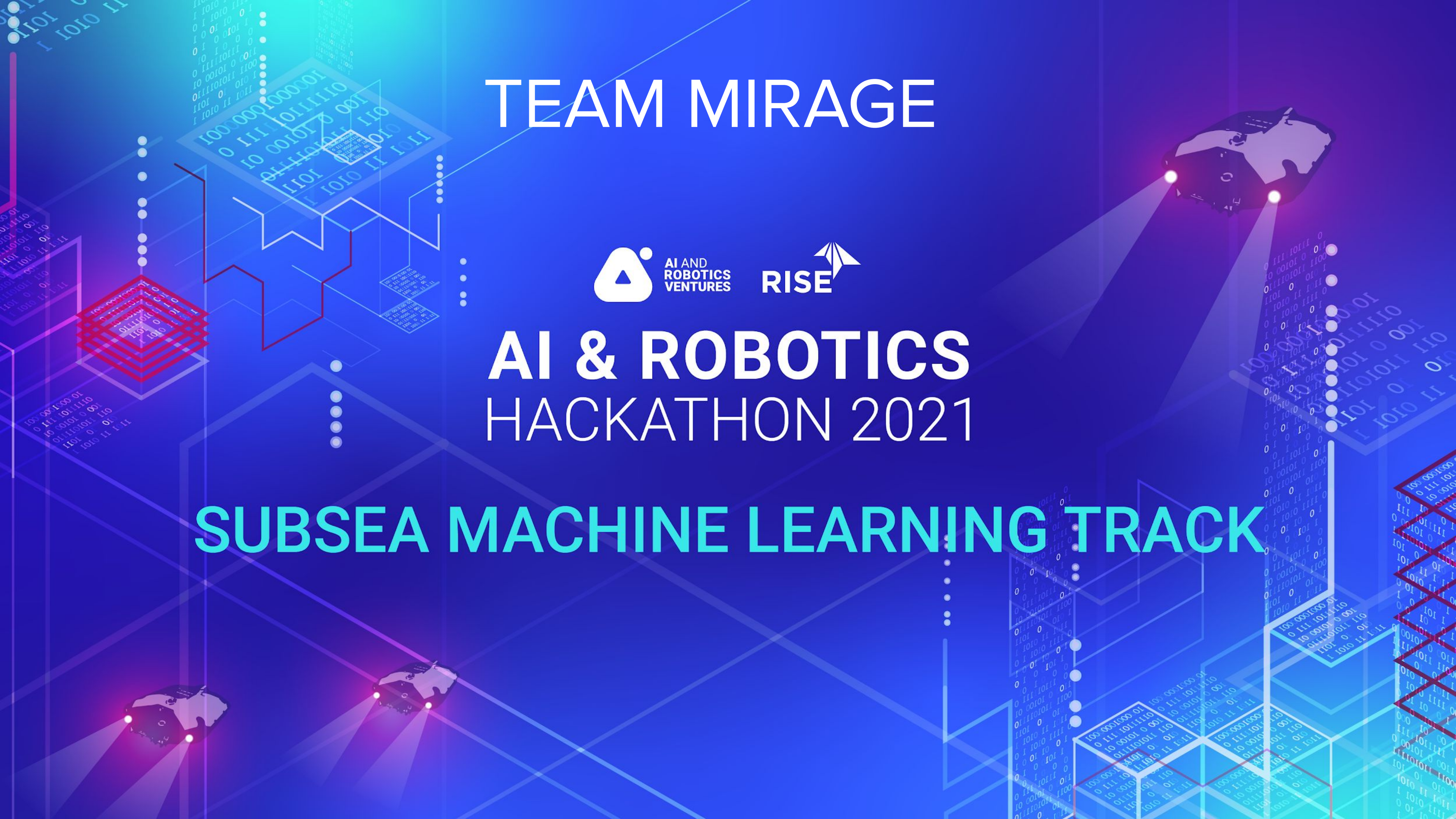


TEAM MIRAGE

AI AND ROBOTICS VENTURES

RISE

AI & ROBOTICS HACKATHON 2021

SUBSEA MACHINE LEARNING TRACK

AGENDA

Problem Statement

Approaches

Building Blocks

Model Selection

Synthetic Data Generation

Data augmentation

Training pipeline

Experiments

Demo and Results

Future improvements

Team Members

Appendix

PROBLEM STATEMENT

Given 3D CAD files and training images,

Generate synthetic data and develop 2D object detection model

Given Data

6 CAD files	**179 training images**	4 **Classes:**
		- 176 labels of pipe - 65 labels of anode - 38 labels of corner - 9 labels of flange

Challenges

- Learning from small data
- Low learning signals (images in training set are very similar)

AI & ROBOTICS
HACKATHON 2021

APPROACHES

AI AND ROBOTICS VENTURES

RISE

Main Focus

1. Generate **synthetic data** from 3D CAD files
2. Utilized **lots of image augmentations** during training

Training Steps

1. Train model using synthetic data
2. Finetune trained model on real data
3. Perform hyperparameter tuning

BUILDING BLOCKS

Data Preparation

- blender
- unity
- roboflow
- FiftyOne

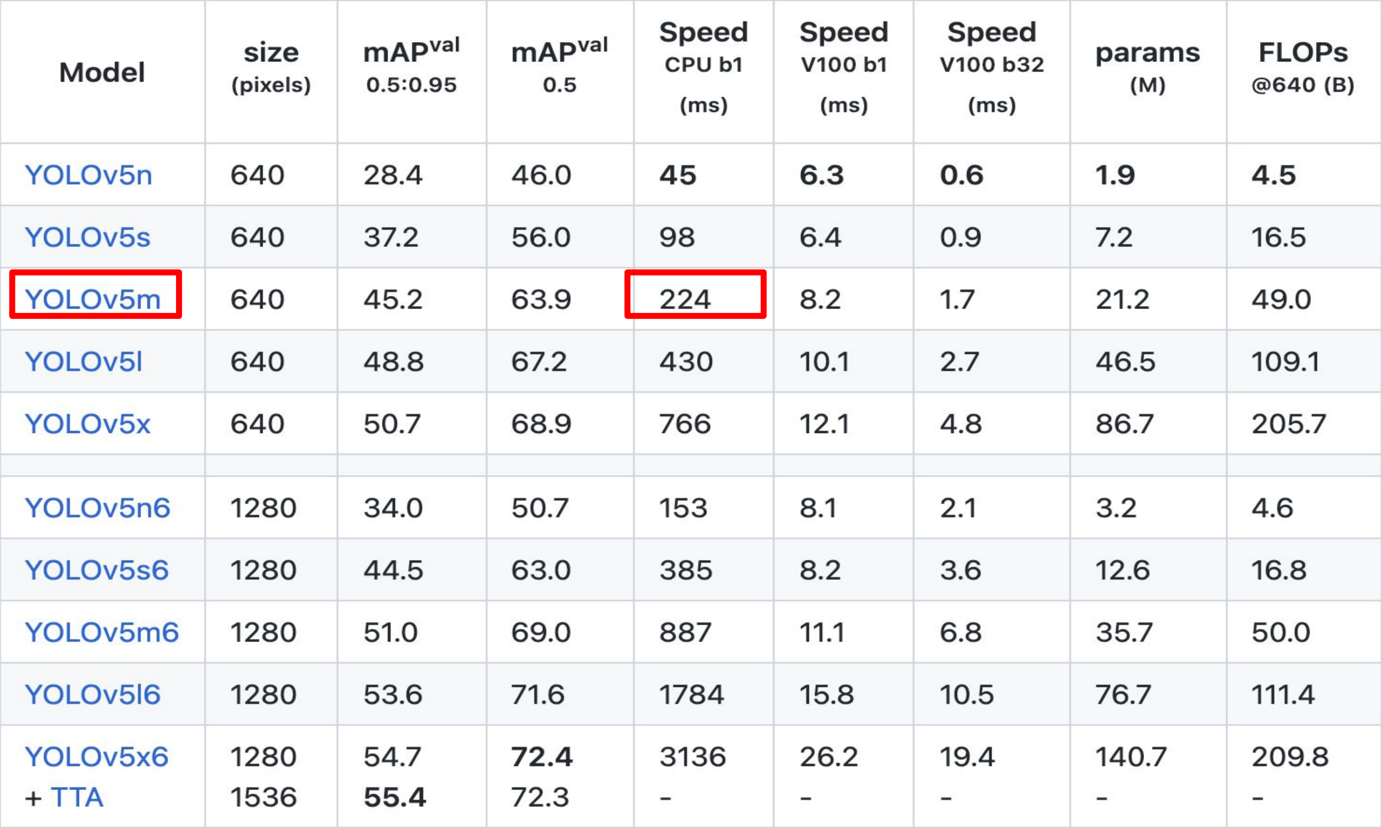
Model Training

- YOLOv5
- Weights & Biases
- PyTorch
- Albumentations
- Amazon EC2
- NVIDIA TESLA

MODEL SELECTION

Requirement: 0.7 second per image with CPU inference

YOLOv5m (40 mb)

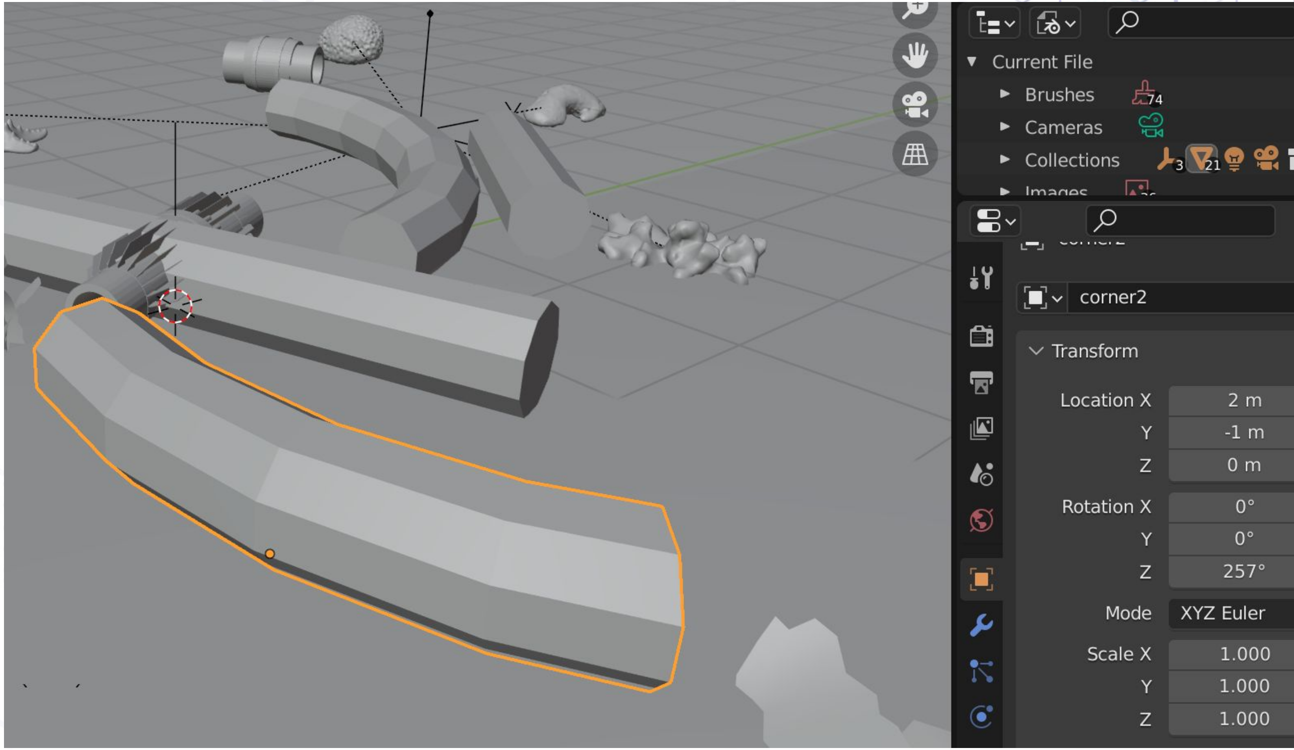


Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.4	46.0	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.2	56.0	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.2	63.9	224	8.2	1.7	21.2	49.0
YOLOv5l	640	48.8	67.2	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	34.0	50.7	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.5	63.0	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.0	69.0	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.6	71.6	1784	15.8	10.5	76.7	111.4
YOLOv5x6 + TTA	1280 1536	54.7 **55.4**	**72.4** 72.3	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

SYNTHETIC DATA GENERATION

Generating synthetic data from Blender

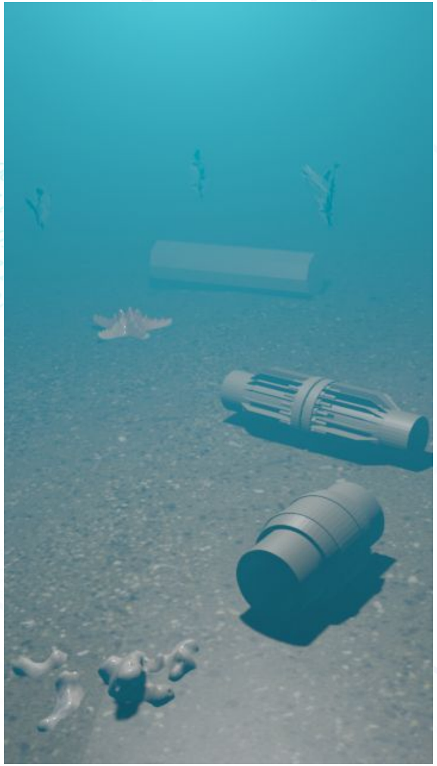
- 10K synthetic images generated
- Cut 3D Models for each class
- Create underwater scene
- Randomize object positions and rotations
- Script to generate bounding box

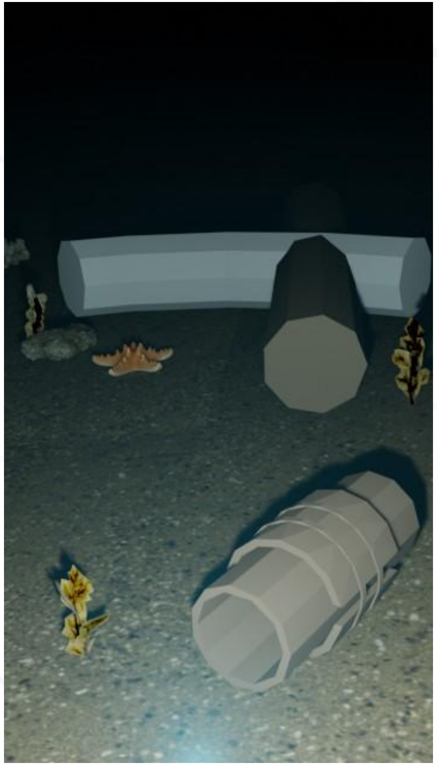


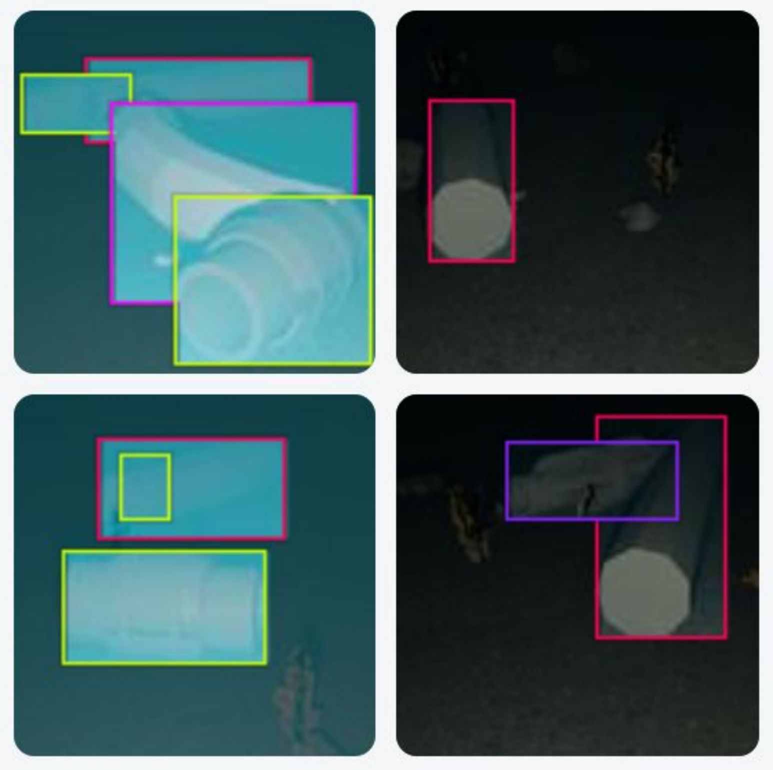
Location X 2 m
Y -1 m
Z 0 m
Rotation X 0°
Y 0°
Z 257°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000

SYNTHETIC IMAGES

Results from synthetic data generation from Blender







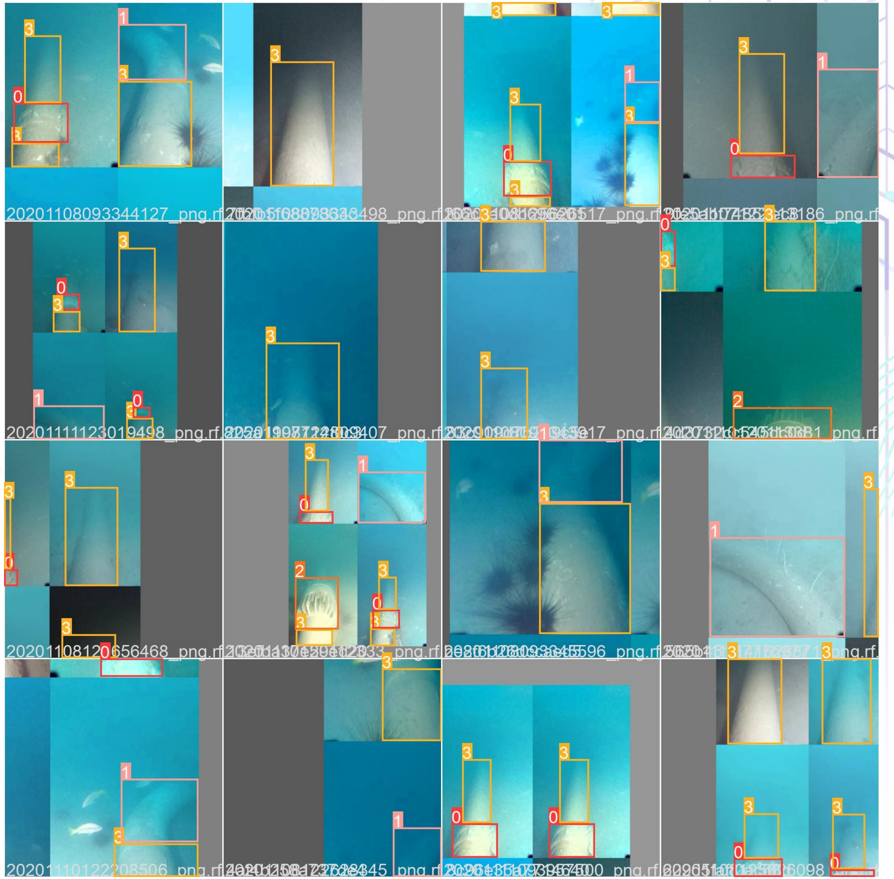
DATA AUGMENTATION

Image Augmentations

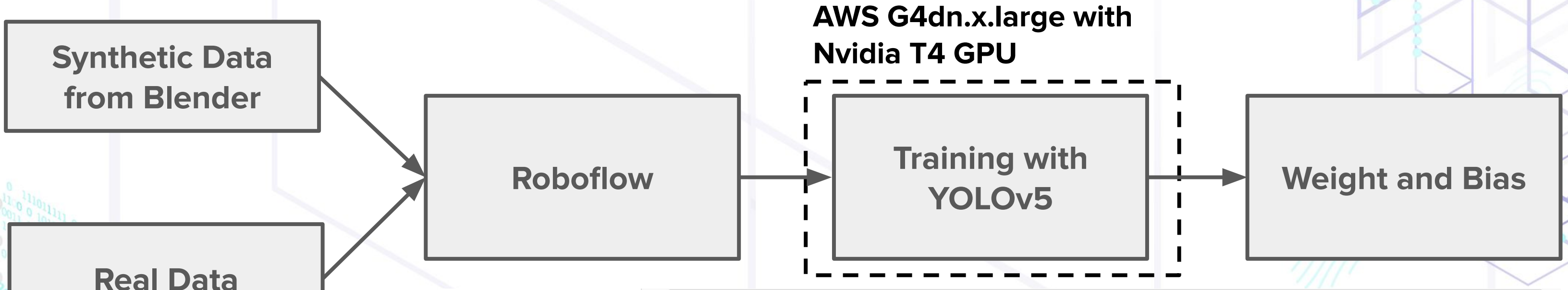
- Mosaic
- Random Brightness
- Random Contrast
- Random Flip
- Random Scale
- Random Crop

Others (did not improve performance)

- Blur
- Gaussian noise
- Random Fog
- Random Gamma
- Image Compression



TRAINING PIPELINE



Synthetic Data from Blender

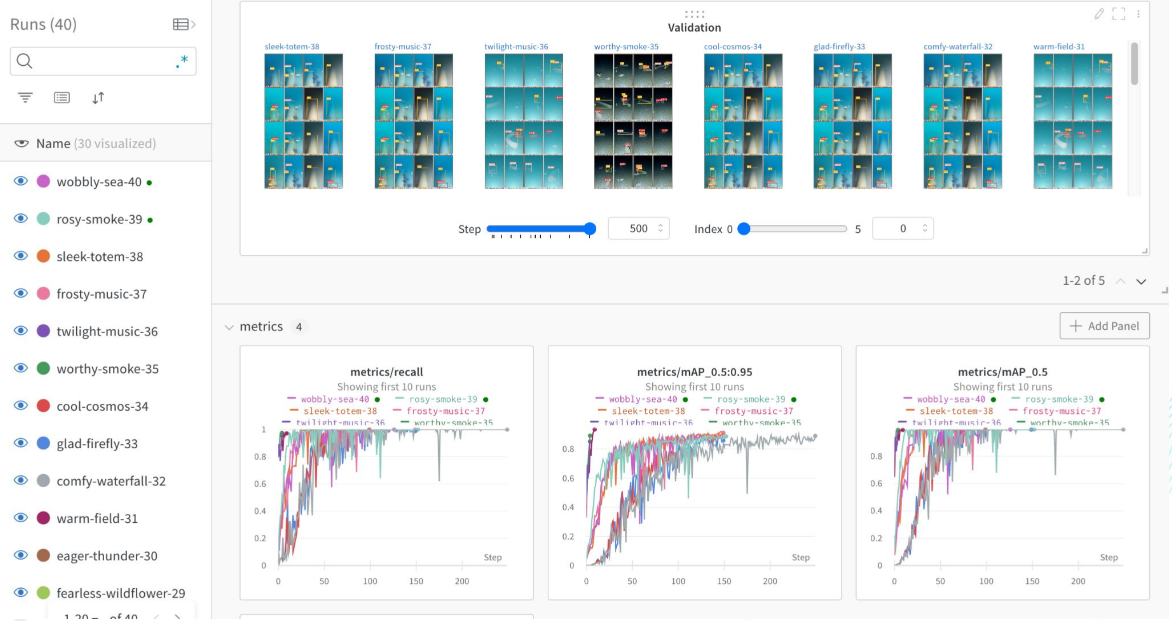
Real Data

Roboflow

AWS G4dn.x.large with Nvidia T4 GPU

Training with YOLOv5

Weight and Bias



Runs (40)

Name (30 visualized)

- wobbly-sea-40 ●
- rosy-smoke-39 ●
- sleek-totem-38
- frosty-music-37
- twilight-music-36
- worthy-smoke-35
- cool-cosmos-34
- glad-firefly-33
- comfy-waterfall-32
- warm-field-31
- eager-thunder-30
- fearless-wildflower-29

1-20 – of 40

Validation

Step 500

Index 0 5 0

1-2 of 5

metrics 4

metrics/recall

Showing first 10 runs

metrics/mAP_0.5:0.95

Showing first 10 runs

metrics/mAP_0.5

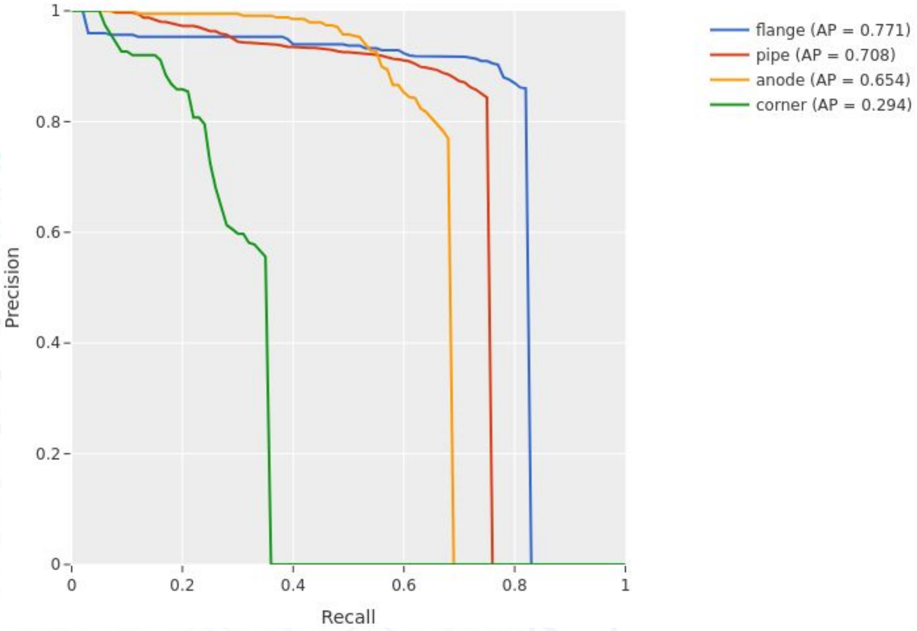
Showing first 10 runs

EXPERIMENTS

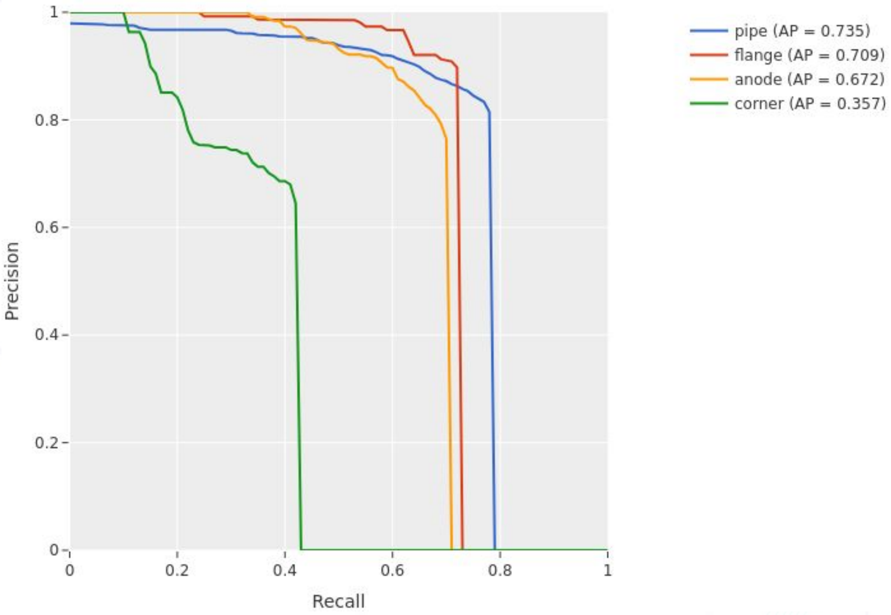
Train with **data augmentation** on only real data (baseline)	0.61
Train on **synthetic data from Unity** and finetune on real data	0.38
Train with **more augmentations** from Albumentation with only real data	0.55
Perform **hyperparameter tuning** using genetic algorithm and train using obtained hyperparameter on only real data	0.55
Train **larger model** with only real data (YOLOv5 Large)	0.53
Train on **synthetic data from Blender** and finetune on real data	**0.6**

Results

Train with real data only

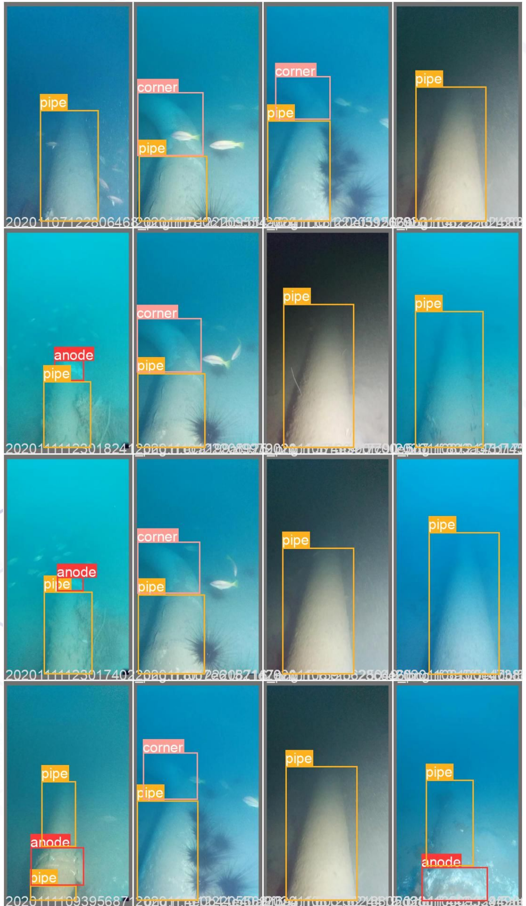


Train with Synthetic and real data

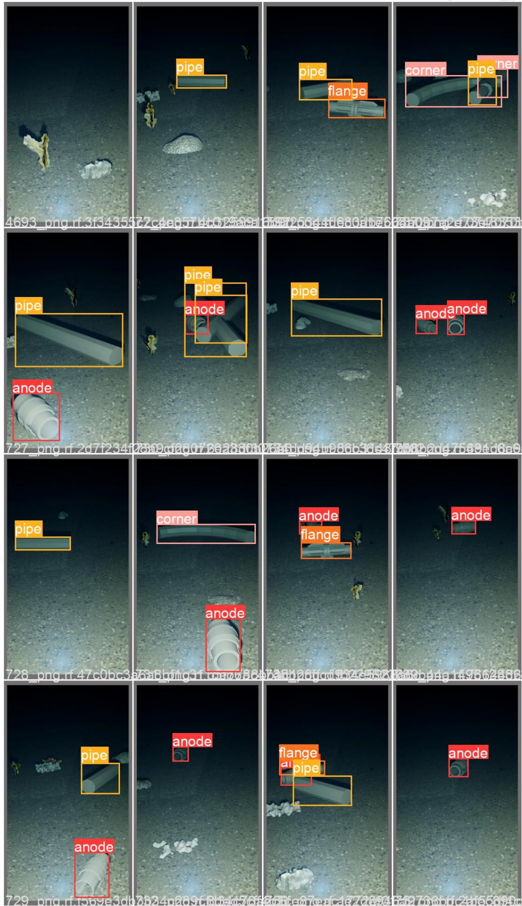


DEMO

Prediction on real data (validation set)



Prediction on synthetic data (validation set)



FUTURE IMPROVEMENT

1. **Generate synthetic images more similar to real world settings**
   - More realistic scenes
   - Need professional 3D artists
2. **Domain randomization**
   - Add more random background objects in synthetic images
3. **Try more image augmentations**
4. **Optimize model inference speed in deployment**
5. **Find the right number of epochs.**
6. **Try other models/ frameworks**
7. **Try separate model for each class.**
8. **Combine real and synthetic data into one dataset**

TEAM MEMBERS

Vasin Srisupavanich

https://www.linkedin.com/in/markvasin/

Deeprom Somkiadcharoen

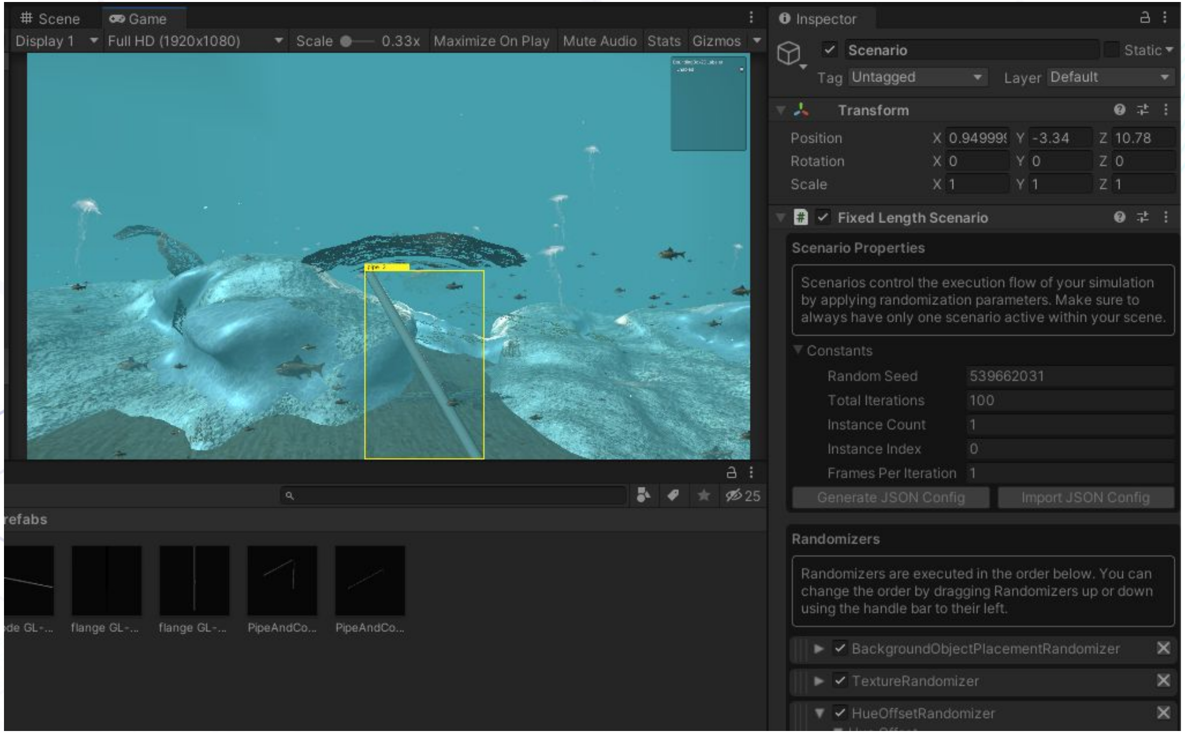
https://www.linkedin.com/in/dprmsmk/

Rachapong Chirarattananon

https://www.linkedin.com/in/paulxiep/

Appendix

Generating synthetic data from Unity3D



☑ Scenario □ Static

Tag Untagged Layer Default

Transform

Position	X 0.94999	Y -3.34	Z 10.78
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ Fixed Length Scenario

Scenario Properties

Scenarios control the execution flow of your simulation by applying randomization parameters. Make sure to always have only one scenario active within your scene.

Constants

Random Seed	539662031
Total Iterations	100
Instance Count	1
Instance Index	0
Frames Per Iteration	1

Randomizers

Randomizers are executed in the order below. You can change the order by dragging Randomizers up or down using the handle bar to their left.

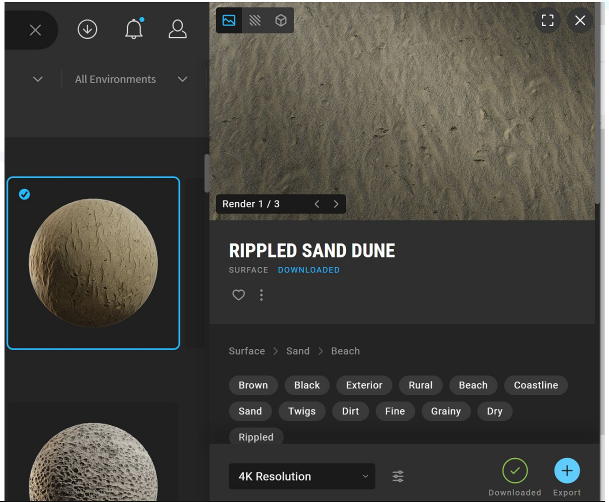
- ☑ BackgroundObjectPlacementRandomizer
- ☑ TextureRandomizer
- ☑ HueOffsetRandomizer

Data Synthesis (Blender)

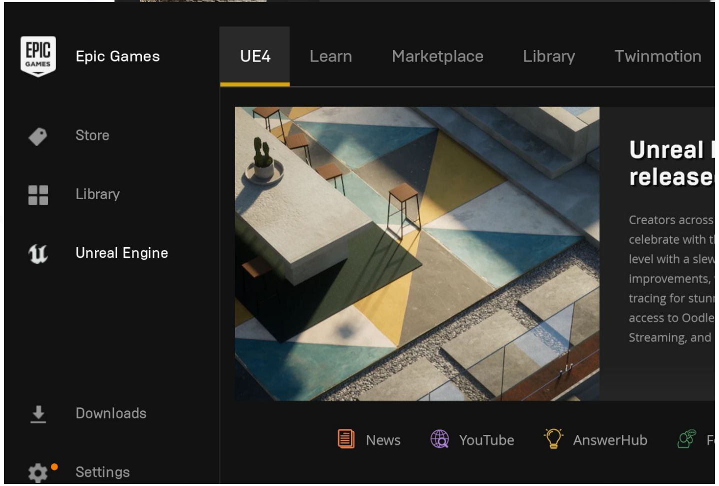
- We found this https://www.youtube.com/watch?v=I2B-x3J0W4I

Data Synthesis (Blender)

- Quixel Bridge

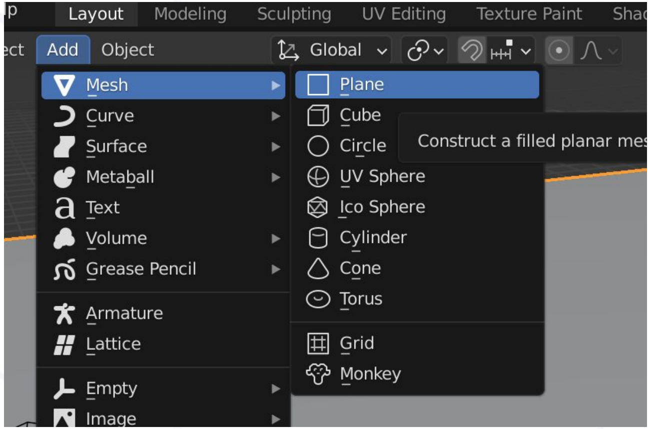


- Install Epic Games!?



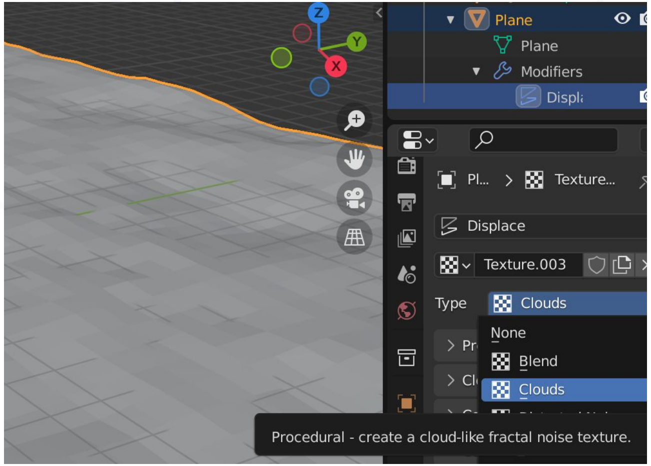
Data Synthesis (Blender)

- Make floor



Construct a filled planar mes

- Make it wavy



Displace

Texture.003

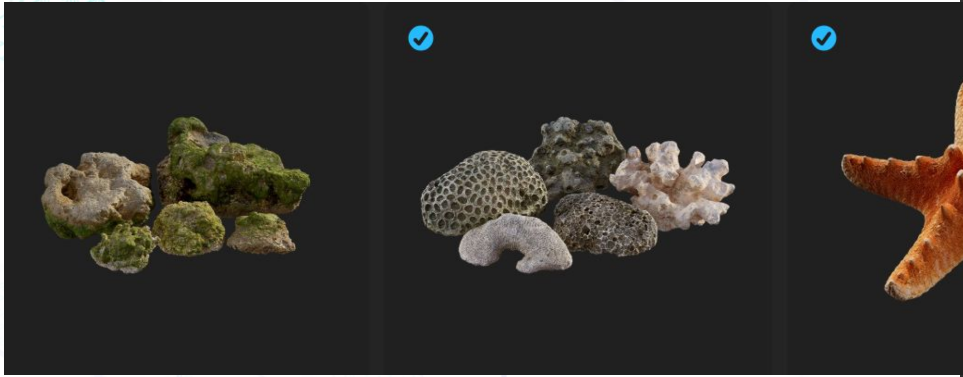
Type Clouds

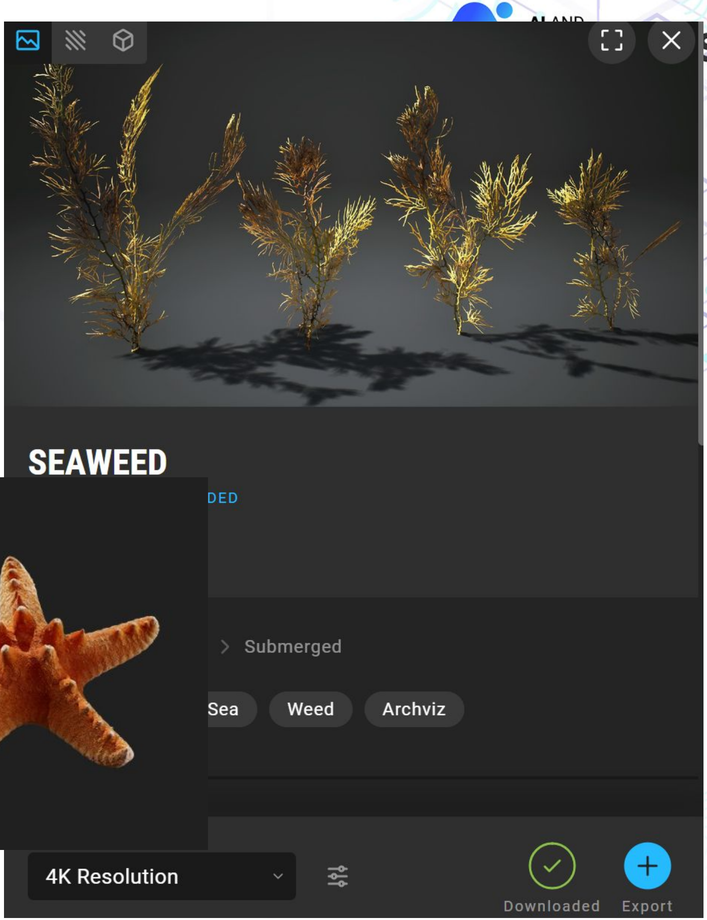
Procedural - create a cloud-like fractal noise texture.

AI & ROBOTICS
HACKATHON 2021

Data Synthesis (Blender)

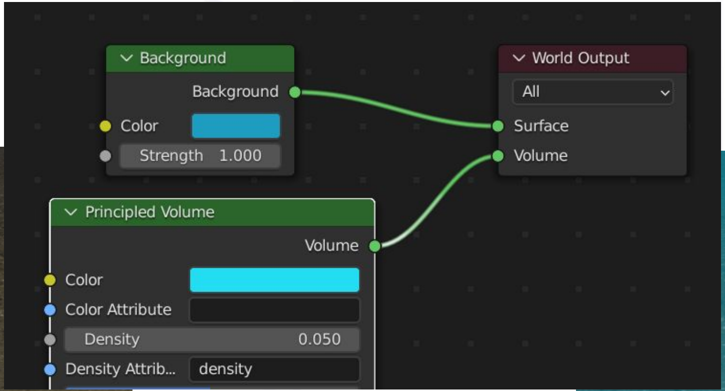
- Import sand texture to seafloor
- Import seabed rocks, seaweed, starfish
- Too bad there's no fish model...





Data Synthesis (Blender)

- Add Shader



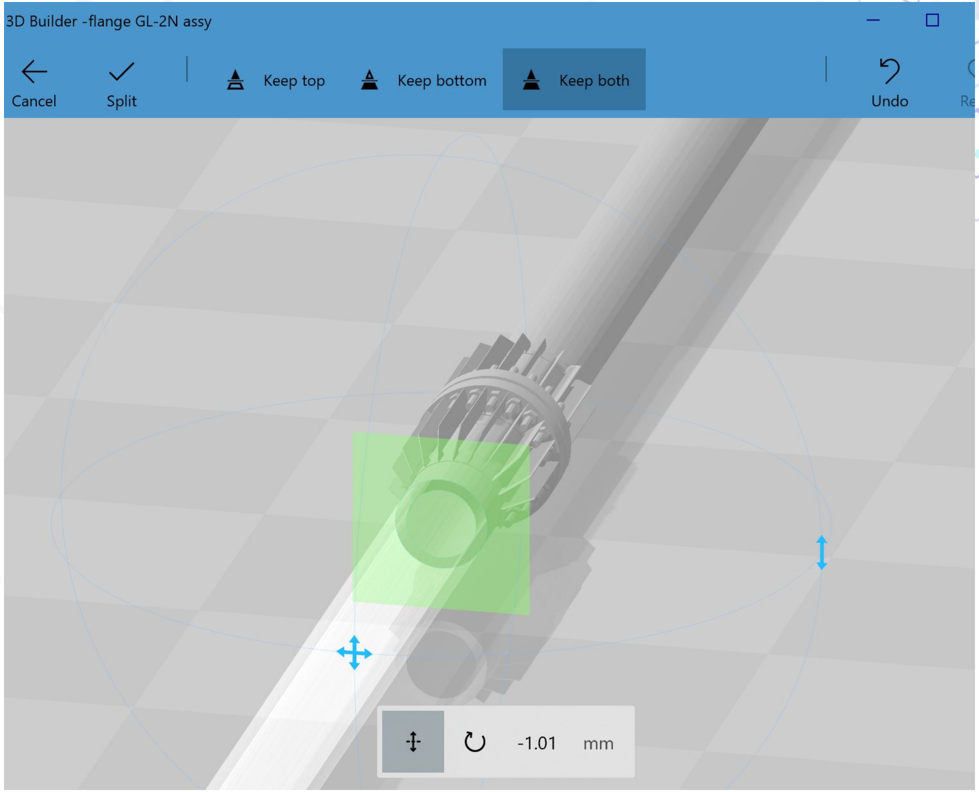
Background
Background
Color
Strength 1.000

World Output
All
Surface
Volume

Principled Volume
Volume
Color
Color Attribute
Density 0.050
Density Attrib... density

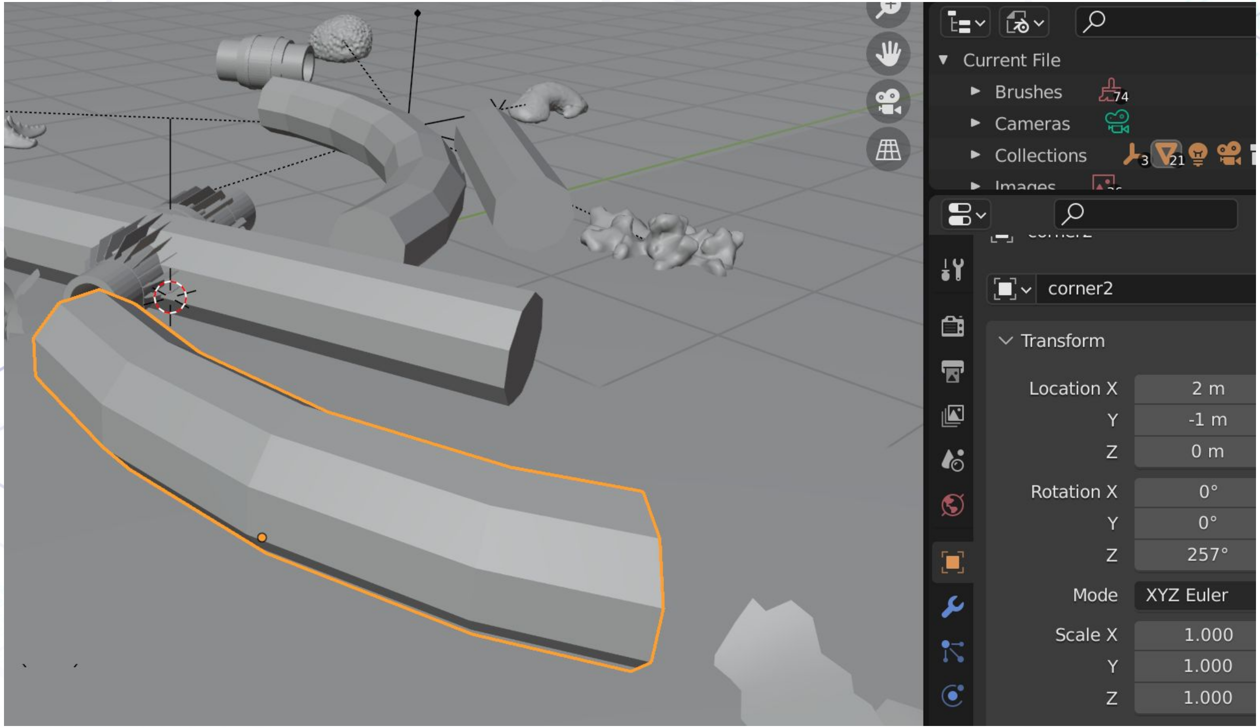
Data Synthesis (Blender)

- Cut 3d models in 3d Builder



Data Synthesis (Blender)

- Import to Blender, find the right resize ratio. (don’t forget to change texture!)



corner2

Transform

Location X 2 m
Y -1 m
Z 0 m

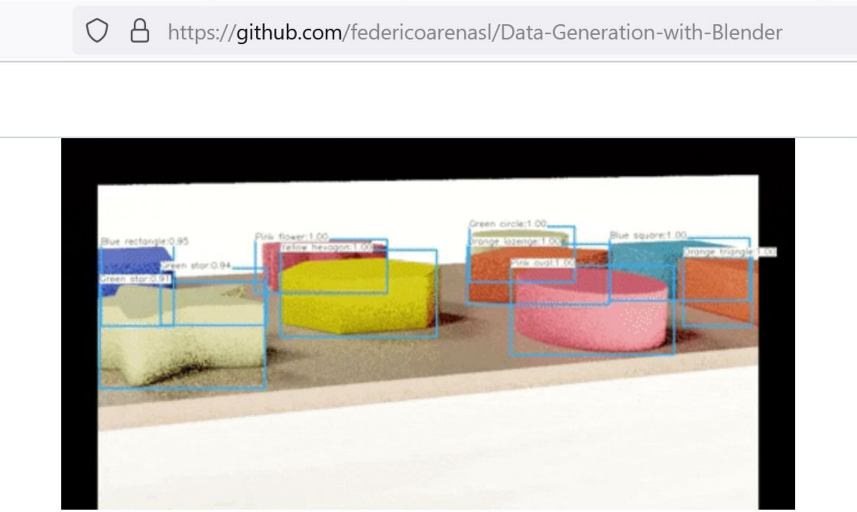
Rotation X 0°
Y 0°
Z 257°

Mode XYZ Euler

Scale X 1.000
Y 1.000
Z 1.000

Data Synthesis (Blender)

- The **HARD** part: Script!
  - Need to take 1000s of different photos
  - With auto-generated bounding boxes!
- Starter code (doesn’t work, we had to fix it)



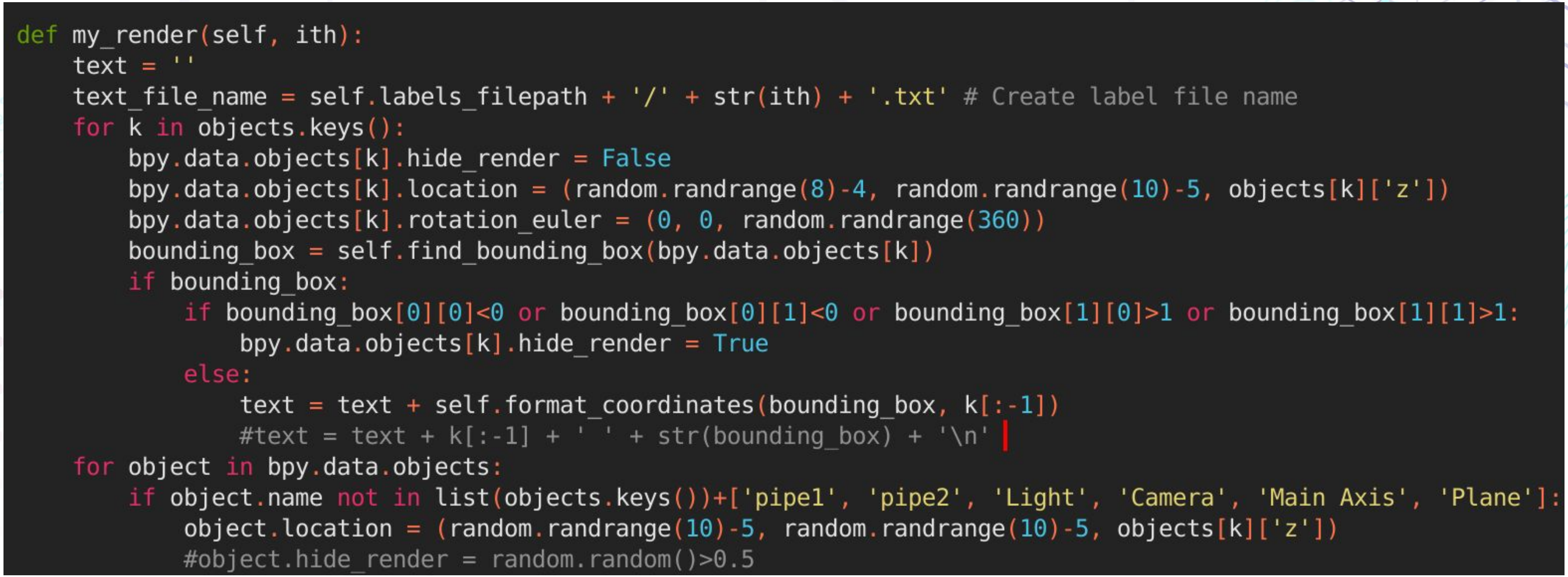
https://github.com/federicoarenasl/Data-Generation-with-Blender

Data Synthesis (Blender)

- Randomize Camera position/rotation.
  - Have to avoid invalid angles
  - Redundant with randomize Object positions
- Randomize Light position/rotation/illuminance
  - Unpredictable due to water shader
- Randomize Object positions
  - Most practical to do.
  - So fix camera and light!

Data Synthesis (Blender)

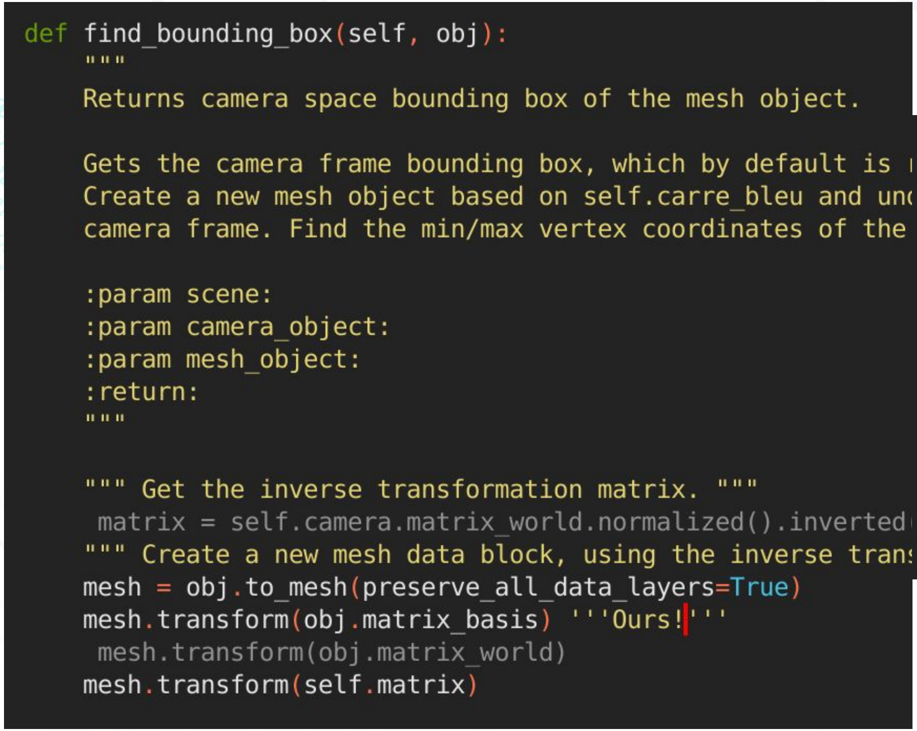
- Replace main rendering loop with our own.



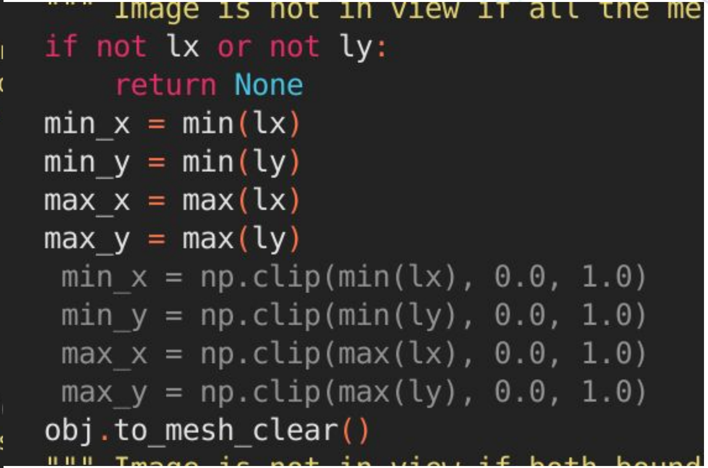
```
def my_render(self, ith):
    text = ''
    text_file_name = self.labels_filepath + '/' + str(ith) + '.txt' # Create label file name
    for k in objects.keys():
        bpy.data.objects[k].hide_render = False
        bpy.data.objects[k].location = (random.randrange(8)-4, random.randrange(10)-5, objects[k]['z'])
        bpy.data.objects[k].rotation_euler = (0, 0, random.randrange(360))
        bounding_box = self.find_bounding_box(bpy.data.objects[k])
        if bounding_box:
            if bounding_box[0][0]<0 or bounding_box[0][1]<0 or bounding_box[1][0]>1 or bounding_box[1][1]>1:
                bpy.data.objects[k].hide_render = True
            else:
                text = text + self.format_coordinates(bounding_box, k[:-1])
                #text = text + k[:-1] + ' ' + str(bounding_box) + '\n'
    for object in bpy.data.objects:
        if object.name not in list(objects.keys())+['pipe1', 'pipe2', 'Light', 'Camera', 'Main Axis', 'Plane']:
            object.location = (random.randrange(10)-5, random.randrange(10)-5, objects[k]['z'])
            #object.hide_render = random.random()>0.5
```

Data Synthesis (Blender)

- Fix the find_bounding_box()



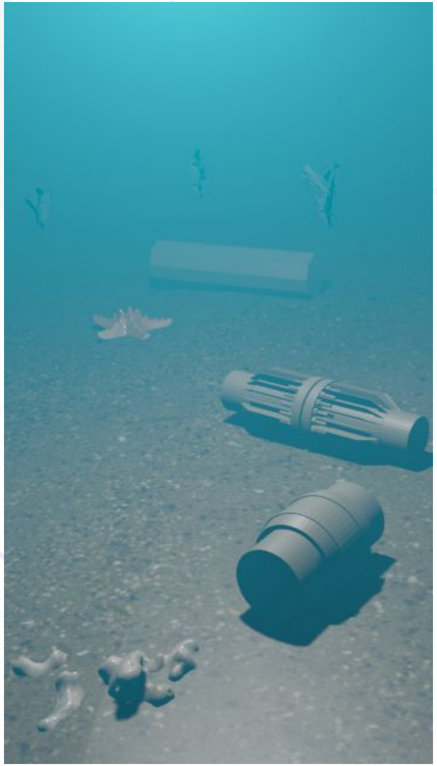
```
def find_bounding_box(self, obj):
    """
    Returns camera space bounding box of the mesh object.

    Gets the camera frame bounding box, which by default is r
    Create a new mesh object based on self.carre_bleu and unc
    camera frame. Find the min/max vertex coordinates of the

    :param scene:
    :param camera_object:
    :param mesh_object:
    :return:
    """

    """ Get the inverse transformation matrix. """
     matrix = self.camera.matrix_world.normalized().inverted(
    """ Create a new mesh data block, using the inverse trans
    mesh = obj.to_mesh(preserve_all_data_layers=True)
    mesh.transform(obj.matrix_basis) '''Ours!'''
     mesh.transform(obj.matrix_world)
    mesh.transform(self.matrix)
```

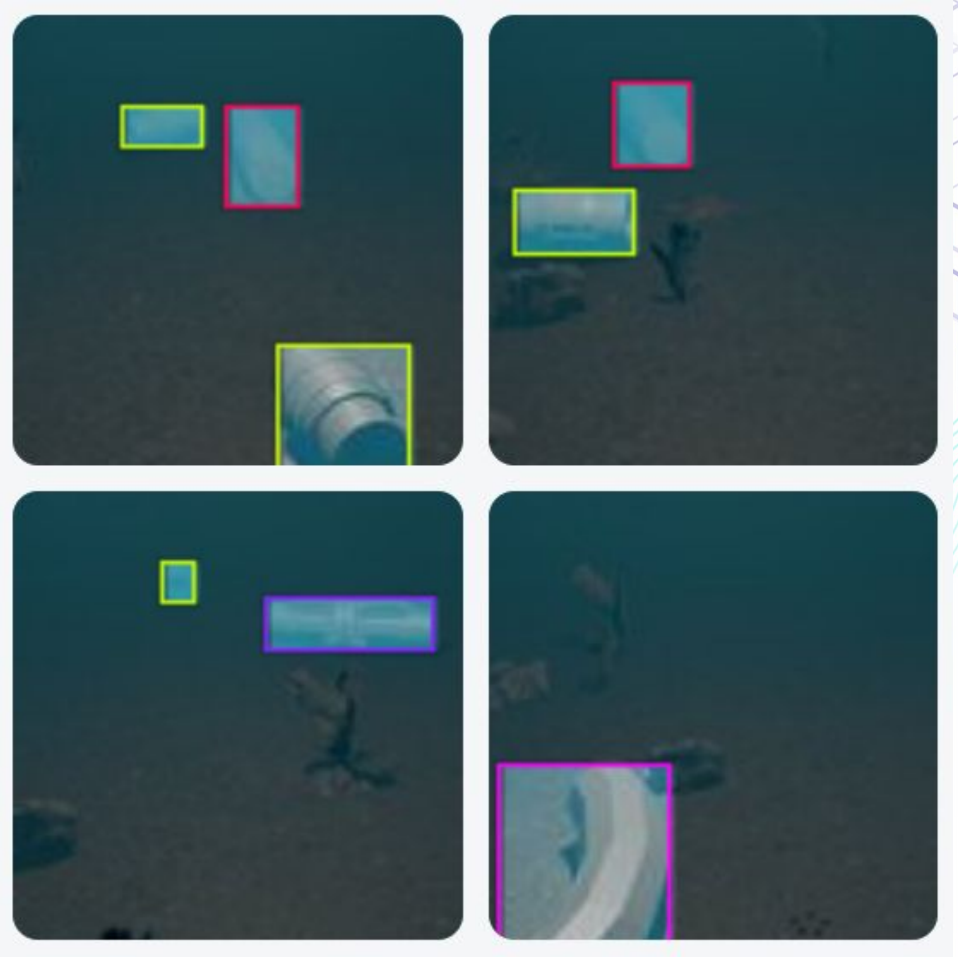


```
""" Image is not in view if all the me
if not lx or not ly:
    return None
min_x = min(lx)
min_y = min(ly)
max_x = max(lx)
max_y = max(ly)
 min_x = np.clip(min(lx), 0.0, 1.0)
 min_y = np.clip(min(ly), 0.0, 1.0)
 max_x = np.clip(max(lx), 0.0, 1.0)
 max_y = np.clip(max(ly), 0.0, 1.0)
obj.to_mesh_clear()
""" Image is not in view if both bound
```

Data Synthesis (Blender)

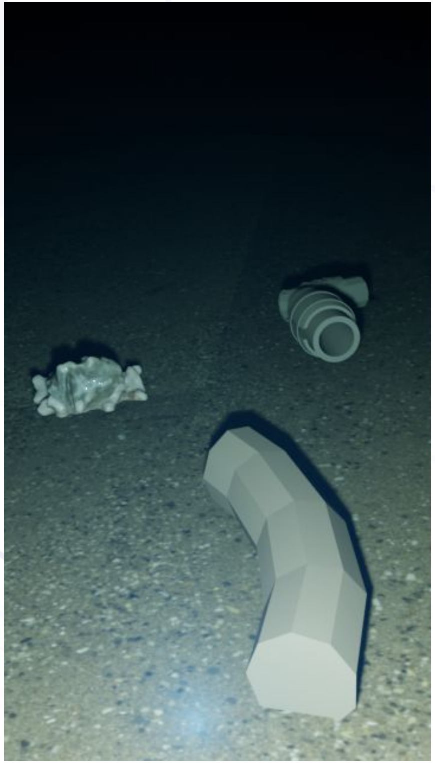
- Sample generated results (day)

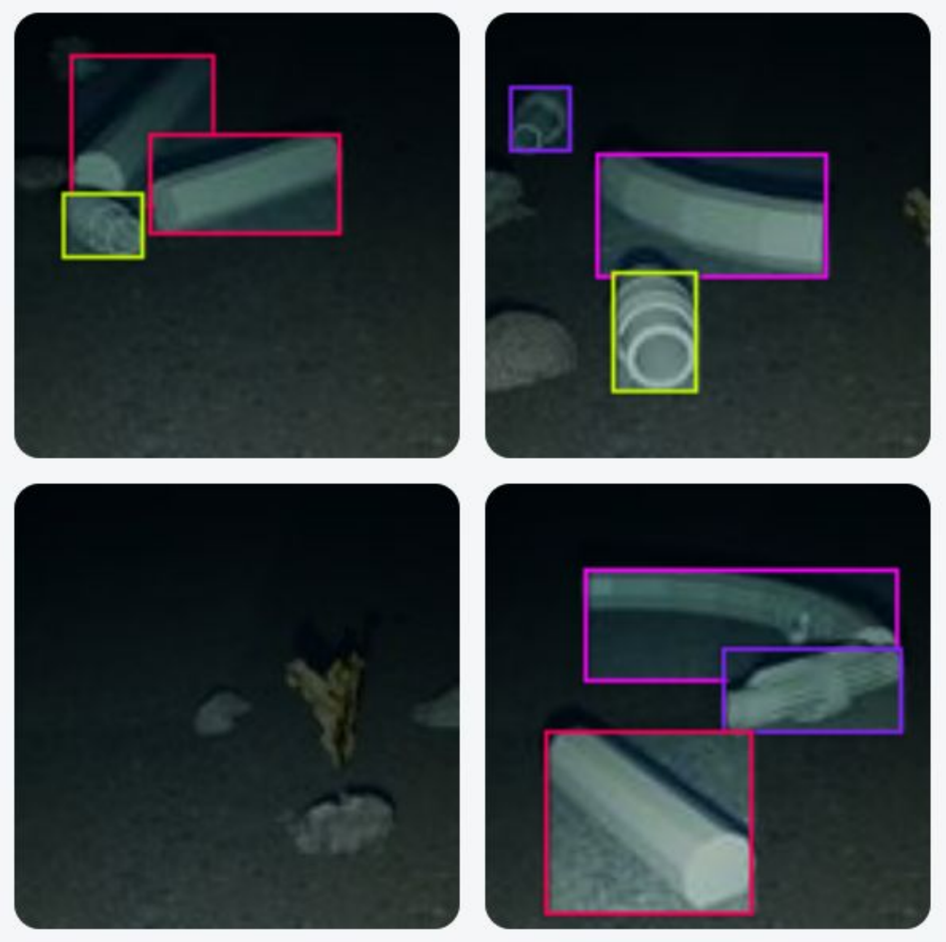


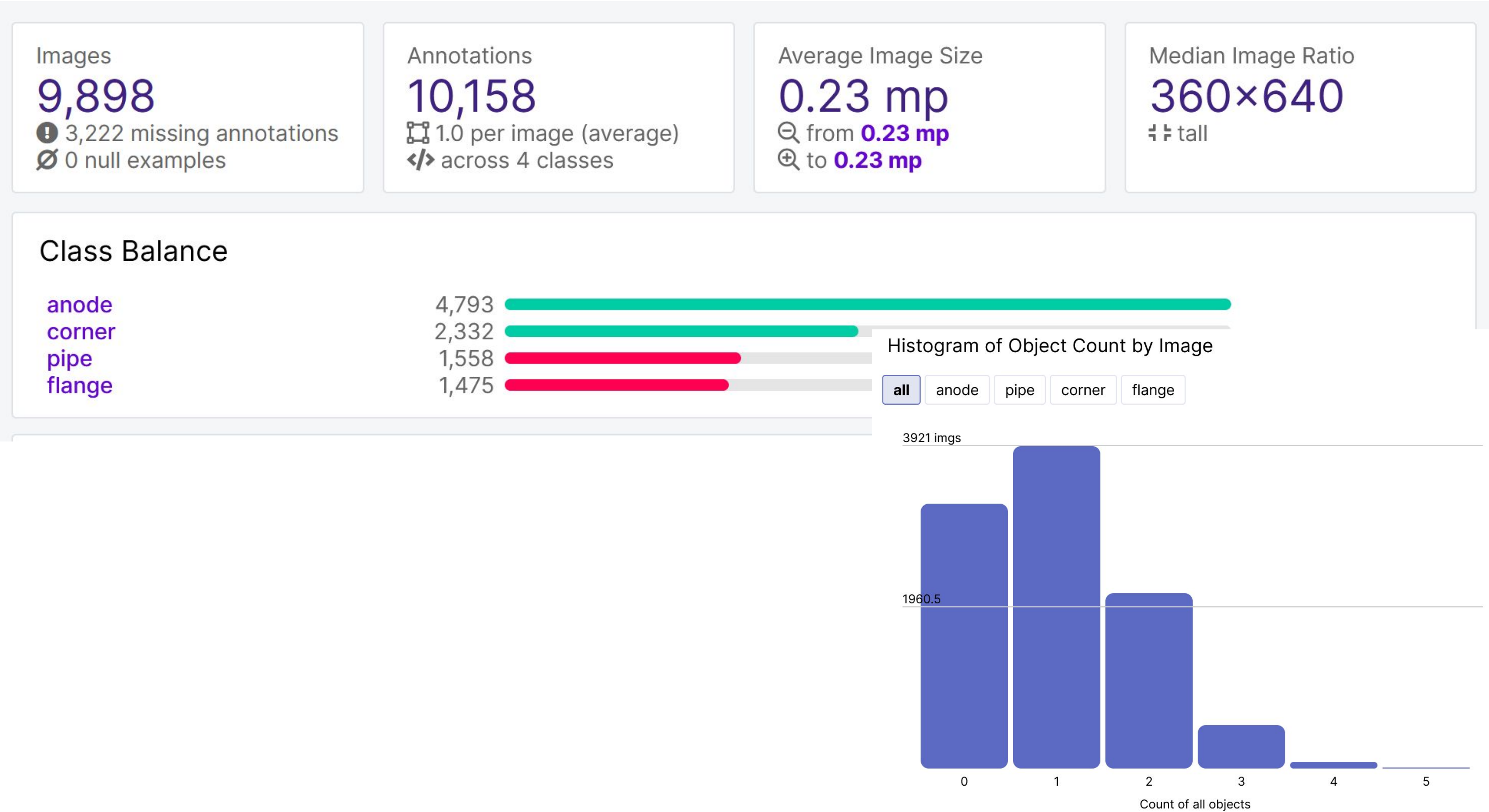


Data Synthesis (Blender)

- Sample generated results (night)







Images

9,898

3,222 missing annotations

0 null examples

Annotations

10,158

1.0 per image (average)

across 4 classes

Average Image Size

0.23 mp

from **0.23 mp**

to **0.23 mp**

Median Image Ratio

360×640

tall

Class Balance

anode	4,793
corner	2,332
pipe	1,558
flange	1,475

Histogram of Object Count by Image