

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

import torch
import os
import json
import matplotlib.pyplot as plt
from tensorboard.backend.event_processing import event_accumulator

# Try installing pre-built wheel
!pip install pyyaml==5.1
!pip install detectron2 -f
!pip install detectron2 -f \
https://dl.fbaipublicfiles.com/detectron2/wheels/cu113/torch1.10/index
.html

Collecting pyyaml==5.1
  Downloading PyYAML-5.1.tar.gz (274 kB)
    0.0/274.2 kB ? eta -:--:--
    274.2/274.2 kB 8.7 MB/s eta
0:00:00
error: subprocess-exited-with-error

  x python setup.py egg_info did not run successfully.
    | exit code: 1
    | See above for output.

  note: This error originates from a subprocess, and is likely not a
problem with pip.
Preparing metadata (setup.py) ... error: metadata-generation-failed

  x Encountered error while generating package metadata.
  | See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.

Usage:
  pip3 install [options] <requirement specifier> [package-index-
options] ...
  pip3 install [options] -r <requirements file> [package-index-
options] ...
  pip3 install [options] [-e] <vcs project url> ...
  pip3 install [options] [-e] <local project path> ...
  pip3 install [options] <archive url/path> ...

-f option requires 1 argument
Looking in links:
https://dl.fbaipublicfiles.com/detectron2/wheels/cu113/torch1.10/index
```

```
.html
Requirement already satisfied: detectron2 in
/usr/local/lib/python3.10/dist-packages (0.6)
Requirement already satisfied: Pillow>=7.1 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (11.0.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from detectron2) (3.8.0)
Requirement already satisfied: pycocotools>=2.0.2 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (2.0.8)
Requirement already satisfied: termcolor>=1.1 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (2.5.0)
Requirement already satisfied: yacs>=0.1.8 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (0.1.8)
Requirement already satisfied: tabulate in
/usr/local/lib/python3.10/dist-packages (from detectron2) (0.9.0)
Requirement already satisfied: cloudpickle in
/usr/local/lib/python3.10/dist-packages (from detectron2) (3.1.0)
Requirement already satisfied: tqdm>4.29.0 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (4.66.6)
Requirement already satisfied: tensorboard in
/usr/local/lib/python3.10/dist-packages (from detectron2) (2.17.1)
Requirement already satisfied: fvcore<0.1.6,>=0.1.5 in
/usr/local/lib/python3.10/dist-packages (from detectron2)
(0.1.5.post20221221)
Requirement already satisfied: iopath<0.1.10,>=0.1.7 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (0.1.9)
Requirement already satisfied: omegaconf<2.4,>=2.1 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (2.3.0)
Requirement already satisfied: hydra-core>=1.1 in
/usr/local/lib/python3.10/dist-packages (from detectron2) (1.3.2)
Requirement already satisfied: black in
/usr/local/lib/python3.10/dist-packages (from detectron2) (24.10.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from detectron2) (24.2)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from fvcore<0.1.6,>=0.1.5-
>detectron2) (1.26.4)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from fvcore<0.1.6,>=0.1.5-
>detectron2) (6.0.2)
Requirement already satisfied: antlr4-python3-runtime==4.9.* in
/usr/local/lib/python3.10/dist-packages (from hydra-core>=1.1-
>detectron2) (4.9.3)
Requirement already satisfied: portalocker in
/usr/local/lib/python3.10/dist-packages (from iopath<0.1.10,>=0.1.7-
>detectron2) (3.0.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(1.3.1)
```

```
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(1.4.7)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2)
(2.8.2)
Requirement already satisfied: click>=8.0.0 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(8.1.7)
Requirement already satisfied: mypy-extensions>=0.4.3 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(1.0.0)
Requirement already satisfied: pathspec>=0.9.0 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(0.12.1)
Requirement already satisfied: platformdirs>=2 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(4.3.6)
Requirement already satisfied: tomli>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(2.1.0)
Requirement already satisfied: typing-extensions>=4.0.1 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2)
(4.12.2)
Requirement already satisfied: absl-py>=0.4 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(1.4.0)
Requirement already satisfied: grpcio>=1.48.2 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(1.67.1)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(3.7)
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(4.25.5)
Requirement already satisfied: setuptools>=41.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(75.1.0)
Requirement already satisfied: six>1.9 in
```

```
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(1.16.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from tensorboard-
>detectron2) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2)
(3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard->detectron2) (3.0.2)

!pip install 'git+https://github.com/facebookresearch/detectron2.git'

Collecting git+https://github.com/facebookresearch/detectron2.git
  Cloning https://github.com/facebookresearch/detectron2.git to
/tmp/pip-req-build-zixznmmq
    Running command git clone --filter=blob:none --quiet
https://github.com/facebookresearch/detectron2.git /tmp/pip-req-build-
zixznmmq
      Resolved https://github.com/facebookresearch/detectron2.git to
commit 754469e176b224d17460612bdaa2cb8112b04cd9
      Preparing metadata (setup.py) ... ent already satisfied: Pillow>=7.1
in /usr/local/lib/python3.10/dist-packages (from detectron2==0.6)
(11.0.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (3.8.0)
Requirement already satisfied: pycocotools>=2.0.2 in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (2.0.8)
Requirement already satisfied: termcolor>=1.1 in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (2.5.0)
Collecting yacs>=0.1.8 (from detectron2==0.6)
  Downloading yacs-0.1.8-py3-none-any.whl.metadata (639 bytes)
Requirement already satisfied: tabulate in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (0.9.0)
Requirement already satisfied: cloudpickle in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (3.1.0)
Requirement already satisfied: tqdm>4.29.0 in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6)
(4.66.6)
Requirement already satisfied: tensorboard in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6)
(2.17.1)
Collecting fvcore<0.1.6,>=0.1.5 (from detectron2==0.6)
  Downloading fvcore-0.1.5.post20221221.tar.gz (50 kB)
0:00:00
      50.2/50.2 kB 3.1 MB/s eta
      etadata (setup.py) ... detectron2==0.6)
      Downloading iopath-0.1.9-py3-none-any.whl.metadata (370 bytes)
      Collecting omegaconf<2.4,>=2.1 (from detectron2==0.6)
```

```
  Downloading omegaconf-2.3.0-py3-none-any.whl.metadata (3.9 kB)
Collecting hydra-core>=1.1 (from detectron2==0.6)
    Downloading hydra_core-1.3.2-py3-none-any.whl.metadata (5.5 kB)
Collecting black (from detectron2==0.6)
    Downloading black-24.10.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.m
etadata (79 kB)
----- 79.2/79.2 kB 4.2 MB/s eta
0:00:00
ent already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (24.2)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from fvcore<0.1.6,>=0.1.5-
>detectron2==0.6) (1.26.4)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from fvcore<0.1.6,>=0.1.5-
>detectron2==0.6) (6.0.2)
Collecting antlr4-python3-runtime==4.9.* (from hydra-core>=1.1-
>detectron2==0.6)
    Downloading antlr4-python3-runtime-4.9.3.tar.gz (117 kB)
----- 117.0/117.0 kB 10.9 MB/s eta
0:00:00
etada (setup.py) ... iopath<0.1.10,>=0.1.7->detectron2==0.6)
    Downloading portalocker-3.0.0-py3-none-any.whl.metadata (8.5 kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (1.4.7)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib-
>detectron2==0.6) (2.8.2)
Requirement already satisfied: click>=8.0.0 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2==0.6)
(8.1.7)
Collecting mypy-extensions>=0.4.3 (from black->detectron2==0.6)
    Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Collecting pathspec>=0.9.0 (from black->detectron2==0.6)
    Downloading pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
```

```
Requirement already satisfied: platformdirs>=2 in  
/usr/local/lib/python3.10/dist-packages (from black->detectron2==0.6)  
(4.3.6)  
Requirement already satisfied: tomli>=1.1.0 in  
/usr/local/lib/python3.10/dist-packages (from black->detectron2==0.6)  
(2.1.0)  
Requirement already satisfied: typing-extensions>=4.0.1 in  
/usr/local/lib/python3.10/dist-packages (from black->detectron2==0.6)  
(4.12.2)  
Requirement already satisfied: absl-py>=0.4 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (1.4.0)  
Requirement already satisfied: grpcio>=1.48.2 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (1.67.1)  
Requirement already satisfied: markdown>=2.6.8 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (3.7)  
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (4.25.5)  
Requirement already satisfied: setuptools>=41.0.0 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (75.1.0)  
Requirement already satisfied: six>1.9 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (1.16.0)  
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (0.7.2)  
Requirement already satisfied: werkzeug>=1.0.1 in  
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6) (3.1.3)  
Requirement already satisfied: MarkupSafe>=2.1.1 in  
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard->detectron2==0.6) (3.0.2)  
Downloading hydra_core-1.3.2-py3-none-any.whl (154 kB) 154.5/154.5 kB 14.4 MB/s eta  
0:00:00  
egaconf-2.3.0-py3-none-any.whl (79 kB) 79.5/79.5 kB 7.4 MB/s eta  
0:00:00  
anylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl  
(1.8 MB) 1.8/1.8 MB 54.4 MB/s eta  
0:00:00  
ypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)  
Downloading pathspec-0.12.1-py3-none-any.whl (31 kB)  
Downloading portalocker-3.0.0-py3-none-any.whl (19 kB)
```

```
Building wheels for collected packages: detectron2, fvcore, antlr4-
python3-runtime
  Building wheel for detectron2 (setup.py) ... e=detectron2-0.6-cp310-
  cp310-linux_x86_64.whl size=6355064
  sha256=7757ee6e2ed1dd3c2bebf8ab830e44ec2963bdb7ea6f061cbbedf5de564fe4f
  1
    Stored in directory:
    /tmp/pip-ephem-wheel-cache-t5pft6qo/wheels/47/e5/15/94c80df2ba85500c5d
    76599cc307c0a7079d0e221bb6fc4375
      Building wheel for fvcore (setup.py) ... e=fvcore-
  0.1.5.post20221221-py3-none-any.whl size=61396
  sha256=bc6f562b2c4b757e9e6f99e079285b1e933cd8a26e99dd901f7fb1a610b5769
  a
    Stored in directory:
    /root/.cache/pip/wheels/01/c0/af/77c1cf53a1be9e42a52b48e5af2169d40ec2e
    89f7362489dd0
      Building wheel for antlr4-python3-runtime (setup.py) ... e:
      filename=antlr4_python3_runtime-4.9.3-py3-none-any.whl size=144555
  sha256=68466dceb25bcb0bbe57c5b1b1b89db098237127a1228931666e606728351a5
  3
    Stored in directory:
    /root/.cache/pip/wheels/12/93/dd/1f6a127edc45659556564c5730f6d4e300888
    f4bca2d4c5a88
Successfully built detectron2 fvcore antlr4-python3-runtime
Installing collected packages: antlr4-python3-runtime, yacs,
portalocker, pathspec, omegaconf, mypy-extensions, iopath, hydra-core,
black, fvcore, detectron2
Successfully installed antlr4-python3-runtime-4.9.3 black-24.10.0
detectron2-0.6 fvcore-0.1.5.post20221221 hydra-core-1.3.2 iopath-0.1.9
mypy-extensions-1.0.0 omegaconf-2.3.0 pathspec-0.12.1 portalocker-
3.0.0 yacs-0.1.8

{"id": "a57801347a904c12810f7d573d75dad9", "pip_warning": {"packages": [
  "pydevd_plugins"
]}

import detectron2
from detectron2 import model_zoo
from detectron2.engine import DefaultTrainer, DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog, DatasetCatalog
from detectron2.evaluation import PascalVOCDetectionEvaluator

import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()
print(f"Detectron2 version: {detectron2.__version__}")

Detectron2 version: 0.6
```

```

!wget http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCTrainval_11-
May-2012.tar
!tar -xf VOCTrainval_11-May-2012.tar
!mkdir -p datasets
# !mv VOCdevkit/VOC2012 datasets/

--2024-11-19 04:04:46--
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCTrainval_11-May-
2012.tar
Resolving host.robots.ox.ac.uk (host.robots.ox.ac.uk)... 129.67.94.152
Connecting to host.robots.ox.ac.uk (host.robots.ox.ac.uk)|129.67.94.152|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1999639040 (1.9G) [application/x-tar]
Saving to: 'VOCTrainval_11-May-2012.tar'

VOCTrainval_11-May- 100%[=====] 1.86G 29.1MB/s   in
66s

2024-11-19 04:05:53 (28.7 MB/s) - 'VOCTrainval_11-May-2012.tar' saved
[1999639040/1999639040]

mv: cannot stat 'VOC2012': No such file or directory
!mv VOCdevkit/VOC2012 datasets/

# Configure the Faster R-CNN model
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_r
cnn_R_50_FPN_3x.yaml"))
cfg.OUTPUT_DIR = 'MyVOCTraining'
cfg.DATASETS.TRAIN = ("voc_2012_train",)
cfg.DATASETS.TEST = ("voc_2012_val",)
cfg.DATA_LOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS =
model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_50_FPN_3x.y
aml")
cfg.SOLVER.IMS_PER_BATCH = 2 # Adjust based on GPU memory
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 3000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 20

# Create and run the trainer
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()

[11/19 04:07:13 d2.engine.defaults]: Model:
GeneralizedRCNN(

```

```
(backbone): FPN(
    (fpn_lateral2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1,
1))
    (fpn_output2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (fpn_lateral3): Conv2d(512, 256, kernel_size=(1, 1), stride=(1,
1))
    (fpn_output3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (fpn_lateral4): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1,
1))
    (fpn_output4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (fpn_lateral5): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1,
1))
    (fpn_output5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (top_block): LastLevelMaxPool()
    (bottom_up): ResNet(
        (stem): BasicStem(
            (conv1): Conv2d(
                3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False
                (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
            )
        )
        (res2): Sequential(
            (0): BottleneckBlock(
                (shortcut): Conv2d(
                    64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
                )
                (conv1): Conv2d(
                    64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
                )
                (conv2): Conv2d(
                    64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
                    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
                )
                (conv3): Conv2d(
                    64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
                )
            )
            (1): BottleneckBlock(
                (conv1): Conv2d(
                    256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
```

```
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv2): Conv2d(
        64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv3): Conv2d(
        64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
)
(2): BottleneckBlock(
    (conv1): Conv2d(
        256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv2): Conv2d(
        64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv3): Conv2d(
        64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
)
)
(res3): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv1): Conv2d(
            256, 128, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
        )
        (conv2): Conv2d(
            128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
            (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
        )
        (conv3): Conv2d(
            128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
    )
)
(1): BottleneckBlock()
```

```
(conv1): Conv2d(
    512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
)
(conv2): Conv2d(
    128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
    (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
)
(conv3): Conv2d(
    128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
)
)
(2): BottleneckBlock(
    (conv1): Conv2d(
        512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv2): Conv2d(
        128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv3): Conv2d(
        128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
    )
)
(3): BottleneckBlock(
    (conv1): Conv2d(
        512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv2): Conv2d(
        128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv3): Conv2d(
        128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
    )
)
)
(res4): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False
        )
        (conv1): Conv2d(
            1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv2): Conv2d(
            512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv3): Conv2d(
            512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
        )
    )
)
```

```
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
    (conv1): Conv2d(
        512, 256, kernel_size=(1, 1), stride=(2, 2), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(1): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(2): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(3): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
```

```
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(4): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(5): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
)
(res5): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)
        )
    )
)
```

```
(conv1): Conv2d(  
    1024, 512, kernel_size=(1, 1), stride=(2, 2), bias=False  
    (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
(conv2): Conv2d(  
    512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,  
1), bias=False  
    (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
(conv3): Conv2d(  
    512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False  
    (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)  
)  
)  
(1): BottleneckBlock(  
    (conv1): Conv2d(  
        2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
    (conv2): Conv2d(  
        512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,  
1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
    (conv3): Conv2d(  
        512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)  
)  
)  
(2): BottleneckBlock(  
    (conv1): Conv2d(  
        2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
    (conv2): Conv2d(  
        512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,  
1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)  
)  
    (conv3): Conv2d(  
        512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False  
        (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)  
)  
)  
)  
)  
(proposal_generator): RPN(  
    (rpn_head): StandardRPNHead(  
        )
```

```

(conv): Conv2d(
    256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)
    (activation): ReLU()
)
(objectness_logits): Conv2d(256, 3, kernel_size=(1, 1),
stride=(1, 1))
(anchor_deltas): Conv2d(256, 12, kernel_size=(1, 1), stride=(1,
1))
)
(anchor_generator): DefaultAnchorGenerator(
    (cell_anchors): BufferList()
)
)
(roi_heads): StandardROIHeads(
    (box_pooler): ROIPooler(
        (level_poolers): ModuleList(
            (0): ROIAlign(output_size=(7, 7), spatial_scale=0.25,
sampling_ratio=0, aligned=True)
            (1): ROIAlign(output_size=(7, 7), spatial_scale=0.125,
sampling_ratio=0, aligned=True)
            (2): ROIAlign(output_size=(7, 7), spatial_scale=0.0625,
sampling_ratio=0, aligned=True)
            (3): ROIAlign(output_size=(7, 7), spatial_scale=0.03125,
sampling_ratio=0, aligned=True)
        )
    )
    (box_head): FastRCNNConvFCHead(
        (flatten): Flatten(start_dim=1, end_dim=-1)
        (fc1): Linear(in_features=12544, out_features=1024, bias=True)
        (fc_relu1): ReLU()
        (fc2): Linear(in_features=1024, out_features=1024, bias=True)
        (fc_relu2): ReLU()
    )
    (box_predictor): FastRCNNOutputLayers(
        (cls_score): Linear(in_features=1024, out_features=21,
bias=True)
        (bbox_pred): Linear(in_features=1024, out_features=80,
bias=True)
    )
)
)
[11/19 04:07:14 d2.data.build]: Removed 0 images with no usable
annotations. 5717 images left.
[11/19 04:07:14 d2.data.build]: Distribution of instances among all 20
categories:
| category | #instances | category | #instances | category |
| #instances |           |           |           |           |
| :-----: | :-----: | :-----: | :-----: | :-----: |
| :-----: |           |           |           |           |

```

aeroplane	470	bicycle	410	bird
592				
boat	508	bottle	749	bus
317				
car	1191	cat	609	chair
1457				
cow	355	diningtable	373	dog
768				
horse	377	motorbike	375	person
5019				
pottedplant	557	sheep	509	sofa
399				
train	327	tvmonitor	412	
total	15774			

[11/19 04:07:14 d2.data.dataset_mapper]: [DatasetMapper] Augmentations used in training: [ResizeShortestEdge(short_edge_length=(640, 672, 704, 736, 768, 800), max_size=1333, sample_style='choice'), RandomFlip()]

[11/19 04:07:14 d2.data.build]: Using training sampler TrainingSampler

[11/19 04:07:14 d2.data.common]: Serializing the dataset using: <class 'detectron2.data.common._TorchSerializedList'>

[11/19 04:07:14 d2.data.common]: Serializing 5717 elements to byte tensors and concatenating them all ...

[11/19 04:07:14 d2.data.common]: Serialized dataset takes 2.63 MiB

[11/19 04:07:14 d2.data.build]: Making batched data loader with batch_size=2

WARNING [11/19 04:07:14 d2.solver.build]: SOLVER.STEPS contains values larger than SOLVER.MAX_ITER. These values will be ignored.

[11/19 04:07:14 d2.checkpoint.detection_checkpoint]:

[DetectionCheckpointer] Loading from https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_R_50_FPN_3x/137849458/model_final_280758.pkl ...

model_final_280758.pkl: 167MB [00:00, 189MB/s]

WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.cls_score.weight' to the model due to incompatible shapes: (81, 1024) in the checkpoint but (21, 1024) in the model! You might want to double check if this is expected.

WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.cls_score.bias' to the model due to incompatible shapes: (81,) in the checkpoint but (21,) in the model! You might want to double check if this is expected.

WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.bbox_pred.weight' to the model due to incompatible shapes: (320, 1024) in the checkpoint but (80, 1024) in the model! You might want to double check if this is expected.

WARNING:fvcore.common.checkpoint:Skip loading parameter

```
'roi_heads.box_predictor.bbox_pred.bias' to the model due to
incompatible shapes: (320,) in the checkpoint but (80,) in the model!
You might want to double check if this is expected.
WARNING:fvcore.common.checkpoint:Some model parameters or buffers are
not found in the checkpoint:
roi_heads.box_predictor.bbox_pred.{bias, weight}
roi_heads.box_predictor.cls_score.{bias, weight}

[11/19 04:07:16 d2.engine.train_loop]: Starting training from
iteration 0

/usr/local/lib/python3.10/dist-packages/torch/functional.py:534:
UserWarning: torch.meshgrid: in an upcoming release, it will be
required to pass the indexing argument. (Triggered internally at
./aten/src/ATen/native/TensorShape.cpp:3595.)
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]

[11/19 04:07:28 d2.utils.events]: eta: 0:18:08 iter: 19 total_loss:
3.915 loss_cls: 3.104 loss_box_reg: 0.7283 loss_rpn_cls: 0.01099
loss_rpn_loc: 0.01875 time: 0.3700 last_time: 0.3378 data_time:
0.0204 last_data_time: 0.0050 lr: 4.9953e-06 max_mem: 2427M
[11/19 04:07:42 d2.utils.events]: eta: 0:19:17 iter: 39 total_loss:
3.744 loss_cls: 2.983 loss_box_reg: 0.6849 loss_rpn_cls: 0.01369
loss_rpn_loc: 0.01333 time: 0.3832 last_time: 0.3601 data_time:
0.0089 last_data_time: 0.0056 lr: 9.9902e-06 max_mem: 2548M
[11/19 04:07:50 d2.utils.events]: eta: 0:19:01 iter: 59 total_loss:
3.46 loss_cls: 2.744 loss_box_reg: 0.6629 loss_rpn_cls: 0.007154
loss_rpn_loc: 0.01025 time: 0.3836 last_time: 0.3188 data_time:
0.0119 last_data_time: 0.0079 lr: 1.4985e-05 max_mem: 2638M
[11/19 04:07:58 d2.utils.events]: eta: 0:18:53 iter: 79 total_loss:
3.158 loss_cls: 2.442 loss_box_reg: 0.7199 loss_rpn_cls: 0.01121
loss_rpn_loc: 0.01641 time: 0.3832 last_time: 0.4234 data_time:
0.0083 last_data_time: 0.0058 lr: 1.998e-05 max_mem: 2638M
[11/19 04:08:05 d2.utils.events]: eta: 0:18:45 iter: 99 total_loss:
2.846 loss_cls: 2.005 loss_box_reg: 0.7985 loss_rpn_cls: 0.01169
loss_rpn_loc: 0.01572 time: 0.3831 last_time: 0.2876 data_time:
0.0164 last_data_time: 0.0179 lr: 2.4975e-05 max_mem: 2638M
[11/19 04:08:13 d2.utils.events]: eta: 0:18:50 iter: 119
total_loss: 2.408 loss_cls: 1.594 loss_box_reg: 0.7791
loss_rpn_cls: 0.004581 loss_rpn_loc: 0.01689 time: 0.3853
last_time: 0.4711 data_time: 0.0082 last_data_time: 0.0123 lr:
2.997e-05 max_mem: 2638M
[11/19 04:08:21 d2.utils.events]: eta: 0:18:17 iter: 139
total_loss: 1.685 loss_cls: 0.9858 loss_box_reg: 0.6523
loss_rpn_cls: 0.005631 loss_rpn_loc: 0.01528 time: 0.3845
last_time: 0.3530 data_time: 0.0138 last_data_time: 0.0147 lr:
3.4965e-05 max_mem: 2638M
[11/19 04:08:28 d2.utils.events]: eta: 0:18:03 iter: 159
total_loss: 1.57 loss_cls: 0.8541 loss_box_reg: 0.6545
loss_rpn_cls: 0.004067 loss_rpn_loc: 0.01338 time: 0.3828
```

```
last_time: 0.3591 data_time: 0.0067 last_data_time: 0.0154 lr:  
3.996e-05 max_mem: 2638M  
[11/19 04:08:36 d2.utils.events]: eta: 0:17:49 iter: 179  
total_loss: 1.638 loss_cls: 0.858 loss_box_reg: 0.7549  
loss_rpn_cls: 0.01185 loss_rpn_loc: 0.01491 time: 0.3833  
last_time: 0.4757 data_time: 0.0138 last_data_time: 0.0053 lr:  
4.4955e-05 max_mem: 2638M  
[11/19 04:08:44 d2.utils.events]: eta: 0:17:36 iter: 199  
total_loss: 1.511 loss_cls: 0.8403 loss_box_reg: 0.7114  
loss_rpn_cls: 0.00989 loss_rpn_loc: 0.01542 time: 0.3832  
last_time: 0.4229 data_time: 0.0121 last_data_time: 0.0151 lr:  
4.995e-05 max_mem: 2638M  
[11/19 04:08:51 d2.utils.events]: eta: 0:17:29 iter: 219  
total_loss: 1.634 loss_cls: 0.8499 loss_box_reg: 0.788  
loss_rpn_cls: 0.006041 loss_rpn_loc: 0.01417 time: 0.3840  
last_time: 0.3513 data_time: 0.0107 last_data_time: 0.0066 lr:  
5.4945e-05 max_mem: 2638M  
[11/19 04:08:59 d2.utils.events]: eta: 0:17:24 iter: 239  
total_loss: 1.514 loss_cls: 0.7315 loss_box_reg: 0.7264  
loss_rpn_cls: 0.006852 loss_rpn_loc: 0.02299 time: 0.3846  
last_time: 0.3245 data_time: 0.0115 last_data_time: 0.0173 lr:  
5.994e-05 max_mem: 2638M  
[11/19 04:09:07 d2.utils.events]: eta: 0:17:22 iter: 259  
total_loss: 1.628 loss_cls: 0.8039 loss_box_reg: 0.7391  
loss_rpn_cls: 0.004308 loss_rpn_loc: 0.0106 time: 0.3850  
last_time: 0.4273 data_time: 0.0094 last_data_time: 0.0055 lr:  
6.4935e-05 max_mem: 2638M  
[11/19 04:09:15 d2.utils.events]: eta: 0:17:17 iter: 279  
total_loss: 1.529 loss_cls: 0.759 loss_box_reg: 0.7388  
loss_rpn_cls: 0.005784 loss_rpn_loc: 0.01916 time: 0.3868  
last_time: 0.4526 data_time: 0.0105 last_data_time: 0.0050 lr:  
6.993e-05 max_mem: 2638M  
[11/19 04:09:23 d2.utils.events]: eta: 0:17:10 iter: 299  
total_loss: 1.526 loss_cls: 0.7587 loss_box_reg: 0.7753  
loss_rpn_cls: 0.01137 loss_rpn_loc: 0.01507 time: 0.3871  
last_time: 0.4608 data_time: 0.0089 last_data_time: 0.0076 lr:  
7.4925e-05 max_mem: 2638M  
[11/19 04:09:31 d2.utils.events]: eta: 0:17:10 iter: 319  
total_loss: 1.347 loss_cls: 0.6764 loss_box_reg: 0.6746  
loss_rpn_cls: 0.00568 loss_rpn_loc: 0.01027 time: 0.3890  
last_time: 0.4239 data_time: 0.0122 last_data_time: 0.0056 lr:  
7.992e-05 max_mem: 2638M  
[11/19 04:09:40 d2.utils.events]: eta: 0:17:12 iter: 339  
total_loss: 1.395 loss_cls: 0.6931 loss_box_reg: 0.7036  
loss_rpn_cls: 0.00784 loss_rpn_loc: 0.01606 time: 0.3898  
last_time: 0.4809 data_time: 0.0114 last_data_time: 0.0151 lr:  
8.4915e-05 max_mem: 2638M  
[11/19 04:09:48 d2.utils.events]: eta: 0:17:11 iter: 359  
total_loss: 1.495 loss_cls: 0.7511 loss_box_reg: 0.7112
```

```
loss_rpn_cls: 0.00683 loss_rpn_loc: 0.0161 time: 0.3906
last_time: 0.3631 data_time: 0.0087 last_data_time: 0.0039 lr:
8.991e-05 max_mem: 2638M
[11/19 04:09:56 d2.utils.events]: eta: 0:17:08 iter: 379
total_loss: 1.586 loss_cls: 0.7348 loss_box_reg: 0.7507
loss_rpn_cls: 0.006725 loss_rpn_loc: 0.02251 time: 0.3920
last_time: 0.4427 data_time: 0.0117 last_data_time: 0.0177 lr:
9.4905e-05 max_mem: 2638M
[11/19 04:10:04 d2.utils.events]: eta: 0:17:03 iter: 399
total_loss: 1.487 loss_cls: 0.7227 loss_box_reg: 0.7981
loss_rpn_cls: 0.004415 loss_rpn_loc: 0.01458 time: 0.3927
last_time: 0.4957 data_time: 0.0086 last_data_time: 0.0153 lr:
9.99e-05 max_mem: 2638M
[11/19 04:10:12 d2.utils.events]: eta: 0:16:58 iter: 419
total_loss: 1.466 loss_cls: 0.7088 loss_box_reg: 0.766
loss_rpn_cls: 0.006929 loss_rpn_loc: 0.01447 time: 0.3933
last_time: 0.4388 data_time: 0.0118 last_data_time: 0.0175 lr:
0.0001049 max_mem: 2638M
[11/19 04:10:21 d2.utils.events]: eta: 0:16:54 iter: 439
total_loss: 1.416 loss_cls: 0.6612 loss_box_reg: 0.7186
loss_rpn_cls: 0.004691 loss_rpn_loc: 0.01994 time: 0.3944
last_time: 0.3901 data_time: 0.0112 last_data_time: 0.0184 lr:
0.00010989 max_mem: 2638M
[11/19 04:10:29 d2.utils.events]: eta: 0:16:48 iter: 459
total_loss: 1.493 loss_cls: 0.7136 loss_box_reg: 0.6691
loss_rpn_cls: 0.01519 loss_rpn_loc: 0.02667 time: 0.3959
last_time: 0.3826 data_time: 0.0106 last_data_time: 0.0051 lr:
0.00011489 max_mem: 2638M
[11/19 04:10:38 d2.utils.events]: eta: 0:16:49 iter: 479
total_loss: 1.311 loss_cls: 0.6116 loss_box_reg: 0.7031
loss_rpn_cls: 0.004468 loss_rpn_loc: 0.01585 time: 0.3976
last_time: 0.4062 data_time: 0.0113 last_data_time: 0.0230 lr:
0.00011988 max_mem: 2638M
[11/19 04:10:46 d2.utils.events]: eta: 0:16:41 iter: 499
total_loss: 1.515 loss_cls: 0.702 loss_box_reg: 0.7347
loss_rpn_cls: 0.009015 loss_rpn_loc: 0.0118 time: 0.3978
last_time: 0.4738 data_time: 0.0091 last_data_time: 0.0054 lr:
0.00012488 max_mem: 2638M
[11/19 04:10:54 d2.utils.events]: eta: 0:16:34 iter: 519
total_loss: 1.539 loss_cls: 0.651 loss_box_reg: 0.7533
loss_rpn_cls: 0.007448 loss_rpn_loc: 0.01878 time: 0.3982
last_time: 0.3508 data_time: 0.0152 last_data_time: 0.0140 lr:
0.00012987 max_mem: 2638M
[11/19 04:11:02 d2.utils.events]: eta: 0:16:21 iter: 539
total_loss: 1.154 loss_cls: 0.5123 loss_box_reg: 0.6287
loss_rpn_cls: 0.005461 loss_rpn_loc: 0.01643 time: 0.3979
last_time: 0.4349 data_time: 0.0086 last_data_time: 0.0166 lr:
0.00013487 max_mem: 2638M
[11/19 04:11:10 d2.utils.events]: eta: 0:16:15 iter: 559
```

```
total_loss: 1.282 loss_cls: 0.5875 loss_box_reg: 0.6975
loss_rpn_cls: 0.005269 loss_rpn_loc: 0.01601 time: 0.3985
last_time: 0.4354 data_time: 0.0130 last_data_time: 0.0054 lr:
0.00013986 max_mem: 2638M
[11/19 04:11:19 d2.utils.events]: eta: 0:16:12 iter: 579
total_loss: 1.291 loss_cls: 0.5306 loss_box_reg: 0.673
loss_rpn_cls: 0.004977 loss_rpn_loc: 0.01472 time: 0.3994
last_time: 0.5001 data_time: 0.0112 last_data_time: 0.0301 lr:
0.00014486 max_mem: 2638M
[11/19 04:11:27 d2.utils.events]: eta: 0:16:01 iter: 599
total_loss: 1.262 loss_cls: 0.5789 loss_box_reg: 0.6922
loss_rpn_cls: 0.01322 loss_rpn_loc: 0.01589 time: 0.3995
last_time: 0.3475 data_time: 0.0104 last_data_time: 0.0012 lr:
0.00014985 max_mem: 2638M
[11/19 04:11:35 d2.utils.events]: eta: 0:15:55 iter: 619
total_loss: 1.34 loss_cls: 0.5485 loss_box_reg: 0.7366
loss_rpn_cls: 0.002919 loss_rpn_loc: 0.01536 time: 0.3995
last_time: 0.4689 data_time: 0.0112 last_data_time: 0.0141 lr:
0.00015485 max_mem: 2638M
[11/19 04:11:43 d2.utils.events]: eta: 0:15:49 iter: 639
total_loss: 1.461 loss_cls: 0.592 loss_box_reg: 0.7776
loss_rpn_cls: 0.005215 loss_rpn_loc: 0.01152 time: 0.3999
last_time: 0.4347 data_time: 0.0072 last_data_time: 0.0055 lr:
0.00015984 max_mem: 2638M
[11/19 04:11:51 d2.utils.events]: eta: 0:15:40 iter: 659
total_loss: 1.163 loss_cls: 0.5158 loss_box_reg: 0.6794
loss_rpn_cls: 0.006015 loss_rpn_loc: 0.01014 time: 0.4001
last_time: 0.4382 data_time: 0.0128 last_data_time: 0.0132 lr:
0.00016484 max_mem: 2638M
[11/19 04:11:59 d2.utils.events]: eta: 0:15:31 iter: 679
total_loss: 1.216 loss_cls: 0.5131 loss_box_reg: 0.6741
loss_rpn_cls: 0.004549 loss_rpn_loc: 0.01591 time: 0.3999
last_time: 0.3471 data_time: 0.0082 last_data_time: 0.0056 lr:
0.00016983 max_mem: 2638M
[11/19 04:12:07 d2.utils.events]: eta: 0:15:21 iter: 699
total_loss: 1.114 loss_cls: 0.4891 loss_box_reg: 0.6032
loss_rpn_cls: 0.003514 loss_rpn_loc: 0.01964 time: 0.3997
last_time: 0.3415 data_time: 0.0120 last_data_time: 0.0035 lr:
0.00017483 max_mem: 2638M
[11/19 04:12:15 d2.utils.events]: eta: 0:15:13 iter: 719
total_loss: 1.223 loss_cls: 0.4862 loss_box_reg: 0.6589
loss_rpn_cls: 0.004256 loss_rpn_loc: 0.01639 time: 0.3997
last_time: 0.4147 data_time: 0.0112 last_data_time: 0.0076 lr:
0.00017982 max_mem: 2638M
[11/19 04:12:23 d2.utils.events]: eta: 0:15:05 iter: 739
total_loss: 1.043 loss_cls: 0.3811 loss_box_reg: 0.5679
loss_rpn_cls: 0.01162 loss_rpn_loc: 0.01831 time: 0.3999
last_time: 0.3456 data_time: 0.0103 last_data_time: 0.0055 lr:
0.00018482 max_mem: 2638M
```

```
[11/19 04:12:31 d2.utils.events]: eta: 0:14:59 iter: 759
total_loss: 0.9488 loss_cls: 0.3621 loss_box_reg: 0.5865
loss_rpn_cls: 0.00233 loss_rpn_loc: 0.01014 time: 0.4002
last_time: 0.3926 data_time: 0.0118 last_data_time: 0.0333 lr:
0.00018981 max_mem: 2638M
[11/19 04:12:40 d2.utils.events]: eta: 0:14:50 iter: 779
total_loss: 1.321 loss_cls: 0.4697 loss_box_reg: 0.7606
loss_rpn_cls: 0.01011 loss_rpn_loc: 0.01509 time: 0.4005
last_time: 0.3739 data_time: 0.0083 last_data_time: 0.0138 lr:
0.00019481 max_mem: 2638M
[11/19 04:12:48 d2.utils.events]: eta: 0:14:41 iter: 799
total_loss: 1.13 loss_cls: 0.4399 loss_box_reg: 0.674 loss_rpn_cls:
0.01067 loss_rpn_loc: 0.01464 time: 0.4004 last_time: 0.3768
data_time: 0.0129 last_data_time: 0.0121 lr: 0.0001998 max_mem:
2638M
[11/19 04:12:55 d2.utils.events]: eta: 0:14:30 iter: 819
total_loss: 0.7711 loss_cls: 0.2777 loss_box_reg: 0.5627
loss_rpn_cls: 0.004999 loss_rpn_loc: 0.01823 time: 0.4001
last_time: 0.3689 data_time: 0.0073 last_data_time: 0.0056 lr:
0.0002048 max_mem: 2638M
[11/19 04:13:03 d2.utils.events]: eta: 0:14:22 iter: 839
total_loss: 1.011 loss_cls: 0.3393 loss_box_reg: 0.5926
loss_rpn_cls: 0.01147 loss_rpn_loc: 0.0182 time: 0.4000
last_time: 0.3201 data_time: 0.0134 last_data_time: 0.0079 lr:
0.00020979 max_mem: 2638M
[11/19 04:13:11 d2.utils.events]: eta: 0:14:14 iter: 859
total_loss: 0.8951 loss_cls: 0.3003 loss_box_reg: 0.5407
loss_rpn_cls: 0.01333 loss_rpn_loc: 0.01206 time: 0.3999
last_time: 0.4829 data_time: 0.0066 last_data_time: 0.0150 lr:
0.00021479 max_mem: 2638M
[11/19 04:13:19 d2.utils.events]: eta: 0:14:06 iter: 879
total_loss: 0.857 loss_cls: 0.3393 loss_box_reg: 0.513
loss_rpn_cls: 0.005167 loss_rpn_loc: 0.02222 time: 0.3997
last_time: 0.4483 data_time: 0.0134 last_data_time: 0.0266 lr:
0.00021978 max_mem: 2638M
[11/19 04:13:27 d2.utils.events]: eta: 0:13:58 iter: 899
total_loss: 0.7483 loss_cls: 0.2456 loss_box_reg: 0.4785
loss_rpn_cls: 0.003474 loss_rpn_loc: 0.01402 time: 0.3998
last_time: 0.4904 data_time: 0.0113 last_data_time: 0.0164 lr:
0.00022478 max_mem: 2638M
[11/19 04:13:35 d2.utils.events]: eta: 0:13:49 iter: 919
total_loss: 0.966 loss_cls: 0.3262 loss_box_reg: 0.6146
loss_rpn_cls: 0.01128 loss_rpn_loc: 0.02288 time: 0.3997
last_time: 0.3865 data_time: 0.0108 last_data_time: 0.0015 lr:
0.00022977 max_mem: 2638M
[11/19 04:13:43 d2.utils.events]: eta: 0:13:41 iter: 939
total_loss: 0.8197 loss_cls: 0.2833 loss_box_reg: 0.5218
loss_rpn_cls: 0.01025 loss_rpn_loc: 0.02502 time: 0.4001
last_time: 0.4286 data_time: 0.0103 last_data_time: 0.0058 lr:
```

```
0.00023477 max_mem: 2638M
[11/19 04:13:51 d2.utils.events]: eta: 0:13:33 iter: 959
total_loss: 0.9252 loss_cls: 0.3159 loss_box_reg: 0.6294
loss_rpn_cls: 0.008346 loss_rpn_loc: 0.02647 time: 0.4000
last_time: 0.4682 data_time: 0.0078 last_data_time: 0.0058 lr:
0.00023976 max_mem: 2638M
[11/19 04:14:00 d2.utils.events]: eta: 0:13:27 iter: 979
total_loss: 0.7675 loss_cls: 0.2348 loss_box_reg: 0.4989
loss_rpn_cls: 0.005452 loss_rpn_loc: 0.015 time: 0.4004
last_time: 0.4282 data_time: 0.0124 last_data_time: 0.0054 lr:
0.00024476 max_mem: 2638M
[11/19 04:14:07 d2.utils.events]: eta: 0:13:17 iter: 999
total_loss: 0.7854 loss_cls: 0.2206 loss_box_reg: 0.4944
loss_rpn_cls: 0.005422 loss_rpn_loc: 0.01257 time: 0.4000
last_time: 0.3927 data_time: 0.0103 last_data_time: 0.0164 lr:
0.00024975 max_mem: 2638M
[11/19 04:14:15 d2.utils.events]: eta: 0:13:09 iter: 1019
total_loss: 0.726 loss_cls: 0.1985 loss_box_reg: 0.4796
loss_rpn_cls: 0.006603 loss_rpn_loc: 0.01925 time: 0.3999
last_time: 0.3825 data_time: 0.0084 last_data_time: 0.0050 lr:
0.00025 max_mem: 2638M
[11/19 04:14:24 d2.utils.events]: eta: 0:13:04 iter: 1039
total_loss: 0.7627 loss_cls: 0.2053 loss_box_reg: 0.4398
loss_rpn_cls: 0.01127 loss_rpn_loc: 0.02885 time: 0.4003
last_time: 0.4421 data_time: 0.0101 last_data_time: 0.0150 lr:
0.00025 max_mem: 2638M
[11/19 04:14:32 d2.utils.events]: eta: 0:12:55 iter: 1059
total_loss: 0.7594 loss_cls: 0.2133 loss_box_reg: 0.4879
loss_rpn_cls: 0.01148 loss_rpn_loc: 0.01481 time: 0.4001
last_time: 0.4333 data_time: 0.0081 last_data_time: 0.0033 lr:
0.00025 max_mem: 2638M
[11/19 04:14:40 d2.utils.events]: eta: 0:12:48 iter: 1079
total_loss: 0.9122 loss_cls: 0.3552 loss_box_reg: 0.4972
loss_rpn_cls: 0.0108 loss_rpn_loc: 0.01757 time: 0.4003
last_time: 0.3755 data_time: 0.0129 last_data_time: 0.0056 lr:
0.00025 max_mem: 2638M
[11/19 04:14:48 d2.utils.events]: eta: 0:12:43 iter: 1099
total_loss: 0.8029 loss_cls: 0.2397 loss_box_reg: 0.5512
loss_rpn_cls: 0.006707 loss_rpn_loc: 0.01541 time: 0.4004
last_time: 0.3523 data_time: 0.0080 last_data_time: 0.0164 lr:
0.00025 max_mem: 2638M
[11/19 04:14:56 d2.utils.events]: eta: 0:12:35 iter: 1119
total_loss: 0.8179 loss_cls: 0.2437 loss_box_reg: 0.4979
loss_rpn_cls: 0.02406 loss_rpn_loc: 0.01436 time: 0.4007
last_time: 0.4386 data_time: 0.0115 last_data_time: 0.0039 lr:
0.00025 max_mem: 2638M
[11/19 04:15:04 d2.utils.events]: eta: 0:12:28 iter: 1139
total_loss: 0.7536 loss_cls: 0.2326 loss_box_reg: 0.4836
loss_rpn_cls: 0.01191 loss_rpn_loc: 0.01909 time: 0.4009
```

```
last_time: 0.3976 data_time: 0.0114 last_data_time: 0.0050 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:12 d2.utils.events]: eta: 0:12:20 iter: 1159  
total_loss: 0.8612 loss_cls: 0.2558 loss_box_reg: 0.495  
loss_rpn_cls: 0.007754 loss_rpn_loc: 0.01339 time: 0.4007  
last_time: 0.4752 data_time: 0.0090 last_data_time: 0.0136 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:21 d2.utils.events]: eta: 0:12:13 iter: 1179  
total_loss: 0.6889 loss_cls: 0.2191 loss_box_reg: 0.4123  
loss_rpn_cls: 0.003365 loss_rpn_loc: 0.01886 time: 0.4011  
last_time: 0.4545 data_time: 0.0158 last_data_time: 0.0209 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:29 d2.utils.events]: eta: 0:12:06 iter: 1199  
total_loss: 0.6534 loss_cls: 0.1596 loss_box_reg: 0.4244  
loss_rpn_cls: 0.007716 loss_rpn_loc: 0.01621 time: 0.4012  
last_time: 0.4751 data_time: 0.0099 last_data_time: 0.0167 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:38 d2.utils.events]: eta: 0:12:00 iter: 1219  
total_loss: 0.7214 loss_cls: 0.202 loss_box_reg: 0.447  
loss_rpn_cls: 0.005793 loss_rpn_loc: 0.01756 time: 0.4016  
last_time: 0.3722 data_time: 0.0109 last_data_time: 0.0164 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:46 d2.utils.events]: eta: 0:11:52 iter: 1239  
total_loss: 0.8118 loss_cls: 0.2797 loss_box_reg: 0.4655  
loss_rpn_cls: 0.004146 loss_rpn_loc: 0.01434 time: 0.4017  
last_time: 0.4494 data_time: 0.0084 last_data_time: 0.0191 lr:  
0.00025 max_mem: 2638M  
[11/19 04:15:54 d2.utils.events]: eta: 0:11:47 iter: 1259  
total_loss: 0.6611 loss_cls: 0.2048 loss_box_reg: 0.3745  
loss_rpn_cls: 0.003764 loss_rpn_loc: 0.02483 time: 0.4019  
last_time: 0.4761 data_time: 0.0116 last_data_time: 0.0122 lr:  
0.00025 max_mem: 2638M  
[11/19 04:16:02 d2.utils.events]: eta: 0:11:37 iter: 1279  
total_loss: 0.751 loss_cls: 0.2512 loss_box_reg: 0.4383  
loss_rpn_cls: 0.005032 loss_rpn_loc: 0.01375 time: 0.4019  
last_time: 0.4510 data_time: 0.0109 last_data_time: 0.0212 lr:  
0.00025 max_mem: 2638M  
[11/19 04:16:10 d2.utils.events]: eta: 0:11:30 iter: 1299  
total_loss: 0.6401 loss_cls: 0.2257 loss_box_reg: 0.3563  
loss_rpn_cls: 0.003755 loss_rpn_loc: 0.01612 time: 0.4020  
last_time: 0.5218 data_time: 0.0083 last_data_time: 0.0032 lr:  
0.00025 max_mem: 2638M  
[11/19 04:16:19 d2.utils.events]: eta: 0:11:22 iter: 1319  
total_loss: 0.6326 loss_cls: 0.1754 loss_box_reg: 0.4119  
loss_rpn_cls: 0.005659 loss_rpn_loc: 0.0231 time: 0.4024  
last_time: 0.3997 data_time: 0.0123 last_data_time: 0.0039 lr:  
0.00025 max_mem: 2638M  
[11/19 04:16:27 d2.utils.events]: eta: 0:11:11 iter: 1339  
total_loss: 0.753 loss_cls: 0.2258 loss_box_reg: 0.4455
```

```
loss_rpn_cls: 0.004585 loss_rpn_loc: 0.02357 time: 0.4023
last_time: 0.3372 data_time: 0.0098 last_data_time: 0.0057 lr:
0.00025 max_mem: 2638M
[11/19 04:16:35 d2.utils.events]: eta: 0:11:04 iter: 1359
total_loss: 0.6104 loss_cls: 0.217 loss_box_reg: 0.3553
loss_rpn_cls: 0.005985 loss_rpn_loc: 0.01792 time: 0.4026
last_time: 0.4412 data_time: 0.0121 last_data_time: 0.0169 lr:
0.00025 max_mem: 2638M
[11/19 04:16:44 d2.utils.events]: eta: 0:10:57 iter: 1379
total_loss: 0.6106 loss_cls: 0.1845 loss_box_reg: 0.3935
loss_rpn_cls: 0.002857 loss_rpn_loc: 0.01925 time: 0.4030
last_time: 0.3846 data_time: 0.0133 last_data_time: 0.0128 lr:
0.00025 max_mem: 2638M
[11/19 04:16:52 d2.utils.events]: eta: 0:10:49 iter: 1399
total_loss: 0.6609 loss_cls: 0.2 loss_box_reg: 0.4409 loss_rpn_cls:
0.005527 loss_rpn_loc: 0.01981 time: 0.4031 last_time: 0.3086
data_time: 0.0072 last_data_time: 0.0056 lr: 0.00025 max_mem:
2638M
[11/19 04:17:00 d2.utils.events]: eta: 0:10:41 iter: 1419
total_loss: 0.512 loss_cls: 0.1685 loss_box_reg: 0.3022
loss_rpn_cls: 0.004585 loss_rpn_loc: 0.01176 time: 0.4032
last_time: 0.3445 data_time: 0.0149 last_data_time: 0.0055 lr:
0.00025 max_mem: 2638M
[11/19 04:17:09 d2.utils.events]: eta: 0:10:33 iter: 1439
total_loss: 0.5783 loss_cls: 0.2121 loss_box_reg: 0.3676
loss_rpn_cls: 0.00511 loss_rpn_loc: 0.01383 time: 0.4034
last_time: 0.4298 data_time: 0.0098 last_data_time: 0.0015 lr:
0.00025 max_mem: 2638M
[11/19 04:17:17 d2.utils.events]: eta: 0:10:24 iter: 1459
total_loss: 0.5822 loss_cls: 0.1791 loss_box_reg: 0.3742
loss_rpn_cls: 0.004577 loss_rpn_loc: 0.01798 time: 0.4034
last_time: 0.3886 data_time: 0.0133 last_data_time: 0.0173 lr:
0.00025 max_mem: 2638M
[11/19 04:17:25 d2.utils.events]: eta: 0:10:15 iter: 1479
total_loss: 0.582 loss_cls: 0.2013 loss_box_reg: 0.3256
loss_rpn_cls: 0.005994 loss_rpn_loc: 0.01517 time: 0.4036
last_time: 0.3558 data_time: 0.0107 last_data_time: 0.0191 lr:
0.00025 max_mem: 2638M
[11/19 04:17:33 d2.utils.events]: eta: 0:10:08 iter: 1499
total_loss: 0.5375 loss_cls: 0.2058 loss_box_reg: 0.3471
loss_rpn_cls: 0.01061 loss_rpn_loc: 0.01638 time: 0.4036
last_time: 0.3678 data_time: 0.0098 last_data_time: 0.0131 lr:
0.00025 max_mem: 2638M
[11/19 04:17:41 d2.utils.events]: eta: 0:10:00 iter: 1519
total_loss: 0.573 loss_cls: 0.1721 loss_box_reg: 0.3439
loss_rpn_cls: 0.007941 loss_rpn_loc: 0.01418 time: 0.4037
last_time: 0.4578 data_time: 0.0100 last_data_time: 0.0154 lr:
0.00025 max_mem: 2638M
[11/19 04:17:50 d2.utils.events]: eta: 0:09:54 iter: 1539
```

```
total_loss: 0.7396 loss_cls: 0.2057 loss_box_reg: 0.3748
loss_rpn_cls: 0.005214 loss_rpn_loc: 0.01611 time: 0.4038
last_time: 0.3064 data_time: 0.0096 last_data_time: 0.0055 lr:
0.00025 max_mem: 2638M
[11/19 04:17:57 d2.utils.events]: eta: 0:09:43 iter: 1559
total_loss: 0.6035 loss_cls: 0.2296 loss_box_reg: 0.3444
loss_rpn_cls: 0.01131 loss_rpn_loc: 0.02399 time: 0.4036
last_time: 0.3646 data_time: 0.0105 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:18:05 d2.utils.events]: eta: 0:09:34 iter: 1579
total_loss: 0.52 loss_cls: 0.2008 loss_box_reg: 0.3568
loss_rpn_cls: 0.004544 loss_rpn_loc: 0.01098 time: 0.4036
last_time: 0.4645 data_time: 0.0077 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:18:14 d2.utils.events]: eta: 0:09:27 iter: 1599
total_loss: 0.665 loss_cls: 0.2343 loss_box_reg: 0.3935
loss_rpn_cls: 0.0128 loss_rpn_loc: 0.02027 time: 0.4039
last_time: 0.3690 data_time: 0.0107 last_data_time: 0.0123 lr:
0.00025 max_mem: 2638M
[11/19 04:18:23 d2.utils.events]: eta: 0:09:20 iter: 1619
total_loss: 0.6123 loss_cls: 0.1973 loss_box_reg: 0.3689
loss_rpn_cls: 0.007078 loss_rpn_loc: 0.02202 time: 0.4042
last_time: 0.4986 data_time: 0.0094 last_data_time: 0.0191 lr:
0.00025 max_mem: 2638M
[11/19 04:18:30 d2.utils.events]: eta: 0:09:09 iter: 1639
total_loss: 0.5172 loss_cls: 0.1539 loss_box_reg: 0.3194
loss_rpn_cls: 0.006882 loss_rpn_loc: 0.01434 time: 0.4040
last_time: 0.2865 data_time: 0.0113 last_data_time: 0.0033 lr:
0.00025 max_mem: 2638M
[11/19 04:18:38 d2.utils.events]: eta: 0:09:01 iter: 1659
total_loss: 0.4314 loss_cls: 0.1381 loss_box_reg: 0.2835
loss_rpn_cls: 0.004156 loss_rpn_loc: 0.01277 time: 0.4040
last_time: 0.4158 data_time: 0.0114 last_data_time: 0.0224 lr:
0.00025 max_mem: 2638M
[11/19 04:18:46 d2.utils.events]: eta: 0:08:52 iter: 1679
total_loss: 0.6296 loss_cls: 0.1748 loss_box_reg: 0.4156
loss_rpn_cls: 0.004307 loss_rpn_loc: 0.01976 time: 0.4037
last_time: 0.3156 data_time: 0.0093 last_data_time: 0.0034 lr:
0.00025 max_mem: 2638M
[11/19 04:18:54 d2.utils.events]: eta: 0:08:44 iter: 1699
total_loss: 0.5188 loss_cls: 0.1528 loss_box_reg: 0.332
loss_rpn_cls: 0.0055 loss_rpn_loc: 0.01487 time: 0.4037
last_time: 0.4273 data_time: 0.0149 last_data_time: 0.0061 lr:
0.00025 max_mem: 2638M
[11/19 04:19:03 d2.utils.events]: eta: 0:08:37 iter: 1719
total_loss: 0.4963 loss_cls: 0.1314 loss_box_reg: 0.371
loss_rpn_cls: 0.00363 loss_rpn_loc: 0.01296 time: 0.4040
last_time: 0.3993 data_time: 0.0090 last_data_time: 0.0153 lr:
0.00025 max_mem: 2638M
```

```
[11/19 04:19:11 d2.utils.events]: eta: 0:08:29 iter: 1739
total_loss: 0.6406 loss_cls: 0.1963 loss_box_reg: 0.3699
loss_rpn_cls: 0.004225 loss_rpn_loc: 0.0173 time: 0.4041
last_time: 0.3386 data_time: 0.0123 last_data_time: 0.0056 lr:
0.00025 max_mem: 2638M
[11/19 04:19:19 d2.utils.events]: eta: 0:08:20 iter: 1759
total_loss: 0.6347 loss_cls: 0.227 loss_box_reg: 0.3898
loss_rpn_cls: 0.009831 loss_rpn_loc: 0.01372 time: 0.4040
last_time: 0.4036 data_time: 0.0092 last_data_time: 0.0330 lr:
0.00025 max_mem: 2638M
[11/19 04:19:27 d2.utils.events]: eta: 0:08:12 iter: 1779
total_loss: 0.5768 loss_cls: 0.1953 loss_box_reg: 0.3468
loss_rpn_cls: 0.00704 loss_rpn_loc: 0.01805 time: 0.4040
last_time: 0.4328 data_time: 0.0100 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:19:35 d2.utils.events]: eta: 0:08:04 iter: 1799
total_loss: 0.6857 loss_cls: 0.2361 loss_box_reg: 0.3725
loss_rpn_cls: 0.01503 loss_rpn_loc: 0.03347 time: 0.4039
last_time: 0.3814 data_time: 0.0126 last_data_time: 0.0124 lr:
0.00025 max_mem: 2638M
[11/19 04:19:43 d2.utils.events]: eta: 0:07:56 iter: 1819
total_loss: 0.6545 loss_cls: 0.254 loss_box_reg: 0.3977
loss_rpn_cls: 0.01062 loss_rpn_loc: 0.01197 time: 0.4038
last_time: 0.4120 data_time: 0.0087 last_data_time: 0.0032 lr:
0.00025 max_mem: 2638M
[11/19 04:19:51 d2.utils.events]: eta: 0:07:48 iter: 1839
total_loss: 0.6162 loss_cls: 0.2126 loss_box_reg: 0.4058
loss_rpn_cls: 0.006696 loss_rpn_loc: 0.01309 time: 0.4039
last_time: 0.4061 data_time: 0.0115 last_data_time: 0.0060 lr:
0.00025 max_mem: 2638M
[11/19 04:19:59 d2.utils.events]: eta: 0:07:41 iter: 1859
total_loss: 0.7378 loss_cls: 0.24 loss_box_reg: 0.4366
loss_rpn_cls: 0.005945 loss_rpn_loc: 0.02032 time: 0.4040
last_time: 0.3990 data_time: 0.0083 last_data_time: 0.0178 lr:
0.00025 max_mem: 2638M
[11/19 04:20:08 d2.utils.events]: eta: 0:07:34 iter: 1879
total_loss: 0.5671 loss_cls: 0.1766 loss_box_reg: 0.3758
loss_rpn_cls: 0.003428 loss_rpn_loc: 0.02165 time: 0.4041
last_time: 0.3580 data_time: 0.0125 last_data_time: 0.0055 lr:
0.00025 max_mem: 2638M
[11/19 04:20:16 d2.utils.events]: eta: 0:07:27 iter: 1899
total_loss: 0.5089 loss_cls: 0.1894 loss_box_reg: 0.2789
loss_rpn_cls: 0.004321 loss_rpn_loc: 0.0208 time: 0.4042
last_time: 0.4480 data_time: 0.0101 last_data_time: 0.0130 lr:
0.00025 max_mem: 2638M
[11/19 04:20:24 d2.utils.events]: eta: 0:07:19 iter: 1919
total_loss: 0.6119 loss_cls: 0.1963 loss_box_reg: 0.3254
loss_rpn_cls: 0.004999 loss_rpn_loc: 0.0149 time: 0.4043
last_time: 0.3781 data_time: 0.0105 last_data_time: 0.0106 lr:
```

```
0.00025 max_mem: 2638M
[11/19 04:20:33 d2.utils.events]: eta: 0:07:11 iter: 1939
total_loss: 0.4969 loss_cls: 0.182 loss_box_reg: 0.295
loss_rpn_cls: 0.005383 loss_rpn_loc: 0.01356 time: 0.4044
last_time: 0.4330 data_time: 0.0110 last_data_time: 0.0107 lr:
0.00025 max_mem: 2638M
[11/19 04:20:41 d2.utils.events]: eta: 0:07:03 iter: 1959
total_loss: 0.6978 loss_cls: 0.2427 loss_box_reg: 0.346
loss_rpn_cls: 0.004019 loss_rpn_loc: 0.01362 time: 0.4046
last_time: 0.4780 data_time: 0.0098 last_data_time: 0.0057 lr:
0.00025 max_mem: 2638M
[11/19 04:20:49 d2.utils.events]: eta: 0:06:55 iter: 1979
total_loss: 0.5863 loss_cls: 0.1948 loss_box_reg: 0.3737
loss_rpn_cls: 0.004342 loss_rpn_loc: 0.01728 time: 0.4046
last_time: 0.4308 data_time: 0.0127 last_data_time: 0.0055 lr:
0.00025 max_mem: 2638M
[11/19 04:20:58 d2.utils.events]: eta: 0:06:47 iter: 1999
total_loss: 0.756 loss_cls: 0.2645 loss_box_reg: 0.4475
loss_rpn_cls: 0.00696 loss_rpn_loc: 0.02589 time: 0.4047
last_time: 0.4541 data_time: 0.0113 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:21:05 d2.utils.events]: eta: 0:06:39 iter: 2019
total_loss: 0.4808 loss_cls: 0.1523 loss_box_reg: 0.3049
loss_rpn_cls: 0.006386 loss_rpn_loc: 0.01676 time: 0.4045
last_time: 0.3204 data_time: 0.0086 last_data_time: 0.0055 lr:
0.00025 max_mem: 2638M
[11/19 04:21:13 d2.utils.events]: eta: 0:06:31 iter: 2039
total_loss: 0.5729 loss_cls: 0.2144 loss_box_reg: 0.3086
loss_rpn_cls: 0.00327 loss_rpn_loc: 0.01537 time: 0.4045
last_time: 0.4794 data_time: 0.0143 last_data_time: 0.0127 lr:
0.00025 max_mem: 2638M
[11/19 04:21:21 d2.utils.events]: eta: 0:06:23 iter: 2059
total_loss: 0.5345 loss_cls: 0.17 loss_box_reg: 0.3403
loss_rpn_cls: 0.003361 loss_rpn_loc: 0.02128 time: 0.4046
last_time: 0.3690 data_time: 0.0104 last_data_time: 0.0040 lr:
0.00025 max_mem: 2638M
[11/19 04:21:29 d2.utils.events]: eta: 0:06:15 iter: 2079
total_loss: 0.6065 loss_cls: 0.2306 loss_box_reg: 0.3592
loss_rpn_cls: 0.008094 loss_rpn_loc: 0.0191 time: 0.4045
last_time: 0.3436 data_time: 0.0144 last_data_time: 0.0059 lr:
0.00025 max_mem: 2638M
[11/19 04:21:38 d2.utils.events]: eta: 0:06:08 iter: 2099
total_loss: 0.4945 loss_cls: 0.1741 loss_box_reg: 0.3153
loss_rpn_cls: 0.00462 loss_rpn_loc: 0.01424 time: 0.4047
last_time: 0.5136 data_time: 0.0094 last_data_time: 0.0218 lr:
0.00025 max_mem: 2638M
[11/19 04:21:46 d2.utils.events]: eta: 0:05:59 iter: 2119
total_loss: 0.6313 loss_cls: 0.2381 loss_box_reg: 0.3467
loss_rpn_cls: 0.003304 loss_rpn_loc: 0.01808 time: 0.4048
last_time: 0.3575 data_time: 0.0128 last_data_time: 0.0058 lr:
```

```
0.00025 max_mem: 2638M
[11/19 04:21:55 d2.utils.events]: eta: 0:05:52 iter: 2139
total_loss: 0.4181 loss_cls: 0.1177 loss_box_reg: 0.2633
loss_rpn_cls: 0.00165 loss_rpn_loc: 0.01243 time: 0.4049
last_time: 0.4335 data_time: 0.0102 last_data_time: 0.0244 lr:
0.00025 max_mem: 2638M
[11/19 04:22:03 d2.utils.events]: eta: 0:05:44 iter: 2159
total_loss: 0.5787 loss_cls: 0.1654 loss_box_reg: 0.3888
loss_rpn_cls: 0.01069 loss_rpn_loc: 0.01641 time: 0.4049
last_time: 0.4700 data_time: 0.0101 last_data_time: 0.0059 lr:
0.00025 max_mem: 2638M
[11/19 04:22:11 d2.utils.events]: eta: 0:05:36 iter: 2179
total_loss: 0.4626 loss_cls: 0.1628 loss_box_reg: 0.3012
loss_rpn_cls: 0.009743 loss_rpn_loc: 0.01063 time: 0.4051
last_time: 0.3752 data_time: 0.0119 last_data_time: 0.0126 lr:
0.00025 max_mem: 2638M
[11/19 04:22:19 d2.utils.events]: eta: 0:05:27 iter: 2199
total_loss: 0.4919 loss_cls: 0.1636 loss_box_reg: 0.3221
loss_rpn_cls: 0.002821 loss_rpn_loc: 0.01242 time: 0.4048
last_time: 0.3056 data_time: 0.0100 last_data_time: 0.0042 lr:
0.00025 max_mem: 2638M
[11/19 04:22:27 d2.utils.events]: eta: 0:05:19 iter: 2219
total_loss: 0.5868 loss_cls: 0.156 loss_box_reg: 0.3511
loss_rpn_cls: 0.0104 loss_rpn_loc: 0.008049 time: 0.4049
last_time: 0.3710 data_time: 0.0136 last_data_time: 0.0126 lr:
0.00025 max_mem: 2638M
[11/19 04:22:35 d2.utils.events]: eta: 0:05:11 iter: 2239
total_loss: 0.5416 loss_cls: 0.15 loss_box_reg: 0.3621
loss_rpn_cls: 0.002697 loss_rpn_loc: 0.01282 time: 0.4049
last_time: 0.3813 data_time: 0.0100 last_data_time: 0.0118 lr:
0.00025 max_mem: 2638M
[11/19 04:22:43 d2.utils.events]: eta: 0:05:02 iter: 2259
total_loss: 0.5446 loss_cls: 0.1464 loss_box_reg: 0.3489
loss_rpn_cls: 0.00725 loss_rpn_loc: 0.01656 time: 0.4049
last_time: 0.3851 data_time: 0.0089 last_data_time: 0.0065 lr:
0.00025 max_mem: 2638M
[11/19 04:22:51 d2.utils.events]: eta: 0:04:53 iter: 2279
total_loss: 0.5331 loss_cls: 0.1548 loss_box_reg: 0.3045
loss_rpn_cls: 0.003299 loss_rpn_loc: 0.01652 time: 0.4048
last_time: 0.3875 data_time: 0.0116 last_data_time: 0.0297 lr:
0.00025 max_mem: 2638M
[11/19 04:22:59 d2.utils.events]: eta: 0:04:45 iter: 2299
total_loss: 0.5644 loss_cls: 0.1716 loss_box_reg: 0.3248
loss_rpn_cls: 0.00749 loss_rpn_loc: 0.01736 time: 0.4048
last_time: 0.3443 data_time: 0.0084 last_data_time: 0.0054 lr:
0.00025 max_mem: 2638M
[11/19 04:23:08 d2.utils.events]: eta: 0:04:37 iter: 2319
total_loss: 0.4598 loss_cls: 0.1332 loss_box_reg: 0.3 loss_rpn_cls:
0.003562 loss_rpn_loc: 0.01436 time: 0.4050 last_time: 0.4060
```

```
data_time: 0.0109 last_data_time: 0.0056 lr: 0.00025 max_mem:  
2638M  
[11/19 04:23:16 d2.utils.events]: eta: 0:04:29 iter: 2339  
total_loss: 0.448 loss_cls: 0.164 loss_box_reg: 0.2818  
loss_rpn_cls: 0.004766 loss_rpn_loc: 0.01246 time: 0.4050  
last_time: 0.4771 data_time: 0.0101 last_data_time: 0.0089 lr:  
0.00025 max_mem: 2638M  
[11/19 04:23:24 d2.utils.events]: eta: 0:04:21 iter: 2359  
total_loss: 0.5544 loss_cls: 0.1496 loss_box_reg: 0.3224  
loss_rpn_cls: 0.01355 loss_rpn_loc: 0.01249 time: 0.4051  
last_time: 0.4141 data_time: 0.0109 last_data_time: 0.0057 lr:  
0.00025 max_mem: 2638M  
[11/19 04:23:33 d2.utils.events]: eta: 0:04:13 iter: 2379  
total_loss: 0.6582 loss_cls: 0.2578 loss_box_reg: 0.3686  
loss_rpn_cls: 0.004065 loss_rpn_loc: 0.01243 time: 0.4052  
last_time: 0.4417 data_time: 0.0100 last_data_time: 0.0055 lr:  
0.00025 max_mem: 2638M  
[11/19 04:23:41 d2.utils.events]: eta: 0:04:05 iter: 2399  
total_loss: 0.5258 loss_cls: 0.1662 loss_box_reg: 0.3269  
loss_rpn_cls: 0.004387 loss_rpn_loc: 0.01222 time: 0.4053  
last_time: 0.4091 data_time: 0.0112 last_data_time: 0.0051 lr:  
0.00025 max_mem: 2638M  
[11/19 04:23:49 d2.utils.events]: eta: 0:03:56 iter: 2419  
total_loss: 0.5423 loss_cls: 0.1611 loss_box_reg: 0.3457  
loss_rpn_cls: 0.0104 loss_rpn_loc: 0.01046 time: 0.4053  
last_time: 0.3687 data_time: 0.0174 last_data_time: 0.0055 lr:  
0.00025 max_mem: 2638M  
[11/19 04:23:57 d2.utils.events]: eta: 0:03:48 iter: 2439  
total_loss: 0.5823 loss_cls: 0.2038 loss_box_reg: 0.3772  
loss_rpn_cls: 0.01076 loss_rpn_loc: 0.01363 time: 0.4054  
last_time: 0.3829 data_time: 0.0092 last_data_time: 0.0082 lr:  
0.00025 max_mem: 2638M  
[11/19 04:24:06 d2.utils.events]: eta: 0:03:41 iter: 2459  
total_loss: 0.4578 loss_cls: 0.1639 loss_box_reg: 0.2892  
loss_rpn_cls: 0.004668 loss_rpn_loc: 0.0233 time: 0.4055  
last_time: 0.3601 data_time: 0.0140 last_data_time: 0.0055 lr:  
0.00025 max_mem: 2638M  
[11/19 04:24:14 d2.utils.events]: eta: 0:03:32 iter: 2479  
total_loss: 0.528 loss_cls: 0.1894 loss_box_reg: 0.311  
loss_rpn_cls: 0.003922 loss_rpn_loc: 0.0172 time: 0.4054  
last_time: 0.3913 data_time: 0.0106 last_data_time: 0.0055 lr:  
0.00025 max_mem: 2638M  
[11/19 04:24:22 d2.utils.events]: eta: 0:03:23 iter: 2499  
total_loss: 0.4342 loss_cls: 0.1454 loss_box_reg: 0.2765  
loss_rpn_cls: 0.002606 loss_rpn_loc: 0.008482 time: 0.4054  
last_time: 0.4113 data_time: 0.0102 last_data_time: 0.0036 lr:  
0.00025 max_mem: 2638M  
[11/19 04:24:30 d2.utils.events]: eta: 0:03:15 iter: 2519  
total_loss: 0.4797 loss_cls: 0.1476 loss_box_reg: 0.2759
```

```
loss_rpn_cls: 0.004354 loss_rpn_loc: 0.01485 time: 0.4053
last_time: 0.4077 data_time: 0.0115 last_data_time: 0.0216 lr:
0.00025 max_mem: 2638M
[11/19 04:24:38 d2.utils.events]: eta: 0:03:07 iter: 2539
total_loss: 0.5 loss_cls: 0.1891 loss_box_reg: 0.2987 loss_rpn_cls:
0.002303 loss_rpn_loc: 0.01365 time: 0.4054 last_time: 0.3500
data_time: 0.0090 last_data_time: 0.0119 lr: 0.00025 max_mem:
2638M
[11/19 04:24:46 d2.utils.events]: eta: 0:03:00 iter: 2559
total_loss: 0.5126 loss_cls: 0.1667 loss_box_reg: 0.3 loss_rpn_cls:
0.003626 loss_rpn_loc: 0.0128 time: 0.4055 last_time: 0.4358
data_time: 0.0141 last_data_time: 0.0057 lr: 0.00025 max_mem:
2638M
[11/19 04:24:54 d2.utils.events]: eta: 0:02:51 iter: 2579
total_loss: 0.5206 loss_cls: 0.1789 loss_box_reg: 0.3053
loss_rpn_cls: 0.005958 loss_rpn_loc: 0.01111 time: 0.4054
last_time: 0.4368 data_time: 0.0094 last_data_time: 0.0048 lr:
0.00025 max_mem: 2638M
[11/19 04:25:02 d2.utils.events]: eta: 0:02:43 iter: 2599
total_loss: 0.5843 loss_cls: 0.1927 loss_box_reg: 0.35
loss_rpn_cls: 0.00359 loss_rpn_loc: 0.01179 time: 0.4052
last_time: 0.4769 data_time: 0.0094 last_data_time: 0.0119 lr:
0.00025 max_mem: 2638M
[11/19 04:25:10 d2.utils.events]: eta: 0:02:34 iter: 2619
total_loss: 0.4929 loss_cls: 0.1814 loss_box_reg: 0.2685
loss_rpn_cls: 0.006322 loss_rpn_loc: 0.01742 time: 0.4052
last_time: 0.5263 data_time: 0.0098 last_data_time: 0.0058 lr:
0.00025 max_mem: 2638M
[11/19 04:25:18 d2.utils.events]: eta: 0:02:26 iter: 2639
total_loss: 0.5923 loss_cls: 0.1759 loss_box_reg: 0.364
loss_rpn_cls: 0.01089 loss_rpn_loc: 0.01984 time: 0.4053
last_time: 0.4661 data_time: 0.0110 last_data_time: 0.0041 lr:
0.00025 max_mem: 2638M
[11/19 04:25:27 d2.utils.events]: eta: 0:02:19 iter: 2659
total_loss: 0.511 loss_cls: 0.1616 loss_box_reg: 0.2892
loss_rpn_cls: 0.00598 loss_rpn_loc: 0.01989 time: 0.4053
last_time: 0.3477 data_time: 0.0106 last_data_time: 0.0057 lr:
0.00025 max_mem: 2638M
[11/19 04:25:35 d2.utils.events]: eta: 0:02:11 iter: 2679
total_loss: 0.5769 loss_cls: 0.1755 loss_box_reg: 0.3187
loss_rpn_cls: 0.007731 loss_rpn_loc: 0.01764 time: 0.4054
last_time: 0.3796 data_time: 0.0091 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:25:43 d2.utils.events]: eta: 0:02:02 iter: 2699
total_loss: 0.4419 loss_cls: 0.1813 loss_box_reg: 0.2742
loss_rpn_cls: 0.003773 loss_rpn_loc: 0.01063 time: 0.4054
last_time: 0.4360 data_time: 0.0124 last_data_time: 0.0148 lr:
0.00025 max_mem: 2638M
[11/19 04:25:51 d2.utils.events]: eta: 0:01:54 iter: 2719
```

```
total_loss: 0.4818 loss_cls: 0.161 loss_box_reg: 0.2614
loss_rpn_cls: 0.005883 loss_rpn_loc: 0.01559 time: 0.4054
last_time: 0.3335 data_time: 0.0089 last_data_time: 0.0214 lr:
0.00025 max_mem: 2638M
[11/19 04:25:59 d2.utils.events]: eta: 0:01:46 iter: 2739
total_loss: 0.4357 loss_cls: 0.1328 loss_box_reg: 0.2502
loss_rpn_cls: 0.001338 loss_rpn_loc: 0.0121 time: 0.4054
last_time: 0.3541 data_time: 0.0117 last_data_time: 0.0204 lr:
0.00025 max_mem: 2638M
[11/19 04:26:07 d2.utils.events]: eta: 0:01:38 iter: 2759
total_loss: 0.4136 loss_cls: 0.1294 loss_box_reg: 0.2456
loss_rpn_cls: 0.004563 loss_rpn_loc: 0.01343 time: 0.4054
last_time: 0.3357 data_time: 0.0160 last_data_time: 0.0256 lr:
0.00025 max_mem: 2638M
[11/19 04:26:16 d2.utils.events]: eta: 0:01:30 iter: 2779
total_loss: 0.6304 loss_cls: 0.1897 loss_box_reg: 0.3995
loss_rpn_cls: 0.003943 loss_rpn_loc: 0.01357 time: 0.4055
last_time: 0.3818 data_time: 0.0091 last_data_time: 0.0076 lr:
0.00025 max_mem: 2638M
[11/19 04:26:24 d2.utils.events]: eta: 0:01:22 iter: 2799
total_loss: 0.4581 loss_cls: 0.1419 loss_box_reg: 0.2941
loss_rpn_cls: 0.006504 loss_rpn_loc: 0.01592 time: 0.4054
last_time: 0.4468 data_time: 0.0135 last_data_time: 0.0054 lr:
0.00025 max_mem: 2638M
[11/19 04:26:32 d2.utils.events]: eta: 0:01:13 iter: 2819
total_loss: 0.5407 loss_cls: 0.1747 loss_box_reg: 0.2873
loss_rpn_cls: 0.003139 loss_rpn_loc: 0.01444 time: 0.4054
last_time: 0.3848 data_time: 0.0104 last_data_time: 0.0018 lr:
0.00025 max_mem: 2638M
[11/19 04:26:40 d2.utils.events]: eta: 0:01:05 iter: 2839
total_loss: 0.546 loss_cls: 0.1822 loss_box_reg: 0.3103
loss_rpn_cls: 0.006461 loss_rpn_loc: 0.01942 time: 0.4055
last_time: 0.3802 data_time: 0.0154 last_data_time: 0.0181 lr:
0.00025 max_mem: 2638M
[11/19 04:26:48 d2.utils.events]: eta: 0:00:57 iter: 2859
total_loss: 0.4469 loss_cls: 0.1196 loss_box_reg: 0.2234
loss_rpn_cls: 0.002144 loss_rpn_loc: 0.01546 time: 0.4055
last_time: 0.5112 data_time: 0.0114 last_data_time: 0.0213 lr:
0.00025 max_mem: 2638M
[11/19 04:26:56 d2.utils.events]: eta: 0:00:49 iter: 2879
total_loss: 0.4261 loss_cls: 0.1365 loss_box_reg: 0.2541
loss_rpn_cls: 0.001831 loss_rpn_loc: 0.009255 time: 0.4055
last_time: 0.4443 data_time: 0.0085 last_data_time: 0.0053 lr:
0.00025 max_mem: 2638M
[11/19 04:27:05 d2.utils.events]: eta: 0:00:41 iter: 2899
total_loss: 0.5043 loss_cls: 0.1718 loss_box_reg: 0.3244
loss_rpn_cls: 0.004388 loss_rpn_loc: 0.01379 time: 0.4056
last_time: 0.4129 data_time: 0.0138 last_data_time: 0.0126 lr:
0.00025 max_mem: 2638M
```

```

[11/19 04:27:13 d2.utils.events]: eta: 0:00:32 iter: 2919
total_loss: 0.5268 loss_cls: 0.1701 loss_box_reg: 0.3197
loss_rpn_cls: 0.004349 loss_rpn_loc: 0.01654 time: 0.4056
last_time: 0.3821 data_time: 0.0106 last_data_time: 0.0037 lr:
0.00025 max_mem: 2638M
[11/19 04:27:21 d2.utils.events]: eta: 0:00:24 iter: 2939
total_loss: 0.5705 loss_cls: 0.1853 loss_box_reg: 0.3566
loss_rpn_cls: 0.007493 loss_rpn_loc: 0.02383 time: 0.4057
last_time: 0.4294 data_time: 0.0149 last_data_time: 0.0084 lr:
0.00025 max_mem: 2638M
[11/19 04:27:29 d2.utils.events]: eta: 0:00:16 iter: 2959
total_loss: 0.4066 loss_cls: 0.1278 loss_box_reg: 0.2713
loss_rpn_cls: 0.002102 loss_rpn_loc: 0.01239 time: 0.4056
last_time: 0.4274 data_time: 0.0110 last_data_time: 0.0241 lr:
0.00025 max_mem: 2638M
[11/19 04:27:37 d2.utils.events]: eta: 0:00:08 iter: 2979
total_loss: 0.403 loss_cls: 0.1084 loss_box_reg: 0.2922
loss_rpn_cls: 0.002218 loss_rpn_loc: 0.01197 time: 0.4056
last_time: 0.5023 data_time: 0.0113 last_data_time: 0.0073 lr:
0.00025 max_mem: 2638M
[11/19 04:27:51 d2.utils.events]: eta: 0:00:00 iter: 2999
total_loss: 0.5597 loss_cls: 0.1818 loss_box_reg: 0.3241
loss_rpn_cls: 0.003884 loss_rpn_loc: 0.01155 time: 0.4057
last_time: 0.5002 data_time: 0.0125 last_data_time: 0.0267 lr:
0.00025 max_mem: 2638M
[11/19 04:27:52 d2.engine.hooks]: Overall training speed: 2998
iterations in 0:20:16 (0.4057 s / it)
[11/19 04:27:52 d2.engine.hooks]: Total training time: 0:20:29
(0:00:13 on hooks)
[11/19 04:27:53 d2.data.build]: Distribution of instances among all 20
categories:


| category    | #instances | category    | #instances | category |
|-------------|------------|-------------|------------|----------|
| #instances  |            |             | <th></th>  |          |
| aeroplane   | 484        | bicycle     | 380        | bird     |
| 629         |            |             |            |          |
| boat        | 491        | bottle      | 733        | bus      |
| 320         |            |             |            |          |
| car         | 1173       | cat         | 618        | chair    |
| 1449        |            |             |            |          |
| cow         | 347        | diningtable | 374        | dog      |
| 773         |            |             |            |          |
| horse       | 373        | motorbike   | 376        | person   |
| 5110        |            |             |            |          |
| pottedplant | 542        | sheep       | 485        | sofa     |
| 387         |            |             |            |          |
| train       | 329        | tvmonitor   | 414        |          |
|             |            |             |            |          |


```

```
| total | 15787 | | |  
[11/19 04:27:53 d2.data.dataset_mapper]: [DatasetMapper] Augmentations  
used in inference: [ResizeShortestEdge(short_edge_length=(800, 800),  
max_size=1333, sample_style='choice')]  
[11/19 04:27:53 d2.data.common]: Serializing the dataset using: <class  
'detectron2.data.common._TorchSerializedList'>  
[11/19 04:27:53 d2.data.common]: Serializing 5823 elements to byte  
tensors and concatenating them all ...  
[11/19 04:27:53 d2.data.common]: Serialized dataset takes 2.67 MiB  
WARNING [11/19 04:27:53 d2.engine.defaults]: No evaluator found. Use  
'DefaultTrainer.test(evaluators=)', or implement its `build_evaluator`  
method.  
  
from detectron2.evaluation import PascalVOCDetectionEvaluator  
# Set up the evaluator  
evaluator = PascalVOCDetectionEvaluator("voc_2012_val")  
  
from detectron2.engine import DefaultPredictor  
from detectron2.data import build_detection_test_loader  
from detectron2.evaluation import inference_on_dataset  
  
# Load the fine-tuned model  
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")  
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5  
predictor = DefaultPredictor(cfg)  
# Run evaluation  
val_loader = build_detection_test_loader(cfg, "voc_2012_val")  
print(inference_on_dataset(predictor.model, val_loader))  
  
[11/19 04:30:15 d2.checkpoint.detection_checkpoint]:  
[DetectionCheckpointer] Loading from MyVOCTraining/model_final.pth ...  
[11/19 04:30:17 d2.data.dataset_mapper]: [DatasetMapper] Augmentations  
used in inference: [ResizeShortestEdge(short_edge_length=(800, 800),  
max_size=1333, sample_style='choice')]  
[11/19 04:30:17 d2.data.common]: Serializing the dataset using: <class  
'detectron2.data.common._TorchSerializedList'>  
[11/19 04:30:17 d2.data.common]: Serializing 5823 elements to byte  
tensors and concatenating them all ...  
[11/19 04:30:17 d2.data.common]: Serialized dataset takes 2.67 MiB  
[11/19 04:30:17 d2.evaluation.evaluator]: Start inference on 5823  
batches  
[11/19 04:30:19 d2.evaluation.evaluator]: Inference done 11/5823.  
Dataloading: 0.0028 s/iter. Inference: 0.0995 s/iter. Eval: 0.0003  
s/iter. Total: 0.1025 s/iter. ETA=0:09:55  
[11/19 04:30:24 d2.evaluation.evaluator]: Inference done 60/5823.  
Dataloading: 0.0020 s/iter. Inference: 0.1000 s/iter. Eval: 0.0002  
s/iter. Total: 0.1023 s/iter. ETA=0:09:49  
[11/19 04:30:29 d2.evaluation.evaluator]: Inference done 106/5823.  
Dataloading: 0.0036 s/iter. Inference: 0.1020 s/iter. Eval: 0.0002
```

```
s/iter. Total: 0.1059 s/iter. ETA=0:10:05
[11/19 04:30:34 d2.evaluation.evaluator]: Inference done 154/5823.
Dataloading: 0.0038 s/iter. Inference: 0.1018 s/iter. Eval: 0.0003
s/iter. Total: 0.1060 s/iter. ETA=0:10:00
[11/19 04:30:39 d2.evaluation.evaluator]: Inference done 202/5823.
Dataloading: 0.0033 s/iter. Inference: 0.1020 s/iter. Eval: 0.0002
s/iter. Total: 0.1057 s/iter. ETA=0:09:54
[11/19 04:30:44 d2.evaluation.evaluator]: Inference done 249/5823.
Dataloading: 0.0036 s/iter. Inference: 0.1024 s/iter. Eval: 0.0003
s/iter. Total: 0.1063 s/iter. ETA=0:09:52
[11/19 04:30:49 d2.evaluation.evaluator]: Inference done 295/5823.
Dataloading: 0.0039 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003
s/iter. Total: 0.1068 s/iter. ETA=0:09:50
[11/19 04:30:54 d2.evaluation.evaluator]: Inference done 344/5823.
Dataloading: 0.0036 s/iter. Inference: 0.1024 s/iter. Eval: 0.0002
s/iter. Total: 0.1063 s/iter. ETA=0:09:42
[11/19 04:30:59 d2.evaluation.evaluator]: Inference done 390/5823.
Dataloading: 0.0038 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003
s/iter. Total: 0.1068 s/iter. ETA=0:09:40
[11/19 04:31:04 d2.evaluation.evaluator]: Inference done 435/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1030 s/iter. Eval: 0.0003
s/iter. Total: 0.1075 s/iter. ETA=0:09:39
[11/19 04:31:10 d2.evaluation.evaluator]: Inference done 483/5823.
Dataloading: 0.0039 s/iter. Inference: 0.1031 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:09:33
[11/19 04:31:15 d2.evaluation.evaluator]: Inference done 530/5823.
Dataloading: 0.0039 s/iter. Inference: 0.1031 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:09:28
[11/19 04:31:20 d2.evaluation.evaluator]: Inference done 576/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1032 s/iter. Eval: 0.0003
s/iter. Total: 0.1076 s/iter. ETA=0:09:24
[11/19 04:31:25 d2.evaluation.evaluator]: Inference done 625/5823.
Dataloading: 0.0038 s/iter. Inference: 0.1031 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:09:17
[11/19 04:31:30 d2.evaluation.evaluator]: Inference done 672/5823.
Dataloading: 0.0039 s/iter. Inference: 0.1031 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:09:12
[11/19 04:31:35 d2.evaluation.evaluator]: Inference done 718/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1030 s/iter. Eval: 0.0003
s/iter. Total: 0.1075 s/iter. ETA=0:09:08
[11/19 04:31:40 d2.evaluation.evaluator]: Inference done 767/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1029 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:09:02
[11/19 04:31:45 d2.evaluation.evaluator]: Inference done 815/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1029 s/iter. Eval: 0.0003
s/iter. Total: 0.1072 s/iter. ETA=0:08:56
[11/19 04:31:50 d2.evaluation.evaluator]: Inference done 861/5823.
Dataloading: 0.0042 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003
s/iter. Total: 0.1073 s/iter. ETA=0:08:52
```

```
[11/19 04:31:55 d2.evaluation.evaluator]: Inference done 910/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1071 s/iter. ETA=0:08:46  
[11/19 04:32:00 d2.evaluation.evaluator]: Inference done 958/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:08:40  
[11/19 04:32:05 d2.evaluation.evaluator]: Inference done 1003/5823.  
Dataloading: 0.0042 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1072 s/iter. ETA=0:08:36  
[11/19 04:32:10 d2.evaluation.evaluator]: Inference done 1051/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1071 s/iter. ETA=0:08:31  
[11/19 04:32:15 d2.evaluation.evaluator]: Inference done 1100/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1025 s/iter. Eval: 0.0003  
s/iter. Total: 0.1069 s/iter. ETA=0:08:24  
[11/19 04:32:20 d2.evaluation.evaluator]: Inference done 1146/5823.  
Dataloading: 0.0042 s/iter. Inference: 0.1025 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:08:20  
[11/19 04:32:25 d2.evaluation.evaluator]: Inference done 1194/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1025 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:08:15  
[11/19 04:32:30 d2.evaluation.evaluator]: Inference done 1241/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:08:10  
[11/19 04:32:36 d2.evaluation.evaluator]: Inference done 1287/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1071 s/iter. ETA=0:08:05  
[11/19 04:32:41 d2.evaluation.evaluator]: Inference done 1335/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:08:00  
[11/19 04:32:46 d2.evaluation.evaluator]: Inference done 1383/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1069 s/iter. ETA=0:07:54  
[11/19 04:32:51 d2.evaluation.evaluator]: Inference done 1429/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1071 s/iter. ETA=0:07:50  
[11/19 04:32:56 d2.evaluation.evaluator]: Inference done 1477/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:45  
[11/19 04:33:01 d2.evaluation.evaluator]: Inference done 1525/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1069 s/iter. ETA=0:07:39  
[11/19 04:33:06 d2.evaluation.evaluator]: Inference done 1571/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:35  
[11/19 04:33:11 d2.evaluation.evaluator]: Inference done 1618/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:30  
[11/19 04:33:16 d2.evaluation.evaluator]: Inference done 1666/5823.
```

```
Dataloading: 0.0040 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:24  
[11/19 04:33:21 d2.evaluation.evaluator]: Inference done 1713/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:19  
[11/19 04:33:26 d2.evaluation.evaluator]: Inference done 1760/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1026 s/iter. Eval: 0.0003  
s/iter. Total: 0.1070 s/iter. ETA=0:07:14  
[11/19 04:33:31 d2.evaluation.evaluator]: Inference done 1807/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1027 s/iter. Eval: 0.0003  
s/iter. Total: 0.1071 s/iter. ETA=0:07:09  
[11/19 04:33:36 d2.evaluation.evaluator]: Inference done 1843/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1033 s/iter. Eval: 0.0003  
s/iter. Total: 0.1078 s/iter. ETA=0:07:09  
[11/19 04:33:41 d2.evaluation.evaluator]: Inference done 1884/5823.  
Dataloading: 0.0042 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:07:05  
[11/19 04:33:46 d2.evaluation.evaluator]: Inference done 1932/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:07:00  
[11/19 04:33:51 d2.evaluation.evaluator]: Inference done 1979/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1035 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:06:55  
[11/19 04:33:57 d2.evaluation.evaluator]: Inference done 2027/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1035 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:06:49  
[11/19 04:34:02 d2.evaluation.evaluator]: Inference done 2076/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1035 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:06:44  
[11/19 04:34:07 d2.evaluation.evaluator]: Inference done 2123/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1035 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:06:39  
[11/19 04:34:12 d2.evaluation.evaluator]: Inference done 2170/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:06:34  
[11/19 04:34:17 d2.evaluation.evaluator]: Inference done 2219/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003  
s/iter. Total: 0.1078 s/iter. ETA=0:06:28  
[11/19 04:34:22 d2.evaluation.evaluator]: Inference done 2266/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003  
s/iter. Total: 0.1078 s/iter. ETA=0:06:23  
[11/19 04:34:27 d2.evaluation.evaluator]: Inference done 2312/5823.  
Dataloading: 0.0041 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003  
s/iter. Total: 0.1078 s/iter. ETA=0:06:18  
[11/19 04:34:32 d2.evaluation.evaluator]: Inference done 2360/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003  
s/iter. Total: 0.1078 s/iter. ETA=0:06:13  
[11/19 04:34:37 d2.evaluation.evaluator]: Inference done 2406/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003
```

```
s/iter. Total: 0.1078 s/iter. ETA=0:06:08
[11/19 04:34:42 d2.evaluation.evaluator]: Inference done 2452/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003
s/iter. Total: 0.1079 s/iter. ETA=0:06:03
[11/19 04:34:47 d2.evaluation.evaluator]: Inference done 2500/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003
s/iter. Total: 0.1078 s/iter. ETA=0:05:58
[11/19 04:34:52 d2.evaluation.evaluator]: Inference done 2547/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003
s/iter. Total: 0.1078 s/iter. ETA=0:05:53
[11/19 04:34:57 d2.evaluation.evaluator]: Inference done 2593/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1034 s/iter. Eval: 0.0003
s/iter. Total: 0.1078 s/iter. ETA=0:05:48
[11/19 04:35:02 d2.evaluation.evaluator]: Inference done 2637/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:43
[11/19 04:35:07 d2.evaluation.evaluator]: Inference done 2684/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:38
[11/19 04:35:12 d2.evaluation.evaluator]: Inference done 2729/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:34
[11/19 04:35:18 d2.evaluation.evaluator]: Inference done 2777/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:28
[11/19 04:35:23 d2.evaluation.evaluator]: Inference done 2824/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:23
[11/19 04:35:28 d2.evaluation.evaluator]: Inference done 2870/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:18
[11/19 04:35:33 d2.evaluation.evaluator]: Inference done 2918/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:13
[11/19 04:35:38 d2.evaluation.evaluator]: Inference done 2965/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:05:08
[11/19 04:35:43 d2.evaluation.evaluator]: Inference done 3010/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003
s/iter. Total: 0.1081 s/iter. ETA=0:05:03
[11/19 04:35:48 d2.evaluation.evaluator]: Inference done 3058/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:04:58
[11/19 04:35:53 d2.evaluation.evaluator]: Inference done 3105/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:04:53
[11/19 04:35:58 d2.evaluation.evaluator]: Inference done 3151/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003
s/iter. Total: 0.1080 s/iter. ETA=0:04:48
```

```
[11/19 04:36:03 d2.evaluation.evaluator]: Inference done 3198/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:43  
[11/19 04:36:08 d2.evaluation.evaluator]: Inference done 3245/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:38  
[11/19 04:36:13 d2.evaluation.evaluator]: Inference done 3290/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:04:33  
[11/19 04:36:18 d2.evaluation.evaluator]: Inference done 3337/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:28  
[11/19 04:36:23 d2.evaluation.evaluator]: Inference done 3385/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:23  
[11/19 04:36:28 d2.evaluation.evaluator]: Inference done 3430/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:18  
[11/19 04:36:33 d2.evaluation.evaluator]: Inference done 3479/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:13  
[11/19 04:36:38 d2.evaluation.evaluator]: Inference done 3527/5823.  
Dataloading: 0.0039 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:07  
[11/19 04:36:44 d2.evaluation.evaluator]: Inference done 3573/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:04:02  
[11/19 04:36:49 d2.evaluation.evaluator]: Inference done 3622/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:03:57  
[11/19 04:36:54 d2.evaluation.evaluator]: Inference done 3670/5823.  
Dataloading: 0.0039 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:03:52  
[11/19 04:36:59 d2.evaluation.evaluator]: Inference done 3715/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:03:47  
[11/19 04:37:04 d2.evaluation.evaluator]: Inference done 3763/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1036 s/iter. Eval: 0.0003  
s/iter. Total: 0.1079 s/iter. ETA=0:03:42  
[11/19 04:37:09 d2.evaluation.evaluator]: Inference done 3804/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:38  
[11/19 04:37:14 d2.evaluation.evaluator]: Inference done 3848/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003  
s/iter. Total: 0.1082 s/iter. ETA=0:03:33  
[11/19 04:37:19 d2.evaluation.evaluator]: Inference done 3896/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:28  
[11/19 04:37:24 d2.evaluation.evaluator]: Inference done 3944/5823.
```

```
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:23  
[11/19 04:37:29 d2.evaluation.evaluator]: Inference done 3990/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:18  
[11/19 04:37:34 d2.evaluation.evaluator]: Inference done 4038/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:12  
[11/19 04:37:39 d2.evaluation.evaluator]: Inference done 4085/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:03:07  
[11/19 04:37:44 d2.evaluation.evaluator]: Inference done 4130/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:03:02  
[11/19 04:37:49 d2.evaluation.evaluator]: Inference done 4178/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1081 s/iter. ETA=0:02:57  
[11/19 04:37:54 d2.evaluation.evaluator]: Inference done 4226/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:02:52  
[11/19 04:37:59 d2.evaluation.evaluator]: Inference done 4272/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:02:47  
[11/19 04:38:04 d2.evaluation.evaluator]: Inference done 4320/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:02:42  
[11/19 04:38:09 d2.evaluation.evaluator]: Inference done 4368/5823.  
Dataloading: 0.0039 s/iter. Inference: 0.1037 s/iter. Eval: 0.0003  
s/iter. Total: 0.1080 s/iter. ETA=0:02:37  
[11/19 04:38:14 d2.evaluation.evaluator]: Inference done 4407/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003  
s/iter. Total: 0.1082 s/iter. ETA=0:02:33  
[11/19 04:38:20 d2.evaluation.evaluator]: Inference done 4450/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003  
s/iter. Total: 0.1083 s/iter. ETA=0:02:28  
[11/19 04:38:25 d2.evaluation.evaluator]: Inference done 4495/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003  
s/iter. Total: 0.1083 s/iter. ETA=0:02:23  
[11/19 04:38:30 d2.evaluation.evaluator]: Inference done 4541/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003  
s/iter. Total: 0.1083 s/iter. ETA=0:02:18  
[11/19 04:38:35 d2.evaluation.evaluator]: Inference done 4588/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003  
s/iter. Total: 0.1083 s/iter. ETA=0:02:13  
[11/19 04:38:40 d2.evaluation.evaluator]: Inference done 4635/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003  
s/iter. Total: 0.1083 s/iter. ETA=0:02:08  
[11/19 04:38:45 d2.evaluation.evaluator]: Inference done 4678/5823.  
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
```

```
s/iter. Total: 0.1084 s/iter. ETA=0:02:04
[11/19 04:38:50 d2.evaluation.evaluator]: Inference done 4725/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:59
[11/19 04:38:55 d2.evaluation.evaluator]: Inference done 4772/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:53
[11/19 04:39:00 d2.evaluation.evaluator]: Inference done 4819/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:48
[11/19 04:39:05 d2.evaluation.evaluator]: Inference done 4866/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:43
[11/19 04:39:10 d2.evaluation.evaluator]: Inference done 4914/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:01:38
[11/19 04:39:15 d2.evaluation.evaluator]: Inference done 4960/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:33
[11/19 04:39:20 d2.evaluation.evaluator]: Inference done 5006/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:28
[11/19 04:39:25 d2.evaluation.evaluator]: Inference done 5054/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:01:23
[11/19 04:39:30 d2.evaluation.evaluator]: Inference done 5101/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1040 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:01:18
[11/19 04:39:35 d2.evaluation.evaluator]: Inference done 5147/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1084 s/iter. ETA=0:01:13
[11/19 04:39:40 d2.evaluation.evaluator]: Inference done 5195/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:01:08
[11/19 04:39:46 d2.evaluation.evaluator]: Inference done 5242/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:01:02
[11/19 04:39:51 d2.evaluation.evaluator]: Inference done 5288/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:57
[11/19 04:39:56 d2.evaluation.evaluator]: Inference done 5336/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:52
[11/19 04:40:01 d2.evaluation.evaluator]: Inference done 5383/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:47
[11/19 04:40:06 d2.evaluation.evaluator]: Inference done 5429/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:42
```

```

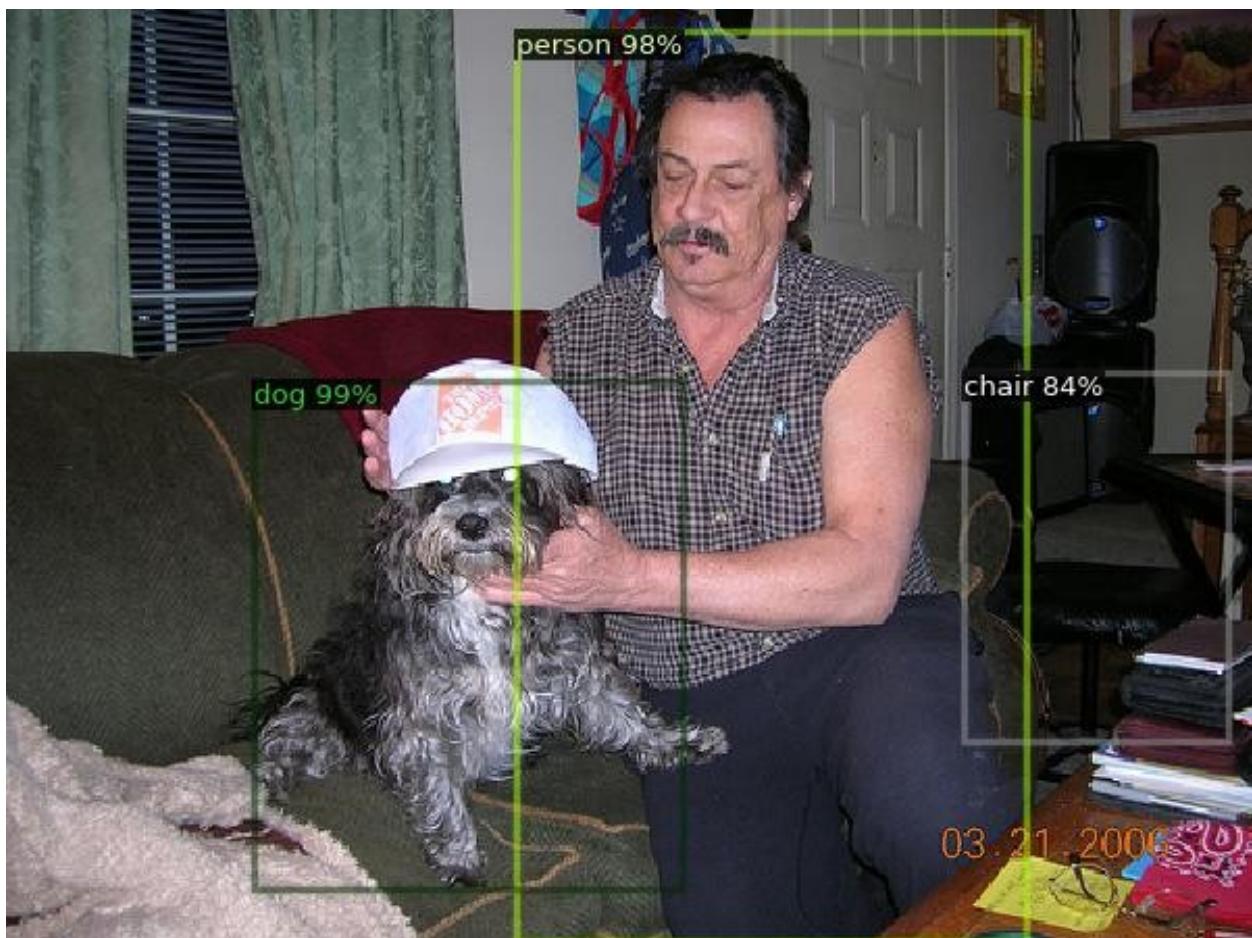
[11/19 04:40:11 d2.evaluation.evaluator]: Inference done 5477/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:37
[11/19 04:40:16 d2.evaluation.evaluator]: Inference done 5525/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:32
[11/19 04:40:21 d2.evaluation.evaluator]: Inference done 5571/5823.
Dataloading: 0.0041 s/iter. Inference: 0.1039 s/iter. Eval: 0.0003
s/iter. Total: 0.1083 s/iter. ETA=0:00:27
[11/19 04:40:26 d2.evaluation.evaluator]: Inference done 5619/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003
s/iter. Total: 0.1082 s/iter. ETA=0:00:22
[11/19 04:40:31 d2.evaluation.evaluator]: Inference done 5667/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003
s/iter. Total: 0.1082 s/iter. ETA=0:00:16
[11/19 04:40:36 d2.evaluation.evaluator]: Inference done 5713/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003
s/iter. Total: 0.1082 s/iter. ETA=0:00:11
[11/19 04:40:41 d2.evaluation.evaluator]: Inference done 5762/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003
s/iter. Total: 0.1082 s/iter. ETA=0:00:06
[11/19 04:40:46 d2.evaluation.evaluator]: Inference done 5810/5823.
Dataloading: 0.0040 s/iter. Inference: 0.1038 s/iter. Eval: 0.0003
s/iter. Total: 0.1082 s/iter. ETA=0:00:01
[11/19 04:40:48 d2.evaluation.evaluator]: Total inference time:
0:10:29.374111 (0.108177 s / iter per device, on 1 devices)
[11/19 04:40:48 d2.evaluation.evaluator]: Total inference pure compute
time: 0:10:03 (0.103793 s / iter per device, on 1 devices)
[11/19 04:40:48 d2.evaluation.pascal_voc_evaluation]: Evaluating
voc_2012_val using 2012 metric. Note that results do not use the
official Matlab API.
OrderedDict([('bbox', {'AP': 49.8443324084568, 'AP50': 74.86616539539328, 'AP75': 57.37197922054671})])

from detectron2.utils.visualizer import Visualizer
from detectron2.data import DatasetCatalog
import random
import cv2
from google.colab.patches import cv2_imshow
# Get validation dataset
dataset_dicts = DatasetCatalog.get("voc_2012_val")

# Visualize predictions on random images
for d in random.sample(dataset_dicts, 3):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im) # Run inference
    v = Visualizer(im[:, :, ::-1],
MetadataCatalog.get("voc_2012_val"), scale=1.2)
    out = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    cv2_imshow(out.get_image()[:, :, ::-1])

```







```
%load_ext tensorboard  
%tensorboard --logdir MyVOCTraining  
<IPython.core.display.Javascript object>
```

DETR:

```
# clone detr
!git clone https://github.com/facebookresearch/detr.git
# !pip install pytorch pytorch torchvision

# download data
%cd /content/
!wget http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar
!tar -xvf VOCtrainval_11-May-2012.tar
!rm VOCtrainval_11-May-2012.tar

# This is to get all the annotation files for data format conversion.
import os
anno_root = "/content/VOCdevkit/VOC2012/Annotations"
xmls = [f for f in os.listdir(anno_root) if f.endswith(".xml")]

with open("anno_path_list.txt", "w") as f:
    for xml in xmls:
        f.write(os.path.join(anno_root, xml)+"\n")

# align the labels in VOC dataset to label index in COCO dataset.
COCO_CLASSES = [
    'N/A', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
    'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A',
    'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse',
    'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack',
    'umbrella', 'N/A', 'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis',
    'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove',
    'skateboard', 'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass',
    'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich',
    'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake',
    'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table', 'N/A',
    'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard',
    'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A',
    'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier',
    'toothbrush'
]

TRANSLATE = {
    "airplane": "aeroplane",
    "dining table": "diningtable",
    "motorcycle": "motorbike",
```

```

    "potted plant": "pottedplant",
    "couch": "sofa",
    "tv": "tvmonitor",
}

label_path = "coco_labels.txt"

i_empty = 0
with open(label_path, "w") as f:
    for i, label in enumerate(COCO_CLASSES[1:]):
        if label == "N/A":
            label = f"empty{i_empty}"
        if label in TRANSLATE:
            label = TRANSLATE[label]
        f.write(f"\n{label}\n")

!mkdir VOC_coco_format
!mkdir VOC_coco_format/train2017/
!mkdir VOC_coco_format/val2017/
!mkdir VOC_coco_format/annotations/

# Generate .json file from all the xml annotation files
# modify from https://github.com/yukkyo/voc2coco/blob/master/voc2coco.py
import os
import json
import xml.etree.ElementTree as ET
from typing import Dict, List
from tqdm import tqdm

def get_label2id(labels_path: str) -> Dict[str, int]:
    """id is 1 start"""
    with open(labels_path, 'r') as f:
        labels_str = f.read().split("\n")
    labels_ids = list(range(1, len(labels_str)+1))
    return dict(zip(labels_str, labels_ids))

def get_annpaths(ann_dir_path: str = None,
                 ann_ids_path: str = None,
                 ext: str = ".",
                 annpaths_list_path: str = None) -> List[str]:

```

```

# If use annotation paths list
if annpaths_list_path is not None:
    with open(annpaths_list_path, 'r') as f:
        ann_paths = f.read().split()
    return ann_paths

# If use annotation ids list
ext_with_dot = '.' + ext if ext != " else "
with open(ann_ids_path, 'r') as f:
    ann_ids = f.read().split()
ann_paths = [os.path.join(ann_dir_path, aid+ext_with_dot) for aid in ann_ids]
return ann_paths

def get_img_ids_from_ann_paths(ann_paths: List[str]):
    img_ids = [os.path.splitext(os.path.basename(ap))[0] for ap in ann_paths]
    num_ids = list(range(1, len(img_ids)+1))
    return dict(zip(img_ids, num_ids))

def get_image_info(annotation_root, img_id_dict=None):
    global id_set
    path = annotation_root.findtext('path')
    if path is None:
        filename = annotation_root.findtext('filename')
    else:
        filename = os.path.basename(path)
    img_name = os.path.basename(filename)
    img_id = os.path.splitext(img_name)[0]
    if img_id_dict is not None:
        assert img_id in img_id_dict, f"Error: {img_id} is not in img_id_dict !"
        img_id = img_id_dict[img_id]

    size = annotation_root.find('size')
    width = int(size.findtext('width'))
    height = int(size.findtext('height'))

    image_info = {
        'file_name': filename,
        'height': height,
        'width': width,
        'id': img_id
    }

```

```

return image_info

def get_coco_annotation_from_obj(obj, label2id):
    label = obj.findtext('name')
    assert label in label2id, f"Error: {label} is not in label2id !"
    category_id = label2id[label]
    bndbox = obj.find('bndbox')
    xmin = int(float(bndbox.findtext('xmin'))) - 1
    ymin = int(float(bndbox.findtext('ymin'))) - 1
    xmax = int(float(bndbox.findtext('xmax')))
    ymax = int(float(bndbox.findtext('ymax')))
    assert xmax > xmin and ymax > ymin, f"Box size error !: (xmin, ymin, xmax, ymax): {xmin, ymin, xmax, ymax}"
    o_width = xmax - xmin
    o_height = ymax - ymin
    ann = {
        'area': o_width * o_height,
        'iscrowd': 0,
        'bbox': [xmin, ymin, o_width, o_height],
        'category_id': category_id,
        'ignore': 0,
        'segmentation': [] # This script is not for segmentation
    }
    return ann

def convert_xmls_to_cocojson(annotation_paths: List[str],
                             label2id: Dict[str, int],
                             output_jsonpath: str,
                             extract_num_from_imgid: bool = True):
    output_json_dict = {
        "images": [],
        "type": "instances",
        "annotations": [],
        "categories": []
    }
    bnd_id = 1 # START_BOUNDING_BOX_ID, TODO input as args ?
    print('Start converting !')

    if extract_num_from_imgid:
        img_id_dict = get_img_ids_from_ann_paths(ann_paths=annotation_paths)
    else:
        img_id_dict = None

```

```

for a_path in tqdm(annotation_paths):
    # Read annotation xml
    ann_tree = ET.parse(a_path)
    ann_root = ann_tree.getroot()

    img_info = get_image_info(annotation_root=ann_root,
                             img_id_dict=img_id_dict)
    img_id = img_info['id']
    output_json_dict['images'].append(img_info)

    for obj in ann_root.findall('object'):
        ann = get_coco_annotation_from_obj(obj=obj, label2id=label2id)
        ann.update({'image_id': img_id, 'id': bnd_id})
        output_json_dict['annotations'].append(ann)
        bnd_id = bnd_id + 1

    for label, label_id in label2id.items():
        category_info = {'supercategory': 'none', 'id': label_id, 'name': label}
        output_json_dict['categories'].append(category_info)

with open(output_jsonpath, 'w') as f:
    output_json = json.dumps(output_json_dict)
    f.write(output_json)

def main():
    label2id = get_label2id(labels_path="coco_labels.txt")
    ann_paths = get_annpaths(
        ann_dir_path=None,
        ann_ids_path=None,
        ext="xml",
        annpaths_list_path="anno_path_list.txt"
    )
    convert_xmlets_to_cocojson(
        annotation_paths=ann_paths,
        label2id=label2id,
        output_jsonpath="VOC_coco_format/output.json",
        extract_num_from_imgid=True
    )

if __name__ == '__main__':
    main()

```

```
!cp VOC_coco_format/output.json VOC_coco_format/annotations/instances_train2017.json  
!cp VOC_coco_format/output.json VOC_coco_format/annotations/instances_val2017.json  
!cp /content/VOCdevkit/VOC2012/JPEGImages/* /content/VOC_coco_format/train2017/  
!cp /content/VOCdevkit/VOC2012/JPEGImages/* /content/VOC_coco_format/val2017/
```

```
%cd detr
```

```
!python main.py --batch_size 2 --resume https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth  
--coco_path /content/VOC_coco_format --output_dir ./output --lr 1e-5 --lr_backbone 1e-6  
--epochs 2
```

```
Not using distributed mode  
git:  
  sha: 29901c51d7fe8712168b8d0d64351170bc0f83e0, status: has uncommitted  
  changes, branch: main  
  
Namespace(lr=1e-05, lr_backbone=1e-06, batch_size=2, weight_decay=0.0001,  
epochs=2, lr_drop=200, clip_max_norm=0.1, frozen_weights=None,  
backbone='resnet50', dilation=False, position_embedding='sine',  
enc_layers=6, dec_layers=6, dim_feedforward=2048, hidden_dim=256,  
dropout=0.1, nheads=8, num_queries=100, pre_norm=False, masks=False,  
aux_loss=True, set_cost_class=1, set_cost_bbox=5, set_cost_giou=2,  
mask_loss_coef=1, dice_loss_coef=1, bbox_loss_coef=5, giou_loss_coef=2,  
eos_coef=0.1, dataset_file='coco', coco_path='/content/VOC_coco_format',  
coco_panoptic_path=None, remove_difficult=False, output_dir='./output',  
device='cuda', seed=42,  
resume='https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth',  
start_epoch=0, eval=False, num_workers=2, world_size=1, dist_url='env://',  
distributed=False)  
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208:  
UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may  
be removed in the future, please use 'weights' instead.  
    warnings.warn(  
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223:  
UserWarning: Arguments other than a weight enum or `None` for 'weights'  
are deprecated since 0.13 and may be removed in the future. The current  
behavior is equivalent to passing  
`weights=ResNet50_Weights.IMAGENET1K_V1`. You can also use  
`weights=ResNet50_Weights.DEFAULT` to get the most up-to-date weights.  
    warnings.warn(msg)  
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, 173MB/s]  
number of params: 41302368
```

```
loading annotations into memory...
Done (t=0.26s)
creating index...
index created!
loading annotations into memory...
Done (t=0.38s)
creating index...
index created!
Downloading: "https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth"
to /root/.cache/torch/hub/checkpoints/detr-r50-e632da11.pth
100% 159M/159M [00:00<00:00, 185MB/s]
Start training
Epoch: [ 0] [ 0/8562] eta: 9:04:51 lr: 0.000010 class_error: 0.00
loss: 4.2602 (4.2602) loss_ce: 0.3053 (0.3053) loss_bbox: 0.2069
(0.2069) loss_giou: 0.1459 (0.1459) loss_ce_0: 0.3270 (0.3270)
loss_bbox_0: 0.3560 (0.3560) loss_giou_0: 0.2422 (0.2422) loss_ce_1:
0.3632 (0.3632) loss_bbox_1: 0.2033 (0.2033) loss_giou_1: 0.1666
(0.1666) loss_ce_2: 0.3123 (0.3123) loss_bbox_2: 0.2202 (0.2202)
loss_giou_2: 0.1547 (0.1547) loss_ce_3: 0.2913 (0.2913) loss_bbox_3:
0.1790 (0.1790) loss_giou_3: 0.1352 (0.1352) loss_ce_4: 0.2924 (0.2924)
loss_bbox_4: 0.2057 (0.2057) loss_giou_4: 0.1531 (0.1531)
loss_ce_unscaled: 0.3053 (0.3053) class_error_unscaled: 0.0000 (0.0000)
loss_bbox_unscaled: 0.0414 (0.0414) loss_giou_unscaled: 0.0729 (0.0729)
cardinality_error_unscaled: 7.5000 (7.5000) loss_ce_0_unscaled: 0.3270
(0.3270) loss_bbox_0_unscaled: 0.0712 (0.0712) loss_giou_0_unscaled:
0.1211 (0.1211) cardinality_error_0_unscaled: 11.5000 (11.5000)
loss_ce_1_unscaled: 0.3632 (0.3632) loss_bbox_1_unscaled: 0.0407 (0.0407)
loss_giou_1_unscaled: 0.0833 (0.0833) cardinality_error_1_unscaled:
9.5000 (9.5000) loss_ce_2_unscaled: 0.3123 (0.3123)
loss_bbox_2_unscaled: 0.0440 (0.0440) loss_giou_2_unscaled: 0.0773
(0.0773) cardinality_error_2_unscaled: 9.0000 (9.0000)
loss_ce_3_unscaled: 0.2913 (0.2913) loss_bbox_3_unscaled: 0.0358 (0.0358)
loss_giou_3_unscaled: 0.0676 (0.0676) cardinality_error_3_unscaled:
8.5000 (8.5000) loss_ce_4_unscaled: 0.2924 (0.2924)
loss_bbox_4_unscaled: 0.0411 (0.0411) loss_giou_4_unscaled: 0.0765
(0.0765) cardinality_error_4_unscaled: 7.5000 (7.5000) time: 3.8182
data: 0.4076 max mem: 1804
Epoch: [ 0] [ 10/8562] eta: 1:32:46 lr: 0.000010 class_error: 33.33
loss: 2.7564 (3.5888) loss_ce: 0.1077 (0.2120) loss_bbox: 0.1718
(0.1918) loss_giou: 0.1459 (0.1762) loss_ce_0: 0.2069 (0.2348)
loss_bbox_0: 0.1728 (0.2265) loss_giou_0: 0.1740 (0.2156) loss_ce_1:
0.2095 (0.2349) loss_bbox_1: 0.1572 (0.1918) loss_giou_1: 0.1666
(0.1965) loss_ce_2: 0.2014 (0.2165) loss_bbox_2: 0.1638 (0.1894)
loss_giou_2: 0.1465 (0.1830) loss_ce_3: 0.1867 (0.2067) loss_bbox_3:
0.1467 (0.1747) loss_giou_3: 0.1352 (0.1750) loss_ce_4: 0.1094 (0.1961)
loss_bbox_4: 0.1679 (0.1907) loss_giou_4: 0.1531 (0.1767)
loss_ce_unscaled: 0.1077 (0.2120) class_error_unscaled: 0.0000 (20.7071)
```

```
loss_bbox_unscaled: 0.0344 (0.0384) loss_giou_unscaled: 0.0729 (0.0881)
cardinality_error_unscaled: 3.0000 (3.6364) loss_ce_0_unscaled: 0.2069
(0.2348) loss_bbox_0_unscaled: 0.0346 (0.0453) loss_giou_0_unscaled:
0.0870 (0.1078) cardinality_error_0_unscaled: 4.0000 (5.3182)
loss_ce_1_unscaled: 0.2095 (0.2349) loss_bbox_1_unscaled: 0.0314 (0.0384)
loss_giou_1_unscaled: 0.0833 (0.0983) cardinality_error_1_unscaled:
3.0000 (4.0909) loss_ce_2_unscaled: 0.2014 (0.2165)
loss_bbox_2_unscaled: 0.0328 (0.0379) loss_giou_2_unscaled: 0.0733
(0.0915) cardinality_error_2_unscaled: 3.0000 (3.8182)
loss_ce_3_unscaled: 0.1867 (0.2067) loss_bbox_3_unscaled: 0.0293 (0.0349)
loss_giou_3_unscaled: 0.0676 (0.0875) cardinality_error_3_unscaled:
2.5000 (3.7273) loss_ce_4_unscaled: 0.1094 (0.1961)
loss_bbox_4_unscaled: 0.0336 (0.0381) loss_giou_4_unscaled: 0.0765
(0.0884) cardinality_error_4_unscaled: 2.5000 (3.5455) time: 0.6509
data: 0.0506 max mem: 2192
Epoch: [0] [ 20/8562] eta: 1:13:42 lr: 0.000010 class_error: 33.33
loss: 3.5930 (4.1941) loss_ce: 0.1503 (0.2382) loss_bbox: 0.1879
(0.2280) loss_giou: 0.1508 (0.2089) loss_ce_0: 0.2036 (0.2674)
loss_bbox_0: 0.2485 (0.2579) loss_giou_0: 0.2123 (0.2573) loss_ce_1:
0.1810 (0.2596) loss_bbox_1: 0.2005 (0.2228) loss_giou_1: 0.1703
(0.2230) loss_ce_2: 0.1624 (0.2434) loss_bbox_2: 0.2053 (0.2292)
loss_giou_2: 0.1808 (0.2175) loss_ce_3: 0.1712 (0.2344) loss_bbox_3:
0.1992 (0.2266) loss_giou_3: 0.1684 (0.2163) loss_ce_4: 0.1353 (0.2347)
loss_bbox_4: 0.1771 (0.2229) loss_giou_4: 0.1645 (0.2060)
loss_ce_unscaled: 0.1503 (0.2382) class_error_unscaled: 22.2222 (21.2188)
loss_bbox_unscaled: 0.0376 (0.0456) loss_giou_unscaled: 0.0754 (0.1044)
cardinality_error_unscaled: 2.0000 (4.0714) loss_ce_0_unscaled: 0.2036
(0.2674) loss_bbox_0_unscaled: 0.0497 (0.0516) loss_giou_0_unscaled:
0.1061 (0.1287) cardinality_error_0_unscaled: 3.0000 (5.5476)
loss_ce_1_unscaled: 0.1810 (0.2596) loss_bbox_1_unscaled: 0.0401 (0.0446)
loss_giou_1_unscaled: 0.0851 (0.1115) cardinality_error_1_unscaled:
2.5000 (4.6905) loss_ce_2_unscaled: 0.1624 (0.2434)
loss_bbox_2_unscaled: 0.0411 (0.0458) loss_giou_2_unscaled: 0.0904
(0.1088) cardinality_error_2_unscaled: 2.0000 (3.9524)
loss_ce_3_unscaled: 0.1712 (0.2344) loss_bbox_3_unscaled: 0.0398 (0.0453)
loss_giou_3_unscaled: 0.0842 (0.1081) cardinality_error_3_unscaled:
2.0000 (4.0238) loss_ce_4_unscaled: 0.1353 (0.2347)
loss_bbox_4_unscaled: 0.0354 (0.0446) loss_giou_4_unscaled: 0.0823
(0.1030) cardinality_error_4_unscaled: 2.0000 (3.8571) time: 0.3527
data: 0.0178 max mem: 3212
Epoch: [0] [ 30/8562] eta: 1:02:55 lr: 0.000010 class_error: 0.00
loss: 3.6990 (4.6104) loss_ce: 0.3055 (0.3118) loss_bbox: 0.1879
(0.2095) loss_giou: 0.1769 (0.2086) loss_ce_0: 0.3473 (0.3451)
loss_bbox_0: 0.2468 (0.2640) loss_giou_0: 0.2197 (0.2608) loss_ce_1:
0.3316 (0.3472) loss_bbox_1: 0.2005 (0.2181) loss_giou_1: 0.1740
(0.2240) loss_ce_2: 0.2626 (0.3200) loss_bbox_2: 0.2053 (0.2141)
loss_giou_2: 0.1862 (0.2205) loss_ce_3: 0.2580 (0.3159) loss_bbox_3:
```

```
0.1889 (0.2099) loss_giou_3: 0.1723 (0.2154) loss_ce_4: 0.3004 (0.3125)
loss_bbox_4: 0.1672 (0.2060) loss_giou_4: 0.1780 (0.2071)
loss_ce_unscaled: 0.3055 (0.3118) class_error_unscaled: 25.0000 (24.1283)
loss_bbox_unscaled: 0.0376 (0.0419) loss_giou_unscaled: 0.0885 (0.1043)
cardinality_error_unscaled: 2.0000 (4.5484) loss_ce_0_unscaled: 0.3473
(0.3451) loss_bbox_0_unscaled: 0.0494 (0.0528) loss_giou_0_unscaled:
0.1099 (0.1304) cardinality_error_0_unscaled: 3.0000 (6.4677)
loss_ce_1_unscaled: 0.3316 (0.3472) loss_bbox_1_unscaled: 0.0401 (0.0436)
loss_giou_1_unscaled: 0.0870 (0.1120) cardinality_error_1_unscaled:
3.0000 (5.3871) loss_ce_2_unscaled: 0.2626 (0.3200)
loss_bbox_2_unscaled: 0.0411 (0.0428) loss_giou_2_unscaled: 0.0931
(0.1102) cardinality_error_2_unscaled: 2.0000 (4.6774)
loss_ce_3_unscaled: 0.2580 (0.3159) loss_bbox_3_unscaled: 0.0378 (0.0420)
loss_giou_3_unscaled: 0.0862 (0.1077) cardinality_error_3_unscaled:
2.0000 (4.6935) loss_ce_4_unscaled: 0.3004 (0.3125)
loss_bbox_4_unscaled: 0.0334 (0.0412) loss_giou_4_unscaled: 0.0890
(0.1035) cardinality_error_4_unscaled: 2.5000 (4.3548) time: 0.3279
data: 0.0165 max mem: 3212
Epoch: [0] [ 40/8562] eta: 0:58:33 lr: 0.000010 class_error: 0.00
loss: 3.6021 (4.3107) loss_ce: 0.2010 (0.2734) loss_bbox: 0.1294
(0.2085) loss_giou: 0.1721 (0.1992) loss_ce_0: 0.2634 (0.3056)
loss_bbox_0: 0.1818 (0.2553) loss_giou_0: 0.1919 (0.2494) loss_ce_1:
0.2899 (0.2986) loss_bbox_1: 0.1468 (0.2285) loss_giou_1: 0.1651
(0.2224) loss_ce_2: 0.1979 (0.2830) loss_bbox_2: 0.1636 (0.2136)
loss_giou_2: 0.1770 (0.2086) loss_ce_3: 0.2058 (0.2797) loss_bbox_3:
0.1436 (0.2051) loss_giou_3: 0.1473 (0.2038) loss_ce_4: 0.2138 (0.2740)
loss_bbox_4: 0.1421 (0.2045) loss_giou_4: 0.1780 (0.1975)
loss_ce_unscaled: 0.2010 (0.2734) class_error_unscaled: 0.0000 (19.4048)
loss_bbox_unscaled: 0.0259 (0.0417) loss_giou_unscaled: 0.0860 (0.0996)
cardinality_error_unscaled: 2.5000 (4.4878) loss_ce_0_unscaled: 0.2634
(0.3056) loss_bbox_0_unscaled: 0.0364 (0.0511) loss_giou_0_unscaled:
0.0960 (0.1247) cardinality_error_0_unscaled: 4.5000 (6.3293)
loss_ce_1_unscaled: 0.2899 (0.2986) loss_bbox_1_unscaled: 0.0294 (0.0457)
loss_giou_1_unscaled: 0.0826 (0.1112) cardinality_error_1_unscaled:
3.5000 (5.1707) loss_ce_2_unscaled: 0.1979 (0.2830)
loss_bbox_2_unscaled: 0.0327 (0.0427) loss_giou_2_unscaled: 0.0885
(0.1043) cardinality_error_2_unscaled: 3.5000 (4.6341)
loss_ce_3_unscaled: 0.2058 (0.2797) loss_bbox_3_unscaled: 0.0287 (0.0410)
loss_giou_3_unscaled: 0.0736 (0.1019) cardinality_error_3_unscaled:
3.5000 (4.6341) loss_ce_4_unscaled: 0.2138 (0.2740)
loss_bbox_4_unscaled: 0.0284 (0.0409) loss_giou_4_unscaled: 0.0890
(0.0988) cardinality_error_4_unscaled: 2.5000 (4.3049) time: 0.3017
data: 0.0129 max mem: 3212
Epoch: [0] [ 50/8562] eta: 0:56:45 lr: 0.000010 class_error: 0.00
loss: 3.1344 (4.3402) loss_ce: 0.1094 (0.2677) loss_bbox: 0.1385
(0.2135) loss_giou: 0.1434 (0.1992) loss_ce_0: 0.1352 (0.2968)
loss_bbox_0: 0.2119 (0.2728) loss_giou_0: 0.1919 (0.2471) loss_ce_1:
```

```
0.1158 (0.2908) loss_bbox_1: 0.1848 (0.2404) loss_giou_1: 0.1651  
(0.2237) loss_ce_2: 0.1098 (0.2771) loss_bbox_2: 0.1687 (0.2282)  
loss_giou_2: 0.1625 (0.2109) loss_ce_3: 0.1195 (0.2745) loss_bbox_3:  
0.1607 (0.2138) loss_giou_3: 0.1632 (0.2051) loss_ce_4: 0.1143 (0.2681)  
loss_bbox_4: 0.1421 (0.2127) loss_giou_4: 0.1487 (0.1980)  
loss_ce_unscaled: 0.1094 (0.2677) class_error_unscaled: 0.0000 (19.3581)  
loss_bbox_unscaled: 0.0277 (0.0427) loss_giou_unscaled: 0.0717 (0.0996)  
cardinality_error_unscaled: 1.5000 (4.1667) loss_ce_0_unscaled: 0.1352  
(0.2968) loss_bbox_0_unscaled: 0.0424 (0.0546) loss_giou_0_unscaled:  
0.0960 (0.1235) cardinality_error_0_unscaled: 2.5000 (5.8137)  
loss_ce_1_unscaled: 0.1158 (0.2908) loss_bbox_1_unscaled: 0.0370 (0.0481)  
loss_giou_1_unscaled: 0.0826 (0.1118) cardinality_error_1_unscaled:  
1.5000 (4.7451) loss_ce_2_unscaled: 0.1098 (0.2771)  
loss_bbox_2_unscaled: 0.0337 (0.0456) loss_giou_2_unscaled: 0.0813  
(0.1054) cardinality_error_2_unscaled: 1.5000 (4.2647)  
loss_ce_3_unscaled: 0.1195 (0.2745) loss_bbox_3_unscaled: 0.0321 (0.0428)  
loss_giou_3_unscaled: 0.0816 (0.1026) cardinality_error_3_unscaled:  
1.5000 (4.2745) loss_ce_4_unscaled: 0.1143 (0.2681)  
loss_bbox_4_unscaled: 0.0284 (0.0425) loss_giou_4_unscaled: 0.0744  
(0.0990) cardinality_error_4_unscaled: 1.5000 (4.0294) time: 0.3341  
data: 0.0139 max mem: 3212  
Epoch: [0] [ 60/8562] eta: 0:56:02 lr: 0.000010 class_error: 14.29  
loss: 3.2288 (4.5572) loss_ce: 0.1720 (0.2909) loss_bbox: 0.1621  
(0.2220) loss_giou: 0.1967 (0.2063) loss_ce_0: 0.2470 (0.3204)  
loss_bbox_0: 0.2119 (0.2830) loss_giou_0: 0.2209 (0.2543) loss_ce_1:  
0.2089 (0.3106) loss_bbox_1: 0.1848 (0.2494) loss_giou_1: 0.1904  
(0.2301) loss_ce_2: 0.2236 (0.3021) loss_bbox_2: 0.1607 (0.2309)  
loss_giou_2: 0.1771 (0.2157) loss_ce_3: 0.1954 (0.2969) loss_bbox_3:  
0.1610 (0.2205) loss_giou_3: 0.2072 (0.2100) loss_ce_4: 0.2092 (0.2917)  
loss_bbox_4: 0.1468 (0.2190) loss_giou_4: 0.2000 (0.2034)  
loss_ce_unscaled: 0.1720 (0.2909) class_error_unscaled: 16.6667 (21.9744)  
loss_bbox_unscaled: 0.0324 (0.0444) loss_giou_unscaled: 0.0983 (0.1032)  
cardinality_error_unscaled: 1.0000 (4.0984) loss_ce_0_unscaled: 0.2470  
(0.3204) loss_bbox_0_unscaled: 0.0424 (0.0566) loss_giou_0_unscaled:  
0.1104 (0.1271) cardinality_error_0_unscaled: 1.5000 (5.7787)  
loss_ce_1_unscaled: 0.2089 (0.3106) loss_bbox_1_unscaled: 0.0370 (0.0499)  
loss_giou_1_unscaled: 0.0952 (0.1150) cardinality_error_1_unscaled:  
1.5000 (4.6475) loss_ce_2_unscaled: 0.2236 (0.3021)  
loss_bbox_2_unscaled: 0.0321 (0.0462) loss_giou_2_unscaled: 0.0885  
(0.1078) cardinality_error_2_unscaled: 1.0000 (4.2377)  
loss_ce_3_unscaled: 0.1954 (0.2969) loss_bbox_3_unscaled: 0.0322 (0.0441)  
loss_giou_3_unscaled: 0.1036 (0.1050) cardinality_error_3_unscaled:  
1.0000 (4.1967) loss_ce_4_unscaled: 0.2092 (0.2917)  
loss_bbox_4_unscaled: 0.0294 (0.0438) loss_giou_4_unscaled: 0.1000  
(0.1017) cardinality_error_4_unscaled: 1.5000 (3.9754) time: 0.3608  
data: 0.0188 max mem: 3212
```

```
Epoch: [0]  [ 70/8562]  eta: 0:54:06  lr: 0.000010  class_error: 0.00
loss: 3.5267 (4.5143)  loss_ce: 0.1964 (0.2829)  loss_bbox: 0.1777
(0.2192)  loss_giou: 0.1967 (0.2076)  loss_ce_0: 0.2504 (0.3135)
loss_bbox_0: 0.2044 (0.2821)  loss_giou_0: 0.2282 (0.2563)  loss_ce_1:
0.2282 (0.3017)  loss_bbox_1: 0.2042 (0.2525)  loss_giou_1: 0.2014
(0.2335)  loss_ce_2: 0.2434 (0.2939)  loss_bbox_2: 0.2127 (0.2290)
loss_giou_2: 0.1771 (0.2177)  loss_ce_3: 0.2289 (0.2886)  loss_bbox_3:
0.2016 (0.2181)  loss_giou_3: 0.2072 (0.2117)  loss_ce_4: 0.2193 (0.2844)
loss_bbox_4: 0.1659 (0.2170)  loss_giou_4: 0.2000 (0.2047)
loss_ce_unscaled: 0.1964 (0.2829)  class_error_unscaled: 25.0000 (21.8640)
loss_bbox_unscaled: 0.0355 (0.0438)  loss_giou_unscaled: 0.0983 (0.1038)
cardinality_error_unscaled: 2.0000 (3.9789)  loss_ce_0_unscaled: 0.2504
(0.3135)  loss_bbox_0_unscaled: 0.0409 (0.0564)  loss_giou_0_unscaled:
0.1141 (0.1282)  cardinality_error_0_unscaled: 3.0000 (5.6127)
loss_ce_1_unscaled: 0.2282 (0.3017)  loss_bbox_1_unscaled: 0.0408 (0.0505)
loss_giou_1_unscaled: 0.1007 (0.1167)  cardinality_error_1_unscaled:
2.0000 (4.4930)  loss_ce_2_unscaled: 0.2434 (0.2939)
loss_bbox_2_unscaled: 0.0425 (0.0458)  loss_giou_2_unscaled: 0.0885
(0.1088)  cardinality_error_2_unscaled: 2.0000 (4.1620)
loss_ce_3_unscaled: 0.2289 (0.2886)  loss_bbox_3_unscaled: 0.0403 (0.0436)
loss_giou_3_unscaled: 0.1036 (0.1058)  cardinality_error_3_unscaled:
2.0000 (4.0704)  loss_ce_4_unscaled: 0.2193 (0.2844)
loss_bbox_4_unscaled: 0.0332 (0.0434)  loss_giou_4_unscaled: 0.1000
(0.1023)  cardinality_error_4_unscaled: 2.0000 (3.8873)  time: 0.3371
data: 0.0186  max mem: 3338
Epoch: [0]  [ 80/8562]  eta: 0:52:21  lr: 0.000010  class_error: 20.00
loss: 3.4347 (4.3782)  loss_ce: 0.1494 (0.2694)  loss_bbox: 0.2055
(0.2157)  loss_giou: 0.1435 (0.2017)  loss_ce_0: 0.2504 (0.3062)
loss_bbox_0: 0.2146 (0.2752)  loss_giou_0: 0.1825 (0.2495)  loss_ce_1:
0.1737 (0.2864)  loss_bbox_1: 0.2141 (0.2501)  loss_giou_1: 0.1904
(0.2281)  loss_ce_2: 0.1856 (0.2803)  loss_bbox_2: 0.2150 (0.2238)
loss_giou_2: 0.1426 (0.2117)  loss_ce_3: 0.1885 (0.2779)  loss_bbox_3:
0.2038 (0.2135)  loss_giou_3: 0.1564 (0.2058)  loss_ce_4: 0.1878 (0.2723)
loss_bbox_4: 0.2108 (0.2121)  loss_giou_4: 0.1454 (0.1987)
loss_ce_unscaled: 0.1494 (0.2694)  class_error_unscaled: 20.0000 (21.5927)
loss_bbox_unscaled: 0.0411 (0.0431)  loss_giou_unscaled: 0.0717 (0.1008)
cardinality_error_unscaled: 2.0000 (4.0309)  loss_ce_0_unscaled: 0.2504
(0.3062)  loss_bbox_0_unscaled: 0.0429 (0.0550)  loss_giou_0_unscaled:
0.0912 (0.1248)  cardinality_error_0_unscaled: 4.0000 (5.6543)
loss_ce_1_unscaled: 0.1737 (0.2864)  loss_bbox_1_unscaled: 0.0428 (0.0500)
loss_giou_1_unscaled: 0.0952 (0.1140)  cardinality_error_1_unscaled:
2.0000 (4.5617)  loss_ce_2_unscaled: 0.1856 (0.2803)
loss_bbox_2_unscaled: 0.0430 (0.0448)  loss_giou_2_unscaled: 0.0713
(0.1059)  cardinality_error_2_unscaled: 2.0000 (4.2346)
loss_ce_3_unscaled: 0.1885 (0.2779)  loss_bbox_3_unscaled: 0.0408 (0.0427)
loss_giou_3_unscaled: 0.0782 (0.1029)  cardinality_error_3_unscaled:
2.0000 (4.1358)  loss_ce_4_unscaled: 0.1878 (0.2723)
```

```
loss bbox_4 unscaled: 0.0422 (0.0424) loss giou 4 unscaled: 0.0727  
(0.0993) cardinality error 4 unscaled: 2.5000 (3.9753) time: 0.2939  
data: 0.0130 max mem: 3338  
Epoch: [0] [ 90/8562] eta: 0:51:43 lr: 0.000010 class error: 33.33  
loss: 3.4347 (4.3557) loss_ce: 0.1772 (0.2669) loss_bbox: 0.1802  
(0.2138) loss_giou: 0.1748 (0.2031) loss_ce_0: 0.2203 (0.3029)  
loss_bbox_0: 0.2293 (0.2761) loss_giou_0: 0.2076 (0.2528) loss_ce_1:  
0.1737 (0.2860) loss_bbox_1: 0.1644 (0.2439) loss_giou_1: 0.1905  
(0.2276) loss_ce_2: 0.1721 (0.2759) loss_bbox_2: 0.1808 (0.2216)  
loss_giou_2: 0.1732 (0.2129) loss_ce_3: 0.1885 (0.2746) loss_bbox_3:  
0.1778 (0.2117) loss_giou_3: 0.1808 (0.2072) loss_ce_4: 0.1878 (0.2678)  
loss_bbox_4: 0.1719 (0.2105) loss_giou_4: 0.1676 (0.2005)  
loss_ce_unscaled: 0.1772 (0.2669) class_error_unscaled: 20.0000 (22.0312)  
loss_bbox_unscaled: 0.0360 (0.0428) loss_giou_unscaled: 0.0874 (0.1015)  
cardinality_error_unscaled: 2.0000 (3.8846) loss_ce_0_unscaled: 0.2203  
(0.3029) loss_bbox_0_unscaled: 0.0459 (0.0552) loss_giou_0_unscaled:  
0.1038 (0.1264) cardinality_error_0_unscaled: 2.5000 (5.4121)  
loss_ce_1_unscaled: 0.1737 (0.2860) loss_bbox_1_unscaled: 0.0329 (0.0488)  
loss_giou_1_unscaled: 0.0953 (0.1138) cardinality_error_1_unscaled:  
2.0000 (4.3901) loss_ce_2_unscaled: 0.1721 (0.2759)  
loss_bbox_2_unscaled: 0.0362 (0.0443) loss_giou_2_unscaled: 0.0866  
(0.1065) cardinality_error_2_unscaled: 2.0000 (4.1099)  
loss_ce_3_unscaled: 0.1885 (0.2746) loss_bbox_3_unscaled: 0.0356 (0.0423)  
loss_giou_3_unscaled: 0.0904 (0.1036) cardinality_error_3_unscaled:  
2.0000 (4.0165) loss_ce_4_unscaled: 0.1878 (0.2678)  
loss_bbox_4_unscaled: 0.0344 (0.0421) loss_giou_4_unscaled: 0.0838  
(0.1002) cardinality_error_4_unscaled: 2.0000 (3.8462) time: 0.3099  
data: 0.0126 max mem: 3338  
Epoch: [0] [ 100/8562] eta: 0:51:45 lr: 0.000010 class error: 0.00  
loss: 3.5832 (4.2961) loss_ce: 0.1983 (0.2626) loss_bbox: 0.1690  
(0.2133) loss_giou: 0.1317 (0.1971) loss_ce_0: 0.2290 (0.2944)  
loss_bbox_0: 0.1946 (0.2762) loss_giou_0: 0.2076 (0.2449) loss_ce_1:  
0.1914 (0.2754) loss_bbox_1: 0.1644 (0.2505) loss_giou_1: 0.1582  
(0.2225) loss_ce_2: 0.1933 (0.2657) loss_bbox_2: 0.1606 (0.2261)  
loss_giou_2: 0.1528 (0.2079) loss_ce_3: 0.1794 (0.2637) loss_bbox_3:  
0.1803 (0.2217) loss_giou_3: 0.1808 (0.2035) loss_ce_4: 0.1816 (0.2587)  
loss_bbox_4: 0.1776 (0.2160) loss_giou_4: 0.1676 (0.1960)  
loss_ce_unscaled: 0.1983 (0.2626) class_error_unscaled: 12.5000 (21.5001)  
loss_bbox_unscaled: 0.0338 (0.0427) loss_giou_unscaled: 0.0659 (0.0985)  
cardinality_error_unscaled: 1.5000 (3.8119) loss_ce_0_unscaled: 0.2290  
(0.2944) loss_bbox_0_unscaled: 0.0389 (0.0552) loss_giou_0_unscaled:  
0.1038 (0.1224) cardinality_error_0_unscaled: 2.0000 (5.2277)  
loss_ce_1_unscaled: 0.1914 (0.2754) loss_bbox_1_unscaled: 0.0329 (0.0501)  
loss_giou_1_unscaled: 0.0791 (0.1113) cardinality_error_1_unscaled:  
1.5000 (4.2822) loss_ce_2_unscaled: 0.1933 (0.2657)  
loss_bbox_2_unscaled: 0.0321 (0.0452) loss_giou_2_unscaled: 0.0764  
(0.1040) cardinality_error_2_unscaled: 1.5000 (4.0297)
```

```
loss_ce_3_unscaled: 0.1794 (0.2637) loss_bbox_3_unscaled: 0.0361 (0.0443)
loss_giou_3_unscaled: 0.0904 (0.1018) cardinality_error_3_unscaled:
2.0000 (3.9356) loss_ce_4_unscaled: 0.1816 (0.2587)
loss_bbox_4_unscaled: 0.0355 (0.0432) loss_giou_4_unscaled: 0.0838
(0.0980) cardinality_error_4_unscaled: 2.0000 (3.7772) time: 0.3535
data: 0.0167 max mem: 3338
Epoch: [0] [ 110/8562] eta: 0:51:29 lr: 0.000010 class_error: 33.33
loss: 3.2804 (4.2654) loss_ce: 0.2154 (0.2582) loss_bbox: 0.1618
(0.2115) loss_giou: 0.1317 (0.1963) loss_ce_0: 0.2201 (0.2879)
loss_bbox_0: 0.1946 (0.2789) loss_giou_0: 0.1689 (0.2472) loss_ce_1:
0.1666 (0.2680) loss_bbox_1: 0.1904 (0.2522) loss_giou_1: 0.1361
(0.2235) loss_ce_2: 0.1869 (0.2614) loss_bbox_2: 0.1615 (0.2245)
loss_giou_2: 0.1296 (0.2071) loss_ce_3: 0.1794 (0.2598) loss_bbox_3:
0.1926 (0.2206) loss_giou_3: 0.1671 (0.2032) loss_ce_4: 0.1561 (0.2555)
loss_bbox_4: 0.1776 (0.2141) loss_giou_4: 0.1625 (0.1955)
loss_ce_unscaled: 0.2154 (0.2582) class_error_unscaled: 10.0000 (21.6102)
loss_bbox_unscaled: 0.0324 (0.0423) loss_giou_unscaled: 0.0659 (0.0981)
cardinality_error_unscaled: 1.5000 (3.7117) loss_ce_0_unscaled: 0.2201
(0.2879) loss_bbox_0_unscaled: 0.0389 (0.0558) loss_giou_0_unscaled:
0.0845 (0.1236) cardinality_error_0_unscaled: 1.5000 (5.0856)
loss_ce_1_unscaled: 0.1666 (0.2680) loss_bbox_1_unscaled: 0.0381 (0.0504)
loss_giou_1_unscaled: 0.0681 (0.1117) cardinality_error_1_unscaled:
1.5000 (4.1712) loss_ce_2_unscaled: 0.1869 (0.2614)
loss_bbox_2_unscaled: 0.0323 (0.0449) loss_giou_2_unscaled: 0.0648
(0.1035) cardinality_error_2_unscaled: 1.5000 (3.9144)
loss_ce_3_unscaled: 0.1794 (0.2598) loss_bbox_3_unscaled: 0.0385 (0.0441)
loss_giou_3_unscaled: 0.0835 (0.1016) cardinality_error_3_unscaled:
1.5000 (3.8288) loss_ce_4_unscaled: 0.1561 (0.2555)
loss_bbox_4_unscaled: 0.0355 (0.0428) loss_giou_4_unscaled: 0.0812
(0.0978) cardinality_error_4_unscaled: 1.5000 (3.6757) time: 0.3616
data: 0.0175 max mem: 3338
Epoch: [0] [ 120/8562] eta: 0:50:44 lr: 0.000010 class_error: 14.29
loss: 3.8373 (4.1767) loss_ce: 0.1408 (0.2484) loss_bbox: 0.1407
(0.2078) loss_giou: 0.1630 (0.1948) loss_ce_0: 0.1531 (0.2807)
loss_bbox_0: 0.2059 (0.2733) loss_giou_0: 0.2095 (0.2443) loss_ce_1:
0.1327 (0.2593) loss_bbox_1: 0.2060 (0.2475) loss_giou_1: 0.1684
(0.2216) loss_ce_2: 0.1360 (0.2519) loss_bbox_2: 0.1624 (0.2210)
loss_giou_2: 0.1870 (0.2061) loss_ce_3: 0.1408 (0.2503) loss_bbox_3:
0.1675 (0.2169) loss_giou_3: 0.1686 (0.2015) loss_ce_4: 0.1401 (0.2460)
loss_bbox_4: 0.1412 (0.2107) loss_giou_4: 0.1665 (0.1943)
loss_ce_unscaled: 0.1408 (0.2484) class_error_unscaled: 0.0000 (20.5621)
loss_bbox_unscaled: 0.0281 (0.0416) loss_giou_unscaled: 0.0815 (0.0974)
cardinality_error_unscaled: 2.5000 (3.6942) loss_ce_0_unscaled: 0.1531
(0.2807) loss_bbox_0_unscaled: 0.0412 (0.0547) loss_giou_0_unscaled:
0.1048 (0.1222) cardinality_error_0_unscaled: 2.0000 (5.0372)
loss_ce_1_unscaled: 0.1327 (0.2593) loss_bbox_1_unscaled: 0.0412 (0.0495)
loss_giou_1_unscaled: 0.0842 (0.1108) cardinality_error_1_unscaled:
```

```
1.5000 (4.1198) loss_ce_2_unscaled: 0.1360 (0.2519)
loss_bbox_2_unscaled: 0.0325 (0.0442) loss_giou_2_unscaled: 0.0935
(0.1031) cardinality_error_2_unscaled: 2.0000 (3.8802)
loss_ce_3_unscaled: 0.1408 (0.2503) loss_bbox_3_unscaled: 0.0335 (0.0434)
loss_giou_3_unscaled: 0.0843 (0.1007) cardinality_error_3_unscaled:
2.0000 (3.8264) loss_ce_4_unscaled: 0.1401 (0.2460)
loss_bbox_4_unscaled: 0.0282 (0.0421) loss_giou_4_unscaled: 0.0832
(0.0971) cardinality_error_4_unscaled: 2.5000 (3.6736) time: 0.3283
data: 0.0136 max mem: 3338
Epoch: [0] [ 130/8562] eta: 0:49:50 lr: 0.000010 class error: 66.67
loss: 3.9698 (4.4127) loss_ce: 0.1592 (0.2652) loss_bbox: 0.2116
(0.2203) loss_giou: 0.1922 (0.2067) loss_ce_0: 0.1667 (0.2956)
loss_bbox_0: 0.2526 (0.2824) loss_giou_0: 0.2754 (0.2556) loss_ce_1:
0.1646 (0.2779) loss_bbox_1: 0.2374 (0.2552) loss_giou_1: 0.2393
(0.2330) loss_ce_2: 0.1705 (0.2665) loss_bbox_2: 0.2247 (0.2322)
loss_giou_2: 0.2660 (0.2186) loss_ce_3: 0.1582 (0.2651) loss_bbox_3:
0.2135 (0.2298) loss_giou_3: 0.2475 (0.2144) loss_ce_4: 0.1596 (0.2622)
loss_bbox_4: 0.2143 (0.2243) loss_giou_4: 0.2381 (0.2078)
loss_ce_unscaled: 0.1592 (0.2652) class_error_unscaled: 20.0000 (21.4878)
loss_bbox_unscaled: 0.0423 (0.0441) loss_giou_unscaled: 0.0961 (0.1033)
cardinality_error_unscaled: 2.5000 (3.7366) loss_ce_0_unscaled: 0.1667
(0.2956) loss_bbox_0_unscaled: 0.0505 (0.0565) loss_giou_0_unscaled:
0.1377 (0.1278) cardinality_error_0_unscaled: 3.5000 (5.1450)
loss_ce_1_unscaled: 0.1646 (0.2779) loss_bbox_1_unscaled: 0.0475 (0.0510)
loss_giou_1_unscaled: 0.1197 (0.1165) cardinality_error_1_unscaled:
2.5000 (4.1870) loss_ce_2_unscaled: 0.1705 (0.2665)
loss_bbox_2_unscaled: 0.0449 (0.0464) loss_giou_2_unscaled: 0.1330
(0.1093) cardinality_error_2_unscaled: 2.0000 (3.9275)
loss_ce_3_unscaled: 0.1582 (0.2651) loss_bbox_3_unscaled: 0.0427 (0.0460)
loss_giou_3_unscaled: 0.1237 (0.1072) cardinality_error_3_unscaled:
2.0000 (3.8664) loss_ce_4_unscaled: 0.1596 (0.2622)
loss_bbox_4_unscaled: 0.0429 (0.0449) loss_giou_4_unscaled: 0.1191
(0.1039) cardinality_error_4_unscaled: 2.5000 (3.7137) time: 0.2941
data: 0.0122 max mem: 3338
Epoch: [0] [ 140/8562] eta: 0:49:43 lr: 0.000010 class error: 0.00
loss: 4.0819 (4.3542) loss_ce: 0.2204 (0.2591) loss_bbox: 0.1851
(0.2174) loss_giou: 0.2102 (0.2057) loss_ce_0: 0.2767 (0.2902)
loss_bbox_0: 0.2711 (0.2784) loss_giou_0: 0.2966 (0.2553) loss_ce_1:
0.2384 (0.2719) loss_bbox_1: 0.2096 (0.2513) loss_giou_1: 0.2193
(0.2321) loss_ce_2: 0.2315 (0.2620) loss_bbox_2: 0.1888 (0.2286)
loss_giou_2: 0.2248 (0.2172) loss_ce_3: 0.2292 (0.2601) loss_bbox_3:
0.1895 (0.2268) loss_giou_3: 0.2216 (0.2132) loss_ce_4: 0.1993 (0.2564)
loss_bbox_4: 0.2044 (0.2215) loss_giou_4: 0.2161 (0.2070)
loss_ce_unscaled: 0.2204 (0.2591) class_error_unscaled: 20.0000 (20.5880)
loss_bbox_unscaled: 0.0370 (0.0435) loss_giou_unscaled: 0.1051 (0.1029)
cardinality_error_unscaled: 1.0000 (3.8262) loss_ce_0_unscaled: 0.2767
(0.2902) loss_bbox_0_unscaled: 0.0542 (0.0557) loss_giou_0_unscaled:
```

```
0.1483 (0.1276) cardinality_error_0_unscaled: 2.0000 (5.1950)
loss_ce_1_unscaled: 0.2384 (0.2719) loss_bbox_1_unscaled: 0.0419 (0.0503)
loss_giou_1_unscaled: 0.1096 (0.1160) cardinality_error_1_unscaled:
1.0000 (4.2624) loss_ce_2_unscaled: 0.2315 (0.2620)
loss_bbox_2_unscaled: 0.0378 (0.0457) loss_giou_2_unscaled: 0.1124
(0.1086) cardinality_error_2_unscaled: 1.0000 (4.0496)
loss_ce_3_unscaled: 0.2292 (0.2601) loss_bbox_3_unscaled: 0.0379 (0.0454)
loss_giou_3_unscaled: 0.1108 (0.1066) cardinality_error_3_unscaled:
1.5000 (4.0035) loss_ce_4_unscaled: 0.1993 (0.2564)
loss_bbox_4_unscaled: 0.0409 (0.0443) loss_giou_4_unscaled: 0.1081
(0.1035) cardinality_error_4_unscaled: 1.5000 (3.8191) time: 0.3159
data: 0.0147 max mem: 3338
Epoch: [0] [ 150/8562] eta: 0:49:11 lr: 0.000010 class_error: 50.00
loss: 3.6109 (4.3254) loss_ce: 0.1172 (0.2537) loss_bbox: 0.1616
(0.2164) loss_giou: 0.1652 (0.2064) loss_ce_0: 0.1997 (0.2849)
loss_bbox_0: 0.1887 (0.2745) loss_giou_0: 0.2263 (0.2546) loss_ce_1:
0.1390 (0.2664) loss_bbox_1: 0.1794 (0.2512) loss_giou_1: 0.1729
(0.2330) loss_ce_2: 0.1721 (0.2571) loss_bbox_2: 0.1670 (0.2289)
loss_giou_2: 0.1872 (0.2179) loss_ce_3: 0.1161 (0.2561) loss_bbox_3:
0.1634 (0.2273) loss_giou_3: 0.1662 (0.2146) loss_ce_4: 0.1227 (0.2516)
loss_bbox_4: 0.1671 (0.2222) loss_giou_4: 0.1612 (0.2087)
loss_ce_unscaled: 0.1172 (0.2537) class_error_unscaled: 0.0000 (20.6479)
loss_bbox_unscaled: 0.0323 (0.0433) loss_giou_unscaled: 0.0826 (0.1032)
cardinality_error_unscaled: 1.0000 (3.6656) loss_ce_0_unscaled: 0.1997
(0.2849) loss_bbox_0_unscaled: 0.0377 (0.0549) loss_giou_0_unscaled:
0.1131 (0.1273) cardinality_error_0_unscaled: 2.0000 (4.9901)
loss_ce_1_unscaled: 0.1390 (0.2664) loss_bbox_1_unscaled: 0.0359 (0.0502)
loss_giou_1_unscaled: 0.0865 (0.1165) cardinality_error_1_unscaled:
1.0000 (4.0728) loss_ce_2_unscaled: 0.1721 (0.2571)
loss_bbox_2_unscaled: 0.0334 (0.0458) loss_giou_2_unscaled: 0.0936
(0.1089) cardinality_error_2_unscaled: 1.0000 (3.8907)
loss_ce_3_unscaled: 0.1161 (0.2561) loss_bbox_3_unscaled: 0.0327 (0.0455)
loss_giou_3_unscaled: 0.0831 (0.1073) cardinality_error_3_unscaled:
1.5000 (3.8477) loss_ce_4_unscaled: 0.1227 (0.2516)
loss_bbox_4_unscaled: 0.0334 (0.0444) loss_giou_4_unscaled: 0.0806
(0.1043) cardinality_error_4_unscaled: 1.0000 (3.6623) time: 0.3261
data: 0.0172 max mem: 3338
Epoch: [0] [ 160/8562] eta: 0:48:52 lr: 0.000010 class_error: 33.33
loss: 2.8527 (4.2568) loss_ce: 0.0501 (0.2466) loss_bbox: 0.1336
(0.2127) loss_giou: 0.1652 (0.2054) loss_ce_0: 0.0982 (0.2781)
loss_bbox_0: 0.1868 (0.2713) loss_giou_0: 0.2471 (0.2554) loss_ce_1:
0.0649 (0.2594) loss_bbox_1: 0.1689 (0.2474) loss_giou_1: 0.2112
(0.2327) loss_ce_2: 0.0721 (0.2512) loss_bbox_2: 0.1632 (0.2243)
loss_giou_2: 0.1872 (0.2164) loss_ce_3: 0.0617 (0.2493) loss_bbox_3:
0.1485 (0.2229) loss_giou_3: 0.1821 (0.2135) loss_ce_4: 0.0482 (0.2443)
loss_bbox_4: 0.1426 (0.2186) loss_giou_4: 0.1726 (0.2073)
loss_ce_unscaled: 0.0501 (0.2466) class_error_unscaled: 0.0000 (20.7216)
```

```
loss_bbox_unscaled: 0.0267 (0.0425) loss_giou_unscaled: 0.0826 (0.1027)
cardinality_error_unscaled: 1.0000 (3.5031) loss_ce_0_unscaled: 0.0982
(0.2781) loss_bbox_0_unscaled: 0.0374 (0.0543) loss_giou_0_unscaled:
0.1235 (0.1277) cardinality_error_0_unscaled: 1.5000 (4.8012)
loss_ce_1_unscaled: 0.0649 (0.2594) loss_bbox_1_unscaled: 0.0338 (0.0495)
loss_giou_1_unscaled: 0.1056 (0.1164) cardinality_error_1_unscaled:
1.0000 (3.8944) loss_ce_2_unscaled: 0.0721 (0.2512)
loss_bbox_2_unscaled: 0.0326 (0.0449) loss_giou_2_unscaled: 0.0936
(0.1082) cardinality_error_2_unscaled: 1.0000 (3.7236)
loss_ce_3_unscaled: 0.0617 (0.2493) loss_bbox_3_unscaled: 0.0297 (0.0446)
loss_giou_3_unscaled: 0.0911 (0.1067) cardinality_error_3_unscaled:
1.0000 (3.6708) loss_ce_4_unscaled: 0.0482 (0.2443)
loss_bbox_4_unscaled: 0.0285 (0.0437) loss_giou_4_unscaled: 0.0863
(0.1037) cardinality_error_4_unscaled: 1.0000 (3.5093) time: 0.3121
data: 0.0150 max mem: 3338
Epoch: [0] [ 170/8562] eta: 0:48:49 lr: 0.000010 class_error: 16.67
loss: 2.8499 (4.2031) loss_ce: 0.0677 (0.2438) loss_bbox: 0.1265
(0.2091) loss_giou: 0.1602 (0.2040) loss_ce_0: 0.0982 (0.2740)
loss_bbox_0: 0.1913 (0.2667) loss_giou_0: 0.2269 (0.2541) loss_ce_1:
0.0649 (0.2549) loss_bbox_1: 0.1362 (0.2426) loss_giou_1: 0.2112
(0.2312) loss_ce_2: 0.0676 (0.2465) loss_bbox_2: 0.1472 (0.2210)
loss_giou_2: 0.1730 (0.2161) loss_ce_3: 0.0632 (0.2444) loss_bbox_3:
0.1401 (0.2196) loss_giou_3: 0.1647 (0.2127) loss_ce_4: 0.0692 (0.2408)
loss_bbox_4: 0.1367 (0.2152) loss_giou_4: 0.1541 (0.2065)
loss_ce_unscaled: 0.0677 (0.2438) class_error_unscaled: 16.6667 (20.4566)
loss_bbox_unscaled: 0.0253 (0.0418) loss_giou_unscaled: 0.0801 (0.1020)
cardinality_error_unscaled: 1.0000 (3.3626) loss_ce_0_unscaled: 0.0982
(0.2740) loss_bbox_0_unscaled: 0.0383 (0.0533) loss_giou_0_unscaled:
0.1134 (0.1270) cardinality_error_0_unscaled: 1.5000 (4.6082)
loss_ce_1_unscaled: 0.0649 (0.2549) loss_bbox_1_unscaled: 0.0272 (0.0485)
loss_giou_1_unscaled: 0.1056 (0.1156) cardinality_error_1_unscaled:
1.0000 (3.7398) loss_ce_2_unscaled: 0.0676 (0.2465)
loss_bbox_2_unscaled: 0.0294 (0.0442) loss_giou_2_unscaled: 0.0865
(0.1080) cardinality_error_2_unscaled: 1.0000 (3.5760)
loss_ce_3_unscaled: 0.0632 (0.2444) loss_bbox_3_unscaled: 0.0280 (0.0439)
loss_giou_3_unscaled: 0.0824 (0.1064) cardinality_error_3_unscaled:
0.5000 (3.5146) loss_ce_4_unscaled: 0.0692 (0.2408)
loss_bbox_4_unscaled: 0.0273 (0.0430) loss_giou_4_unscaled: 0.0771
(0.1033) cardinality_error_4_unscaled: 1.0000 (3.3684) time: 0.3354
data: 0.0136 max mem: 3652
Epoch: [0] [ 180/8562] eta: 0:48:41 lr: 0.000010 class_error: 0.00
loss: 2.9393 (4.1550) loss_ce: 0.1584 (0.2414) loss_bbox: 0.1372
(0.2072) loss_giou: 0.1468 (0.2023) loss_ce_0: 0.1599 (0.2721)
loss_bbox_0: 0.1892 (0.2619) loss_giou_0: 0.2055 (0.2510) loss_ce_1:
0.1072 (0.2523) loss_bbox_1: 0.1481 (0.2381) loss_giou_1: 0.1833
(0.2282) loss_ce_2: 0.1700 (0.2453) loss_bbox_2: 0.1409 (0.2172)
loss_giou_2: 0.1670 (0.2135) loss_ce_3: 0.1612 (0.2422) loss_bbox_3:
```

```
0.1625 (0.2162) loss_giou_3: 0.1542 (0.2101) loss_ce_4: 0.1278 (0.2384)
loss_bbox_4: 0.1537 (0.2129) loss_giou_4: 0.1522 (0.2046)
loss_ce_unscaled: 0.1584 (0.2414) class_error_unscaled: 8.3333 (20.2472)
loss_bbox_unscaled: 0.0274 (0.0414) loss_giou_unscaled: 0.0734 (0.1012)
cardinality_error_unscaled: 1.0000 (3.2707) loss_ce_0_unscaled: 0.1599
(0.2721) loss_bbox_0_unscaled: 0.0378 (0.0524) loss_giou_0_unscaled:
0.1028 (0.1255) cardinality_error_0_unscaled: 1.0000 (4.4945)
loss_ce_1_unscaled: 0.1072 (0.2523) loss_bbox_1_unscaled: 0.0296 (0.0476)
loss_giou_1_unscaled: 0.0916 (0.1141) cardinality_error_1_unscaled:
1.0000 (3.6464) loss_ce_2_unscaled: 0.1700 (0.2453)
loss_bbox_2_unscaled: 0.0282 (0.0434) loss_giou_2_unscaled: 0.0835
(0.1068) cardinality_error_2_unscaled: 1.0000 (3.4751)
loss_ce_3_unscaled: 0.1612 (0.2422) loss_bbox_3_unscaled: 0.0325 (0.0432)
loss_giou_3_unscaled: 0.0771 (0.1051) cardinality_error_3_unscaled:
1.0000 (3.4254) loss_ce_4_unscaled: 0.1278 (0.2384)
loss_bbox_4_unscaled: 0.0307 (0.0426) loss_giou_4_unscaled: 0.0761
(0.1023) cardinality_error_4_unscaled: 1.0000 (3.2735) time: 0.3450
data: 0.0155 max mem: 3652
Epoch: [0] [ 190/8562] eta: 0:48:16 lr: 0.000010 class_error: 42.86
loss: 3.1717 (4.1406) loss_ce: 0.1229 (0.2375) loss_bbox: 0.1820
(0.2066) loss_giou: 0.1733 (0.2049) loss_ce_0: 0.1599 (0.2674)
loss_bbox_0: 0.2034 (0.2622) loss_giou_0: 0.2112 (0.2535) loss_ce_1:
0.1072 (0.2489) loss_bbox_1: 0.1589 (0.2370) loss_giou_1: 0.2103
(0.2300) loss_ce_2: 0.1338 (0.2418) loss_bbox_2: 0.1691 (0.2164)
loss_giou_2: 0.1721 (0.2158) loss_ce_3: 0.1201 (0.2382) loss_bbox_3:
0.1755 (0.2150) loss_giou_3: 0.1594 (0.2118) loss_ce_4: 0.1187 (0.2348)
loss_bbox_4: 0.1689 (0.2118) loss_giou_4: 0.1850 (0.2069)
loss_ce_unscaled: 0.1229 (0.2375) class_error_unscaled: 0.0000 (19.8216)
loss_bbox_unscaled: 0.0364 (0.0413) loss_giou_unscaled: 0.0867 (0.1025)
cardinality_error_unscaled: 1.0000 (3.2199) loss_ce_0_unscaled: 0.1599
(0.2674) loss_bbox_0_unscaled: 0.0407 (0.0524) loss_giou_0_unscaled:
0.1056 (0.1268) cardinality_error_0_unscaled: 1.5000 (4.4136)
loss_ce_1_unscaled: 0.1072 (0.2489) loss_bbox_1_unscaled: 0.0318 (0.0474)
loss_giou_1_unscaled: 0.1052 (0.1150) cardinality_error_1_unscaled:
1.0000 (3.5864) loss_ce_2_unscaled: 0.1338 (0.2418)
loss_bbox_2_unscaled: 0.0338 (0.0433) loss_giou_2_unscaled: 0.0860
(0.1079) cardinality_error_2_unscaled: 1.0000 (3.4346)
loss_ce_3_unscaled: 0.1201 (0.2382) loss_bbox_3_unscaled: 0.0351 (0.0430)
loss_giou_3_unscaled: 0.0797 (0.1059) cardinality_error_3_unscaled:
1.0000 (3.3743) loss_ce_4_unscaled: 0.1187 (0.2348)
loss_bbox_4_unscaled: 0.0338 (0.0424) loss_giou_4_unscaled: 0.0925
(0.1034) cardinality_error_4_unscaled: 1.0000 (3.2225) time: 0.3193
data: 0.0154 max mem: 3652
Epoch: [0] [ 200/8562] eta: 0:48:02 lr: 0.000010 class_error: 0.00
loss: 3.4570 (4.1455) loss_ce: 0.0957 (0.2364) loss_bbox: 0.1914
(0.2081) loss_giou: 0.2017 (0.2058) loss_ce_0: 0.1882 (0.2695)
loss_bbox_0: 0.2530 (0.2618) loss_giou_0: 0.2268 (0.2525) loss_ce_1:
```

```
0.1425 (0.2510) loss_bbox_1: 0.2016 (0.2375) loss_giou_1: 0.2182  
(0.2298) loss_ce_2: 0.1062 (0.2406) loss_bbox_2: 0.1882 (0.2174)  
loss_giou_2: 0.2035 (0.2158) loss_ce_3: 0.1126 (0.2369) loss_bbox_3:  
0.1819 (0.2162) loss_giou_3: 0.1945 (0.2120) loss_ce_4: 0.0994 (0.2335)  
loss_bbox_4: 0.1888 (0.2131) loss_giou_4: 0.2051 (0.2075)  
loss_ce_unscaled: 0.0957 (0.2364) class_error_unscaled: 0.0000 (19.4546)  
loss_bbox_unscaled: 0.0383 (0.0416) loss_giou_unscaled: 0.1008 (0.1029)  
cardinality_error_unscaled: 1.0000 (3.1692) loss_ce_0_unscaled: 0.1882  
(0.2695) loss_bbox_0_unscaled: 0.0506 (0.0524) loss_giou_0_unscaled:  
0.1134 (0.1262) cardinality_error_0_unscaled: 1.5000 (4.3383)  
loss_ce_1_unscaled: 0.1425 (0.2510) loss_bbox_1_unscaled: 0.0403 (0.0475)  
loss_giou_1_unscaled: 0.1091 (0.1149) cardinality_error_1_unscaled:  
1.0000 (3.5174) loss_ce_2_unscaled: 0.1062 (0.2406)  
loss_bbox_2_unscaled: 0.0376 (0.0435) loss_giou_2_unscaled: 0.1017  
(0.1079) cardinality_error_2_unscaled: 1.5000 (3.3831)  
loss_ce_3_unscaled: 0.1126 (0.2369) loss_bbox_3_unscaled: 0.0364 (0.0432)  
loss_giou_3_unscaled: 0.0972 (0.1060) cardinality_error_3_unscaled:  
1.5000 (3.3184) loss_ce_4_unscaled: 0.0994 (0.2335)  
loss_bbox_4_unscaled: 0.0378 (0.0426) loss_giou_4_unscaled: 0.1026  
(0.1037) cardinality_error_4_unscaled: 1.0000 (3.1617) time: 0.3101  
data: 0.0133 max mem: 3652  
Epoch: [0] [ 210/8562] eta: 0:47:53 lr: 0.000010 class_error: 0.00  
loss: 3.4570 (4.1596) loss_ce: 0.1229 (0.2370) loss_bbox: 0.1780  
(0.2081) loss_giou: 0.1610 (0.2069) loss_ce_0: 0.2284 (0.2726)  
loss_bbox_0: 0.2499 (0.2613) loss_giou_0: 0.2430 (0.2543) loss_ce_1:  
0.1989 (0.2524) loss_bbox_1: 0.2016 (0.2372) loss_giou_1: 0.1844  
(0.2305) loss_ce_2: 0.1395 (0.2414) loss_bbox_2: 0.1700 (0.2174)  
loss_giou_2: 0.1721 (0.2172) loss_ce_3: 0.1297 (0.2381) loss_bbox_3:  
0.1708 (0.2161) loss_giou_3: 0.1665 (0.2130) loss_ce_4: 0.1273 (0.2347)  
loss_bbox_4: 0.1734 (0.2130) loss_giou_4: 0.1584 (0.2086)  
loss_ce_unscaled: 0.1229 (0.2370) class_error_unscaled: 0.0000 (19.5264)  
loss_bbox_unscaled: 0.0356 (0.0416) loss_giou_unscaled: 0.0805 (0.1035)  
cardinality_error_unscaled: 1.5000 (3.2322) loss_ce_0_unscaled: 0.2284  
(0.2726) loss_bbox_0_unscaled: 0.0500 (0.0523) loss_giou_0_unscaled:  
0.1215 (0.1271) cardinality_error_0_unscaled: 1.5000 (4.4431)  
loss_ce_1_unscaled: 0.1989 (0.2524) loss_bbox_1_unscaled: 0.0403 (0.0474)  
loss_giou_1_unscaled: 0.0922 (0.1153) cardinality_error_1_unscaled:  
1.5000 (3.5995) loss_ce_2_unscaled: 0.1395 (0.2414)  
loss_bbox_2_unscaled: 0.0340 (0.0435) loss_giou_2_unscaled: 0.0861  
(0.1086) cardinality_error_2_unscaled: 1.5000 (3.4692)  
loss_ce_3_unscaled: 0.1297 (0.2381) loss_bbox_3_unscaled: 0.0342 (0.0432)  
loss_giou_3_unscaled: 0.0833 (0.1065) cardinality_error_3_unscaled:  
1.5000 (3.4028) loss_ce_4_unscaled: 0.1273 (0.2347)  
loss_bbox_4_unscaled: 0.0347 (0.0426) loss_giou_4_unscaled: 0.0792  
(0.1043) cardinality_error_4_unscaled: 1.0000 (3.2251) time: 0.3256  
data: 0.0126 max mem: 3652
```

```
Epoch: [0]  [ 220/8562]  eta: 0:47:44  lr: 0.000010  class_error: 54.55
loss: 3.5411 (4.1443)  loss_ce: 0.1229 (0.2360)  loss_bbox: 0.1567
(0.2087)  loss_giou: 0.1423 (0.2058)  loss_ce_0: 0.1912 (0.2706)
loss_bbox_0: 0.2655 (0.2623)  loss_giou_0: 0.2432 (0.2530)  loss_ce_1:
0.1482 (0.2520)  loss_bbox_1: 0.1523 (0.2347)  loss_giou_1: 0.1732
(0.2279)  loss_ce_2: 0.1387 (0.2396)  loss_bbox_2: 0.1700 (0.2183)
loss_giou_2: 0.1721 (0.2156)  loss_ce_3: 0.1297 (0.2368)  loss_bbox_3:
0.1708 (0.2167)  loss_giou_3: 0.1522 (0.2116)  loss_ce_4: 0.1273 (0.2339)
loss_bbox_4: 0.1557 (0.2135)  loss_giou_4: 0.1584 (0.2073)
loss_ce_unscaled: 0.1229 (0.2360)  class_error_unscaled: 20.0000 (19.4327)
loss_bbox_unscaled: 0.0313 (0.0417)  loss_giou_unscaled: 0.0711 (0.1029)
cardinality_error_unscaled: 0.5000 (3.1222)  loss_ce_0_unscaled: 0.1912
(0.2706)  loss_bbox_0_unscaled: 0.0531 (0.0525)  loss_giou_0_unscaled:
0.1216 (0.1265)  cardinality_error_0_unscaled: 1.0000 (4.3122)
loss_ce_1_unscaled: 0.1482 (0.2520)  loss_bbox_1_unscaled: 0.0305 (0.0469)
loss_giou_1_unscaled: 0.0866 (0.1139)  cardinality_error_1_unscaled:
0.5000 (3.4842)  loss_ce_2_unscaled: 0.1387 (0.2396)
loss_bbox_2_unscaled: 0.0340 (0.0437)  loss_giou_2_unscaled: 0.0861
(0.1078)  cardinality_error_2_unscaled: 0.5000 (3.3529)
loss_ce_3_unscaled: 0.1297 (0.2368)  loss_bbox_3_unscaled: 0.0342 (0.0433)
loss_giou_3_unscaled: 0.0761 (0.1058)  cardinality_error_3_unscaled:
0.5000 (3.2964)  loss_ce_4_unscaled: 0.1273 (0.2339)
loss_bbox_4_unscaled: 0.0311 (0.0427)  loss_giou_4_unscaled: 0.0792
(0.1037)  cardinality_error_4_unscaled: 0.5000 (3.1244)  time: 0.3293
data: 0.0130  max mem: 3652
Epoch: [0]  [ 230/8562]  eta: 0:47:41  lr: 0.000010  class_error: 20.00
loss: 3.2989 (4.1210)  loss_ce: 0.1697 (0.2328)  loss_bbox: 0.1567
(0.2093)  loss_giou: 0.1517 (0.2051)  loss_ce_0: 0.1949 (0.2681)
loss_bbox_0: 0.2566 (0.2627)  loss_giou_0: 0.2337 (0.2518)  loss_ce_1:
0.1635 (0.2494)  loss_bbox_1: 0.1636 (0.2343)  loss_giou_1: 0.1691
(0.2264)  loss_ce_2: 0.1252 (0.2365)  loss_bbox_2: 0.1857 (0.2185)
loss_giou_2: 0.1724 (0.2146)  loss_ce_3: 0.1737 (0.2336)  loss_bbox_3:
0.1902 (0.2168)  loss_giou_3: 0.1557 (0.2104)  loss_ce_4: 0.1769 (0.2309)
loss_bbox_4: 0.1557 (0.2136)  loss_giou_4: 0.1410 (0.2062)
loss_ce_unscaled: 0.1697 (0.2328)  class_error_unscaled: 0.0000 (18.9233)
loss_bbox_unscaled: 0.0313 (0.0419)  loss_giou_unscaled: 0.0759 (0.1025)
cardinality_error_unscaled: 0.5000 (3.1688)  loss_ce_0_unscaled: 0.1949
(0.2681)  loss_bbox_0_unscaled: 0.0513 (0.0525)  loss_giou_0_unscaled:
0.1169 (0.1259)  cardinality_error_0_unscaled: 2.0000 (4.3788)
loss_ce_1_unscaled: 0.1635 (0.2494)  loss_bbox_1_unscaled: 0.0327 (0.0469)
loss_giou_1_unscaled: 0.0846 (0.1132)  cardinality_error_1_unscaled:
0.5000 (3.5108)  loss_ce_2_unscaled: 0.1252 (0.2365)
loss_bbox_2_unscaled: 0.0371 (0.0437)  loss_giou_2_unscaled: 0.0862
(0.1073)  cardinality_error_2_unscaled: 1.0000 (3.3874)
loss_ce_3_unscaled: 0.1737 (0.2336)  loss_bbox_3_unscaled: 0.0380 (0.0434)
loss_giou_3_unscaled: 0.0779 (0.1052)  cardinality_error_3_unscaled:
1.0000 (3.3268)  loss_ce_4_unscaled: 0.1769 (0.2309)
```

```
loss_bbox_4_unscaled: 0.0311 (0.0427) loss_giou_4_unscaled: 0.0705  
(0.1031) cardinality_error_4_unscaled: 1.0000 (3.1645) time: 0.3369  
data: 0.0167 max mem: 3652  
Epoch: [0] [ 240/8562] eta: 0:47:36 lr: 0.000010 class_error: 0.00  
loss: 2.7311 (4.0723) loss_ce: 0.1413 (0.2293) loss_bbox: 0.1488  
(0.2069) loss_giou: 0.1517 (0.2038) loss_ce_0: 0.2087 (0.2637)  
loss_bbox_0: 0.1933 (0.2598) loss_giou_0: 0.2097 (0.2502) loss_ce_1:  
0.1743 (0.2461) loss_bbox_1: 0.1636 (0.2313) loss_giou_1: 0.1824  
(0.2247) loss_ce_2: 0.1767 (0.2326) loss_bbox_2: 0.1440 (0.2157)  
loss_giou_2: 0.1611 (0.2127) loss_ce_3: 0.1604 (0.2302) loss_bbox_3:  
0.1492 (0.2139) loss_giou_3: 0.1557 (0.2088) loss_ce_4: 0.1512 (0.2274)  
loss_bbox_4: 0.1405 (0.2106) loss_giou_4: 0.1398 (0.2048)  
loss_ce_unscaled: 0.1413 (0.2293) class_error_unscaled: 0.0000 (18.8494)  
loss_bbox_unscaled: 0.0298 (0.0414) loss_giou_unscaled: 0.0759 (0.1019)  
cardinality_error_unscaled: 2.0000 (3.1100) loss_ce_0_unscaled: 0.2087  
(0.2637) loss_bbox_0_unscaled: 0.0387 (0.0520) loss_giou_0_unscaled:  
0.1049 (0.1251) cardinality_error_0_unscaled: 2.0000 (4.2988)  
loss_ce_1_unscaled: 0.1743 (0.2461) loss_bbox_1_unscaled: 0.0327 (0.0463)  
loss_giou_1_unscaled: 0.0912 (0.1124) cardinality_error_1_unscaled:  
1.5000 (3.4336) loss_ce_2_unscaled: 0.1767 (0.2326)  
loss_bbox_2_unscaled: 0.0288 (0.0431) loss_giou_2_unscaled: 0.0805  
(0.1063) cardinality_error_2_unscaled: 1.5000 (3.3195)  
loss_ce_3_unscaled: 0.1604 (0.2302) loss_bbox_3_unscaled: 0.0298 (0.0428)  
loss_giou_3_unscaled: 0.0779 (0.1044) cardinality_error_3_unscaled:  
1.0000 (3.2593) loss_ce_4_unscaled: 0.1512 (0.2274)  
loss_bbox_4_unscaled: 0.0281 (0.0421) loss_giou_4_unscaled: 0.0699  
(0.1024) cardinality_error_4_unscaled: 1.5000 (3.1037) time: 0.3416  
data: 0.0164 max mem: 3652  
Epoch: [0] [ 250/8562] eta: 0:47:34 lr: 0.000010 class_error: 22.22  
loss: 3.0128 (4.0848) loss_ce: 0.1834 (0.2308) loss_bbox: 0.1292  
(0.2055) loss_giou: 0.1710 (0.2054) loss_ce_0: 0.1566 (0.2629)  
loss_bbox_0: 0.1920 (0.2616) loss_giou_0: 0.2248 (0.2533) loss_ce_1:  
0.1748 (0.2439) loss_bbox_1: 0.1422 (0.2329) loss_giou_1: 0.2051  
(0.2283) loss_ce_2: 0.1710 (0.2318) loss_bbox_2: 0.1239 (0.2148)  
loss_giou_2: 0.1483 (0.2147) loss_ce_3: 0.1626 (0.2314) loss_bbox_3:  
0.1257 (0.2123) loss_giou_3: 0.1603 (0.2105) loss_ce_4: 0.1922 (0.2293)  
loss_bbox_4: 0.1265 (0.2089) loss_giou_4: 0.1582 (0.2065)  
loss_ce_unscaled: 0.1834 (0.2308) class_error_unscaled: 23.0769 (19.4391)  
loss_bbox_unscaled: 0.0258 (0.0411) loss_giou_unscaled: 0.0855 (0.1027)  
cardinality_error_unscaled: 0.5000 (3.0359) loss_ce_0_unscaled: 0.1566  
(0.2629) loss_bbox_0_unscaled: 0.0384 (0.0523) loss_giou_0_unscaled:  
0.1124 (0.1266) cardinality_error_0_unscaled: 1.0000 (4.2211)  
loss_ce_1_unscaled: 0.1748 (0.2439) loss_bbox_1_unscaled: 0.0284 (0.0466)  
loss_giou_1_unscaled: 0.1025 (0.1141) cardinality_error_1_unscaled:  
0.5000 (3.3645) loss_ce_2_unscaled: 0.1710 (0.2318)  
loss_bbox_2_unscaled: 0.0248 (0.0430) loss_giou_2_unscaled: 0.0741  
(0.1073) cardinality_error_2_unscaled: 1.0000 (3.2610)
```

```
loss_ce_3_unscaled: 0.1626 (0.2314) loss_bbox_3_unscaled: 0.0251 (0.0425)
loss_giou_3_unscaled: 0.0802 (0.1052) cardinality_error_3_unscaled:
0.5000 (3.1873) loss_ce_4_unscaled: 0.1922 (0.2293)
loss_bbox_4_unscaled: 0.0253 (0.0418) loss_giou_4_unscaled: 0.0791
(0.1033) cardinality_error_4_unscaled: 0.5000 (3.0378) time: 0.3443
data: 0.0136 max mem: 3652
Epoch: [0] [ 260/8562] eta: 0:47:25 lr: 0.000010 class_error: 33.33
loss: 3.4941 (4.0560) loss_ce: 0.1834 (0.2263) loss_bbox: 0.1643
(0.2059) loss_giou: 0.1809 (0.2048) loss_ce_0: 0.2017 (0.2603)
loss_bbox_0: 0.2088 (0.2609) loss_giou_0: 0.2475 (0.2513) loss_ce_1:
0.1287 (0.2402) loss_bbox_1: 0.1923 (0.2329) loss_giou_1: 0.2051
(0.2271) loss_ce_2: 0.1462 (0.2280) loss_bbox_2: 0.1942 (0.2149)
loss_giou_2: 0.2116 (0.2137) loss_ce_3: 0.1237 (0.2268) loss_bbox_3:
0.1874 (0.2128) loss_giou_3: 0.2292 (0.2100) loss_ce_4: 0.1798 (0.2246)
loss_bbox_4: 0.1608 (0.2094) loss_giou_4: 0.2128 (0.2062)
loss_ce_unscaled: 0.1834 (0.2263) class_error_unscaled: 20.0000 (19.2818)
loss_bbox_unscaled: 0.0329 (0.0412) loss_giou_unscaled: 0.0905 (0.1024)
cardinality_error_unscaled: 1.0000 (2.9655) loss_ce_0_unscaled: 0.2017
(0.2603) loss_bbox_0_unscaled: 0.0418 (0.0522) loss_giou_0_unscaled:
0.1238 (0.1256) cardinality_error_0_unscaled: 1.5000 (4.1207)
loss_ce_1_unscaled: 0.1287 (0.2402) loss_bbox_1_unscaled: 0.0385 (0.0466)
loss_giou_1_unscaled: 0.1025 (0.1135) cardinality_error_1_unscaled:
1.0000 (3.2816) loss_ce_2_unscaled: 0.1462 (0.2280)
loss_bbox_2_unscaled: 0.0388 (0.0430) loss_giou_2_unscaled: 0.1058
(0.1068) cardinality_error_2_unscaled: 1.0000 (3.1839)
loss_ce_3_unscaled: 0.1237 (0.2268) loss_bbox_3_unscaled: 0.0375 (0.0426)
loss_giou_3_unscaled: 0.1146 (0.1050) cardinality_error_3_unscaled:
1.0000 (3.1188) loss_ce_4_unscaled: 0.1798 (0.2246)
loss_bbox_4_unscaled: 0.0322 (0.0419) loss_giou_4_unscaled: 0.1064
(0.1031) cardinality_error_4_unscaled: 1.0000 (2.9674) time: 0.3374
data: 0.0139 max mem: 3652
Epoch: [0] [ 270/8562] eta: 0:47:21 lr: 0.000010 class_error: 0.00
loss: 2.9274 (4.0290) loss_ce: 0.0821 (0.2240) loss_bbox: 0.1722
(0.2049) loss_giou: 0.1505 (0.2032) loss_ce_0: 0.1193 (0.2585)
loss_bbox_0: 0.2513 (0.2610) loss_giou_0: 0.1653 (0.2506) loss_ce_1:
0.0985 (0.2387) loss_bbox_1: 0.1792 (0.2320) loss_giou_1: 0.1484
(0.2255) loss_ce_2: 0.1122 (0.2263) loss_bbox_2: 0.1845 (0.2138)
loss_giou_2: 0.1494 (0.2119) loss_ce_3: 0.0908 (0.2247) loss_bbox_3:
0.1629 (0.2112) loss_giou_3: 0.1414 (0.2081) loss_ce_4: 0.0887 (0.2229)
loss_bbox_4: 0.1683 (0.2077) loss_giou_4: 0.1507 (0.2043)
loss_ce_unscaled: 0.0821 (0.2240) class_error_unscaled: 0.0000 (19.4374)
loss_bbox_unscaled: 0.0344 (0.0410) loss_giou_unscaled: 0.0753 (0.1016)
cardinality_error_unscaled: 1.0000 (2.8985) loss_ce_0_unscaled: 0.1193
(0.2585) loss_bbox_0_unscaled: 0.0503 (0.0522) loss_giou_0_unscaled:
0.0826 (0.1253) cardinality_error_0_unscaled: 1.5000 (4.0461)
loss_ce_1_unscaled: 0.0985 (0.2387) loss_bbox_1_unscaled: 0.0358 (0.0464)
loss_giou_1_unscaled: 0.0742 (0.1127) cardinality_error_1_unscaled:
```

```
1.0000 (3.2122) loss_ce_2_unscaled: 0.1122 (0.2263)
loss_bbox_2_unscaled: 0.0369 (0.0428) loss_giou_2_unscaled: 0.0747
(0.1060) cardinality_error_2_unscaled: 1.0000 (3.1218)
loss_ce_3_unscaled: 0.0908 (0.2247) loss_bbox_3_unscaled: 0.0326 (0.0422)
loss_giou_3_unscaled: 0.0707 (0.1040) cardinality_error_3_unscaled:
1.0000 (3.0554) loss_ce_4_unscaled: 0.0887 (0.2229)
loss_bbox_4_unscaled: 0.0337 (0.0415) loss_giou_4_unscaled: 0.0753
(0.1021) cardinality_error_4_unscaled: 1.0000 (2.9022) time: 0.3330
data: 0.0166 max mem: 3652
Epoch: [0] [ 280/8562] eta: 0:47:17 lr: 0.000010 class error: 33.33
loss: 2.7483 (4.0251) loss_ce: 0.0821 (0.2234) loss_bbox: 0.1665
(0.2046) loss_giou: 0.1480 (0.2035) loss_ce_0: 0.0954 (0.2574)
loss_bbox_0: 0.2538 (0.2618) loss_giou_0: 0.1988 (0.2512) loss_ce_1:
0.0680 (0.2368) loss_bbox_1: 0.1664 (0.2326) loss_giou_1: 0.1796
(0.2260) loss_ce_2: 0.0952 (0.2250) loss_bbox_2: 0.1677 (0.2137)
loss_giou_2: 0.1526 (0.2124) loss_ce_3: 0.0808 (0.2233) loss_bbox_3:
0.1494 (0.2108) loss_giou_3: 0.1403 (0.2086) loss_ce_4: 0.0829 (0.2220)
loss_bbox_4: 0.1447 (0.2074) loss_giou_4: 0.1485 (0.2046)
loss_ce_unscaled: 0.0821 (0.2234) class_error_unscaled: 0.0000 (19.4744)
loss_bbox_unscaled: 0.0333 (0.0409) loss_giou_unscaled: 0.0740 (0.1018)
cardinality_error_unscaled: 0.5000 (2.8167) loss_ce_0_unscaled: 0.0954
(0.2574) loss_bbox_0_unscaled: 0.0508 (0.0524) loss_giou_0_unscaled:
0.0994 (0.1256) cardinality_error_0_unscaled: 1.5000 (3.9448)
loss_ce_1_unscaled: 0.0680 (0.2368) loss_bbox_1_unscaled: 0.0333 (0.0465)
loss_giou_1_unscaled: 0.0898 (0.1130) cardinality_error_1_unscaled:
0.5000 (3.1246) loss_ce_2_unscaled: 0.0952 (0.2250)
loss_bbox_2_unscaled: 0.0335 (0.0427) loss_giou_2_unscaled: 0.0763
(0.1062) cardinality_error_2_unscaled: 0.5000 (3.0338)
loss_ce_3_unscaled: 0.0808 (0.2233) loss_bbox_3_unscaled: 0.0299 (0.0422)
loss_giou_3_unscaled: 0.0701 (0.1043) cardinality_error_3_unscaled:
0.5000 (2.9644) loss_ce_4_unscaled: 0.0829 (0.2220)
loss_bbox_4_unscaled: 0.0289 (0.0415) loss_giou_4_unscaled: 0.0743
(0.1023) cardinality_error_4_unscaled: 0.5000 (2.8185) time: 0.3407
data: 0.0163 max mem: 3652
Epoch: [0] [ 290/8562] eta: 0:47:11 lr: 0.000010 class error: 0.00
loss: 2.3117 (3.9815) loss_ce: 0.0425 (0.2192) loss_bbox: 0.1665
(0.2036) loss_giou: 0.1105 (0.2018) loss_ce_0: 0.0800 (0.2537)
loss_bbox_0: 0.1859 (0.2593) loss_giou_0: 0.1273 (0.2488) loss_ce_1:
0.0533 (0.2336) loss_bbox_1: 0.1656 (0.2305) loss_giou_1: 0.1503
(0.2239) loss_ce_2: 0.0491 (0.2215) loss_bbox_2: 0.1750 (0.2124)
loss_giou_2: 0.1191 (0.2103) loss_ce_3: 0.0510 (0.2196) loss_bbox_3:
0.1871 (0.2097) loss_giou_3: 0.1141 (0.2066) loss_ce_4: 0.0471 (0.2178)
loss_bbox_4: 0.1668 (0.2065) loss_giou_4: 0.1139 (0.2029)
loss_ce_unscaled: 0.0425 (0.2192) class_error_unscaled: 0.0000 (18.8052)
loss_bbox_unscaled: 0.0333 (0.0407) loss_giou_unscaled: 0.0553 (0.1009)
cardinality_error_unscaled: 0.5000 (2.8454) loss_ce_0_unscaled: 0.0800
(0.2537) loss_bbox_0_unscaled: 0.0372 (0.0519) loss_giou_0_unscaled:
```

```
0.0636 (0.1244) cardinality_error_0_unscaled: 1.5000 (3.9794)
loss_ce_1_unscaled: 0.0533 (0.2336) loss_bbox_1_unscaled: 0.0331 (0.0461)
loss_giou_1_unscaled: 0.0752 (0.1119) cardinality_error_1_unscaled:
1.0000 (3.1649) loss_ce_2_unscaled: 0.0491 (0.2215)
loss_bbox_2_unscaled: 0.0350 (0.0425) loss_giou_2_unscaled: 0.0596
(0.1051) cardinality_error_2_unscaled: 0.5000 (3.0773)
loss_ce_3_unscaled: 0.0510 (0.2196) loss_bbox_3_unscaled: 0.0374 (0.0419)
loss_giou_3_unscaled: 0.0571 (0.1033) cardinality_error_3_unscaled:
0.5000 (2.9983) loss_ce_4_unscaled: 0.0471 (0.2178)
loss_bbox_4_unscaled: 0.0334 (0.0413) loss_giou_4_unscaled: 0.0570
(0.1015) cardinality_error_4_unscaled: 1.0000 (2.8557) time: 0.3371
data: 0.0126 max mem: 3652
Epoch: [0] [ 300/8562] eta: 0:47:05 lr: 0.000010 class_error: 0.00
loss: 2.4794 (3.9479) loss_ce: 0.0495 (0.2176) loss_bbox: 0.1486
(0.2022) loss_giou: 0.1056 (0.1990) loss_ce_0: 0.1186 (0.2527)
loss_bbox_0: 0.1859 (0.2594) loss_giou_0: 0.1561 (0.2465) loss_ce_1:
0.0931 (0.2310) loss_bbox_1: 0.1512 (0.2295) loss_giou_1: 0.1298
(0.2211) loss_ce_2: 0.0559 (0.2197) loss_bbox_2: 0.1701 (0.2109)
loss_giou_2: 0.1175 (0.2076) loss_ce_3: 0.0871 (0.2179) loss_bbox_3:
0.1546 (0.2078) loss_giou_3: 0.1050 (0.2035) loss_ce_4: 0.0530 (0.2164)
loss_bbox_4: 0.1465 (0.2050) loss_giou_4: 0.1049 (0.2002)
loss_ce_unscaled: 0.0495 (0.2176) class_error_unscaled: 0.0000 (18.7895)
loss_bbox_unscaled: 0.0297 (0.0404) loss_giou_unscaled: 0.0528 (0.0995)
cardinality_error_unscaled: 1.0000 (2.8073) loss_ce_0_unscaled: 0.1186
(0.2527) loss_bbox_0_unscaled: 0.0372 (0.0519) loss_giou_0_unscaled:
0.0780 (0.1232) cardinality_error_0_unscaled: 1.5000 (3.9336)
loss_ce_1_unscaled: 0.0931 (0.2310) loss_bbox_1_unscaled: 0.0302 (0.0459)
loss_giou_1_unscaled: 0.0649 (0.1106) cardinality_error_1_unscaled:
1.0000 (3.1096) loss_ce_2_unscaled: 0.0559 (0.2197)
loss_bbox_2_unscaled: 0.0340 (0.0422) loss_giou_2_unscaled: 0.0588
(0.1038) cardinality_error_2_unscaled: 0.5000 (3.0299)
loss_ce_3_unscaled: 0.0871 (0.2179) loss_bbox_3_unscaled: 0.0309 (0.0416)
loss_giou_3_unscaled: 0.0525 (0.1018) cardinality_error_3_unscaled:
1.0000 (2.9518) loss_ce_4_unscaled: 0.0530 (0.2164)
loss_bbox_4_unscaled: 0.0293 (0.0410) loss_giou_4_unscaled: 0.0524
(0.1001) cardinality_error_4_unscaled: 1.0000 (2.8189) time: 0.3322
data: 0.0131 max mem: 3652
Epoch: [0] [ 310/8562] eta: 0:47:02 lr: 0.000010 class_error: 0.00
loss: 2.6501 (3.9256) loss_ce: 0.0637 (0.2137) loss_bbox: 0.1756
(0.2037) loss_giou: 0.1085 (0.1980) loss_ce_0: 0.1095 (0.2484)
loss_bbox_0: 0.2900 (0.2607) loss_giou_0: 0.1765 (0.2453) loss_ce_1:
0.0931 (0.2280) loss_bbox_1: 0.1608 (0.2294) loss_giou_1: 0.1298
(0.2188) loss_ce_2: 0.0756 (0.2159) loss_bbox_2: 0.1701 (0.2127)
loss_giou_2: 0.1158 (0.2068) loss_ce_3: 0.0772 (0.2147) loss_bbox_3:
0.1642 (0.2091) loss_giou_3: 0.1049 (0.2024) loss_ce_4: 0.0644 (0.2130)
loss_bbox_4: 0.1513 (0.2061) loss_giou_4: 0.1175 (0.1990)
loss_ce_unscaled: 0.0637 (0.2137) class_error_unscaled: 0.0000 (18.7748)
```

```
loss_bbox_unscaled: 0.0351 (0.0407) loss_giou_unscaled: 0.0542 (0.0990)
cardinality_error_unscaled: 1.0000 (2.7765) loss_ce_0_unscaled: 0.1095
(0.2484) loss_bbox_0_unscaled: 0.0580 (0.0521) loss_giou_0_unscaled:
0.0882 (0.1226) cardinality_error_0_unscaled: 2.0000 (3.8923)
loss_ce_1_unscaled: 0.0931 (0.2280) loss_bbox_1_unscaled: 0.0322 (0.0459)
loss_giou_1_unscaled: 0.0649 (0.1094) cardinality_error_1_unscaled:
1.5000 (3.0852) loss_ce_2_unscaled: 0.0756 (0.2159)
loss_bbox_2_unscaled: 0.0340 (0.0425) loss_giou_2_unscaled: 0.0579
(0.1034) cardinality_error_2_unscaled: 1.0000 (2.9920)
loss_ce_3_unscaled: 0.0772 (0.2147) loss_bbox_3_unscaled: 0.0328 (0.0418)
loss_giou_3_unscaled: 0.0525 (0.1012) cardinality_error_3_unscaled:
1.0000 (2.9244) loss_ce_4_unscaled: 0.0644 (0.2130)
loss_bbox_4_unscaled: 0.0303 (0.0412) loss_giou_4_unscaled: 0.0588
(0.0995) cardinality_error_4_unscaled: 1.0000 (2.7926) time: 0.3381
data: 0.0162 max mem: 3652
Epoch: [0] [ 320/8562] eta: 0:46:59 lr: 0.000010 class_error: 0.00
loss: 2.6354 (3.9228) loss_ce: 0.0655 (0.2124) loss_bbox: 0.2235
(0.2043) loss_giou: 0.1263 (0.1980) loss_ce_0: 0.0966 (0.2465)
loss_bbox_0: 0.2522 (0.2611) loss_giou_0: 0.1803 (0.2453) loss_ce_1:
0.0820 (0.2265) loss_bbox_1: 0.1847 (0.2302) loss_giou_1: 0.1192
(0.2193) loss_ce_2: 0.0756 (0.2152) loss_bbox_2: 0.2226 (0.2137)
loss_giou_2: 0.1308 (0.2066) loss_ce_3: 0.0772 (0.2140) loss_bbox_3:
0.2073 (0.2094) loss_giou_3: 0.1223 (0.2021) loss_ce_4: 0.0687 (0.2118)
loss_bbox_4: 0.2084 (0.2070) loss_giou_4: 0.1296 (0.1994)
loss_ce_unscaled: 0.0655 (0.2124) class_error_unscaled: 0.0000 (18.7957)
loss_bbox_unscaled: 0.0447 (0.0409) loss_giou_unscaled: 0.0632 (0.0990)
cardinality_error_unscaled: 1.0000 (2.7601) loss_ce_0_unscaled: 0.0966
(0.2465) loss_bbox_0_unscaled: 0.0504 (0.0522) loss_giou_0_unscaled:
0.0902 (0.1227) cardinality_error_0_unscaled: 2.0000 (3.8738)
loss_ce_1_unscaled: 0.0820 (0.2265) loss_bbox_1_unscaled: 0.0369 (0.0460)
loss_giou_1_unscaled: 0.0596 (0.1097) cardinality_error_1_unscaled:
1.5000 (3.0685) loss_ce_2_unscaled: 0.0756 (0.2152)
loss_bbox_2_unscaled: 0.0445 (0.0427) loss_giou_2_unscaled: 0.0654
(0.1033) cardinality_error_2_unscaled: 1.0000 (2.9844)
loss_ce_3_unscaled: 0.0772 (0.2140) loss_bbox_3_unscaled: 0.0415 (0.0419)
loss_giou_3_unscaled: 0.0612 (0.1011) cardinality_error_3_unscaled:
1.0000 (2.9128) loss_ce_4_unscaled: 0.0687 (0.2118)
loss_bbox_4_unscaled: 0.0417 (0.0414) loss_giou_4_unscaled: 0.0648
(0.0997) cardinality_error_4_unscaled: 1.0000 (2.7835) time: 0.3444
data: 0.0153 max mem: 3652
Epoch: [0] [ 330/8562] eta: 0:46:54 lr: 0.000010 class_error: 33.33
loss: 2.8610 (3.9142) loss_ce: 0.0776 (0.2102) loss_bbox: 0.2096
(0.2053) loss_giou: 0.1422 (0.1976) loss_ce_0: 0.1122 (0.2442)
loss_bbox_0: 0.2491 (0.2627) loss_giou_0: 0.1806 (0.2453) loss_ce_1:
0.1181 (0.2235) loss_bbox_1: 0.2053 (0.2312) loss_giou_1: 0.1493
(0.2189) loss_ce_2: 0.1037 (0.2129) loss_bbox_2: 0.2079 (0.2149)
loss_giou_2: 0.1511 (0.2067) loss_ce_3: 0.0828 (0.2118) loss_bbox_3:
```

```
0.2112 (0.2106) loss_giou_3: 0.1472 (0.2019) loss_ce_4: 0.0781 (0.2095)
loss_bbox_4: 0.2195 (0.2079) loss_giou_4: 0.1394 (0.1990)
loss_ce_unscaled: 0.0776 (0.2102) class_error_unscaled: 0.0000 (18.6407)
loss_bbox_unscaled: 0.0419 (0.0411) loss_giou_unscaled: 0.0711 (0.0988)
cardinality_error_unscaled: 0.5000 (2.7190) loss_ce_0_unscaled: 0.1122
(0.2442) loss_bbox_0_unscaled: 0.0498 (0.0525) loss_giou_0_unscaled:
0.0903 (0.1227) cardinality_error_0_unscaled: 1.0000 (3.8127)
loss_ce_1_unscaled: 0.1181 (0.2235) loss_bbox_1_unscaled: 0.0411 (0.0462)
loss_giou_1_unscaled: 0.0746 (0.1094) cardinality_error_1_unscaled:
1.0000 (3.0317) loss_ce_2_unscaled: 0.1037 (0.2129)
loss_bbox_2_unscaled: 0.0416 (0.0430) loss_giou_2_unscaled: 0.0756
(0.1033) cardinality_error_2_unscaled: 1.0000 (2.9486)
loss_ce_3_unscaled: 0.0828 (0.2118) loss_bbox_3_unscaled: 0.0422 (0.0421)
loss_giou_3_unscaled: 0.0736 (0.1009) cardinality_error_3_unscaled:
1.0000 (2.8792) loss_ce_4_unscaled: 0.0781 (0.2095)
loss_bbox_4_unscaled: 0.0439 (0.0416) loss_giou_4_unscaled: 0.0697
(0.0995) cardinality_error_4_unscaled: 1.0000 (2.7462) time: 0.3403
data: 0.0127 max mem: 3652
Epoch: [0] [ 340/8562] eta: 0:46:46 lr: 0.000010 class_error: 0.00
loss: 3.3288 (3.9227) loss_ce: 0.1050 (0.2099) loss_bbox: 0.2058
(0.2057) loss_giou: 0.2017 (0.1990) loss_ce_0: 0.1608 (0.2447)
loss_bbox_0: 0.2364 (0.2621) loss_giou_0: 0.2463 (0.2458) loss_ce_1:
0.1404 (0.2238) loss_bbox_1: 0.2216 (0.2314) loss_giou_1: 0.2385
(0.2202) loss_ce_2: 0.1286 (0.2132) loss_bbox_2: 0.2181 (0.2152)
loss_giou_2: 0.2219 (0.2078) loss_ce_3: 0.1183 (0.2117) loss_bbox_3:
0.2092 (0.2109) loss_giou_3: 0.1819 (0.2031) loss_ce_4: 0.1132 (0.2093)
loss_bbox_4: 0.2149 (0.2083) loss_giou_4: 0.2080 (0.2006)
loss_ce_unscaled: 0.1050 (0.2099) class_error_unscaled: 0.0000 (18.4118)
loss_bbox_unscaled: 0.0412 (0.0411) loss_giou_unscaled: 0.1008 (0.0995)
cardinality_error_unscaled: 1.0000 (2.7317) loss_ce_0_unscaled: 0.1608
(0.2447) loss_bbox_0_unscaled: 0.0473 (0.0524) loss_giou_0_unscaled:
0.1232 (0.1229) cardinality_error_0_unscaled: 2.0000 (3.8226)
loss_ce_1_unscaled: 0.1404 (0.2238) loss_bbox_1_unscaled: 0.0443 (0.0463)
loss_giou_1_unscaled: 0.1193 (0.1101) cardinality_error_1_unscaled:
1.0000 (3.0367) loss_ce_2_unscaled: 0.1286 (0.2132)
loss_bbox_2_unscaled: 0.0436 (0.0430) loss_giou_2_unscaled: 0.1109
(0.1039) cardinality_error_2_unscaled: 1.0000 (2.9531)
loss_ce_3_unscaled: 0.1183 (0.2117) loss_bbox_3_unscaled: 0.0418 (0.0422)
loss_giou_3_unscaled: 0.0909 (0.1015) cardinality_error_3_unscaled:
1.0000 (2.8798) loss_ce_4_unscaled: 0.1132 (0.2093)
loss_bbox_4_unscaled: 0.0430 (0.0417) loss_giou_4_unscaled: 0.1040
(0.1003) cardinality_error_4_unscaled: 1.0000 (2.7507) time: 0.3299
data: 0.0134 max mem: 3652
Epoch: [0] [ 350/8562] eta: 0:46:50 lr: 0.000010 class_error: 0.00
loss: 4.1889 (3.9958) loss_ce: 0.1588 (0.2136) loss_bbox: 0.2058
(0.2091) loss_giou: 0.2232 (0.2046) loss_ce_0: 0.2293 (0.2487)
loss_bbox_0: 0.2399 (0.2637) loss_giou_0: 0.2468 (0.2516) loss_ce_1:
```

```
0.2172 (0.2286) loss_bbox_1: 0.2324 (0.2335) loss_giou_1: 0.2725  
(0.2254) loss_ce_2: 0.2589 (0.2170) loss_bbox_2: 0.2284 (0.2182)  
loss_giou_2: 0.2407 (0.2130) loss_ce_3: 0.2419 (0.2157) loss_bbox_3:  
0.2092 (0.2143) loss_giou_3: 0.2162 (0.2084) loss_ce_4: 0.1534 (0.2129)  
loss_bbox_4: 0.2149 (0.2117) loss_giou_4: 0.2232 (0.2060)  
loss_ce_unscaled: 0.1588 (0.2136) class_error_unscaled: 10.0000 (18.4376)  
loss_bbox_unscaled: 0.0412 (0.0418) loss_giou_unscaled: 0.1116 (0.1023)  
cardinality_error_unscaled: 1.5000 (2.7066) loss_ce_0_unscaled: 0.2293  
(0.2487) loss_bbox_0_unscaled: 0.0480 (0.0527) loss_giou_0_unscaled:  
0.1234 (0.1258) cardinality_error_0_unscaled: 2.0000 (3.8048)  
loss_ce_1_unscaled: 0.2172 (0.2286) loss_bbox_1_unscaled: 0.0465 (0.0467)  
loss_giou_1_unscaled: 0.1362 (0.1127) cardinality_error_1_unscaled:  
1.5000 (3.0171) loss_ce_2_unscaled: 0.2589 (0.2170)  
loss_bbox_2_unscaled: 0.0457 (0.0436) loss_giou_2_unscaled: 0.1203  
(0.1065) cardinality_error_2_unscaled: 1.5000 (2.9402)  
loss_ce_3_unscaled: 0.2419 (0.2157) loss_bbox_3_unscaled: 0.0418 (0.0429)  
loss_giou_3_unscaled: 0.1081 (0.1042) cardinality_error_3_unscaled:  
2.0000 (2.8647) loss_ce_4_unscaled: 0.1534 (0.2129)  
loss_bbox_4_unscaled: 0.0430 (0.0423) loss_giou_4_unscaled: 0.1116  
(0.1030) cardinality_error_4_unscaled: 1.5000 (2.7251) time: 0.3477  
data: 0.0160 max mem: 3652  
Epoch: [0] [ 360/8562] eta: 0:46:48 lr: 0.000010 class_error: 0.00  
loss: 3.1412 (3.9867) loss_ce: 0.1154 (0.2122) loss_bbox: 0.1887  
(0.2089) loss_giou: 0.1967 (0.2048) loss_ce_0: 0.2170 (0.2485)  
loss_bbox_0: 0.2108 (0.2627) loss_giou_0: 0.2323 (0.2507) loss_ce_1:  
0.1543 (0.2281) loss_bbox_1: 0.1944 (0.2330) loss_giou_1: 0.1977  
(0.2250) loss_ce_2: 0.1624 (0.2165) loss_bbox_2: 0.1763 (0.2175)  
loss_giou_2: 0.1966 (0.2129) loss_ce_3: 0.1424 (0.2148) loss_bbox_3:  
0.1811 (0.2138) loss_giou_3: 0.1699 (0.2081) loss_ce_4: 0.1412 (0.2118)  
loss_bbox_4: 0.1761 (0.2112) loss_giou_4: 0.1773 (0.2060)  
loss_ce_unscaled: 0.1154 (0.2122) class_error_unscaled: 0.0000 (18.2676)  
loss_bbox_unscaled: 0.0377 (0.0418) loss_giou_unscaled: 0.0983 (0.1024)  
cardinality_error_unscaled: 1.0000 (2.6981) loss_ce_0_unscaled: 0.2170  
(0.2485) loss_bbox_0_unscaled: 0.0422 (0.0525) loss_giou_0_unscaled:  
0.1161 (0.1254) cardinality_error_0_unscaled: 2.0000 (3.7895)  
loss_ce_1_unscaled: 0.1543 (0.2281) loss_bbox_1_unscaled: 0.0389 (0.0466)  
loss_giou_1_unscaled: 0.0988 (0.1125) cardinality_error_1_unscaled:  
1.5000 (3.0111) loss_ce_2_unscaled: 0.1624 (0.2165)  
loss_bbox_2_unscaled: 0.0353 (0.0435) loss_giou_2_unscaled: 0.0983  
(0.1065) cardinality_error_2_unscaled: 1.5000 (2.9349)  
loss_ce_3_unscaled: 0.1424 (0.2148) loss_bbox_3_unscaled: 0.0362 (0.0428)  
loss_giou_3_unscaled: 0.0849 (0.1041) cardinality_error_3_unscaled:  
1.5000 (2.8643) loss_ce_4_unscaled: 0.1412 (0.2118)  
loss_bbox_4_unscaled: 0.0352 (0.0422) loss_giou_4_unscaled: 0.0887  
(0.1030) cardinality_error_4_unscaled: 1.0000 (2.7161) time: 0.3611  
data: 0.0158 max mem: 3652
```

DONE (t=30.89s).

IoU metric: bbox

Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.617
Average Precision (AP) @[IoU=0.50 | area= all | maxDets=100] = 0.773
Average Precision (AP) @[IoU=0.75 | area= all | maxDets=100] = 0.672
Average Precision (AP) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.184
Average Precision (AP) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.438
Average Precision (AP) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.726
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 1] = 0.524
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 10] = 0.774
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.811
Average Recall (AR) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.399
Average Recall (AR) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.680
Average Recall (AR) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.900

```
import math

from PIL import Image
import requests
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'

import ipywidgets as widgets
from IPython.display import display, clear_output

import torch
from torch import nn
from torchvision.models import resnet50
import torchvision.transforms as T
torch.set_grad_enabled(False);

# COCO classes
CLASSES = [
    'N/A', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
    'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A',
    'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse',
    'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack',
    'umbrella', 'N/A', 'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis',
    'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove',
```

```

'skateboard', 'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass',
'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich',
'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake',
'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table', 'N/A',
'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard',
'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A',
'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier',
'toothbrush'
]

# colors for visualization
COLORS = [[0.000, 0.447, 0.741], [0.850, 0.325, 0.098], [0.929, 0.694, 0.125],
[0.494, 0.184, 0.556], [0.466, 0.674, 0.188], [0.301, 0.745, 0.933]]


# standard PyTorch mean-std input image normalization
transform = T.Compose([
    T.Resize(800),
    T.ToTensor(),
    T.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

# for output bounding box post-processing
def box_cxcywh_to_xyxy(x):
    x_c, y_c, w, h = x.unbind(1)
    b = [(x_c - 0.5 * w), (y_c - 0.5 * h),
          (x_c + 0.5 * w), (y_c + 0.5 * h)]
    return torch.stack(b, dim=1)

def rescale_bboxes(out_bbox, size):
    img_w, img_h = size
    b = box_cxcywh_to_xyxy(out_bbox)
    b = b * torch.tensor([img_w, img_h, img_w, img_h], dtype=torch.float32)
    return b

def plot_results(pil_img, prob, boxes):
    plt.figure(figsize=(16,10))
    plt.imshow(pil_img)
    ax = plt.gca()
    colors = COLORS * 100
    for p, (xmin, ymin, xmax, ymax), c in zip(prob, boxes.tolist(), colors):
        ax.add_patch(plt.Rectangle((xmin, ymin), xmax - xmin, ymax - ymin,
                                  fill=False, color=c, linewidth=3))
    cl = p.argmax()

```

```
text = f'{CLASSES[cl]}: {p[cl]:0.2f}'  
ax.text(xmin, ymin, text, fontsize=15,  
        bbox=dict(facecolor='yellow', alpha=0.5))  
plt.axis('off')  
plt.show()  
checkpoint = torch.load('/content/checkpoint.pth')  
print(checkpoint.keys()) # See what keys are in the checkpoint  
  
model = torch.hub.load('facebookresearch/detr', 'detr_resnet50')  
checkpoint = torch.load('/content/checkpoint.pth')  
model.load_state_dict(checkpoint['model'])  
model.eval()  
  
Downloading: "https://github.com/facebookresearch/detr/zipball/main" to  
/root/.cache/torch/hub/main.zip  
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208:  
UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may  
be removed in the future, please use 'weights' instead.  
    warnings.warn(  
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223:  
UserWarning: Arguments other than a weight enum or `None` for 'weights'  
are deprecated since 0.13 and may be removed in the future. The current  
behavior is equivalent to passing  
`weights=ResNet50_Weights.IMAGENET1K_V1`. You can also use  
`weights=ResNet50_Weights.DEFAULT` to get the most up-to-date weights.  
    warnings.warn(msg)  
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, 165MB/s]  
<ipython-input-12-5f63326786dd>:2: FutureWarning: You are using  
`torch.load` with `weights_only=False` (the current default value), which  
uses the default pickle module implicitly. It is possible to construct  
malicious pickle data which will execute arbitrary code during unpickling  
(See  
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models  
for more details). In a future release, the default value for  
`weights_only` will be flipped to `True`. This limits the functions that  
could be executed during unpickling. Arbitrary objects will no longer be  
allowed to be loaded via this mode unless they are explicitly allowlisted  
by the user via `torch.serialization.add_safe_globals`. We recommend you  
start setting `weights_only=True` for any use case where you don't have  
full control of the loaded file. Please open an issue on GitHub for any  
issues related to this experimental feature.  
checkpoint = torch.load('/content/checkpoint.pth')
```



```
        (relu) : ReLU(inplace=True)
    )
)
)
)
(1) : PositionEmbeddingSine()
)
)
```

```
url = 'http://images.cocodataset.org/val2017/000000039769.jpg'
im = Image.open(requests.get(url, stream=True).raw)
# mean-std normalize the input image (batch-size: 1)
img = transform(im).unsqueeze(0)

# propagate through the model
outputs = model(img)

# keep only predictions with 0.7+ confidence
probas = outputs['pred_logits'].softmax(-1)[0, :, :-1]
keep = probas.max(-1).values > 0.9

# convert boxes from [0; 1] to image scales
bboxes_scaled = rescale_bboxes(outputs['pred_boxes'][0, keep], im.size)
```

```
plot_results(im,
```

```
    couch: 0.95
```

```
    cat: 0.99
```



```
    cat: 1.00
```



```
    probas[keep], bboxes_scaled)
```

```
# use lists to store the outputs via up-values
```

```
conv_features, enc_attn_weights, dec_attn_weights = [], [], []
```

```
hooks = [
```

```
    model.backbone[-2].register_forward_hook(  
        lambda self, input, output: conv_features.append(output)  
)  
    model.transformer.encoder.layers[-1].self_attn.register_forward_hook(  
        lambda self, input, output: enc_attn_weights.append(output[1])  
)  
    model.transformer.decoder.layers[-1].multihead_attn.register_forward_hook(  
        lambda self, input, output: dec_attn_weights.append(output[1])  
)  
]
```

```

# propagate through the model
outputs = model(img)

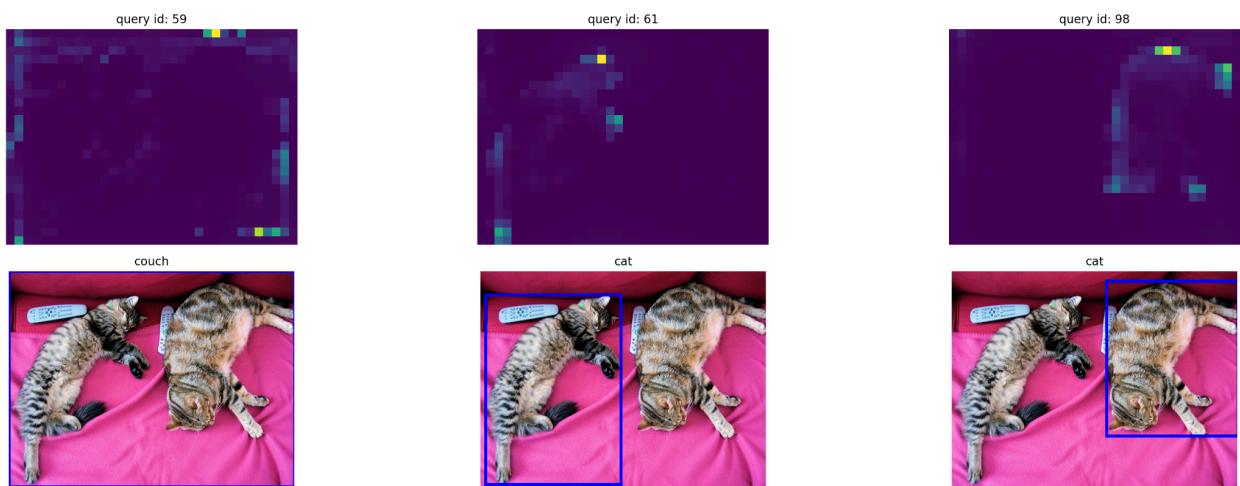
for hook in hooks:
    hook.remove()

# don't need the list anymore
conv_features = conv_features[0]
enc_attn_weights = enc_attn_weights[0]
dec_attn_weights = dec_attn_weights[0]

# get the feature map shape
h, w = conv_features['0'].tensors.shape[-2:]

fig, axs = plt.subplots(ncols=len(bboxes_scaled), nrows=2, figsize=(22, 7))
colors = COLORS * 100
for idx, ax_i, (xmin, ymin, xmax, ymax) in zip(keep.nonzero(), axs.T, bboxes_scaled):
    ax = ax_i[0]
    ax.imshow(dec_attn_weights[0, idx].view(h, w))
    ax.axis('off')
    ax.set_title(f'query id: {idx.item()}')
    ax = ax_i[1]
    ax.imshow(im)
    ax.add_patch(plt.Rectangle((xmin, ymin), xmax - xmin, ymax - ymin,
                               fill=False, color='blue', linewidth=3))
    ax.axis('off')
    ax.set_title(CLASSES[probas[idx].argmax()])
fig.tight_layout()

```



```
# output of the CNN
f_map = conv_features[0]
print("Encoder attention: ", enc_attn_weights[0].shape)
print("Feature map: ", f_map.tensors.shape)
Encoder attention: torch.Size([850, 850])
Feature map: torch.Size([1, 2048, 25, 34])
```

```
# get the HxW shape of the feature maps of the CNN
shape = f_map.tensors.shape[-2:]
# and reshape the self-attention to a more interpretable shape
sattn = enc_attn_weights[0].reshape(shape + shape)
print("Reshaped self-attention:", sattn.shape)
```

```
# downsampling factor for the CNN, is 32 for DETR and 16 for DETR DC5
fact = 32
```

```
# let's select 4 reference points for visualization
idxs = [(200, 200), (280, 400), (200, 600), (440, 800), ]
```

```
# here we create the canvas
fig = plt.figure(constrained_layout=True, figsize=(25 * 0.7, 8.5 * 0.7))
# and we add one plot per reference point
gs = fig.add_gridspec(2, 4)
axs = [
    fig.add_subplot(gs[0, 0]),
    fig.add_subplot(gs[1, 0]),
    fig.add_subplot(gs[0, -1]),
    fig.add_subplot(gs[1, -1]),
]
```

```
# for each one of the reference points, let's plot the self-attention
# for that point
for idx_o, ax in zip(idxs, axs):
    idx = (idx_o[0] // fact, idx_o[1] // fact)
    ax.imshow(sattn[..., idx[0], idx[1]], cmap='cividis',
interpolation='nearest')
    ax.axis('off')
    ax.set_title(f'self-attention{idx_o}')
```

```
# and now let's add the central image, with the reference points as red
circles
fcenter_ax = fig.add_subplot(gs[:, 1:-1])
fcenter_ax.imshow(im)
for (y, x) in idxs:
    scale = im.height / img.shape[-2]
```

```

x = ((x // fact) + 0.5) * fact
y = ((y // fact) + 0.5) * fact
fcenter_ax.add_patch(plt.Circle((x * scale, y * scale), fact // 2,
color='r'))
fcenter_ax.axis('off')

```



```

class AttentionVisualizer:
    def __init__(self, model, transform):
        self.model = model
        self.transform = transform

        self.url = ""
        self.cur_url = None
        self.pil_img = None
        self.tensor_img = None

        self.conv_features = None
        self.enc_attn_weights = None
        self.dec_attn_weights = None

    def setup_widgets():
        self.sliders = [
            widgets.Text(
                value='http://images.cocodataset.org/val2017/000000039769.jpg',
                placeholder='Type something',
                description='URL (ENTER):',

```

```

        disabled=False,
        continuous_update=False,
        layout=widgets.Layout(width='100%')
    ),
    widgets.FloatSlider(min=0, max=0.99,
                         step=0.02, description='X coordinate', value=0.72,
                         continuous_update=False,
                         layout=widgets.Layout(width='50%')
    ),
    widgets.FloatSlider(min=0, max=0.99,
                         step=0.02, description='Y coordinate', value=0.40,
                         continuous_update=False,
                         layout=widgets.Layout(width='50%')),
    widgets.Checkbox(
        value=False,
        description='Direction of self attention',
        disabled=False,
        indent=False,
        layout=widgets.Layout(width='50%'),
    ),
    widgets.Checkbox(
        value=True,
        description='Show red dot in attention',
        disabled=False,
        indent=False,
        layout=widgets.Layout(width='50%'),
    )
]
self.o = widgets.Output()

def compute_features(self, img):
    model = self.model
    # use lists to store the outputs via up-values
    conv_features, enc_attn_weights, dec_attn_weights = [], [], []

    hooks = [
        model.backbone[-2].register_forward_hook(
            lambda self, input, output: conv_features.append(output)
        ),

```

```

model.transformer.encoder.layers[-1].self_attn.register_forward_hook(
    lambda self, input, output:
enc_attn_weights.append(output[1])
) , 

model.transformer.decoder.layers[-1].multihead_attn.register_forward_hook(
    lambda self, input, output:
dec_attn_weights.append(output[1])
) ,
]

# propagate through the model
outputs = model(img)

for hook in hooks:
    hook.remove()

# don't need the list anymore
self.conv_features = conv_features[0]
self.dec_attn_weights = dec_attn_weights[0]
# get the HxW shape of the feature maps of the CNN
shape = self.conv_features['0'].tensors.shape[-2:]
# and reshape the self-attention to a more interpretable shape
self.enc_attn_weights = enc_attn_weights[0].reshape(shape + shape)

def compute_on_image(self, url):
    if url != self.url:
        self.url = url
        self.pil_img = Image.open(requests.get(url, stream=True).raw)
        # mean-std normalize the input image (batch-size: 1)
        self.tensor_img = self.transform(self.pil_img).unsqueeze(0)
        self.compute_features(self.tensor_img)

def update_chart(self, change):
    with self.o:
        clear_output()

        # j and i are the x and y coordinates of where to look at
        # sattn_dir is which direction to consider in the
self-attention matrix

```

```

        # sattn_dot displays a red dot or not in the self-attention map
        url, j, i, sattn_dir, sattn_dot = [s.value for s in
self.sliders]

        fig, axs = plt.subplots(ncols=2, nrows=1, figsize=(9, 4))
        self.compute_on_image(url)

        # convert reference point to absolute coordinates
        j = int(j * self.tensor_img.shape[-1])
        i = int(i * self.tensor_img.shape[-2])

        # how much was the original image upsampled before feeding it
to the model
        scale = self.pil_img.height / self.tensor_img.shape[-2]

        # compute the downsampling factor for the model
        # it should be 32 for standard DETR and 16 for DC5
        sattn = self.enc_attn_weights
        fact = 2 ** round(math.log2(self.tensor_img.shape[-1] /
sattn.shape[-1]))

        # round the position at the downsampling factor
        x = ((j // fact) + 0.5) * fact
        y = ((i // fact) + 0.5) * fact

        axs[0].imshow(self.pil_img)
        axs[0].axis('off')
        axs[0].add_patch(plt.Circle((x * scale, y * scale), fact // 2,
color='r'))

        idx = (i // fact, j // fact)

        if sattn_dir:
            sattn_map = sattn[idx[0], idx[1], ...]
        else:
            sattn_map = sattn[..., idx[0], idx[1]]

        axs[1].imshow(sattn_map, cmap='cividis',
interpolation='nearest')
        if sattn_dot:

```

```

        axs[1].add_patch(plt.Circle((idx[1], idx[0]), 1, color='r'))
        axs[1].axis('off')
        axs[1].set_title(f'self-attention{(i, j)}')

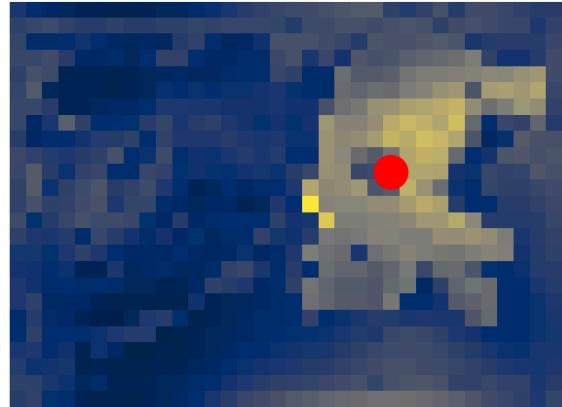
    plt.show()

def run(self):
    for s in self.sliders:
        s.observe(self.update_chart, 'value')
    self.update_chart(None)
    url, x, y, d, sattn_d = self.sliders
    res = widgets.VBox([
        [
            url,
            widgets.HBox([x, y]),
            widgets.HBox([d, sattn_d]),
            self.o
        ]
    ])
    return res
w = AttentionVisualizer(model, transform)
w.run()

```



self-attention(320, 767)



Qualitative and Quantitative Comparison:

1. Detectron Results (VOC 2012 Validation):
 - Average Precision (AP): 49.84%
 - AP at IoU 50%: 74.87%
 - AP at IoU 75%: 57.37%
2. DETR Results:
 - Overall Average Precision (AP): 61.7%
 - AP at IoU 50%: 77.3%
 - AP at IoU 75%: 67.2%

Performance Breakdown:

- Small Objects AP: DETR at 18.4%
- Medium Objects AP: DETR at 43.8%
- Large Objects AP: DETR at 72.6%

Key Observations:

1. DETR shows superior overall performance across most metrics
2. Both models perform better at lower IoU thresholds (50%)
3. DETR demonstrates stronger performance on large objects
4. DETR has better average recall, especially for multiple object detection

The Qualitative results have been shown above along with the code.

