【全人類がわかるE資格コース PyTorch版】プロダクト開発演習

6. Labolo Tomato事前学習モデルの検証

公開されているLaboro Tomato Datasetにより学習済みのMask R-CNNモデルが実際に動作することを確認する。 google Colab環境でGPUを使用する。 モデル環境の構築の後、test data、アノテーションデータを使って公開されている性能がでるのかを評価し、その正当性を確認（validation）する。
更に、今回新たに準備したトマト画像（静止画、動画）データによって、正しく判別が行えるかを検証（verification）する。

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

```
Mounted at /content/drive
```

6.1 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
# MMDetction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

# 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
!pip install mmcv-full==1.3.17 -f https://download.openmmlab.com/mmcv/dist/cu111/torc
# 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
!git clone https://github.com/open-mmlab/mmdetection.git -b v2.23.0 --depth 1    ### t
%cd mmdetection

!pip install -e .
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pu
blic/simple/
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.9.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torch-1.9.0%2Bcu111-cp37-cp37m-li
nux_x86_64.whl (2041.3 MB)
     |████████████                    | 834.1 MB 1.3 MB/s eta 0:15:51tcmalloc: large a
lloc 1147494400 bytes == 0x3a586000 @  0x7f5b24134615 0x592b76 0x4df71e 0x59afff 0x515
655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576 0x593fce 0x548ae9 0x5127f1 0x598e3b 0
x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68 0x4bc98a 0x532e76 0x594b72 0x515600 0x5495
76 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x5118f8 0x593dd7
     |████████████████                | 1055.7 MB 1.2 MB/s eta 0:13:18tcmalloc: large
alloc 1434370048 bytes == 0x7ebdc000 @  0x7f5b24134615 0x592b76 0x4df71e 0x59afff 0x51
5655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576 0x593fce 0x548ae9 0x5127f1 0x598e3b
0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68 0x4bc98a 0x532e76 0x594b72 0x515600 0x549
576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x5118f8 0x593dd7
     |████████████████████            | 1336.2 MB 1.2 MB/s eta 0:09:51tcmalloc: large
alloc 1792966656 bytes == 0x3a0e000 @  0x7f5b24134615 0x592b76 0x4df71e 0x59afff 0x515
655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576 0x593fce 0x548ae9 0x5127f1 0x598e3b 0
x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68 0x4bc98a 0x532e76 0x594b72 0x515600 0x5495
76 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x5118f8 0x593dd7
     |████████████████████████        | 1691.1 MB 1.2 MB/s eta 0:04:54tcmalloc: large
alloc 2241208320 bytes == 0x6e7f6000 @  0x7f5b24134615 0x592b76 0x4df71e 0x59afff 0x51
5655 0x549576 0x593fce 0x548ae9 0x51566f 0x549576 0x593fce 0x548ae9 0x5127f1 0x598e3b
0x511f68 0x598e3b 0x511f68 0x598e3b 0x511f68 0x4bc98a 0x532e76 0x594b72 0x515600 0x549
576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x5118f8 0x593dd7
     |████████████████████████████████| 2041.3 MB 1.2 MB/s eta 0:00:01tcmalloc: large
alloc 2041348096 bytes == 0xf4158000 @  0x7f5b241331e7 0x4a3940 0x4a39cc 0x592b76 0x4d
f71e 0x59afff 0x515655 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576
0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x593dd7 0x511
e2c 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x548ae9
tcmalloc: large alloc 2551685120 bytes == 0x16dc20000 @  0x7f5b24134615 0x592b76 0x4df
71e 0x59afff 0x515655 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576 0
x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x549576 0x593fce 0x511e2c 0x593dd7 0x511e
2c 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576
     |████████████████████████████████| 2041.3 MB 7.2 kB/s
Collecting torchvision==0.10.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torchvision-0.10.0%2Bcu111-cp37-c
p37m-linux_x86_64.whl (23.2 MB)
     |████████████████████████████████| 23.2 MB 1.3 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pack
ages (from torch==1.9.0+cu111) (4.1.1)
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python3.7/dist-packages
(from torchvision==0.10.0+cu111) (7.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from t
orchvision==0.10.0+cu111) (1.21.6)
Installing collected packages: torch, torchvision
  Attempting uninstall: torch
    Found existing installation: torch 1.11.0+cu113
    Uninstalling torch-1.11.0+cu113:
      Successfully uninstalled torch-1.11.0+cu113
  Attempting uninstall: torchvision
    Found existing installation: torchvision 0.12.0+cu113
    Uninstalling torchvision-0.12.0+cu113:
      Successfully uninstalled torchvision-0.12.0+cu113
ERROR: pip's dependency resolver does not currently take into account all the packages
```

```
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 incom
patible.
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/pu
blic/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 9.3 MB/s
Requirement already satisfied: opencv-python>=3 in /usr/local/lib/python3.7/dist-packa
ges (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)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (fr
om mmcv-full==1.3.17) (21.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from m
mcv-full==1.3.17) (1.21.6)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from
 mmcv-full==1.3.17) (3.13)
Collecting yapf
  Downloading yapf-0.32.0-py2.py3-none-any.whl (190 kB)
     |████████████████████████████████| 190 kB 5.1 MB/s
Collecting addict
  Downloading addict-2.4.0-py3-none-any.whl (3.8 kB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/di
st-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 | 30.57 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/pu
blic/simple/
Obtaining file:///content/mmdetection
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (f
rom mmdet==2.23.0) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from m
mdet==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 mmd
et==2.23.0) (1.15.0)
Collecting terminaltables
```

```
  Downloading terminaltables-3.1.10-py2.py3-none-any.whl (15 kB)
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: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pack
ages (from matplotlib->mmdet==2.23.0) (1.4.3)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-p
ackages (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: typing-extensions in /usr/local/lib/python3.7/dist-pack
ages (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_KINETO -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=O
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
```

6.2 LABORO TOMATOデータ、学習済みモデルの移植、動作確認

Mask R-CNN r50 FPN 1x8 COCOモデル

Laboro Tomato dataset（ローカルコピーしたもの）を、google DriveからMMdetectionの実行環境にコピー（シンボリックリンク）
MMDetectionフレームワークで利用できるようモデルの環境設定(config）を、LaboroTomatoの公開情報に基づき変更する。

- Datasetパス情報の変更
- 識別クラス数の変更（80 => 6）
- 学習済みcheckpointファイル（パラメタデータファイル）のロード

である。
https://github.com/open-mmlab/mmdetection

In [6]:
```
!cd '/content/mmdetection'

# laboro_tomato datasetを./data/にシンボリックリンク
!mkdir -p ./data
!ln -s '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato' './data/
```

In [7]:
```
# datasetクラスの登録
!ln -s '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/labor
    './mmdet/datasets/laboro_tomato.py'
```

In [8]:
```
# configのセット
from mmcv import Config
cfg = Config.fromfile('./configs/mask_rcnn/mask_rcnn_r50_fpn_1x_coco.py')

# 学習済みcheckpoint を./checkponts/にシンボリックリンク
!mkdir -p ./checkpoints
!ln -s '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/labor
    './checkpoints/laboro_tomato_48ep.pth'
```

In [9]:
```
# Add dataset names to mmdet/datasets/__init__.py
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/__init__
    './mmdet/datasets/__init__.py'

# Cnfiguration files setup
# step1: add laboro_tomato_base.py to configs/_base_/datasets/
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/laboro_t
    './configs/_base_/datasets/laboro_tomato_base.py'
```

```
# step2: add laboro_tomato_instance.py to configs/_base_/datasets/
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/laboro_t
    './configs/_base_/datasets/laboro_tomato_instance.py'

# step3: overwrite class numbers at model configuration file configs/_base_/models/mas
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/mask_rcn
    './configs/_base_/models/mask_rcnn_r50_fpn.py'

# step4: overwrite base instance configuration file name at configs/mask_rcnn/mask_rcn
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato_etc/mask_rcn
    './configs/mask_rcnn/mask_rcnn_r50_fpn_1x_coco.py'
```

In [10]:

```
# 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_fpn_1x_coco.py'
# Setup a checkpoint file to load
checkpoint = './checkpoints/laboro_tomato_48ep.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

# 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()
```

Out[10]:

```
load checkpoint from local path: ./checkpoints/laboro_tomato_48ep.pth
MaskRCNN(
  (backbone): ResNet(
    (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=Fal
se)
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=T
rue)
    (relu): ReLU(inplace=True)
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=Fal
se)
    (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_sta
ts=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bia
s=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=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)
        (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_st
ats=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_sta
ts=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bia
s=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=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)
      )
      (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_sta
ts=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bia
s=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_sta
ts=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=(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=(2, 2), 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)
        (relu): ReLU(inplace=True)
      )
    )
    (layer3): ResLayer(
      (0): Bottleneck(
        (conv1): Conv2d(512, 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=(2, 2), 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=(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=(2, 2), 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)
        (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': 'torchvision://resnet50'}
  (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=7, bias=True)
      (fc_reg): Linear(in_features=1024, out_features=24, 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'}}, {'typ
e': 'Normal', 'std': 0.001, 'override': {'name': 'fc_reg'}}, {'type': 'Xavier', 'distr
ibution': 'uniform', 'override': [{'name': 'shared_fcs'}, {'name': 'cls_fcs'}, {'nam
e': '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, poo
l_mode=avg, aligned=True, use_torchvision=False)
        (2): RoIAlign(output_size=(14, 14), spatial_scale=0.0625, sampling_ratio=0, po
ol_mode=avg, aligned=True, use_torchvision=False)
        (3): RoIAlign(output_size=(14, 14), spatial_scale=0.03125, sampling_ratio=0, p
ool_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, 6, kernel_size=(1, 1), stride=(1, 1))
      (relu): ReLU(inplace=True)
    )
  )
)
```

6.3 test dataによるLaboro Tomato 事前学習modelの正当性確認

In [12]:

```
# single-gpu testing
!python tools/test.py configs/mask_rcnn/mask_rcnn_r50_fpn_1x_coco.py ¥
                      checkpoints/laboro_tomato_48ep.pth --show --eval "bbox" "segm"
```

```
/content/mmdetection/mmdet/utils/setup_env.py:33: UserWarning: Setting OMP_NUM_THREADS
environment variable for each process to be 1 in default, to avoid your system being o
verloaded, please further tune the variable for optimal performance in your applicatio
n as needed.
  f'Setting OMP_NUM_THREADS environment variable for each process '
/content/mmdetection/mmdet/utils/setup_env.py:43: UserWarning: Setting MKL_NUM_THREADS
environment variable for each process to be 1 in default, to avoid your system being o
verloaded, please further tune the variable for optimal performance in your applicatio
n as needed.
  f'Setting MKL_NUM_THREADS environment variable for each process '
loading annotations into memory...
Done (t=0.04s)
creating index...
index created!
load checkpoint from local path: checkpoints/laboro_tomato_48ep.pth
[                                                  ] 0/161, elapsed: 0s, ETA:/usr/loca
l/lib/python3.7/dist-packages/torch/nn/functional.py:718: UserWarning: Named tensors a
nd 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 i
nternally at  /pytorch/c10/core/TensorImpl.h:1156.)
  return torch.max_pool2d(input, kernel_size, stride, padding, dilation, ceil_mode)
<Figure size 3024.01x4032.01 with 1 Axes>
[  ] 1/161, 0.1 task/s, elapsed: 9s, ETA:  1431s<Figure size 3120.01x4160.01 with 1 Ax
es>
[  ] 2/161, 0.2 task/s, elapsed: 12s, ETA:   981s<Figure size 3024.01x4032.01 with 1 A
xes>
[  ] 3/161, 0.2 task/s, elapsed: 18s, ETA:   956s<Figure size 3120.01x4160.01 with 1 A
```

```
xes>
[  ] 4/161, 0.2 task/s, elapsed: 23s, ETA:   914s<Figure size 3120.01x4160.01 with 1 A
xes>
[  ] 5/161, 0.2 task/s, elapsed: 26s, ETA:   819s<Figure size 3120.01x4160.01 with 1 A
xes>
[  ] 6/161, 0.2 task/s, elapsed: 28s, ETA:   735s<Figure size 3120.01x4160.01 with 1 A
xes>
[  ] 7/161, 0.2 task/s, elapsed: 33s, ETA:   726s<Figure size 3024.01x4032.01 with 1 A
xes>
[  ] 8/161, 0.2 task/s, elapsed: 35s, ETA:   676s<Figure size 3024.01x4032.01 with 1 A
xes>
[  ] 9/161, 0.2 task/s, elapsed: 38s, ETA:   636s<Figure size 3120.01x4160.01 with 1 A
xes>
[  ] 10/161, 0.2 task/s, elapsed: 40s, ETA:   607s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 11/161, 0.2 task/s, elapsed: 45s, ETA:   613s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 12/161, 0.3 task/s, elapsed: 47s, ETA:   587s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 13/161, 0.3 task/s, elapsed: 50s, ETA:   565s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 14/161, 0.3 task/s, elapsed: 53s, ETA:   553s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 15/161, 0.3 task/s, elapsed: 55s, ETA:   535s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 16/161, 0.3 task/s, elapsed: 57s, ETA:   518s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 17/161, 0.3 task/s, elapsed: 61s, ETA:   513s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 18/161, 0.3 task/s, elapsed: 64s, ETA:   506s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 19/161, 0.3 task/s, elapsed: 66s, ETA:   494s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 20/161, 0.3 task/s, elapsed: 69s, ETA:   487s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 21/161, 0.3 task/s, elapsed: 74s, ETA:   495s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 22/161, 0.3 task/s, elapsed: 78s, ETA:   490s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 23/161, 0.3 task/s, elapsed: 80s, ETA:   479s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 24/161, 0.3 task/s, elapsed: 82s, ETA:   468s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 25/161, 0.3 task/s, elapsed: 84s, ETA:   458s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 26/161, 0.3 task/s, elapsed: 88s, ETA:   455s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 27/161, 0.3 task/s, elapsed: 90s, ETA:   449s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 28/161, 0.3 task/s, elapsed: 94s, ETA:   445s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 29/161, 0.3 task/s, elapsed: 96s, ETA:   436s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 30/161, 0.3 task/s, elapsed: 99s, ETA:   431s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 31/161, 0.3 task/s, elapsed: 102s, ETA:   427s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 32/161, 0.3 task/s, elapsed: 104s, ETA:   420s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 33/161, 0.3 task/s, elapsed: 107s, ETA:   415s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 34/161, 0.3 task/s, elapsed: 109s, ETA:   408s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 35/161, 0.3 task/s, elapsed: 111s, ETA:   401s<Figure size 3120.01x4160.01 with 1
```

```
Axes>
[  ] 36/161, 0.3 task/s, elapsed: 113s, ETA:   394s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 37/161, 0.3 task/s, elapsed: 117s, ETA:   391s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 38/161, 0.3 task/s, elapsed: 120s, ETA:   388s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 39/161, 0.3 task/s, elapsed: 123s, ETA:   386s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 40/161, 0.3 task/s, elapsed: 127s, ETA:   385s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 41/161, 0.3 task/s, elapsed: 131s, ETA:   382s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 42/161, 0.3 task/s, elapsed: 133s, ETA:   378s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 43/161, 0.3 task/s, elapsed: 137s, ETA:   376s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 44/161, 0.3 task/s, elapsed: 140s, ETA:   372s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 45/161, 0.3 task/s, elapsed: 144s, ETA:   370s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 46/161, 0.3 task/s, elapsed: 150s, ETA:   376s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 47/161, 0.3 task/s, elapsed: 153s, ETA:   371s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 48/161, 0.3 task/s, elapsed: 156s, ETA:   366s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 49/161, 0.3 task/s, elapsed: 159s, ETA:   363s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 50/161, 0.3 task/s, elapsed: 163s, ETA:   362s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 51/161, 0.3 task/s, elapsed: 167s, ETA:   361s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 52/161, 0.3 task/s, elapsed: 170s, ETA:   357s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 53/161, 0.3 task/s, elapsed: 176s, ETA:   358s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 54/161, 0.3 task/s, elapsed: 179s, ETA:   355s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 55/161, 0.3 task/s, elapsed: 189s, ETA:   364s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 56/161, 0.3 task/s, elapsed: 192s, ETA:   359s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 57/161, 0.3 task/s, elapsed: 196s, ETA:   358s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 58/161, 0.3 task/s, elapsed: 200s, ETA:   356s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 59/161, 0.3 task/s, elapsed: 210s, ETA:   363s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 60/161, 0.3 task/s, elapsed: 218s, ETA:   366s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 61/161, 0.3 task/s, elapsed: 220s, ETA:   361s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 62/161, 0.3 task/s, elapsed: 223s, ETA:   356s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 63/161, 0.3 task/s, elapsed: 227s, ETA:   353s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 64/161, 0.3 task/s, elapsed: 230s, ETA:   348s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 65/161, 0.3 task/s, elapsed: 237s, ETA:   350s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 66/161, 0.3 task/s, elapsed: 241s, ETA:   346s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 67/161, 0.3 task/s, elapsed: 243s, ETA:   341s<Figure size 3120.01x4160.01 with 1
```

```
Axes>
[  ] 68/161, 0.3 task/s, elapsed: 247s, ETA:   338s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 69/161, 0.3 task/s, elapsed: 250s, ETA:   334s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 70/161, 0.3 task/s, elapsed: 253s, ETA:   330s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 71/161, 0.3 task/s, elapsed: 261s, ETA:   330s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 72/161, 0.3 task/s, elapsed: 268s, ETA:   331s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 73/161, 0.3 task/s, elapsed: 273s, ETA:   329s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 74/161, 0.3 task/s, elapsed: 278s, ETA:   326s<Figure size 4160.01x3120.01 with 1
Axes>
[  ] 75/161, 0.3 task/s, elapsed: 284s, ETA:   326s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 76/161, 0.3 task/s, elapsed: 292s, ETA:   326s<Figure size 4160.01x3120.01 with 1
Axes>
[  ] 77/161, 0.3 task/s, elapsed: 296s, ETA:   323s<Figure size 3120.01x4160.01 with 1
Axes>
[  ] 78/161, 0.3 task/s, elapsed: 302s, ETA:   321s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 79/161, 0.3 task/s, elapsed: 309s, ETA:   321s<Figure size 3024.01x4032.01 with 1
Axes>
[  ] 80/161, 0.3 task/s, elapsed: 311s, ETA:   315s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 81/161, 0.3 task/s, elapsed: 314s, ETA:   310s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 82/161, 0.3 task/s, elapsed: 317s, ETA:   305s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 83/161, 0.3 task/s, elapsed: 322s, ETA:   302s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 84/161, 0.3 task/s, elapsed: 329s, ETA:   302s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 85/161, 0.3 task/s, elapsed: 334s, ETA:   299s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 86/161, 0.3 task/s, elapsed: 339s, ETA:   296s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 87/161, 0.3 task/s, elapsed: 343s, ETA:   291s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 88/161, 0.3 task/s, elapsed: 349s, ETA:   289s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 89/161, 0.3 task/s, elapsed: 352s, ETA:   285s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 90/161, 0.3 task/s, elapsed: 359s, ETA:   283s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 91/161, 0.2 task/s, elapsed: 365s, ETA:   281s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 92/161, 0.2 task/s, elapsed: 369s, ETA:   277s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 93/161, 0.2 task/s, elapsed: 372s, ETA:   272s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 94/161, 0.2 task/s, elapsed: 377s, ETA:   268s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 95/161, 0.2 task/s, elapsed: 383s, ETA:   266s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 96/161, 0.2 task/s, elapsed: 387s, ETA:   262s<Figure size 3120.01x4160.01 with 1
Axes>
[> ] 97/161, 0.2 task/s, elapsed: 391s, ETA:   258s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 98/161, 0.2 task/s, elapsed: 394s, ETA:   254s<Figure size 3024.01x4032.01 with 1
Axes>
[> ] 99/161, 0.2 task/s, elapsed: 397s, ETA:   249s<Figure size 3120.01x4160.01 with 1
```

```
Axes>
[> ] 100/161, 0.2 task/s, elapsed: 401s, ETA:   244s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 101/161, 0.3 task/s, elapsed: 403s, ETA:   239s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 102/161, 0.2 task/s, elapsed: 409s, ETA:   237s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 103/161, 0.3 task/s, elapsed: 412s, ETA:   232s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 104/161, 0.2 task/s, elapsed: 418s, ETA:   229s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 105/161, 0.2 task/s, elapsed: 421s, ETA:   225s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 106/161, 0.3 task/s, elapsed: 424s, ETA:   220s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 107/161, 0.3 task/s, elapsed: 426s, ETA:   215s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 108/161, 0.3 task/s, elapsed: 430s, ETA:   211s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 109/161, 0.3 task/s, elapsed: 432s, ETA:   206s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 110/161, 0.3 task/s, elapsed: 437s, ETA:   202s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 111/161, 0.3 task/s, elapsed: 442s, ETA:   199s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 112/161, 0.3 task/s, elapsed: 445s, ETA:   195s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 113/161, 0.3 task/s, elapsed: 450s, ETA:   191stcmalloc: large alloc 1219280896 b
ytes == 0x14331a000 @  0x7fed4f8a21e7 0x7fed4d1b30ce 0x7fed4d209cf5 0x7fed4d2b286d 0x7
fed4d2b317f 0x7fed4d2b32d0 0x4bc4ab 0x7fed4d1f4944 0x59371f 0x515244 0x549576 0x593fce
0x548ae9 0x5127f1 0x549e0e 0x4bca8a 0x7fed4d1f4944 0x59371f 0x515244 0x549576 0x593fce
0x548ae9 0x5127f1 0x549576 0x593fce 0x548ae9 0x5127f1 0x549576 0x593fce 0x5118f8 0x593
dd7
<Figure size 3024.01x4032.01 with 1 Axes>
[> ] 114/161, 0.2 task/s, elapsed: 470s, ETA:   194s<Figure size 4160.01x3120.01 with
1 Axes>
[> ] 115/161, 0.2 task/s, elapsed: 476s, ETA:   191s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 116/161, 0.2 task/s, elapsed: 481s, ETA:   187s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 117/161, 0.2 task/s, elapsed: 486s, ETA:   183s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 118/161, 0.2 task/s, elapsed: 490s, ETA:   178s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 119/161, 0.2 task/s, elapsed: 492s, ETA:   174s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 120/161, 0.2 task/s, elapsed: 494s, ETA:   169s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 121/161, 0.2 task/s, elapsed: 497s, ETA:   164s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 122/161, 0.2 task/s, elapsed: 501s, ETA:   160s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 123/161, 0.2 task/s, elapsed: 508s, ETA:   157s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 124/161, 0.2 task/s, elapsed: 512s, ETA:   153s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 125/161, 0.2 task/s, elapsed: 517s, ETA:   149s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 126/161, 0.2 task/s, elapsed: 526s, ETA:   146s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 127/161, 0.2 task/s, elapsed: 531s, ETA:   142s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 128/161, 0.2 task/s, elapsed: 534s, ETA:   138s<Figure size 3024.01x4032.01 with
1 Axes>
```

```
[> ] 129/161, 0.2 task/s, elapsed: 537s, ETA:   133s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 130/161, 0.2 task/s, elapsed: 542s, ETA:   129s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 131/161, 0.2 task/s, elapsed: 545s, ETA:   125s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 132/161, 0.2 task/s, elapsed: 548s, ETA:   120s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 133/161, 0.2 task/s, elapsed: 551s, ETA:   116s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 134/161, 0.2 task/s, elapsed: 557s, ETA:   112s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 135/161, 0.2 task/s, elapsed: 560s, ETA:   108s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 136/161, 0.2 task/s, elapsed: 567s, ETA:   104s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 137/161, 0.2 task/s, elapsed: 576s, ETA:   101s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 138/161, 0.2 task/s, elapsed: 582s, ETA:    97s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 139/161, 0.2 task/s, elapsed: 586s, ETA:    93s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 140/161, 0.2 task/s, elapsed: 598s, ETA:    90s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 141/161, 0.2 task/s, elapsed: 600s, ETA:    85s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 142/161, 0.2 task/s, elapsed: 604s, ETA:    81s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 143/161, 0.2 task/s, elapsed: 610s, ETA:    77s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 144/161, 0.2 task/s, elapsed: 613s, ETA:    72s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 145/161, 0.2 task/s, elapsed: 617s, ETA:    68s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 146/161, 0.2 task/s, elapsed: 621s, ETA:    64s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 147/161, 0.2 task/s, elapsed: 625s, ETA:    60s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 148/161, 0.2 task/s, elapsed: 630s, ETA:    55s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 149/161, 0.2 task/s, elapsed: 636s, ETA:    51s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 150/161, 0.2 task/s, elapsed: 640s, ETA:    47s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 151/161, 0.2 task/s, elapsed: 644s, ETA:    43s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 152/161, 0.2 task/s, elapsed: 652s, ETA:    39s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 153/161, 0.2 task/s, elapsed: 657s, ETA:    34s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 154/161, 0.2 task/s, elapsed: 662s, ETA:    30s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 155/161, 0.2 task/s, elapsed: 665s, ETA:    26s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 156/161, 0.2 task/s, elapsed: 670s, ETA:    21s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 157/161, 0.2 task/s, elapsed: 675s, ETA:    17s<Figure size 3024.01x4032.01 with
1 Axes>
[> ] 158/161, 0.2 task/s, elapsed: 678s, ETA:    13s<Figure size 3120.01x4160.01 with
1 Axes>
[> ] 159/161, 0.2 task/s, elapsed: 685s, ETA:     9s<Figure size 4160.01x3120.01 with
1 Axes>
[> ] 160/161, 0.2 task/s, elapsed: 691s, ETA:     4s<Figure size 3120.01x4160.01 with
1 Axes>
```

```
[>>] 161/161, 0.2 task/s, elapsed: 695s, ETA:     0s
Evaluating bbox...
Loading and preparing results...
DONE (t=0.00s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.82s).
Accumulating evaluation results...
DONE (t=0.09s).

 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.647
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.822
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.735
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.146
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.681
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.742
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.742
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.742
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.186
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.780


Evaluating segm...
/content/mmdetection/mmdet/datasets/coco.py:474: UserWarning: The key "bbox" is delete
d for more accurate mask AP of small/medium/large instances since v2.12.0. This does n
ot change the overall mAP calculation.
  UserWarning)
Loading and preparing results...
DONE (t=0.05s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *segm*
DONE (t=1.05s).
Accumulating evaluation results...
/usr/local/lib/python3.7/dist-packages/pycocotools/cocoeval.py:378: DeprecationWarnin
g: `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 spe
cifically wanted the numpy scalar type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/rel
ease/1.20.0-notes.html#deprecations
  tp_sum = np.cumsum(tps, axis=1).astype(dtype=np.float)
DONE (t=0.10s).

 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.660
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.818
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.736
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.131
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.697
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.751
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.751
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.751
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.187
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.791

OrderedDict([('bbox_mAP', 0.647), ('bbox_mAP_50', 0.822), ('bbox_mAP_75', 0.735), ('bb
ox_mAP_s', 0.0), ('bbox_mAP_m', 0.146), ('bbox_mAP_l', 0.681), ('bbox_mAP_copypaste',
'0.647 0.822 0.735 0.000 0.146 0.681'), ('segm_mAP', 0.66), ('segm_mAP_50', 0.818),
```

```
('segm_mAP_75', 0.736), ('segm_mAP_s', 0.0), ('segm_mAP_m', 0.131), ('segm_mAP_l', 0.6
97), ('segm_mAP_copypaste', '0.660 0.818 0.736 0.000 0.131 0.697')])
```

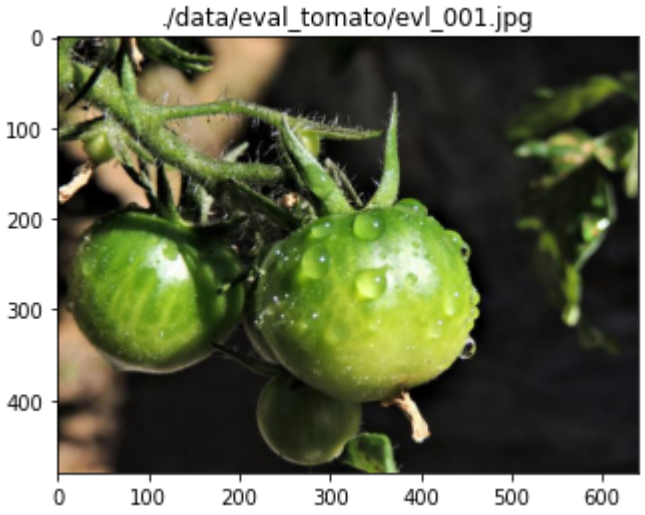
6.4 新たに準備したデータによる検証（verification）

In [13]:

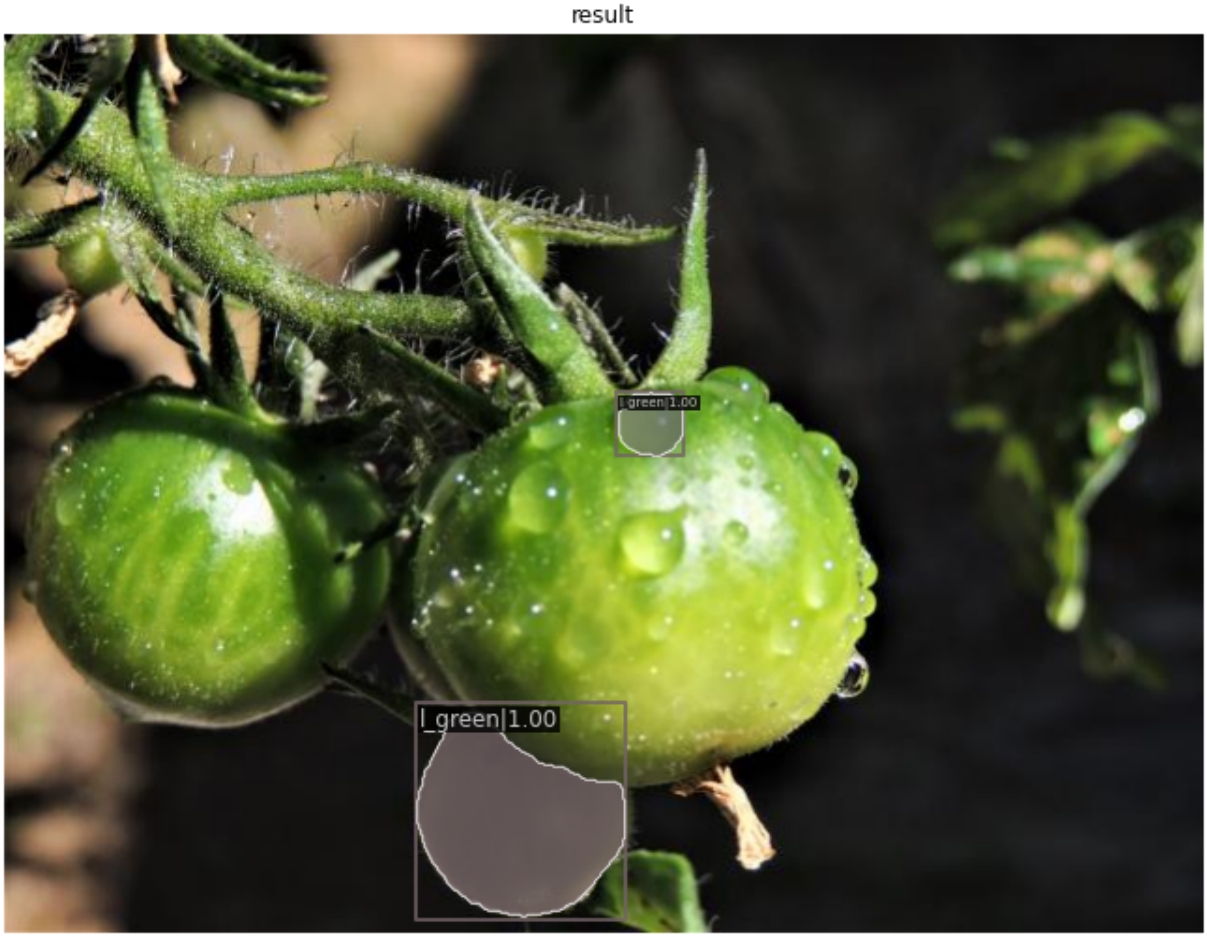
```
# フリー素材写真での評価（トマト、ミニトマト8枚、リンゴ：2枚）
# 前処理：写真の幅を360ピクセルに揃え、./data/eval_tomatoにロード
from sys import argv
import os
import glob
from PIL import Image

width = 640

src = glob.glob('/content/drive/MyDrive/Colab Notebooks/product_develop/eval_tomato/*.

!mkdir -p './data/eval_tomato/'
dst = './data/eval_tomato/' # リサイズ画像の保存フォルダ

for f in src:
    img = Image.open(f)
    original_width, original_height = img.size
    scale = width / original_width
    height = int(original_height * scale)
    img = img.resize((width,height))
    img.save(dst + os.path.basename(f))
```

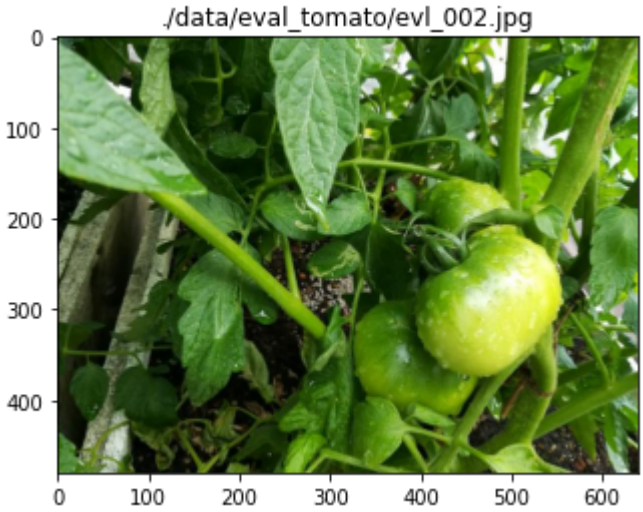
In [14]:

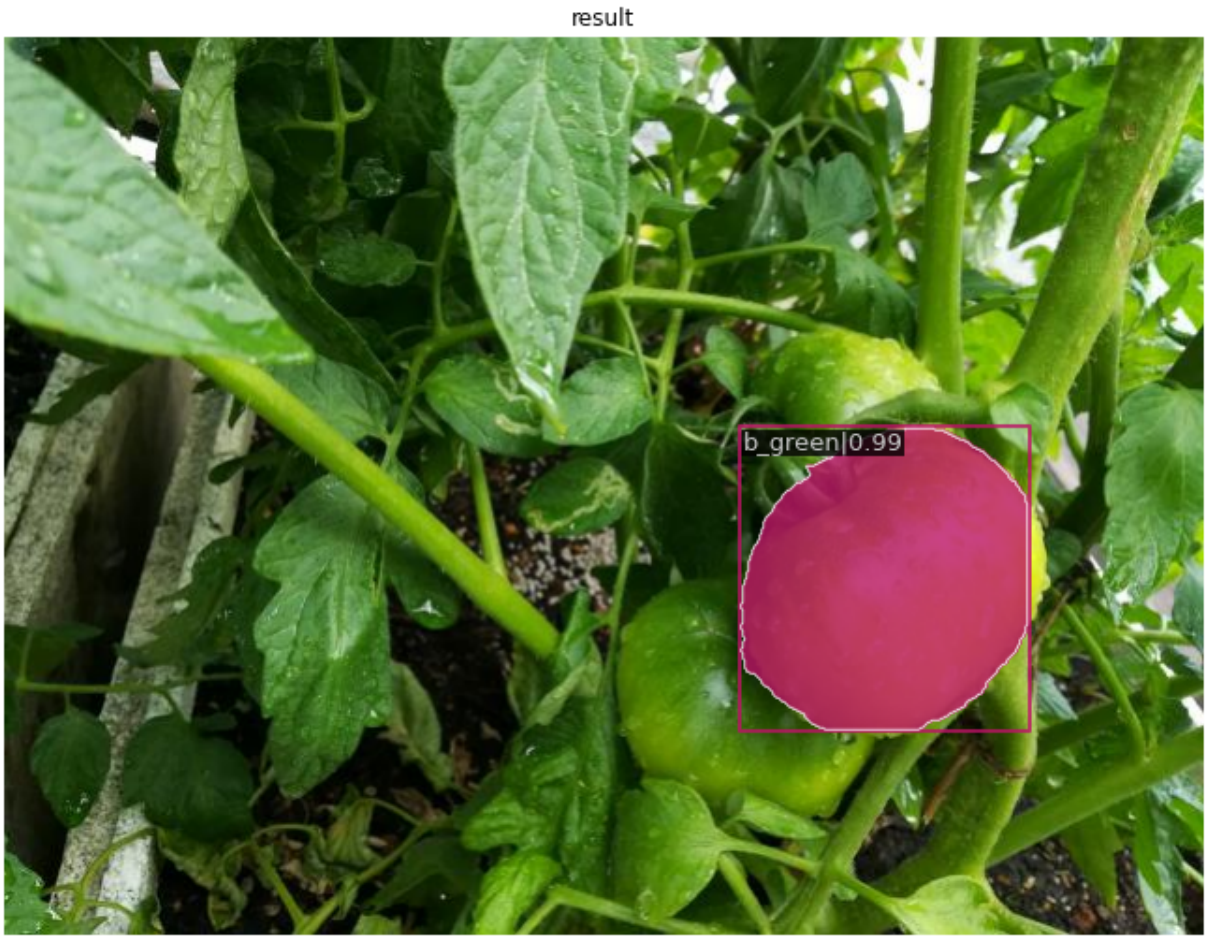
```
# 事前学習済みDetectorによる推定結果の評価
import mmcv
import matplotlib.pyplot as plt
import os
import glob
import pickle

!mkdir -p './data/eval_result/'
dst = './data/eval_result/'
src_imgs = sorted(glob.glob('./data/eval_tomato/*.jpg')) # オリジナル画像のパスと拡張

results = []

for i, f in enumerate(src_imgs):
    img = mmcv.imread(f)
    # plt.figure(figsize=(10, 10))
    plt.title(str(f))
    plt.imshow(mmcv.bgr2rgb(img))
    plt.show()

    # 検出器での物体検出推定demo実行
    # # Use the detector to do inference
    result = inference_detector(model, img)
    # Let's plot the result
    show_result_pyplot(model, img, result, score_thr=0.3)
    # save result data
    results.append(result)

# 推定・mask結果ファイルをpickle形式で保存
with open(dst+'results.bin', 'wb') as p:
    pickle.dump(dst+'results.bin', p)
```

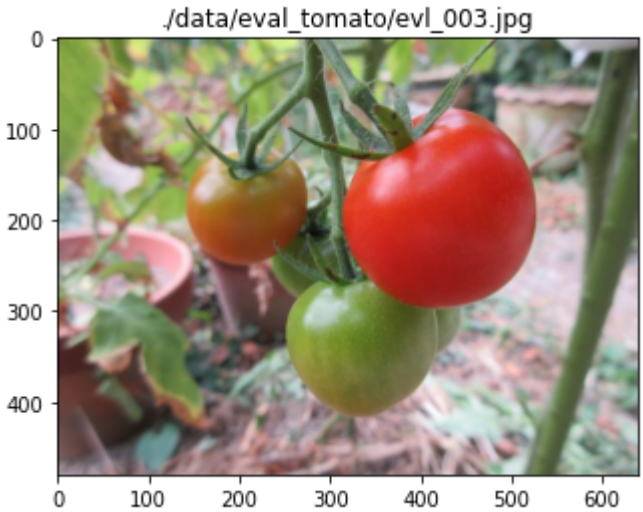



```
/content/mmdetection/mmdet/datasets/utils.py:70: UserWarning: "ImageToTensor" pipeline is replaced by "DefaultFormatBundle" for batch inference. It is recommended to manually replace it in the test data pipeline in your config file.
  'data pipeline in your config file.', UserWarning)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:718: 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)
```

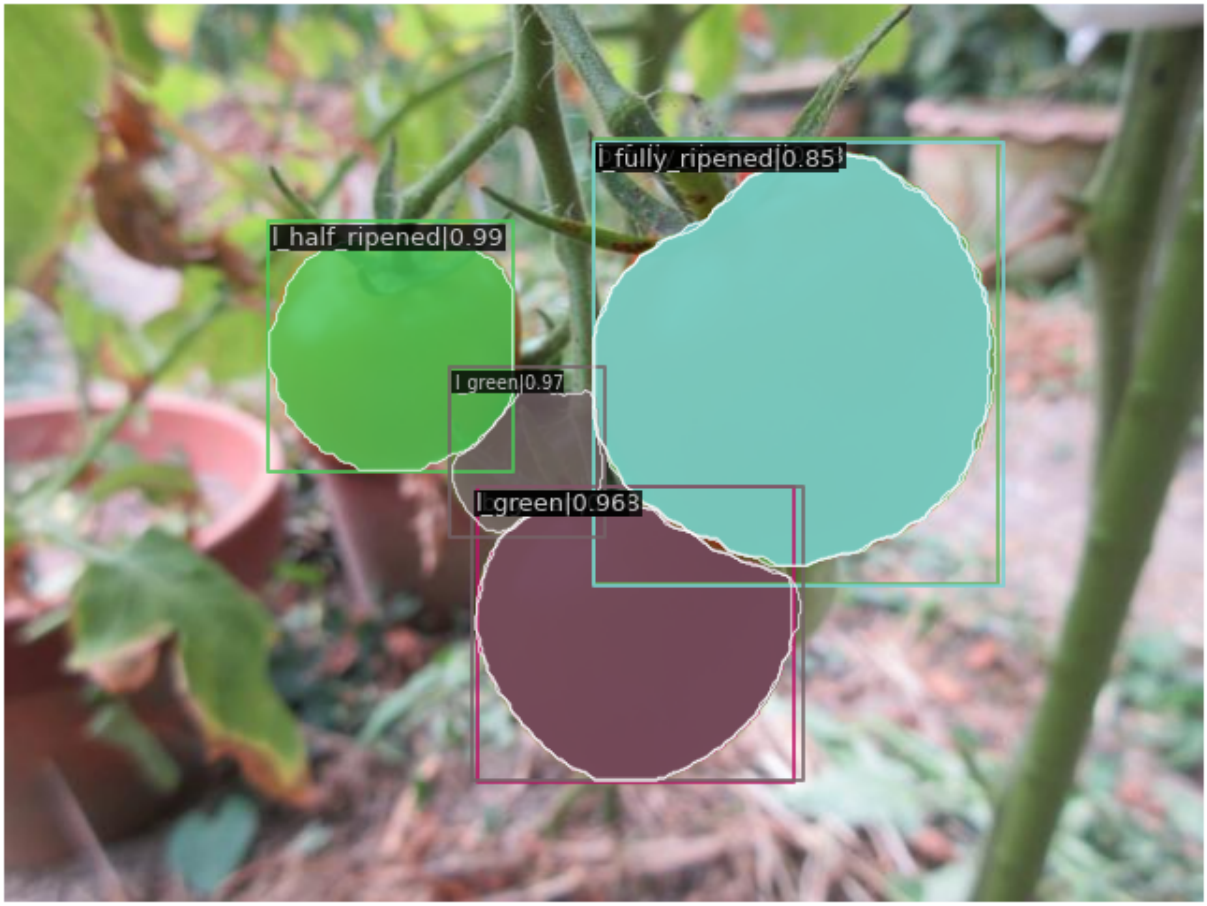


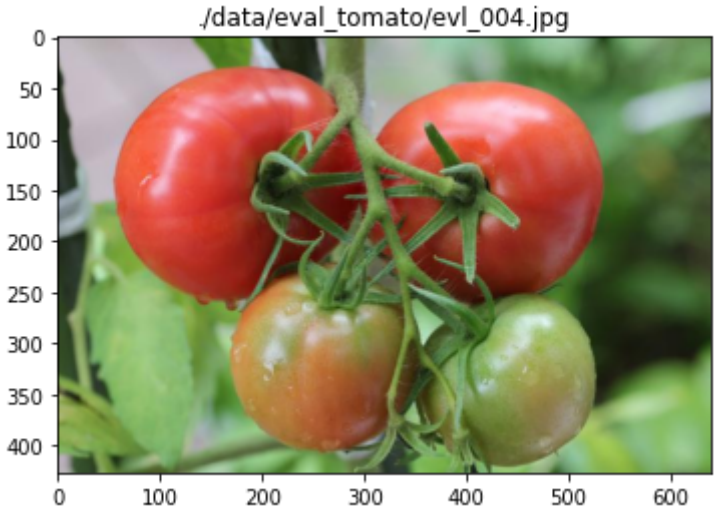

./data/eval_tomato/evl_002.jpg
0
100
200
300
400
0
100
200
300
400
500
600


result
b_green|0.99

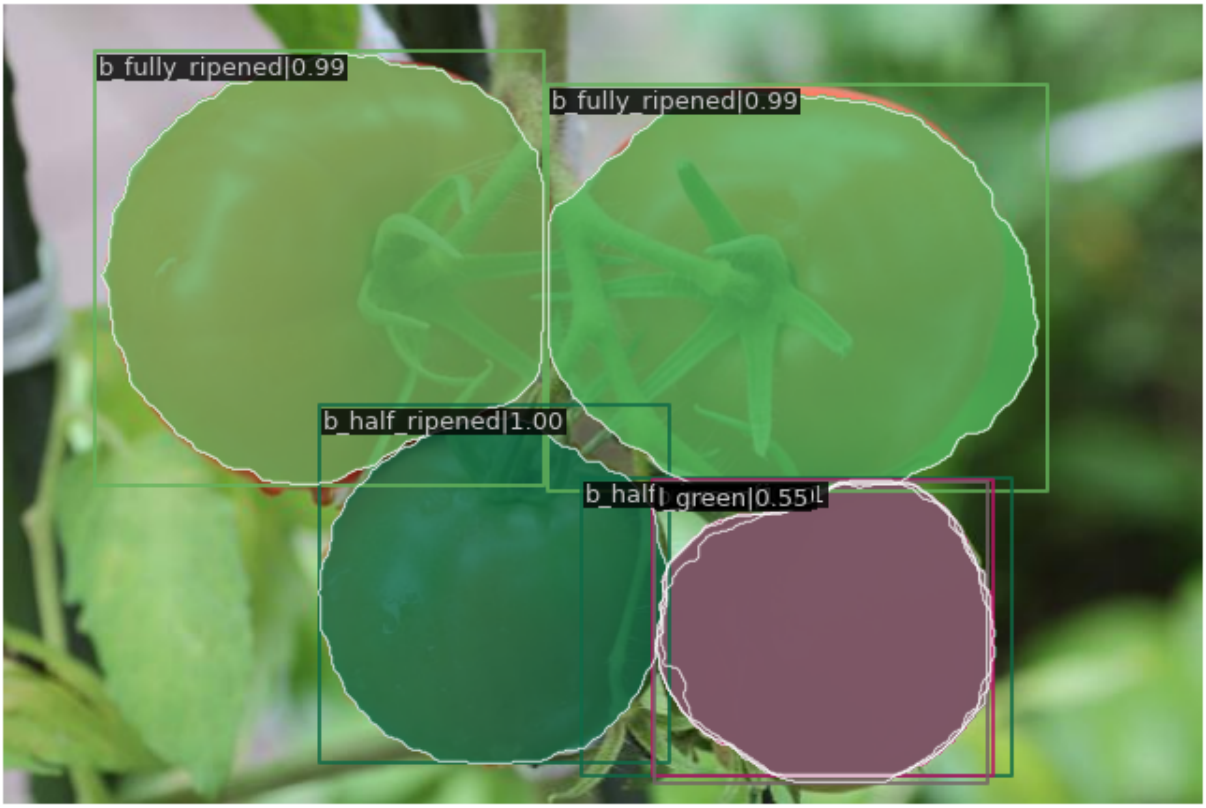

./data/eval_tomato/evl_003.jpg
0
100
200
300
400
0
100
200
300
400
500
600

result


l_fully_ripened|0.85
l_half_ripened|0.99
l_green|0.97
l_green|0.963


./data/eval_tomato/evl_004.jpg
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

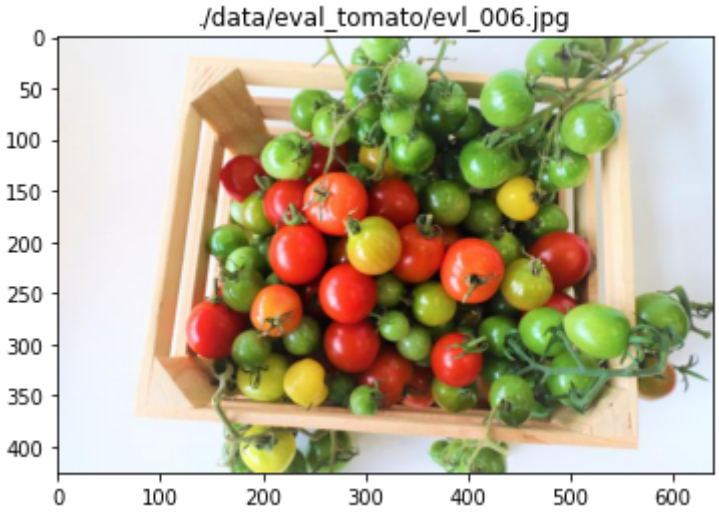
result



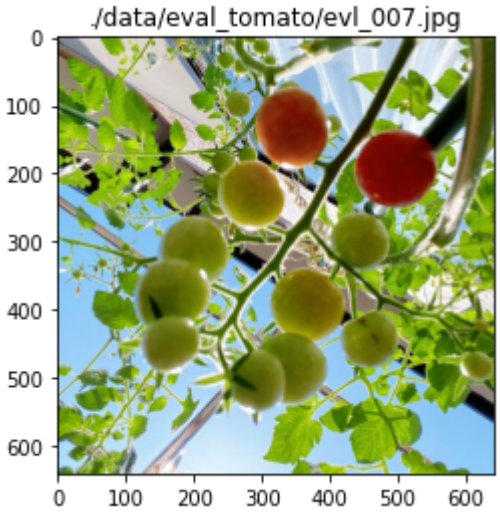
./data/eval_tomato/evl_005.jpg

result


l_fully_ripened|0.63
b_fully_ripened|0.99
l_half_ripened|0.857
b_fully_ripened|0.67
l_fully_ripened|0.87
b_half_ripened|0.44
b_green|1.00

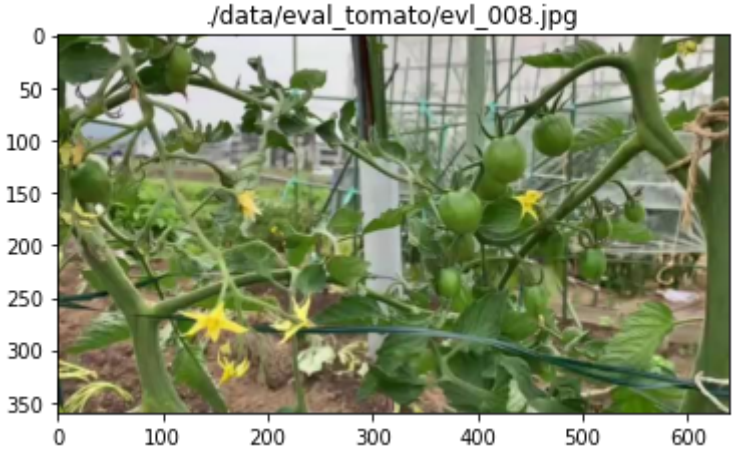

./data/eval_tomato/evl_006.jpg


result


./data/eval_tomato/evl_007.jpg

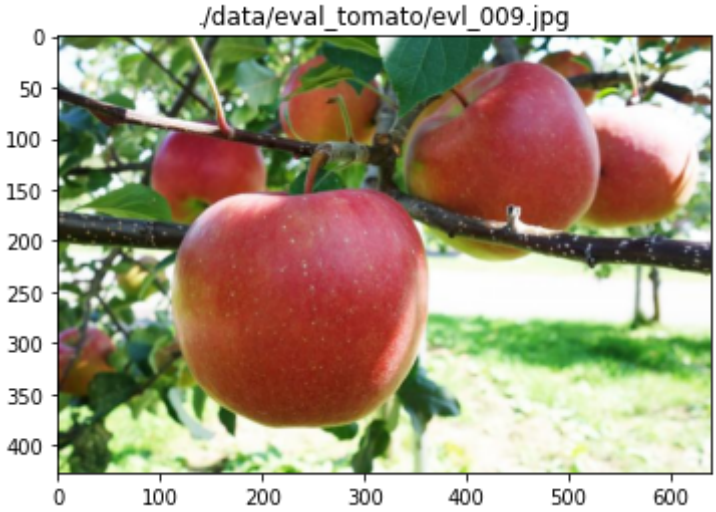
result


l_fully_ripened|1.00
l_fully_ripened|1.00
l_half_ripened|0.96
l_green|1.00
l_green|1.00
l_green|1.00
l_green|0.95
l_green|0.99
l_green|1.00
b_half_ripened|0.48
b_green|1.00

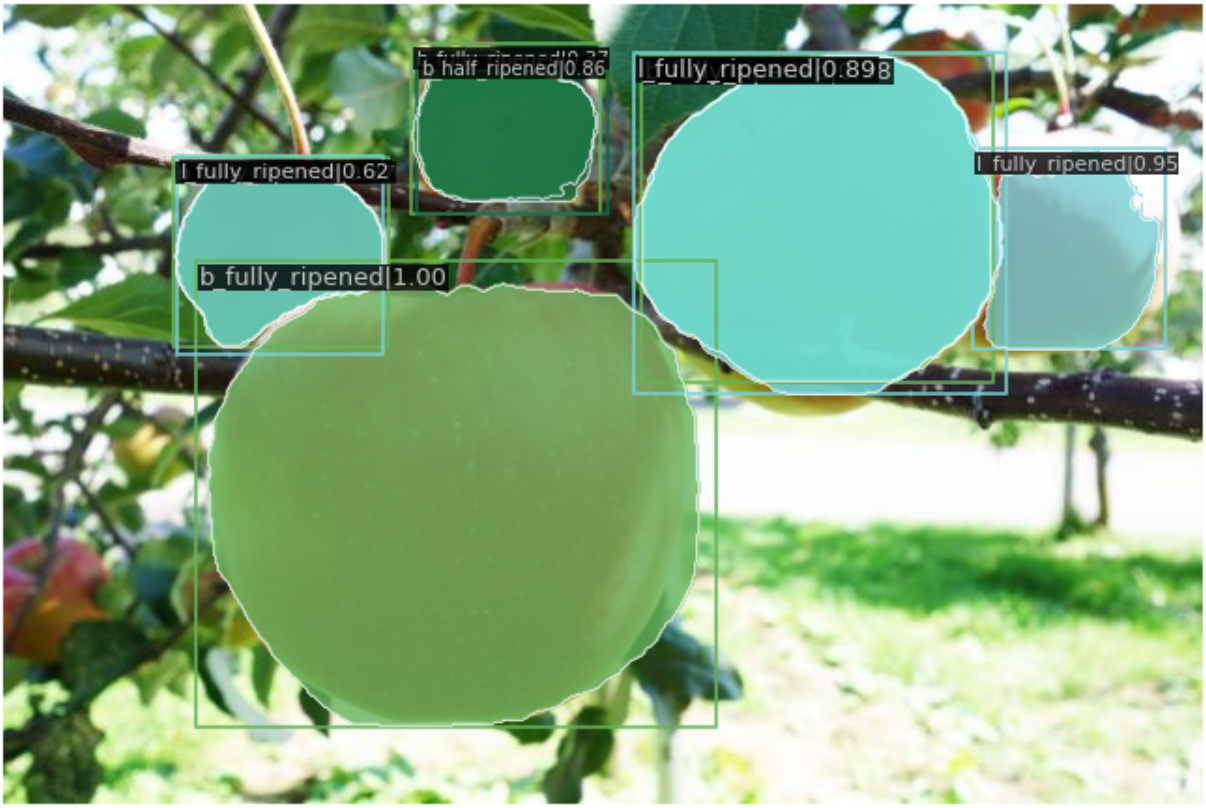

./data/eval_tomato/evl_008.jpg
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600

result

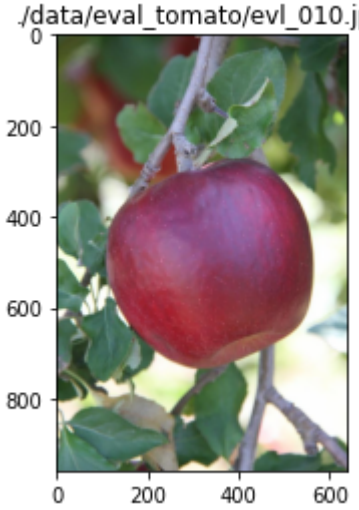

l green 0.98
l green 1.00
l green 1.00
b green 0.97


./data/eval_tomato/evl_009.jpg
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

result



./data/eval_tomato/evl_010.jpg



result

4.2 動画のセグメンテーション

In [ ]:

```
# 動画のセグメンテーション
# 動画データのロード
```

```
!cd '/content/mmdetection'

# laboro_tomato datasetを./data/にシンボリックリンク
!mkdir -p ./data
!cp -Ri '/content/drive/MyDrive/Colab Notebooks/product_develop/video_tomato' './data'
```

In [18]:

```
# 動画セグメンテーション実行
src_movs = sorted(glob.glob('./data/video_tomato/*.mp4')) # オリジナル画像のパスと拡張

device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')

#!mkdir -p './data/video_tomato/'
#dst = './data/video_tomato' # リサイズ画像の保存フォルダ
#!ln -s '/content/drive/MyDrive/Colab Notebooks/product_develop/video_tomato' './data'

!python "./demo/video_demo.py" "./data/video_tomato/tomato3.mp4"¥
        "./configs/mask_rcnn/mask_rcnn_r50_fpn_1x_coco.py"¥
        "./checkpoints/laboro_tomato_48ep.pth"¥
        --out "./data/eval_result/tomato3.mp4"
```

```
load checkpoint from local path: ./checkpoints/laboro_tomato_48ep.pth
[                                                  ] 0/1752, elapsed: 0s, ETA:/conten
t/mmdetection/mmdet/datasets/utils.py:70: UserWarning: "ImageToTensor" pipeline is rep
laced by "DefaultFormatBundle" for batch inference. It is recommended to manually repl
ace 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 chang
e. Please do not use them for anything important until they are released as stable. (T
riggered internally at  /pytorch/c10/core/TensorImpl.h:1156.)
  return torch.max_pool2d(input, kernel_size, stride, padding, dilation, ceil_mode)
[>>] 1752/1752, 5.3 task/s, elapsed: 333s, ETA:     0s
```

In [ ]: