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

```
cd /content/drive/MyDrive/diagram_correction/Detectron #change the path to your drive
```

```
!pip install -U torch==1.9.0 torchvision
!pip install git+https://github.com/facebookresearch/fvcore.git
import torch, torchvision
torch.__version__
```

```
#install detecron2
!git clone https://github.com/facebookresearch/detectron2 detectron2_repo
!pip install -e detectron2_repo
```

```
############################## RESTART THE RUNTIME BEFORE GOING TO NEXT STEP ###############
```

```
cd /content/drive/MyDrive/diagram_correction/Detectron
```

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

# import some common libraries
import matplotlib.pyplot as plt
import numpy as np
import cv2
from google.colab.patches import cv2_imshow

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

```
ls
```

```
#register datset
from detectron2.data.datasets import register_coco_instances
register_coco_instances("rooled1", {}, "./rooled_combined_json.json", "./rooled_collected_
```

```
#create metadata
rooled_metadata = MetadataCatalog.get("rooled1") #use a metadata name
dataset_dicts = DatasetCatalog.get("rooled1") #registered name
```

```
#verify the dataset
import random
```

```
for d in random.sample(dataset_dicts, 10):
    img = cv2.imread(d["file_name"])
    visualizer = Visualizer(img[:, :, ::-1], metadata=rooled_metadata, scale=0.5) #metadat
    vis = visualizer.draw_dataset_dict(d)
    cv2_imshow(vis.get_image()[:, :, ::-1])


#congiguring model values
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
import os

cfg = get_cfg()
cfg.merge_from_file("./detectron2_repo/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FP
cfg.DATASETS.TRAIN = ("rooled1",) #registered name
cfg.DATASETS.TEST = ()   # no metrics implemented for this dataset
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = "detectron2://COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x/13784960
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 tra
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 10 #change the total number of classes for which you hav

CUDA_LAUNCH_BLOCKING=1


#training a model
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()


#preparing a prediction model
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 = ("rooled", )
predictor = DefaultPredictor(cfg)


#prediction with the trained model
from detectron2.utils.visualizer import ColorMode
from detectron2.utils.visualizer import Visualizer
import random
from detectron2.data import DatasetCatalog
dataset_dicts = DatasetCatalog.get("rooled1") #registered name
for d in random.sample(dataset_dicts,10):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)

    print(outputs)

    v= Visualizer(im[:, :, ::-1],
                  metadata=rooled_metadata, #metadata name
```

```
                    scale=0.8,
                    instance_mode=ColorMode.IMAGE_BW   # remove the colors of unsegmented p
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    cv2_imshow(v.get_image()[:, :, ::-1])


#dump all the model configuration in a file
with open("rooled.yaml", "w") as f: #define a name for congiration file
  f.write(cfg.dump())
```