094179a41214e59426dbd&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=107595270&response-content-disposition=attachment%3B%20filename%3Dballoon_dataset.zip&response-content-type=application%2Foctet-stream [following]
--2022-06-25 10:30:06-- https://objects.githubusercontent.com/github-production-relea
```

```
se-asset-2e65be/107595270/737339e2-2b83-11e8-856a-188034eb3468?X-Amz-Algorithm=AWS4-HM
AC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220625%2Fus-east-1%2Fs3%2Faws4_req
uest&X-Amz-Date=20220625T103006Z&X-Amz-Expires=300&X-Amz-Signature=842d86b91b15f687331
5239844a6fc87b7cade53211094179a41214e59426dbd&X-Amz-SignedHeaders=host&actor_id=0&key_
id=0&repo_id=107595270&response-content-disposition=attachment%3B%20filename%3Dballoon_
dataset.zip&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com) ... 185.199.10
8.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com) |185.199.10
8.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38741381 (37M) [application/octet-stream]
Saving to: 'balloon_dataset.zip'

balloon_dataset.zip 100%[=====] 36.95M 15.8MB/s in 2.3s

2022-06-25 10:30:09 (15.8 MB/s) - 'balloon_dataset.zip' saved [38741381/38741381]
```

```
In [12]: # Check the directory structure of the tiny data

# Install tree first
!apt-get -q install tree
!tree balloon
```

```
Reading package lists...
Building dependency tree...
Reading state information...
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 49 not upgraded.
Need to get 40.7 kB of archives.
After this operation, 105 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tree amd64 1.7.0-5 [40.7
kB]
Fetched 40.7 kB in 0s (675 kB/s)
Selecting previously unselected package tree.
(Reading database ... 155639 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-5_amd64.deb ...
Unpacking tree (1.7.0-5) ...
Setting up tree (1.7.0-5) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
balloon
├── train
│   ├── 10464445726_6f1e3bbe6a_k.jpg
│   ├── 12037308314_e16fb3a0f7_k.jpg
│   ├── 120853323_d4788431b9_b.jpg
│   ├── 12288043903_fe1ea17a4e_k.jpg
│   ├── 12288355124_5e340d3de3_k.jpg
│   ├── 12288446656_2c6a90e6f5_k.jpg
│   ├── 126700562_8e27720147_b.jpg
│   ├── 1297451346_5b92bdac08_b.jpg
│   ├── 14321263043_b76ef054d3_k.jpg
│   ├── 145053828_e0e748717c_b.jpg
│   ├── 14666848163_8be8e37562_k.jpg
│   ├── 15290896925_884ab33fd3_k.jpg
│   ├── 15331928994_d5b82eb368_k.jpg
│   ├── 154446334_5d41cd1375_b.jpg
│   ├── 155815494_800fc9aa32_b.jpg
│   └── 15717689633_5f7f78c28e_k.jpg
```

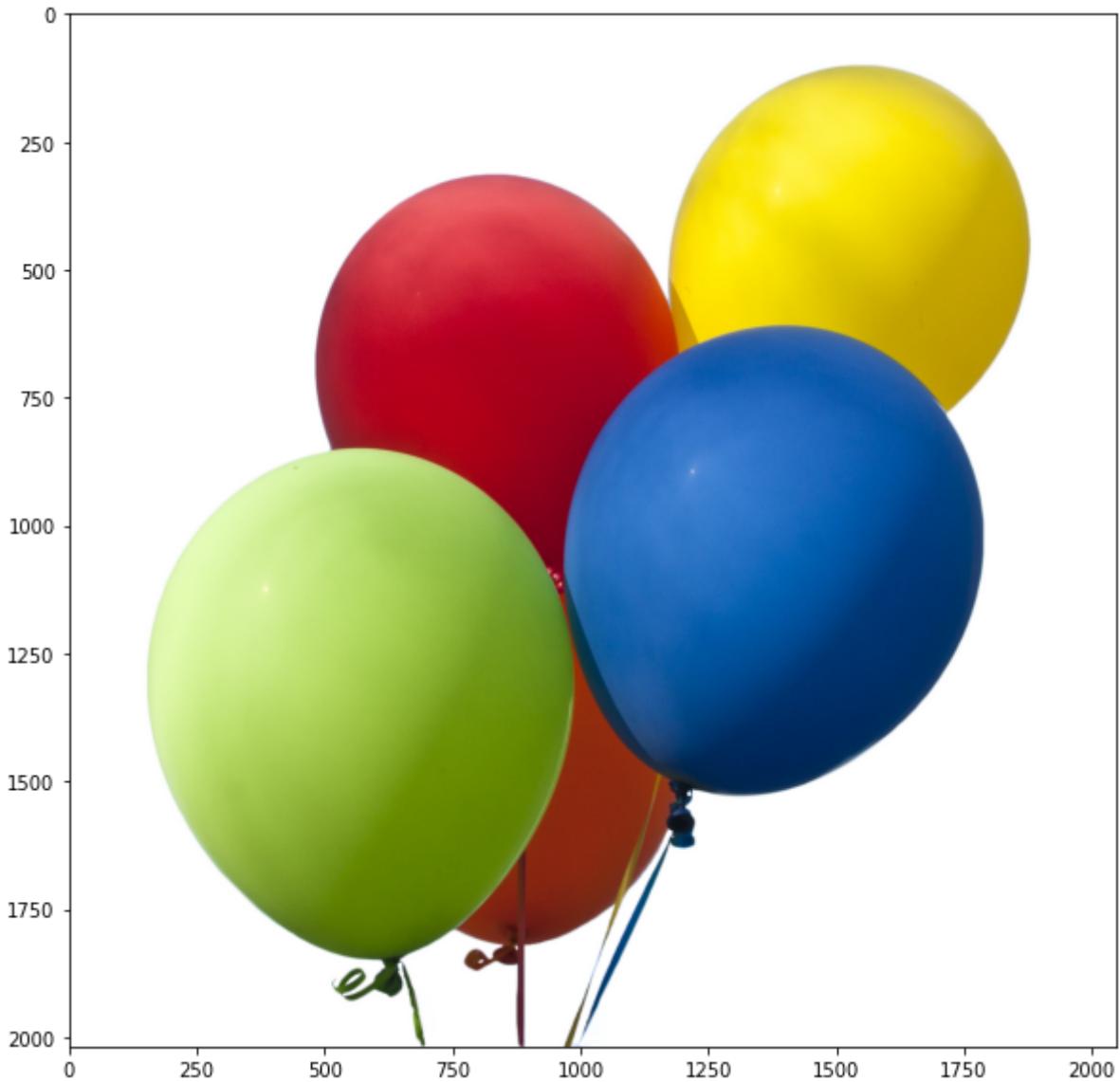
```
    ├── 16435593892_2aa8118f4a_k.jpg
    ├── 17156759330_5af4f5a5b8_k.jpg
    ├── 17178818589_16e58fc1e5_k.jpg
    ├── 18849792632_aad23ad513_k.jpg
    ├── 2311771643_f46392fcc0_b.jpg
    ├── 2354829160_3f65a6bf6f_b.jpg
    ├── 2385899600_94b68350af_b.jpg
    ├── 24362039530_b151b41a52_k.jpg
    ├── 25899693952_7c8b8b9edc_k.jpg
    ├── 2685563244_b0d5f7eb67_b.jpg
    ├── 2937599387_80e7d6e050_b.jpg
    ├── 321888854_3723b6f10b_b.jpg
    ├── 332344155_71be3a3b22_b.jpg
    ├── 3342804367_f43682bb80_b.jpg
    ├── 34020010494_e5cb88e1c4_k.jpg
    ├── 351678851_e2aeebdafdf_b.jpg
    ├── 3646097131_e3e1215843_b.jpg
    ├── 3927754171_9011487133_b.jpg
    ├── 3945575930_ce99a7e98d_b.jpg
    ├── 4057490235_2ffdf7d68b_b.jpg
    ├── 4543126482_92254ef046_b.jpg
    ├── 4552737035_3a0a4105fb_b.jpg
    ├── 485227412_e335662bb5_b.jpg
    ├── 4864857993_edb62f16ef_b.jpg
    ├── 4887227769_acd2e6127d_b.jpg
    ├── 489752654_777853a0ba_b.jpg
    ├── 4955354786_337a598e4a_b.jpg
    ├── 5013250607_26359229b6_b.jpg
    ├── 5178670692_63a4365c9c_b.jpg
    ├── 518678836_94d58e3839_b.jpg
    ├── 5253122239_38b1e7f61c_b.jpg
    ├── 53500107_d24b11b3c2_b.jpg
    ├── 5560377994_cb597a4af5_b.jpg
    ├── 5674044810_2d9e2243ff_b.jpg
    ├── 605521662_a470fef77f_b.jpg
    ├── 6483318883_21facf57cd_b.jpg
    ├── 699765866_abaad7274d_b.jpg
    ├── 7178882742_f090f3ce56_k.jpg
    ├── 7308740338_591f27b631_k.jpg
    ├── 7488015492_0583857ca0_k.jpg
    ├── 8436015314_3a678c1005_k.jpg
    ├── 873768102_7d931e5fa3_b.jpg
    ├── 8758393087_76fcfd56bd3_k.jpg
    ├── 9210739293_2b0e0d991e_k.jpg
    ├── 9330497995_4cf0438cb6_k.jpg
    └── via_region_data.json

