

!nvidia-smi

Thu Apr 27 03:12:07 2023

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
NVIDIA-SMI		525.85.12		Driver Version: 525.85.12			CUDA Version: 12.0		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
GPU	Name		Persistence-M		Bus-Id	Disp.A	Volatile Uncorr. ECC		
Fan	Temp	Perf	Pwr:Usage/Cap		Memory-Usage		GPU-Util	Compute M.	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
0	Tesla	T4	Off		00000000:00:04:0	Off		0	
N/A	48C	P8	10W / 70W		0MiB / 15360MiB		0%	Default	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Processes:									
GPU	GI	CI	PID	Type	Process name			GPU Memory	
	ID	ID						Usage	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
No running processes found									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									

```
import os
os.environ['CUDA_LAUNCH_BLOCKING'] = "1"
```

!pwd

/content

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

Mounted at /content/drive

!pip install pyyaml==5.1

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyyaml==5.1
  Downloading PyYAML-5.1.tar.gz (274 kB)
    274.2/274.2 kB 13.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pyyaml
  Building wheel for pyyaml (setup.py) ... done
  Created wheel for pyyaml: filename=PyYAML-5.1-cp39-cp39-linux_x86_64.whl size=44089 sha256=54465fb37963dc596388e7c769df06d3b8c9d6e9bfc
  Stored in directory: /root/.cache/pip/wheels/68/be/8f/b6c454cd264e0b349b47f8ee00755511f277618af9e5dae20d
Successfully built pyyaml
Installing collected packages: pyyaml
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 6.0
    Uninstalling PyYAML-6.0:
      Successfully uninstalled PyYAML-6.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
flax 0.6.9 requires PyYAML>=5.4.1, but you have pyyaml 5.1 which is incompatible.
dask 2022.12.1 requires pyyaml>=5.3.1, but you have pyyaml 5.1 which is incompatible.
Successfully installed pyyaml-5.1
```

!pip install torch==1.9.0+cu102 torchvision==0.10.0+cu102 -f https://download.pytorch.org/whl/torch_stable.html

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Looking in links: https://download.pytorch.org/whl/torch\_stable.html
Collecting torch==1.9.0+cu102
  Downloading https://download.pytorch.org/whl/cu102/torch-1.9.0%2Bcu102-cp39-cp39-linux\_x86\_64.whl (831.4 MB)
    831.4/831.4 MB 1.6 MB/s eta 0:00:00
Collecting torchvision==0.10.0+cu102
  Downloading https://download.pytorch.org/whl/cu102/torchvision-0.10.0%2Bcu102-cp39-cp39-linux\_x86\_64.whl (22.0 MB)
    22.0/22.0 MB 73.8 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.9/dist-packages (from torch==1.9.0+cu102) (4.5.0)
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python3.9/dist-packages (from torchvision==0.10.0+cu102) (8.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from torchvision==0.10.0+cu102) (1.22.4)
Installing collected packages: torch, torchvision
  Attempting uninstall: torch
    Found existing installation: torch 2.0.0+cu118
    Uninstalling torch-2.0.0+cu118:
      Successfully uninstalled torch-2.0.0+cu118
```

```
Attempting uninstall: torchvision
Found existing installation: torchvision 0.15.1+cu118
Uninstalling torchvision-0.15.1+cu118:
Successfully uninstalled torchvision-0.15.1+cu118
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
torchtext 0.15.1 requires torch==2.0.0, but you have torch 1.9.0+cu102 which is incompatible.
torchdata 0.6.0 requires torch==2.0.0, but you have torch 1.9.0+cu102 which is incompatible.
torchaudio 2.0.1+cu118 requires torch==2.0.0, but you have torch 1.9.0+cu102 which is incompatible.
Successfully installed torch-1.9.0+cu102 torchvision-0.10.0+cu102
```

```
!pip install detectron2 -f https://dl.fbaipublicfiles.com/detectron2/wheels/cu102/torch1.9/index.html
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Looking in links: https://dl.fbaipublicfiles.com/detectron2/wheels/cu102/torch1.9/index.html
Collecting detectron2
  Downloading https://dl.fbaipublicfiles.com/detectron2/wheels/cu102/torch1.9/detectron2-0.6%2Bcu102
    6.3/6.3 MB 29.6 MB/s eta 0:00:00
Requirement already satisfied: Pillow>=7.1 in /usr/local/lib/python3.9/dist-packages (from detectron2)
Collecting omegaconf>=2.1
  Downloading omegaconf-2.3.0-py3-none-any.whl (79 kB)
    79.5/79.5 kB 7.9 MB/s eta 0:00:00
Requirement already satisfied: pycocotools>=2.0.2 in /usr/local/lib/python3.9/dist-packages (from detectron2)
Requirement already satisfied: future in /usr/local/lib/python3.9/dist-packages (from detectron2)
Collecting iopath<0.1.10,>=0.1.7
  Downloading iopath-0.1.9-py3-none-any.whl (27 kB)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.9/dist-packages (from detectron2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from detectron2)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.9/dist-packages (from detectron2)
Collecting fvc<0.1.6,>=0.1.5
  Downloading fvc-0.1.5.post20221221.tar.gz (50 kB)
    50.2/50.2 kB 7.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting yacs>=0.1.8
  Downloading yacs-0.1.8-py3-none-any.whl (14 kB)
Collecting hydra-core>=1.1
  Downloading hydra-core-1.3.2-py3-none-any.whl (154 kB)

import torch, torchvision
print(torch.__version__, torch.cuda.is_available())
assert torch.__version__.startswith("1.9") # please mrch 1.9 if Colab changes its default version

1.9.0+cu102 True

  Downloading black-21.4b2-py3-none-any.whl (130 kB)

# Some basic setup:
# Setup detectron2 logger
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# import some common libraries
import numpy as np
import os, json, cv2, random
from google.colab.patches import cv2_imshow

# 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

Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from detectron2)
!unzip '/content/trail_images.zip' -d '/content'

Archive: /content/trail_images.zip
  inflating: /content/trail_images/1.jpg
  inflating: /content/trail_images/2.jpg
  inflating: /content/trail_images/3.jpg
  inflating: /content/trail_images/4.jpg
  inflating: /content/trail_images/5.jpg
  inflating: /content/trail_images/6.jpg
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.9/dist-packages (from detectron2)
from detectron2.data.datasets import register_coco_instances
register_coco_instances("sample2", {}, "/content/trail.json", "/content/trail_images")
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.9/dist-packages (from detectron2)
sample_metadata = MetadataCatalog.get("sample2")
dataset_dicts = DatasetCatalog.get("sample2")

WARNING [04/27 03:14:48 d2.data.datasets.coco]:
Category ids in annotations are not in [1, #categories]! We'll apply a mapping for you.

[04/27 03:14:48 d2.data.datasets.coco]: Loaded 6 images in COCO format from /content/trail.json

import random

for d in random.sample(dataset_dicts, 4):
    img = cv2.imread(d['file_name'])
    visualizer = Visualizer(img[:, :, :-1], metadata=sample_metadata, scale=0.5)
    vis = visualizer.draw_dataset_dict(d)
    cv2_imshow(vis.get_image()[:, :, :-1])

```



```
!git clone https://github.com/facebookresearch/detectron2 detectron2_repo
```

```
Cloning into 'detectron2_repo'...
remote: Enumerating objects: 15007, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 15007 (delta 10), reused 21 (delta 4), pack-reused 14975
Receiving objects: 100% (15007/15007), 6.10 MiB | 20.07 MiB/s, done.
Resolving deltas: 100% (10875/10875), done.
```

```
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
import os
```

```
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_50_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("sample2",)
cfg.DATASETS.TEST = () # no metrics implemented for this dataset
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_50_FPN_3x.yaml") # initialize from model zoo
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.02
cfg.SOLVER.MAX_ITER = 300 # 300 iterations seems good enough, but you can certainly train longer
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 640 # faster, and good enough for this toy dataset
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 4
```

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

```
[04/27 03:15:08 d2.engine.defaults]: Model:
GeneralizedRCNN(
  (backbone): FPN(
    (fpn_lateral2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral3): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral4): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral5): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (top_block): LastLevelMaxPool()
    (bottom_up): ResNet(
      (stem): BasicStem(
        (conv1): Conv2d(
          3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False
          (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
        )
      )
    )
    (res2): Sequential(
      (0): BottleneckBlock(
        (shortcut): Conv2d(
          64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
          (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv1): Conv2d(
          64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
          (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
        )
        (conv2): Conv2d(
          64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
          (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
        )
      )
      (conv3): Conv2d(
```

```

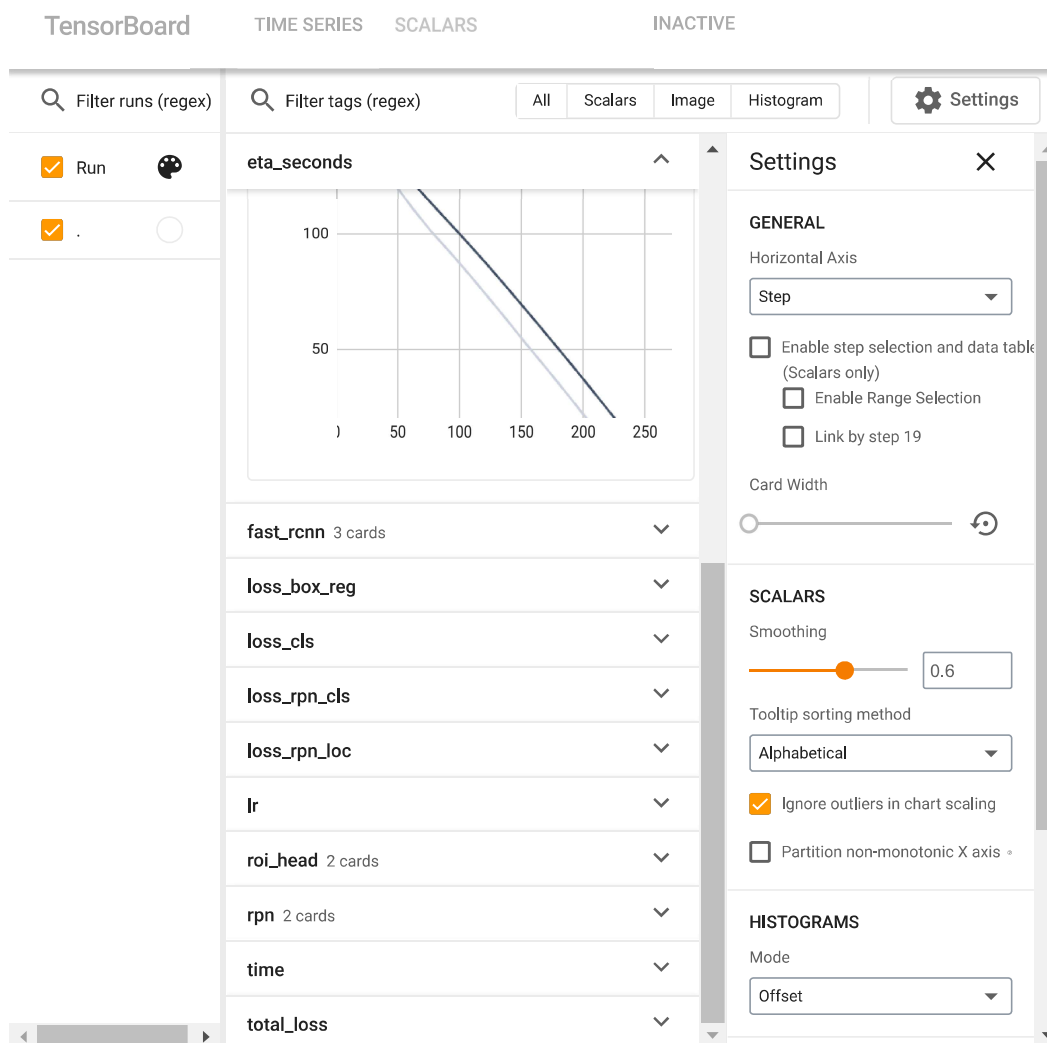
        64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
)
(1): BottleneckBlock(
  (conv1): Conv2d(
    256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
  )
  (conv2): Conv2d(
    64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
  )
  (conv3): Conv2d(
    64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
  )
)
)
(2): BottleneckBlock(
  (conv1): Conv2d(
    256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
  )
)
)

```

```

# Look at training curves in tensorboard:
%load_ext tensorboard
%tensorboard --logdir output

```



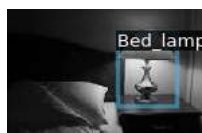
```

cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5 # set the testing threshold for this model
cfg.DATASETS.TEST = ("sample2",)
predictor = DefaultPredictor(cfg)

```

```
from detectron2.utils.visualizer import ColorMode

for d in random.sample(dataset_dicts, 4):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)
    v = Visualizer(im[:, :, ::-1],
                  metadata=sample_metadata,
                  scale=0.8,
                  instance_mode=ColorMode.IMAGE_BW # remove the colors of unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    cv2.imshow(v.get_image()[:, :, ::-1])
```



```
# Download Config File
f= open("config.yaml", "w")
f.write(cfg.dump())
f.close()
```