faster-r-cnn-epoch-2000

August 24, 2023

[ ]:
```
!python -m 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-azvqnd2k
  Running command git clone --filter=blob:none --quiet
https://github.com/facebookresearch/detectron2.git /tmp/pip-req-build-azvqnd2k
  Resolved https://github.com/facebookresearch/detectron2.git to commit
a2e43eab54d28ffbd59f5e9b4e3193b82faeb70f
  Preparing metadata (setup.py) … done
Requirement already satisfied: Pillow>=7.1 in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (8.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (3.7.1)
Requirement already satisfied: pycocotools>=2.0.2 in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (2.0.6)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (2.3.0)
Collecting yacs>=0.1.8 (from detectron2==0.6)
  Downloading yacs-0.1.8-py3-none-any.whl (14 kB)
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) (2.2.1)
Requirement already satisfied: tqdm>4.29.0 in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (4.65.0)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.10/dist-
packages (from detectron2==0.6) (2.12.3)
Collecting fvcore<0.1.6,>=0.1.5 (from detectron2==0.6)
  Downloading fvcore-0.1.5.post20221221.tar.gz (50 kB)
                           50.2/50.2 kB
1.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting iopath<0.1.10,>=0.1.7 (from detectron2==0.6)
  Downloading iopath-0.1.9-py3-none-any.whl (27 kB)
Collecting omegaconf>=2.1 (from detectron2==0.6)
  Downloading omegaconf-2.3.0-py3-none-any.whl (79 kB)
                           79.5/79.5 kB
```

```
5.5 MB/s eta 0:00:00
Collecting hydra-core>=1.1 (from detectron2==0.6)
  Downloading hydra_core-1.3.2-py3-none-any.whl (154 kB)
                           154.5/154.5 kB
15.2 MB/s eta 0:00:00
Collecting black (from detectron2==0.6)
  Downloading
black-23.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
                            1.7/1.7 MB
59.1 MB/s eta 0:00:00
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from detectron2==0.6) (23.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from fvcore<0.1.6,>=0.1.5->detectron2==0.6) (1.22.4)
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) (6.0.1)
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
17.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting portalocker (from iopath<0.1.10,>=0.1.7->detectron2==0.6)
  Downloading portalocker-2.7.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2==0.6)
(1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->detectron2==0.6) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2==0.6)
(4.41.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2==0.6)
(1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->detectron2==0.6)
(3.1.0)
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) (8.1.6)
Collecting mypy-extensions>=0.4.3 (from black->detectron2==0.6)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Collecting pathspec>=0.9.0 (from black->detectron2==0.6)
  Downloading pathspec-0.11.1-py3-none-any.whl (29 kB)
Requirement already satisfied: platformdirs>=2 in
/usr/local/lib/python3.10/dist-packages (from black->detectron2==0.6) (3.9.1)
```

```
Requirement already satisfied: tomli>=1.1.0 in /usr/local/lib/python3.10/dist-
packages (from black->detectron2==0.6) (2.0.1)
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) (1.56.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(2.17.3)
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(1.0.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(3.4.3)
Requirement already satisfied: protobuf>=3.19.6 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(2.27.1)
Requirement already satisfied: setuptools>=41.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(67.7.2)
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)
(0.7.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard->detectron2==0.6)
(2.3.6)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.10/dist-
packages (from tensorboard->detectron2==0.6) (0.40.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-
auth<3,>=1.6.3->tensorboard->detectron2==0.6) (5.3.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-
auth<3,>=1.6.3->tensorboard->detectron2==0.6) (0.3.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-
packages (from google-auth<3,>=1.6.3->tensorboard->detectron2==0.6) (1.16.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-
packages (from google-auth<3,>=1.6.3->tensorboard->detectron2==0.6) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<1.1,>=0.5->tensorboard->detectron2==0.6) (1.3.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2==0.6) (1.26.16)
```

```
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2==0.6) (2023.5.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2==0.6) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.21.0->tensorboard->detectron2==0.6) (3.4)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard->detectron2==0.6) (2.1.3)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard->detectron2==0.6) (0.5.0)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib<1.1,>=0.5->tensorboard->detectron2==0.6) (3.2.2)
Building wheels for collected packages: detectron2, fvcore,
antlr4-python3-runtime
  Building wheel for detectron2 (setup.py) … done
  Created wheel for detectron2:
filename=detectron2-0.6-cp310-cp310-linux_x86_64.whl size=6111766
sha256=2f1b7bc10f1697a9549bff725d11654b0fc5e0a07f06203b41ed8f5ff3855a9e
  Stored in directory: /tmp/pip-ephem-wheel-cache-
qqm1mfp1/wheels/47/e5/15/94c80df2ba85500c5d76599cc307c0a7079d0e221bb6fc4375
  Building wheel for fvcore (setup.py) … done
  Created wheel for fvcore: filename=fvcore-0.1.5.post20221221-py3-none-any.whl
size=61405
sha256=266bab7527c4345665204d148af9b7877f55cb8cad9c2250d9ad34a0890f0532
  Stored in directory: /root/.cache/pip/wheels/01/c0/af/77c1cf53a1be9e42a52b48e5
af2169d40ec2e89f7362489dd0
  Building wheel for antlr4-python3-runtime (setup.py) … done
  Created wheel for antlr4-python3-runtime:
filename=antlr4_python3_runtime-4.9.3-py3-none-any.whl size=144554
sha256=df5bb2c698ca289400fd30a7edd483c8d3d26fc6dae7a21f6546e01ff738033e
  Stored in directory: /root/.cache/pip/wheels/12/93/dd/1f6a127edc45659556564c57
30f6d4e300888f4bca2d4c5a88
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-23.7.0 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-0.11.1 portalocker-2.7.0 yacs-0.1.8
```

[ ]:
```
!python -m pip install pyyaml==5.1
```

```
Collecting pyyaml==5.1
```

```
  Downloading PyYAML-5.1.tar.gz (274 kB)
                              274.2/274.2
kB 4.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Building wheels for collected packages: pyyaml
  Building wheel for pyyaml (setup.py) … done
  Created wheel for pyyaml: filename=PyYAML-5.1-cp310-cp310-linux_x86_64.whl
size=44090
sha256=b311b1f4ddf117483a7daf69c189984f91e1d0cdd108ac47e4db4d86a5a2e639
  Stored in directory: /root/.cache/pip/wheels/70/83/31/975b737609aba39a4099d471
d5684141c1fdc3404f97e7f68a
Successfully built pyyaml
Installing collected packages: pyyaml
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 6.0.1
    Uninstalling PyYAML-6.0.1:
      Successfully uninstalled PyYAML-6.0.1
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
dask 2022.12.1 requires pyyaml>=5.3.1, but you have pyyaml 5.1 which is
incompatible.
flax 0.7.0 requires PyYAML>=5.4.1, but you have pyyaml 5.1 which is
incompatible.
Successfully installed pyyaml-5.1
```

[ ]:
```
import torch, detectron2
!nvcc --version
TORCH_VERSION = ".".join(torch.__version__.split(".")[:2])
CUDA_VERSION = torch.__version__.split("+")[-1]
print("torch: ", TORCH_VERSION, "; cuda: ", CUDA_VERSION)
print("detectron2:", detectron2.__version__)
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
torch:  2.0 ; cuda:  cu118
detectron2: 0.6
```

[ ]:
```
import detectron2
from detectron2.utils.logger import setup_logger
```

```
setup_logger()

# import some common libraries
import numpy as np
import cv2
import matplotlib.pyplot as plt

# import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog, DatasetCatalog
```

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

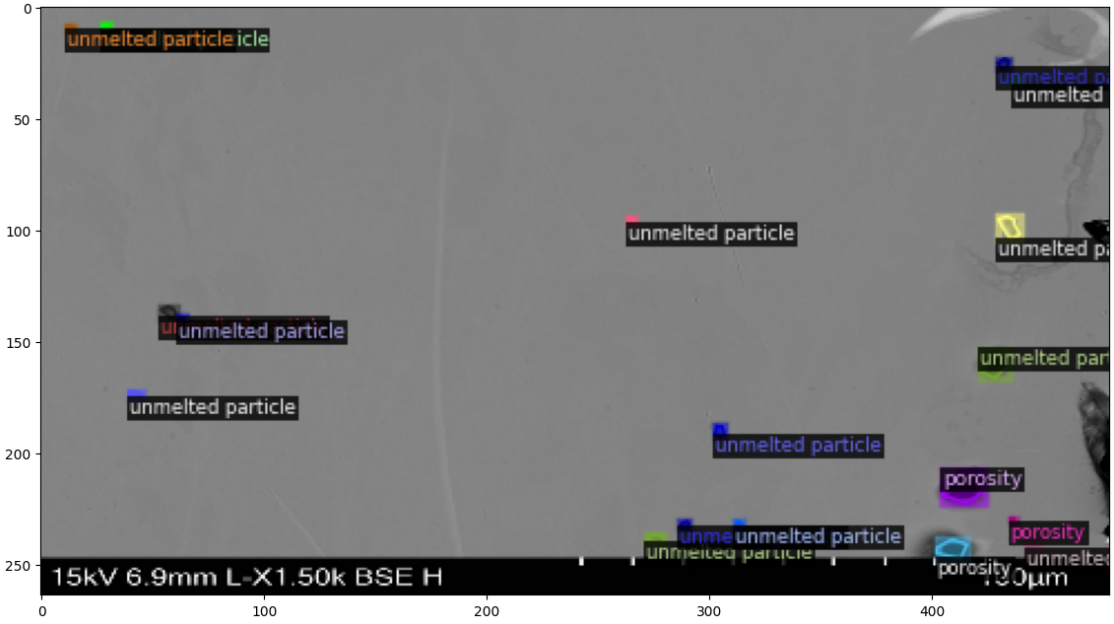
```
Mounted at /content/drive
```

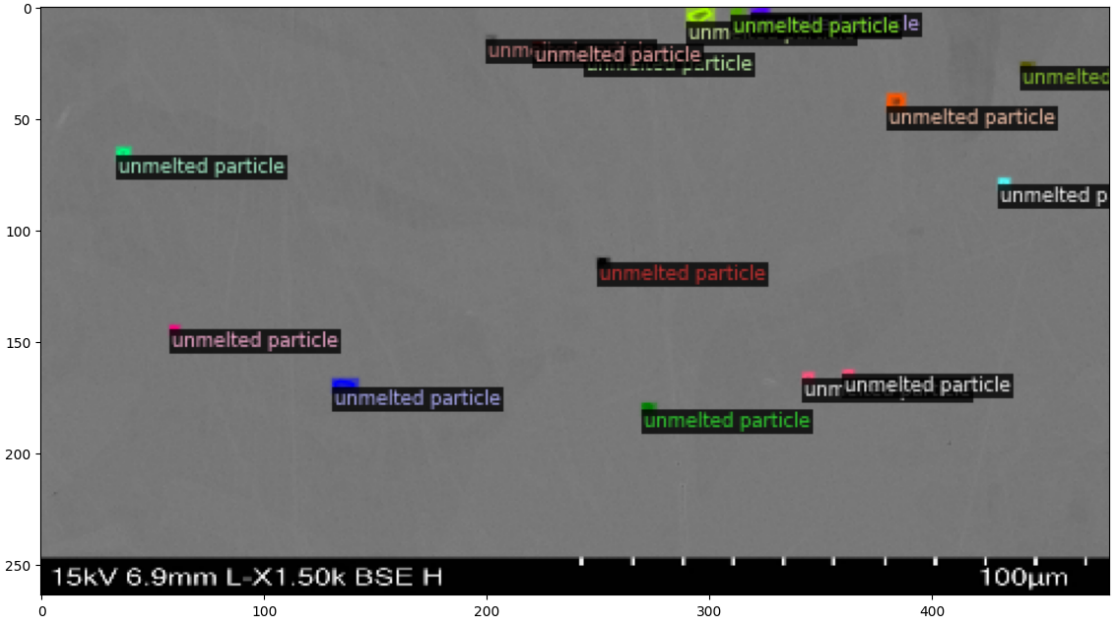
[ ]:
```
DatasetCatalog.remove("p_train")
DatasetCatalog.remove("p_test")
```

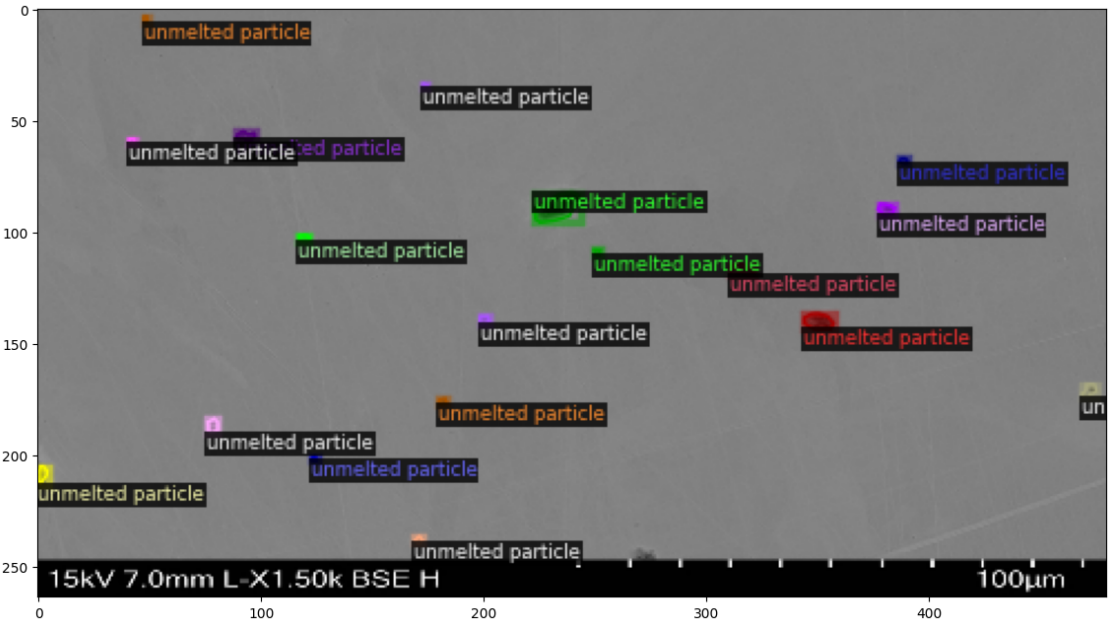
[ ]:
```
import os
import numpy as np
import json
from detectron2.structures import BoxMode

def get_r_dicts(directory):

    classes = ['unmelted particle', 'porosity', 'microcrack']
    dataset_dicts = []
    for idx, filename in enumerate([file for file in os.listdir(directory) if␣
 ↪file.endswith('.json')]):
        json_file = os.path.join(directory, filename)
        with open(json_file) as f:
            img_anns = json.load(f)

        record = {}

        filename = os.path.join(directory, img_anns["imagePath"])

        record["file_name"] = filename
        record["image_id"] = idx
        record["height"] = 528
        record["width"] = 960

        annos = img_anns["shapes"]
```

```
        objs = []
        for anno in annos:
            px = [a[0] for a in anno['points']]
            py = [a[1] for a in anno['points']]
            poly = [(x, y) for x, y in zip(px, py)]
            poly = [p for x in poly for p in x]
            obj = {
                "bbox": [np.min(px), np.min(py), np.max(px), np.max(py)],
                "bbox_mode": BoxMode.XYXY_ABS,
                "segmentation": [poly],
                "category_id": classes.index(anno['label']),
                "iscrowd": 0
            }
            objs.append(obj)
        record["annotations"] = objs
        dataset_dicts.append(record)
    return dataset_dicts

from detectron2.data import DatasetCatalog, MetadataCatalog
for d in ["train", "test"]:
    DatasetCatalog.register("p_" + d, lambda d=d: get_r_dicts('/content/drive/
 ↪MyDrive/Mahabub/' + d))
    MetadataCatalog.get("p_" + d).set(thing_classes=['unmelted particle',␣
 ↪'porosity', 'microcrack'])
r_metadata = MetadataCatalog.get("p_train")
```

[ ]:
```
import random

dataset_dicts = get_r_dicts("/content/drive/MyDrive/Mahabub/train")
for d in random.sample(dataset_dicts, 3):
    img = cv2.imread(d["file_name"])
    v = Visualizer(img[:, :, ::-1], metadata=r_metadata, scale=0.5)
    v = v.draw_dataset_dict(d)
    plt.figure(figsize = (14, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()
```


unmelted particle
unmelted particle
unmelted particle
unmelted particle
unmelted particle
porosity
porosity
15kV 6.9mm L-X1.50k BSE H


unmelted particle
unmelted particle
unmelted particle
unmelted particle
unmelted particle
unmelted particle
unmelted particle
unmelted particle
15kV 6.9mm L-X1.50k BSE H
100µm



[ ]:
```
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
from detectron2 import model_zoo

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/
 ↪faster_rcnn_R_50_FPN_1x.yaml"))
cfg.DATASETS.TRAIN = ("p_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/
 ↪faster_rcnn_R_50_FPN_1x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 2000
cfg.SOLVER.STEPS = []        # do not decay learning rate
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 3

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

```
[07/21 19:57:51 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
            (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=4, bias=True)
      (bbox_pred): Linear(in_features=1024, out_features=12, bias=True)
    )
  )
)
[07/21 19:57:51 d2.data.build]: Removed 0 images with no usable annotations. 42
images left.
[07/21 19:57:51 d2.data.build]: Distribution of instances among all 3
categories:
|   category    | #instances   |  category  | #instances   |  category  |
#instances   |
|:-------------:|:-------------|:----------:|:-------------|:----------:|:------
-------|
| unmelted pa.. | 639          |  porosity  | 67           | microcrack | 9
|
|               |              |            |              |            |
|
```

```
|     total     | 715          |            |               |            |
|
[07/21 19:57:51 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()]
[07/21 19:57:51 d2.data.build]: Using training sampler TrainingSampler
[07/21 19:57:51 d2.data.common]: Serializing the dataset using: <class
'detectron2.data.common._TorchSerializedList'>
[07/21 19:57:51 d2.data.common]: Serializing 42 elements to byte tensors and
concatenating them all …
[07/21 19:57:51 d2.data.common]: Serialized dataset takes 0.16 MiB
[07/21 19:57:51 d2.checkpoint.detection_checkpoint]: [DetectionCheckpointer]
Loading from https://dl.fbaipublicfiles.com/detectron2/COCO-
Detection/faster_rcnn_R_50_FPN_1x/137257794/model_final_b275ba.pkl …

model_final_b275ba.pkl: 167MB [00:00, 232MB/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 (4, 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 (4,) 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 (12, 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 (12,) 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}

[07/21 19:57:52 d2.engine.train_loop]: Starting training from iteration 0

/usr/local/lib/python3.10/dist-packages/torch/functional.py:504: 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:3483.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]

[07/21 19:58:14 d2.utils.events]:  eta: 0:29:36  iter: 19  total_loss: 3.389
loss_cls: 1.378  loss_box_reg: 0.6909  loss_rpn_cls: 1.131  loss_rpn_loc: 0.2496
time: 0.8888  last_time: 1.0795  data_time: 0.4448  last_data_time: 0.6154   lr:
4.9953e-06  max_mem: 2457M
[07/21 19:58:26 d2.utils.events]:  eta: 0:16:37  iter: 39  total_loss: 2.535
```

```
loss_cls: 1.298  loss_box_reg: 0.7093  loss_rpn_cls: 0.3476  loss_rpn_loc:
0.2334    time: 0.6664  last_time: 0.4746  data_time: 0.0137  last_data_time:
0.0292   lr: 9.9902e-06  max_mem: 2458M
[07/21 19:58:36 d2.utils.events]:  eta: 0:15:34  iter: 59  total_loss: 2.092
loss_cls: 1.105  loss_box_reg: 0.7166  loss_rpn_cls: 0.0763  loss_rpn_loc:
0.2178    time: 0.5983  last_time: 0.4811  data_time: 0.0109  last_data_time:
0.0079   lr: 1.4985e-05  max_mem: 2458M
[07/21 19:58:45 d2.utils.events]:  eta: 0:15:25  iter: 79  total_loss: 1.895
loss_cls: 0.8734  loss_box_reg: 0.6645  loss_rpn_cls: 0.06235  loss_rpn_loc:
0.2248    time: 0.5659  last_time: 0.4183  data_time: 0.0134  last_data_time:
0.0077   lr: 1.998e-05  max_mem: 2458M
[07/21 19:58:55 d2.utils.events]:  eta: 0:15:20  iter: 99  total_loss: 1.602
loss_cls: 0.7071  loss_box_reg: 0.6546  loss_rpn_cls: 0.04351  loss_rpn_loc:
0.2055    time: 0.5501  last_time: 0.5108  data_time: 0.0093  last_data_time:
0.0103   lr: 2.4975e-05  max_mem: 2458M
[07/21 19:59:04 d2.utils.events]:  eta: 0:15:11  iter: 119  total_loss: 1.433
loss_cls: 0.5588  loss_box_reg: 0.6299  loss_rpn_cls: 0.05256  loss_rpn_loc:
0.2183    time: 0.5374  last_time: 0.4490  data_time: 0.0108  last_data_time:
0.0053   lr: 2.997e-05  max_mem: 2458M
[07/21 19:59:14 d2.utils.events]:  eta: 0:15:04  iter: 139  total_loss: 1.356
loss_cls: 0.494  loss_box_reg: 0.5995  loss_rpn_cls: 0.04068  loss_rpn_loc:
0.2088    time: 0.5315  last_time: 0.5058  data_time: 0.0124  last_data_time:
0.0053   lr: 3.4965e-05  max_mem: 2458M
[07/21 19:59:24 d2.utils.events]:  eta: 0:14:57  iter: 159  total_loss: 1.348
loss_cls: 0.4683  loss_box_reg: 0.5806  loss_rpn_cls: 0.04593  loss_rpn_loc:
0.2192    time: 0.5273  last_time: 0.5185  data_time: 0.0181  last_data_time:
0.0091   lr: 3.996e-05  max_mem: 2458M
[07/21 19:59:34 d2.utils.events]:  eta: 0:14:49  iter: 179  total_loss: 1.289
loss_cls: 0.4235  loss_box_reg: 0.62  loss_rpn_cls: 0.03602  loss_rpn_loc:
0.1973    time: 0.5230  last_time: 0.5047  data_time: 0.0092  last_data_time:
0.0055   lr: 4.4955e-05  max_mem: 2458M
[07/21 19:59:44 d2.utils.events]:  eta: 0:14:47  iter: 199  total_loss: 1.192
loss_cls: 0.3894  loss_box_reg: 0.5594  loss_rpn_cls: 0.04299  loss_rpn_loc:
0.2108    time: 0.5211  last_time: 0.5068  data_time: 0.0146  last_data_time:
0.0067   lr: 4.995e-05  max_mem: 2459M
[07/21 19:59:54 d2.utils.events]:  eta: 0:14:44  iter: 219  total_loss: 1.207
loss_cls: 0.3596  loss_box_reg: 0.6116  loss_rpn_cls: 0.04293  loss_rpn_loc:
0.2037    time: 0.5192  last_time: 0.4792  data_time: 0.0108  last_data_time:
0.0236   lr: 5.4945e-05  max_mem: 2459M
[07/21 20:00:04 d2.utils.events]:  eta: 0:14:35  iter: 239  total_loss: 1.152
loss_cls: 0.3379  loss_box_reg: 0.5694  loss_rpn_cls: 0.03014  loss_rpn_loc:
0.206    time: 0.5165  last_time: 0.4752  data_time: 0.0112  last_data_time:
0.0256   lr: 5.994e-05  max_mem: 2459M
[07/21 20:00:14 d2.utils.events]:  eta: 0:14:25  iter: 259  total_loss: 1.075
loss_cls: 0.287  loss_box_reg: 0.5379  loss_rpn_cls: 0.03224  loss_rpn_loc:
0.1983    time: 0.5151  last_time: 0.5000  data_time: 0.0125  last_data_time:
0.0061   lr: 6.4935e-05  max_mem: 2459M
[07/21 20:00:24 d2.utils.events]:  eta: 0:14:16  iter: 279  total_loss: 1.152
```

```
loss_cls: 0.3071  loss_box_reg: 0.5656  loss_rpn_cls: 0.04731  loss_rpn_loc:
0.2091    time: 0.5132  last_time: 0.4449  data_time: 0.0099  last_data_time:
0.0076   lr: 6.993e-05  max_mem: 2459M
[07/21 20:00:33 d2.utils.events]:  eta: 0:14:07  iter: 299  total_loss: 1.093
loss_cls: 0.2917  loss_box_reg: 0.5664  loss_rpn_cls: 0.0315  loss_rpn_loc:
0.2058    time: 0.5114  last_time: 0.5012  data_time: 0.0092  last_data_time:
0.0071   lr: 7.4925e-05  max_mem: 2459M
[07/21 20:00:43 d2.utils.events]:  eta: 0:13:57  iter: 319  total_loss: 1.096
loss_cls: 0.2748  loss_box_reg: 0.5925  loss_rpn_cls: 0.03856  loss_rpn_loc:
0.2089    time: 0.5105  last_time: 0.4488  data_time: 0.0151  last_data_time:
0.0218   lr: 7.992e-05  max_mem: 2459M
[07/21 20:00:53 d2.utils.events]:  eta: 0:13:49  iter: 339  total_loss: 0.9908
loss_cls: 0.253  loss_box_reg: 0.5075  loss_rpn_cls: 0.0314  loss_rpn_loc:
0.2039    time: 0.5099  last_time: 0.4945  data_time: 0.0126  last_data_time:
0.0053   lr: 8.4915e-05  max_mem: 2459M
[07/21 20:01:03 d2.utils.events]:  eta: 0:13:40  iter: 359  total_loss: 1.047
loss_cls: 0.2426  loss_box_reg: 0.5547  loss_rpn_cls: 0.03701  loss_rpn_loc:
0.21    time: 0.5083  last_time: 0.5136  data_time: 0.0076  last_data_time:
0.0065   lr: 8.991e-05  max_mem: 2459M
[07/21 20:01:13 d2.utils.events]:  eta: 0:13:30  iter: 379  total_loss: 0.9958
loss_cls: 0.2408  loss_box_reg: 0.5071  loss_rpn_cls: 0.04039  loss_rpn_loc:
0.2012    time: 0.5079  last_time: 0.4619  data_time: 0.0121  last_data_time:
0.0051   lr: 9.4905e-05  max_mem: 2459M
[07/21 20:01:23 d2.utils.events]:  eta: 0:13:21  iter: 399  total_loss: 0.9962
loss_cls: 0.2342  loss_box_reg: 0.4927  loss_rpn_cls: 0.03574  loss_rpn_loc:
0.2097    time: 0.5072  last_time: 0.5011  data_time: 0.0115  last_data_time:
0.0065   lr: 9.99e-05  max_mem: 2459M
[07/21 20:01:33 d2.utils.events]:  eta: 0:13:12  iter: 419  total_loss: 1.036
loss_cls: 0.2455  loss_box_reg: 0.5391  loss_rpn_cls: 0.03309  loss_rpn_loc:
0.2028    time: 0.5068  last_time: 0.4849  data_time: 0.0073  last_data_time:
0.0278   lr: 0.0001049  max_mem: 2459M
[07/21 20:01:43 d2.utils.events]:  eta: 0:13:02  iter: 439  total_loss: 1.006
loss_cls: 0.2414  loss_box_reg: 0.5405  loss_rpn_cls: 0.03749  loss_rpn_loc:
0.1933    time: 0.5065  last_time: 0.5088  data_time: 0.0114  last_data_time:
0.0070   lr: 0.00010989  max_mem: 2459M
[07/21 20:01:53 d2.utils.events]:  eta: 0:12:53  iter: 459  total_loss: 0.9855
loss_cls: 0.2145  loss_box_reg: 0.512  loss_rpn_cls: 0.03028  loss_rpn_loc:
0.2053    time: 0.5067  last_time: 0.5127  data_time: 0.0114  last_data_time:
0.0164   lr: 0.00011489  max_mem: 2459M
[07/21 20:02:03 d2.utils.events]:  eta: 0:12:43  iter: 479  total_loss: 0.9543
loss_cls: 0.2324  loss_box_reg: 0.5102  loss_rpn_cls: 0.0358  loss_rpn_loc:
0.1961    time: 0.5066  last_time: 0.4731  data_time: 0.0144  last_data_time:
0.0262   lr: 0.00011988  max_mem: 2459M
[07/21 20:02:13 d2.utils.events]:  eta: 0:12:34  iter: 499  total_loss: 0.9725
loss_cls: 0.2158  loss_box_reg: 0.5117  loss_rpn_cls: 0.03729  loss_rpn_loc:
0.2013    time: 0.5064  last_time: 0.5068  data_time: 0.0120  last_data_time:
0.0077   lr: 0.00012488  max_mem: 2459M
[07/21 20:02:23 d2.utils.events]:  eta: 0:12:24  iter: 519  total_loss: 0.9803
```

```
loss_cls: 0.2289  loss_box_reg: 0.5081  loss_rpn_cls: 0.03088  loss_rpn_loc:
0.1876    time: 0.5063  last_time: 0.5089  data_time: 0.0140  last_data_time:
0.0057   lr: 0.00012987  max_mem: 2459M
[07/21 20:02:33 d2.utils.events]:  eta: 0:12:14  iter: 539  total_loss: 0.9504
loss_cls: 0.2239  loss_box_reg: 0.4858  loss_rpn_cls: 0.03621  loss_rpn_loc:
0.2079    time: 0.5060  last_time: 0.5335  data_time: 0.0092  last_data_time:
0.0188   lr: 0.00013487  max_mem: 2459M
[07/21 20:02:43 d2.utils.events]:  eta: 0:12:05  iter: 559  total_loss: 0.8955
loss_cls: 0.2055  loss_box_reg: 0.477  loss_rpn_cls: 0.02882  loss_rpn_loc:
0.1989    time: 0.5058  last_time: 0.5040  data_time: 0.0089  last_data_time:
0.0071   lr: 0.00013986  max_mem: 2459M
[07/21 20:02:53 d2.utils.events]:  eta: 0:11:55  iter: 579  total_loss: 0.8896
loss_cls: 0.2104  loss_box_reg: 0.4513  loss_rpn_cls: 0.02985  loss_rpn_loc:
0.1875    time: 0.5056  last_time: 0.4410  data_time: 0.0144  last_data_time:
0.0068   lr: 0.00014486  max_mem: 2459M
[07/21 20:03:03 d2.utils.events]:  eta: 0:11:45  iter: 599  total_loss: 0.9611
loss_cls: 0.2127  loss_box_reg: 0.5116  loss_rpn_cls: 0.02654  loss_rpn_loc:
0.1916    time: 0.5053  last_time: 0.5195  data_time: 0.0077  last_data_time:
0.0058   lr: 0.00014985  max_mem: 2459M
[07/21 20:03:13 d2.utils.events]:  eta: 0:11:35  iter: 619  total_loss: 0.9528
loss_cls: 0.2105  loss_box_reg: 0.483  loss_rpn_cls: 0.03678  loss_rpn_loc:
0.2047    time: 0.5051  last_time: 0.5066  data_time: 0.0083  last_data_time:
0.0062   lr: 0.00015485  max_mem: 2459M
[07/21 20:03:23 d2.utils.events]:  eta: 0:11:25  iter: 639  total_loss: 0.8813
loss_cls: 0.2137  loss_box_reg: 0.4519  loss_rpn_cls: 0.02882  loss_rpn_loc:
0.2017    time: 0.5049  last_time: 0.5086  data_time: 0.0134  last_data_time:
0.0075   lr: 0.00015984  max_mem: 2459M
[07/21 20:03:34 d2.utils.events]:  eta: 0:11:15  iter: 659  total_loss: 0.9292
loss_cls: 0.2066  loss_box_reg: 0.5052  loss_rpn_cls: 0.02367  loss_rpn_loc:
0.1932    time: 0.5050  last_time: 0.5118  data_time: 0.0132  last_data_time:
0.0070   lr: 0.00016484  max_mem: 2459M
[07/21 20:03:43 d2.utils.events]:  eta: 0:11:05  iter: 679  total_loss: 0.9448
loss_cls: 0.2076  loss_box_reg: 0.4758  loss_rpn_cls: 0.02427  loss_rpn_loc:
0.2133    time: 0.5045  last_time: 0.4741  data_time: 0.0079  last_data_time:
0.0066   lr: 0.00016983  max_mem: 2459M
[07/21 20:03:53 d2.utils.events]:  eta: 0:10:55  iter: 699  total_loss: 0.8655
loss_cls: 0.186  loss_box_reg: 0.4489  loss_rpn_cls: 0.02518  loss_rpn_loc:
0.2043    time: 0.5044  last_time: 0.5073  data_time: 0.0154  last_data_time:
0.0122   lr: 0.00017483  max_mem: 2459M
[07/21 20:04:03 d2.utils.events]:  eta: 0:10:45  iter: 719  total_loss: 0.9025
loss_cls: 0.2077  loss_box_reg: 0.4817  loss_rpn_cls: 0.03218  loss_rpn_loc:
0.1866    time: 0.5042  last_time: 0.5073  data_time: 0.0145  last_data_time:
0.0049   lr: 0.00017982  max_mem: 2459M
[07/21 20:04:13 d2.utils.events]:  eta: 0:10:35  iter: 739  total_loss: 0.947
loss_cls: 0.2041  loss_box_reg: 0.4736  loss_rpn_cls: 0.03904  loss_rpn_loc:
0.2023    time: 0.5037  last_time: 0.5108  data_time: 0.0074  last_data_time:
0.0086   lr: 0.00018482  max_mem: 2459M
[07/21 20:04:23 d2.utils.events]:  eta: 0:10:25  iter: 759  total_loss: 0.9057
```

```
loss_cls: 0.2079  loss_box_reg: 0.4759  loss_rpn_cls: 0.02757  loss_rpn_loc:
0.193    time: 0.5038  last_time: 0.5076  data_time: 0.0140  last_data_time:
0.0051   lr: 0.00018981  max_mem: 2459M
[07/21 20:04:33 d2.utils.events]:  eta: 0:10:15  iter: 779  total_loss: 0.8695
loss_cls: 0.1818  loss_box_reg: 0.4572  loss_rpn_cls: 0.03313  loss_rpn_loc:
0.191    time: 0.5038  last_time: 0.5031  data_time: 0.0158  last_data_time:
0.0056   lr: 0.00019481  max_mem: 2459M
[07/21 20:04:43 d2.utils.events]:  eta: 0:10:05  iter: 799  total_loss: 0.8578
loss_cls: 0.1963  loss_box_reg: 0.466  loss_rpn_cls: 0.02887  loss_rpn_loc:
0.1998    time: 0.5035  last_time: 0.5275  data_time: 0.0106  last_data_time:
0.0114   lr: 0.0001998  max_mem: 2459M
[07/21 20:04:53 d2.utils.events]:  eta: 0:09:55  iter: 819  total_loss: 0.8745
loss_cls: 0.1988  loss_box_reg: 0.4783  loss_rpn_cls: 0.03046  loss_rpn_loc:
0.1895    time: 0.5036  last_time: 0.5123  data_time: 0.0159  last_data_time:
0.0162   lr: 0.0002048  max_mem: 2459M
[07/21 20:05:03 d2.utils.events]:  eta: 0:09:45  iter: 839  total_loss: 0.844
loss_cls: 0.1947  loss_box_reg: 0.4445  loss_rpn_cls: 0.02725  loss_rpn_loc:
0.1991    time: 0.5037  last_time: 0.5273  data_time: 0.0113  last_data_time:
0.0253   lr: 0.00020979  max_mem: 2459M
[07/21 20:05:13 d2.utils.events]:  eta: 0:09:35  iter: 859  total_loss: 0.8837
loss_cls: 0.1885  loss_box_reg: 0.4493  loss_rpn_cls: 0.02789  loss_rpn_loc:
0.2089    time: 0.5036  last_time: 0.5082  data_time: 0.0080  last_data_time:
0.0089   lr: 0.00021479  max_mem: 2459M
[07/21 20:05:24 d2.utils.events]:  eta: 0:09:25  iter: 879  total_loss: 0.8751
loss_cls: 0.1843  loss_box_reg: 0.4445  loss_rpn_cls: 0.02054  loss_rpn_loc:
0.1878    time: 0.5036  last_time: 0.5031  data_time: 0.0130  last_data_time:
0.0061   lr: 0.00021978  max_mem: 2460M
[07/21 20:05:34 d2.utils.events]:  eta: 0:09:15  iter: 899  total_loss: 0.8943
loss_cls: 0.1828  loss_box_reg: 0.4621  loss_rpn_cls: 0.02659  loss_rpn_loc:
0.2077    time: 0.5035  last_time: 0.5087  data_time: 0.0102  last_data_time:
0.0076   lr: 0.00022478  max_mem: 2460M
[07/21 20:05:44 d2.utils.events]:  eta: 0:09:05  iter: 919  total_loss: 0.8532
loss_cls: 0.1844  loss_box_reg: 0.4622  loss_rpn_cls: 0.02866  loss_rpn_loc:
0.1771    time: 0.5034  last_time: 0.4818  data_time: 0.0092  last_data_time:
0.0091   lr: 0.00022977  max_mem: 2460M
[07/21 20:05:54 d2.utils.events]:  eta: 0:08:55  iter: 939  total_loss: 0.8615
loss_cls: 0.1807  loss_box_reg: 0.4722  loss_rpn_cls: 0.02832  loss_rpn_loc:
0.1889    time: 0.5035  last_time: 0.5065  data_time: 0.0149  last_data_time:
0.0089   lr: 0.00023477  max_mem: 2460M
[07/21 20:06:04 d2.utils.events]:  eta: 0:08:45  iter: 959  total_loss: 0.8227
loss_cls: 0.1846  loss_box_reg: 0.4388  loss_rpn_cls: 0.02451  loss_rpn_loc:
0.1831    time: 0.5033  last_time: 0.5119  data_time: 0.0112  last_data_time:
0.0099   lr: 0.00023976  max_mem: 2460M
[07/21 20:06:13 d2.utils.events]:  eta: 0:08:35  iter: 979  total_loss: 0.8871
loss_cls: 0.1893  loss_box_reg: 0.4717  loss_rpn_cls: 0.02605  loss_rpn_loc:
0.1923    time: 0.5031  last_time: 0.4737  data_time: 0.0094  last_data_time:
0.0288   lr: 0.00024476  max_mem: 2460M
[07/21 20:06:23 d2.utils.events]:  eta: 0:08:24  iter: 999  total_loss: 0.8739
```

```
loss_cls: 0.1889  loss_box_reg: 0.448  loss_rpn_cls: 0.03051  loss_rpn_loc:
0.1904    time: 0.5028  last_time: 0.5041  data_time: 0.0082  last_data_time:
0.0069   lr: 0.00024975  max_mem: 2460M
[07/21 20:06:33 d2.utils.events]:  eta: 0:08:14  iter: 1019  total_loss: 0.8539
loss_cls: 0.1777  loss_box_reg: 0.451  loss_rpn_cls: 0.02414  loss_rpn_loc:
0.1775    time: 0.5028  last_time: 0.4600  data_time: 0.0092  last_data_time:
0.0053   lr: 0.00025  max_mem: 2460M
[07/21 20:06:43 d2.utils.events]:  eta: 0:08:04  iter: 1039  total_loss: 0.8387
loss_cls: 0.1638  loss_box_reg: 0.4521  loss_rpn_cls: 0.02573  loss_rpn_loc:
0.1895    time: 0.5026  last_time: 0.5296  data_time: 0.0109  last_data_time:
0.0287   lr: 0.00025  max_mem: 2460M
[07/21 20:06:53 d2.utils.events]:  eta: 0:07:54  iter: 1059  total_loss: 0.8994
loss_cls: 0.183  loss_box_reg: 0.489  loss_rpn_cls: 0.03153  loss_rpn_loc:
0.1864    time: 0.5024  last_time: 0.4431  data_time: 0.0119  last_data_time:
0.0074   lr: 0.00025  max_mem: 2460M
[07/21 20:07:03 d2.utils.events]:  eta: 0:07:44  iter: 1079  total_loss: 0.8499
loss_cls: 0.1801  loss_box_reg: 0.4489  loss_rpn_cls: 0.02652  loss_rpn_loc:
0.1963    time: 0.5022  last_time: 0.4562  data_time: 0.0101  last_data_time:
0.0055   lr: 0.00025  max_mem: 2460M
[07/21 20:07:13 d2.utils.events]:  eta: 0:07:34  iter: 1099  total_loss: 0.8412
loss_cls: 0.1664  loss_box_reg: 0.45  loss_rpn_cls: 0.02814  loss_rpn_loc:
0.1865    time: 0.5021  last_time: 0.5365  data_time: 0.0117  last_data_time:
0.0314   lr: 0.00025  max_mem: 2460M
[07/21 20:07:23 d2.utils.events]:  eta: 0:07:24  iter: 1119  total_loss: 0.8384
loss_cls: 0.1752  loss_box_reg: 0.4484  loss_rpn_cls: 0.02736  loss_rpn_loc:
0.1848    time: 0.5019  last_time: 0.5055  data_time: 0.0115  last_data_time:
0.0057   lr: 0.00025  max_mem: 2460M
[07/21 20:07:33 d2.utils.events]:  eta: 0:07:14  iter: 1139  total_loss: 0.8542
loss_cls: 0.1809  loss_box_reg: 0.434  loss_rpn_cls: 0.02298  loss_rpn_loc:
0.1866    time: 0.5022  last_time: 0.5104  data_time: 0.0162  last_data_time:
0.0080   lr: 0.00025  max_mem: 2460M
[07/21 20:07:43 d2.utils.events]:  eta: 0:07:04  iter: 1159  total_loss: 0.822
loss_cls: 0.1657  loss_box_reg: 0.4536  loss_rpn_cls: 0.02117  loss_rpn_loc:
0.1924    time: 0.5021  last_time: 0.5162  data_time: 0.0084  last_data_time:
0.0053   lr: 0.00025  max_mem: 2460M
[07/21 20:07:53 d2.utils.events]:  eta: 0:06:54  iter: 1179  total_loss: 0.842
loss_cls: 0.1712  loss_box_reg: 0.4564  loss_rpn_cls: 0.02899  loss_rpn_loc:
0.1857    time: 0.5018  last_time: 0.5093  data_time: 0.0089  last_data_time:
0.0053   lr: 0.00025  max_mem: 2460M
[07/21 20:08:03 d2.utils.events]:  eta: 0:06:44  iter: 1199  total_loss: 0.823
loss_cls: 0.1617  loss_box_reg: 0.4418  loss_rpn_cls: 0.02474  loss_rpn_loc:
0.1833    time: 0.5018  last_time: 0.5068  data_time: 0.0121  last_data_time:
0.0062   lr: 0.00025  max_mem: 2460M
[07/21 20:08:13 d2.utils.events]:  eta: 0:06:34  iter: 1219  total_loss: 0.8018
loss_cls: 0.1527  loss_box_reg: 0.427  loss_rpn_cls: 0.0284  loss_rpn_loc:
0.1824    time: 0.5016  last_time: 0.5200  data_time: 0.0101  last_data_time:
0.0223   lr: 0.00025  max_mem: 2460M
[07/21 20:08:23 d2.utils.events]:  eta: 0:06:24  iter: 1239  total_loss: 0.8346
```

```
loss_cls: 0.1704  loss_box_reg: 0.4276  loss_rpn_cls: 0.0291  loss_rpn_loc:
0.1897    time: 0.5016  last_time: 0.5065  data_time: 0.0092  last_data_time:
0.0063   lr: 0.00025  max_mem: 2460M
[07/21 20:08:33 d2.utils.events]:  eta: 0:06:14  iter: 1259  total_loss: 0.8039
loss_cls: 0.1643  loss_box_reg: 0.4036  loss_rpn_cls: 0.02778  loss_rpn_loc:
0.1863    time: 0.5015  last_time: 0.5149  data_time: 0.0142  last_data_time:
0.0067   lr: 0.00025  max_mem: 2460M
[07/21 20:08:43 d2.utils.events]:  eta: 0:06:04  iter: 1279  total_loss: 0.848
loss_cls: 0.1699  loss_box_reg: 0.4594  loss_rpn_cls: 0.03087  loss_rpn_loc:
0.1718    time: 0.5016  last_time: 0.5227  data_time: 0.0159  last_data_time:
0.0188   lr: 0.00025  max_mem: 2460M
[07/21 20:08:53 d2.utils.events]:  eta: 0:05:54  iter: 1299  total_loss: 0.8093
loss_cls: 0.1674  loss_box_reg: 0.4326  loss_rpn_cls: 0.03009  loss_rpn_loc:
0.1839    time: 0.5014  last_time: 0.5041  data_time: 0.0086  last_data_time:
0.0054   lr: 0.00025  max_mem: 2460M
[07/21 20:09:03 d2.utils.events]:  eta: 0:05:44  iter: 1319  total_loss: 0.7906
loss_cls: 0.1554  loss_box_reg: 0.4057  loss_rpn_cls: 0.02665  loss_rpn_loc:
0.1795    time: 0.5014  last_time: 0.5140  data_time: 0.0145  last_data_time:
0.0111   lr: 0.00025  max_mem: 2460M
[07/21 20:09:12 d2.utils.events]:  eta: 0:05:33  iter: 1339  total_loss: 0.8777
loss_cls: 0.1653  loss_box_reg: 0.4539  loss_rpn_cls: 0.02828  loss_rpn_loc:
0.1959    time: 0.5011  last_time: 0.4939  data_time: 0.0111  last_data_time:
0.0252   lr: 0.00025  max_mem: 2460M
[07/21 20:09:22 d2.utils.events]:  eta: 0:05:23  iter: 1359  total_loss: 0.8369
loss_cls: 0.1623  loss_box_reg: 0.4108  loss_rpn_cls: 0.03059  loss_rpn_loc:
0.1785    time: 0.5011  last_time: 0.5114  data_time: 0.0097  last_data_time:
0.0099   lr: 0.00025  max_mem: 2460M
[07/21 20:09:32 d2.utils.events]:  eta: 0:05:13  iter: 1379  total_loss: 0.8035
loss_cls: 0.1546  loss_box_reg: 0.4378  loss_rpn_cls: 0.02434  loss_rpn_loc:
0.182    time: 0.5013  last_time: 0.5139  data_time: 0.0130  last_data_time:
0.0091   lr: 0.00025  max_mem: 2460M
[07/21 20:09:43 d2.utils.events]:  eta: 0:05:03  iter: 1399  total_loss: 0.8001
loss_cls: 0.1495  loss_box_reg: 0.4338  loss_rpn_cls: 0.02526  loss_rpn_loc:
0.1834    time: 0.5013  last_time: 0.5042  data_time: 0.0096  last_data_time:
0.0063   lr: 0.00025  max_mem: 2460M
[07/21 20:09:52 d2.utils.events]:  eta: 0:04:53  iter: 1419  total_loss: 0.7755
loss_cls: 0.1522  loss_box_reg: 0.4103  loss_rpn_cls: 0.02737  loss_rpn_loc:
0.1758    time: 0.5012  last_time: 0.5201  data_time: 0.0092  last_data_time:
0.0076   lr: 0.00025  max_mem: 2460M
[07/21 20:10:03 d2.utils.events]:  eta: 0:04:43  iter: 1439  total_loss: 0.7634
loss_cls: 0.1504  loss_box_reg: 0.4075  loss_rpn_cls: 0.02651  loss_rpn_loc:
0.1745    time: 0.5012  last_time: 0.5162  data_time: 0.0129  last_data_time:
0.0071   lr: 0.00025  max_mem: 2460M
[07/21 20:10:13 d2.utils.events]:  eta: 0:04:33  iter: 1459  total_loss: 0.8033
loss_cls: 0.1515  loss_box_reg: 0.4205  loss_rpn_cls: 0.03192  loss_rpn_loc:
0.1862    time: 0.5014  last_time: 0.5237  data_time: 0.0132  last_data_time:
0.0070   lr: 0.00025  max_mem: 2460M
[07/21 20:10:23 d2.utils.events]:  eta: 0:04:23  iter: 1479  total_loss: 0.784
```

```
loss_cls: 0.1596  loss_box_reg: 0.4087  loss_rpn_cls: 0.02442  loss_rpn_loc:
0.1661    time: 0.5013  last_time: 0.5078  data_time: 0.0083  last_data_time:
0.0078   lr: 0.00025  max_mem: 2460M
[07/21 20:10:33 d2.utils.events]:  eta: 0:04:12  iter: 1499  total_loss: 0.8177
loss_cls: 0.1565  loss_box_reg: 0.4394  loss_rpn_cls: 0.0264  loss_rpn_loc:
0.186    time: 0.5013  last_time: 0.4592  data_time: 0.0150  last_data_time:
0.0062   lr: 0.00025  max_mem: 2460M
[07/21 20:10:43 d2.utils.events]:  eta: 0:04:02  iter: 1519  total_loss: 0.7781
loss_cls: 0.1542  loss_box_reg: 0.4049  loss_rpn_cls: 0.01658  loss_rpn_loc:
0.1617    time: 0.5013  last_time: 0.5259  data_time: 0.0146  last_data_time:
0.0223   lr: 0.00025  max_mem: 2460M
[07/21 20:10:53 d2.utils.events]:  eta: 0:03:52  iter: 1539  total_loss: 0.8439
loss_cls: 0.1569  loss_box_reg: 0.4433  loss_rpn_cls: 0.03017  loss_rpn_loc:
0.1729    time: 0.5012  last_time: 0.5084  data_time: 0.0072  last_data_time:
0.0094   lr: 0.00025  max_mem: 2460M
[07/21 20:11:03 d2.utils.events]:  eta: 0:03:42  iter: 1559  total_loss: 0.7678
loss_cls: 0.1511  loss_box_reg: 0.3971  loss_rpn_cls: 0.03138  loss_rpn_loc:
0.1772    time: 0.5012  last_time: 0.4616  data_time: 0.0125  last_data_time:
0.0055   lr: 0.00025  max_mem: 2460M
[07/21 20:11:13 d2.utils.events]:  eta: 0:03:32  iter: 1579  total_loss: 0.7551
loss_cls: 0.1401  loss_box_reg: 0.4088  loss_rpn_cls: 0.02351  loss_rpn_loc:
0.174    time: 0.5011  last_time: 0.5026  data_time: 0.0106  last_data_time:
0.0067   lr: 0.00025  max_mem: 2460M
[07/21 20:11:22 d2.utils.events]:  eta: 0:03:22  iter: 1599  total_loss: 0.7431
loss_cls: 0.1449  loss_box_reg: 0.4152  loss_rpn_cls: 0.0278  loss_rpn_loc:
0.1749    time: 0.5010  last_time: 0.5328  data_time: 0.0074  last_data_time:
0.0254   lr: 0.00025  max_mem: 2460M
[07/21 20:11:32 d2.utils.events]:  eta: 0:03:12  iter: 1619  total_loss: 0.77
loss_cls: 0.1511  loss_box_reg: 0.4161  loss_rpn_cls: 0.02403  loss_rpn_loc:
0.1698    time: 0.5010  last_time: 0.5148  data_time: 0.0131  last_data_time:
0.0183   lr: 0.00025  max_mem: 2460M
[07/21 20:11:43 d2.utils.events]:  eta: 0:03:02  iter: 1639  total_loss: 0.7163
loss_cls: 0.1446  loss_box_reg: 0.3997  loss_rpn_cls: 0.02483  loss_rpn_loc:
0.1694    time: 0.5010  last_time: 0.5031  data_time: 0.0152  last_data_time:
0.0055   lr: 0.00025  max_mem: 2460M
[07/21 20:11:53 d2.utils.events]:  eta: 0:02:52  iter: 1659  total_loss: 0.8059
loss_cls: 0.1538  loss_box_reg: 0.4282  loss_rpn_cls: 0.02244  loss_rpn_loc:
0.1795    time: 0.5010  last_time: 0.5332  data_time: 0.0115  last_data_time:
0.0273   lr: 0.00025  max_mem: 2460M
[07/21 20:12:03 d2.utils.events]:  eta: 0:02:41  iter: 1679  total_loss: 0.7493
loss_cls: 0.1462  loss_box_reg: 0.3842  loss_rpn_cls: 0.02939  loss_rpn_loc:
0.1667    time: 0.5010  last_time: 0.5123  data_time: 0.0127  last_data_time:
0.0167   lr: 0.00025  max_mem: 2460M
[07/21 20:12:13 d2.utils.events]:  eta: 0:02:31  iter: 1699  total_loss: 0.7625
loss_cls: 0.146  loss_box_reg: 0.4251  loss_rpn_cls: 0.02153  loss_rpn_loc:
0.1705    time: 0.5009  last_time: 0.4993  data_time: 0.0104  last_data_time:
0.0083   lr: 0.00025  max_mem: 2460M
[07/21 20:12:23 d2.utils.events]:  eta: 0:02:21  iter: 1719  total_loss: 0.794
```

```
loss_cls: 0.145  loss_box_reg: 0.4268  loss_rpn_cls: 0.02385  loss_rpn_loc:
0.1701    time: 0.5010  last_time: 0.5296  data_time: 0.0099  last_data_time:
0.0299   lr: 0.00025  max_mem: 2460M
[07/21 20:12:33 d2.utils.events]:  eta: 0:02:11  iter: 1739  total_loss: 0.7645
loss_cls: 0.1469  loss_box_reg: 0.4062  loss_rpn_cls: 0.02329  loss_rpn_loc:
0.1714    time: 0.5009  last_time: 0.4991  data_time: 0.0108  last_data_time:
0.0076   lr: 0.00025  max_mem: 2460M
[07/21 20:12:43 d2.utils.events]:  eta: 0:02:01  iter: 1759  total_loss: 0.7724
loss_cls: 0.1371  loss_box_reg: 0.4121  loss_rpn_cls: 0.02403  loss_rpn_loc:
0.1748    time: 0.5009  last_time: 0.5036  data_time: 0.0119  last_data_time:
0.0078   lr: 0.00025  max_mem: 2460M
[07/21 20:12:53 d2.utils.events]:  eta: 0:01:51  iter: 1779  total_loss: 0.704
loss_cls: 0.1356  loss_box_reg: 0.3941  loss_rpn_cls: 0.02364  loss_rpn_loc:
0.1719    time: 0.5009  last_time: 0.5271  data_time: 0.0088  last_data_time:
0.0133   lr: 0.00025  max_mem: 2460M
[07/21 20:13:03 d2.utils.events]:  eta: 0:01:41  iter: 1799  total_loss: 0.7502
loss_cls: 0.1508  loss_box_reg: 0.3933  loss_rpn_cls: 0.02278  loss_rpn_loc:
0.1586    time: 0.5010  last_time: 0.4621  data_time: 0.0127  last_data_time:
0.0141   lr: 0.00025  max_mem: 2460M
[07/21 20:13:13 d2.utils.events]:  eta: 0:01:31  iter: 1819  total_loss: 0.7336
loss_cls: 0.1427  loss_box_reg: 0.3987  loss_rpn_cls: 0.02192  loss_rpn_loc:
0.1686    time: 0.5010  last_time: 0.4575  data_time: 0.0155  last_data_time:
0.0077   lr: 0.00025  max_mem: 2460M
[07/21 20:13:23 d2.utils.events]:  eta: 0:01:20  iter: 1839  total_loss: 0.7604
loss_cls: 0.1456  loss_box_reg: 0.4191  loss_rpn_cls: 0.02649  loss_rpn_loc:
0.1717    time: 0.5009  last_time: 0.5158  data_time: 0.0131  last_data_time:
0.0059   lr: 0.00025  max_mem: 2460M
[07/21 20:13:33 d2.utils.events]:  eta: 0:01:10  iter: 1859  total_loss: 0.7479
loss_cls: 0.1385  loss_box_reg: 0.3946  loss_rpn_cls: 0.02157  loss_rpn_loc:
0.1615    time: 0.5009  last_time: 0.5083  data_time: 0.0071  last_data_time:
0.0099   lr: 0.00025  max_mem: 2460M
[07/21 20:13:43 d2.utils.events]:  eta: 0:01:00  iter: 1879  total_loss: 0.7396
loss_cls: 0.1373  loss_box_reg: 0.4178  loss_rpn_cls: 0.01618  loss_rpn_loc:
0.1722    time: 0.5009  last_time: 0.5038  data_time: 0.0115  last_data_time:
0.0064   lr: 0.00025  max_mem: 2460M
[07/21 20:13:53 d2.utils.events]:  eta: 0:00:50  iter: 1899  total_loss: 0.7863
loss_cls: 0.1532  loss_box_reg: 0.4178  loss_rpn_cls: 0.02054  loss_rpn_loc:
0.1662    time: 0.5009  last_time: 0.5064  data_time: 0.0133  last_data_time:
0.0101   lr: 0.00025  max_mem: 2460M
[07/21 20:14:03 d2.utils.events]:  eta: 0:00:40  iter: 1919  total_loss: 0.7363
loss_cls: 0.1426  loss_box_reg: 0.4036  loss_rpn_cls: 0.02464  loss_rpn_loc:
0.168    time: 0.5009  last_time: 0.5371  data_time: 0.0114  last_data_time:
0.0330   lr: 0.00025  max_mem: 2460M
[07/21 20:14:13 d2.utils.events]:  eta: 0:00:30  iter: 1939  total_loss: 0.737
loss_cls: 0.1379  loss_box_reg: 0.4141  loss_rpn_cls: 0.02441  loss_rpn_loc:
0.1639    time: 0.5009  last_time: 0.4934  data_time: 0.0135  last_data_time:
0.0063   lr: 0.00025  max_mem: 2460M
[07/21 20:14:23 d2.utils.events]:  eta: 0:00:20  iter: 1959  total_loss: 0.7117
```

```
loss_cls: 0.1451  loss_box_reg: 0.3927  loss_rpn_cls: 0.01779  loss_rpn_loc:
0.1601    time: 0.5009  last_time: 0.5041  data_time: 0.0111  last_data_time:
0.0054   lr: 0.00025  max_mem: 2460M
[07/21 20:14:33 d2.utils.events]:  eta: 0:00:10  iter: 1979  total_loss: 0.7286
loss_cls: 0.1349  loss_box_reg: 0.3941  loss_rpn_cls: 0.02489  loss_rpn_loc:
0.1644    time: 0.5008  last_time: 0.5132  data_time: 0.0065  last_data_time:
0.0054   lr: 0.00025  max_mem: 2460M
[07/21 20:14:48 d2.utils.events]:  eta: 0:00:00  iter: 1999  total_loss: 0.7437
loss_cls: 0.1473  loss_box_reg: 0.4105  loss_rpn_cls: 0.01927  loss_rpn_loc:
0.1701    time: 0.5007  last_time: 0.5013  data_time: 0.0115  last_data_time:
0.0053   lr: 0.00025  max_mem: 2460M
[07/21 20:14:48 d2.engine.hooks]: Overall training speed: 1998 iterations in
0:16:40 (0.5007 s / it)
[07/21 20:14:48 d2.engine.hooks]: Total training time: 0:16:50 (0:00:10 on
hooks)
```

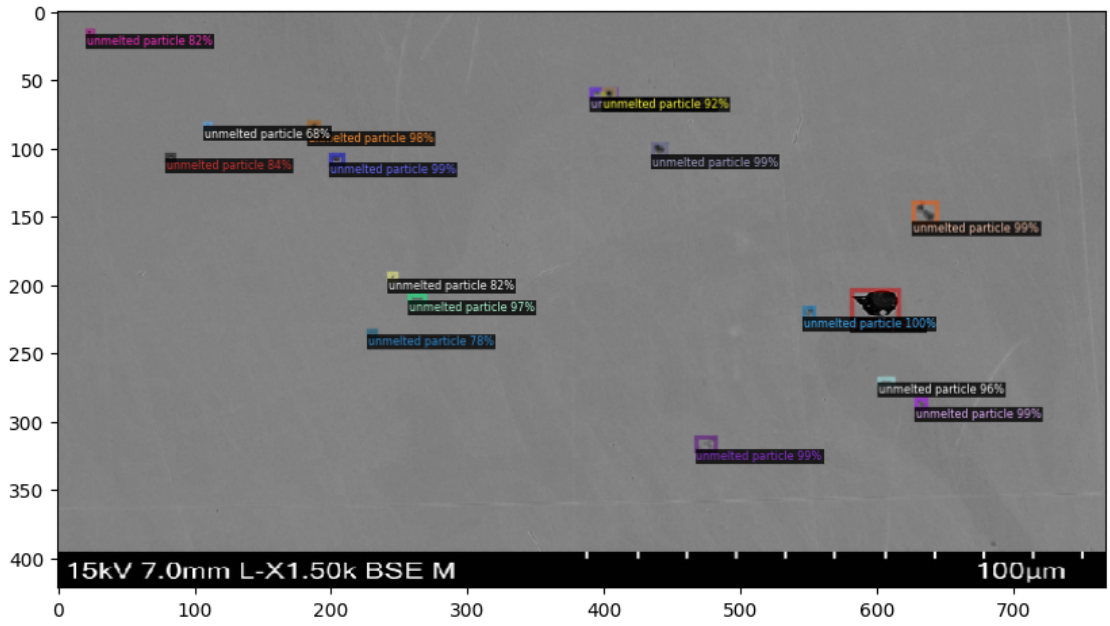
[ ]:
```
# Look at training curves in tensorboard:
%reload_ext tensorboard
%tensorboard --logdir output
```

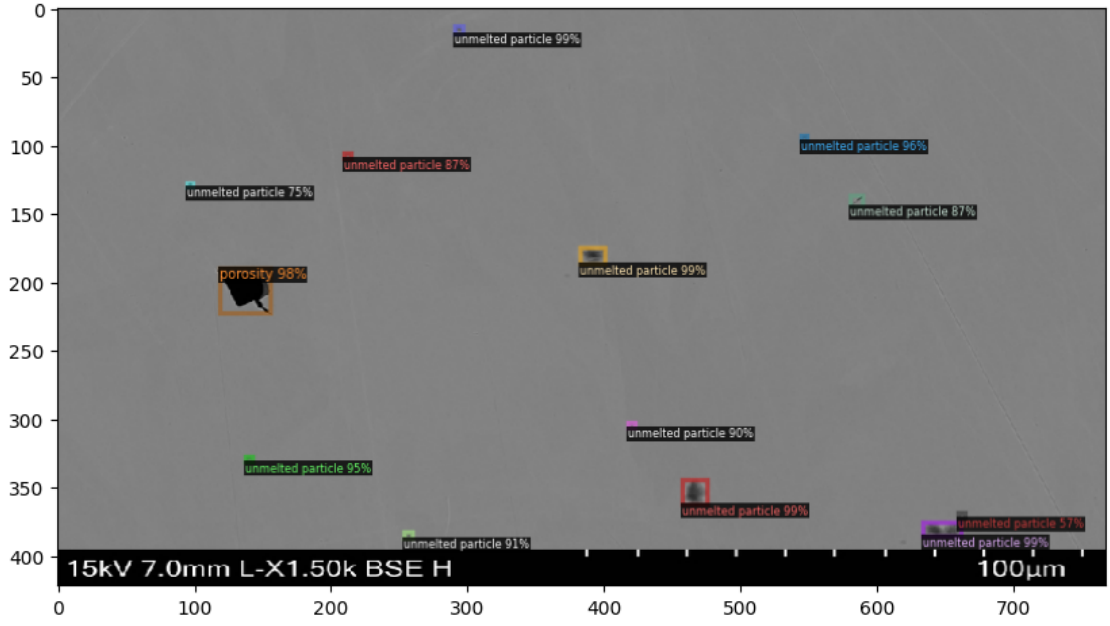
```
<IPython.core.display.Javascript object>
```

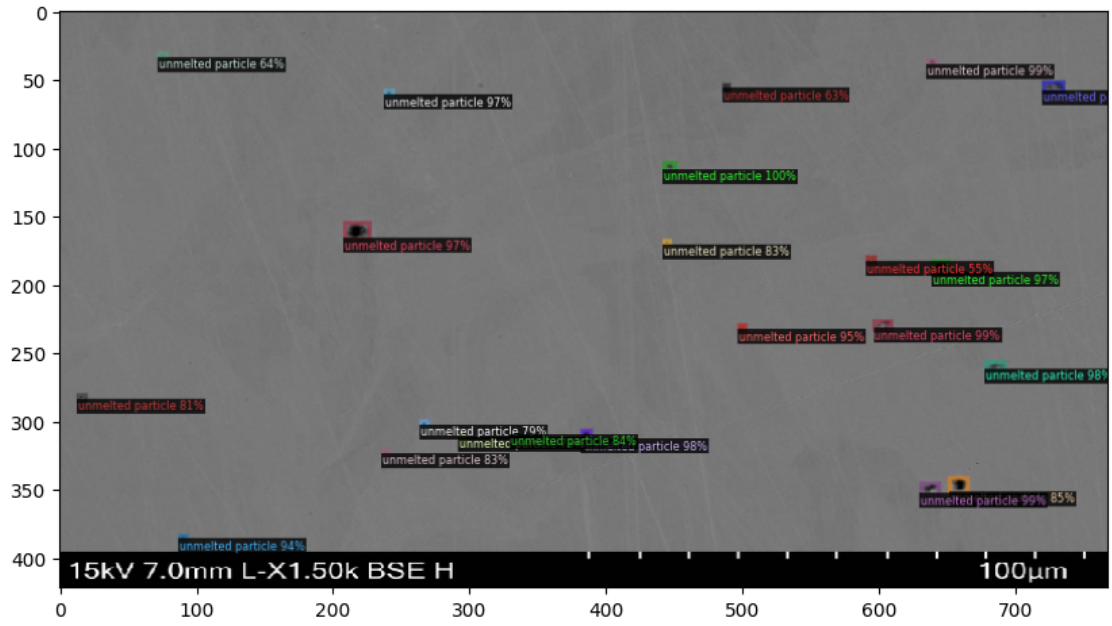
[ ]:
```
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
cfg.DATASETS.TEST = ("p_test", )
predictor = DefaultPredictor(cfg)
```

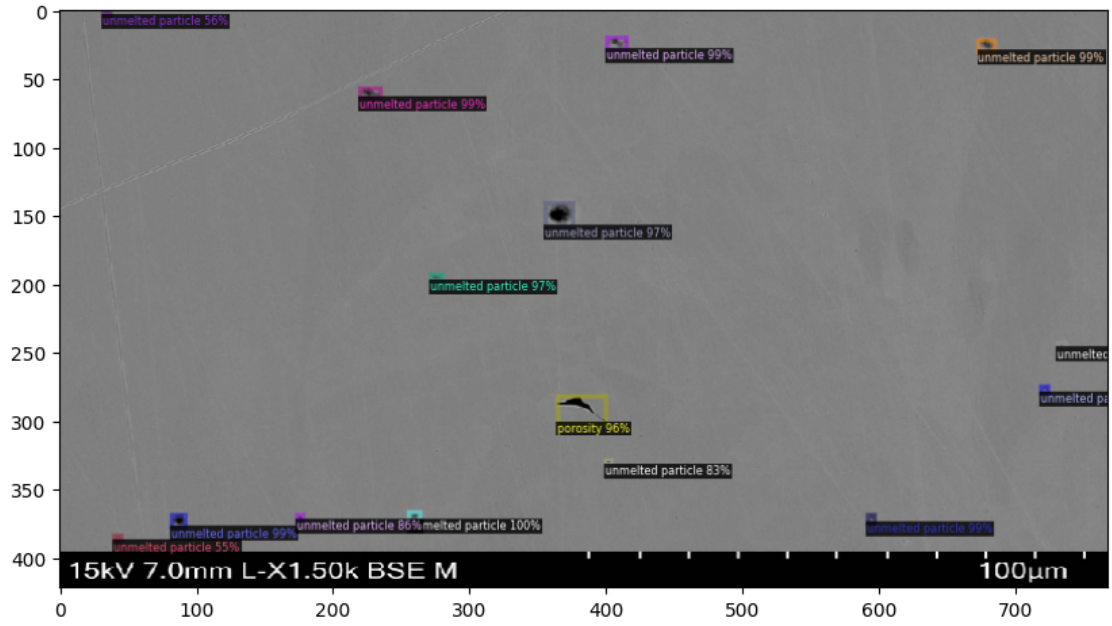
```
[07/21 20:17:08 d2.checkpoint.detection_checkpoint]: [DetectionCheckpointer]
Loading from ./output/model_final.pth …
```

[ ]:
```
from detectron2.utils.visualizer import ColorMode
dataset_dicts = get_r_dicts('/content/drive/MyDrive/Mahabub/train')
for d in random.sample(dataset_dicts, 4):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)
    v = Visualizer(im[:, :, ::-1],
                   metadata=r_metadata,
                   scale=0.8,
                   instance_mode=ColorMode.IMAGE_BW   # remove the colors of␣
 ↪unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    plt.figure(figsize = (10, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()
```


unmelted particle 82%
unmelted particle 92%
unmelted particle 68%
unmelted particle 98%
unmelted particle 84%
unmelted particle 99%
unmelted particle 99%
unmelted particle 99%
unmelted particle 82%
unmelted particle 97%
unmelted particle 78%
unmelted particle 100%
unmelted particle 96%
unmelted particle 99%
unmelted particle 99%
15kV 7.0mm L-X1.50k BSE M
100µm

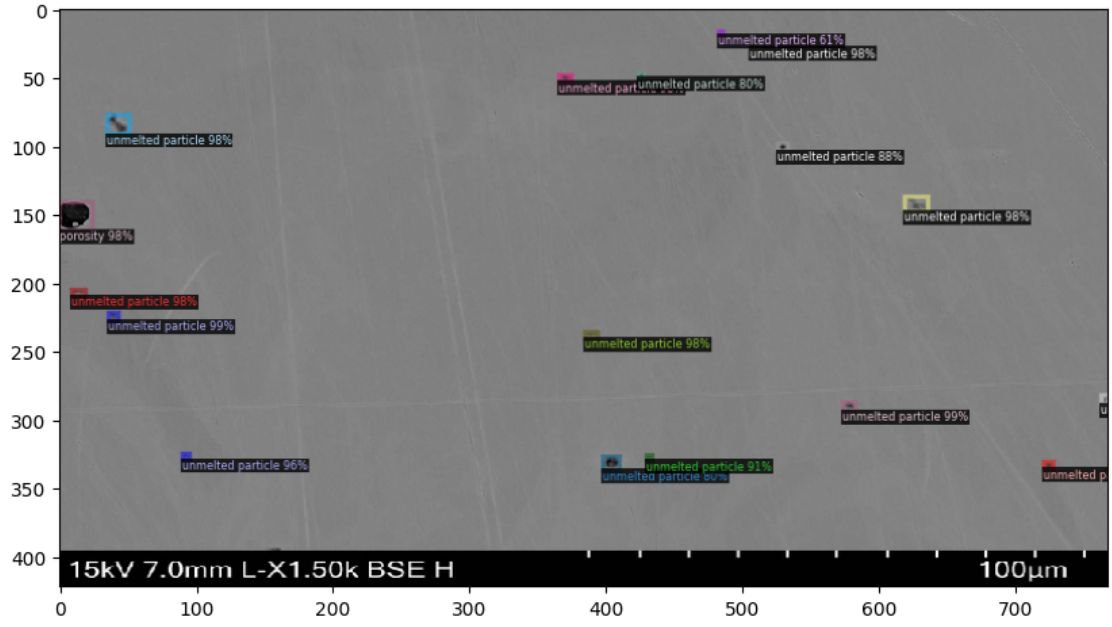

unmelted particle 99%
unmelted particle 96%
unmelted particle 87%
unmelted particle 75%
unmelted particle 87%
porosity 98%
unmelted particle 99%
unmelted particle 90%
unmelted particle 95%
unmelted particle 99%
unmelted particle 57%
unmelted particle 91%
unmelted particle 99%
15kV 7.0mm L-X1.50k BSE H
100µm

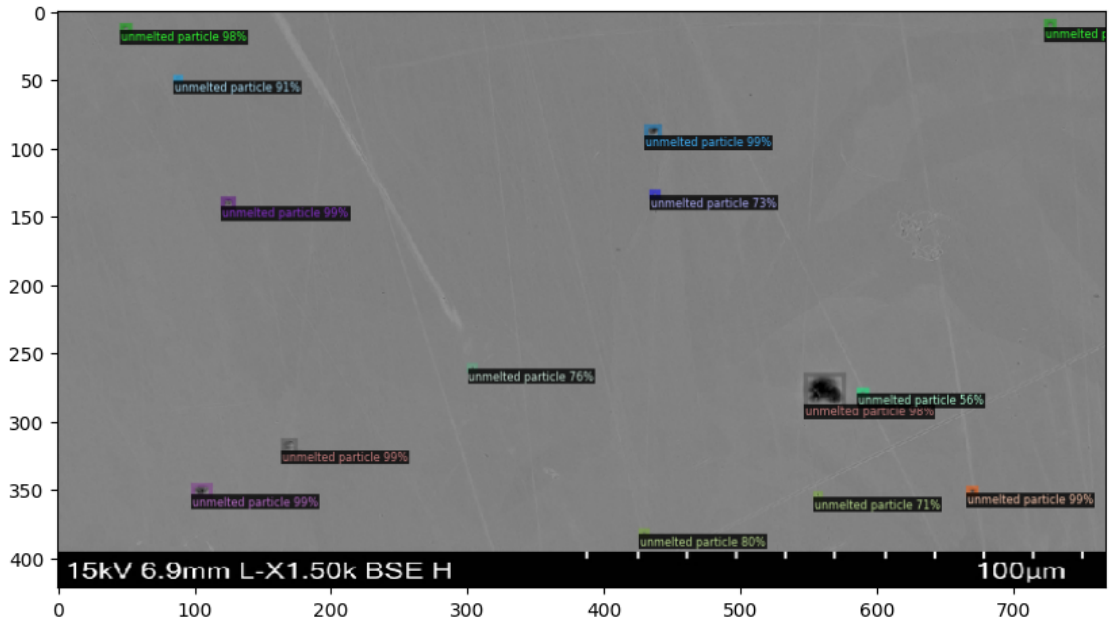


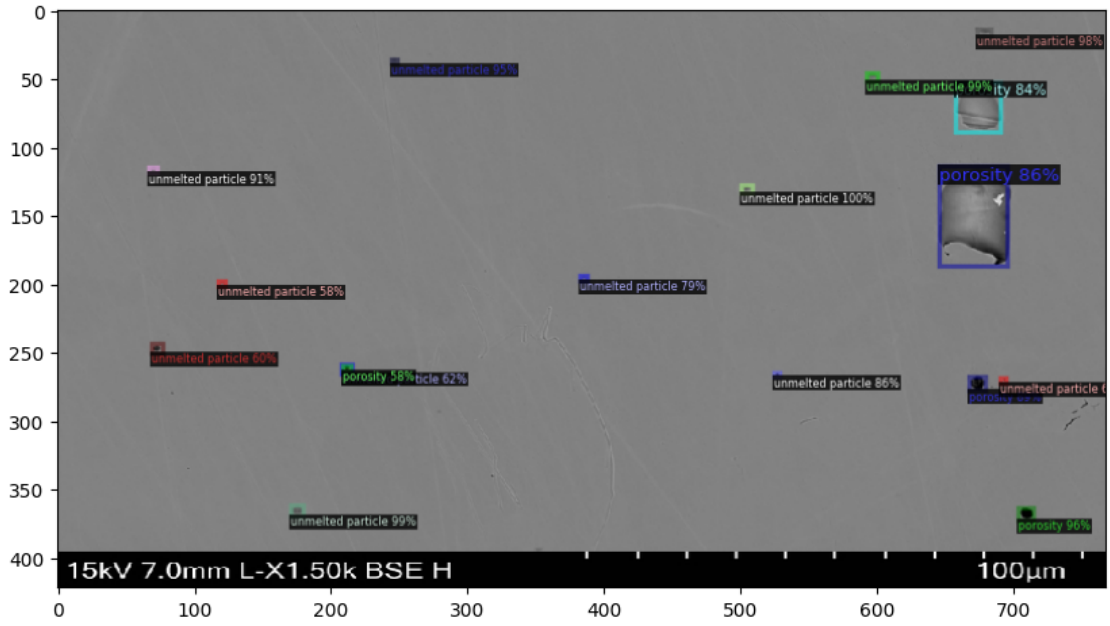


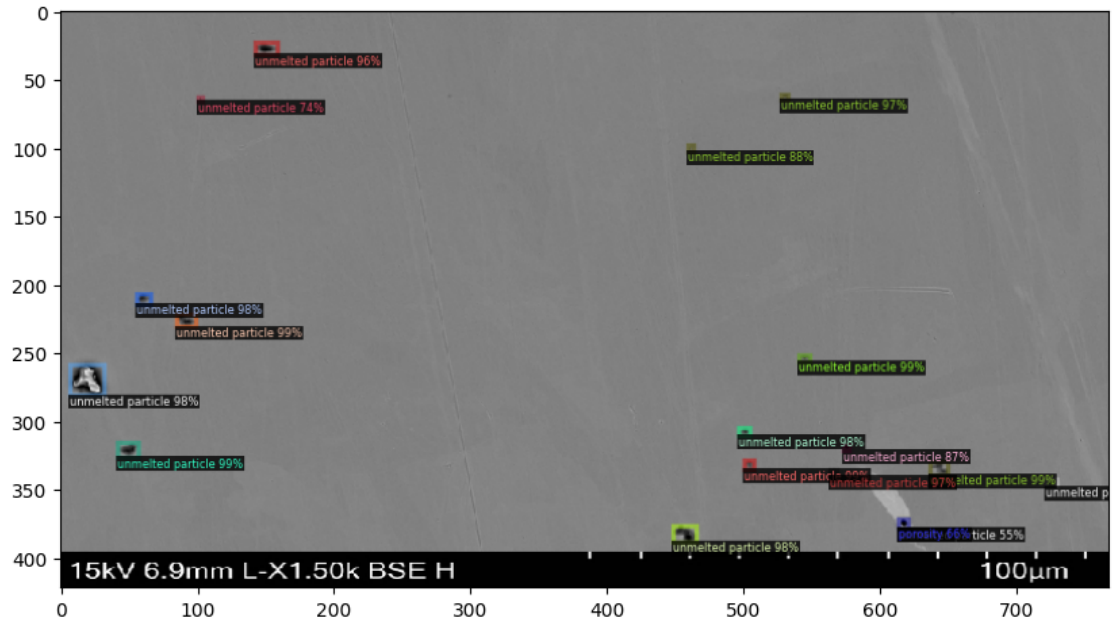

```
[ ]: from detectron2.utils.visualizer import ColorMode
     dataset_dicts = get_r_dicts('/content/drive/MyDrive/Mahabub/test')
     for d in random.sample(dataset_dicts, 4):
         im = cv2.imread(d["file_name"])
         outputs = predictor(im)
```

```
    v = Visualizer(im[:, :, ::-1],
                   metadata=r_metadata,
                   scale=0.8,
                   instance_mode=ColorMode.IMAGE_BW   # remove the colors of␣
↪unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    plt.figure(figsize = (10, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()
```




unmelted particle 98%
unmelted particle 91%
unmelted particle 99%
unmelted particle 99%
unmelted particle 73%
unmelted particle 76%
unmelted particle 56%
unmelted particle 99%
unmelted particle 99%
unmelted particle 71%
unmelted particle 99%
unmelted particle 80%
15kV 6.9mm L-X1.50k BSE H
100µm


unmelted particle 98%
unmelted particle 99%
unmelted particle 91%
porosity 86%
unmelted particle 100%
unmelted particle 58%
unmelted particle 79%
unmelted particle 60%
porosity 58%
unmelted particle 86%
unmelted particle 99%
porosity 96%
15kV 7.0mm L-X1.50k BSE H
100µm



[ ]:
```
from detectron2.evaluation import COCOEvaluator, inference_on_dataset
from detectron2.data import build_detection_test_loader
evaluator = COCOEvaluator("p_train", ['bbox'], False, output_dir="./output/")
val_loader = build_detection_test_loader(cfg, "p_train")
print(inference_on_dataset(predictor.model, val_loader, evaluator))
```

```
[07/21 20:17:54 d2.evaluation.coco_evaluation]: Trying to convert 'p_train' to
COCO format …
[07/21 20:17:54 d2.data.datasets.coco]: Converting annotations of dataset
'p_train' to COCO format …)
[07/21 20:17:54 d2.data.datasets.coco]: Converting dataset dicts into COCO
format
[07/21 20:17:54 d2.data.datasets.coco]: Conversion finished, #images: 42,
#annotations: 715
[07/21 20:17:54 d2.data.datasets.coco]: Caching COCO format annotations at
'./output/p_train_coco_format.json' …
[07/21 20:17:54 d2.data.dataset_mapper]: [DatasetMapper] Augmentations used in
inference: [ResizeShortestEdge(short_edge_length=(800, 800), max_size=1333,
sample_style='choice')]
[07/21 20:17:54 d2.data.common]: Serializing the dataset using: <class
'detectron2.data.common._TorchSerializedList'>
[07/21 20:17:54 d2.data.common]: Serializing 42 elements to byte tensors and
concatenating them all …
[07/21 20:17:54 d2.data.common]: Serialized dataset takes 0.16 MiB
[07/21 20:17:54 d2.evaluation.evaluator]: Start inference on 42 batches
[07/21 20:17:56 d2.evaluation.evaluator]: Inference done 11/42. Dataloading:
0.0014 s/iter. Inference: 0.1090 s/iter. Eval: 0.0003 s/iter. Total: 0.1107
```

```
s/iter. ETA=0:00:03
[07/21 20:17:59 d2.evaluation.evaluator]: Total inference time: 0:00:04.145786
(0.112048 s / iter per device, on 1 devices)
[07/21 20:17:59 d2.evaluation.evaluator]: Total inference pure compute time:
0:00:04 (0.108921 s / iter per device, on 1 devices)
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Preparing results for COCO
format …
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Saving results to
./output/coco_instances_results.json
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Evaluating predictions with
unofficial COCO API…
Loading and preparing results…
DONE (t=0.00s)
creating index…
index created!
[07/21 20:17:59 d2.evaluation.fast_eval_api]: Evaluate annotation type *bbox*
[07/21 20:17:59 d2.evaluation.fast_eval_api]: COCOeval_opt.evaluate() finished
in 0.02 seconds.
[07/21 20:17:59 d2.evaluation.fast_eval_api]: Accumulating evaluation results…
[07/21 20:17:59 d2.evaluation.fast_eval_api]: COCOeval_opt.accumulate() finished
in 0.01 seconds.
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.331
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.568
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.309
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.316
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.591
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.133
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.324
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.373
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.360
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.600
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl  |
|:------:|:------:|:------:|:------:|:------:|:-----:|
| 33.141 | 56.827 | 30.940 | 31.580 | 59.145 |  nan  |
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Some metrics cannot be computed
and is shown as NaN.
[07/21 20:17:59 d2.evaluation.coco_evaluation]: Per-category bbox AP:
| category          | AP     | category   | AP     | category   | AP    |
|:------------------|:-------|:-----------|:-------|:-----------|:------|
| unmelted particle | 42.747 | porosity   | 56.675 | microcrack | 0.000 |
OrderedDict([('bbox', {'AP': 33.14057936251722, 'AP50': 56.82736369549149,
'AP75': 30.939837821455303, 'APs': 31.580461289864736, 'APm': 59.14466446644665,
'APl': nan, 'AP-unmelted particle': 42.74711543399843, 'AP-porosity':
56.67462265355322, 'AP-microcrack': 0.0})])
```