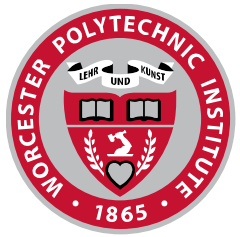


WPI

RBE 596 -- Using 3D Semantic Instance Segmentation for Robotics Pick-and-Place

Yihao (Charles) Cai

Table of Content

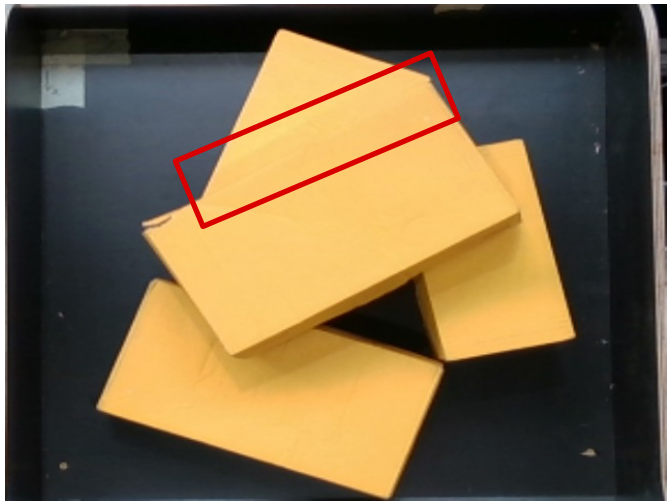
- Background
- Dataset & Experiment
- Model Analysis
- Summary

Background

Background

- **Pick-and-Place** = pick items in cluttered environment (tote)
- **Procedure** = Object Detection + Item Picking
- **Object Detection** =
  - ➢ **Solution** = using RGB-D based approach to produce affordance-map (object-agnostic), then generate pick-points for item-picking
  - ➢ **Limitation** = cannot segment objects well in some cases (loss of depth)
- **New Approach** = using 3D Semantics Instance Segmentation

