

【全人類がわかるE資格コース PyTorch版】 プロダクト開発演習

7. YOLACT model検証

公開されているLaboro Tomato Datasetを使い、別の特徴を持つYOLACTモデルにて、インスタンス・セグメンテーションの検証を行う。

Mask R-CNNモデルと比較し、YOLACTモデルは判定(inference)速度が速く、条件次第では30fpsを上回るとされている。もしもその速度がエッジデバイスでも出るのであれば、WebCamからの動画入力に対して、ローカルにリアルタイムでセグメンテーション、クラス判定が行えると期待できるため、今回比較対象として取り上げる。

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

Mounted at /content/drive

7.1 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 = NVIDIA cuda compiler 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
!git clone https://github.com/open-mmlab/mmdetection.git -b v2.23.0 --depth 1    ### 1
%cd mmdetection

!pip install -e .
```

```

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 108 kB/s]
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)
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)
Collecting addict
    Downloading addict-2.4.0-py3-none-any.whl (3.8 kB)
Collecting yapf
    Downloading yapf-0.32.0-py2.py3-none-any.whl (190 kB)
[██████████| 190 kB 8.8 MB/s]
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% (1093/1093), done.
remote: Total 1575 (delta 614), reused 783 (delta 467), pack-reused 0
Receiving objects: 100% (1575/1575), 16.33 MiB | 15.38 MiB/s, done.
Resolving deltas: 100% (614/614), 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: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages
  (from matplotlib->mmdet==2.23.0) (0.11.0)
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: 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: 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: typing-extensions in /usr/local/lib/python3.7/dist-p
  ackages (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: - GCC 7.3 - C++ Version: 2014
  02 - Intel(R) Math Kernel Library Version 2020.0.0 Product Build 20191122 for Intel
  (R) 64 architecture applications - Intel(R) MKL-DNN v2.1.2 (Git Hash 98be7e8afa711d
  c9b66c8ff3504129cb82013cdb) - OpenMP 201511 (a. k. a. OpenMP 4.5) - NNPACK is enab
  led - CPU capability usage: AVX2 - CUDA Runtime 11.1 - NVCC architecture flag
  s: -gencode;arch=compute_37,code=sm_37;-gencode;arch=compute_50,code=sm_50;-gencode;
  arch=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
  - CuDNN 8.0.5 - Magma 2.5.2 - 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=O
  N, USE_MPI=OFF, USE_NCCL=ON, USE_NNPACK=ON, USE_OPENMP=ON, ',
  '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
```

7.2 YOLACTモデルでの学習、評価・検証

YOLACT r50 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の公開情報とMMDetectionのチュートリアルを参考に変更する。

```
In [6]: # laboro_tomato datasetを./data/にシンボリックリンクをコピー
!cd '/content/mm detection'
!mkdir -p ./data
!ln -s '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato' './data/'
```

```
In [7]: # datasets annotation coco format JSONファイルの確認 (training用、test用)
# laboro_tomato/annotations/train.json laboro_tomato/annotations/test.json

!cd /content/mm detection/
!ls -al ./data
print('`n/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato')
!ls -al '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato'

total 12
drwxr-xr-x 2 root root 4096 Jun 25 06:26 .
drwxr-xr-x 17 root root 4096 Jun 25 06:26 ..
lrwxrwxrwx 1 root root 68 Jun 25 06:26 laboro_tomato -> '/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato'

/content/drive/MyDrive/Colab Notebooks/product_develop/laboro_tomato
total 16
drwx----- 2 root root 4096 Jun 14 09:16 annotations
drwx----- 2 root root 4096 Jun 17 09:54 .ipynb_checkpoints
drwx----- 2 root root 4096 Jun 14 09:16 test
drwx----- 2 root root 4096 Jun 14 09:16 train
```

```
In [9]: # Add datasets to mm detection
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/yolact_tomato_etc/laboro_tomato.py' './mmdet/datasets/laboro_tomato.py'

# Add dataset names to mmdet/datasets/__init__.py
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/yolact_tomato_etc/__init__.py' './mmdet/datasets/__init__.py'

# step3: overwrite class numbers at model configuration file configs/yolact/yolact_r50
!cp '/content/drive/MyDrive/Colab Notebooks/product_develop/yolact_tomato_etc/yolact_r50/configs/yolact/yolact_r50_1x8_coco_tomato.py'
```

In [11]:

```
# configの切り替え to YOLACTモデルへ
# a yolact model achieving 29 mAP on MS COCO at 42.5 fps evaluated on a single GPU mod
# yolact_r50_1x8_coco.py [Enhance]: Optimize augmentation pipeline to speed up trainir

from mmcv import Config

cfg = Config.fromfile('configs/yolact/yolact_r50_1x8_coco_tomato.py')

# download the pre-trained checkpoints for inference and finetuning.
!mkdir -p checkpoints
!wget -c https://download.openmmlab.com/mmdetection/v2.0/yolact/yolact_r50_1x8_coco/yo
    -0 checkpoints/yolact_r50_1x8_coco_20200908-f38d58df.pth

--2022-06-25 06:28:02-- https://download.openmmlab.com/mmdetection/v2.0/yolact/yolact_
_r50_1x8_coco/yolact_r50_1x8_coco_20200908-f38d58df.pth
Resolving download.openmmlab.com (download.openmmlab.com)... 47.88.36.72
Connecting to download.openmmlab.com (download.openmmlab.com)|47.88.36.72|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK

The file is already fully retrieved; nothing to do.
```

In [12]:

```
# 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 = 'LaboroTomato'

### cfg.data.test.ann_file = 'data/laboro_tomato/annotation/test.json'
### cfg.data.test.img_prefix = 'data/laboro_tomato/test/'
### cfg.data.test.classes = ('LaboroTomato',)

### cfg.data.train.ann_file = 'data/laboro_tomato/annotation/train.json'
### cfg.data.train.img_prefix = 'data/laboro_tomato/train/'
### cfg.data.train.classes = ('LaboroTomato',)

### cfg.data.val.ann_file = 'data/laboro_tomato/annotation/test.json'
### cfg.data.val.img_prefix = 'data/laboro_tomato/test/'
### cfg.data.val.classes = ('LaboroTomato',)

# modify num classes of the model in box head and mask head
### cfg.model.bbox_head.num_classes = 6
### cfg.model.mask_head.num_classes = 6
### cfg.model.segm_head.num_classes = 6

# We can still use the pre-trained Mask RCNN model to obtain a higher performance
###cfg.load_from = 'checkpoints/yolact_r50_1x8_coco_20200908-f38d58df.pth'

# Set up working dir to save files and logs.
#cfg.work_dir = ./data/laboro_tomato

# 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.lr = 0.01
### cfg.lr_config.warmup = None
### cfg.log_config.interval = 10

### add by Fujiwara
### cfg.runner.max_epochs=48
```

```
# 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:

```
checkpoint_config = dict(interval=12)
log_config = dict(
    interval=50,
    hooks=[dict(type='TextLoggerHook'),
           dict(type='TensorboardLoggerHook')])
custom_hooks = [dict(type='NumClassCheckHook')]
dist_params = dict(backend='nccl')
log_level = 'INFO'
load_from = 'checkpoints/yolact_r50_1x8_coco_20200908-f38d58df.pth'
resume_from = None
workflow = [('train', 1)]
opencv_num_threads = 0
mp_start_method = 'fork'
img_size = 550
model = dict(
    type='YOLACT',
    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=True),
        norm_eval=False,
        zero_init_residual=False,
        style='pytorch',
        init_cfg=dict(type='Pretrained', checkpoint='torchvision://resnet50')),
    neck=dict(
        type='FPN',
        in_channels=[256, 512, 1024, 2048],
        out_channels=256,
        start_level=1,
        add_extra_convs='on_input',
        num_outs=5,
        upsample_cfg=dict(mode='bilinear')),
    bbox_head=dict(
        type='YOLACTHead',
        num_classes=6,
        in_channels=256,
        feat_channels=256,
        anchor_generator=dict(
            type='AnchorGenerator',
            octave_base_scale=3,
            scales_per_octave=1,
```

```

        base_sizes=[8, 16, 32, 64, 128],
        ratios=[0.5, 1.0, 2.0],
        strides=[
            7.971014492753623, 15.714285714285714, 30.5555555555555557,
            61.111111111111114, 110.0
        ],
        centers=[(3.9855072463768115, 3.9855072463768115),
                  (7.857142857142857, 7.857142857142857),
                  (15.277777777777779, 15.277777777777779),
                  (30.5555555555555557, 30.5555555555555557), (55.0, 55.0)],
        bbox_coder=dict(
            type='DeltaXYWHBoxCoder',
            target_means=[0.0, 0.0, 0.0, 0.0],
            target_stds=[0.1, 0.1, 0.2, 0.2]),
        loss_cls=dict(
            type='CrossEntropyLoss',
            use_sigmoid=False,
            reduction='none',
            loss_weight=1.0),
        loss_bbox=dict(type='SmoothL1Loss', beta=1.0, loss_weight=1.5),
        num_head_convs=1,
        num_protos=32,
        use_ohem=True),
    mask_head=dict(
        type='YOLACTProtonet',
        in_channels=256,
        num_protos=32,
        num_classes=6,
        max_masks_to_train=100,
        loss_mask_weight=6.125),
    segm_head=dict(
        type='YOLACTSegmHead',
        num_classes=6,
        in_channels=256,
        loss_segm=dict(
            type='CrossEntropyLoss', use_sigmoid=True, loss_weight=1.0)),
train_cfg=dict(
    assigner=dict(
        type='MaxIoUAssigner',
        pos_iou_thr=0.5,
        neg_iou_thr=0.4,
        min_pos_iou=0.0,
        ignore_iou_thr=-1,
        gt_max_assign_all=False),
    allowed_border=-1,
    pos_weight=-1,
    neg_pos_ratio=3,
    debug=False),
    test_cfg=dict(
        nms_pre=1000,
        min_bbox_size=0,
        score_thr=0.05,
        iou_thr=0.5,
        top_k=200,
        max_per_img=100))
dataset_type = 'LaboroTomato'
data_root = 'data/laboro_tomato/'
img_norm_cfg = dict(
    mean=[123.68, 116.78, 103.94], std=[58.4, 57.12, 57.38], to_rgb=True)
train_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='LoadAnnotations', with_bbox=True, with_mask=True),
    dict(type='FilterAnnotations', min_gt_bbox_wh=(4.0, 4.0)),
    dict(

```

```

        type='Expand',
        mean=[123. 68, 116. 78, 103. 94],
        to_rgb=True,
        ratio_range=(1, 4)),
    dict(
        type='MinIoURandomCrop',
        min_iou=(0.1, 0.3, 0.5, 0.7, 0.9),
        min_crop_size=0.3),
    dict(type='Resize', img_scale=(550, 550), keep_ratio=False),
    dict(type='RandomFlip', flip_ratio=0.5),
    dict(
        type='PhotoMetricDistortion',
        brightness_delta=32,
        contrast_range=(0.5, 1.5),
        saturation_range=(0.5, 1.5),
        hue_delta=18),
    dict(
        type='Normalize',
        mean=[123. 68, 116. 78, 103. 94],
        std=[58. 4, 57. 12, 57. 38],
        to_rgb=True),
    dict(type='DefaultFormatBundle'),
    dict(type='Collect', keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
]
test_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(
        type='MultiScaleFlipAug',
        img_scale=(550, 550),
        flip=False,
        transforms=[
            dict(type='Resize', keep_ratio=False),
            dict(
                type='Normalize',
                mean=[123. 68, 116. 78, 103. 94],
                std=[58. 4, 57. 12, 57. 38],
                to_rgb=True),
            dict(type='ImageToTensor', keys=['img']),
            dict(type='Collect', keys=['img'])
        ])
]
data = dict(
    samples_per_gpu=4,
    workers_per_gpu=1,
    train=dict(
        type='LaboroTomato',
        ann_file='data/laboro_tomato/annotations/train.json',
        img_prefix='data/laboro_tomato/train/',
        pipeline=[
            dict(type='LoadImageFromFile'),
            dict(type='LoadAnnotations', with_bbox=True, with_mask=True),
            dict(type='FilterAnnotations', min_gt_bbox_wh=(4.0, 4.0)),
            dict(
                type='Expand',
                mean=[123. 68, 116. 78, 103. 94],
                to_rgb=True,
                ratio_range=(1, 4)),
            dict(
                type='MinIoURandomCrop',
                min_iou=(0.1, 0.3, 0.5, 0.7, 0.9),
                min_crop_size=0.3),
            dict(type='Resize', img_scale=(550, 550), keep_ratio=False),
            dict(type='RandomFlip', flip_ratio=0.5),
            dict(

```

```

        type='PhotoMetricDistortion',
        brightness_delta=32,
        contrast_range=(0.5, 1.5),
        saturation_range=(0.5, 1.5),
        hue_delta=18),
    dict(
        type='Normalize',
        mean=[123.68, 116.78, 103.94],
        std=[58.4, 57.12, 57.38],
        to_rgb=True),
    dict(type='DefaultFormatBundle'),
    dict(
        type='Collect',
        keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
),
val=dict(
    type='LaboroTomato',
    ann_file='data/laboro_tomato/annotations/test.json',
    img_prefix='data/laboro_tomato/test/',
    pipeline=[
        dict(type='LoadImageFromFile'),
        dict(
            type='MultiScaleFlipAug',
            img_scale=(550, 550),
            flip=False,
            transforms=[
                dict(type='Resize', keep_ratio=False),
                dict(
                    type='Normalize',
                    mean=[123.68, 116.78, 103.94],
                    std=[58.4, 57.12, 57.38],
                    to_rgb=True),
                dict(type='ImageToTensor', keys=['img']),
                dict(type='Collect', keys=['img'])
            ])
),
test=dict(
    type='LaboroTomato',
    ann_file='data/laboro_tomato/annotations/test.json',
    img_prefix='data/laboro_tomato/test/',
    pipeline=[
        dict(type='LoadImageFromFile'),
        dict(
            type='MultiScaleFlipAug',
            img_scale=(550, 550),
            flip=False,
            transforms=[
                dict(type='Resize', keep_ratio=False),
                dict(
                    type='Normalize',
                    mean=[123.68, 116.78, 103.94],
                    std=[58.4, 57.12, 57.38],
                    to_rgb=True),
                dict(type='ImageToTensor', keys=['img']),
                dict(type='Collect', keys=['img'])
            ])
),
])
optimizer = dict(type='SGD', lr=0.001, momentum=0.9, weight_decay=0.0005)
optimizer_config = dict()
lr_config = dict(
    policy='step',
    warmup='linear',
    warmup_iters=500,
    warmup_ratio=0.1,
)

```

```

        step=[20, 42, 49, 52])
runner = dict(type='EpochBasedRunner', max_epochs=12)
cudnn_benchmark = True
evaluation = dict(metric=['bbox', 'segm'], interval=12)
seed = 0
gpu_ids = range(0, 1)

```

7.3 train dataによる学習の実行

In [13]:

```

# カスタマイズしたDetectorの学習  train

# Trains using the resnet-50 backbone with a batch size of 8 on a single GPU.
!./tools/dist_train.sh configs/yolact/yolact_r50_1x8_coco_tomato.py 1

## single-gpu train

# !python tools/train.py ./configs/yolact/yolact_r50_1x8_coco_tomato.py \
#                         --work-dir ./laboro_tomato

```

```

/usr/local/lib/python3.7/dist-packages/torch/distributed/launch.py:164: DeprecationWarning: The 'warn' method is deprecated, use 'warning' instead
    "The module torch.distributed.launch is deprecated"
The module torch.distributed.launch is deprecated and going to be removed in future. Migrate to torch.distributed.run
WARNING:torch.distributed.run:--use_env is deprecated and will be removed in future releases.
Please read local_rank from `os.environ('LOCAL_RANK')` instead.
INFO:torch.distributed.launcher.api:Starting elastic_operator with launch configs:
entrypoint      : ./tools/train.py
min_nodes       : 1
max_nodes       : 1
nproc_per_node  : 1
run_id          : none
rdzv_backend    : static
rdzv_endpoint   : 127.0.0.1:29500
rdzv_configs    : {'rank': 0, 'timeout': 900}
max_restarts    : 3
monitor_interval: 5
log_dir         : None
metrics_cfg     : {}

INFO:torch.distributed.elastic.agent.server.local_elastic_agent:log directory set to:
/tmp/torchelastic_x198_l4o/none_4hgehydz
INFO:torch.distributed.elastic.agent.server.api:[default] starting workers for entrypoint: python3
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous'ing worker group
/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/utils/store.py:53: FutureWarning: This is an experimental API and will be changed in future.
    "This is an experimental API and will be changed in future.", FutureWarning
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous complete for workers. Result:
    restart_count=0
    master_addr=127.0.0.1
    master_port=29500
    group_rank=0
    group_world_size=1
    local_ranks=[0]
    role_ranks=[0]
    global_ranks=[0]
    role_world_sizes=[1]
    global_world_sizes=[1]

```

INFO:torch.distributed.elastic.agent.server.api:[default] Starting worker group
 INFO:torch.distributed.elastic.multiprocessing:Setting worker0 reply file to: /tmp/torchelastic_x198_l4o/none_4hgehydz/attempts_0/0/error.json
 2022-06-25 06:29:20,120 - mmdet - INFO - Environment info:

```
sys.platform: linux
Python: 3.7.13 (default, Apr 24 2022, 01:04:09) [GCC 7.5.0]
CUDA available: True
GPU 0: Tesla P100-PCIE-16GB
CUDA_HOME: /usr/local/cuda
NVCC: Build cuda_11.1.TC455_06.29190527_0
GCC: gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
PyTorch: 1.9.0+cu111
PyTorch compiling details: PyTorch built with:
- GCC 7.3
- C++ Version: 201402
- Intel(R) Math Kernel Library Version 2020.0.0 Product Build 20191122 for Intel(R)
64 architecture applications
- Intel(R) MKL-DNN v2.1.2 (Git Hash 98be7e8afa711dc9b66c8ff3504129cb82013cdb)
- OpenMP 201511 (a.k.a. OpenMP 4.5)
- NNPACK is enabled
- CPU capability usage: AVX2
- CUDA Runtime 11.1
- NVCC architecture flags: -gencode;arch=compute_37,code=sm_37;-gencode;arch=compute_50,code=sm_50;-gencode;arch=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
- CuDNN 8.0.5
- Magma 2.5.2
- 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/bin/c++, CXX_FLAGS= -Wno-deprecated-d -fvisibility-inlines-hidden -DUSE_PTHREADPOOL -fopenmp -DNDEBUG -DUSE_KINETIC -DUSE_FBGEMM -DUSE_QNNPACK -DUSE_PYTORCH_QNNPACK -DUSE_XNNPACK -DSYMBOLICATE_MOBILE_DEBUG_HANDLE -O2 -fPIC -Wno-narrowing -Wall -Wextra -Werror=return-type -Wno-missing-field-initializers -Wno-type-limits -Wno-array-bounds -Wno-unknown-pragmas -Wno-sign-compare -Wno-unused-parameter -Wno-unused-variable -Wno-unused-function -Wno-unused-result -Wno-unused-local-typedefs -Wno-strict-overflow -Wno-strict-aliasing -Wno-error=deprecated-declarations -Wno-stringop-overflow -Wno-psabi -Wno-error=pedantic -Wno-error=redundant-decls -Wno-error=old-style-cast -fdiagnostics-color=always -faligned-new -Wno-unused-but-set-variable -Wno-maybe-uninitialized -fno-math-errno -fno-trapping-math -Werror=format -Wno-stringop-overflow, LAPACK_INFO=mkl, PERF_WITH_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=ON, USE_MPI=OFF, USE_NCCL=ON, USE_NNPACK=ON, USE_OPENMP=ON,
```

TorchVision: 0.10.0+cu111
 OpenCV: 4.1.2
 MMCV: 1.3.17
 MMCV Compiler: GCC 7.3
 MMCV CUDA Compiler: 11.1
 MMDetection: 2.23.0+3e26931

```
2022-06-25 06:29:20,518 - mmdet - INFO - Distributed training: True
2022-06-25 06:29:20,996 - mmdet - INFO - Config:
checkpoint_config = dict(interval=1)
log_config = dict(interval=50, hooks=[dict(type='TextLoggerHook')])
custom_hooks = [dict(type='NumClassCheckHook')]
dist_params = dict(backend='nccl')
log_level = 'INFO'
load_from = 'checkpoints/yolact_r50_1x8_coco_20200908-f38d58df.pth'
resume_from = None
workflow = [('train', 1)]
```

```
opencv_num_threads = 0
mp_start_method = 'fork'
img_size = 550
model = dict(
    type='YOLACT',
    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=True),
        norm_eval=False,
        zero_init_residual=False,
        style='pytorch',
        init_cfg=dict(type='Pretrained', checkpoint='torchvision://resnet50')),
    neck=dict(
        type='FPN',
        in_channels=[256, 512, 1024, 2048],
        out_channels=256,
        start_level=1,
        add_extra_convs='on_input',
        num_outs=5,
        upsample_cfg=dict(mode='bilinear')),
    bbox_head=dict(
        type='YOLACTHead',
        num_classes=6,
        in_channels=256,
        feat_channels=256,
        anchor_generator=dict(
            type='AnchorGenerator',
            octave_base_scale=3,
            scales_per_octave=1,
            base_sizes=[8, 16, 32, 64, 128],
            ratios=[0.5, 1.0, 2.0],
            strides=[
                7.971014492753623, 15.714285714285714, 30.555555555555557,
                61.111111111111114, 110.0
            ],
            centers=[(3.9855072463768115, 3.9855072463768115),
                     (7.857142857142857, 7.857142857142857),
                     (15.277777777777779, 15.277777777777779),
                     (30.555555555555557, 30.555555555555557), (55.0, 55.0)]),
        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]),
        loss_cls=dict(
            type='CrossEntropyLoss',
            use_sigmoid=False,
            reduction='none',
            loss_weight=1.0),
        loss_bbox=dict(type='SmoothL1Loss', beta=1.0, loss_weight=1.5),
        num_head_convs=1,
        num_protos=32,
        use_ohem=True),
    mask_head=dict(
        type='YOLACTProtonet',
        in_channels=256,
        num_protos=32,
        num_classes=6,
        max_masks_to_train=100,
        loss_mask_weight=6.125),
    segm_head=dict()
```

```

        type='YOLOACTSegmHead',
        num_classes=6,
        in_channels=256,
        loss_segm=dict(
            type='CrossEntropyLoss', use_sigmoid=True, loss_weight=1.0)),
train_cfg=dict(
    assigner=dict(
        type='MaxIoUAssigner',
        pos_iou_thr=0.5,
        neg_iou_thr=0.4,
        min_pos_iou=0.0,
        ignore_iou_thr=-1,
        gt_max_assign_all=False),
    allowed_border=-1,
    pos_weight=-1,
    neg_pos_ratio=3,
    debug=False),
test_cfg=dict(
    nms_pre=1000,
    min_bbox_size=0,
    score_thr=0.05,
    iou_thr=0.5,
    top_k=200,
    max_per_img=100))
dataset_type = 'LaboroTomato'
data_root = 'data/laboro_tomato/'
img_norm_cfg = dict(
    mean=[123.68, 116.78, 103.94], std=[58.4, 57.12, 57.38], to_rgb=True)
train_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='LoadAnnotations', with_bbox=True, with_mask=True),
    dict(type='FilterAnnotations', min_gt_bbox_wh=(4.0, 4.0)),
    dict(
        type='Expand',
        mean=[123.68, 116.78, 103.94],
        to_rgb=True,
        ratio_range=(1, 4)),
    dict(
        type='MinIoURandomCrop',
        min_ious=(0.1, 0.3, 0.5, 0.7, 0.9),
        min_crop_size=0.3),
    dict(type='Resize', img_scale=(550, 550), keep_ratio=False),
    dict(type='RandomFlip', flip_ratio=0.5),
    dict(
        type='PhotoMetricDistortion',
        brightness_delta=32,
        contrast_range=(0.5, 1.5),
        saturation_range=(0.5, 1.5),
        hue_delta=18),
    dict(
        type='Normalize',
        mean=[123.68, 116.78, 103.94],
        std=[58.4, 57.12, 57.38],
        to_rgb=True),
    dict(type='DefaultFormatBundle'),
    dict(type='Collect', keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
]
test_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(
        type='MultiScaleFlipAug',
        img_scale=(550, 550),
        flip=False,
        transforms=[

```

```

        dict(type='Resize', keep_ratio=False),
        dict(
            type='Normalize',
            mean=[123. 68, 116. 78, 103. 94],
            std=[58. 4, 57. 12, 57. 38],
            to_rgb=True),
        dict(type='ImageToTensor', keys=['img']),
        dict(type='Collect', keys=['img'])
    ])
]
data = dict(
    samples_per_gpu=4,
    workers_per_gpu=1,
    train=dict(
        type='LaboroTomato',
        ann_file='data/laboro_tomato/annotations/train.json',
        img_prefix='data/laboro_tomato/train/',
        pipeline=[
            dict(type='LoadImageFromFile'),
            dict(type='LoadAnnotations', with_bbox=True, with_mask=True),
            dict(type='FilterAnnotations', min_gt_bbox_wh=(4.0, 4.0)),
            dict(
                type='Expand',
                mean=[123. 68, 116. 78, 103. 94],
                to_rgb=True,
                ratio_range=(1, 4)),
            dict(
                type='MinIoURandomCrop',
                min_ious=(0.1, 0.3, 0.5, 0.7, 0.9),
                min_crop_size=0.3),
            dict(type='Resize', img_scale=(550, 550), keep_ratio=False),
            dict(type='RandomFlip', flip_ratio=0.5),
            dict(
                type='PhotoMetricDistortion',
                brightness_delta=32,
                contrast_range=(0.5, 1.5),
                saturation_range=(0.5, 1.5),
                hue_delta=18),
            dict(
                type='Normalize',
                mean=[123. 68, 116. 78, 103. 94],
                std=[58. 4, 57. 12, 57. 38],
                to_rgb=True),
            dict(type='DefaultFormatBundle'),
            dict(
                type='Collect',
                keys=['img', 'gt_bboxes', 'gt_labels', 'gt_masks'])
        ]),
    val=dict(
        type='LaboroTomato',
        ann_file='data/laboro_tomato/annotations/test.json',
        img_prefix='data/laboro_tomato/test/',
        pipeline=[
            dict(type='LoadImageFromFile'),
            dict(
                type='MultiScaleFlipAug',
                img_scale=(550, 550),
                flip=False,
                transforms=[
                    dict(type='Resize', keep_ratio=False),
                    dict(
                        type='Normalize',
                        mean=[123. 68, 116. 78, 103. 94],
                        std=[58. 4, 57. 12, 57. 38],

```

```

        to_rgb=True),
        dict(type='ImageToTensor', keys=['img']),
        dict(type='Collect', keys=['img']))
    ],
    test=dict(
        type='LaboroTomato',
        ann_file='data/laboro_tomato/annotations/test.json',
        img_prefix='data/laboro_tomato/test/',
        pipeline=[
            dict(type='LoadImageFromFile'),
            dict(
                type='MultiScaleFlipAug',
                img_scale=(550, 550),
                flip=False,
                transforms=[
                    dict(type='Resize', keep_ratio=False),
                    dict(
                        type='Normalize',
                        mean=[123.68, 116.78, 103.94],
                        std=[58.4, 57.12, 57.38],
                        to_rgb=True),
                    dict(type='ImageToTensor', keys=['img']),
                    dict(type='Collect', keys=['img'])
                ])
            ],
        )))
optimizer = dict(type='SGD', lr=0.001, momentum=0.9, weight_decay=0.0005)
optimizer_config = dict()
lr_config = dict(
    policy='step',
    warmup='linear',
    warmup_iters=500,
    warmup_ratio=0.1,
    step=[20, 42, 49, 52])
runner = dict(type='EpochBasedRunner', max_epochs=12)
cudnn_benchmark = True
evaluation = dict(metric=['bbox', 'segm'])
work_dir = './work_dirs/yolact_r50_1x8_coco_tomato'
auto_resume = False
gpu_ids = range(0, 1)

```

```

2022-06-25 06:29:21,097 - mmdet - INFO - Set random seed to 0, deterministic: False
2022-06-25 06:29:21,493 - mmdet - INFO - initialize ResNet with init_cfg {'type': 'Pretrained', 'checkpoint': 'torchvision://resnet50'}
2022-06-25 06:29:21,494 - mmcv - INFO - load model from: torchvision://resnet50
2022-06-25 06:29:21,494 - mmcv - INFO - load checkpoint from torchvision path: torchvision://resnet50
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to /root/.cache/torch/hub/checkpoints/resnet50-0676ba61.pth
100% 97.8M/97.8M [00:00<00:00, 155MB/s]
2022-06-25 06:29:22,363 - mmcv - WARNING - The model and loaded state dict do not match exactly

```

unexpected key in source state_dict: fc.weight, fc.bias

```

2022-06-25 06:29:22,389 - mmdet - INFO - initialize FPN with init_cfg {'type': 'Xavier', 'layer': 'Conv2d', 'distribution': 'uniform'}
2022-06-25 06:29:22,448 - mmdet - INFO - initialize YOLACTHead with init_cfg {'type': 'Xavier', 'distribution': 'uniform', 'bias': 0, 'layer': 'Conv2d'}
2022-06-25 06:29:22,460 - mmdet - INFO - initialize YOLACTSegmHead with init_cfg {'type': 'Xavier', 'distribution': 'uniform', 'override': {'name': 'segm_conv'}}
2022-06-25 06:29:22,461 - mmdet - INFO - initialize YOLACTProtonet with init_cfg {'type': 'Xavier', 'distribution': 'uniform', 'override': {'name': 'protonet'}}
loading annotations into memory...

```

```
Done (t=0.57s)
creating index...
index created!
loading annotations into memory...
Done (t=0.37s)
creating index...
index created!
2022-06-25 06:29:26,510 - mmdet - INFO - load checkpoint from local path: checkpoints/
yolact_r50_1x8_coco_20200908-f38d58df.pth
2022-06-25 06:29:26,642 - mmdet - WARNING - The model and loaded state dict do not match exactly

size mismatch for bbox_head.conv_cls.weight: copying a param with shape torch.Size([24 3, 256, 3, 3]) from checkpoint, the shape in current model is torch.Size([21, 256, 3, 3]).
size mismatch for bbox_head.conv_cls.bias: copying a param with shape torch.Size([24 3]) from checkpoint, the shape in current model is torch.Size([21]).
size mismatch for segm_head.segm_conv.weight: copying a param with shape torch.Size([8 0, 256, 1, 1]) from checkpoint, the shape in current model is torch.Size([6, 256, 1, 1]).
size mismatch for segm_head.segm_conv.bias: copying a param with shape torch.Size([8 0]) from checkpoint, the shape in current model is torch.Size([6]).
2022-06-25 06:29:26,646 - mmdet - INFO - Start running, host: root@35f615b8aec0, work_dir: /content/mmdetection/work_dirs/yolact_r50_1x8_coco_tomato
2022-06-25 06:29:26,646 - mmdet - INFO - Hooks will be executed in the following order:
before_run:
(VERY_HIGH   ) StepLrUpdaterHook
(NORMAL      ) CheckpointHook
(LOW         ) DistEvalHook
(VERY_LOW    ) TextLoggerHook
-----
before_train_epoch:
(VERY_HIGH   ) StepLrUpdaterHook
(NORMAL      ) NumClassCheckHook
(NORMAL      ) DistSamplerSeedHook
(LOW         ) IterTimerHook
(LOW         ) DistEvalHook
(VERY_LOW    ) TextLoggerHook
-----
before_train_iter:
(VERY_HIGH   ) StepLrUpdaterHook
(LOW         ) IterTimerHook
(LOW         ) DistEvalHook
-----
after_train_iter:
(ABOVE_NORMAL) OptimizerHook
(NORMAL      ) CheckpointHook
(LOW         ) IterTimerHook
(LOW         ) DistEvalHook
(VERY_LOW    ) TextLoggerHook
-----
after_train_epoch:
(NORMAL      ) CheckpointHook
(LOW         ) DistEvalHook
(VERY_LOW    ) TextLoggerHook
-----
before_val_epoch:
(NORMAL      ) NumClassCheckHook
(NORMAL      ) DistSamplerSeedHook
(LOW         ) IterTimerHook
(VERY_LOW    ) TextLoggerHook
-----
before_val_iter:
```

```
(LOW      ) IterTimerHook
-----
after_val_iter:
(LOW      ) IterTimerHook
-----
after_val_epoch:
(VERY_LOW ) TextLoggerHook
-----
after_run:
(VERY_LOW ) TextLoggerHook
-----
2022-06-25 06:29:26,647 - mmdet - INFO - workflow: [('train', 1)], max: 12 epochs
2022-06-25 06:29:26,647 - mmdet - INFO - Checkpoints will be saved to /content/mmdetection/work_dirs/yolact_r50_1x8_coco_tomato by HardDiskBackend.
/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)
tcmalloc: large alloc 1132191744 bytes == 0xab93e000 @ 0x7f7cc88b5001 0x7f7cc61c41af
0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511
e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0
x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e
0e 0x593fce 0x511e2c
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Default upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample for details.
    "See the documentation of nn.Upsample for details.".format(mode)
tcmalloc: large alloc 3055255552 bytes == 0x7f7a8de48000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 3055255552 bytes == 0x7f79d7c90000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 06:29:57,648 - mmcv - INFO - Reducer buckets have been rebuilt in this iteration.
tcmalloc: large alloc 1715789824 bytes == 0x8358a000 @ 0x7f7cc88b5001 0x7f7cc61c41af
0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511
e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0
x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e
0e 0x593fce 0x511e2c
tcmalloc: large alloc 2899779584 bytes == 0x7f79d7c90000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 2899779584 bytes == 0x7f7a84a02000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 2286108672 bytes == 0x7f79d7c90000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 3351330816 bytes == 0x7f79d7c90000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
```

6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 3351330816 bytes == 0x7f791007c000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 7535525888 bytes == 0x7f791007c000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 7535525888 bytes == 0x7f774ee0c000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 6368739328 bytes == 0x7f774ee0c000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 06:37:05, 331 - mmdet - INFO - Epoch [1][50/161] lr: 1.882e-04, eta: 4:
47:44, time: 9.173, data_time: 8.774, memory: 11156, loss_cls: 6.3770, loss_bbox: 1.72
87, loss_segm: 1.2932, loss_mask: 2.9444, loss: 12.3432
tcmalloc: large alloc 6816825344 bytes == 0x7f774ee0c000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 6816825344 bytes == 0x7f78e5314000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 06:44:42, 516 - mmdet - INFO - Epoch [1][100/161] lr: 2.782e-04, eta: 4:
39:38, time: 9.144, data_time: 8.888, memory: 11156, loss_cls: 3.9494, loss_bbox: 1.47
32, loss_segm: 0.4750, loss_mask: 2.6606, loss: 8.5582
tcmalloc: large alloc 7363977216 bytes == 0x7f774ee0c000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 7363977216 bytes == 0x7f7905ce2000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 06:52:34, 873 - mmdet - INFO - Epoch [1][150/161] lr: 3.682e-04, eta: 4:
34:51, time: 9.447, data_time: 9.191, memory: 11156, loss_cls: 3.5532, loss_bbox: 1.48
35, loss_segm: 0.3855, loss_mask: 2.6015, loss: 8.0237
2022-06-25 06:54:13, 065 - mmdet - INFO - Saving checkpoint at 1 epochs
[] 0/161, elapsed: 0s, ETA:tcmalloc:
large alloc 1219280896 bytes == 0x8358a000 @ 0x7f7cc8895b6b 0x7f7cc88b5379 0x7f7c11d7
126e 0x7f7c11d729e2 0x7f7c13f9ed43 0x7f7c134d4e4b 0x7f7c13c2febfb 0x7f7c13c14750 0x7f7c
1381d724 0x7f7c134c8498 0x7f7c13da25e0 0x7f7c13f6d4e7 0x7f7cb6683dbe 0x7f7cb67cfdee 0x
59b1b0 0x515655 0x593dd7 0x548ae9 0x51566f 0x549e0e 0x593fce 0x548ae9 0x5127f1 0x549e0
e 0x4bca8a 0x59c019 0x595ef6 0x5134a6 0x549576 0x4bca8a 0x59c019
tcmalloc: large alloc 1219280896 bytes == 0xcc056000 @ 0x7f7cc88b31e7 0x7f7cc61c40ce

0x7f7cc621e726 0x7f7cc62b3841 0x59b076 0x515655 0x593dd7 0x548ae9 0x51566f 0x549e0e 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bca8a 0x59c019 0x595ef6 0x5134a6 0x549576 0x4bca8a 0x59c019 0x595ef6 0x5134a6 0x549576 0x4bca8a 0x5134a6 0x549e0e 0x4bca8a 0x59c019 0x595ef6 0x5134a6
 [>>] 161/161, 0.3 task/s, elapsed: 634s, ETA: 0s

2022-06-25 07:04:49,156 - mmdet - INFO - Evaluating bbox...

Loading and preparing results...

DONE (t=0.12s)

creating index...

index created!

Running per image evaluation...

Evaluate annotation type *bbox*

DONE (t=1.52s).

Accumulating evaluation results...

DONE (t=0.25s).

2022-06-25 07:04:51,076 - mmdet - INFO -

Average Precision (AP) @[IoU=0.50:0.95 area= all maxDets=100]	= 0.146
Average Precision (AP) @[IoU=0.50 area= all maxDets=1000]	= 0.228
Average Precision (AP) @[IoU=0.75 area= all maxDets=1000]	= 0.169
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]	= 0.009
Average Precision (AP) @[IoU=0.50:0.95 area= large maxDets=1000]	= 0.160
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=100]	= 0.501
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=300]	= 0.501
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=1000]	= 0.501
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]	= 0.012
Average Recall (AR) @[IoU=0.50:0.95 area= large maxDets=1000]	= 0.542

2022-06-25 07:04:51,076 - mmdet - INFO - Evaluating segm...

/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)

Loading and preparing results...

DONE (t=0.22s)

creating index...

index created!

Running per image evaluation...

Evaluate annotation type *segm*

DONE (t=2.06s).

Accumulating evaluation results...

/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)

DONE (t=0.26s).

2022-06-25 07:04:53,741 - mmdet - INFO -

Average Precision (AP) @[IoU=0.50:0.95 area= all maxDets=100]	= 0.159
Average Precision (AP) @[IoU=0.50 area= all maxDets=1000]	= 0.225
Average Precision (AP) @[IoU=0.75 area= all maxDets=1000]	= 0.178
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]	= 0.006
Average Precision (AP) @[IoU=0.50:0.95 area= large maxDets=1000]	= 0.175
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=100]	= 0.548
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=300]	= 0.548
Average Recall (AR) @[IoU=0.50:0.95 area= all maxDets=1000]	= 0.548
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]	= 0.012
Average Recall (AR) @[IoU=0.50:0.95 area= large maxDets=1000]	= 0.592

2022-06-25 07:04:53, 768 - mmdet - INFO - Exp name: yolact_r50_1x8_coco_tomato.py
2022-06-25 07:04:53, 768 - mmdet - INFO - Epoch [1][161] bbox_mAP: 0.1460, bbox_mAP_50: 0.2280, bbox_mAP_75: 0.1690, bbox_mAP_s: 0.0000, bbox_mAP_m: 0.0090, bbox_mAP_l: 0.1600, bbox_mAP_copypaste: 0.146 0.228 0.169 0.000 0.009 0.160, segm_mAP: 0.1590, segm_mAP_50: 0.2250, segm_mAP_75: 0.1780, segm_mAP_s: 0.0000, segm_mAP_m: 0.0060, segm_mAP_l: 0.1750, segm_mAP_copypaste: 0.159 0.225 0.178 0.000 0.006 0.175
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Default upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample for details.
"See the documentation of nn.Upsample for details.".format(mode)
tcmalloc: large alloc 2949111808 bytes == 0x14566e000 @ 0x7f7cc88b5001 0x7f7cc61c41af
0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511
e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0
x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e
0e 0x593fce 0x511e2c
tcmalloc: large alloc 2949111808 bytes == 0x7f7699802000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 2949111808 bytes == 0x14566e000 @ 0x7f7cc88b31e7 0x7f7cc61c40ce
0x7f7cc621acf5 0x7f7cc621ae08 0x7f7cc62ad0b9 0x7f7cc62afa25 0x4d3969 0x512147 0x4bc98a
0x533274 0x4d3969 0x512147 0x549e0e 0x4bcb19 0x532e76 0x594b72 0x515600 0x4bc98a 0x532
e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0
x549e0e 0x593fce 0x5118f8
tcmalloc: large alloc 2949111808 bytes == 0x7f75e9b84000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x4bc98a 0x533274 0x4d3969 0x512147
tcmalloc: large alloc 2436276224 bytes == 0x14566e000 @ 0x7f7cc88b31e7 0x7f7cc61c40ce
0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc6205944 0x
59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc6205944 0
x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x5156
00 0x593dd7 0x511e2c 0x549e0e 0x4bcb19
tcmalloc: large alloc 2485755904 bytes == 0x8358a000 @ 0x7f7cc88b5001 0x7f7cc61c41af
0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511
e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0
x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e
0e 0x593fce 0x511e2c
tcmalloc: large alloc 5248884736 bytes == 0x7f75e9b84000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 5248884736 bytes == 0x7f74b0dca000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 4346945536 bytes == 0x7f74b0dca000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0
x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x5156
6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x
549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 4346945536 bytes == 0x7f75b3f5c000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 07:12:18, 488 - mmdet - INFO - Epoch [2][50/161] lr: 4.780e-04, eta: 4:
09:09, time: 8.894, data_time: 8.636, memory: 11156, loss_cls: 3.2713, loss_bbox: 1.39

95, loss_segm: 0.3044, loss_mask: 2.5994, loss: 7.5745
tcmalloc: large alloc 5435564032 bytes == 0x7f73c7012000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 5435564032 bytes == 0x7f750afd4000 @ 0x7f7cc88b31e7 0x7f7cc61c40ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc6205944 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059444 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 7037829120 bytes == 0x7f7283050000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 7037829120 bytes == 0x7f742681c000 @ 0x7f7cc88b31e7 0x7f7cc61c40ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc6205944 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059444 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 07:19:51, 793 - mmdet - INFO - Epoch [2][100/161] lr: 5.680e-04, eta: 4:03:57, time: 9.066, data_time: 8.807, memory: 11156, loss_cls: 2.9833, loss_bbox: 1.2190, loss_segm: 0.3547, loss_mask: 2.3476, loss: 6.9046
2022-06-25 07:26:58, 557 - mmdet - INFO - Epoch [2][150/161] lr: 6.580e-04, eta: 3:55:40, time: 8.535, data_time: 8.277, memory: 11156, loss_cls: 2.8339, loss_bbox: 1.2458, loss_segm: 0.2944, loss_mask: 2.3149, loss: 6.6890
2022-06-25 07:28:44, 740 - mmdet - INFO - Saving checkpoint at 2 epochs
[>>] 161/161, 0.1 task/s, elapsed: 1202s, ETA: 0s

2022-06-25 07:48:49, 185 - mmdet - INFO - Evaluating bbox...

Loading and preparing results...

DONE (t=0.01s)

creating index...

index created!

Running per image evaluation...

Evaluate annotation type *bbox*

DONE (t=1.67s).

Accumulating evaluation results...

DONE (t=0.28s).

2022-06-25 07:48:51, 327 - mmdet - INFO -

Average Precision (AP)	@[IoU=0.50:0.95]	area=	all	maxDets=100	= 0.269
Average Precision (AP)	@[IoU=0.50]	area=	all	maxDets=1000	= 0.426
Average Precision (AP)	@[IoU=0.75]	area=	all	maxDets=1000	= 0.314
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	= 0.076
Average Precision (AP)	@[IoU=0.50:0.95]	area=	large	maxDets=1000	= 0.292
Average Recall (AR)	@[IoU=0.50:0.95]	area=	all	maxDets=100	= 0.546
Average Recall (AR)	@[IoU=0.50:0.95]	area=	all	maxDets=300	= 0.546
Average Recall (AR)	@[IoU=0.50:0.95]	area=	all	maxDets=1000	= 0.546
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	= 0.103
Average Recall (AR)	@[IoU=0.50:0.95]	area=	large	maxDets=1000	= 0.586

2022-06-25 07:48:51, 328 - mmdet - INFO - Evaluating segm...

/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)

Loading and preparing results...

DONE (t=0.19s)

creating index...

index created!

Running per image evaluation...

```
Evaluate annotation type *segm*
DONE (t=1.89s).
```

```
Accumulating evaluation results...
```

```
/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)
```

```
DONE (t=0.28s).
```

```
2022-06-25 07:48:53,934 - mmdet - INFO -
```

Average Precision (AP)	@[IoU=0.50:0.95]	area= all	maxDets=100	= 0.285
Average Precision (AP)	@[IoU=0.50]	area= all	maxDets=1000	= 0.415
Average Precision (AP)	@[IoU=0.75]	area= all	maxDets=1000	= 0.312
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	= 0.008
Average Precision (AP)	@[IoU=0.50:0.95]	area= large	maxDets=1000	= 0.310
Average Recall (AR)	@[IoU=0.50:0.95]	area= all	maxDets=100	= 0.584
Average Recall (AR)	@[IoU=0.50:0.95]	area= all	maxDets=300	= 0.584
Average Recall (AR)	@[IoU=0.50:0.95]	area= all	maxDets=1000	= 0.584
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	= 0.044
Average Recall (AR)	@[IoU=0.50:0.95]	area= large	maxDets=1000	= 0.626

```
2022-06-25 07:48:53,960 - mmdet - INFO - Exp name: yolact_r50_1x8_coco_tomato.py
```

```
2022-06-25 07:48:53,960 - mmdet - INFO - Epoch(val) [2][161] bbox_mAP: 0.2690, bbox_mAP_50: 0.4260, bbox_mAP_75: 0.3140, bbox_mAP_s: 0.0000, bbox_mAP_m: 0.0760, bbox_mAP_l: 0.2920, bbox_mAP_copypaste: 0.269 0.426 0.314 0.000 0.076 0.292, segm_mAP: 0.2850, segm_mAP_50: 0.4150, segm_mAP_75: 0.3120, segm_mAP_s: 0.0000, segm_mAP_m: 0.0080, segm_mAP_l: 0.3100, segm_mAP_copypaste: 0.285 0.415 0.312 0.000 0.008 0.310
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Default upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample for details.
```

```
"See the documentation of nn.Upsample for details.".format(mode)
```

```
tcmalloc: large alloc 1464639488 bytes == 0xa0724000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 5879439360 bytes == 0x7f75930a4000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 5879439360 bytes == 0x7f7434992000 @ 0x7f7cc88b31e7 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 5879439360 bytes == 0x7f7434992000 @ 0x7f7cc88b31e7 0x7f7cc61c41af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bc98a 0x7f7cc620594 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 3040116736 bytes == 0x8358a000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 3040116736 bytes == 0x7f7434992000 @ 0x7f7cc88b31e7 0x7f7cc61c41af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bc98a 0x7f7cc620594 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 4332331008 bytes == 0x7f7434992000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 4332331008 bytes == 0x7f7434992000 @ 0x7f7cc88b5001 0x7f7cc61c41af 0x7f7cc621ac23 0x7f7cc621ba87 0x7f7cc62bd823 0x5936cc 0x548c51 0x5127f1 0x593dd7 0x511e2c 0x4bc98a 0x532e76 0x594b72 0x515600 0x4bc98a 0x532e76 0x594b72 0x548cc1 0x51566f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
```

6f 0x593dd7 0x511e2c 0x4bc98a 0x533274 0x4d3969 0x512147 0x549e0e 0x593fce 0x5118f8 0x549e0e 0x593fce 0x511e2c
tcmalloc: large alloc 4332331008 bytes == 0x7f7536d34000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 3636903936 bytes == 0x7f7434992000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 3636903936 bytes == 0x7f750d5fe000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 4507852800 bytes == 0x7f7434992000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 4507852800 bytes == 0x7f7541498000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 5481791488 bytes == 0x7f7434992000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
2022-06-25 07:54:25, 152 - mmdet - INFO - Epoch [3][50/161] lr: 7.678e-04, eta: 3:
32:45, time: 6.623, data_time: 6.365, memory: 11156, loss_cls: 2.6653, loss_bbox: 1.25
15, loss_segm: 0.2750, loss_mask: 2.4019, loss: 6.5937
tcmalloc: large alloc 5751922688 bytes == 0x7f735e2cc000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 9699958784 bytes == 0x7f735e2cc000 @ 0x7f7cc88b5001 0x7f7cc61c4
1af 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 9699958784 bytes == 0x7f711c032000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 9410412544 bytes == 0x7f735e2cc000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x593dd7 0x511e2c 0x4bc98a 0x532e76
tcmalloc: large alloc 9410412544 bytes == 0x7f6eeb1ba000 @ 0x7f7cc88b31e7 0x7f7cc61c4
0ce 0x7f7cc621acf5 0x7f7cc62c386d 0x7f7cc62c417f 0x7f7cc62c42d0 0x4bc4ab 0x7f7cc620594
4 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bcb19 0x7f7cc62059
44 0x59371f 0x515244 0x549576 0x593fce 0x548ae9 0x51566f 0x4bc98a 0x532b86 0x594a96 0x
515600 0x4bc98a 0x533274 0x4d3969 0x512147
Traceback (most recent call last):
File "/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py", line 9
90, in _try_get_data
data = self._data_queue.get(timeout=timeout)

```

File "/usr/lib/python3.7/multiprocessing/queues.py", line 104, in get
    if not self._poll(timeout):
File "/usr/lib/python3.7/multiprocessing/connection.py", line 257, in poll
    return self._poll(timeout)
File "/usr/lib/python3.7/multiprocessing/connection.py", line 414, in _poll
    r = wait([self], timeout)
File "/usr/lib/python3.7/multiprocessing/connection.py", line 921, in wait
    ready = selector.select(timeout)
File "/usr/lib/python3.7/selectors.py", line 415, in select
    fd_event_list = self._selector.poll(timeout)
File "/usr/local/lib/python3.7/dist-packages/torch/utils/data/_utils/signal_handling.py", line 66, in handler
    _error_if_any_worker_fails()
RuntimeError: DataLoader worker (pid 1764) is killed by signal: Killed.

```

The above exception was the direct cause of the following exception:

```

Traceback (most recent call last):
  File "./tools/train.py", line 220, in <module>
    main()
  File "./tools/train.py", line 216, in main
    meta=meta)
  File "/content/mmdetection/mmdet/apis/train.py", line 208, in train_detector
    runner.run(data_loaders, cfg.workflow)
  File "/usr/local/lib/python3.7/dist-packages/mmcv/runner/epoch_based_runner.py", line 127, in run
    epoch_runner(data_loaders[i], **kwargs)
  File "/usr/local/lib/python3.7/dist-packages/mmcv/runner/epoch_based_runner.py", line 47, in train
    for i, data_batch in enumerate(self.data_loader):
  File "/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py", line 521, in __next__
    data = self._next_data()
  File "/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py", line 186, in _next_data
    idx, data = self._get_data()
    success, data = self._try_get_data()
  File "/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py", line 1003, in _try_get_data
    raise RuntimeError('DataLoader worker (pid(s) {}) exited unexpectedly'.format(pids_str)) from e
RuntimeError: DataLoader worker (pid(s) 1764) exited unexpectedly
ERROR:torch.distributed.elastic.multiprocessing.api:failed (exitcode: 1) local_rank: 0 (pid: 604) of binary: /usr/bin/python3
ERROR:torch.distributed.elastic.agent.server.local_elastic_agent:[default] Worker group failed
INFO:torch.distributed.elastic.agent.server.api:[default] Worker group FAILED. 3/3 attempts left; will restart worker group
INFO:torch.distributed.elastic.agent.server.api:[default] Stopping worker group
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous'ing worker group
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous complete for workers. Result:
  restart_count=1
  master_addr=127.0.0.1
  master_port=29500
  group_rank=0
  group_world_size=1
  local_ranks=[0]
  role_ranks=[0]
  global_ranks=[0]
  role_world_sizes=[1]
  global_world_sizes=[1]

```

```

INFO:torch.distributed.elastic.agent.server.api:[default] Starting worker group
INFO:torch.distributed.elastic.multiprocessing:Setting worker0 reply file to: /tmp/tor
chelastic_x198_l4o/none_4hgehydz/attempts_1/0/error.json
Traceback (most recent call last):
  File "/usr/lib/python3.7/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/usr/lib/python3.7/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/launch.py", line 173,
in <module>
    main()
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/launch.py", line 169,
in main
    run(args)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/run.py", line 624, in
run
    )(*cmd_args)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/launcher/api.py", lin
e 116, in __call__
    return launch_agent(self._config, self._entrypoint, list(args))
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/multiprocessi
ng/errors/_init_.py", line 348, in wrapper
    return f(*args, **kwargs)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/launcher/api.py", lin
e 238, in launch_agent
    result = agent.run()
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/metrics/api.p
y", line 125, in wrapper
    result = f(*args, **kwargs)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/agent/server/
api.py", line 700, in run
    result = self._invoke_run(role)
  File "/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/agent/server/
api.py", line 828, in _invoke_run
    time.sleep(monitor_interval)
KeyboardInterrupt

```

```

In [ ]: # load tensorboard in colab
%load_ext tensorboard

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

```

7.4 test dataによるYOLACT学習modelの正当性確認

学習中Epoch2まで完了の後、Epoch3途中で google ColabのRun Timeエラー発生！！

エラー発生前のEpoch 2までの学習結果をtestデータを使い評価する。このためconfigデータを書き換える

```

In [18]: # overwrite class numbers at model configuration file configs/yolact_r50_1x8_cc
!cp 'work_dirs/yolact_r50_1x8_coco_tomato/yolact_r50_1x8_coco_tomato.py' @@
'./configs/yolact_r50_1x8_coco_tomato_20220625-ep2.py'

# download the latest trained checkpoints for inference and finetuning.
# !mkdir -p checkpoints
!cp 'work_dirs/yolact_r50_1x8_coco_tomato/epoch_2.pth' @@
'./checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth'

# test datasetsによる評価 (test) 以下のスクリプトを利用できる

```

```

config = 'configs/yolact/yolact_r50_1x8_coco_tomato_20220625-ep2.py'
checkpoint = 'checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth'

# Trains using the resnet-50 backbone with a batch size of 8 on a single GPU.
!./tools/dist_test.sh 'configs/yolact/yolact_r50_1x8_coco_tomato_20220625-ep2.py' \
    'checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth' 1¥
    --show --eval "bbox" "segm"

```

```

/usr/local/lib/python3.7/dist-packages/torch/distributed/launch.py:164: DeprecationWarning: The 'warn' method is deprecated, use 'warning' instead
    "The module torch.distributed.launch is deprecated"
The module torch.distributed.launch is deprecated and going to be removed in future. Migrate to torch.distributed.run
WARNING:torch.distributed.run:--use_env is deprecated and will be removed in future releases.
Please read local_rank from `os.environ('LOCAL_RANK')` instead.
INFO:torch.distributed.launcher.api:Starting elastic_operator with launch configs:
entrypoint      : ./tools/test.py
min_nodes       : 1
max_nodes       : 1
nproc_per_node  : 1
run_id          : none
rdzv_backend    : static
rdzv_endpoint   : 127.0.0.1:29500
rdzv_configs    : {'rank': 0, 'timeout': 900}
max_restarts    : 3
monitor_interval: 5
log_dir         : None
metrics_cfg     : {}

INFO:torch.distributed.elastic.agent.server.local_elastic_agent:log directory set to:
/tmp/torchelastic_tfdmr219/none_4p3mnfa8
INFO:torch.distributed.elastic.agent.server.api:[default] starting workers for entrypoint: python3
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous'ing worker group
/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/utils/store.py:53: FutureWarning: This is an experimental API and will be changed in future.
    "This is an experimental API and will be changed in future.", FutureWarning
INFO:torch.distributed.elastic.agent.server.api:[default] Rendezvous complete for workers. Result:
    restart_count=0
    master_addr=127.0.0.1
    master_port=29500
    group_rank=0
    group_world_size=1
    local_ranks=[0]
    role_ranks=[0]
    global_ranks=[0]
    role_world_sizes=[1]
    global_world_sizes=[1]

INFO:torch.distributed.elastic.agent.server.api:[default] Starting worker group
INFO:torch.distributed.elastic.multiprocessing:Setting worker0 reply file to: /tmp/torchelastic_tfdmr219/none_4p3mnfa8/attempts_0/0/error.json
loading annotations into memory...
Done (t=0.04s)
creating index...
index created!
load checkpoint from local path: checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth
[                                         ] 0/161, elapsed: 0s, ETA:/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 i

```

```

internally at /pytorch/c10/core/TensorImpl.h:1156.)
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, ceil_mode)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Default upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample for details.
    "See the documentation of nn.Upsample for details.".format(mode)
[ ] 2/161, 0.2 task/s, elapsed: 13s, ETA: 1002stcmalloc: large alloc 1219280896 bytes == 0x92a8a000 @ 0x7f4b6238fb6b 0x7f4b623af379 0x7f4aab86b26e 0x7f4aab86c9e2 0x7f4aa
da98d43 0x7f4aacfccee4b 0x7f4aad729ebf 0x7f4aad70e750 0x7f4aad317724 0x7f4aacfc2498 0x7
f4aad89c5e0 0x7f4aada674e7 0x7f4b5017ddbe 0x7f4b502c9dee 0x59b1b0 0x515655 0x593dd7 0x
548ae9 0x51566f 0x549e0e 0x593fce 0x548ae9 0x5127f1 0x549e0e 0x4bca8a 0x59c019 0x595ef
6 0x5134a6 0x549576 0x4bca8a 0x59c019
tcmalloc: large alloc 1219280896 bytes == 0xdb556000 @ 0x7f4b623ad1e7 0x7f4b5fcbe0ce
0x7f4b5fd18726 0x7f4b5fdad841 0x59b076 0x515655 0x593dd7 0x548ae9 0x51566f 0x549e0e 0x
593fce 0x548ae9 0x5127f1 0x549e0e 0x4bca8a 0x59c019 0x595ef6 0x5134a6 0x549576 0x4bca8
a 0x59c019 0x595ef6 0x5134a6 0x549576 0x4bca8a 0x5134a6 0x549e0e 0x4bca8a 0x59c019 0x5
95ef6 0x5134a6
[>>] 161/161, 0.3 task/s, elapsed: 506s, ETA:      0s
Evaluating bbox...
Loading and preparing results...
DONE (t=0.01s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=1.82s).
Accumulating evaluation results...
DONE (t=0.26s).

```

Average Precision	(AP) @ [IoU=0.50:0.95 area= all maxDets=100]	= 0.269
Average Precision	(AP) @ [IoU=0.50 area= all maxDets=1000]	= 0.426
Average Precision	(AP) @ [IoU=0.75 area= all maxDets=1000]	= 0.314
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]	= 0.076
Average Precision	(AP) @ [IoU=0.50:0.95 area= large maxDets=1000]	= 0.292
Average Recall	(AR) @ [IoU=0.50:0.95 area= all maxDets=100]	= 0.546
Average Recall	(AR) @ [IoU=0.50:0.95 area= all maxDets=300]	= 0.546
Average Recall	(AR) @ [IoU=0.50:0.95 area= all maxDets=1000]	= 0.546
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]	= 0.103
Average Recall	(AR) @ [IoU=0.50:0.95 area= large maxDets=1000]	= 0.586

```

Evaluating segm...
/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)
Loading and preparing results...
DONE (t=0.21s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *segm*
DONE (t=2.21s).
Accumulating evaluation results...
/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)

```

DONE (t=0.26s).

```

Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.285
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=1000 ] = 0.415
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=1000 ] = 0.312
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 ] = 0.008
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.310
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.584
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=300 ] = 0.584
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=1000 ] = 0.584
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 ] = 0.044
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.626

OrderedDict([('bbox_mAP', 0.269), ('bbox_mAP_50', 0.426), ('bbox_mAP_75', 0.314), ('bbox_mAP_s', 0.0), ('bbox_mAP_m', 0.076), ('bbox_mAP_l', 0.292), ('bbox_mAP_copypaste', '0.269 0.426 0.314 0.000 0.076 0.292'), ('segm_mAP', 0.285), ('segm_mAP_50', 0.415), ('segm_mAP_75', 0.312), ('segm_mAP_s', 0.0), ('segm_mAP_m', 0.008), ('segm_mAP_l', 0.31), ('segm_mAP_copypaste', '0.285 0.415 0.312 0.000 0.008 0.310')])

INFO:torch.distributed.elastic.agent.server.api:[default] worker group successfully finished. Waiting 300 seconds for other agents to finish.
INFO:torch.distributed.elastic.agent.server.api:Local worker group finished (SUCCEEDED). Waiting 300 seconds for other agents to finish
/usr/local/lib/python3.7/dist-packages/torch/distributed/elastic/utils/store.py:71: FutureWarning: This is an experimental API and will be changed in future.
    "This is an experimental API and will be changed in future.", FutureWarning
INFO:torch.distributed.elastic.agent.server.api:Done waiting for other agents. Elapsed: 0.0004987716674804688 seconds
[{"name": "torchelastic.worker.status.SUCCEEDED", "source": "WORKER", "timestamp": 0, "metadata": {"run_id": "none", "global_rank": 0, "group_rank": 0, "worker_id": "2379", "role": "default", "hostname": "35f615b8aec0", "state": "SUCCEEDED", "total_run_time": 535, "rdzv_backend": "static", "raw_error": null, "metadata": "{$group_world_size": 1, "{$entry_point": "{$python3", "{$local_rank": [0], "{$role_rank": [0], "{$role_world_size": [1]}}, "agent_restarts": 0}}}
[{"name": "torchelastic.worker.status.SUCCEEDED", "source": "AGENT", "timestamp": 0, "metadata": {"run_id": "none", "global_rank": null, "group_rank": 0, "worker_id": null, "role": "default", "hostname": "35f615b8aec0", "state": "SUCCEEDED", "total_run_time": 535, "rdzv_backend": "static", "raw_error": null, "metadata": "{$group_world_size": 1, "{$entry_point": "{$python3}}", "agent_restarts": 0}]


```

7.5 学習済みconfig、checkpointを使った検証

新たに準備したデータ（静止画、動画）による検証(verification)

detectorの構築を行い、検証用の画像、動画ファイルを判別させ、出力結果を目視評価する。

In [19]:

```

# 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

# configの変更 to YOLACTモデルへ
# a yolact model achieving 29 mAP on MS COCO at 42.5 fps evaluated on a single GPU model
# yolact_r50_1x8_coco.py [Enhance]: Optimize augmentation pipeline to speed up training

# overwrite class numbers at model configuration file configs/yolact/yolact_r50_1x8_coco.py
!cp 'work_dirs/yolact_r50_1x8_coco_tomato/yolact_r50_1x8_coco_tomato.py' configs/yolact/yolact_r50_1x8_coco_tomato_20220625-ep2.py'

```

```
# download the latest trained checkpoints for inference and finetuning.
# !mkdir -p checkpoints
!cp 'work_dirs/yolact_r50_1x8_coco_tomato/epoch_2.pth' \
    'checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth'

# Setup a checkpoint file to load
config = 'configs/yolact/yolact_r50_1x8_coco_tomato_20220625-ep2.py'
checkpoint = 'checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.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()
```

load checkpoint from local path: checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth

Out[19]:

```
YOLACT(
  (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_st  
ats=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_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)  
)  
(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_st  
ats=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_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=(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)
```

```
ats=True)
    (conv2) : Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=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_st
ats=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(512, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (1): ConvModule(
            (conv): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
        )
        (2): 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(2048, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        )
        (4): ConvModule(
            (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        )
    )
)
init_cfg={'type': 'Xavier', 'layer': 'Conv2d', 'distribution': 'uniform'}
(bbox_head): YOLACTHead(
    (loss_cls): CrossEntropyLoss(avg_non_ignore=False)
    (loss_bbox): SmoothL1Loss()
    (relu): ReLU(inplace=True)
)

```

```
(head_convs): ModuleList(
    (0): ConvModule(
        (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (activate): ReLU(inplace=True)
    )
)
(conv_cls): Conv2d(256, 21, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(conv_reg): Conv2d(256, 12, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(conv_coeff): Conv2d(256, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
)
init_cfg={'type': 'Xavier', 'distribution': 'uniform', 'bias': 0, 'layer': 'Conv2d'}
(segm_head): YOLACTSegmHead(
    (loss_segm): CrossEntropyLoss(avg_non_ignore=False)
    (segm_conv): Conv2d(256, 6, kernel_size=(1, 1), stride=(1, 1))
)
init_cfg={'type': 'Xavier', 'distribution': 'uniform', 'override': {'name': 'segm_cv'}}
(mask_head): YOLACTProtonet(
    (protonet): Sequential(
        (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): ReLU(inplace=True)
        (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (3): ReLU(inplace=True)
        (4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (5): ReLU(inplace=True)
        (6): InterpolateModule()
        (7): ReLU(inplace=True)
        (8): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (9): ReLU(inplace=True)
        (10): Conv2d(256, 32, kernel_size=(1, 1), stride=(1, 1))
        (11): ReLU(inplace=True)
    )
)
init_cfg={'type': 'Xavier', 'distribution': 'uniform', 'override': {'name': 'protone'}}
```

In [23]:

```
# 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 [24]:

```
# 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名

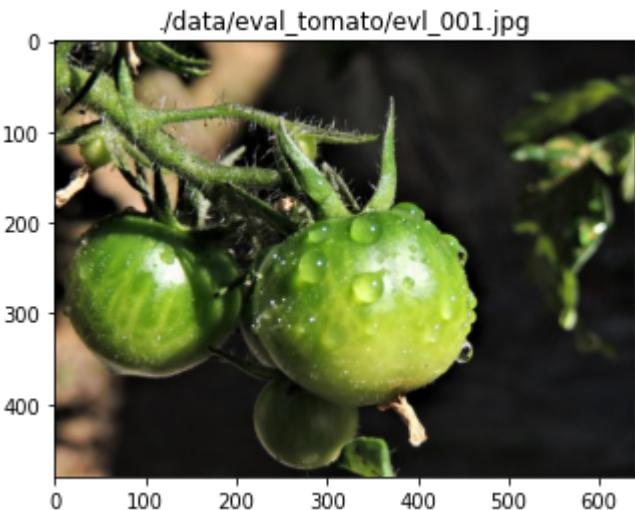
results = []

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_yolact.bin', 'wb') as p:
    pickle.dump(dst+'results_yolact.bin', p)

```

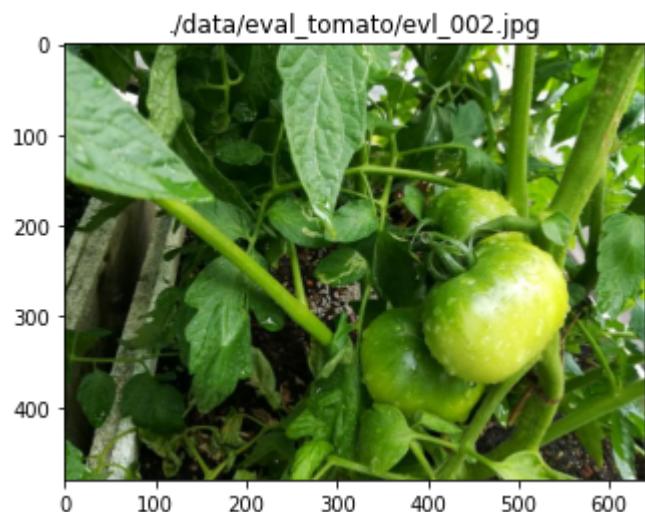


```

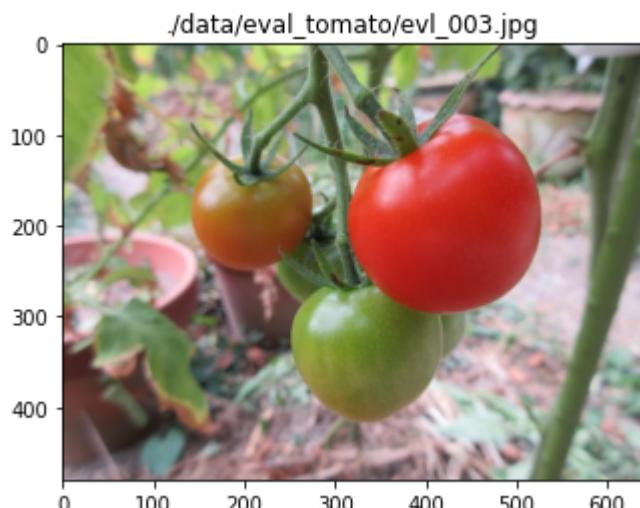
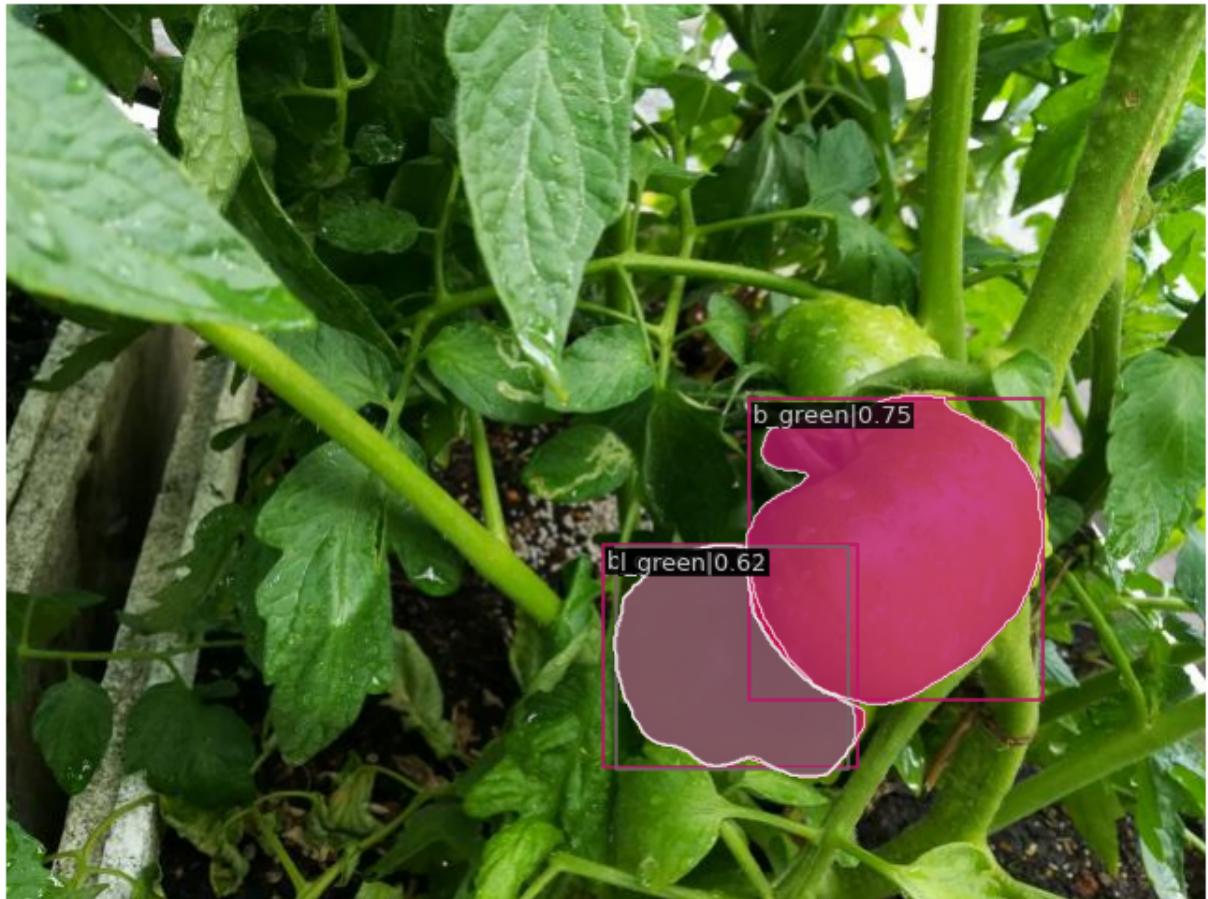
/content/mmdetection/mmdet/datasets/utils.py:70: UserWarning: "ImageToTensor" pipeline
is replaced by "DefaultFormatBundle" for batch inference. It is recommended to manually
replace it in the test data pipeline in your config file.
'data pipeline in your config file.', UserWarning)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:718: 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. (T
riggered internally at /pytorch/c10/core/TensorImpl.h:1156.)
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, ceil_mode)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Defau
lt upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.
0. Please specify align_corners=True if the old behavior is desired. See the documenta
tion of nn.Upsample for details.
    "See the documentation of nn.Upsample for details.".format(mode)

```

result



result



result



./data/eval_tomato/evl_004.jpg



result



./data/eval_tomato/evl_005.jpg



result



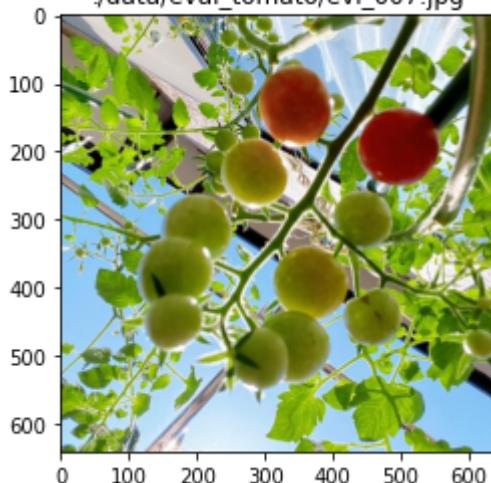
./data/eval_tomato/evl_006.jpg



result



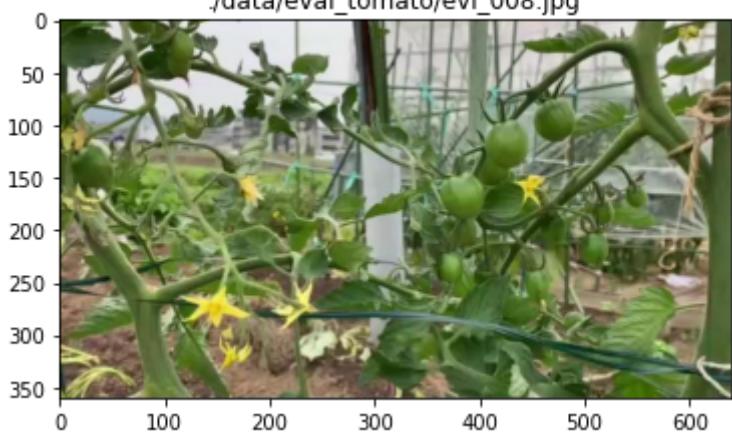
./data/eval_tomato/evl_007.jpg



result



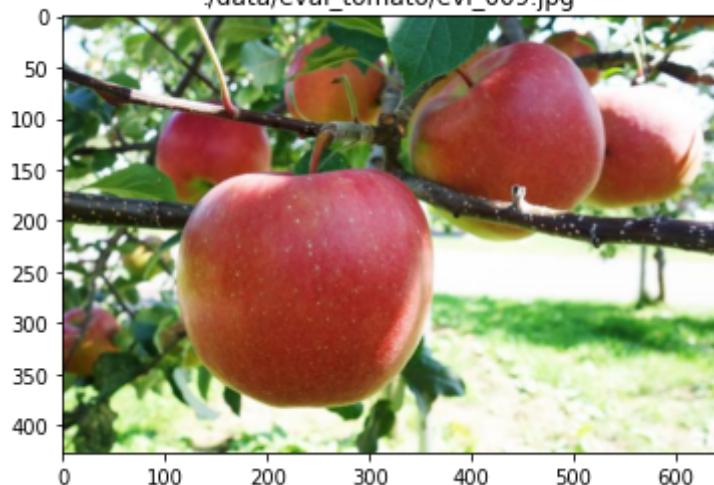
./data/eval_tomato/evl_008.jpg



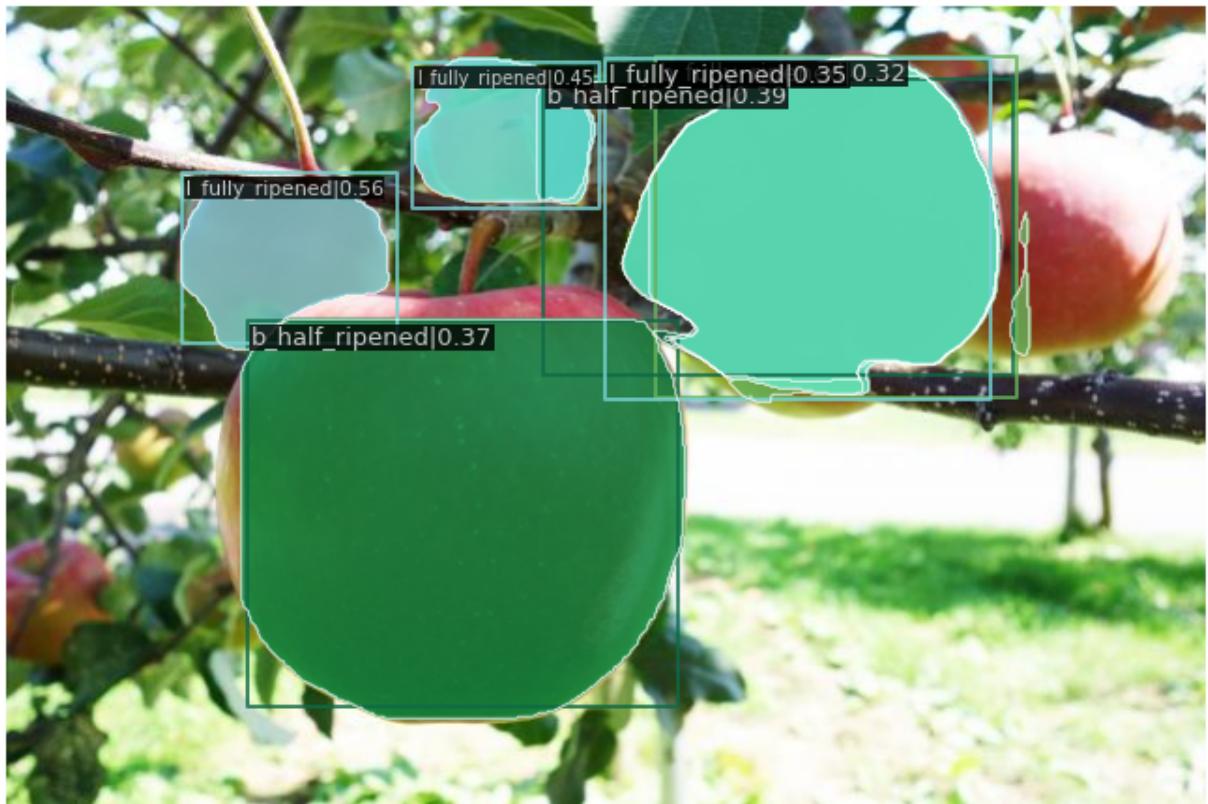
result



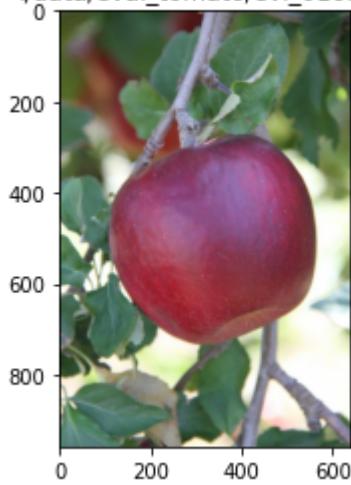
./data/eval_tomato/evl_009.jpg



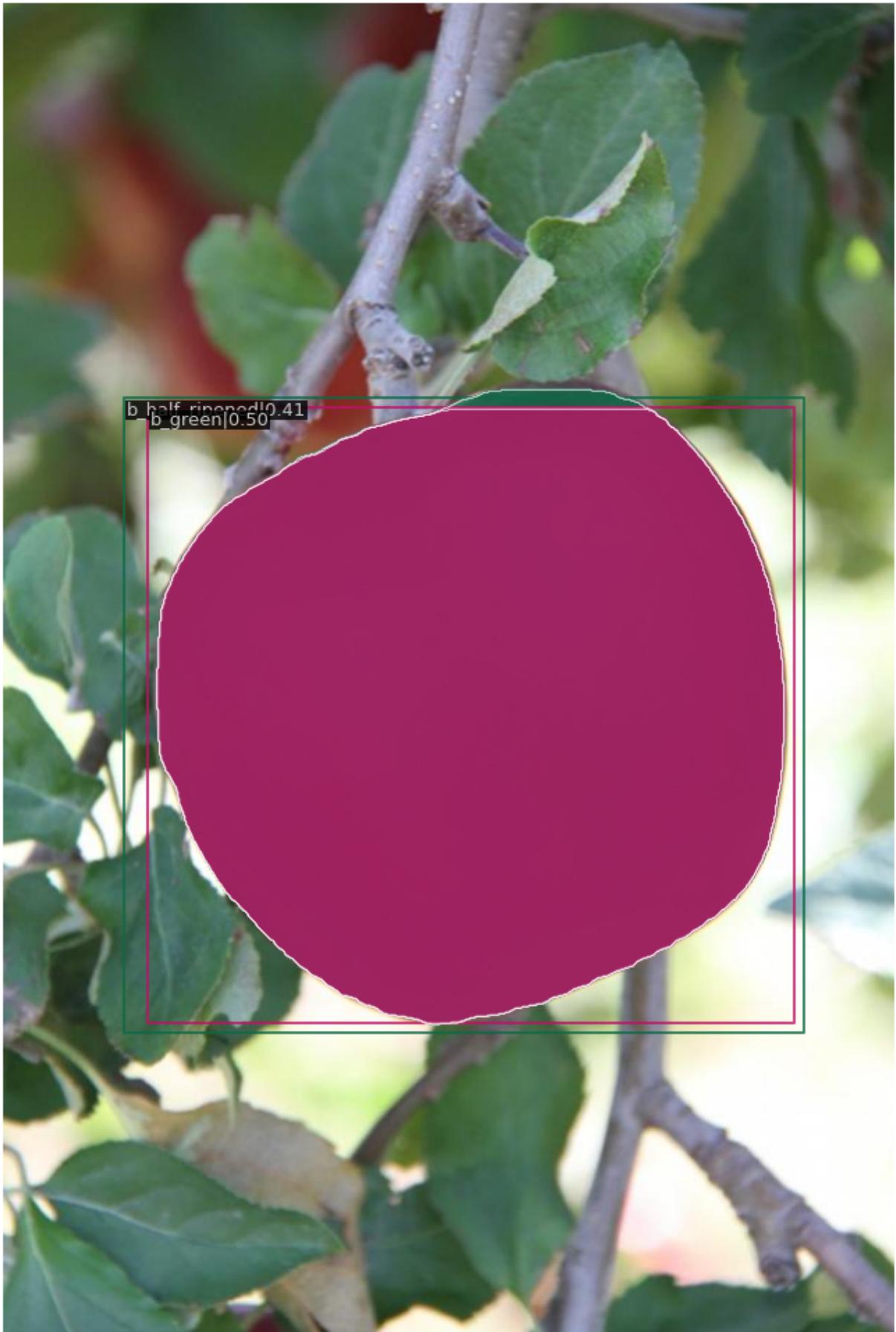
result



./data/eval_tomato/evl_010.jpg



result



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 [26]:

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

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

mov_name = 'tomato3.mp4'
out_file = dst+mov_name
!python "./demo/video_demo.py" "./data/video_tomato/tomato3.mp4" \
    'configs/yolact/yolact_r50_1x8_coco_tomato_20220625-ep2.py' \
    'checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth' \
    --out "./data/eval_result/tomato3_yolact.mp4"
```

```
load checkpoint from local path: checkpoints/yolact_r50_1x8_coco_tomato_20220625-ep2.pth
[...]
] 0/1752, elapsed: 0s, ETA:/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)
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:3613: UserWarning: Default upsampling behavior when mode=bilinear is changed to align_corners=False since 0.4.0. Please specify align_corners=True if the old behavior is desired. See the documentation of nn.Upsample for details.
    "See the documentation of nn.Upsample for details.".format(mode)
[>>] 1752/1752, 6.8 task/s, elapsed: 257s, ETA:      0s
```

In []: