```
!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-9ouujjrw
  Running command git clone --filter=blob:none --quiet
https://github.com/facebookresearch/detectron2.git /tmp/pip-req-build-
9ouujjrw
  Resolved https://github.com/facebookresearch/detectron2.git to
commit 57bdb21249d5418c130d54e2ebdc94dda7a4c01a
  Preparing metadata (setup.py) ... ent 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 1.2 MB/s eta
0:00:00
etadata (setup.py) ...  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 9.5 MB/s eta
0:00:00
 detectron2==0.6)
  Downloading hydra_core-1.3.2-py3-none-any.whl (154 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 154.5/154.5 kB 16.8 MB/s eta
0:00:00
 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 29.2 MB/s eta
```

```
0:00:00
ent 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 11.5 MB/s eta
0:00:00
etadata (setup.py) ...  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) ... e=detectron2-0.6-cp310-
cp310-linux_x86_64.whl size=6114338
sha256=a976f533f006e201800add615bf643d37ff5d549f91bc646ae20bb1ec78ed43
8
  Stored in directory:
/tmp/pip-ephem-wheel-cache-95vy96fp/wheels/47/e5/15/94c80df2ba85500c5d
76599cc307c0a7079d0e221bb6fc4375
  Building wheel for fvcore (setup.py) ... e=fvcore-
0.1.5.post20221221-py3-none-any.whl size=61405
sha256=19e3992237906d1add92de6e691831cbb09c52f4e4b185f32c163555a980b07
d
  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=144554
sha256=0a49064e8f234bb6260243dd510726907c4cee5959bb9594c03216486193d1c
1
  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-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)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/274.2 kB ? eta -:--:--
━━━━━━━━━━━━━━━━ ━━━━━━━━━━━━━━━━━━━━━━━ 112.6/274.2 kB 3.4 MB/s eta
0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 274.2/274.2 kB 5.2
MB/s eta 0:00:00
etadata (setup.py) ... l
  Building wheel for pyyaml (setup.py) ... l: filename=PyYAML-5.1-
cp310-cp310-linux_x86_64.whl size=44090
sha256=4fbb5a68d2a7e7acbc6f818aaf6f79187097536b5ba78ac141fdebfec478685
8
  Stored in directory:
/root/.cache/pip/wheels/70/83/31/975b737609aba39a4099d471d5684141c1fdc
3404f97e7f68a
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
```

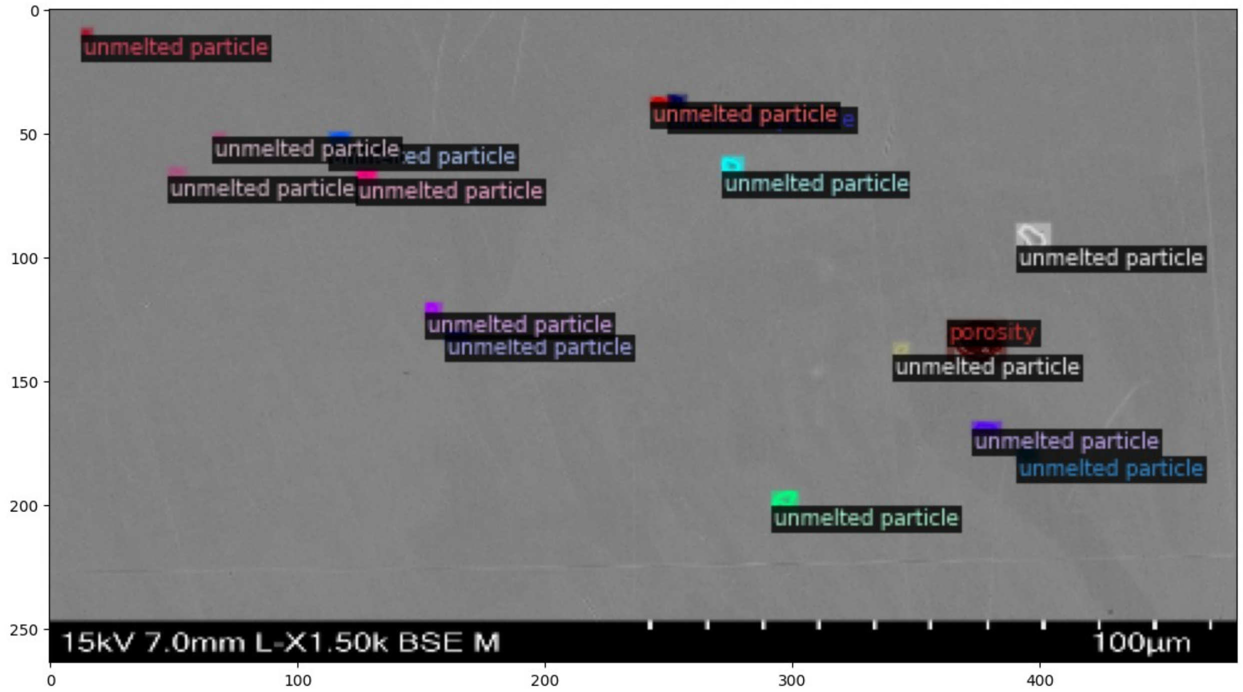
```
import torch, detectron2
!nvcc --version
TORCH_VERSION = ".".join(torch.__version__.split(".")[:2])
CUDA_VERSION = torch.__version__.split("+")[-1]
print("torch: ", TORCH_VERSION, "; cuda: ", CUDA_VERSION)
print("detectron2:", detectron2.__version__)
```

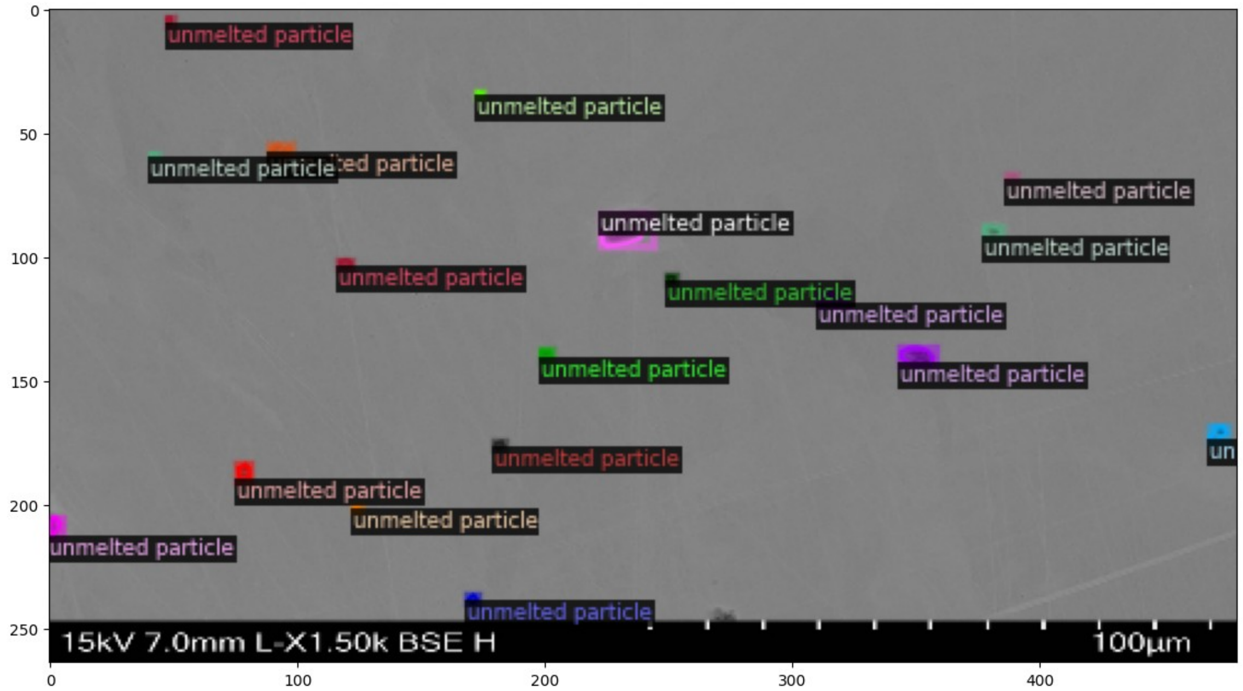
```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
torch:  2.0 ; cuda:  cu118
detectron2: 0.6
```

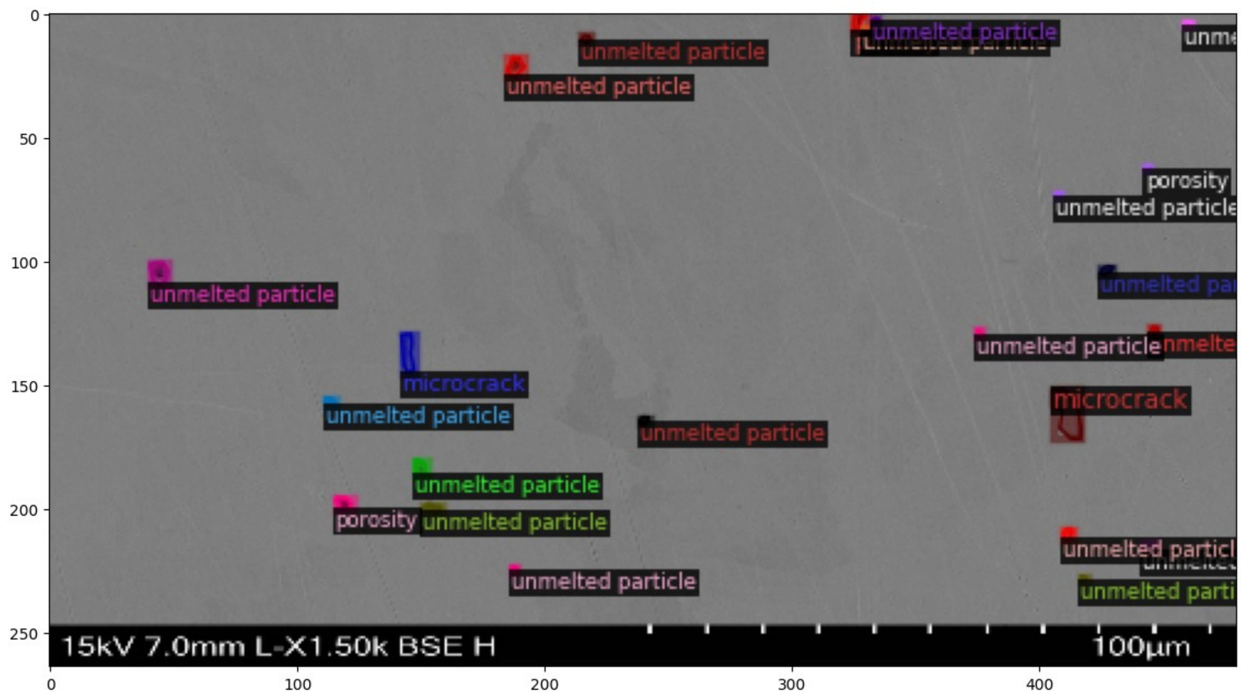
```
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# 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
```

```
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()
```


0
50
100
150
200
250
unmelted particle
porosity
15kV 7.0mm L-X1.50k BSE M
100µm
0
100
200
300
400


0
50
100
150
200
250
unmelted particle
un
15kV 7.0mm L-X1.50k BSE H
100µm
0
100
200
300
400



```
DatasetCatalog.remove("p_train")
DatasetCatalog.remove("p_test")

---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-16-9f969f888a51> in <cell line: 1>()
----> 1 DatasetCatalog.remove("p_train")
      2 DatasetCatalog.remove("p_test")

/usr/local/lib/python3.10/dist-packages/detectron2/data/catalog.py in remove(self, name)
     71         Alias of ``pop``.
     72         """
---> 73         self.pop(name)
     74 
     75     def __str__(self):

/usr/lib/python3.10/_collections_abc.py in pop(self, key, default)
    955         '''
    956         try:
--> 957             value = self[key]
    958         except KeyError:
    959             if default is self.__marker:

/usr/lib/python3.10/collections/__init__.py in __getitem__(self, key)
   1104         if hasattr(self.__class__, "__missing__"):
```

```
   1105             return self.__class__.__missing__(self, key)
-> 1106         raise KeyError(key)
   1107 
   1108     def __setitem__(self, key, item):

KeyError: 'p_train'
```

```python
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
from detectron2.model_zoo import model_zoo

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/retinanet_R_101_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("p_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = 
model_zoo.get_checkpoint_url("COCO-Detection/retinanet_R_101_FPN_3x.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.RETINANET.NUM_CLASSES = 3

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

```
WARNING:fvcore.common.config:Loading config
/usr/local/lib/python3.10/dist-packages/detectron2/model_zoo/configs/
COCO-Detection/../Base-RetinaNet.yaml with yaml.unsafe_load. Your
machine may be at risk if the file contains malicious content.

[07/25 11:20:58 d2.engine.defaults]: Model:
RetinaNet(
  (backbone): FPN(
    (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): LastLevelP6P7(
      (p6): Conv2d(2048, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
      (p7): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
    )
    (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)
          )
        )
        (6): 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)
          )
        )
        (7): 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)
          )
        )
        (8): 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)
          )
        )
        (9): 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)
          )
        )
        (10): 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)
          )
        )
        (11): 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)
          )
        )
        (12): 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)
          )
        )
        (13): 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)
          )
        )
        (14): 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)
          )
        )
        (15): 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)
          )
        )
        (16): 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)
          )
        )
        (17): 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)
          )
        )
        (18): 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)
          )
        )
        (19): 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)
          )
        )
        (20): 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)
          )
        )
        (21): 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)
          )
        )
        (22): 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)
          )
        )
      )
    )
  )
  (head): RetinaNetHead(
    (cls_subnet): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
```

```
      (1): ReLU()
      (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (3): ReLU()
      (4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (5): ReLU()
      (6): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (7): ReLU()
    )
    (bbox_subnet): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (1): ReLU()
      (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (3): ReLU()
      (4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (5): ReLU()
      (6): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
      (7): ReLU()
    )
    (cls_score): Conv2d(256, 27, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (bbox_pred): Conv2d(256, 36, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
  )
  (anchor_generator): DefaultAnchorGenerator(
    (cell_anchors): BufferList()
  )
)
[07/25 11:20:58 d2.data.build]: Removed 0 images with no usable
annotations. 42 images left.
[07/25 11:20:58 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          |            |              |
|               |
[07/25 11:20:58 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()]
[07/25 11:20:58 d2.data.build]: Using training sampler TrainingSampler
[07/25 11:20:58 d2.data.common]: Serializing the dataset using: <class
'detectron2.data.common._TorchSerializedList'>
[07/25 11:20:58 d2.data.common]: Serializing 42 elements to byte
tensors and concatenating them all ...
[07/25 11:20:58 d2.data.common]: Serialized dataset takes 0.16 MiB
[07/25 11:20:58 d2.checkpoint.detection_checkpoint]:
[DetectionCheckpointer] Loading from
https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/retinanet_R_1
01_FPN_3x/190397697/model_final_971ab9.pkl ...

model_final_971ab9.pkl: 228MB [00:04, 46.1MB/s]

WARNING:fvcore.common.checkpoint:Skip loading parameter
'head.cls_score.weight' to the model due to incompatible shapes: (720,
256, 3, 3) in the checkpoint but (27, 256, 3, 3) in the model! You
might want to double check if this is expected.
WARNING:fvcore.common.checkpoint:Skip loading parameter
'head.cls_score.bias' to the model due to incompatible shapes: (720,)
in the checkpoint but (27,) 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:
head.cls_score.{bias, weight}
WARNING:fvcore.common.checkpoint:The checkpoint state_dict contains
keys that are not used by the model:
  pixel_mean
  pixel_std

[07/25 11:21:03 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]

[07/25 11:21:17 d2.utils.events]:  eta: 0:10:53  iter: 19  total_loss:
1.898  loss_cls: 1.337  loss_box_reg: 0.5127    time: 0.5389
last_time: 0.2966  data_time: 0.0591  last_data_time: 0.0160   lr:
4.9953e-06  max_mem: 2833M
[07/25 11:21:26 d2.utils.events]:  eta: 0:09:42  iter: 39  total_loss:
1.342  loss_cls: 0.9546  loss_box_reg: 0.3698    time: 0.4002
last_time: 0.2556  data_time: 0.0075  last_data_time: 0.0091   lr:
9.9902e-06  max_mem: 2916M
[07/25 11:21:32 d2.utils.events]:  eta: 0:09:39  iter: 59  total_loss:
```

```
1.214  loss_cls: 0.8363  loss_box_reg: 0.3802    time: 0.3696
last_time: 0.2571  data_time: 0.0172  last_data_time: 0.0183   lr:
1.4985e-05  max_mem: 2916M
[07/25 11:21:37 d2.utils.events]:  eta: 0:09:20  iter: 79  total_loss:
1.089  loss_cls: 0.7315  loss_box_reg: 0.3571    time: 0.3419
last_time: 0.2470  data_time: 0.0082  last_data_time: 0.0061   lr:
1.998e-05  max_mem: 2916M
[07/25 11:21:43 d2.utils.events]:  eta: 0:09:06  iter: 99  total_loss:
1.077  loss_cls: 0.6711  loss_box_reg: 0.3927    time: 0.3278
last_time: 0.2880  data_time: 0.0096  last_data_time: 0.0103   lr:
2.4975e-05  max_mem: 2916M
[07/25 11:21:49 d2.utils.events]:  eta: 0:09:03  iter: 119
total_loss: 0.9542  loss_cls: 0.5712  loss_box_reg: 0.3491    time:
0.3238  last_time: 0.2899  data_time: 0.0138  last_data_time: 0.0080
lr: 2.997e-05  max_mem: 2916M
[07/25 11:21:54 d2.utils.events]:  eta: 0:08:46  iter: 139
total_loss: 0.9094  loss_cls: 0.5334  loss_box_reg: 0.3735    time:
0.3144  last_time: 0.2410  data_time: 0.0093  last_data_time: 0.0066
lr: 3.4965e-05  max_mem: 2916M
[07/25 11:22:01 d2.utils.events]:  eta: 0:08:47  iter: 159
total_loss: 0.8444  loss_cls: 0.4552  loss_box_reg: 0.3222    time:
0.3157  last_time: 0.3932  data_time: 0.0163  last_data_time: 0.0227
lr: 3.996e-05  max_mem: 2916M
[07/25 11:22:06 d2.utils.events]:  eta: 0:08:36  iter: 179
total_loss: 0.7493  loss_cls: 0.4229  loss_box_reg: 0.3589    time:
0.3107  last_time: 0.2479  data_time: 0.0108  last_data_time: 0.0106
lr: 4.4955e-05  max_mem: 2916M
[07/25 11:22:11 d2.utils.events]:  eta: 0:08:16  iter: 199
total_loss: 0.8247  loss_cls: 0.4436  loss_box_reg: 0.3443    time:
0.3057  last_time: 0.3068  data_time: 0.0079  last_data_time: 0.0255
lr: 4.995e-05  max_mem: 2916M
[07/25 11:22:17 d2.utils.events]:  eta: 0:08:19  iter: 219
total_loss: 0.8362  loss_cls: 0.4395  loss_box_reg: 0.3844    time:
0.3051  last_time: 0.2436  data_time: 0.0135  last_data_time: 0.0075
lr: 5.4945e-05  max_mem: 2916M
[07/25 11:22:22 d2.utils.events]:  eta: 0:07:59  iter: 239
total_loss: 0.7496  loss_cls: 0.4138  loss_box_reg: 0.3384    time:
0.2997  last_time: 0.2066  data_time: 0.0088  last_data_time: 0.0056
lr: 5.994e-05  max_mem: 2916M
[07/25 11:22:28 d2.utils.events]:  eta: 0:07:47  iter: 259
total_loss: 0.6571  loss_cls: 0.3488  loss_box_reg: 0.313    time:
0.2976  last_time: 0.2917  data_time: 0.0093  last_data_time: 0.0058
lr: 6.4935e-05  max_mem: 2916M
[07/25 11:22:33 d2.utils.events]:  eta: 0:07:46  iter: 279
total_loss: 0.6601  loss_cls: 0.3461  loss_box_reg: 0.3373    time:
0.2967  last_time: 0.3037  data_time: 0.0081  last_data_time: 0.0080
lr: 6.993e-05  max_mem: 2916M
[07/25 11:22:38 d2.utils.events]:  eta: 0:07:31  iter: 299
total_loss: 0.652  loss_cls: 0.3337  loss_box_reg: 0.3176    time:
```

```
0.2933  last_time: 0.2124  data_time: 0.0096  last_data_time: 0.0091
lr: 7.4925e-05  max_mem: 2916M
[07/25 11:22:44 d2.utils.events]:  eta: 0:07:28  iter: 319
total_loss: 0.7376  loss_cls: 0.3626  loss_box_reg: 0.3792    time:
0.2942  last_time: 0.3536  data_time: 0.0096  last_data_time: 0.0124
lr: 7.992e-05  max_mem: 2916M
[07/25 11:22:50 d2.utils.events]:  eta: 0:07:23  iter: 339
total_loss: 0.6173  loss_cls: 0.2935  loss_box_reg: 0.3294    time:
0.2929  last_time: 0.2999  data_time: 0.0079  last_data_time: 0.0073
lr: 8.4915e-05  max_mem: 2916M
[07/25 11:22:55 d2.utils.events]:  eta: 0:07:17  iter: 359
total_loss: 0.5881  loss_cls: 0.2813  loss_box_reg: 0.3242    time:
0.2916  last_time: 0.2426  data_time: 0.0067  last_data_time: 0.0057
lr: 8.991e-05  max_mem: 2916M
[07/25 11:23:02 d2.utils.events]:  eta: 0:07:15  iter: 379
total_loss: 0.6232  loss_cls: 0.2795  loss_box_reg: 0.3251    time:
0.2928  last_time: 0.2487  data_time: 0.0183  last_data_time: 0.0053
lr: 9.4905e-05  max_mem: 2916M
[07/25 11:23:07 d2.utils.events]:  eta: 0:07:11  iter: 399
total_loss: 0.5777  loss_cls: 0.2686  loss_box_reg: 0.3108    time:
0.2922  last_time: 0.3035  data_time: 0.0099  last_data_time: 0.0082
lr: 9.99e-05  max_mem: 2916M
[07/25 11:23:13 d2.utils.events]:  eta: 0:07:04  iter: 419
total_loss: 0.6232  loss_cls: 0.2818  loss_box_reg: 0.3244    time:
0.2920  last_time: 0.2892  data_time: 0.0113  last_data_time: 0.0226
lr: 0.0001049  max_mem: 2916M
[07/25 11:23:19 d2.utils.events]:  eta: 0:06:58  iter: 439
total_loss: 0.5455  loss_cls: 0.2584  loss_box_reg: 0.2989    time:
0.2915  last_time: 0.2955  data_time: 0.0091  last_data_time: 0.0103
lr: 0.00010989  max_mem: 2916M
[07/25 11:23:24 d2.utils.events]:  eta: 0:06:52  iter: 459
total_loss: 0.5962  loss_cls: 0.2516  loss_box_reg: 0.3166    time:
0.2904  last_time: 0.2484  data_time: 0.0079  last_data_time: 0.0075
lr: 0.00011489  max_mem: 2916M
[07/25 11:23:30 d2.utils.events]:  eta: 0:06:53  iter: 479
total_loss: 0.5636  loss_cls: 0.2575  loss_box_reg: 0.3079    time:
0.2914  last_time: 0.2468  data_time: 0.0171  last_data_time: 0.0076
lr: 0.00011988  max_mem: 2916M
[07/25 11:23:35 d2.utils.events]:  eta: 0:06:41  iter: 499
total_loss: 0.5418  loss_cls: 0.2414  loss_box_reg: 0.3037    time:
0.2898  last_time: 0.2116  data_time: 0.0083  last_data_time: 0.0087
lr: 0.00012488  max_mem: 2916M
[07/25 11:23:41 d2.utils.events]:  eta: 0:06:35  iter: 519
total_loss: 0.5535  loss_cls: 0.2746  loss_box_reg: 0.2808    time:
0.2895  last_time: 0.3319  data_time: 0.0094  last_data_time: 0.0229
lr: 0.00012987  max_mem: 2932M
[07/25 11:23:47 d2.utils.events]:  eta: 0:06:32  iter: 539
total_loss: 0.5542  loss_cls: 0.2335  loss_box_reg: 0.3208    time:
0.2896  last_time: 0.2667  data_time: 0.0090  last_data_time: 0.0059
```

```
lr: 0.00013487  max_mem: 2932M
[07/25 11:23:52 d2.utils.events]:  eta: 0:06:25  iter: 559
total_loss: 0.5331  loss_cls: 0.2238  loss_box_reg: 0.3102    time:
0.2888  last_time: 0.3183  data_time: 0.0079  last_data_time: 0.0089
lr: 0.00013986  max_mem: 2932M
[07/25 11:23:58 d2.utils.events]:  eta: 0:06:24  iter: 579
total_loss: 0.5427  loss_cls: 0.2479  loss_box_reg: 0.279    time:
0.2896  last_time: 0.3561  data_time: 0.0160  last_data_time: 0.0068
lr: 0.00014486  max_mem: 2932M
[07/25 11:24:04 d2.utils.events]:  eta: 0:06:19  iter: 599
total_loss: 0.5354  loss_cls: 0.2206  loss_box_reg: 0.308    time:
0.2891  last_time: 0.3023  data_time: 0.0100  last_data_time: 0.0060
lr: 0.00014985  max_mem: 2932M
[07/25 11:24:09 d2.utils.events]:  eta: 0:06:10  iter: 619
total_loss: 0.492  loss_cls: 0.2141  loss_box_reg: 0.2719    time:
0.2883  last_time: 0.3891  data_time: 0.0082  last_data_time: 0.0119
lr: 0.00015485  max_mem: 2932M
[07/25 11:24:16 d2.utils.events]:  eta: 0:06:08  iter: 639
total_loss: 0.543  loss_cls: 0.2301  loss_box_reg: 0.3034    time:
0.2892  last_time: 0.2080  data_time: 0.0129  last_data_time: 0.0070
lr: 0.00015984  max_mem: 2932M
[07/25 11:24:21 d2.utils.events]:  eta: 0:05:59  iter: 659
total_loss: 0.4984  loss_cls: 0.195  loss_box_reg: 0.2852    time:
0.2884  last_time: 0.2529  data_time: 0.0098  last_data_time: 0.0074
lr: 0.00016484  max_mem: 2932M
[07/25 11:24:27 d2.utils.events]:  eta: 0:05:57  iter: 679
total_loss: 0.5158  loss_cls: 0.2149  loss_box_reg: 0.3012    time:
0.2895  last_time: 0.2539  data_time: 0.0146  last_data_time: 0.0126
lr: 0.00016983  max_mem: 2932M
[07/25 11:24:33 d2.utils.events]:  eta: 0:05:52  iter: 699
total_loss: 0.4627  loss_cls: 0.1968  loss_box_reg: 0.2675    time:
0.2892  last_time: 0.2998  data_time: 0.0088  last_data_time: 0.0092
lr: 0.00017483  max_mem: 2932M
[07/25 11:24:38 d2.utils.events]:  eta: 0:05:42  iter: 719
total_loss: 0.5467  loss_cls: 0.2132  loss_box_reg: 0.3089    time:
0.2884  last_time: 0.2506  data_time: 0.0102  last_data_time: 0.0128
lr: 0.00017982  max_mem: 2932M
[07/25 11:24:44 d2.utils.events]:  eta: 0:05:41  iter: 739
total_loss: 0.4644  loss_cls: 0.1969  loss_box_reg: 0.2673    time:
0.2891  last_time: 0.3119  data_time: 0.0161  last_data_time: 0.0074
lr: 0.00018482  max_mem: 2932M
[07/25 11:24:50 d2.utils.events]:  eta: 0:05:34  iter: 759
total_loss: 0.4581  loss_cls: 0.1881  loss_box_reg: 0.269    time:
0.2886  last_time: 0.2601  data_time: 0.0104  last_data_time: 0.0072
lr: 0.00018981  max_mem: 2932M
[07/25 11:24:56 d2.utils.events]:  eta: 0:05:30  iter: 779
total_loss: 0.5036  loss_cls: 0.1826  loss_box_reg: 0.2979    time:
0.2892  last_time: 0.3019  data_time: 0.0110  last_data_time: 0.0193
lr: 0.00019481  max_mem: 2932M
```

```
[07/25 11:25:02 d2.utils.events]:  eta: 0:05:23  iter: 799
total_loss: 0.4101  loss_cls: 0.164  loss_box_reg: 0.242    time:
0.2887  last_time: 0.2241  data_time: 0.0093  last_data_time: 0.0068
lr: 0.0001998  max_mem: 2932M
[07/25 11:25:07 d2.utils.events]:  eta: 0:05:15  iter: 819
total_loss: 0.4922  loss_cls: 0.1933  loss_box_reg: 0.2786    time:
0.2878  last_time: 0.2495  data_time: 0.0089  last_data_time: 0.0091
lr: 0.0002048  max_mem: 2932M
[07/25 11:25:13 d2.utils.events]:  eta: 0:05:13  iter: 839
total_loss: 0.4722  loss_cls: 0.169  loss_box_reg: 0.2765    time:
0.2885  last_time: 0.2975  data_time: 0.0135  last_data_time: 0.0058
lr: 0.00020979  max_mem: 2932M
[07/25 11:25:18 d2.utils.events]:  eta: 0:05:07  iter: 859
total_loss: 0.4291  loss_cls: 0.1609  loss_box_reg: 0.2591    time:
0.2880  last_time: 0.2497  data_time: 0.0097  last_data_time: 0.0082
lr: 0.00021479  max_mem: 2932M
[07/25 11:25:24 d2.utils.events]:  eta: 0:05:03  iter: 879
total_loss: 0.4322  loss_cls: 0.1665  loss_box_reg: 0.2682    time:
0.2882  last_time: 0.2257  data_time: 0.0124  last_data_time: 0.0066
lr: 0.00021978  max_mem: 2932M
[07/25 11:25:30 d2.utils.events]:  eta: 0:04:58  iter: 899
total_loss: 0.4773  loss_cls: 0.1971  loss_box_reg: 0.2855    time:
0.2883  last_time: 0.3308  data_time: 0.0082  last_data_time: 0.0055
lr: 0.00022478  max_mem: 2932M
[07/25 11:25:36 d2.utils.events]:  eta: 0:04:54  iter: 919
total_loss: 0.3905  loss_cls: 0.1719  loss_box_reg: 0.2201    time:
0.2884  last_time: 0.3159  data_time: 0.0102  last_data_time: 0.0113
lr: 0.00022977  max_mem: 2932M
[07/25 11:25:43 d2.utils.events]:  eta: 0:04:49  iter: 939
total_loss: 0.4373  loss_cls: 0.1593  loss_box_reg: 0.2642    time:
0.2893  last_time: 0.2525  data_time: 0.0149  last_data_time: 0.0067
lr: 0.00023477  max_mem: 2932M
[07/25 11:25:48 d2.utils.events]:  eta: 0:04:43  iter: 959
total_loss: 0.4123  loss_cls: 0.1585  loss_box_reg: 0.2487    time:
0.2889  last_time: 0.2918  data_time: 0.0092  last_data_time: 0.0058
lr: 0.00023976  max_mem: 2932M
[07/25 11:25:54 d2.utils.events]:  eta: 0:04:38  iter: 979
total_loss: 0.4576  loss_cls: 0.1672  loss_box_reg: 0.2805    time:
0.2892  last_time: 0.3940  data_time: 0.0073  last_data_time: 0.0066
lr: 0.00024476  max_mem: 2932M
[07/25 11:26:00 d2.utils.events]:  eta: 0:04:32  iter: 999
total_loss: 0.3946  loss_cls: 0.1454  loss_box_reg: 0.2335    time:
0.2890  last_time: 0.2456  data_time: 0.0089  last_data_time: 0.0073
lr: 0.00024975  max_mem: 2932M
[07/25 11:26:05 d2.utils.events]:  eta: 0:04:24  iter: 1019
total_loss: 0.4281  loss_cls: 0.1604  loss_box_reg: 0.2676    time:
0.2885  last_time: 0.3365  data_time: 0.0071  last_data_time: 0.0079
lr: 0.00025  max_mem: 2932M
[07/25 11:26:11 d2.utils.events]:  eta: 0:04:20  iter: 1039
```

```
total_loss: 0.4316  loss_cls: 0.1485  loss_box_reg: 0.2804    time:
0.2891  last_time: 0.3082  data_time: 0.0081  last_data_time: 0.0051
lr: 0.00025  max_mem: 2932M
[07/25 11:26:17 d2.utils.events]:  eta: 0:04:11  iter: 1059
total_loss: 0.4312  loss_cls: 0.1641  loss_box_reg: 0.2569    time:
0.2888  last_time: 0.2277  data_time: 0.0077  last_data_time: 0.0075
lr: 0.00025  max_mem: 2932M
[07/25 11:26:23 d2.utils.events]:  eta: 0:04:09  iter: 1079
total_loss: 0.3822  loss_cls: 0.155  loss_box_reg: 0.2262    time:
0.2889  last_time: 0.2423  data_time: 0.0092  last_data_time: 0.0119
lr: 0.00025  max_mem: 2933M
[07/25 11:26:29 d2.utils.events]:  eta: 0:04:04  iter: 1099
total_loss: 0.4115  loss_cls: 0.1493  loss_box_reg: 0.246    time:
0.2893  last_time: 0.3578  data_time: 0.0076  last_data_time: 0.0059
lr: 0.00025  max_mem: 2933M
[07/25 11:26:34 d2.utils.events]:  eta: 0:03:56  iter: 1119
total_loss: 0.371  loss_cls: 0.1371  loss_box_reg: 0.2381    time:
0.2889  last_time: 0.1974  data_time: 0.0079  last_data_time: 0.0091
lr: 0.00025  max_mem: 2933M
[07/25 11:26:41 d2.utils.events]:  eta: 0:03:53  iter: 1139
total_loss: 0.4015  loss_cls: 0.1447  loss_box_reg: 0.2621    time:
0.2894  last_time: 0.2518  data_time: 0.0169  last_data_time: 0.0067
lr: 0.00025  max_mem: 2933M
[07/25 11:26:46 d2.utils.events]:  eta: 0:03:46  iter: 1159
total_loss: 0.4044  loss_cls: 0.1542  loss_box_reg: 0.2589    time:
0.2890  last_time: 0.2489  data_time: 0.0087  last_data_time: 0.0067
lr: 0.00025  max_mem: 2933M
[07/25 11:26:52 d2.utils.events]:  eta: 0:03:42  iter: 1179
total_loss: 0.3583  loss_cls: 0.1276  loss_box_reg: 0.2335    time:
0.2892  last_time: 0.2796  data_time: 0.0106  last_data_time: 0.0142
lr: 0.00025  max_mem: 2933M
[07/25 11:26:58 d2.utils.events]:  eta: 0:03:38  iter: 1199
total_loss: 0.402  loss_cls: 0.1527  loss_box_reg: 0.2519    time:
0.2892  last_time: 0.2015  data_time: 0.0099  last_data_time: 0.0077
lr: 0.00025  max_mem: 2933M
[07/25 11:27:03 d2.utils.events]:  eta: 0:03:31  iter: 1219
total_loss: 0.3374  loss_cls: 0.1303  loss_box_reg: 0.213    time:
0.2889  last_time: 0.2983  data_time: 0.0080  last_data_time: 0.0082
lr: 0.00025  max_mem: 2933M
[07/25 11:27:09 d2.utils.events]:  eta: 0:03:28  iter: 1239
total_loss: 0.3911  loss_cls: 0.1303  loss_box_reg: 0.247    time:
0.2891  last_time: 0.2958  data_time: 0.0148  last_data_time: 0.0248
lr: 0.00025  max_mem: 2933M
[07/25 11:27:15 d2.utils.events]:  eta: 0:03:23  iter: 1259
total_loss: 0.3475  loss_cls: 0.1227  loss_box_reg: 0.2129    time:
0.2888  last_time: 0.2170  data_time: 0.0095  last_data_time: 0.0068
lr: 0.00025  max_mem: 2933M
[07/25 11:27:21 d2.utils.events]:  eta: 0:03:18  iter: 1279
total_loss: 0.3463  loss_cls: 0.1254  loss_box_reg: 0.2277    time:
```

```
0.2888  last_time: 0.2898  data_time: 0.0088  last_data_time: 0.0193
lr: 0.00025  max_mem: 2933M
[07/25 11:27:27 d2.utils.events]:  eta: 0:03:14  iter: 1299
total_loss: 0.375  loss_cls: 0.1275  loss_box_reg: 0.243    time:
0.2891  last_time: 0.2427  data_time: 0.0120  last_data_time: 0.0065
lr: 0.00025  max_mem: 2933M
[07/25 11:27:32 d2.utils.events]:  eta: 0:03:08  iter: 1319
total_loss: 0.3391  loss_cls: 0.1131  loss_box_reg: 0.2216    time:
0.2888  last_time: 0.3229  data_time: 0.0084  last_data_time: 0.0060
lr: 0.00025  max_mem: 2933M
[07/25 11:27:38 d2.utils.events]:  eta: 0:03:02  iter: 1339
total_loss: 0.3546  loss_cls: 0.1228  loss_box_reg: 0.2201    time:
0.2889  last_time: 0.3129  data_time: 0.0119  last_data_time: 0.0258
lr: 0.00025  max_mem: 2933M
[07/25 11:27:44 d2.utils.events]:  eta: 0:02:56  iter: 1359
total_loss: 0.328  loss_cls: 0.12  loss_box_reg: 0.2137    time:
0.2888  last_time: 0.3208  data_time: 0.0090  last_data_time: 0.0070
lr: 0.00025  max_mem: 2933M
[07/25 11:27:49 d2.utils.events]:  eta: 0:02:51  iter: 1379
total_loss: 0.3467  loss_cls: 0.1137  loss_box_reg: 0.2321    time:
0.2887  last_time: 0.3107  data_time: 0.0107  last_data_time: 0.0180
lr: 0.00025  max_mem: 2933M
[07/25 11:27:56 d2.utils.events]:  eta: 0:02:45  iter: 1399
total_loss: 0.3683  loss_cls: 0.1364  loss_box_reg: 0.2283    time:
0.2890  last_time: 0.2216  data_time: 0.0083  last_data_time: 0.0072
lr: 0.00025  max_mem: 2933M
[07/25 11:28:01 d2.utils.events]:  eta: 0:02:39  iter: 1419
total_loss: 0.3881  loss_cls: 0.1202  loss_box_reg: 0.2345    time:
0.2887  last_time: 0.2540  data_time: 0.0081  last_data_time: 0.0068
lr: 0.00025  max_mem: 2933M
[07/25 11:28:07 d2.utils.events]:  eta: 0:02:34  iter: 1439
total_loss: 0.3022  loss_cls: 0.1028  loss_box_reg: 0.21    time:
0.2891  last_time: 0.4277  data_time: 0.0148  last_data_time: 0.0332
lr: 0.00025  max_mem: 2933M
[07/25 11:28:13 d2.utils.events]:  eta: 0:02:28  iter: 1459
total_loss: 0.3418  loss_cls: 0.1177  loss_box_reg: 0.2241    time:
0.2888  last_time: 0.2504  data_time: 0.0087  last_data_time: 0.0054
lr: 0.00025  max_mem: 2933M
[07/25 11:28:18 d2.utils.events]:  eta: 0:02:21  iter: 1479
total_loss: 0.3189  loss_cls: 0.1102  loss_box_reg: 0.2212    time:
0.2886  last_time: 0.2637  data_time: 0.0079  last_data_time: 0.0062
lr: 0.00025  max_mem: 2933M
[07/25 11:28:24 d2.utils.events]:  eta: 0:02:18  iter: 1499
total_loss: 0.3248  loss_cls: 0.1065  loss_box_reg: 0.2139    time:
0.2889  last_time: 0.3436  data_time: 0.0107  last_data_time: 0.0062
lr: 0.00025  max_mem: 2933M
[07/25 11:28:30 d2.utils.events]:  eta: 0:02:12  iter: 1519
total_loss: 0.2947  loss_cls: 0.09337  loss_box_reg: 0.1959    time:
0.2887  last_time: 0.3386  data_time: 0.0093  last_data_time: 0.0059
```

```
lr: 0.00025  max_mem: 2933M
[07/25 11:28:36 d2.utils.events]:  eta: 0:02:07  iter: 1539
total_loss: 0.3202  loss_cls: 0.1131  loss_box_reg: 0.2114    time:
0.2890  last_time: 0.3257  data_time: 0.0129  last_data_time: 0.0306
lr: 0.00025  max_mem: 2933M
[07/25 11:28:42 d2.utils.events]:  eta: 0:02:01  iter: 1559
total_loss: 0.2794  loss_cls: 0.09443  loss_box_reg: 0.1783    time:
0.2890  last_time: 0.2508  data_time: 0.0091  last_data_time: 0.0077
lr: 0.00025  max_mem: 2933M
[07/25 11:28:48 d2.utils.events]:  eta: 0:01:55  iter: 1579
total_loss: 0.2951  loss_cls: 0.1025  loss_box_reg: 0.2068    time:
0.2889  last_time: 0.3952  data_time: 0.0097  last_data_time: 0.0060
lr: 0.00025  max_mem: 2933M
[07/25 11:28:54 d2.utils.events]:  eta: 0:01:50  iter: 1599
total_loss: 0.293  loss_cls: 0.1096  loss_box_reg: 0.1861    time:
0.2892  last_time: 0.2447  data_time: 0.0125  last_data_time: 0.0084
lr: 0.00025  max_mem: 2933M
[07/25 11:28:59 d2.utils.events]:  eta: 0:01:45  iter: 1619
total_loss: 0.3113  loss_cls: 0.09173  loss_box_reg: 0.2019    time:
0.2890  last_time: 0.2973  data_time: 0.0084  last_data_time: 0.0055
lr: 0.00025  max_mem: 2933M
[07/25 11:29:06 d2.utils.events]:  eta: 0:01:39  iter: 1639
total_loss: 0.2679  loss_cls: 0.08808  loss_box_reg: 0.1852    time:
0.2894  last_time: 0.4436  data_time: 0.0132  last_data_time: 0.0306
lr: 0.00025  max_mem: 2933M
[07/25 11:29:11 d2.utils.events]:  eta: 0:01:34  iter: 1659
total_loss: 0.335  loss_cls: 0.1045  loss_box_reg: 0.2239    time:
0.2892  last_time: 0.3349  data_time: 0.0082  last_data_time: 0.0051
lr: 0.00025  max_mem: 2933M
[07/25 11:29:17 d2.utils.events]:  eta: 0:01:28  iter: 1679
total_loss: 0.3034  loss_cls: 0.09743  loss_box_reg: 0.2036    time:
0.2890  last_time: 0.2622  data_time: 0.0093  last_data_time: 0.0203
lr: 0.00025  max_mem: 2933M
[07/25 11:29:23 d2.utils.events]:  eta: 0:01:22  iter: 1699
total_loss: 0.3534  loss_cls: 0.1181  loss_box_reg: 0.2276    time:
0.2895  last_time: 0.2460  data_time: 0.0127  last_data_time: 0.0075
lr: 0.00025  max_mem: 2933M
[07/25 11:29:28 d2.utils.events]:  eta: 0:01:17  iter: 1719
total_loss: 0.308  loss_cls: 0.104  loss_box_reg: 0.1978    time:
0.2890  last_time: 0.3030  data_time: 0.0085  last_data_time: 0.0070
lr: 0.00025  max_mem: 2933M
[07/25 11:29:34 d2.utils.events]:  eta: 0:01:10  iter: 1739
total_loss: 0.2836  loss_cls: 0.0934  loss_box_reg: 0.1862    time:
0.2891  last_time: 0.3960  data_time: 0.0116  last_data_time: 0.0059
lr: 0.00025  max_mem: 2933M
[07/25 11:29:41 d2.utils.events]:  eta: 0:01:06  iter: 1759
total_loss: 0.2937  loss_cls: 0.08975  loss_box_reg: 0.2042    time:
0.2894  last_time: 0.3137  data_time: 0.0111  last_data_time: 0.0057
lr: 0.00025  max_mem: 2933M
```

```
[07/25 11:29:46 d2.utils.events]:  eta: 0:01:00  iter: 1779
total_loss: 0.2816  loss_cls: 0.07974  loss_box_reg: 0.1917    time:
0.2893  last_time: 0.2762  data_time: 0.0075  last_data_time: 0.0089
lr: 0.00025  max_mem: 2933M
[07/25 11:29:53 d2.utils.events]:  eta: 0:00:55  iter: 1799
total_loss: 0.2546  loss_cls: 0.08115  loss_box_reg: 0.1726    time:
0.2896  last_time: 0.2601  data_time: 0.0143  last_data_time: 0.0216
lr: 0.00025  max_mem: 2933M
[07/25 11:29:59 d2.utils.events]:  eta: 0:00:50  iter: 1819
total_loss: 0.3138  loss_cls: 0.1011  loss_box_reg: 0.1969    time:
0.2897  last_time: 0.2998  data_time: 0.0094  last_data_time: 0.0069
lr: 0.00025  max_mem: 2933M
[07/25 11:30:05 d2.utils.events]:  eta: 0:00:45  iter: 1839
total_loss: 0.2967  loss_cls: 0.09357  loss_box_reg: 0.2011    time:
0.2902  last_time: 0.2800  data_time: 0.0109  last_data_time: 0.0080
lr: 0.00025  max_mem: 2933M
[07/25 11:30:10 d2.utils.events]:  eta: 0:00:39  iter: 1859
total_loss: 0.2646  loss_cls: 0.07796  loss_box_reg: 0.1805    time:
0.2899  last_time: 0.2939  data_time: 0.0072  last_data_time: 0.0052
lr: 0.00025  max_mem: 2933M
[07/25 11:30:16 d2.utils.events]:  eta: 0:00:33  iter: 1879
total_loss: 0.2573  loss_cls: 0.07447  loss_box_reg: 0.1799    time:
0.2896  last_time: 0.4247  data_time: 0.0104  last_data_time: 0.0059
lr: 0.00025  max_mem: 2933M
[07/25 11:30:22 d2.utils.events]:  eta: 0:00:27  iter: 1899
total_loss: 0.2752  loss_cls: 0.08372  loss_box_reg: 0.1788    time:
0.2898  last_time: 0.3119  data_time: 0.0167  last_data_time: 0.0083
lr: 0.00025  max_mem: 2933M
[07/25 11:30:28 d2.utils.events]:  eta: 0:00:22  iter: 1919
total_loss: 0.2944  loss_cls: 0.09137  loss_box_reg: 0.2042    time:
0.2898  last_time: 0.2218  data_time: 0.0081  last_data_time: 0.0121
lr: 0.00025  max_mem: 2933M
[07/25 11:30:35 d2.utils.events]:  eta: 0:00:16  iter: 1939
total_loss: 0.275  loss_cls: 0.08577  loss_box_reg: 0.1877    time:
0.2904  last_time: 0.2741  data_time: 0.0155  last_data_time: 0.0073
lr: 0.00025  max_mem: 2933M
[07/25 11:30:40 d2.utils.events]:  eta: 0:00:11  iter: 1959
total_loss: 0.2537  loss_cls: 0.08225  loss_box_reg: 0.1816    time:
0.2903  last_time: 0.2557  data_time: 0.0084  last_data_time: 0.0096
lr: 0.00025  max_mem: 2933M
[07/25 11:30:46 d2.utils.events]:  eta: 0:00:05  iter: 1979
total_loss: 0.2563  loss_cls: 0.08571  loss_box_reg: 0.1791    time:
0.2903  last_time: 0.3197  data_time: 0.0108  last_data_time: 0.0206
lr: 0.00025  max_mem: 2933M
[07/25 11:30:53 d2.utils.events]:  eta: 0:00:00  iter: 1999
total_loss: 0.2599  loss_cls: 0.07856  loss_box_reg: 0.1832    time:
0.2904  last_time: 0.2115  data_time: 0.0096  last_data_time: 0.0068
lr: 0.00025  max_mem: 2933M
[07/25 11:30:54 d2.engine.hooks]: Overall training speed: 1998
iterations in 0:09:40 (0.2904 s / it)
```

```
[07/25 11:30:54 d2.engine.hooks]: Total training time: 0:09:46
(0:00:05 on hooks)
```

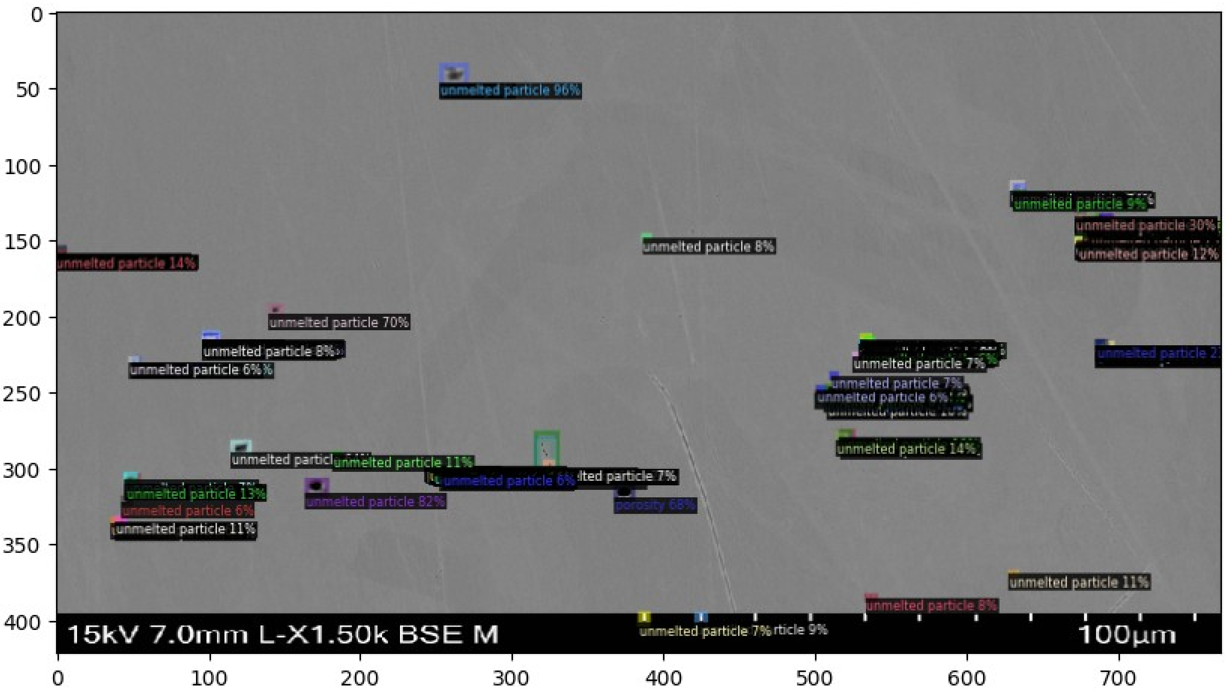
```
# Look at training curves in tensorboard:

%load_ext tensorboard
%tensorboard --logdir output
```

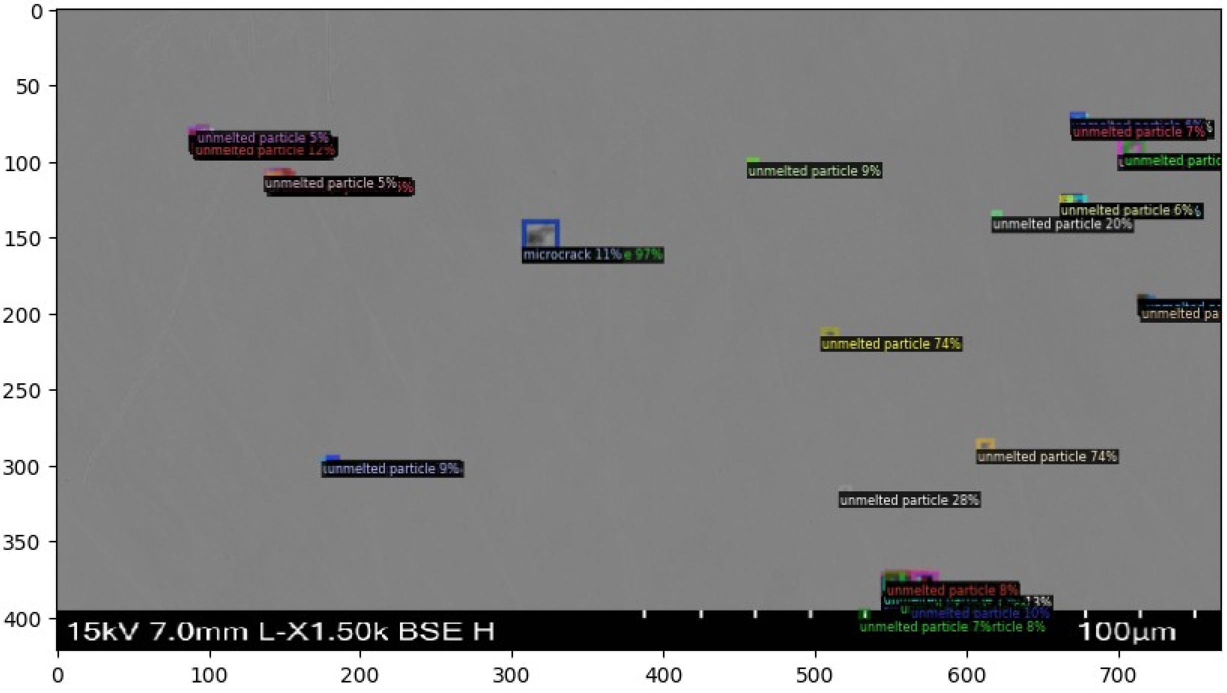
```
<IPython.core.display.Javascript object>
```

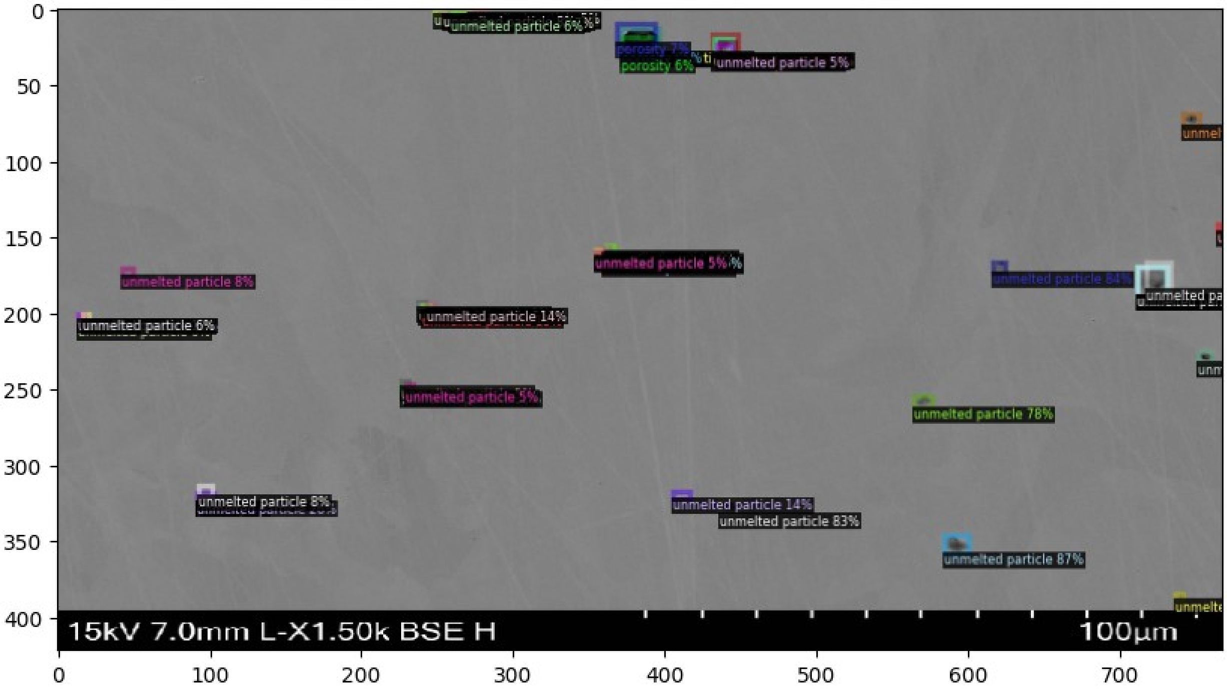
```
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
cfg.DATASETS.TEST = ("p_test", )
predictor = DefaultPredictor(cfg)
```

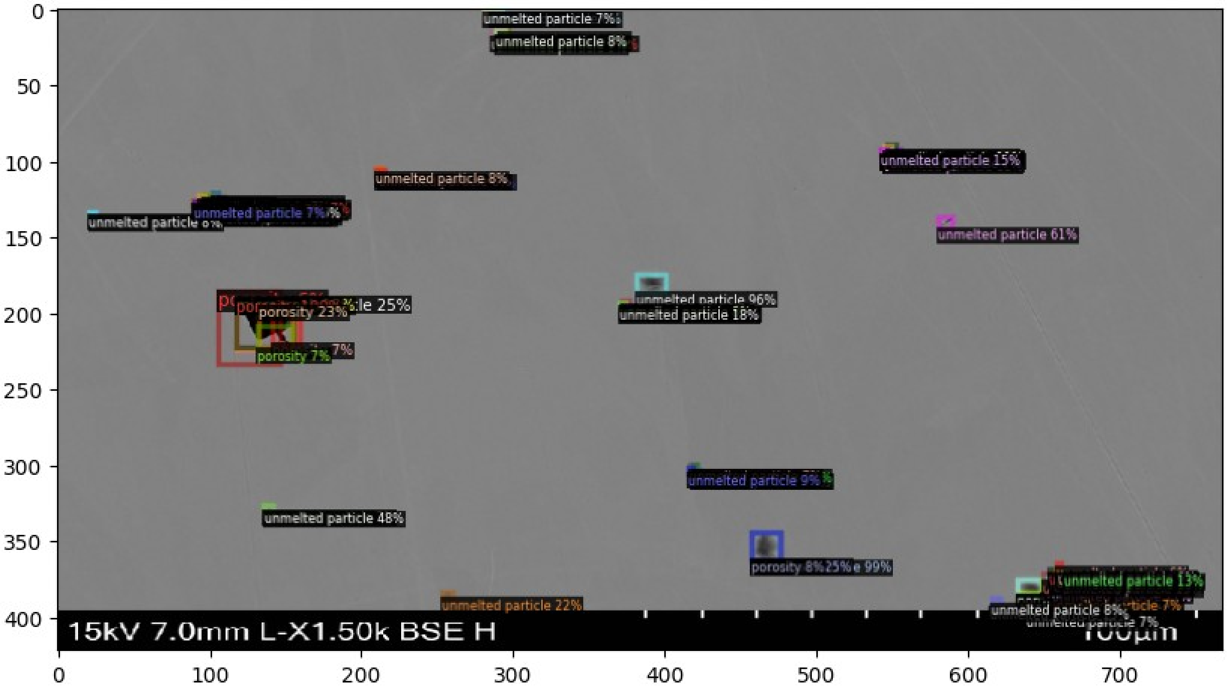
```
[07/25 11:31:48 d2.checkpoint.detection_checkpoint]:
[DetectionCheckpointer] Loading from ./output/model_final.pth ...
```

```
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()
```

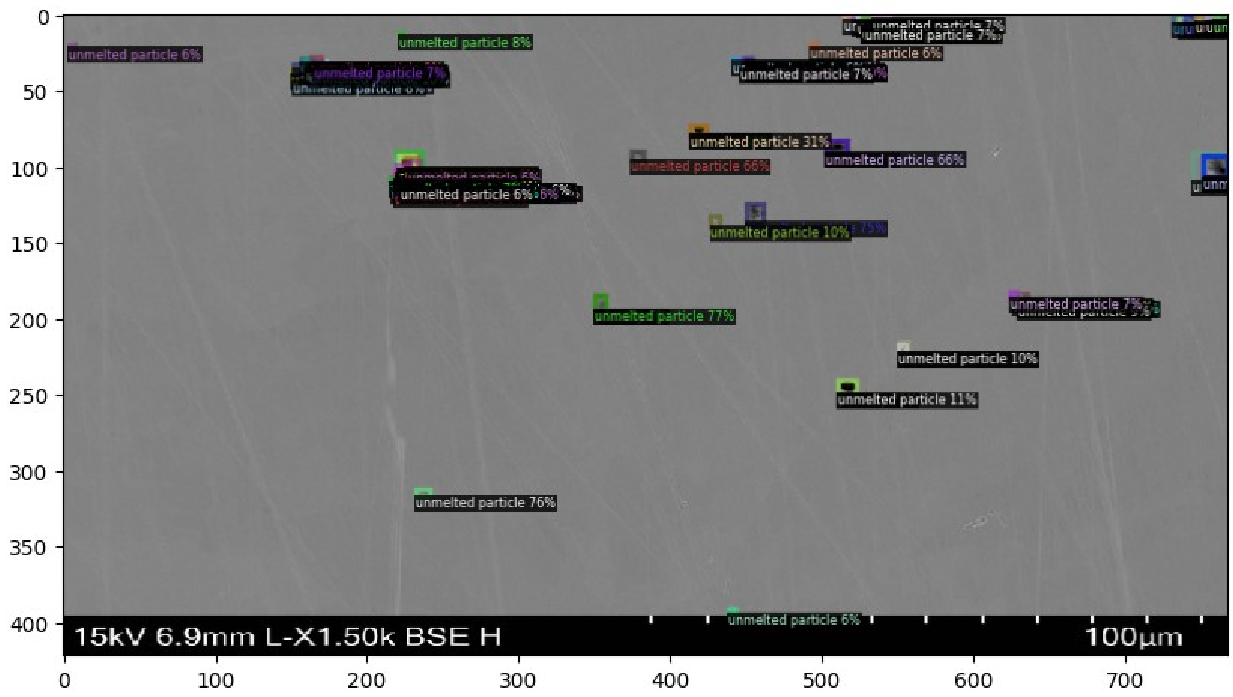

unmelted particle 96%
unmelted particle 9%
unmelted particle 30%
unmelted particle 12%
unmelted particle 8%
unmelted particle 14%
unmelted particle 70%
unmelted particle 8%
unmelted particle 6%
unmelted particle 7%
unmelted particle 7%
unmelted particle 6%
unmelted particle 10%
unmelted particle 14%
unmelted particle 11%
unmelted particle 6%
lted particle 7%
unmelted particle 13%
unmelted particle 6%
unmelted particle 82%
porosity 68%
unmelted particle 11%
unmelted particle 11%
unmelted particle 8%
unmelted particle 7%
rticle 9%
15kV 7.0mm L-X1.50k BSE M
100μm
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700

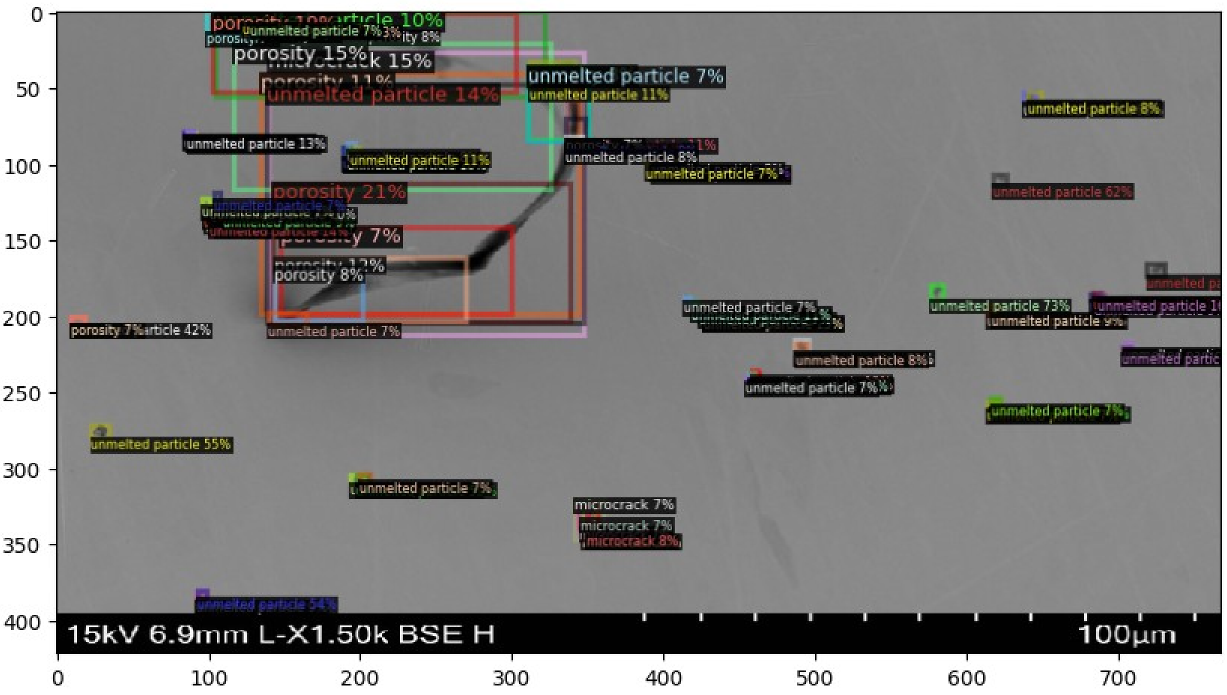

unmelted particle 5%
unmelted particle 12%
unmelted particle 7%
unmelted particle 5%
unmelted particle 9%
unmelted particle 6%
unmelted particle 20%
microcrack 11%
e 97%
unmelted pa
unmelted particle 74%
unmelted particle 74%
unmelted particle 9%
unmelted particle 28%
unmelted particle 8%
unmelted particle 10%
unmelted particle 7%
rticle 8%
15kV 7.0mm L-X1.50k BSE H
100μm
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700



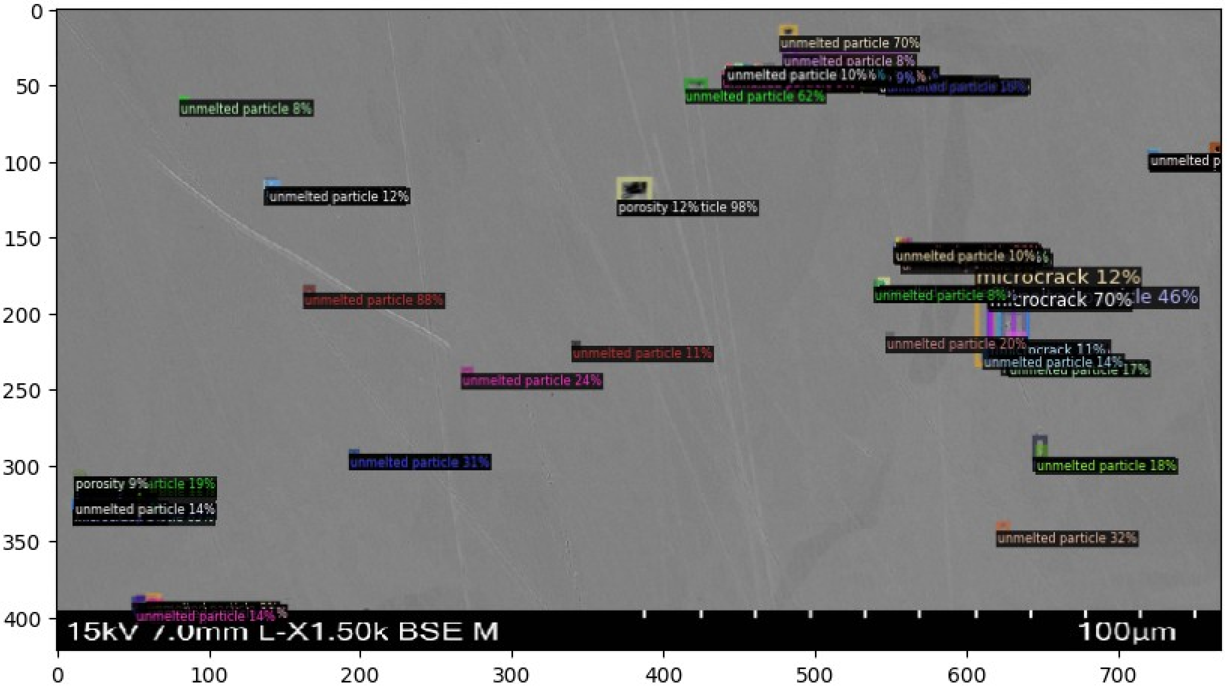


```
from detectron2.utils.visualizer import ColorMode
dataset_dicts = get_r_dicts('/content/drive/MyDrive/Mahabub/test')
for d in random.sample(dataset_dicts, 4):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)
    v = Visualizer(im[:, :, ::-1],
                    metadata=r_metadata,
                    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()
```

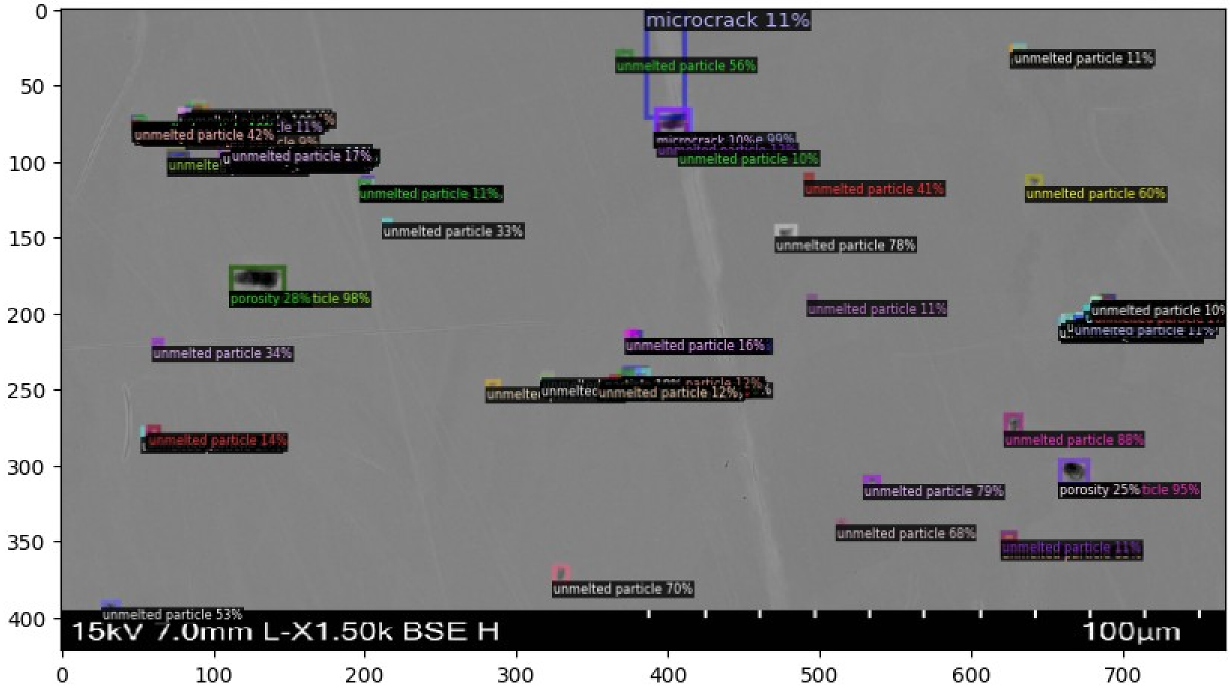




porosity 15%
microcrack 15%
porosity 11%
unmelted particle 14%
unmelted particle 7%
unmelted particle 11%
unmelted particle 8%
unmelted particle 13%
unmelted particle 11%
unmelted particle 8%
unmelted particle 7%
unmelted particle 62%
porosity 21%
porosity 7%
porosity 12%
porosity 8%
unmelted particle 7%
unmelted particle 73%
unmelted particle 9%
porosity 7%
unmelted particle 7%
unmelted particle 8%
unmelted particle 7%
unmelted particle 7%
unmelted particle 55%
unmelted particle 7%
microcrack 7%
microcrack 7%
microcrack 8%
unmelted particle 54%
15kV 6.9mm L-X1.50k BSE H
100µm
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700



unmelted particle 70%
unmelted particle 8%
unmelted particle 10%
unmelted particle 62%
unmelted particle 8%
unmelted particle 12%
porosity 12%
unmelted particle 10%
microcrack 12%
microcrack 70%
unmelted particle 8%
unmelted particle 88%
unmelted particle 11%
unmelted particle 20%
unmelted particle 14%
unmelted particle 17%
unmelted particle 24%
unmelted particle 31%
unmelted particle 18%
porosity 9%
unmelted particle 14%
unmelted particle 32%
unmelted particle 14%
15kV 7.0mm L-X1.50k BSE M
100µm
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700



```
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))
```

```
[07/25 11:32:48 d2.evaluation.coco_evaluation]: Trying to convert
'p_train' to COCO format ...
[07/25 11:32:48 d2.data.datasets.coco]: Converting annotations of
dataset 'p_train' to COCO format ...)
[07/25 11:32:48 d2.data.datasets.coco]: Converting dataset dicts into
COCO format
[07/25 11:32:49 d2.data.datasets.coco]: Conversion finished, #images:
42, #annotations: 715
[07/25 11:32:49 d2.data.datasets.coco]: Caching COCO format
annotations at './output/p_train_coco_format.json' ...
[07/25 11:32:49 d2.data.dataset_mapper]: [DatasetMapper] Augmentations
used in inference: [ResizeShortestEdge(short_edge_length=(800, 800),
max_size=1333, sample_style='choice')]
[07/25 11:32:49 d2.data.common]: Serializing the dataset using: <class
'detectron2.data.common._TorchSerializedList'>
[07/25 11:32:49 d2.data.common]: Serializing 42 elements to byte
tensors and concatenating them all ...
[07/25 11:32:49 d2.data.common]: Serialized dataset takes 0.16 MiB
[07/25 11:32:49 d2.evaluation.evaluator]: Start inference on 42
batches
[07/25 11:32:50 d2.evaluation.evaluator]: Inference done 11/42.
```

```
Dataloading: 0.0020 s/iter. Inference: 0.0564 s/iter. Eval: 0.0005
s/iter. Total: 0.0589 s/iter. ETA=0:00:01
[07/25 11:32:52 d2.evaluation.evaluator]: Total inference time:
0:00:02.265970 (0.061242 s / iter per device, on 1 devices)
[07/25 11:32:52 d2.evaluation.evaluator]: Total inference pure compute
time: 0:00:02 (0.055835 s / iter per device, on 1 devices)
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Preparing results for
COCO format ...
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Saving results to
./output/coco_instances_results.json
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Evaluating predictions
with unofficial COCO API...
Loading and preparing results...
DONE (t=0.01s)
creating index...
index created!
[07/25 11:32:52 d2.evaluation.fast_eval_api]: Evaluate annotation type
*bbox*
[07/25 11:32:52 d2.evaluation.fast_eval_api]: COCOeval_opt.evaluate()
finished in 0.04 seconds.
[07/25 11:32:52 d2.evaluation.fast_eval_api]: Accumulating evaluation
results...
[07/25 11:32:52 d2.evaluation.fast_eval_api]:
COCOeval_opt.accumulate() finished in 0.02 seconds.
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all |
maxDets=100 ] = 0.539
 Average Precision  (AP) @[ IoU=0.50      | area=   all |
maxDets=100 ] = 0.817
 Average Precision  (AP) @[ IoU=0.75      | area=   all |
maxDets=100 ] = 0.554
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small |
maxDets=100 ] = 0.511
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium |
maxDets=100 ] = 0.923
 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 ] = 0.276
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=
10 ] = 0.557
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all |
maxDets=100 ] = 0.634
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small |
maxDets=100 ] = 0.606
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium |
maxDets=100 ] = 0.988
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large |
maxDets=100 ] = -1.000
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Evaluation results for
```

```
bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl  |
|:------:|:------:|:------:|:------:|:------:|:-----:|
| 53.878 | 81.661 | 55.377 | 51.118 | 92.266 |  nan  |
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Some metrics cannot be
computed and is shown as NaN.
[07/25 11:32:52 d2.evaluation.coco_evaluation]: Per-category bbox AP:
| category          | AP     | category   | AP     | category   | AP
|
|:------------------|:-------|:-----------|:-------|:-----------|:----
---|
| unmelted particle | 45.444 | porosity   | 66.404 | microcrack |
49.787 |
OrderedDict([('bbox', {'AP': 53.878198180479174, 'AP50':
81.66097785790757, 'AP75': 55.37691012964614, 'APs':
51.11774899305674, 'APm': 92.26622662266226, 'APl': nan, 'AP-unmelted
particle': 45.44378159428122, 'AP-porosity': 66.40377703606681, 'AP-
microcrack': 49.7870359110895})])
```