

In [1]: !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-gyockyxk
    Running command git clone --filter=blob:none --quiet https://github.com/facebookresearch/detectron2.git /tmp/pip-req-build-gyockyxk
      Resolved https://github.com/facebookresearch/detectron2.git to commit 9604f5995cc628e19f0e4fd913453b4d7d61db3f
        Preparing metadata (setup.py) ... [?251[?25hdone
Requirement already satisfied: Pillow==7.1 in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (11.1.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (3.10.0)
Requirement already satisfied: pycocotools>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (2.0.8)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (2.5.0)
Collecting yacs>=0.1.8 (from detectron2==0.6)
  Downloading yacs-0.1.8-py3-none-any.whl.metadata (639 bytes)
Requirement already satisfied: tabulate in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (0.9.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (3.1.1)
Requirement already satisfied: tqdm>4.29.0 in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (4.67.1)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (2.18.0)
Collecting fvcore<0.1.6,>=0.1.5 (from detectron2==0.6)
  Downloading fvcore-0.1.5.post20221221.tar.gz (50 kB)
[2K   [90m----- [0m [32m50.2/50.2 kB [0m [31m3.3 MB/s [0m eta [36m0:00:00 [0m
[?25h  Preparing metadata (setup.py) ... [?251[?25hdone
Collecting iopath<0.1.10,>=0.1.7 (from detectron2==0.6)
  Downloading iopath-0.1.9-py3-none-any.whl.metadata (370 bytes)
Collecting omegaconf<2.4,>=2.1 (from detectron2==0.6)
  Downloading omegaconf-2.3.0-py3-none-any.whl.metadata (3.9 kB)
Collecting hydra-core>=1.1 (from detectron2==0.6)
  Downloading hydra_core-1.3.2-py3-none-any.whl.metadata (5.5 kB)
Collecting black (from detectron2==0.6)
  Downloading black-25.1.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.metadata (81 kB)
[2K   [90m----- [0m [32m81.3/81.3 kB [0m [31m4.7 MB/s [0m eta [36m0:00:00 [0m
[?25hRequirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from detectron2==0.6) (24.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from fvcore<0.1.6,>=0.1.5->detectron2==0.6) (1.26.4)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from fvcore<0.1.6,>=0.1.5->detectron2==0.6) (6.0.2)
Collecting antlr4-python3-runtime==4.9.* (from hydra-core>=1.1->detectron2==0.6)
  Downloading antlr4-python3-runtime-4.9.3.tar.gz (117 kB)
[2K   [90m----- [0m [32m117.0/117.0 kB [0m [31m10.5 MB/s [0m eta [36m0:00:00 [0m
[?25h  Preparing metadata (setup.py) ... [?251[?25hdone
Collecting portalocker (from iopath<0.1.10,>=0.1.7->detectron2==0.6)
  Downloading portalocker-3.1.1-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (4.55.8)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (1.4.8)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib->detectron2==0.6) (2.8.2)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from black->detectron2==0.6) (8.1.8)
Collecting mypy_extensions>=0.4.3 (from black->detectron2==0.6)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Collecting pathspec>=0.9.0 (from black->detectron2==0.6)
  Downloading pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: platformdirs>=2 in /usr/local/lib/python3.11/dist-packages (from black->detectron2==0.6) (4.3.6)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (1.4.0)
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (1.70.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (3.7)
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (4.25.6)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (75.1.0)
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (1.17.0)
Requirement already satisfied: tensorflow-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow->detectron2==0.6) (3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorflow->detectron2==0.6) (3.1)
Downloading hydra_core-1.3.2-py3-none-any.whl (154 kB)
[2K   [90m----- [0m [32m154.5/154.5 kB [0m [31m9.2 MB/s [0m eta [36m0:00:00 [0m
[?25hDownloaded iopath-0.1.9-py3-none-any.whl (27 kB)
Downloading omegaconf-2.3.0-py3-none-any.whl (79 kB)
[2K   [90m----- [0m [32m79.5/79.5 kB [0m [31m7.1 MB/s [0m eta [36m0:00:00 [0m
[?25hDownloading yacs-0.1.8-py3-none-any.whl (14 kB)
Downloading black-25.1.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl (1.7 MB)
[2K   [90m----- [0m [32m1.7/1.7 MB [0m [31m39.6 MB/s [0m eta [36m0:00:00 [0m
[?25hDownloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading pathspec-0.12.1-py3-none-any.whl (31 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Building wheels for collected packages: detectron2, fvcore, antlr4-python3-runtime
  Building wheel for detectron2 (setup.py) ... [?251[?25hdone
  Created wheel for detectron2: filename=detectron2-0.6-cp311-cp311-linux_x86_64.whl size=6380954 sha256=f94028829bb61f8b185497618884303e3e96fc5f32a
  Stored in directory: /tmp/pip-ephem-wheel-cache-9fpycce7g/wheels/17/d9/40/60db98e485aa9455d653e29d1046601ce96fe23647f60c1c5a
  Building wheel for fvcore (setup.py) ... [?251[?25hdone
  Created wheel for fvcore: filename=fvcore-0.1.5.post20221221-py3-none-any.whl size=61396 sha256=1707c672421936e848e94b42a89b0c87beaff97ab70d3a91f2
  Stored in directory: /root/.cache/pip/wheels/65/71/95/3b8fde5c65c6e4a806e0867c1651dcc71a1cb2f3430e8f355f
  Building wheel for antlr4-python3-runtime (setup.py) ... [?251[?25hdone
  Created wheel for antlr4-python3-runtime: filename=antlr4_python3_runtime-4.9.3-py3-none-any.whl size=144555 sha256=4d33e32f16c70803f6be4e2e7e25c
  Stored in directory: /root/.cache/pip/wheels/1a/97/32/461f837398029ad76911109f07047fd1d7b661a147c7c56d
Successfully built detectron2 fvcore antlr4-python3-runtime
Installing collected packages: antlr4-python3-runtime, yacs, portalocker, pathspec, omegaconf, mypy_extensions, iopath, hydra-core, black, fvcore, fvcore-0.1.5.post20221221 hydra-core-1.3.2 iopath-0.1.9 mypy_extensions
Successfully installed antlr4-python3-runtime-4.9.3 black-25.1.0 detectron2-0.6 fvcore-0.1.5.post20221221 hydra-core-1.3.2 iopath-0.1.9 mypy_extensions
```

In [2]:

```
import torch
if torch.cuda.is_available():
    GPU = True
    print('gpu available')
else:
    GPU = False
    print('no gpu')
```

gpu available

In [3]:

```
import torch, torchvision
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()
```

```

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

import pandas as pd
import numpy as np
import os
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm
import cv2
import random
from ast import literal_eval

```

In [4]: !pip install -U aicrowd-cli

```

Collecting aicrowd-cli
  Downloading aicrowd_cli-0.1.16-py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (8.1.8)
Requirement already satisfied: GitPython in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (3.1.44)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (2.32.3)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (1.0.0)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (13.9.4)
Requirement already satisfied: toml in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (0.10.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (4.67.1)
Requirement already satisfied: pyzmq in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (24.0.1)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (from aicrowd-cli) (8.0.4)
Collecting semver (from aicrowd-cli)
  Downloading semver-3.0.4-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from GitPython->aicrowd-cli) (4.0.12)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.11/dist-packages (from python-slugify->aicrowd-cli) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->aicrowd-cli) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->aicrowd-cli) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->aicrowd-cli) (2.3.0)
Requirement already satisfied: certifi=>2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->aicrowd-cli) (2025.1.31)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->aicrowd-cli) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->aicrowd-cli) (2.18.0)
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.11/dist-packages (from gitdb<5,>=4.0.1->GitPython->aicrowd-cli) (5.0.2)
Requirement already satisfied: mdurl=>0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->aicrowd-cli) (0.1.2)
Downloading aicrowd_cli-0.1.16-py3-none-any.whl (50 kB)
[2K  [90m-----] 0m [32m51.0/51.0 kB [0m [31m4.9 MB/s [0m eta [36m0:00:00 [0m
?25hDownloading semver-3.0.4-py3-none-any.whl (17 kB)
Installing collected packages: semver, aicrowd-cli
Successfully installed aicrowd-cli-0.1.16 semver-3.0.4

```

In [5]: API\_KEY = "c9eff61ed0464a10bb065b62a4f29ef7"  
!aicrowd login --api-key \$API\_KEY

```

[32mA API Key valid [0m
[33mGitlab oauth token invalid or absent.
It is highly recommended to simply run `aicrowd login` without passing the API Key. [0m
[32mSaved details successfully! [0m

```

In [6]: !aicrowd dataset download --challenge debris-detection

```

sample_submission.csv: 100% 1.98M/1.98M [00:01<00:00, 1.53MB/s]
test.zip: 100% 259M/259M [00:25<00:00, 10.1MB/s]
train.csv: 100% 1.78M/1.78M [00:01<00:00, 1.52MB/s]
train.zip: 100% 1.04G/1.04G [01:39<00:00, 10.4MB/s]
val.csv: 100% 176k/176k [00:00<00:00, 270kB/s]
val.zip: 100% 104M/104M [00:16<00:00, 6.25MB/s]

```

In [7]: !rm -rf data  
!mkdir data  
  
!unzip train.zip -d data/train > /dev/null  
  
!unzip val.zip -d data/val > /dev/null  
  
!unzip test.zip -d data/test > /dev/null  
  
!mv train.csv data/train.csv  
!mv val.csv data/val.csv

In [8]: data\_path = "data"  
  
train\_df = pd.read\_csv(os.path.join(data\_path, "train.csv"))  
val\_df = pd.read\_csv(os.path.join(data\_path, "val.csv"))

In [9]: train\_df

Out [9]: 

ImageID	bboxes
0 0	[[282, 308, 58, 86], [228, 256, 191, 218], [10...

ImageID	bboxes
1 1	[[97, 127, 297, 332], [315, 352, 136, 168], [7...
2 2	[[45, 76, 244, 269], [154, 182, 457, 484], [20...
3 3	[[252, 282, 86, 110], [315, 333, 379, 412], [2...
4 4	[[175, 203, 406, 440], [57, 86, 433, 464], [26...
... ...	...
19995 19995	[[226, 247, 396, 424], [301, 323, 454, 479], [...
19996 19996	[[228, 256, 149, 177], [42, 71, 294, 331], [28...
19997 19997	[[443, 474, 153, 181], [138, 166, 193, 219], [...
19998 19998	[[34, 70, 178, 217], [400, 443, 127, 161], [43...
19999 19999	[[184, 207, 419, 448], [388, 410, 427, 457], [...

20000 rows × 2 columns

```
In [10]: def show_images(images, num = 5):

    images_to_show = np.random.choice(images, num)

    for image_id in images_to_show:

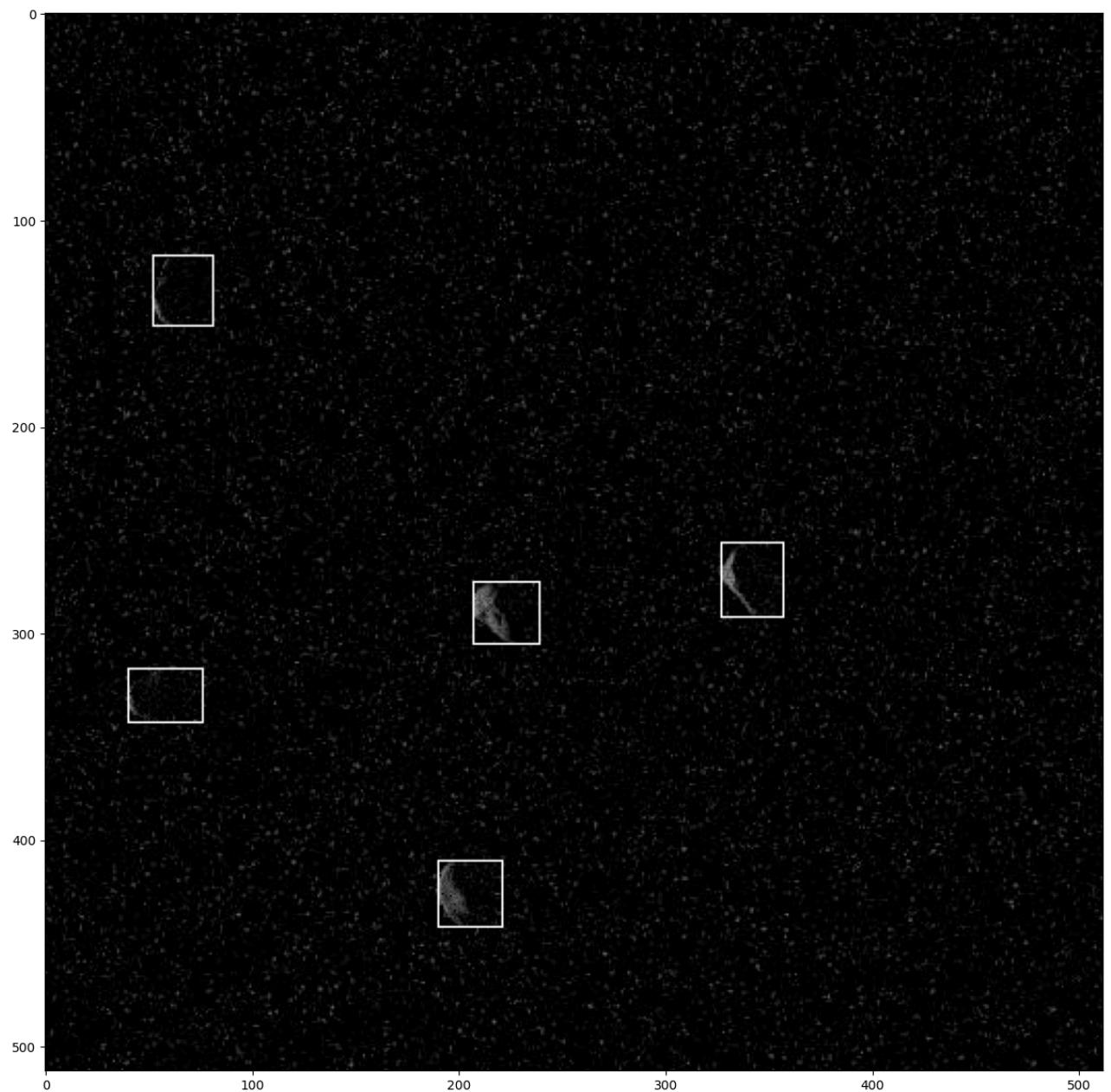
        image = Image.open(f"data/train/{image_id}.jpg")

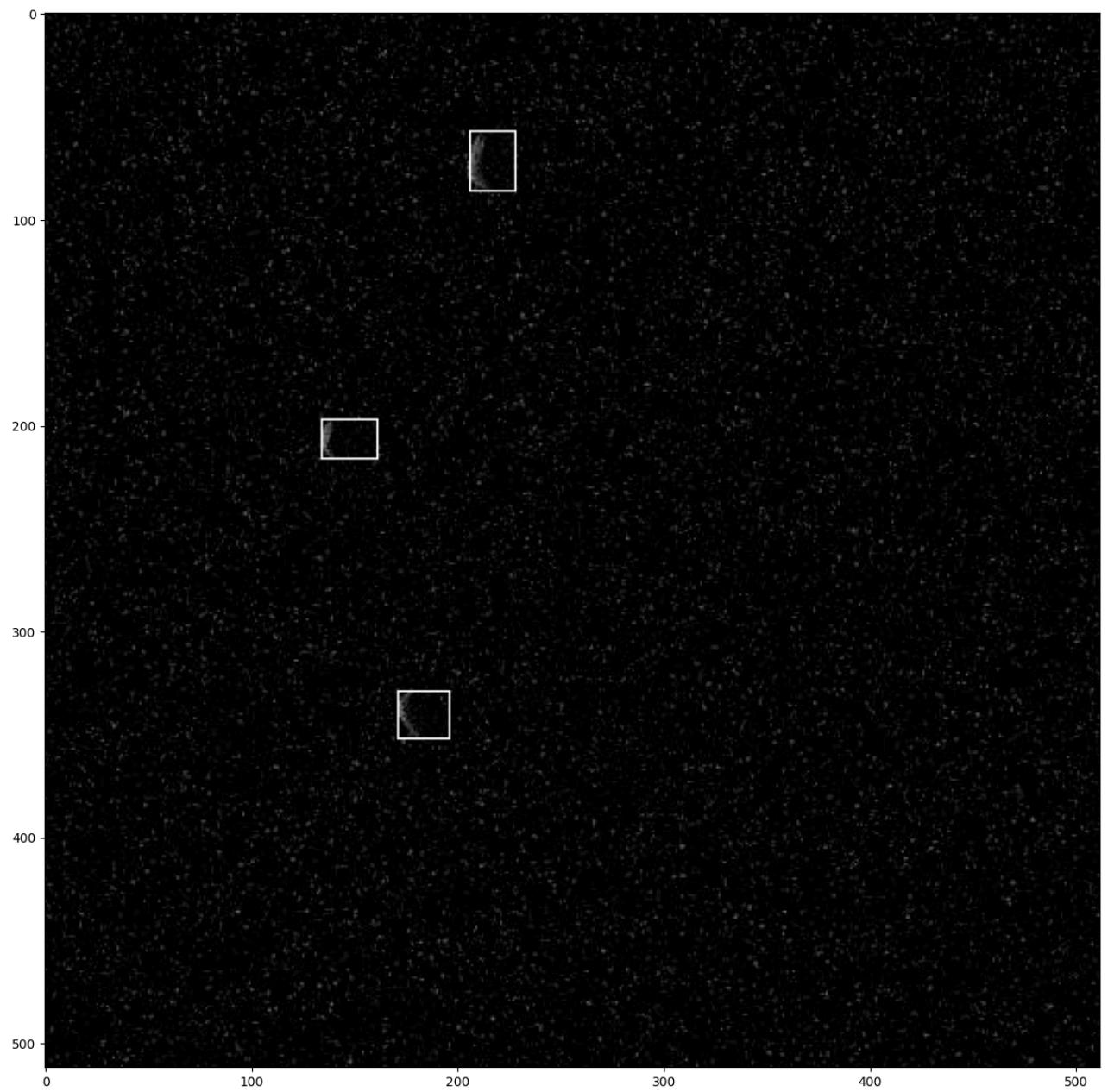
        bboxes = literal_eval(train_df.loc[train_df['ImageID'] == image_id]['bboxes'].values[0])

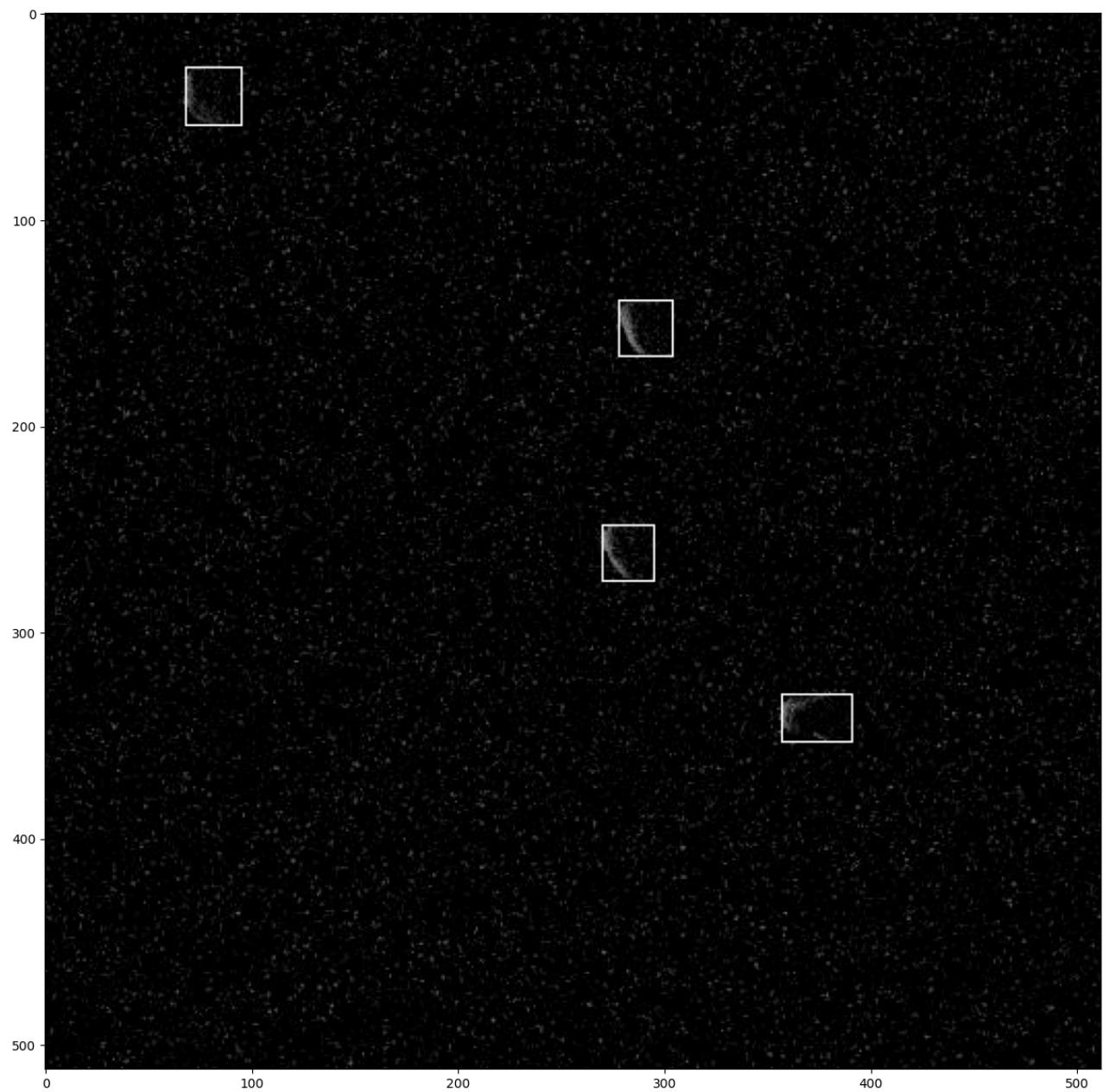
        draw = ImageDraw.Draw(image)
        for bbox in bboxes:
            draw.rectangle([bbox[0], bbox[2], bbox[1], bbox[3]], width=1)

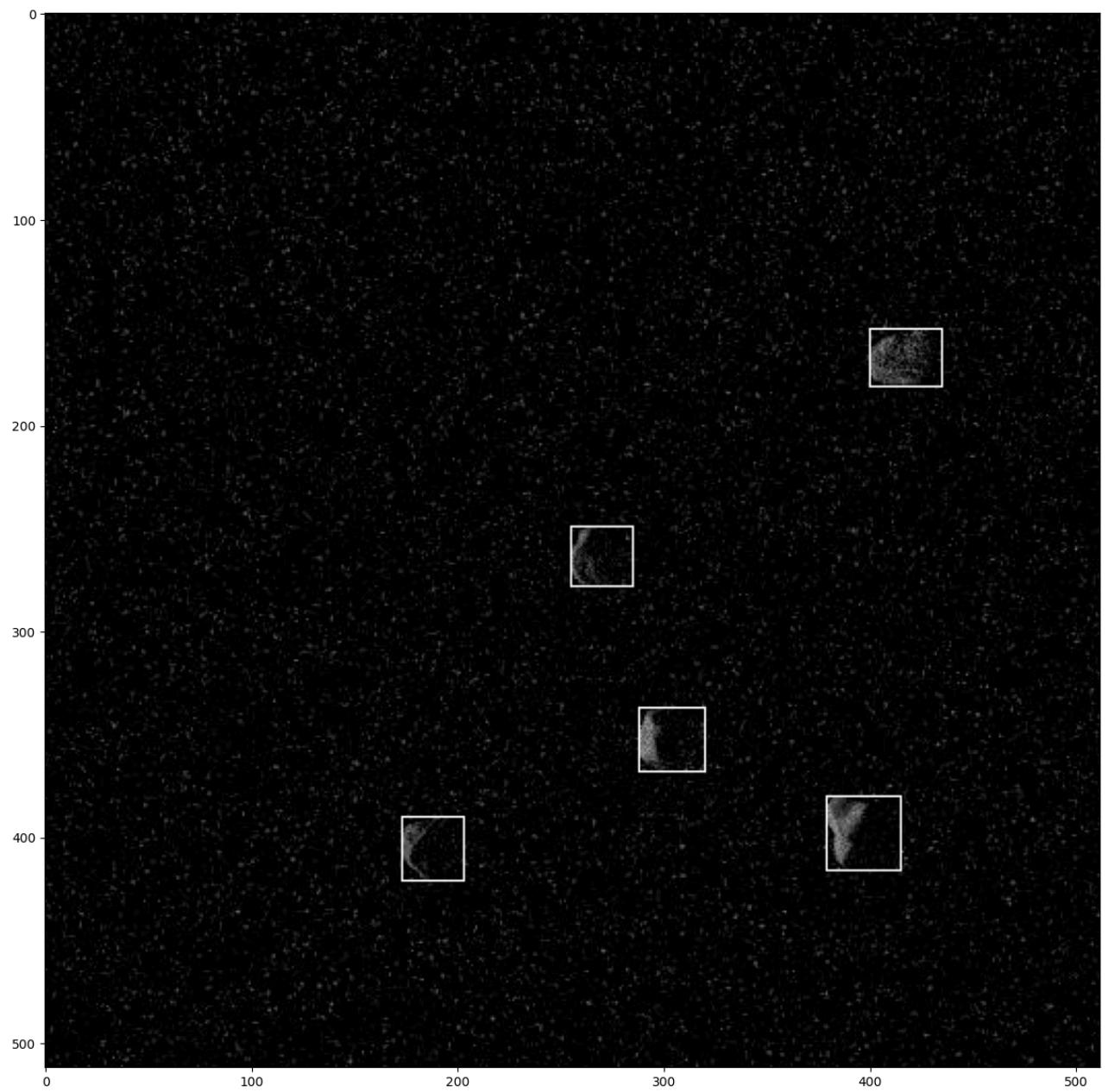
        plt.figure(figsize = (15,15))
        plt.imshow(image)
        plt.show()

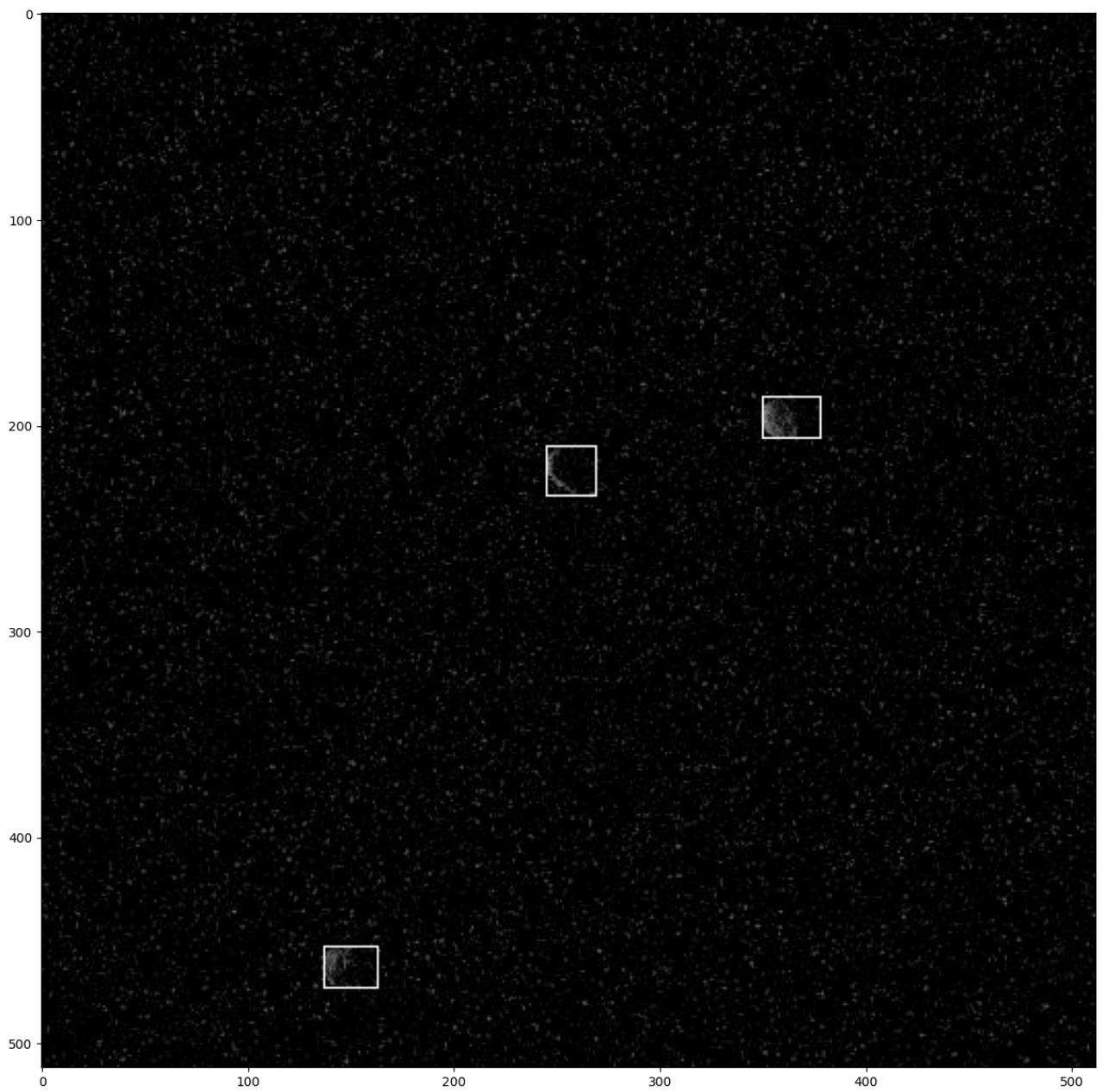
show_images(train_df['ImageID'].unique(), num = 5)
```











```
In [11]: dict_dataset = []
def get_dataset_dicts():

    for index, row in train_df.iterrows():

        image = Image.open(os.path.join(data_path, f"train/{row['ImageID']}.jpg"))
        w, h = image.size

        ann_lst = []

        bboxes = literal_eval(row['bboxes'])

        for n, bbox in enumerate(bboxes):

            ann_dict = {'bbox': [bbox[0], bbox[2], bbox[1], bbox[3]],
                       'bbox_mode': BoxMode.XYXY_ABS,
                       'category_id': 0, #i[1]['category_id'].values[0],
                       'iscrowd': 0}

            ann_lst.append(ann_dict)

        image_dict = {'annotations': ann_lst,
                      'file_name': os.path.join(data_path, f"train/{row['ImageID']}.jpg"),
                      'height': h,
                      'image_id': row["ImageID"], #i[1]['image_category_id'].values[0],
                      'width': w}

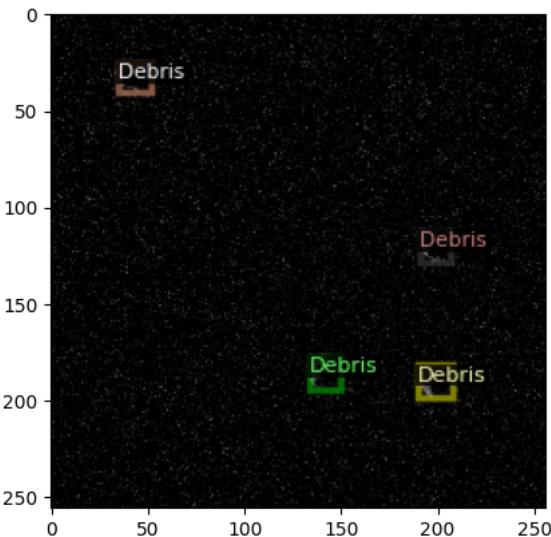
        dict_dataset.append(image_dict)

    return dict_dataset

dict_dataset = get_dataset_dicts()
```

```
In [12]: d = f"debris_train{np.random.randint(10000)}"
DatasetCatalog.register(d, lambda d=d : get_dataset_dicts())
MetadataCatalog.get(d).set(thing_classes=["Debris"])
obj_metadata = MetadataCatalog.get(d)
```

```
In [13]: for i in random.sample(dict_dataset, 3):
    img = cv2.imread(i["file_name"])
    visualizer = Visualizer(img, metadata=obj_metadata, scale=0.5)
    out = visualizer.draw_dataset_dict(i)
    plt.imshow(out.get_image())
```



```
In [14]: cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_50_DC5_3x.yaml"))
cfg.DATASETS.TRAIN = (d,)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_50_DC5_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 200
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
```

```
In [15]: trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

```
[02/08 17:42:56 d2.engine.defaults]: Model:
GeneralizedRCNN(
  (backbone): 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=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)
        )
        (conv1): Conv2d(
            1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv2): Conv2d(
            512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(2, 2), dilation=(2, 2), 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=(2, 2), dilation=(2, 2), 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=(2, 2), dilation=(2, 2), 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(
            2048, 2048, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)
            (activation): ReLU()
        )
        (objectness_logits): Conv2d(2048, 15, kernel_size=(1, 1), stride=(1, 1))
        (anchor_deltas): Conv2d(2048, 60, kernel_size=(1, 1), stride=(1, 1))
    )
    (anchor_generator): DefaultAnchorGenerator(
        (cell_anchors): BufferList()
    )
)
)
(roi_heads): StandardROIHeads(
    (box_pooler): ROIAlign(
        (level_poolers): ModuleList(
            (0): ROIAlign(output_size=(7, 7), spatial_scale=0.0625, sampling_ratio=0, aligned=True)
        )
    )
    (box_head): FastRCNNConvFCHead(
        (flatten): Flatten(start_dim=1, end_dim=-1)
        (fc1): Linear(in_features=100352, 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=2, bias=True)
  (bbox_pred): Linear(in_features=1024, out_features=4, bias=True)
)
)
)
[02/08 17:43:02 d2.data.build]: Removed 0 images with no usable annotations. 40000 images left.
[02/08 17:43:03 d2.data.build]: Distribution of instances among all 1 categories:
| category | #instances |
|:-----:|:-----:|
| Debris | 150544 |
|
[02/08 17:43:03 d2.data.dataset_mapper]: [DatasetMapper] Augmentations used in training: [ResizeShortestEdge(short_edge_length=(640, 672, 704, 736,
[02/08 17:43:03 d2.data.build]: Using training sampler TrainingSampler
[02/08 17:43:03 d2.data.common]: Serializing the dataset using: <class 'detectron2.data.common._TorchSerializedList'>
[02/08 17:43:03 d2.data.common]: Serializing 40000 elements to byte tensors and concatenating them all ...
[02/08 17:43:03 d2.data.common]: Serialized dataset takes 11.71 MiB
[02/08 17:43:03 d2.data.build]: Making batched data loader with batch_size=2
WARNING [02/08 17:43:03 d2.solver.build]: SOLVER.STEPS contains values larger than SOLVER.MAX_ITER. These values will be ignored.
[02/08 17:43:03 d2.checkpoint.detection_checkpoint]: [DetectionCheckpointer] Loading from https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/f...
```

```
model_final_68d202.pkl: 663MB [00:29, 22.7MB/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 (2, 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 (2,) 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 (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.bbox_pred.bias' to the model due to incompatible
shapes: (320,) in the checkpoint but (4,) 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}
```

```
[02/08 17:43:33 d2.engine.train_loop]: Starting training from iteration 0
```

```
/usr/local/lib/python3.11/dist-packages/torch/functional.py:534: UserWarning: torch.meshgrid: in an upcoming release, it will be
required to pass the indexing argument. (Triggered internally at ../aten/src/ATen/native/TensorShape.cpp:3595.)
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
```

```
[02/08 17:43:46 d2.utils.events]: eta: 0:01:17 iter: 19 total_loss: 1.323 loss_cls: 0.6995 loss_box_reg: 0.5783 loss_rpn_cls: 0.0372 loss_rpn_box_reg: 0.0001
[02/08 17:44:00 d2.utils.events]: eta: 0:01:08 iter: 39 total_loss: 1.02 loss_cls: 0.4191 loss_box_reg: 0.5612 loss_rpn_cls: 0.03149 loss_rpn_box_reg: 0.0001
[02/08 17:44:09 d2.utils.events]: eta: 0:01:00 iter: 59 total_loss: 0.9249 loss_cls: 0.2664 loss_box_reg: 0.6021 loss_rpn_cls: 0.01916 loss_rpn_box_reg: 0.0001
[02/08 17:44:18 d2.utils.events]: eta: 0:00:52 iter: 79 total_loss: 0.8618 loss_cls: 0.2085 loss_box_reg: 0.6513 loss_rpn_cls: 0.01333 loss_rpn_box_reg: 0.0001
[02/08 17:44:27 d2.utils.events]: eta: 0:00:43 iter: 99 total_loss: 0.8208 loss_cls: 0.1412 loss_box_reg: 0.6476 loss_rpn_cls: 0.00563 loss_rpn_box_reg: 0.0001
[02/08 17:44:36 d2.utils.events]: eta: 0:00:34 iter: 119 total_loss: 0.8411 loss_cls: 0.1236 loss_box_reg: 0.7024 loss_rpn_cls: 0.007235 loss_rpn_box_reg: 0.0001
[02/08 17:44:45 d2.utils.events]: eta: 0:00:26 iter: 139 total_loss: 0.798 loss_cls: 0.0946 loss_box_reg: 0.6841 loss_rpn_cls: 0.001694 loss_rpn_box_reg: 0.0001
[02/08 17:44:54 d2.utils.events]: eta: 0:00:17 iter: 159 total_loss: 0.72 loss_cls: 0.08165 loss_box_reg: 0.6314 loss_rpn_cls: 0.002156 loss_rpn_box_reg: 0.0001
[02/08 17:45:03 d2.utils.events]: eta: 0:00:08 iter: 179 total_loss: 0.6003 loss_cls: 0.07363 loss_box_reg: 0.5188 loss_rpn_cls: 0.00271 loss_rpn_box_reg: 0.0001
[02/08 17:45:17 d2.utils.events]: eta: 0:00:00 iter: 199 total_loss: 0.6002 loss_cls: 0.06004 loss_box_reg: 0.5199 loss_rpn_cls: 0.0003947 loss_rpn_box_reg: 0.0001
[02/08 17:45:18 d2.engine.hooks]: Overall training speed: 198 iterations in 0:01:28 (0.4494 s / it)
[02/08 17:45:18 d2.engine.hooks]: Total training time: 0:01:39 (0:00:10 on hooks)
```

In [16]: %load\_ext tensorboard

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

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

In [17]:

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

```
[02/08 17:50:55 d2.checkpoint.detection_checkpoint]: [DetectionCheckpointer] Loading from ./output/model_final.pth ...
```

```
/usr/local/lib/python3.11/dist-packages/fvcore/common/checkpoint.py:252: FutureWarning: You are using `torch.load` with
`weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct
malicious pickle data which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full
control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
    return torch.load(f, map_location=torch.device("cpu"))
```

In [18]:

```
test_imgs_paths = os.listdir(os.path.join(data_path, "test"))

predictions = {"ImageID":[], "bboxes":[]}

for test_img_path in tqdm(test_imgs_paths):

    img = cv2.imread(os.path.join(data_path, "test", test_img_path))
    h, w, _ = img.shape

    model_predictions = predictor(img)

    bboxes = model_predictions['instances'].pred_boxes.tensor.cpu().numpy().tolist()
    scores = model_predictions['instances'].scores.cpu().numpy().tolist()

    for n, bbox in enumerate(bboxes):

        bboxes[n] == bbox.append(scores[n])

    image_id = test_img_path.split('.')[0]

    predictions['ImageID'].append(image_id)
    predictions['bboxes'].append(bboxes)
```

```
0% | 0/5000 [00:00<?, ?it/s]
```

```
In [19]: submission = pd.DataFrame(predictions)
submission
```

Out [19]:

	ImageID	bboxes
0	2535	[[360.4900207519531, 421.6808776855469, 386.99...]
1	4336	[[28.393911361694336, 408.4031677246094, 55.63...]
2	4821	[[409.55499267578125, 198.4417724609375, 436.6...]
3	3623	[[177.0428466796875, 55.4532470703125, 204.588...]
4	685	[[74.48613739013672, 53.49018859863281, 102.86...]
...	...	...
4995	2031	[[350.2088317871094, 340.33819580078125, 379.0...]
4996	2132	[[149.4950714111328, 234.63926696777344, 174.7...]
4997	871	[[373.0574035644531, 211.2058563232422, 401.59...]
4998	2228	[[350.3152160644531, 369.7606201171875, 377.53...]
4999	1746	[[238.6121063232422, 264.8551330566406, 265.14...]

5000 rows × 2 columns

```
In [20]: submission.to_csv("submission.csv", index=False)
```

```
In [21]: import os
import cv2
import matplotlib.pyplot as plt

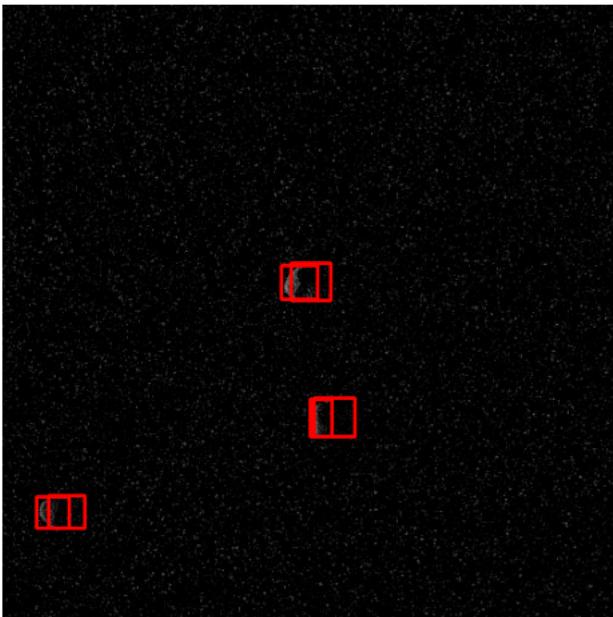
# Select an image to visualize
image_id = submission["ImageID"][1] # Get first image ID
bboxes = submission["bboxes"][1] # Get corresponding bounding boxes

# Load the image
img_path = os.path.join(data_path, "test", f"{image_id}.jpg")
img = cv2.imread(img_path)

if img is None:
    print(f"Error: Image {img_path} not found!")
else:
    # Convert from BGR to RGB (Matplotlib displays in RGB)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Draw bounding boxes
    for bbox in bboxes:
        x_min, y_min, x_max, y_max, score = bbox # Ensure correct order
        color = (255, 0, 0) # Red color for boxes
        thickness = 2
        img = cv2.rectangle(img, (int(x_min), int(y_min)), (int(x_max), int(y_max)), color, thickness)

    # Display image
    plt.figure(figsize=(8, 6))
    plt.imshow(img)
    plt.axis("off") # Hide axis
    plt.title(f"Detected Space Debris - {image_id}")
    plt.show()
```



```
In [24]: import cv2
import torch
import numpy as np
import IPython.display
from google.colab.patches import cv2_imshow
from google.colab import output
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2 import model_zoo
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog
import time
import base64
import numpy as np

# 1 Load Trained Faster R-CNN Model
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_50_DC5_3x.yaml"))
cfg.MODEL.WEIGHTS = "output/model_final.pth" # Ensure this path is correct
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5 # Confidence threshold
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1 # Only one class (Debris)
predictor = DefaultPredictor(cfg)

# 2 Function to Capture Image from Google Colab Camera
def take_photo(filename='photo.jpg'):
    js = '''
    async function takePhoto() {
        const div = document.createElement('div');
        const video = document.createElement('video');
        const btn = document.createElement('button');
        btn.textContent = 'Take Photo';
        div.appendChild(video);
        div.appendChild(btn);
        document.body.appendChild(div);

        const stream = await navigator.mediaDevices.getUserMedia({video: true});
        video.srcObject = stream;
        await video.play();

        await new Promise((resolve) => btn.onclick = resolve);
        const canvas = document.createElement('canvas');
        canvas.width = video.videoWidth;
        canvas.height = video.videoHeight;
        canvas.getContext('2d').drawImage(video, 0, 0);
        stream.getTracks().forEach(track => track.stop());

        document.body.removeChild(div);
        return canvas.toDataURL('image/jpeg');
    }
    takePhoto();
    ...
    display(IPython.display.Javascript(js))
    data = output.eval_js('takePhoto()')

```

```

binary = np.frombuffer(base64.b64decode(data.split(',')[1]), dtype=np.uint8)
img = cv2.imdecode(binary, cv2.IMREAD_COLOR)
cv2.imwrite(filename, img)
return filename

# ③ Capture an Image from the Camera
image_path = take_photo()

# ④ Read the Image and Run Object Detection
frame = cv2.imread(image_path)
image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
outputs = predictor(image)

# ⑤ Draw Bounding Boxes
metadata = MetadataCatalog.get("debris_test") # Adjust dataset name if needed
v = Visualizer(frame[:, :, ::-1], metadata=metadata, scale=0.8)
v = v.draw_instance_predictions(outputs["instances"].to("cpu"))

# ⑥ Display the Detected Image
cv2.imshow(v.get_image()[:, :, ::-1])

```

[02/08 18:33:35 d2.checkpoint.detection\_checkpoint]: [DetectionCheckpointer] Loading from output/model\_final.pth ...

```

/usr/local/lib/python3.11/dist-packages/fvcore/common/checkpoint.py:252: FutureWarning: You are using `torch.load` with
`weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct
malicious pickle data which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full
control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
    return torch.load(f, map_location=torch.device("cpu"))

```

<IPython.core.display.Javascript object>



In [ ]: