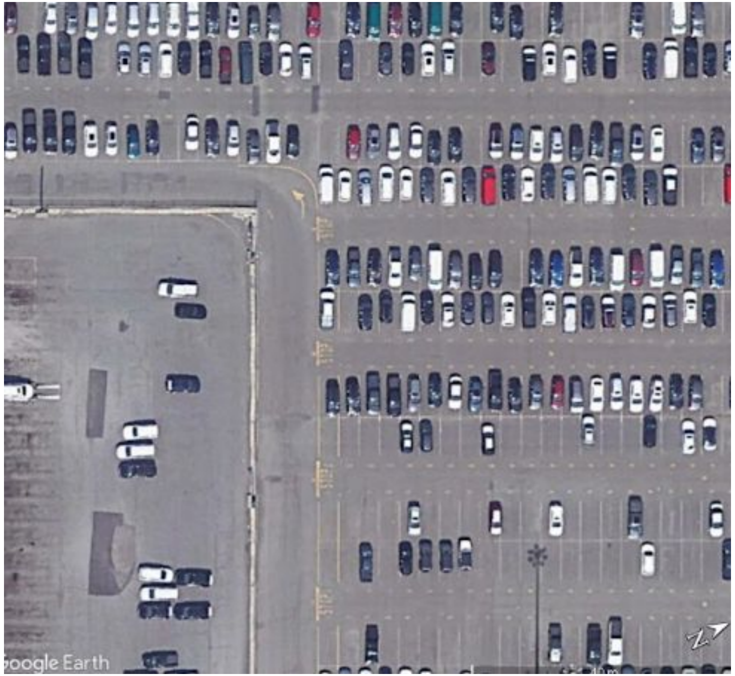
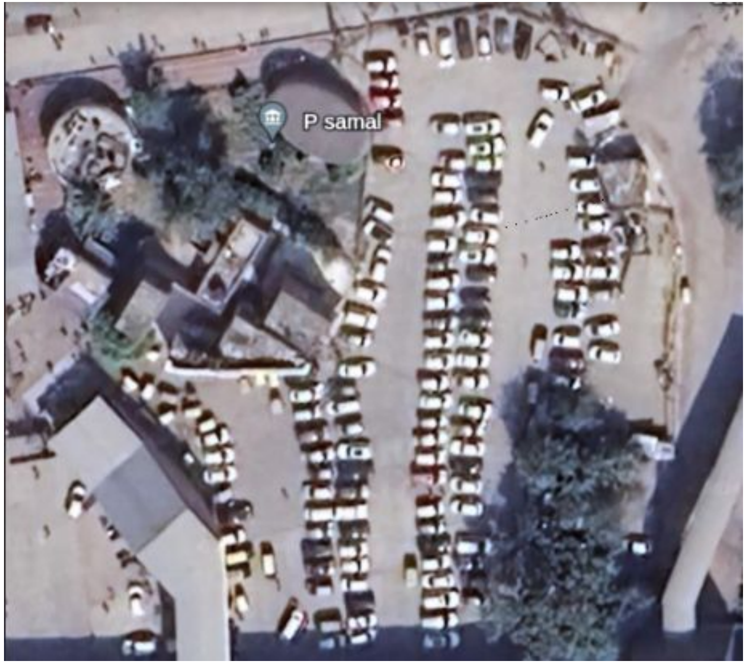
Prompt image (107.jpg)

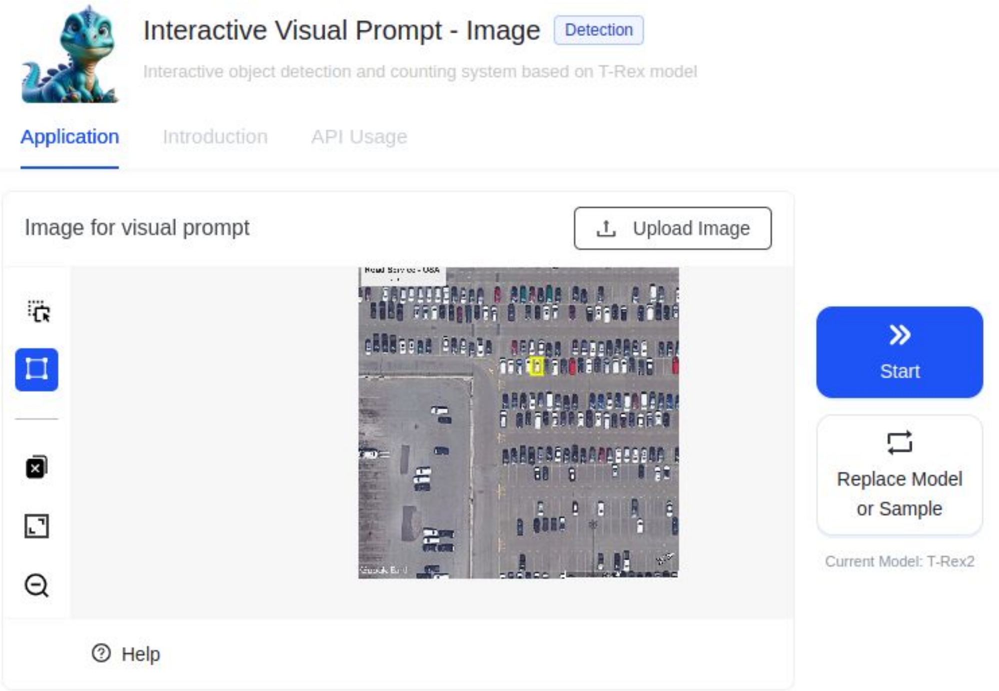


Test image (test_5.png)



Using Deep Data Space

https://deepdataspace.com/playground/ivp



Interactive Visual Prompt - Image

Interactive object detection and counting system based on T-Rex model

Image for visual prompt

Current Model: T-Rex2

Result of “Interactive visual prompt”

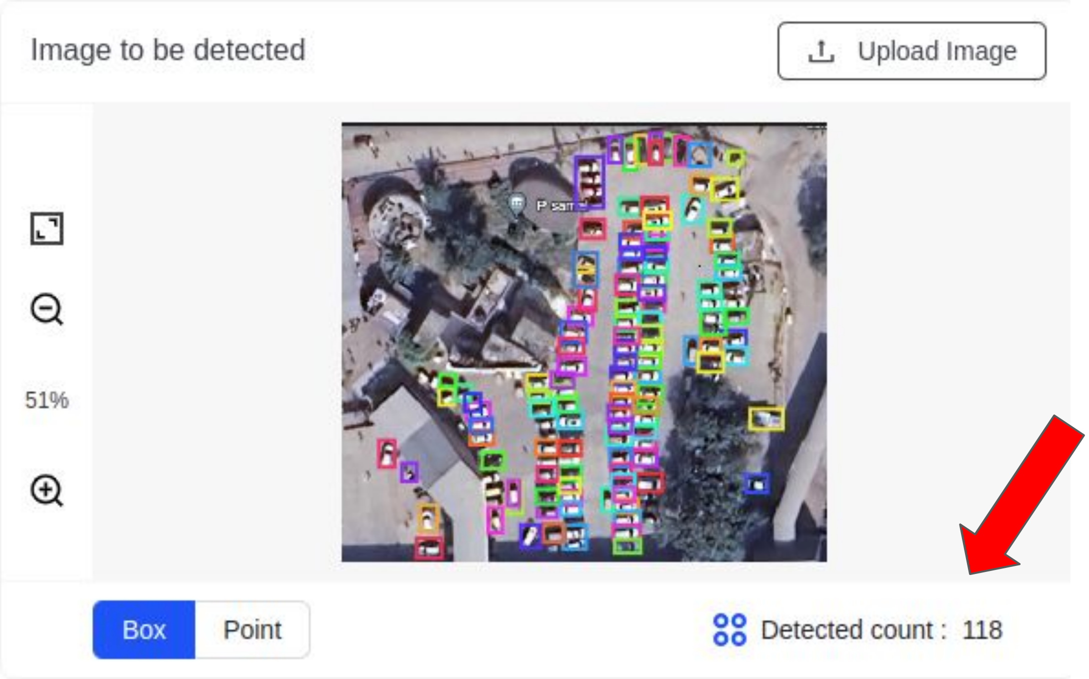
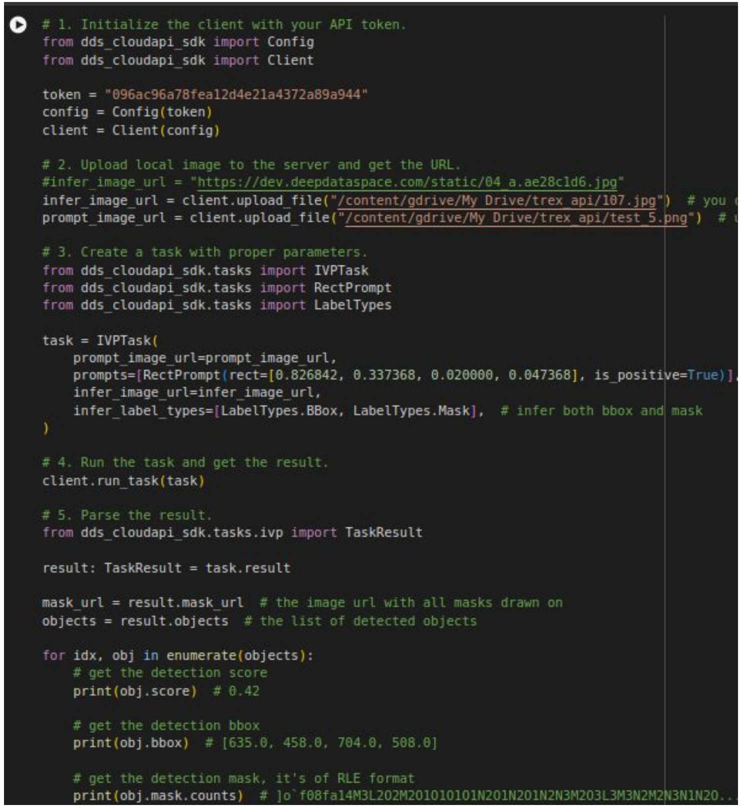


Image to be detected

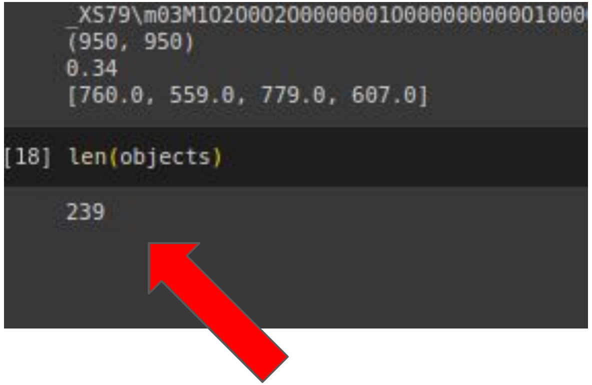
Detected count : 118

Predicts 118 vehicles

Using Interactive Visual Prompt (iVP) API



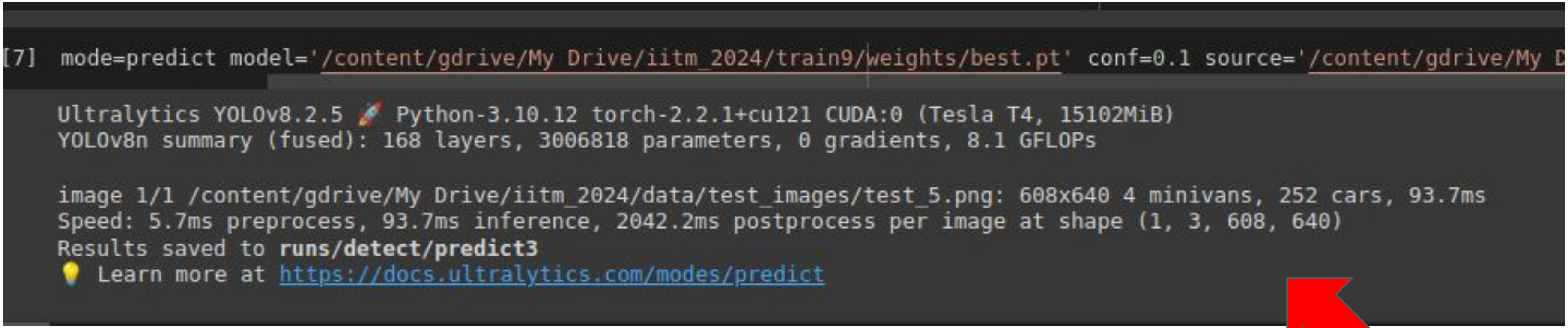
```
# 1. Initialize the client with your API token.
from dds_cloudapi_sdk import Config
from dds_cloudapi_sdk import Client

token = "096ac96a78fea12d4e21a4372a89a944"
config = Config(token)
client = Client(config)

# 2. Upload local image to the server and get the URL.
#infer_image_url = "https://dev.deepdataspace.com/static/04_a.ae28c1d6.jpg"
infer_image_url = client.upload_file("/content/gdrive/My Drive/trex_api/107.jpg")  # you
prompt_image_url = client.upload_file("/content/gdrive/My Drive/trex_api/test_5.png")  #

# 3. Create a task with proper parameters.
from dds_cloudapi_sdk.tasks import IVPTask
from dds_cloudapi_sdk.tasks import RectPrompt
from dds_cloudapi_sdk.tasks import LabelTypes

task = IVPTask(
    prompt_image_url=prompt_image_url,
    prompts=[RectPrompt(rect=[0.826842, 0.337368, 0.020000, 0.047368], is_positive=True)],
    infer_image_url=infer_image_url,
    infer_label_types=[LabelTypes.BBox, LabelTypes.Mask],  # infer both bbox and mask
)

# 4. Run the task and get the result.
client.run_task(task)

# 5. Parse the result.
from dds_cloudapi_sdk.tasks.ivp import TaskResult

result: TaskResult = task.result

mask_url = result.mask_url  # the image url with all masks drawn on
objects = result.objects  # the list of detected objects

for idx, obj in enumerate(objects):
    # get the detection score
    print(obj.score)  # 0.42

    # get the detection bbox
    print(obj.bbox)  # [635.0, 458.0, 704.0, 508.0]

    # get the detection mask, it's of RLE format
    print(obj.mask.counts)  # ]o`f08fa14M3L2O2M2O1O1O1O1N2O1N2O1N2N3M2O3L3M3N2M2N3N1N2O..
```



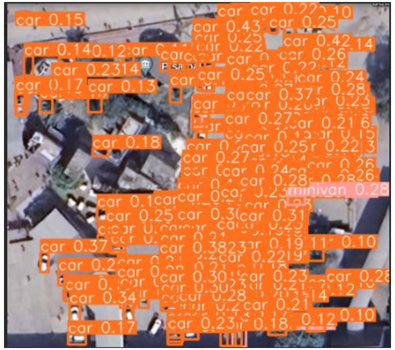
```
_XS79\m03M1O2000200000001000000000001000
(950, 950)
0.34
[760.0, 559.0, 779.0, 607.0]

[18] len(objects)

239
```

Predicts 239 vehicles

Training Yolov8 on custom dataset



```
[7]  mode=predict model='/content/gdrive/My Drive/iitm_2024/train9/weights/best.pt'  conf=0.1 source='/content/gdrive/My D

    Ultralytics YOLOv8.2.5 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
    YOLOv8n summary (fused): 168 layers, 3006818 parameters, 0 gradients, 8.1 GFLOPs

    image 1/1 /content/gdrive/My Drive/iitm_2024/data/test_images/test_5.png: 608x640 4 minivans, 252 cars, 93.7ms
    Speed: 5.7ms preprocess, 93.7ms inference, 2042.2ms postprocess per image at shape (1, 3, 608, 640)
    Results saved to runs/detect/predict3
    💡 Learn more at https://docs.ultralytics.com/modes/predict
```



Predicts 256 vehicles