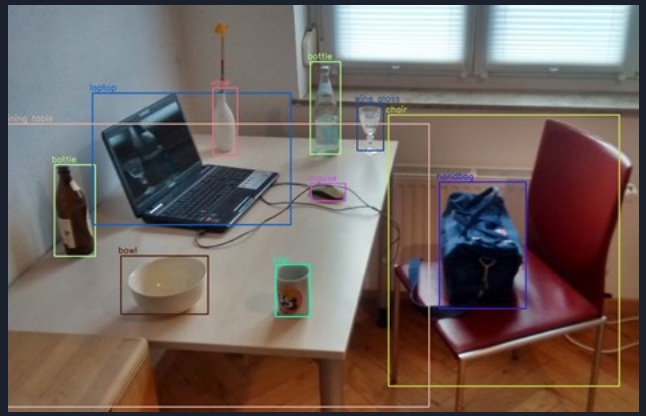
DETECTRON2

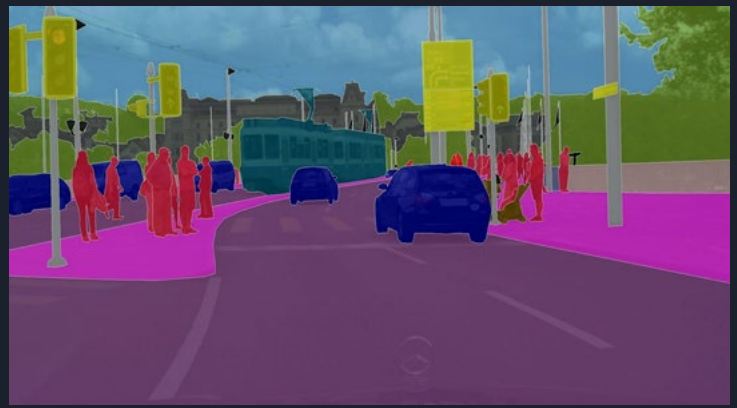
FacebookAI’s framework

Object Detection



Keypoint Detection

Semantic Segmentation



Panoptic Segmentation

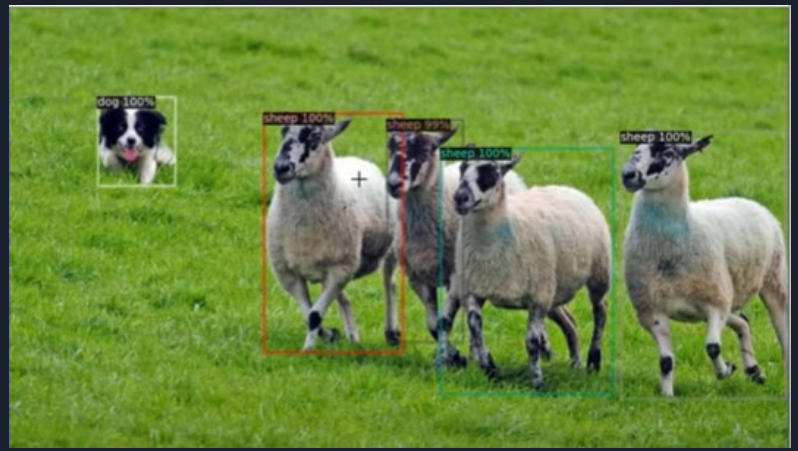
LVIS Instance Segmentation (Large Vocabulary Instance Segmentation)

OBJECT DETECTION

```python
class Detector:
    def __init__(self):
        self.cfg = get_cfg()

        # Load model config and pretrained model
        self.cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml"))
        self.cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml")

        self.cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.7
        self.cfg.MODEL.DEVICE = "cuda" #cpu or cuda

        self.predictor = DefaultPredictor(self.cfg)
```

```python
def onImage(self, imagePath):
    image = cv2.imread(imagePath)
    predictions = self.predictor(image)

    viz = Visualizer(image[:,:,::-1], metadata = MetadataCatalog.get(cfg.DATASETS.TRAIN[0]),
    instance_mode = ColorMode.IMAGE_BW)

    output = viz.draw_instance_predictions(predictions["instances"].to("cpu"))

    cv2.imshow("Result", output.get_image()[:,:,::-1])
    cv2.waitKey(0)
```

```python
from Detector import *

detector = Detector()

detector.onImage("images/3.jpg")
```



MASKED RCNN

```python
class Detector:
    def __init__(self, model_type = "OD"):
        self.cfg = get_cfg()

        # Load model config and pretrained model
        if model_type == "OD": # object detection
            self.cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml"))
            self.cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml")
        elif model_type == "IS": #instance segmentation
            self.cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x
            self.cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN

        self.cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.7
        self.cfg.MODEL.DEVICE = "cuda" #cpu or cuda

        self.predictor = DefaultPredictor(self.cfg)
```

```python
from Detector import *

detector = Detector(model_type="IS")

detector.onImage("images/3.jpg")
```

```
def onImage(self, imagePath):
    image = cv2.imread(imagePath)
    predictions = self.predictor(image)

    viz = Visualizer(image[:,:,::-1], metadata = MetadataCatalog.get(self.cfg.DATASETS.TRAIN[0]),
    instance_mode = ColorMode.SEGMENTATION)

    output = viz.draw_instance_predictions

    cv2.imshow("Result", output.get_image()[:,:,::-1])
    cv2.waitKey(0)
```

segmentation
InstanceSegmentation

for more: https://www.youtube.com/watch?v=Pb3opEFP94U&t=569s