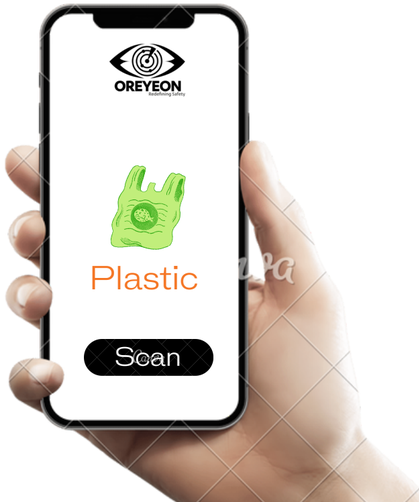
ZAKA
WWW.ZAKA.AI

×

OREYEON
Redefining Safety

Waste Detection in the Wild



Problem:

- Kill or stunt plant growth.
- Small animals stuck in small bottles and starve to death.
- Birds, small animals and water creatures eat or get tangled in plastic bags or plastic 6-ring holders.

Literature:

How the problem has been previously solved

1) Litter Detection with Deep Learning: A Comparative Study:

Table 6. Litter detection results on TACO (best results appear in bold).

Methods	AP50	AP@	AR@	F1@
RetinaNet [41]	50.6	26.7	37.1	31.1
Faster R-CNN [28]	51.1	28.1	36.9	31.9
Mask R-CNN [43]	52.3	29.2	38.6	33.2
EfficientDet-d0 [30]	32.7	23.8	28.4	25.9
EfficientDet-d5 [30]	42.3	35.2	40.3	37.6
YOLO-v5s [32]	54.7	38.8	58.1	46.5
YOLO-v5x [32]	**63.3**	**48.4**	**66.4**	**56.0**

- YOLO-v5s proved to be the most promising approach to be run on mobile devices
- No segmentation was performed, only based on object detection

2) A Multi-Level Approach to Waste Object Segmentation:

Dataset: TACO (Test)	Backbone	IoU	mIoU	Prec	Mean
Baseline Approaches					
FCN-8s [17]	VGG-16	70.43	84.31	85.50	92.21
DeepLabv3 [23]	ResNet-101	**83.02**	**90.99**	**88.37**	**94.00**
Proposed Multi-Level (ML) Model					
FCN-8s-ML	VGG-16	74.21 (+3.78)	86.35 (+2.04)	90.36 (+4.86)	94.65 (+2.44)
DeepLabv3-ML	ResNet-101	**86.58** (+3.56)	**92.90** (+1.91)	**92.52** (+4.15)	**96.07** (+2.07)

- Using the TACO dataset deeplabv3 performed well
- Introduced their dataset MJU which is an indoor dataset consisting of RGBD images
- Here they used TACO-1 where all classes where combined into 1 class only called ‘litter’. Segmentation was performed.

Literature:

How the problem has been previously solved

3) Instance Segmentation of Multiclass Litter and Imbalanced Dataset Handling –A Deep Learning Model Comparison:

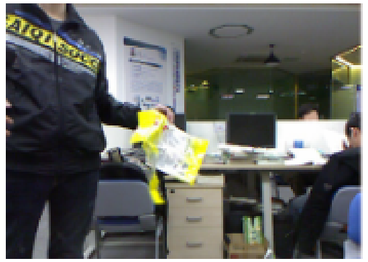
Metric	area	Mask R-CNN score	DetectoRS score	diff
mAP	all	0.127	0.167	+0.040
$mAP_{0.5}$	all	0.159	0.203	+0.044
$mAP_{0.75}$	all	0.136	0.178	+0.042
mAP	small	0.022	0.044	+0.022
mAP	medium	0.064	0.140	+0.076
mAP	large	0.150	0.194	+0.044
mAR	all	0.320	0.479	+0.159
mAR	small	0.021	0.058	+0.037
mAR	medium	0.093	0.252	+0.159
mAR	large	0.367	0.507	+0.140

DetecroRS performed better but it demands more GPU than Mask R-CNN

Our Solution

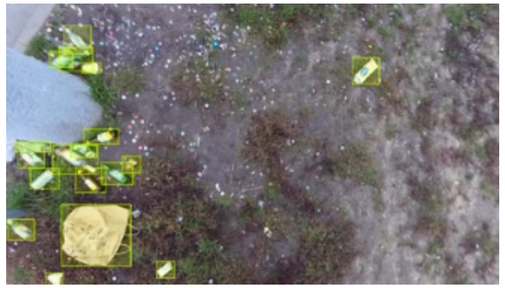
Dataset :

MJU Dataset



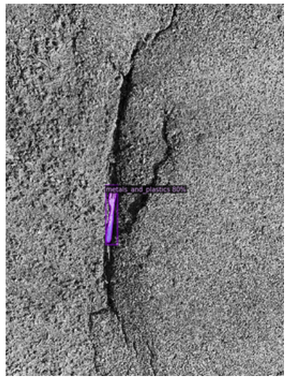
2475 images and 2532 annotations

UAVVASTE



772 images and 3716 annotations

TACO



1500 images with 4784 annotations

Dataset :

The format of the annotations is provided in COCO format

TACO: Trash Annotations in Context

This dataset labeled its images under 60 categories corresponding to 28 super categories

is open-source, and anyone could help in labelling

1500 images with 4784 annotations (avg 3.19 object per images)

Dataset :

COCO Format:


```
dataset.keys()
```

```
dict_keys(['info', 'images', 'annotations', 'scene_annotations', 'licenses', 'categories', 'scene_categories'])
```

```
dataset['images'][1]
```

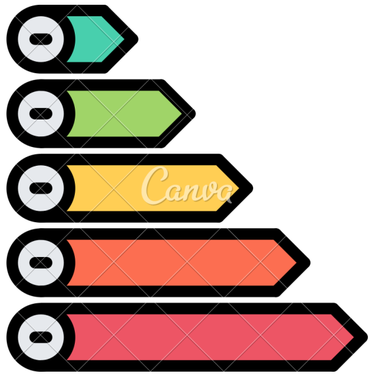
```
{'id': 1,
 'width': 1537,
 'height': 2049,
 'file_name': 'batch_1/000008.jpg',
 'license': None,
 'flickr_url': 'https://farm66.staticflickr.com/65535/47803331152_ee00755a2e_o.png',
 'coco_url': None,
 'date_captured': None,
 'flickr_640_url': 'https://farm66.staticflickr.com/65535/47803331152_19beae025a_z.jpg'}
```

```
dataset['annotations'][7]
```

```
{'id': 8,
 'image_id': 3,
 'category_id': 7,
 'segmentation': [[643.0,
   1453.0,
   649.0,
   1445.0,
   653.0,
   1442.0,
   657.0,
   1450.0,
   663.0,
   1459.0,
   662.0,
   1467.0,
   656.0,
   1478.0,
   651.0,
   1481.0,
   644.0,
   1476.0,
   643.0,
   1465.0,
   638.0,
   1459.0,
   634.0,
   1459.0,
   643.0,
   1453.0]],
 'area': 578.5,
 'bbox': [634.0, 1442.0, 29.0, 39.0],
 'iscrowd': 0}
```

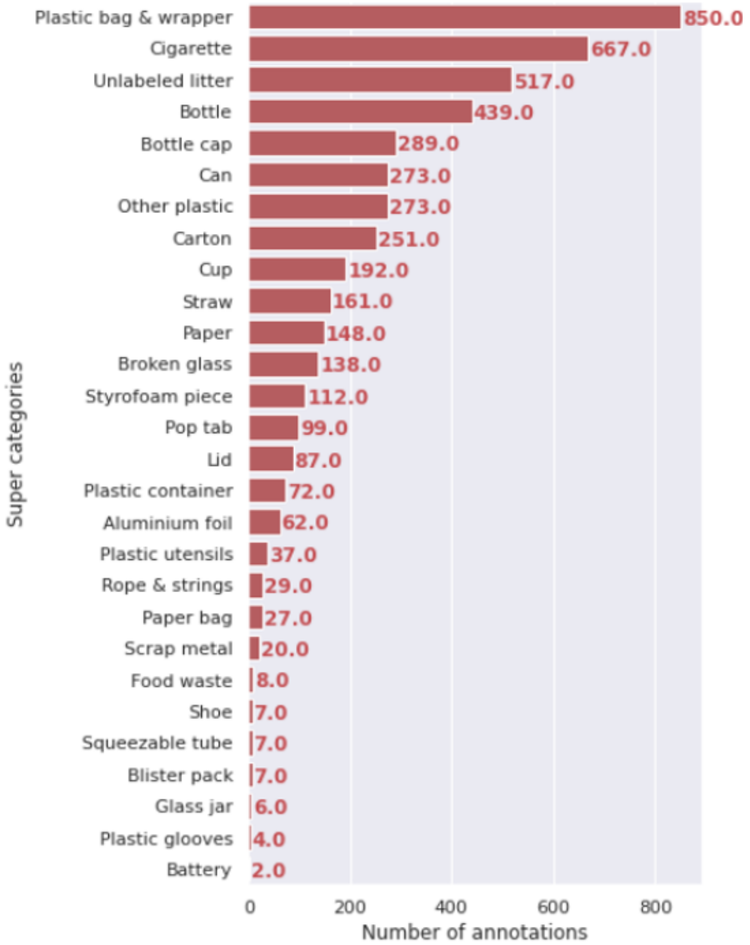
Dataset :

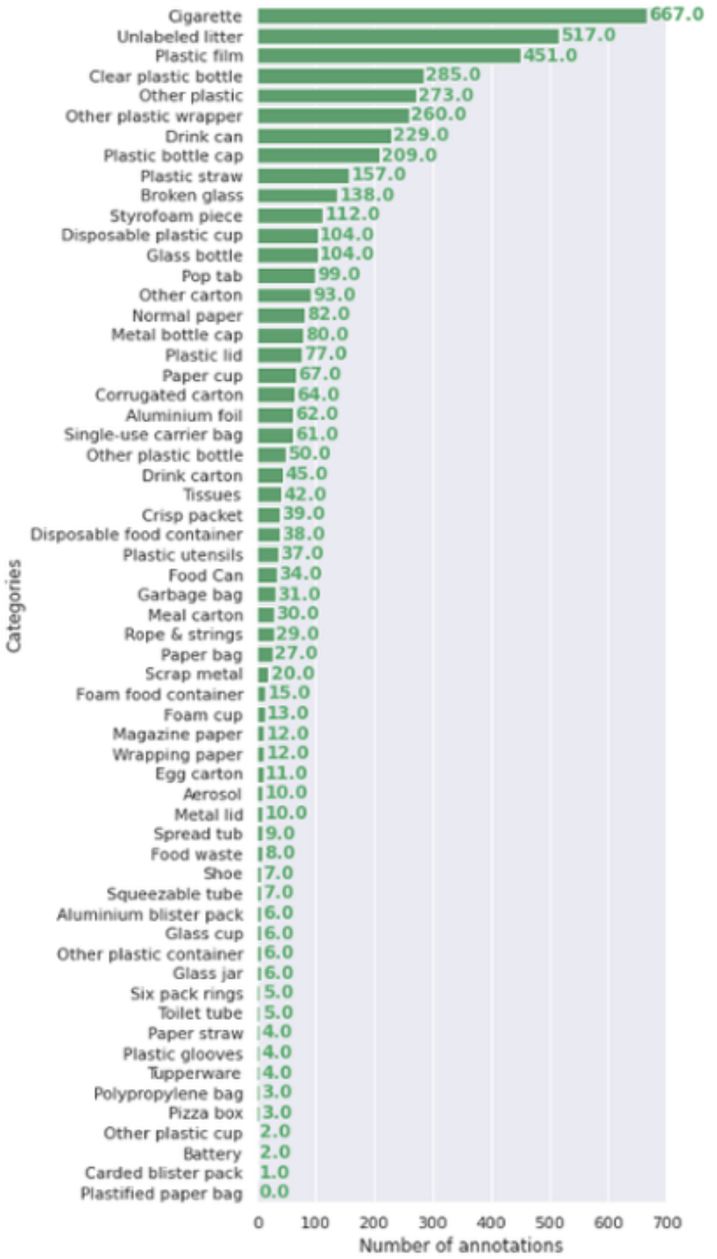
COCO Format:



Categories

There are 6 categories with 3 or fewer image representations, namely pizza box, polypropylene bag, battery, other plastic cup, carded blister pack, and plastified paper bag.

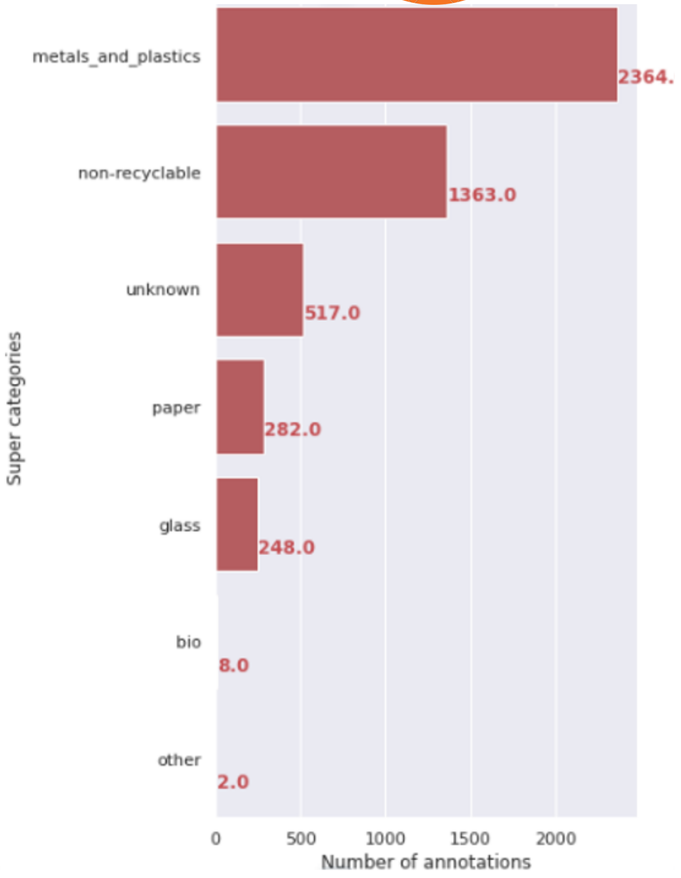




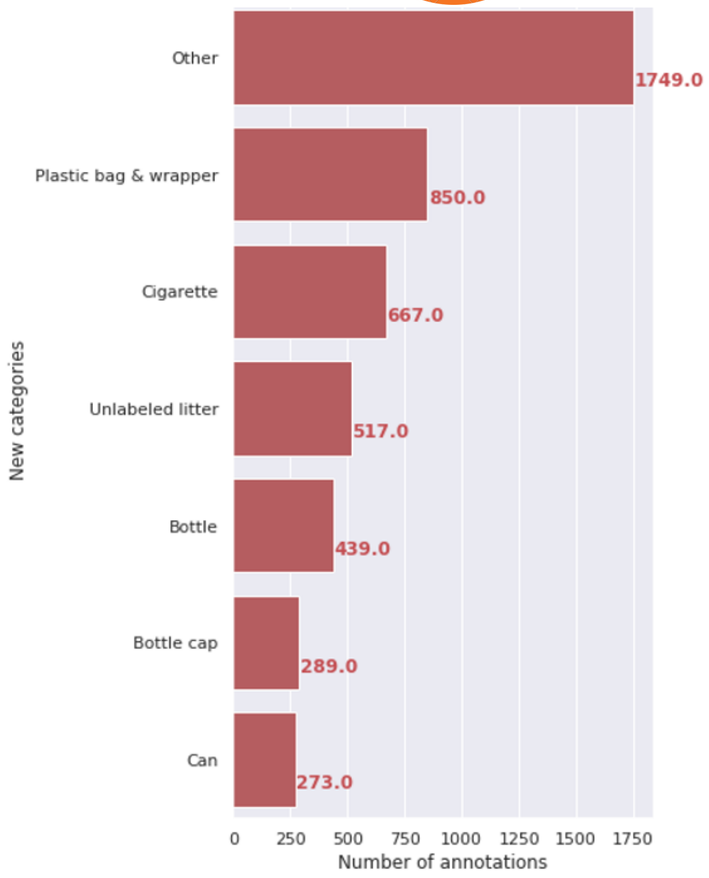
Grouping:

Grouping methods

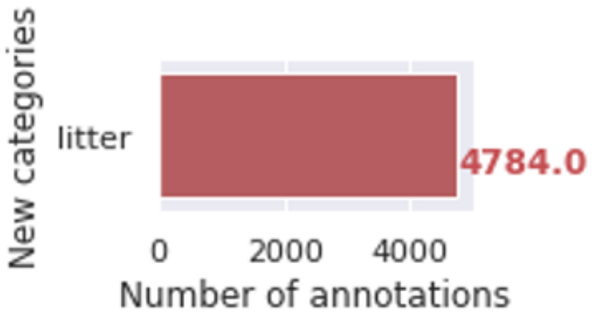
1



2



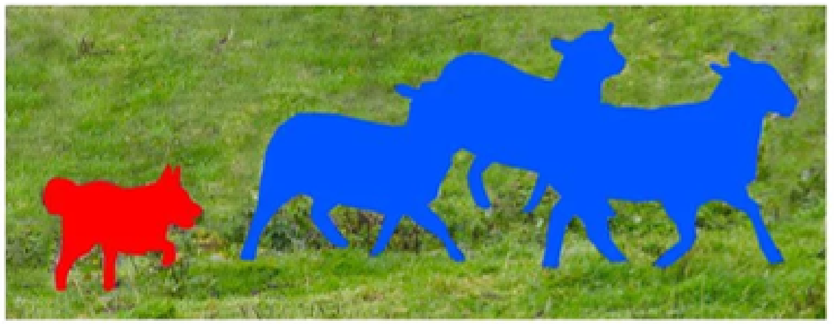
3



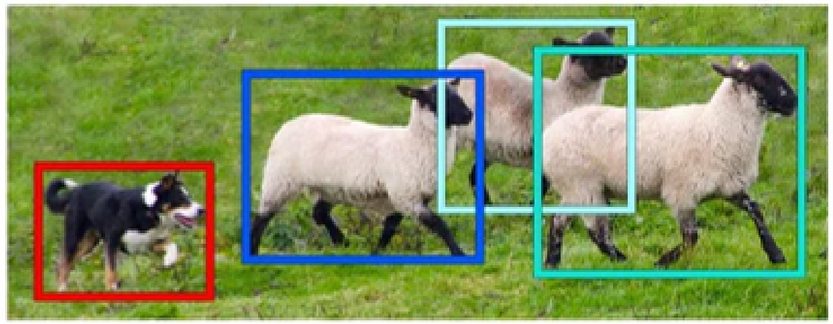
Model:

Object Detection and Segmentation

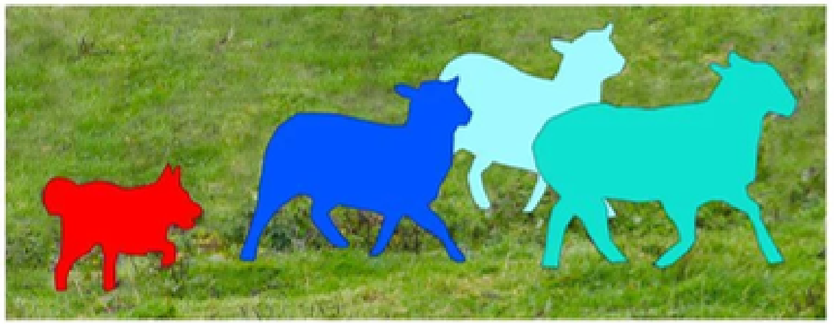
Image Recognition



Semantic Segmentation



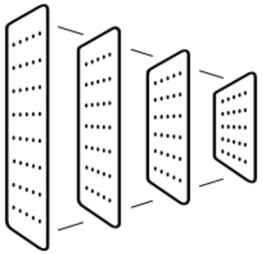
Object Detection



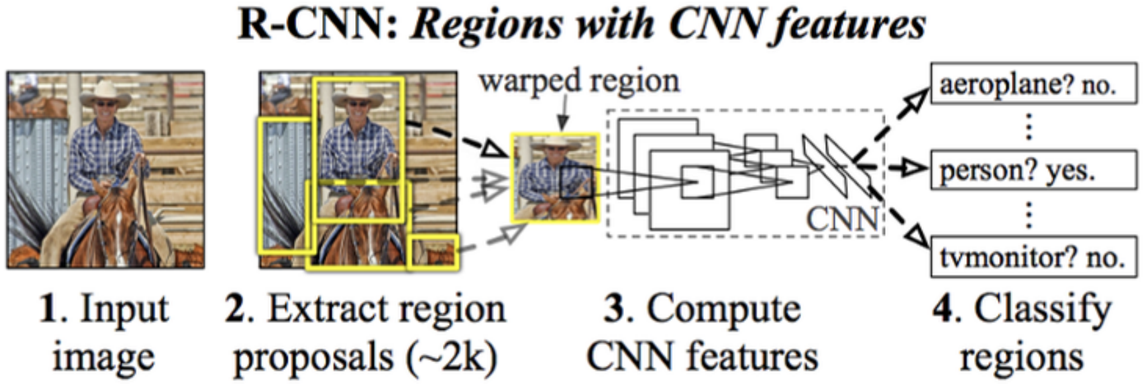
Instance Segmentation

Model:

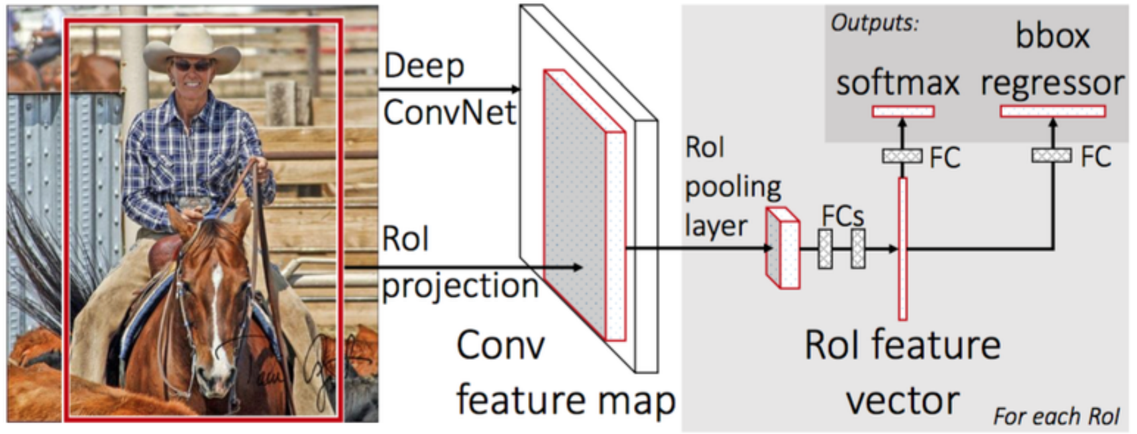
Object Detection and Segmentation



R-CNN

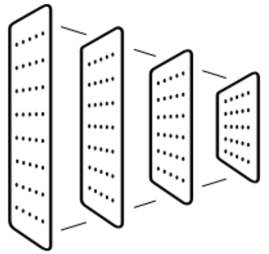


Fast R-CNN

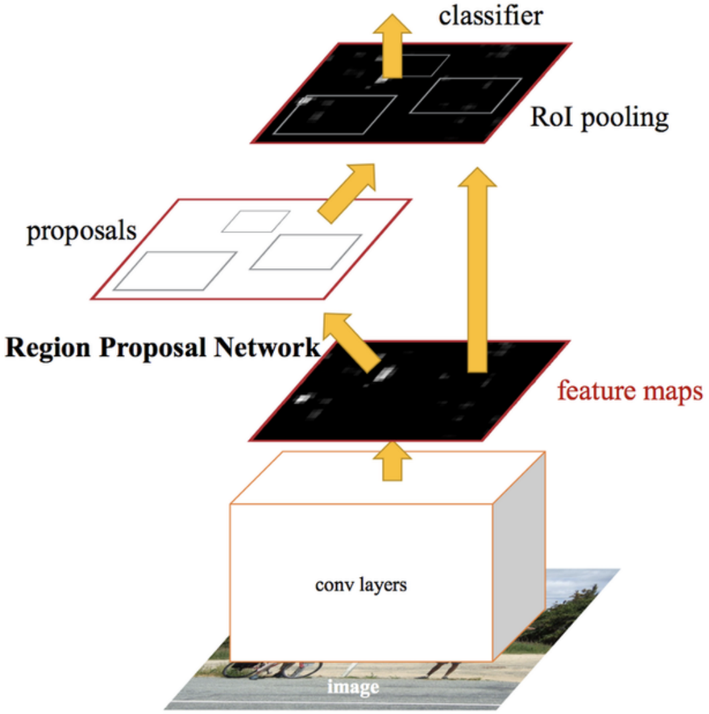


Model:

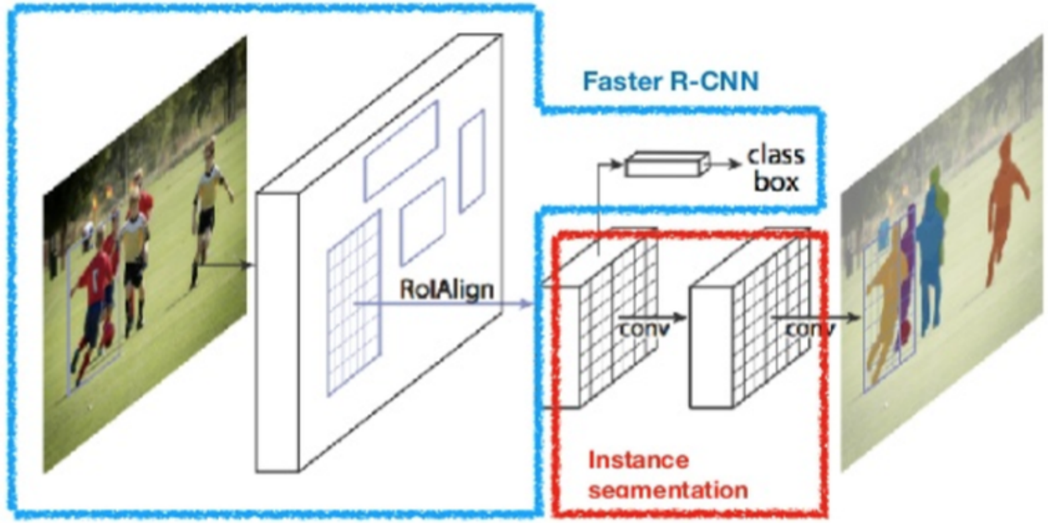
Object Detection and Segmentation



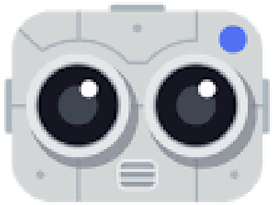
Faster R-CNN



Mask R-CNN



Model:



Detectron2

FACEBOOK AI

main ▾ detectron2 / configs / COCO-InstanceSegmentation / Go to file Add file ▾ ···

ppwwyyxx and facebook-github-bot set default pre-trained model to empty ... 5b29850 on Sep 20, 2021 History

..

mask_rcnn_R_101_C4_3x.yaml	Initial commit	3 years ago
mask_rcnn_R_101_DC5_3x.yaml	Initial commit	3 years ago
mask_rcnn_R_101_FPN_3x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_C4_1x.py	set default pre-trained model to empty	14 months ago
mask_rcnn_R_50_C4_1x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_C4_3x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_DC5_1x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_DC5_3x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_FPN_1x.py	set default pre-trained model to empty	14 months ago
mask_rcnn_R_50_FPN_1x.yaml	Initial commit	3 years ago
mask_rcnn_R_50_FPN_1x_giou.yaml	Configurable loss for rpn box regression and giou support	2 years ago
mask_rcnn_R_50_FPN_3x.yaml	Initial commit	3 years ago
mask_rcnn_X_101_32x8d_FPN_3x.yaml	Initial commit	3 years ago
mask_rcnn_regnetx_4gf_dds_fpn_1x.py	Add RegNet support	2 years ago
mask_rcnn_regnety_4gf_dds_fpn_1x.py	Add RegNet support	2 years ago

Model:

Detectron2

- Install:

```
!pip install 'git+https://github.com/facebookresearch/detectron2.git' > /dev/null
```

- Register dataset:

The division was 80% training, 10% validation and 10% testing. Before that we kept 50 images on the side for inference.

```
Training images: 1200, Validation images: 125, Testing images: 125 ,and Inference images: 50
Training annotations: 3794, Validation annotations: 416, Testing annotations: 383, and Inference annotations: 191
```

```
register_coco_instances('train',{},'/content/SplittedData/train.json', '/content/TACO/data')
register_coco_instances('test',{},'/content/SplittedData/test.json', '/content/TACO/data')
register_coco_instances('val',{},'/content/SplittedData/val.json', '/content/TACO/data')
```

- Create Dataset Dict:

```
dataset_dicts_train = get_detection_dataset_dicts(['train'])
```

- Metadata:

```
_dataset_metadata_train = MetadataCatalog.get('train')
_dataset_metadata_train.thing_colors = [cc['color'] for cc in builtin_meta.COCO_CATEGORIES]
```

Model:

Detectron2

```
cfg = get_cfg()

cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml"))

cfg.DATASETS.TRAIN = ("train",)
cfg.DATASETS.TEST = ("test_val",)
cfg.DATALOADER.NUM_WORKERS = 2
cfg.DATALOADER.SAMPLE_TRAIN = 'RepeatFactorTrainingSampler'
cfg.DATALOADER.REPEAT_THRESHOLD = 0.3
```

```
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.0025
cfg.SOLVER.MAX_ITER = 3000
cfg.SOLVER.CHECKPOINT_PERIOD = 1000


# minimum image size for the train set
cfg.INPUT.MIN_SIZE_TRAIN = (800,)
# maximum image size for the train set
cfg.INPUT.MAX_SIZE_TRAIN = 1333
# minimum image size for the test set
cfg.INPUT.MIN_SIZE_TEST = 800
# maximum image size for the test set
cfg.INPUT.MAX_SIZE_TEST = 1333
```

```
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 512 #128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = NUM_CLASSES

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

trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
```

Model:

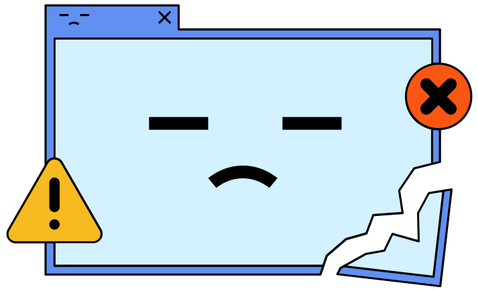
Detectron2

```python
from detectron2.data import transforms as T
train_augmentations = [
    T.RandomBrightness(0.5, 2),
    # T.RandomContrast(0.5, 2),
    T.RandomSaturation(0.5, 2),
    T.RandomFlip(prob=0.5, horizontal=True, vertical=False),
    # T.RandomFlip(prob=0.5, horizontal=False, vertical=True),
]

from detectron2.data import DatasetMapper, build_detection_train_loader
class MyTrainer(DefaultTrainer):
    # #@classmethod
    # def build_evaluator(cls, cfg, dataset_name, output_folder=None):
    #     if output_folder is None:
    #         output_folder = os.path.join(cfg.OUTPUT_DIR,"inference")
    #     return COCOEvaluator(dataset_name, cfg, True, output_folder)
    @classmethod
    def build_train_loader(cls, cfg):
        # if "SemanticSegmentor" in cfg.MODEL.META_ARCHITECTURE:
        mapper = DatasetMapper(cfg, is_train=True, augmentations=train_augmentations)
        # else:
            # mapper = None
        return build_detection_train_loader(cfg, mapper=mapper)

trainer = MyTrainer(cfg)
# trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
```

Custom Augmentation using detectron2: ERROR



9 frames

/usr/local/lib/python3.7/dist-packages/detectron2/layers/wrappers.py in forward(self, x)

```
    112
    113         x = F.conv2d(
--> 114             x, self.weight, self.bias, self.stride, self.padding, self.dilation, self.groups
    115         )
    116         if self.norm is not None:
```

RuntimeError: CUDA out of memory. Tried to allocate 976.00 MiB (GPU 0; 14.76 GiB total capacity; 10.65 GiB already allocated; 681.75 MiB free; 12.78 GiB reserved in total by PyTorch) If reserved memory is >> allocated memory try setting max_split_size_mb to avoid fragmentation. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF

SEARCH STACK OVERFLOW

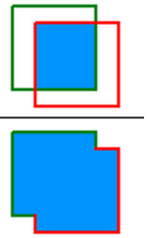
Metrics:

Intersection over Union (IoU)

$$\text{IoU} = \frac{\text{area}(gt \cap pd)}{\text{area}(gt \cup pd)}$$

Degree of overlap between the ground(gt) truth and prediction(pd)

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$



ranges between 0 and 1

Average Precision

$$AP@\alpha = \int_0^1 p(r)\,dr$$

Area Under the Precision-Recall Curve(AUC-PR) evaluated at α IoU threshold

AP50 and A75 then they just mean AP calculated at IoU=0.5 and IoU=0.75

Mean Average Precision (mAP) $\overline{X}$

$$\text{mAP@}\alpha = \frac{1}{n}\sum_{i=1}^{n} \text{AP}_i \quad \text{for n classes}$$

Calculated individually for each class

Results:

Model 1:
Mask-RCNN_R_50_FPN_3x

Model 2:
Mask-RCNN_R_101_FPN_3x

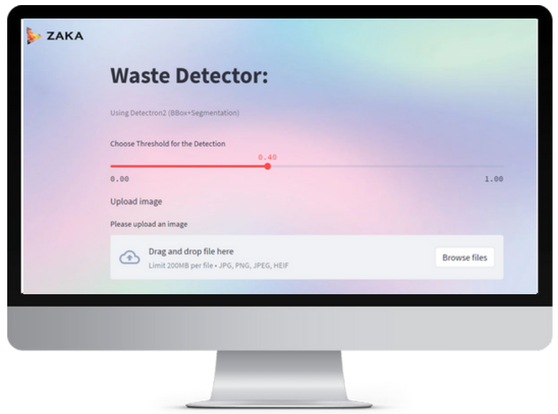
		segm					
		AP	AP50	AP75	APs	APm	APl
Group 1	Model 1	6.666	9.972	6.933	**1.334**	**5.070**	10.376
	Model 2	**7.789**	**11.280**	**7.800**	1.175	3.884	**11.530**
Group 2	Model 1	18.018	24.856	19.339	**2.163**	6.429	21.844
	Model 2	**19.997**	**28.186**	**19.536**	1.651	**8.571**	**23.305**
Group 3	Model 1	**40.114**	**54.666**	**43.881**	**3.607**	18.866	**56.331**
	Model 2	39.832	53.365	43.374	2.067	**19.057**	55.992
		bbox					
		AP	AP50	AP75	APs	APm	APl
Group 1	Model 1	6.851	10.093	7.881	**2.555**	**5.277**	9.810
	Model 2	**8.269**	**12.224**	**8.743**	1.804	4.662	**11.263**
Group 2	Model 1	18.188	25.321	20.742	**3.322**	6.638	21.458
	Model 2	**20.831**	**28.212**	**24.407**	2.830	**10.856**	**23.078**
Group 3	Model 1	**40.656**	**54.591**	**45.149**	**7.997**	21.529	**56.021**
	Model 2	40.149	53.437	44.287	4.453	**21.693**	55.010

AP Across Scales:

AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

Deployment:

- Create web apps for data science and machine learning in a short time
- Compatibility
- Simple to use



Streamlit

Compatible with
major libraries & frameworks

scikit-learn · Altair · bokeh · LaTeX
Keras · plotly · OpenCV · Vega-Lite
PyTorch · NumPy · seaborn · DECK.GL
TensorFlow · python · matplotlib · pandas

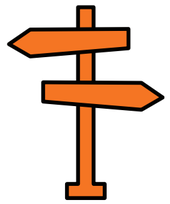
Deployment:

Streamlit

1) Upload Code to Github

Nabil-Miri Update main.py	29a8d2f 7 hours ago	**100** commits
.idea	edits for local deployment	6 days ago
assets	edits for local deployment	6 days ago
hide	edits for local deployment	6 days ago
.gitattributes	Add design file	6 days ago
README.md	Initial commit	6 days ago
img.jpg	edit	8 hours ago
main.py	Update main.py	7 hours ago
model_final.pth	Add design file	6 days ago
my_metadata	edits for local deployment	6 days ago
requirements.txt	Update requirements.txt	9 hours ago

Deployment:



Streamlit

← Back

Deploy an app

Repository — Paste GitHub URL

Nabil-Miri/mystreamlit

Branch

main

Main file path

main.py

Advanced settings...

Deploy!

Requirments.txt :

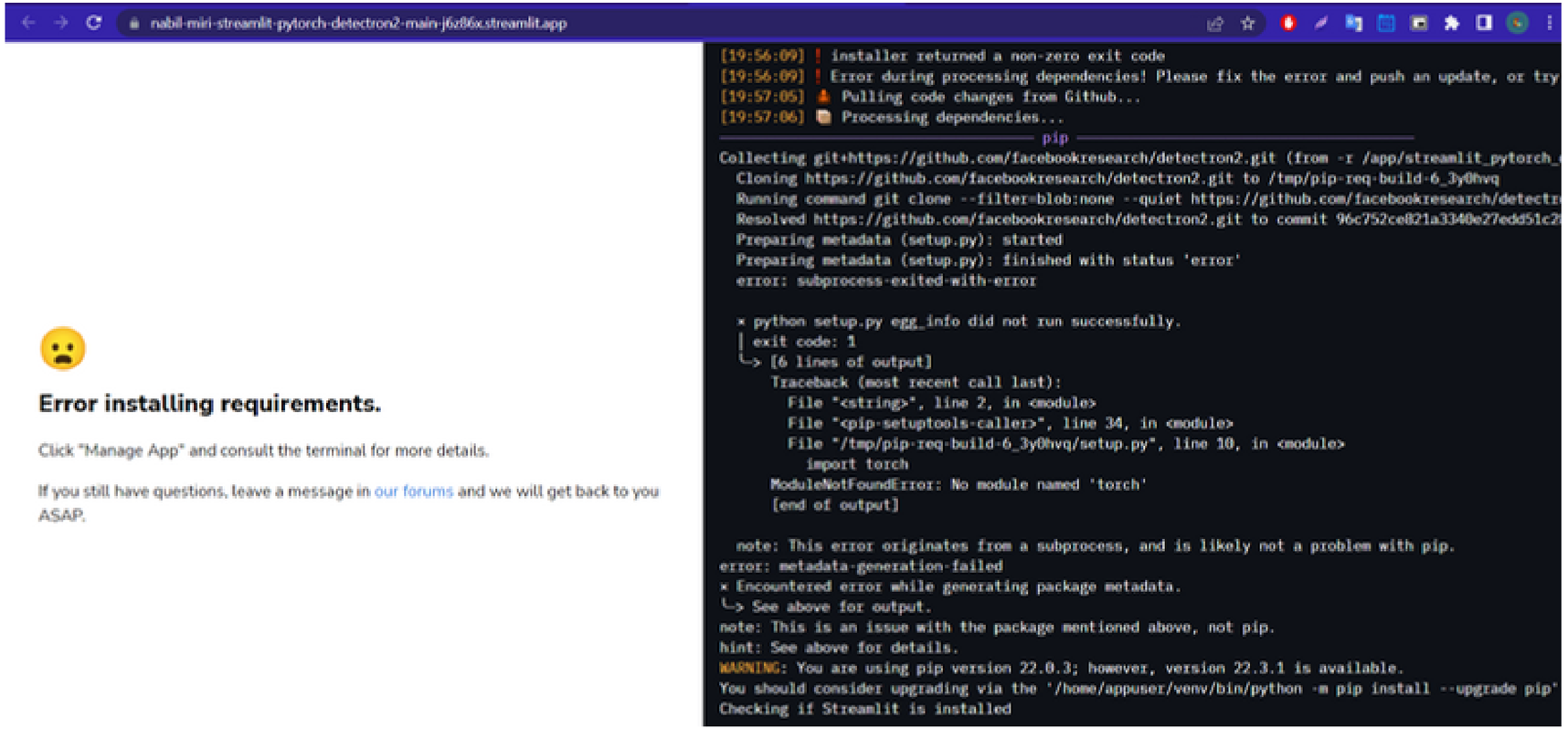
```
1  pyyaml==5.1
2  torch==1.9.0
3  torchvision==0.10.0
4  streamlit
5  opencv-contrib-python-headless
6  git+https://github.com/facebookresearch/detectron2.git #detectron2
```

Deployment:

Streamlit Cloud:

Requirments.txt :

```
1  pyyaml==5.1
2  torch==1.9.0
3  torchvision==0.10.0
4  streamlit
5  opencv-contrib-python-headless
6  git+https://github.com/facebookresearch/detectron2.git #detectron2
```



Deployment:

Solution: (Interesting Way) 🤓

1 Comment the detectronv2 line

```
pyyaml==5.1
torch==1.9.0
torchvision==0.10.0
streamlit
opencv-contrib-python-headless
#git+https://github.com/facebookresearch/detectron2.git #detectron2
```

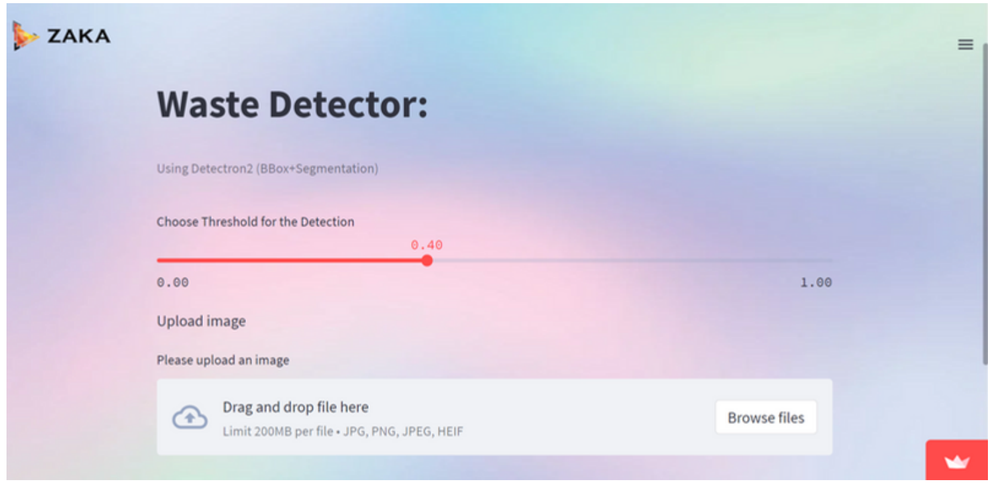
2 Uncomment again the line

```
pyyaml==5.1
torch==1.9.0
torchvision==0.10.0
streamlit
opencv-contrib-python-headless
git+https://github.com/facebookresearch/detectron2.git #detectron2
```

Streamlit Starts Running:

```
[19:57:09] ! installer returned a non-zero exit code
[19:57:42] ! Streamlit server consistently failed status checks
[19:57:42] ! Please fix the errors, push an update to the git repo, or re
[20:03:49] 🐙 Pulling code changes from Github...
[20:03:50] 📦 Processing dependencies...
                                    pip
Collecting pyyaml==5.1
  Downloading PyYAML-5.1.tar.gz (274 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 274.2/274.2 KB 12.4 MB/s eta 0:0
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting torch==1.9.0
  Downloading torch-1.9.0-cp39-cp39-manylinux1_x86_64.whl (831.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 831.4/831.4 MB 283.5 MB/s eta 0:0
Collecting torchvision==0.10.0
  Downloading torchvision-0.10.0-cp39-cp39-manylinux1_x86_64.whl (22.1 MB
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22.1/22.1 MB 223.0 MB/s eta 0:0
Collecting streamlit
  Downloading streamlit-1.14.1-py2.py3-none-any.whl (9.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.2/9.2 MB 225.1 MB/s eta 0:00
Collecting opencv-contrib-python-headless
  Downloading opencv_contrib_python_headless-4.6.0.66-cp36-abi3-manylinux
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.5/54.5 MB 260.7 MB/s eta 0:0
```

Streamlit Works:



Deployment:

Solution: (Boring Way)

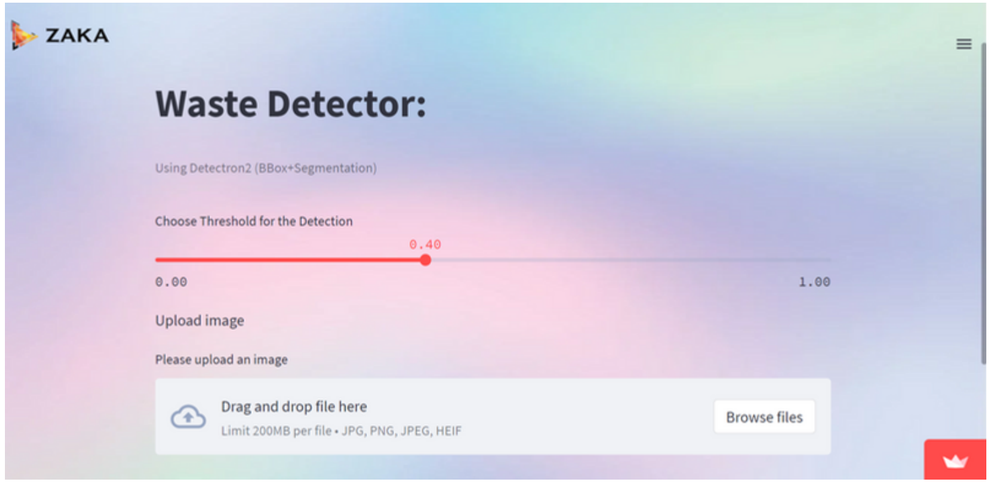
1 Replace the line

```
pyyaml==5.1
torch==1.9.0
torchvision==0.10.0
streamlit

-f https://dl.fbaipublicfiles.com/detectron2/wheels/cpu/torch1.9/index.html
detectron2

opencv-contrib-python-headless
```

Streamlit Works:



Install Pre-Built Detectron2 (Linux only)

Choose from this table to install v0.6 (Oct 2021):

CUDA	torch 1.10	torch 1.9	torch 1.8
11.3	▶ install		
11.1	▶ install	▶ install	▶ install
10.2	▶ install	▶ install	▶ install
10.1			▶ install
cpu	▶ install	▶ install	▶ install

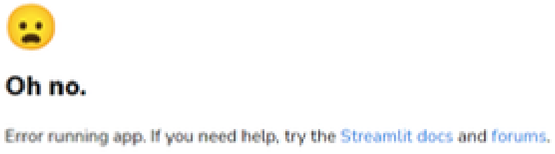
CUDA	torch 1.10	torch 1.9
11.3	▶ install	
11.1	▶ install	▶ install
10.2	▶ install	▶ install
10.1		

cpu	▶ install	▼ install
		`python -m pip install detectron2 -f \`
		`https://dl.fbaipublicfiles.com/detectron2/wheels/cpu/torch1.9/index.html`

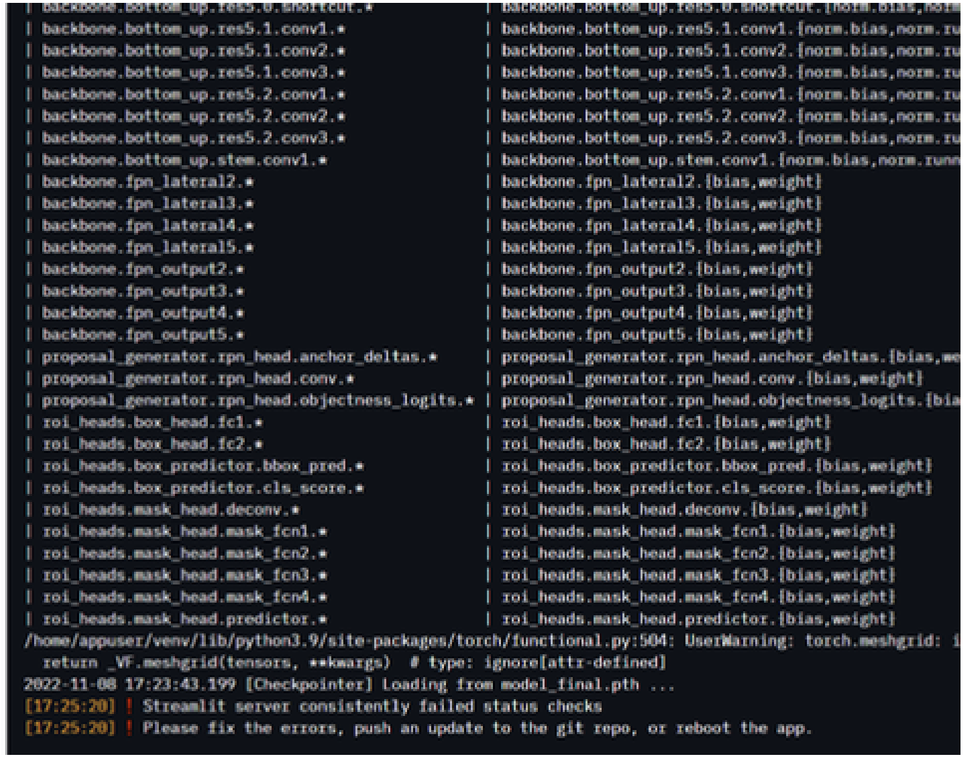
Deployment:

Error: App Crashing after 3 trials

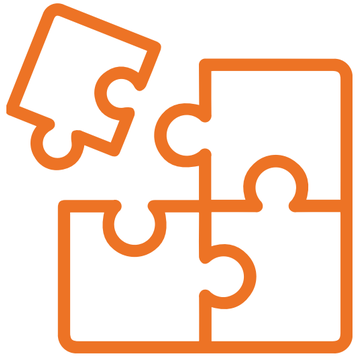
Hello again old friend



App is exceeding the 1GB resource limit



Deployment:



Solution: Cache

```python
@st.cache
def inference(predictor, img):
    return predictor(img)
```

```python
@st.cache
def output_image(img_array, outputs):
```

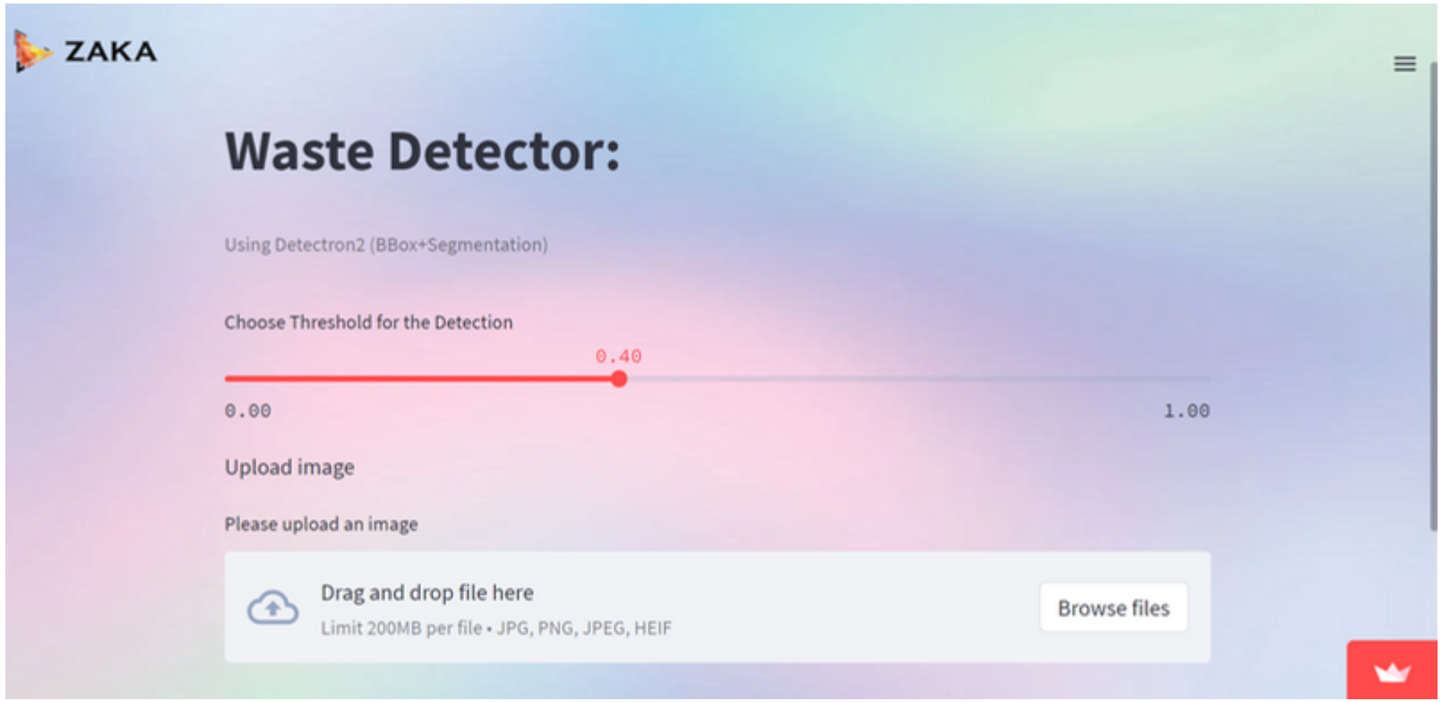
```python
@st.cache(persist=True)
def initialization():
    """Loads configuration and model for the prediction.

    Returns:
        cfg (detectron2.config.config.CfgNode): Configuration for the model.
        predictor (detectron2.engine.defaults.DefaultPredicto): Model to use.
            by the model.

    """
```

Deployment:

Final App:



Link:

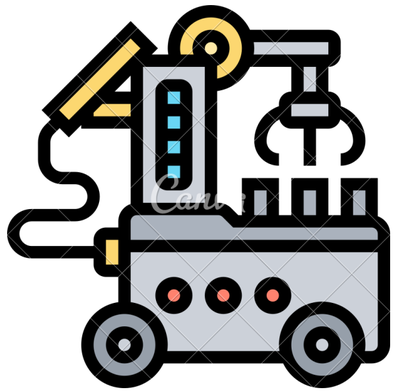
SCAN ME

Benefits :

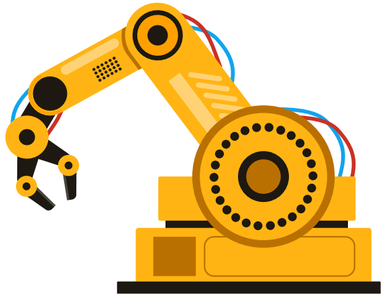
This technology could save on costs or reduce expenses for disposal

Increase the recycling process efficiency thus creating a better environment

Possible applications



Autonomous mobile robot
to detect, collect and recycle unwanted waste



Automated waste recycling line

Future Work:

- [x] Perform "Iterative Stratification" data splitting based on annotations
- [x] Add more photos (test the idea of combining datasets)
- [x] Yolov7 segmentation
- [x] Mobile app (Yolov7s or tiny - D2GO - EfficientDet-B0....)

Team :

AVENGERS

Mentor

MRAD SLEIMAN

NABIL MIRI

ABDULRAHIM EL MOHAMAD

thank you!

ZAKA

WWW.ZAKA.AI

OREYEON

Redefining Safety