Cluttered items



RGB Image

Affordance Map

Pick-and-place

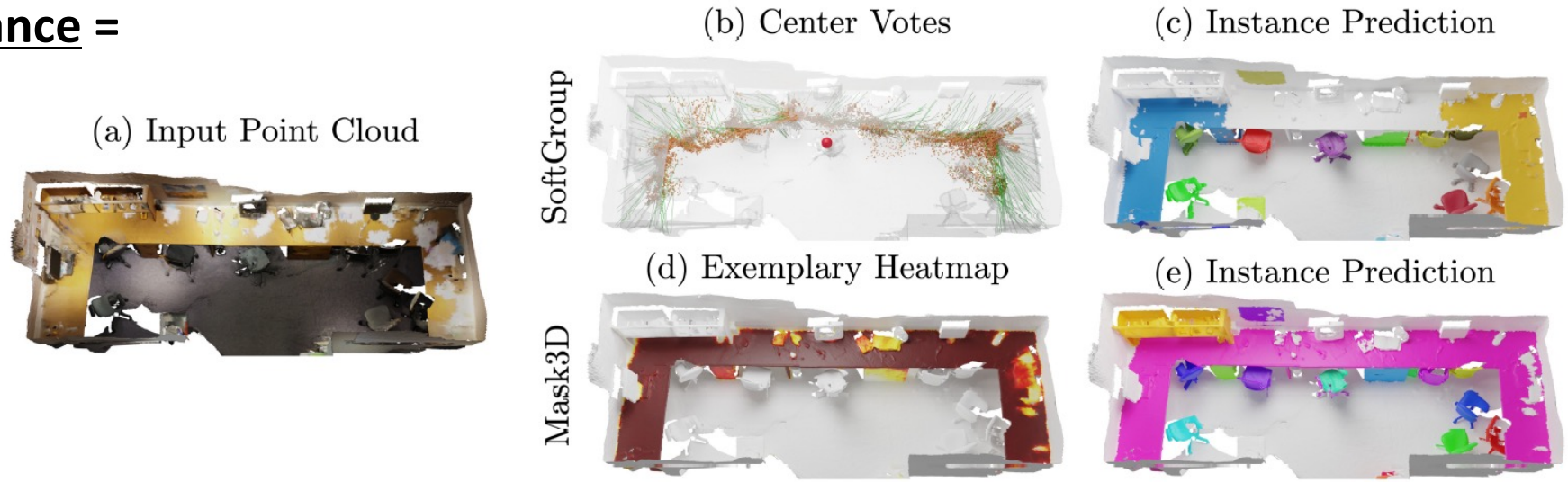
3D Semantic Instance Segmentation

- **Neural Network (Model)** = Mask3D (state-of-the-art algorithm in 3DIS)
  - **Property** = the first Transformer-based model for 3DIS
  - **Metrics** = use IoU (Intersection over Union) and make thresholding
  - **Test Datasets** = Scannet/Scannet200/S3DIS/STPLS3D (benchmark data)
  - **Data Features** = Depth(x,y,z) + RGB + Normals + SegmentID + Class(Label) + InstanceID

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

X	Y	Z	Red	Green	Blue	Norm_X	Norm_Y	Norm_Z	SegID	Class	ID
-15.219	3.727	2.658	210	178	153	1	1	1	1	2	3

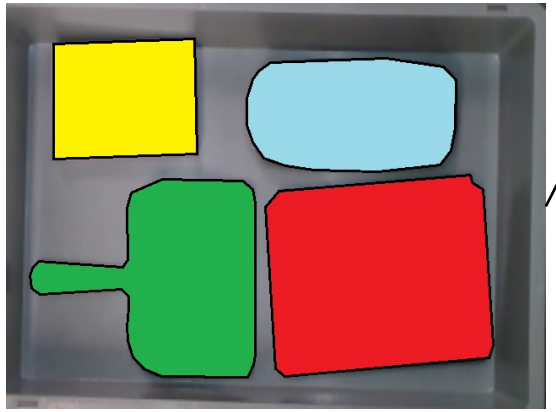
- **Performance** =



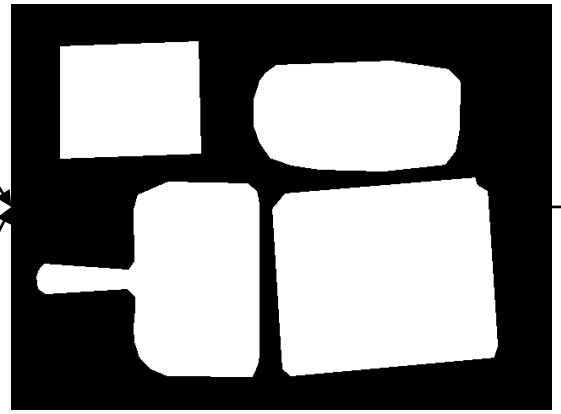
Dataset & Experiment

Point Cloud Label Pipeline

- **Approach** = generate mask for PCD labels by drawing RGB cover
- **Pipeline** =



Masks



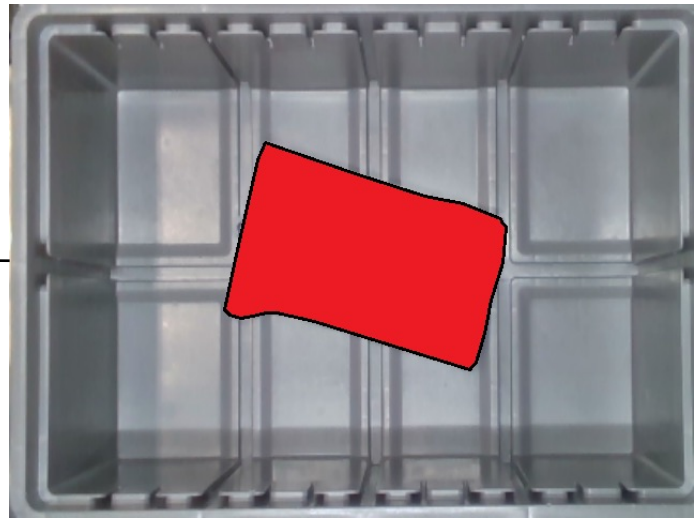
RGB

Depth

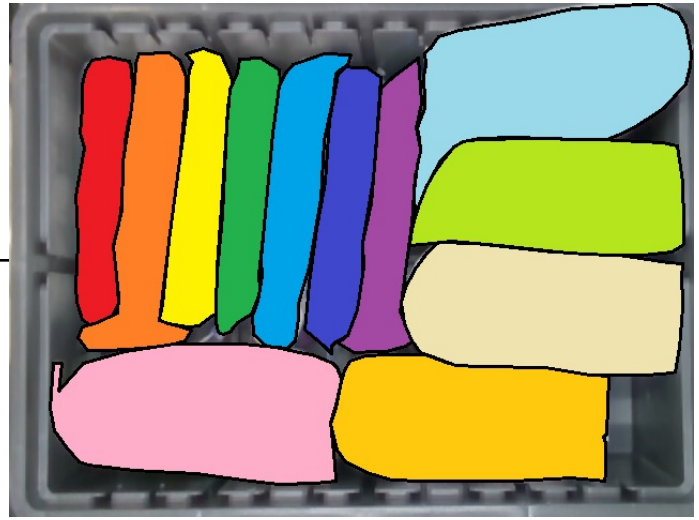
PCD with Label

- **<u>Limitation</u> =** label effort depends on degree of clutter in an image

Simple



Complex



– Data Label

- **Pure Color Item**

White Box

Piab Box

Pencil Box

Ipad Cover

Dustpan

Envelope

- **Amazon Envelope**

Envelope_bm3

Envelope_bm4

Envelope_bm5

Envelope_big

Envelope_small

– Data Label

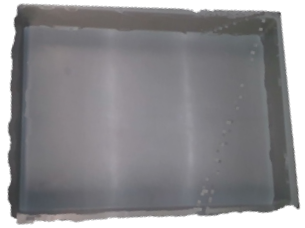
- **Reflective Materials**

Thumbtack

White Cylinder

Tablet Strip

Orange Cylinder



Flat Tote

- **Colorful Boxes**

Aspercreme Box

Aspirin Box

Bandage Box

Vicks Box

Walgreens Box

- **Fashion Clothes**

Hanes

Socks

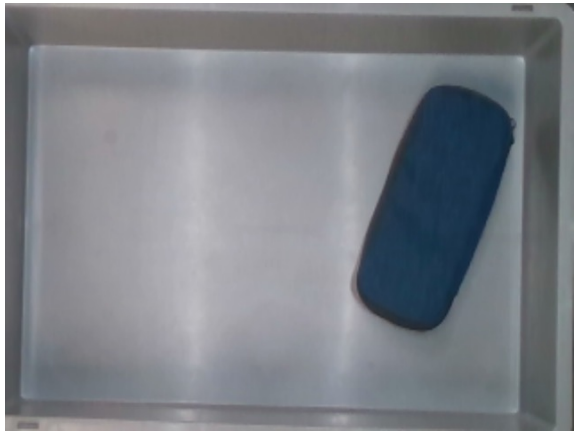
Sweater

Black Jacket

Red_hm

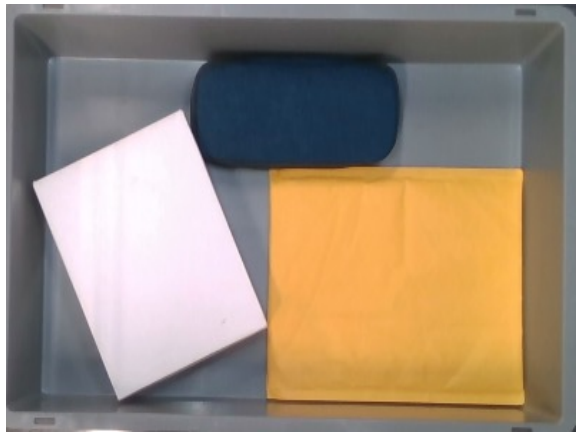
– Scenario Types

- **Single Item**

- **Clutter** (No occlusion)






– Scenario Types

- **Overlap** (with occlusion)


- **Instance** (objects with same type)




– Test

Dataset			Test Scenarios		Training (7:2:1)		
Name	Label	Feature	Type	Data Scale	Time Cost	Accuracy	Inference
Pure Color Items	white box pencil box piab box ipad cover envelope dustpan flat_tote	All items have pure color on its surface	Single-item	60	1.75h -> 300 epoch	100.00%	3 sec
			Clutter	60 x 4 = 240	17.5h -> 500 epoch	99.40%	4 sec
			Overlap	60 x 4 = 240	16h -> 500 epoch	98.30%	4 sec
			Instance	60 x 4 = 240	16h -> 500 epoch	99.00%	4 - 5 sec
			Clutter + Overlap + Instance	180 x 4 = 720	15h -> 300 epoch	98.00%	5 sec
Amazon Envelopes	envelope_bm3 envelope_bm4 envelope_bm5 envelope_big envelope_small flat_tote	Items have similar shape and color on surfaces	Clutter + Overlap	Data Augmentation 99 x 4 = 396	21h -> 400 epoch	99.60%	4 - 5 sec
Reflective Materials	thumbtack white cylinder orange cylinder tablet strip flat_tote	Items surfaces are either reflective or translucent	Clutter + Overlap + Instance	113 x 4 = 452	36h -> 600 epoch	84.00%	2 - 3 sec
Colorful Boxes	aspercreme box aspirin box bandage box vicks box walgreens box uneven_tote	All items are with colorful cover on its surface	Clutter + Overlap + Instance	197 x 4 = 788	40h -> 400 epoch	94.70%	3 sec
Fashion Clothes	hanes socks sweater black jacket red hm uneven_tote	Items with reflective surfaces and subtle deformation	Clutter + Instance	170 x 4 = 760	48h -> 600 epoch	92.10%	2 - 3 sec
			Clutter + Overlap + Instance	200 x 4 = 800	55h -> 600 epoch	92.70%	2 - 3 sec

Q&A

Experiment Result

Pure Color Items

– Inference Result (Clutter)

- **Scene** = PiabBox + IPad Cover + WhiteBox

RGB Point Cloud

2-3s (AMAX)

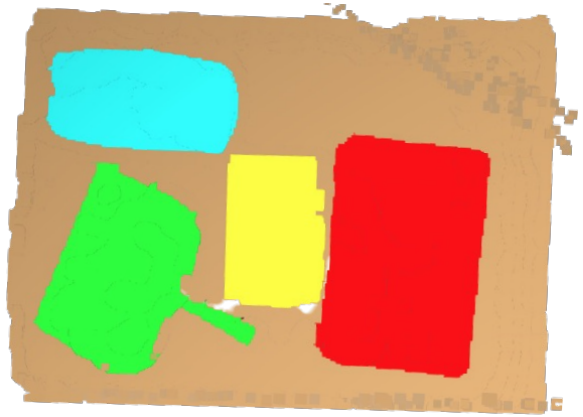
Semantic Segmentation

Instance Segmentation

- **Scene** = Dustpan + PiabBox + IPad Cover + PencilBox

RGB Point Cloud

2-3s (AMAX)



Semantic Segmentation

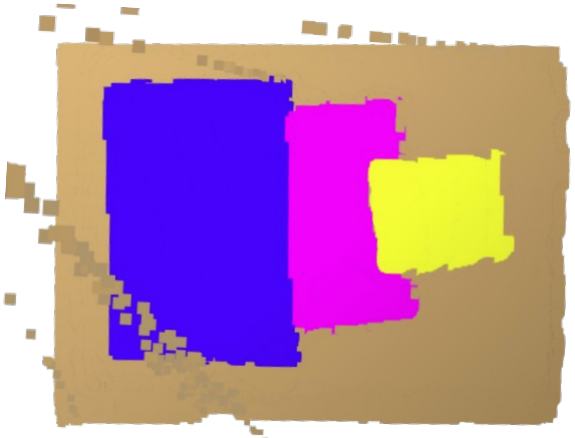
Instance Segmentation

– Inference Result (Overlap)

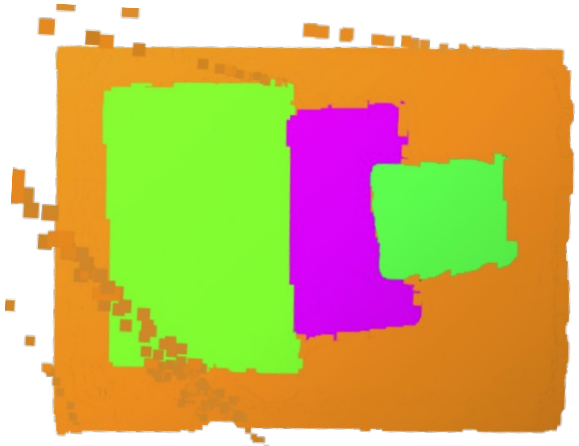
- **Scene** = Envelope + PiabBox + WhiteBox

RGB Point Cloud

3-4s (AMAX)



Semantic Segmentation

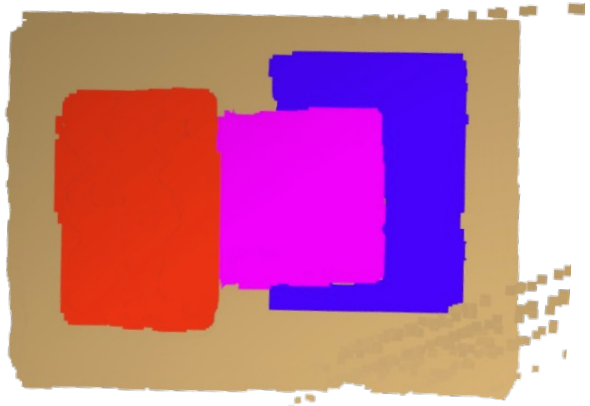


Instance Segmentation

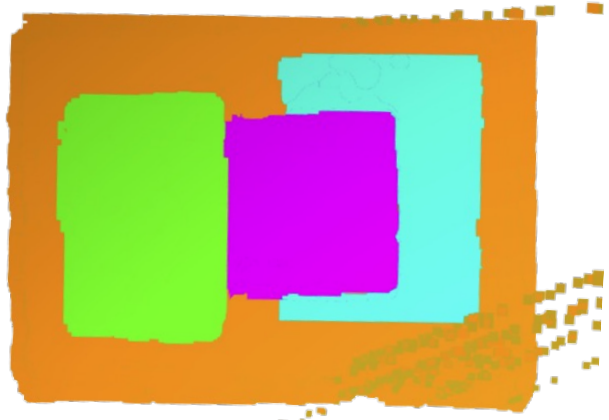
- **Scene** = Envelope + IPad Cover + WhiteBox

RGB Point Cloud

3-4s (AMAX)



Semantic Segmentation



Instance Segmentation

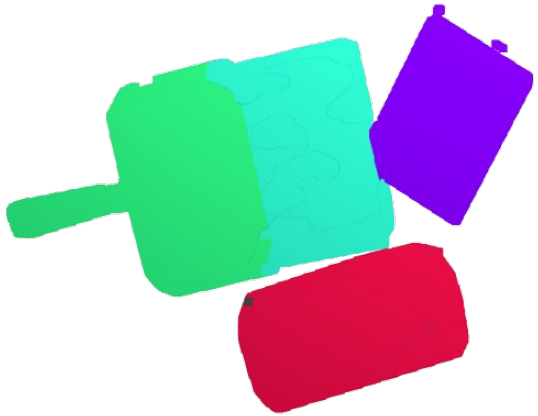
– Inference Result (No Tote)

- **Scene** = Dustpan + PiabBox + IPad Cover + PencilBox

RGB Point Cloud

3-4s (AMAX)

Semantic Segmentation

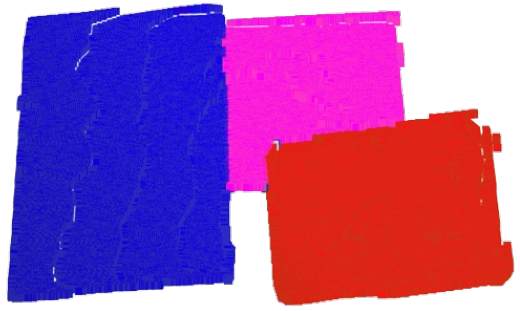


Instance Segmentation

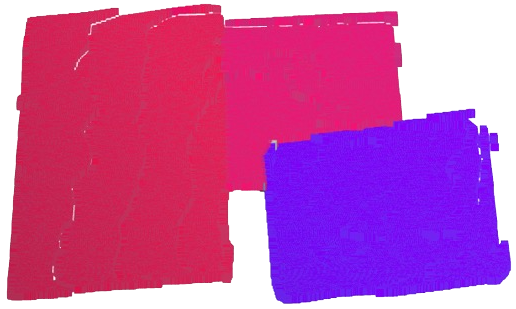
- **Scene** = Envelope + WhiteBox + IPad Cover

RGB Point Cloud

3-4s (AMAX)



Semantic Segmentation



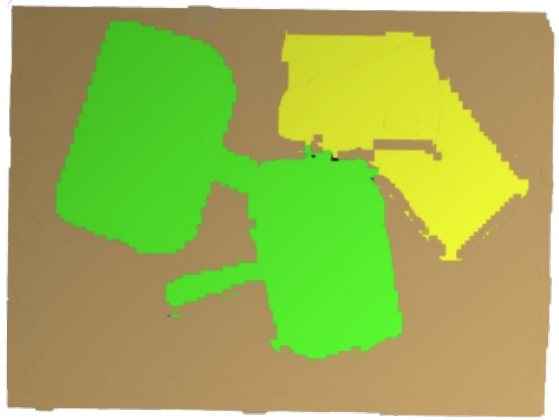
Instance Segmentation

– Inference Result (Instance)

- **Scene** = Dustpan x2 + PiabBox x2

RGB Point Cloud

~5s (AMAX)



Semantic Segmentation

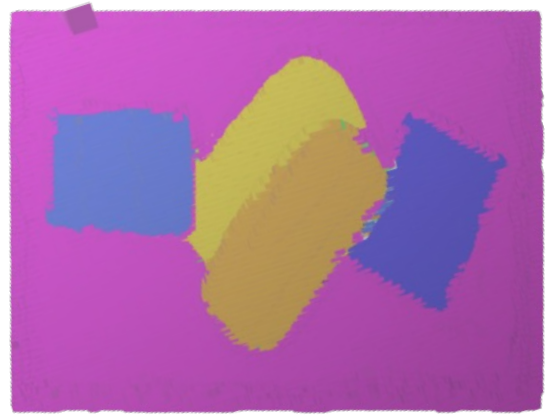
Instance Segmentation

- **Scene** = PencilBox x2 + PiabBox x2

RGB Point Cloud

~5s (AMAX)

Semantic Segmentation



Instance Segmentation

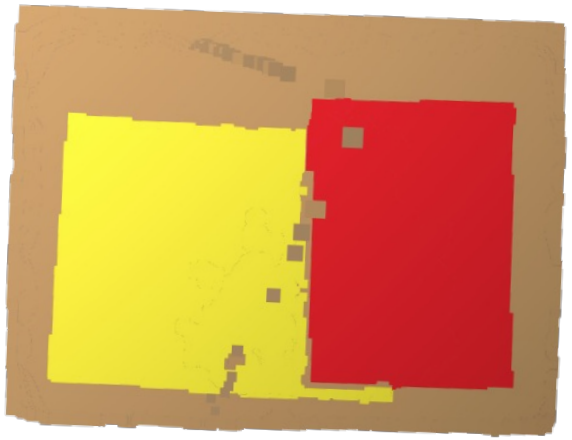
Amazon Envelope

– Inference Result

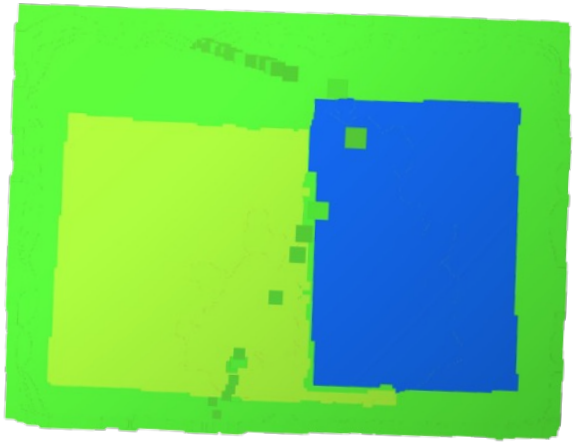
- **Scene** = Envelope bm3 + White Big

RGB Point Cloud

4-5s (AMAX)



Semantic Segmentation

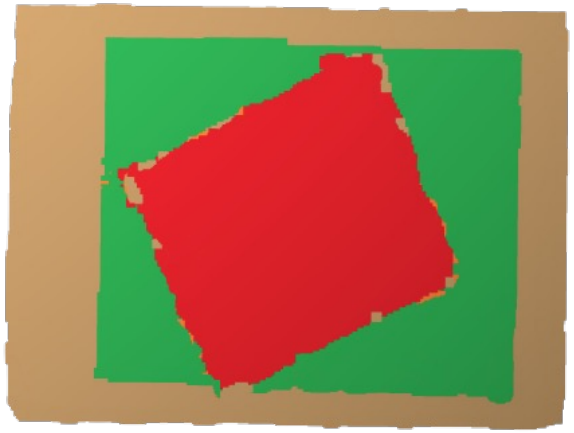


Instance Segmentation

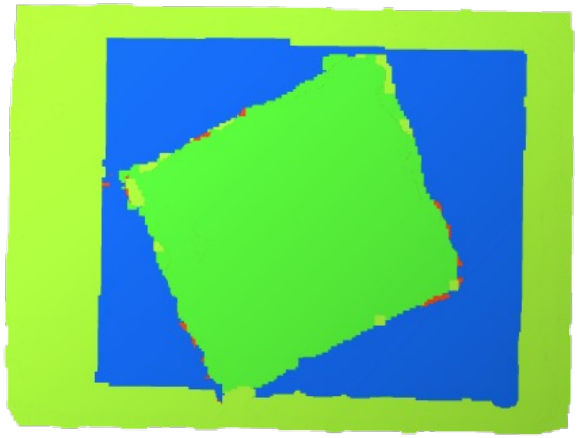
- **Scene** = Envelope bm5 + bm3

RGB Point Cloud

4-5s (AMAX)



Semantic Segmentation



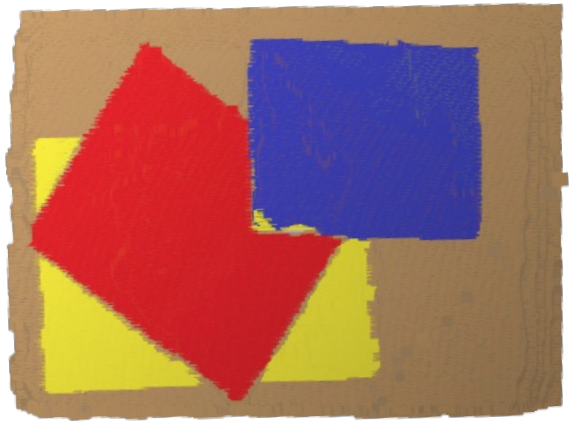
Instance Segmentation

– Inference Result

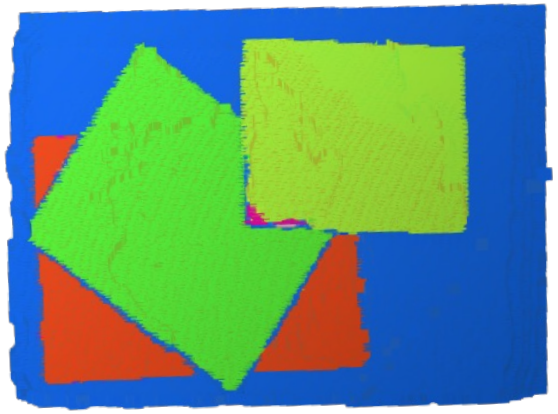
- **Scene** = Envelope bm3 + White Big + Small

RGB Point Cloud

4-5s (AMAX)