    └── val
        ├── 14898532020_ba6199dd22_k.jpg
        ├── 16335852991_f55de7958d_k.jpg
        ├── 24631331976_defa3bb61f_k.jpg
        ├── 2917282960_06beee649a_b.jpg
        ├── 3800636873_ace2c2795f_b.jpg
        ├── 3825919971_93fb1ec581_b.jpg
        ├── 410488422_5f8991f26e_b.jpg
        ├── 4581425993_72b9b15fc0_b.jpg
        ├── 4838031651_3e7b5ea5c7_b.jpg
        ├── 5555705118_3390d70abe_b.jpg
        ├── 5603212091_2dfe16ea72_b.jpg
        ├── 6810773040_3d81036d05_k.jpg
        ├── 8053085540_a72bd21a64_k.jpg
        └── via_region_data.json
```

2 directories, 76 files

```
In [13]: # Let's take a look at the dataset image
import mmcv
import matplotlib.pyplot as plt

img = mmcv.imread('balloon/train/10464445726_6f1e3bbe6a_k.jpg')
plt.figure(figsize=(15, 10))
plt.imshow(mmcv.bgr2rgb(img))
plt.show()
```



```
In [14]: # Check the label of a single image
annotation = mmcv.load('balloon/train/via_region_data.json')
```

```
In [15]: # The annotation is a dict, and its values looks like the following
annotation['34020010494_e5cb88e1c4_k.jpg1115004']
```

```
Out[15]: {'base64_img_data': '',
 'file_attributes': {},
 'filename': '34020010494_e5cb88e1c4_k.jpg',
 'fileref': '',
 'regions': {'0': {'region_attributes': {},
 'shape_attributes': {'all_points_x': [1020,
 1000,
 994,
 1003,
 1023,
```

```
1050,  
1089,  
1134,  
1190,  
1265,  
1321,  
1361,  
1403,  
1428,  
1442,  
1445,  
1441,  
1427,  
1400,  
1361,  
1316,  
1269,  
1228,  
1198,  
1207,  
1210,  
1190,  
1177,  
1172,  
1174,  
1170,  
1153,  
1127,  
1104,  
1061,  
1032,  
1020],  
'all_points_y': [963,  
899,  
841,  
787,  
738,  
700,  
663,  
638,  
621,  
619,  
643,  
672,  
720,  
765,  
800,  
860,  
896,  
942,  
990,  
1035,  
1079,  
1112,  
1129,  
1134,  
1144,  
1153,  
1166,  
1166,  
1150,  
1136,  
1129,  
1122,
```

```

1112,
1084,
1037,
989,
963],
'name': 'polygon'}}},
'size': 1115004}

```

上記アノテーションデータをMS COCOアノテーションフォーマットに変換する必要がある。そのデータ構造は下記の通りである。

According to the above observation, each single image has a corresponding annotation containing keys filename and regions that are necessary for training. We need to read annotations of each image and convert them into COCO format as below:

```

{
  "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,
}]

```

Note: We only list the necessary keys for training, as shown above. For a full COCO format, please see [here](#).

In [16]:

```

import os.path as osp

def convert_balloon_to_coco(ann_file, out_file, image_prefix):
    data_infos = mmcv.load(ann_file)

    annotations = []
    images = []
    obj_count = 0
    for idx, v in enumerate(mmcv.track_iter_progress(data_infos.values())):
        filename = v['filename']

```

```

img_path = osp.join(image_prefix, filename)
height, width = mmcv.imread(img_path).shape[:2]

images.append(dict(
    id=idx,
    file_name=filename,
    height=height,
    width=width))

bboxes = []
labels = []
masks = []
for _, obj in v['regions'].items():
    assert not obj['region_attributes']
    obj = obj['shape_attributes']
    px = obj['all_points_x']
    py = obj['all_points_y']
    poly = [(x + 0.5, y + 0.5) for x, y in zip(px, py)]
    poly = [p for x in poly for p in x]

    x_min, y_min, x_max, y_max = (
        min(px), min(py), max(px), max(py))

    data_anno = dict(
        image_id=idx,
        id=obj['id'],
        category_id=0,
        bbox=[x_min, y_min, x_max - x_min, y_max - y_min],
        area=(x_max - x_min) * (y_max - y_min),
        segmentation=[poly],
        iscrowd=0)
    annotations.append(data_anno)
    obj['id'] += 1

coco_format_json = dict(
    images=images,
    annotations=annotations,
    categories=[{'id': 0, 'name': 'balloon'}])
mmcv.dump(coco_format_json, out_file)

```

In [17]:

```

convert_balloon_to_coco(
    'balloon/train/via_region_data.json',
    'balloon/train/annotation_coco.json',
    'balloon/train/')
convert_balloon_to_coco(
    'balloon/val/via_region_data.json',
    'balloon/val/annotation_coco.json',
    'balloon/val/')

```

```
[>>>>>>>>>>>>>>>>>>] 61/61, 32.2 task/s, elapsed: 2s, ETA:      0s
[>>>>>>>>>>>>>>>>>] 13/13, 30.7 task/s, elapsed: 0s, ETA:      0s
```

モデルをmask_rcnn_r50_caffe_fpn_mstrain-poly_1x_coco.pyに切り替え、実行環境を設定(config)する。

※Mask R-CNNアーキテクチャ、backbone network: ResNet-50, neck: FPN, MS-COCO2017での学習に対応

config環境に読み込まれる学習済みパラメタデータ（checkpointファイル）は、
mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-0.408_segm_mAP-0.37_20200504_163245-42aa3d00.pth

その学習評価結果は COCO 80クラスの判別において、bbox mAP: 40.8%, segm mAP: 37%をマークしていることが、ファイル名から推定できる。

In [18]:

```
# configの変更
from mmcv import Config
cfg = Config.fromfile('./configs/mask_rcnn/mask_rcnn_r50_caffe_fpn_mstrain-poly_1x_coco.py')
```

In [19]:

```
# modelのコンフィグレーション
from mmdet.apis import set_random_seed

# Modify dataset type and path
cfg.dataset_type = 'COCODataset'

cfg.data.test.ann_file = 'balloon/val/annotation_coco.json'
cfg.data.test.img_prefix = 'balloon/val/'
cfg.data.test.classes = ('balloon',)

cfg.data.train.ann_file = 'balloon/train/annotation_coco.json'
cfg.data.train.img_prefix = 'balloon/train/'
cfg.data.train.classes = ('balloon',)

cfg.data.val.ann_file = 'balloon/val/annotation_coco.json'
cfg.data.val.img_prefix = 'balloon/val/'
cfg.data.val.classes = ('balloon',)

# modify num classes of the model in box head and mask head
cfg.model.roi_head.bbox_head.num_classes = 1
cfg.model.roi_head.mask_head.num_classes = 1

# We can still use the pre-trained Mask RCNN model to obtain a higher performance
cfg.load_from = 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-0.432.pth'

# Set up working dir to save files and logs.
cfg.work_dir = './tutorial_exps'

# The original learning rate (LR) is set for 8-GPU training.
# We divide it by 8 since we only use one GPU.
cfg.optimizer.lr = 0.02 / 8
cfg.optimizer_config.warmup = None
cfg.log_config.interval = 10

# We can set the evaluation interval to reduce the evaluation times
cfg.evaluation.interval = 12
# We can set the checkpoint saving interval to reduce the storage cost
cfg.checkpoint_config.interval = 12

# Set seed thus the results are more reproducible
cfg.seed = 0
set_random_seed(0, deterministic=False)
cfg.gpu_ids = range(1)

# We can also use tensorboard to log the training process
cfg.log_config.hooks = [
    dict(type='TextLoggerHook'),
    dict(type='TensorboardLoggerHook')
]

# We can initialize the logger for training and have a look
# at the final config used for training
print(f'Config:\n{cfg.pretty_text}'')
```

```
Config:  
model = dict(  
    type='MaskRCNN',  
    backbone=dict(  
        type='ResNet',  
        depth=50,  
        num_stages=4,  
        out_indices=(0, 1, 2, 3),  
        frozen_stages=1,  
        norm_cfg=dict(type='BN', requires_grad=False),  
        norm_eval=True,  
        style='caffe',  
        init_cfg=dict(  
            type='Pretrained',  
            checkpoint='open-mmlab://detectron2/resnet50_caffe')),  
    neck=dict(  
        type='FPN',  
        in_channels=[256, 512, 1024, 2048],  
        out_channels=256,  
        num_outs=5),  
    rpn_head=dict(  
        type='RPNHead',  
        in_channels=256,  
        feat_channels=256,  
        anchor_generator=dict(  
            type='AnchorGenerator',  
            scales=[8],  
            ratios=[0.5, 1.0, 2.0],  
            strides=[4, 8, 16, 32, 64]),  
        bbox_coder=dict(  
            type='DeltaXYWHBBoxCoder',  
            target_means=[0.0, 0.0, 0.0, 0.0],  
            target_stds=[1.0, 1.0, 1.0, 1.0]),  
        loss_cls=dict(  
            type='CrossEntropyLoss', use_sigmoid=True, loss_weight=1.0),  
        loss_bbox=dict(type='L1Loss', loss_weight=1.0)),  
    roi_head=dict(  
        type='StandardRoIHead',  
        bbox_roi_extractor=dict(  
            type='SingleRoIExtractor',  
            roi_layer=dict(type='RoIAlign', output_size=7, sampling_ratio=0),  
            out_channels=256,  
            featmap_strides=[4, 8, 16, 32]),  
        bbox_head=dict(  
            type='Shared2FCBBoxHead',  
            in_channels=256,  
            fc_out_channels=1024,  
            roi_feat_size=7,  
            num_classes=1,  
            bbox_coder=dict(  
                type='DeltaXYWHBBoxCoder',  
                target_means=[0.0, 0.0, 0.0, 0.0],  
                target_stds=[0.1, 0.1, 0.2, 0.2]),  
            reg_class_agnostic=False,  
            loss_cls=dict(  
                type='CrossEntropyLoss', use_sigmoid=False, loss_weight=1.0),  
            loss_bbox=dict(type='L1Loss', loss_weight=1.0)),  
        mask_roi_extractor=dict(  
            type='SingleRoIExtractor',  
            roi_layer=dict(type='RoIAlign', output_size=14, sampling_ratio=0),  
            out_channels=256,  
            featmap_strides=[4, 8, 16, 32]),  
        mask_head=dict(  
            type='FCNMaskHead',
```

```

        num_convs=4,
        in_channels=256,
        conv_out_channels=256,
        num_classes=1,
        loss_mask=dict(
            type='CrossEntropyLoss', use_mask=True, loss_weight=1.0))),
train_cfg=dict(
    rpn=dict(
        assigner=dict(
            type='MaxIoUAssigner',
            pos_iou_thr=0.7,
            neg_iou_thr=0.3,
            min_pos_iou=0.3,
            match_low_quality=True,
            ignore_iou_thr=-1),
        sampler=dict(
            type='RandomSampler',
            num=256,
            pos_fraction=0.5,
            neg_pos_ub=-1,
            add_gt_as_proposals=False),
        allowed_border=-1,
        pos_weight=-1,
        debug=False),
    rpn_proposal=dict(
        nms_pre=2000,
        max_per_img=1000,
        nms=dict(type='nms', iou_threshold=0.7),
        min_bbox_size=0),
    rcnn=dict(
        assigner=dict(
            type='MaxIoUAssigner',
            pos_iou_thr=0.5,
            neg_iou_thr=0.5,
            min_pos_iou=0.5,
            match_low_quality=True,
            ignore_iou_thr=-1),
        sampler=dict(
            type='RandomSampler',
            num=512,
            pos_fraction=0.25,
            neg_pos_ub=-1,
            add_gt_as_proposals=True),
        mask_size=28,
        pos_weight=-1,
        debug=False)),
    test_cfg=dict(
        rpn=dict(
            nms_pre=1000,
            max_per_img=1000,
            nms=dict(type='nms', iou_threshold=0.7),
            min_bbox_size=0),
        rcnn=dict(
            score_thr=0.05,
            nms=dict(type='nms', iou_threshold=0.5),
            max_per_img=100,
            mask_thr_binary=0.5)))
dataset_type = 'COCODataset'
data_root = 'data/coco/'
img_norm_cfg = dict(
    mean=[103.53, 116.28, 123.675], std=[1.0, 1.0, 1.0], to_rgb=False)
train_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(

```

```

        type='LoadAnnotations',
        with_bbox=True,
        with_mask=True,
        poly2mask=False),
    dict(
        type='Resize',
        img_scale=[(1333, 640), (1333, 672), (1333, 704), (1333, 736),
                   (1333, 768), (1333, 800)],
        multiscale_mode='value',
        keep_ratio=True),
    dict(type='RandomFlip', flip_ratio=0.5),
    dict(
        type='Normalize',
        mean=[103.53, 116.28, 123.675],
        std=[1.0, 1.0, 1.0],
        to_rgb=False),
    dict(type='Pad', size_divisor=32),
    dict(type='DefaultFormatBundle'),
    dict(type='Collect', keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
]
test_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(
        type='MultiScaleFlipAug',
        img_scale=(1333, 800),
        flip=False,
        transforms=[
            dict(type='Resize', keep_ratio=True),
            dict(type='RandomFlip'),
            dict(
                type='Normalize',
                mean=[103.53, 116.28, 123.675],
                std=[1.0, 1.0, 1.0],
                to_rgb=False),
            dict(type='Pad', size_divisor=32),
            dict(type='ImageToTensor', keys=['img']),
            dict(type='Collect', keys=['img'])
        ])
]
data = dict(
    samples_per_gpu=2,
    workers_per_gpu=2,
    train=dict(
        type='CocoDataset',
        ann_file='balloon/train/annotation_coco.json',
        img_prefix='balloon/train/',
        pipeline=[
            dict(type='LoadImageFromFile'),
            dict(
                type='LoadAnnotations',
                with_bbox=True,
                with_mask=True,
                poly2mask=False),
            dict(
                type='Resize',
                img_scale=[(1333, 640), (1333, 672), (1333, 704), (1333, 736),
                           (1333, 768), (1333, 800)],
                multiscale_mode='value',
                keep_ratio=True),
            dict(type='RandomFlip', flip_ratio=0.5),
            dict(
                type='Normalize',
                mean=[103.53, 116.28, 123.675],
                std=[1.0, 1.0, 1.0],
                to_rgb=False)
        ])
)

```

```

        to_rgb=False),
        dict(type='Pad', size_divisor=32),
        dict(type='DefaultFormatBundle'),
        dict(
            type='Collect',
            keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
    ],
    classes=('balloon', )),
val=dict(
    type='CocoDataset',
    ann_file='balloon/val/annotation_coco.json',
    img_prefix='balloon/val/',
    pipeline=[
        dict(type='LoadImageFromFile'),
        dict(
            type='MultiScaleFlipAug',
            img_scale=(1333, 800),
            flip=False,
            transforms=[
                dict(type='Resize', keep_ratio=True),
                dict(type='RandomFlip'),
                dict(
                    type='Normalize',
                    mean=[103.53, 116.28, 123.675],
                    std=[1.0, 1.0, 1.0],
                    to_rgb=False),
                dict(type='Pad', size_divisor=32),
                dict(type='ImageToTensor', keys=['img']),
                dict(type='Collect', keys=['img'])
            ])
    ],
    classes=('balloon', )),
test=dict(
    type='CocoDataset',
    ann_file='balloon/val/annotation_coco.json',
    img_prefix='balloon/val/',
    pipeline=[
        dict(type='LoadImageFromFile'),
        dict(
            type='MultiScaleFlipAug',
            img_scale=(1333, 800),
            flip=False,
            transforms=[
                dict(type='Resize', keep_ratio=True),
                dict(type='RandomFlip'),
                dict(
                    type='Normalize',
                    mean=[103.53, 116.28, 123.675],
                    std=[1.0, 1.0, 1.0],
                    to_rgb=False),
                dict(type='Pad', size_divisor=32),
                dict(type='ImageToTensor', keys=['img']),
                dict(type='Collect', keys=['img'])
            ])
    ],
    classes=('balloon', )))
evaluation = dict(metric=['bbox', 'segm'], interval=12)
optimizer = dict(type='SGD', lr=0.0025, momentum=0.9, weight_decay=0.0001)
optimizer_config = dict(grad_clip=None)
lr_config = dict(
    policy='step',
    warmup=None,
    warmup_iters=500,
    warmup_ratio=0.001,
)

```

```

        step=[8, 11])
runner = dict(type='EpochBasedRunner', max_epochs=12)
checkpoint_config = dict(interval=12)
log_config = dict(
    interval=10,
    hooks=[dict(type='TextLoggerHook'),
           dict(type='TensorboardLoggerHook')])
custom_hooks = [dict(type='NumClassCheckHook')]
dist_params = dict(backend='nccl')
log_level = 'INFO'
load_from = 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-0.408__segm_mAP-0.37_20200504_163245-42aa3d00.pth'
resume_from = None
workflow = [('train', 1)]
opencv_num_threads = 0
mp_start_method = 'fork'
work_dir = './tutorial_exps'
seed = 0
gpu_ids = range(0, 1)

```

In [20]:

```

# カスタマイズしたDetectorの学習
from mmdet.datasets import build_dataset
from mmdet.models import build_detector
from mmdet.apis import train_detector

# Build dataset
datasets = [build_dataset(cfg.data.train)]

# Build the detector
model = build_detector(cfg.model)

# Add an attribute for visualization convenience
model.CLASSES = datasets[0].CLASSES

# Create work_dir
mmcv.mkdir_or_exist(osp.abspath(cfg.work_dir))
train_detector(model, datasets, cfg, distributed=False, validate=True)

```

loading annotations into memory...

Done (t=0.01s)

creating index...

index created!

2022-06-25 10:32:08,553 - mmdet - INFO - load checkpoint from local path: checkpoints/mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-0.408__segm_mAP-0.37_20200504_163245-42aa3d00.pth

2022-06-25 10:32:08,700 - mmdet - WARNING - The model and loaded state dict do not match exactly

size mismatch for roi_head.bbox_head.fc_cls.weight: copying a param with shape torch.Size([81, 1024]) from checkpoint, the shape in current model is torch.Size([2, 1024]).
size mismatch for roi_head.bbox_head.fc_cls.bias: copying a param with shape torch.Size([81]) from checkpoint, the shape in current model is torch.Size([2]).

size mismatch for roi_head.bbox_head.fc_reg.weight: copying a param with shape torch.Size([320, 1024]) from checkpoint, the shape in current model is torch.Size([4, 1024]).
size mismatch for roi_head.bbox_head.fc_reg.bias: copying a param with shape torch.Size([320]) from checkpoint, the shape in current model is torch.Size([4]).

size mismatch for roi_head.mask_head.conv_logits.weight: copying a param with shape torch.Size([80, 256, 1, 1]) from checkpoint, the shape in current model is torch.Size([1, 256, 1, 1]).

size mismatch for roi_head.mask_head.conv_logits.bias: copying a param with shape torch.Size([80]) from checkpoint, the shape in current model is torch.Size([1]).

```
2022-06-25 10:32:08,710 - mmdet - INFO - Start running, host: root@5fa4014c792d, work_
dir: /content/mmdetection/tutorial_exps
2022-06-25 10:32:08,712 - mmdet - INFO - Hooks will be executed in the following orde
r:
before_run:
(VERY_HIGH    ) StepLrUpdaterHook
(NORMAL       ) CheckpointHook
(LOW          ) EvalHook
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
before_train_epoch:
(VERY_HIGH    ) StepLrUpdaterHook
(NORMAL       ) NumClassCheckHook
(LOW          ) IterTimerHook
(LOW          ) EvalHook
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
before_train_iter:
(VERY_HIGH    ) StepLrUpdaterHook
(LOW          ) IterTimerHook
(LOW          ) EvalHook
-----
after_train_iter:
(ABOVE_NORMAL) OptimizerHook
(NORMAL       ) CheckpointHook
(LOW          ) IterTimerHook
(LOW          ) EvalHook
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
after_train_epoch:
(NORMAL       ) CheckpointHook
(LOW          ) EvalHook
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
before_val_epoch:
(NORMAL       ) NumClassCheckHook
(LOW          ) IterTimerHook
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
before_val_iter:
(LOW          ) IterTimerHook
-----
after_val_iter:
(LOW          ) IterTimerHook
-----
after_val_epoch:
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
after_run:
(VERY_LOW     ) TextLoggerHook
(VERY_LOW     ) TensorboardLoggerHook
-----
2022-06-25 10:32:08,716 - mmdet - INFO - workflow: [('train', 1)], max: 12 epochs
2022-06-25 10:32:08,719 - mmdet - INFO - Checkpoints will be saved to /content/mmdete
ction/tutorial_exps by HardDiskBackend.
loading annotations into memory...
Done (t=0.00s)
```

```
creating index...
index created!
```

```
2022-06-25 10:32:24, 549 - mmdet - INFO - Epoch [1][10/31]      lr: 2.500e-03, eta: 0: 05:32, time: 0.919, data_time: 0.246, memory: 3641, loss_rpn_cls: 0.0300, loss_rpn_bbo_x: 0.0167, loss_cls: 0.3624, acc: 84.1895, loss_bbox: 0.4058, loss_mask: 0.4689, loss: 1.2838
2022-06-25 10:32:27, 837 - mmdet - INFO - Epoch [1][20/31]      lr: 2.500e-03, eta: 0: 03:39, time: 0.328, data_time: 0.024, memory: 3691, loss_rpn_cls: 0.0352, loss_rpn_bbo_x: 0.0144, loss_cls: 0.1400, acc: 95.9668, loss_bbox: 0.3011, loss_mask: 0.1147, loss: 0.6054
2022-06-25 10:32:31, 002 - mmdet - INFO - Epoch [1][30/31]      lr: 2.500e-03, eta: 0: 02:58, time: 0.316, data_time: 0.023, memory: 3691, loss_rpn_cls: 0.0181, loss_rpn_bbo_x: 0.0102, loss_cls: 0.0674, acc: 97.4805, loss_bbox: 0.1337, loss_mask: 0.1237, loss: 0.3531
2022-06-25 10:32:36, 991 - mmdet - INFO - Epoch [2][10/31]      lr: 2.500e-03, eta: 0: 02:51, time: 0.556, data_time: 0.243, memory: 3691, loss_rpn_cls: 0.0094, loss_rpn_bbo_x: 0.0074, loss_cls: 0.0526, acc: 97.9492, loss_bbox: 0.1029, loss_mask: 0.0650, loss: 0.2371
2022-06-25 10:32:40, 244 - mmdet - INFO - Epoch [2][20/31]      lr: 2.500e-03, eta: 0: 02:34, time: 0.327, data_time: 0.024, memory: 3691, loss_rpn_cls: 0.0141, loss_rpn_bbo_x: 0.0128, loss_cls: 0.0731, acc: 97.5098, loss_bbox: 0.1277, loss_mask: 0.1085, loss: 0.3362
2022-06-25 10:32:43, 559 - mmdet - INFO - Epoch [2][30/31]      lr: 2.500e-03, eta: 0: 02:21, time: 0.331, data_time: 0.021, memory: 3691, loss_rpn_cls: 0.0131, loss_rpn_bbo_x: 0.0120, loss_cls: 0.0808, acc: 97.2363, loss_bbox: 0.1312, loss_mask: 0.1081, loss: 0.3452
2022-06-25 10:32:49, 561 - mmdet - INFO - Epoch [3][10/31]      lr: 2.500e-03, eta: 0: 02:18, time: 0.551, data_time: 0.234, memory: 3691, loss_rpn_cls: 0.0092, loss_rpn_bbo_x: 0.0053, loss_cls: 0.0431, acc: 98.5938, loss_bbox: 0.0647, loss_mask: 0.0592, loss: 0.1816
2022-06-25 10:32:52, 796 - mmdet - INFO - Epoch [3][20/31]      lr: 2.500e-03, eta: 0: 02:09, time: 0.323, data_time: 0.022, memory: 3691, loss_rpn_cls: 0.0041, loss_rpn_bbo_x: 0.0110, loss_cls: 0.0614, acc: 97.6660, loss_bbox: 0.1055, loss_mask: 0.0856, loss: 0.2675
2022-06-25 10:32:56, 193 - mmdet - INFO - Epoch [3][30/31]      lr: 2.500e-03, eta: 0: 02:01, time: 0.339, data_time: 0.021, memory: 3691, loss_rpn_cls: 0.0043, loss_rpn_bbo_x: 0.0141, loss_cls: 0.0776, acc: 96.9336, loss_bbox: 0.1397, loss_mask: 0.0870, loss: 0.3227
2022-06-25 10:33:02, 272 - mmdet - INFO - Epoch [4][10/31]      lr: 2.500e-03, eta: 0: 01:58, time: 0.563, data_time: 0.241, memory: 3691, loss_rpn_cls: 0.0056, loss_rpn_bbo_x: 0.0100, loss_cls: 0.0677, acc: 97.4219, loss_bbox: 0.1148, loss_mask: 0.0772, loss: 0.2753
2022-06-25 10:33:05, 444 - mmdet - INFO - Epoch [4][20/31]      lr: 2.500e-03, eta: 0: 01:51, time: 0.317, data_time: 0.021, memory: 3691, loss_rpn_cls: 0.0042, loss_rpn_bbo_x: 0.0094, loss_cls: 0.0524, acc: 98.0273, loss_bbox: 0.0775, loss_mask: 0.0592, loss: 0.2028
2022-06-25 10:33:08, 925 - mmdet - INFO - Epoch [4][30/31]      lr: 2.500e-03, eta: 0: 01:45, time: 0.347, data_time: 0.022, memory: 3795, loss_rpn_cls: 0.0045, loss_rpn_bbo_x: 0.0102, loss_cls: 0.0621, acc: 97.5195, loss_bbox: 0.0975, loss_mask: 0.0706, loss: 0.2449
2022-06-25 10:33:14, 944 - mmdet - INFO - Epoch [5][10/31]      lr: 2.500e-03, eta: 0: 01:42, time: 0.559, data_time: 0.235, memory: 3795, loss_rpn_cls: 0.0031, loss_rpn_bbo_x: 0.0084, loss_cls: 0.0431, acc: 98.4277, loss_bbox: 0.0863, loss_mask: 0.0633, loss: 0.2042
2022-06-25 10:33:18, 208 - mmdet - INFO - Epoch [5][20/31]      lr: 2.500e-03, eta: 0: 01:36, time: 0.327, data_time: 0.022, memory: 3856, loss_rpn_cls: 0.0056, loss_rpn_bbo_x: 0.0102, loss_cls: 0.0505, acc: 98.0371, loss_bbox: 0.0782, loss_mask: 0.0568, loss: 0.2013
2022-06-25 10:33:21, 604 - mmdet - INFO - Epoch [5][30/31]      lr: 2.500e-03, eta: 0: 01:31, time: 0.336, data_time: 0.020, memory: 3857, loss_rpn_cls: 0.0027, loss_rpn_bbo_x: 0.0120, loss_cls: 0.0472, acc: 98.2520, loss_bbox: 0.0880, loss_mask: 0.0813, loss: 0.2312
2022-06-25 10:33:27, 625 - mmdet - INFO - Epoch [6][10/31]      lr: 2.500e-03, eta: 0: 01:27, time: 0.557, data_time: 0.240, memory: 3857, loss_rpn_cls: 0.0033, loss_rpn_bbo_x: 0.0102, loss_cls: 0.0524, acc: 98.4277, loss_bbox: 0.0863, loss_mask: 0.0633, loss: 0.2042
```

x: 0.0059, loss_cls: 0.0341, acc: 98.7207, loss_bbox: 0.0613, loss_mask: 0.0542, loss: 0.1589
2022-06-25 10:33:30, 834 - mmdet - INFO - Epoch [6][20/31] lr: 2.500e-03, eta: 0:01:22, time: 0.324, data_time: 0.022, memory: 3857, loss_rpn_cls: 0.0061, loss_rpn_bbo
x: 0.0102, loss_cls: 0.0548, acc: 97.7734, loss_bbox: 0.0930, loss_mask: 0.0867, loss: 0.2509
2022-06-25 10:33:34, 100 - mmdet - INFO - Epoch [6][30/31] lr: 2.500e-03, eta: 0:01:17, time: 0.325, data_time: 0.019, memory: 3857, loss_rpn_cls: 0.0026, loss_rpn_bbo
x: 0.0094, loss_cls: 0.0440, acc: 98.3301, loss_bbox: 0.0761, loss_mask: 0.0662, loss: 0.1983
2022-06-25 10:33:40, 094 - mmdet - INFO - Epoch [7][10/31] lr: 2.500e-03, eta: 0:01:13, time: 0.550, data_time: 0.235, memory: 3865, loss_rpn_cls: 0.0037, loss_rpn_bbo
x: 0.0063, loss_cls: 0.0268, acc: 99.0137, loss_bbox: 0.0506, loss_mask: 0.0590, loss: 0.1464
2022-06-25 10:33:43, 369 - mmdet - INFO - Epoch [7][20/31] lr: 2.500e-03, eta: 0:01:08, time: 0.330, data_time: 0.024, memory: 3865, loss_rpn_cls: 0.0010, loss_rpn_bbo
x: 0.0054, loss_cls: 0.0341, acc: 98.6523, loss_bbox: 0.0605, loss_mask: 0.0544, loss: 0.1554
2022-06-25 10:33:46, 867 - mmdet - INFO - Epoch [7][30/31] lr: 2.500e-03, eta: 0:01:04, time: 0.348, data_time: 0.021, memory: 3865, loss_rpn_cls: 0.0032, loss_rpn_bbo
x: 0.0138, loss_cls: 0.0588, acc: 97.6270, loss_bbox: 0.1037, loss_mask: 0.0842, loss: 0.2637
2022-06-25 10:33:52, 941 - mmdet - INFO - Epoch [8][10/31] lr: 2.500e-03, eta: 0:01:00, time: 0.564, data_time: 0.236, memory: 3865, loss_rpn_cls: 0.0013, loss_rpn_bbo
x: 0.0069, loss_cls: 0.0349, acc: 98.7305, loss_bbox: 0.0694, loss_mask: 0.0683, loss: 0.1808
2022-06-25 10:33:56, 217 - mmdet - INFO - Epoch [8][20/31] lr: 2.500e-03, eta: 0:00:55, time: 0.330, data_time: 0.025, memory: 3865, loss_rpn_cls: 0.0020, loss_rpn_bbo
x: 0.0067, loss_cls: 0.0362, acc: 98.5156, loss_bbox: 0.0678, loss_mask: 0.0628, loss: 0.1756
2022-06-25 10:33:59, 570 - mmdet - INFO - Epoch [8][30/31] lr: 2.500e-03, eta: 0:00:51, time: 0.334, data_time: 0.021, memory: 3865, loss_rpn_cls: 0.0018, loss_rpn_bbo
x: 0.0099, loss_cls: 0.0364, acc: 98.5742, loss_bbox: 0.0820, loss_mask: 0.0657, loss: 0.1959
2022-06-25 10:34:05, 508 - mmdet - INFO - Epoch [9][10/31] lr: 2.500e-04, eta: 0:00:47, time: 0.551, data_time: 0.234, memory: 3865, loss_rpn_cls: 0.0010, loss_rpn_bbo
x: 0.0105, loss_cls: 0.0338, acc: 98.7695, loss_bbox: 0.0698, loss_mask: 0.0516, loss: 0.1667
2022-06-25 10:34:08, 798 - mmdet - INFO - Epoch [9][20/31] lr: 2.500e-04, eta: 0:00:42, time: 0.329, data_time: 0.022, memory: 3865, loss_rpn_cls: 0.0027, loss_rpn_bbo
x: 0.0081, loss_cls: 0.0408, acc: 98.4180, loss_bbox: 0.0697, loss_mask: 0.0568, loss: 0.1782
2022-06-25 10:34:12, 120 - mmdet - INFO - Epoch [9][30/31] lr: 2.500e-04, eta: 0:00:38, time: 0.331, data_time: 0.020, memory: 3865, loss_rpn_cls: 0.0014, loss_rpn_bbo
x: 0.0075, loss_cls: 0.0289, acc: 98.7891, loss_bbox: 0.0611, loss_mask: 0.0689, loss: 0.1677
2022-06-25 10:34:18, 098 - mmdet - INFO - Epoch [10][10/31] lr: 2.500e-04, eta: 0:00:34, time: 0.555, data_time: 0.240, memory: 3865, loss_rpn_cls: 0.0022, loss_rpn_bbo
x: 0.0068, loss_cls: 0.0291, acc: 98.8477, loss_bbox: 0.0551, loss_mask: 0.0566, loss: 0.1497
2022-06-25 10:34:21, 469 - mmdet - INFO - Epoch [10][20/31] lr: 2.500e-04, eta: 0:00:29, time: 0.337, data_time: 0.023, memory: 3865, loss_rpn_cls: 0.0017, loss_rpn_bbo
x: 0.0102, loss_cls: 0.0475, acc: 98.0078, loss_bbox: 0.0810, loss_mask: 0.0681, loss: 0.2086
2022-06-25 10:34:24, 707 - mmdet - INFO - Epoch [10][30/31] lr: 2.500e-04, eta: 0:00:25, time: 0.323, data_time: 0.021, memory: 3865, loss_rpn_cls: 0.0014, loss_rpn_bbo
x: 0.0041, loss_cls: 0.0237, acc: 99.2090, loss_bbox: 0.0445, loss_mask: 0.0478, loss: 0.1215
2022-06-25 10:34:30, 786 - mmdet - INFO - Epoch [11][10/31] lr: 2.500e-04, eta: 0:00:21, time: 0.565, data_time: 0.246, memory: 3865, loss_rpn_cls: 0.0017, loss_rpn_bbo
x: 0.0125, loss_cls: 0.0428, acc: 98.2910, loss_bbox: 0.0775, loss_mask: 0.0668, loss: 0.2013
2022-06-25 10:34:34, 167 - mmdet - INFO - Epoch [11][20/31] lr: 2.500e-04, eta: 0:00:17, time: 0.338, data_time: 0.023, memory: 3865, loss_rpn_cls: 0.0010, loss_rpn_bbo

```
x: 0.0050, loss_cls: 0.0277, acc: 99.0527, loss_bbox: 0.0577, loss_mask: 0.0586, loss: 0.1501
2022-06-25 10:34:37,446 - mmdet - INFO - Epoch [11][30/31]      lr: 2.500e-04, eta: 0: 00:12, time: 0.325, data_time: 0.021, memory: 3865, loss_rpn_cls: 0.0007, loss_rpn_bbo
x: 0.0037, loss_cls: 0.0246, acc: 99.1113, loss_bbox: 0.0443, loss_mask: 0.0527, loss: 0.1260
2022-06-25 10:34:43,549 - mmdet - INFO - Epoch [12][10/31]      lr: 2.500e-05, eta: 0: 00:08, time: 0.569, data_time: 0.242, memory: 3865, loss_rpn_cls: 0.0008, loss_rpn_bbo
x: 0.0073, loss_cls: 0.0343, acc: 98.6523, loss_bbox: 0.0624, loss_mask: 0.0602, loss: 0.1650
2022-06-25 10:34:46,883 - mmdet - INFO - Epoch [12][20/31]      lr: 2.500e-05, eta: 0: 00:04, time: 0.334, data_time: 0.023, memory: 3865, loss_rpn_cls: 0.0011, loss_rpn_bbo
x: 0.0067, loss_cls: 0.0394, acc: 98.3691, loss_bbox: 0.0703, loss_mask: 0.0654, loss: 0.1828
2022-06-25 10:34:50,111 - mmdet - INFO - Epoch [12][30/31]      lr: 2.500e-05, eta: 0: 00:00, time: 0.322, data_time: 0.021, memory: 3865, loss_rpn_cls: 0.0016, loss_rpn_bbo
x: 0.0055, loss_cls: 0.0225, acc: 99.2383, loss_bbox: 0.0384, loss_mask: 0.0484, loss: 0.1164
2022-06-25 10:34:50,425 - mmdet - INFO - Saving checkpoint at 12 epochs
[>>>>>>>>>>>>>>>>>>] 13/13, 5.6 task/s, elapsed: 2s, ETA:      0s
```

2022-06-25 10:34:55,015 - mmdet - INFO - Evaluating bbox...

2022-06-25 10:34:55,056 - mmdet - INFO -

Average Precision (AP) @ [IoU=0.50:0.95 area= all maxDets=100]	= 0.799
Average Precision (AP) @ [IoU=0.50 area= all maxDets=1000]	= 0.898
Average Precision (AP) @ [IoU=0.75 area= all maxDets=1000]	= 0.888
Average Precision (AP) @ [IoU=0.50:0.95 area= small maxDets=1000]	= 0.303
Average Precision (AP) @ [IoU=0.50:0.95 area=medium maxDets=1000]	= 0.628
Average Precision (AP) @ [IoU=0.50:0.95 area= large maxDets=1000]	= 0.881
Average Recall (AR) @ [IoU=0.50:0.95 area= all maxDets=100]	= 0.840
Average Recall (AR) @ [IoU=0.50:0.95 area= all maxDets=300]	= 0.840
Average Recall (AR) @ [IoU=0.50:0.95 area= all maxDets=1000]	= 0.840
Average Recall (AR) @ [IoU=0.50:0.95 area= small maxDets=1000]	= 0.300
Average Recall (AR) @ [IoU=0.50:0.95 area=medium maxDets=1000]	= 0.725
Average Recall (AR) @ [IoU=0.50:0.95 area= large maxDets=1000]	= 0.908

2022-06-25 10:34:55,058 - mmdet - INFO - Evaluating segm...

Loading and preparing results...

DONE (t=0.00s)

creating index...

index created!

Running per image evaluation...

Evaluate annotation type *bbox*

DONE (t=0.02s).

Accumulating evaluation results...

DONE (t=0.01s).

Loading and preparing results...

DONE (t=0.00s)

creating index...

index created!

Running per image evaluation...

Evaluate annotation type *segm*

DONE (t=0.03s).

Accumulating evaluation results...

/content/mmdetection/mmdet/datasets/coco.py:474: UserWarning: The key "bbox" is deleted for more accurate mask AP of small/medium/large instances since v2.12.0. This does not change the overall mAP calculation.

UserWarning)

/usr/local/lib/python3.7/dist-packages/pycocotools/cocoeval.py:378: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```

tp_sum = np.cumsum(tps, axis=1).astype(dtype=np.float)
2022-06-25 10:34:55,116 - mmdet - INFO -
Average Precision (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.811
Average Precision (AP) @[ IoU=0.50      | area=   all | maxDets=1000] = 0.882
Average Precision (AP) @[ IoU=0.75      | area=   all | maxDets=1000] = 0.882
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.404
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.655
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.879
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.852
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.852
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000] = 0.852
Average Recall    (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.400
Average Recall    (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.792
Average Recall    (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.897

2022-06-25 10:34:55,118 - mmdet - INFO - Epoch(val) [12][13]    bbox_mAP: 0.7990, bbox
_mAP_50: 0.8980, bbox_mAP_75: 0.8880, bbox_mAP_s: 0.3030, bbox_mAP_m: 0.6280, bbox_mAP
_l: 0.8810, bbox_mAP_copypaste: 0.799 0.898 0.888 0.303 0.628 0.881, segm_mAP: 0.8110,
segm_mAP_50: 0.8820, segm_mAP_75: 0.8820, segm_mAP_s: 0.4040, segm_mAP_m: 0.6550, segm
_mAP_l: 0.8790, segm_mAP_copypaste: 0.811 0.882 0.882 0.404 0.655 0.879
DONE (t=0.02s).

```

5.4 学習プロセスのlogの解釈

このデモでは、学習データが少ないので、学習済みのMask R-CNN modelをfineチューニングした。もともとcoco datasetを使って80クラスの識別クラスであったが今回はballoon識別1クラスに減らしている。このため、事前学習の最後の全結合層FCの重みパラメタは使われない。同じく事前学習済みのmask_head.conv_logitsの重みも使われない。

学習後の評価はデフォルトのCOCO-style evaluationで行われている。

valDatasetに対するbboxAP:79.9, maskAP81.1という高いスコアが出ている。

In [21]:

```

# load tensorboard in colab
%load_ext tensorboard

# see curves in tensorboard
%tensorboard --logdir ./tutorial_exps

```

```

/usr/lib/python3.7/subprocess.py:883: ResourceWarning: subprocess 930 is still running
  ResourceWarning, source=self)
ResourceWarning: Enable tracemalloc to get the object allocation traceback

```

5.5 フайнチューニング後の動作確認

新たに学習させたバルーンのインスタンスセグメンテーションdetectorについて、画像データを与えて動作確認を行う。

In [22]:

```

# 学習したDetectorのTest
img = mmcv.imread('balloon/train/7178882742_f090f3ce56_k.jpg')

model.cfg = cfg
result = inference_detector(model, img)
show_result_pyplot(model, img, result)

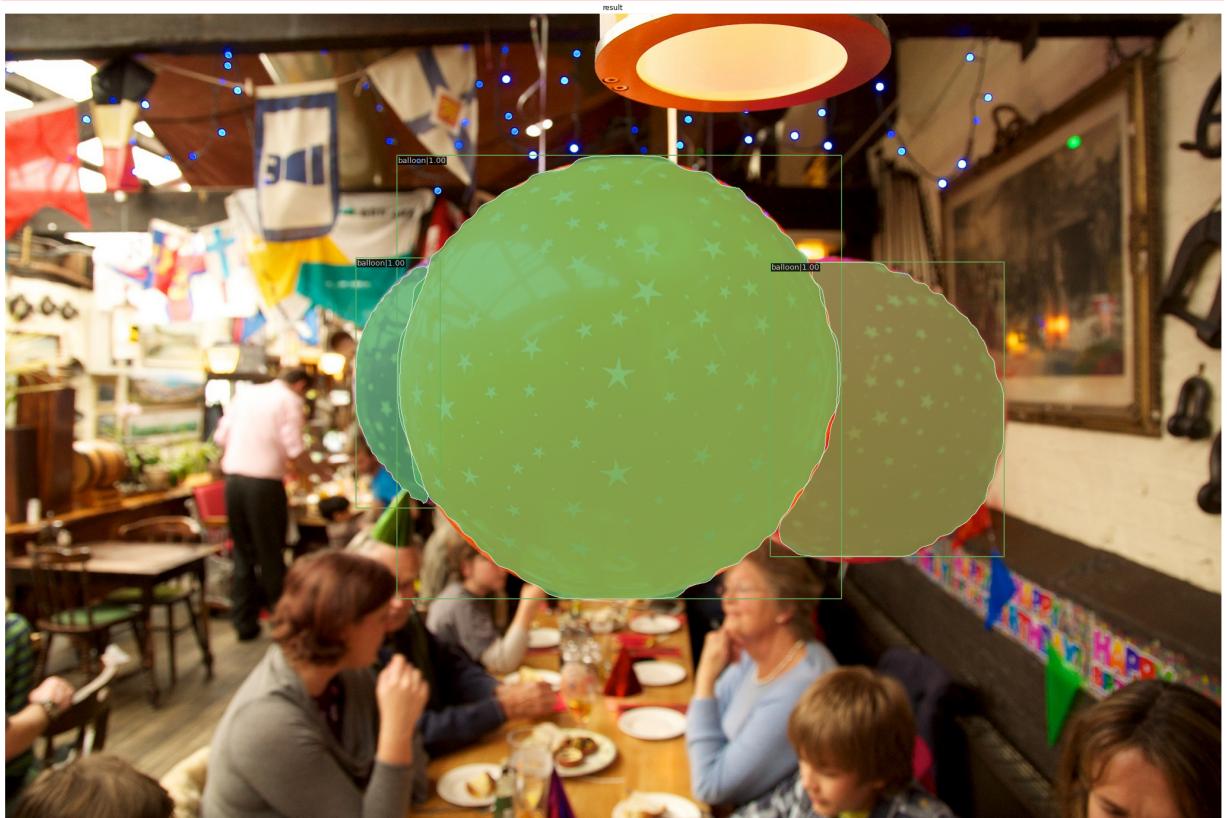
```

```

/content/mmdetection/mmdet/datasets/utils.py:70: UserWarning: "ImageToTensor" pipeline
is replaced by "DefaultFormatBundle" for batch inference. It is recommended to manuall

```

y replace it in the test data pipeline in your config file.
' data pipeline in your config file.', UserWarning)



バルーンに対してのみ、正しく認識し、マスク処理が行われていることが確認できる。