Semantic Segmentation

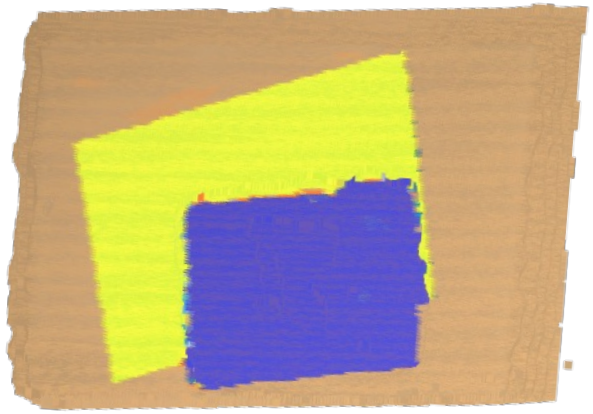


Instance Segmentation

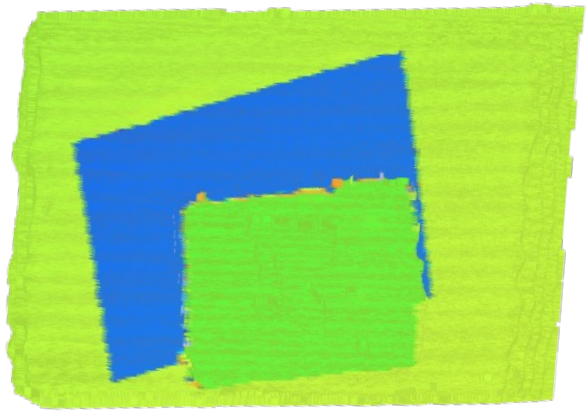
- **Scene** = Envelope White Big + White Small

RGB Point Cloud

4-5s (AMAX)



Semantic Segmentation



Instance Segmentation

Reflective Materials

– Inference Result (Without Noise)

- **Scene** = Tablet strip x2

RGB Point Cloud

2-3s (AMAX)

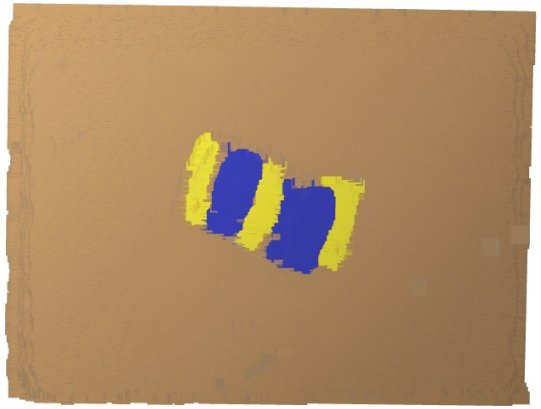
Semantic Segmentation

Instance Segmentation

- **Scene** = Orange Cylinder x3 + White Cylinder x3

RGB Point Cloud

2-3s (AMAX)



Semantic Segmentation

Instance Segmentation

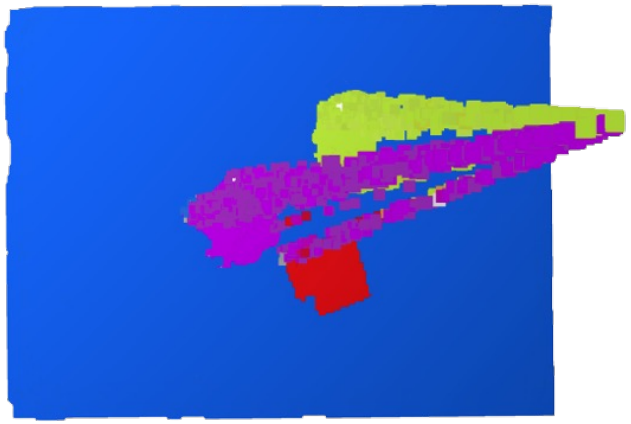
– Inference Result (With Noise)

- **Scene** = Thumbtack Box x3

RGB Point Cloud

2-3s (AMAX)

Semantic Segmentation

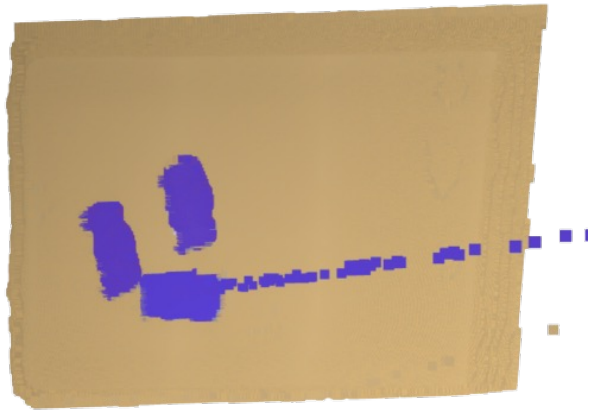


Instance Segmentation

- **Scene** = White Cylinder x3

RGB Point Cloud

2-3s (AMAX)



Semantic Segmentation

Instance Segmentation

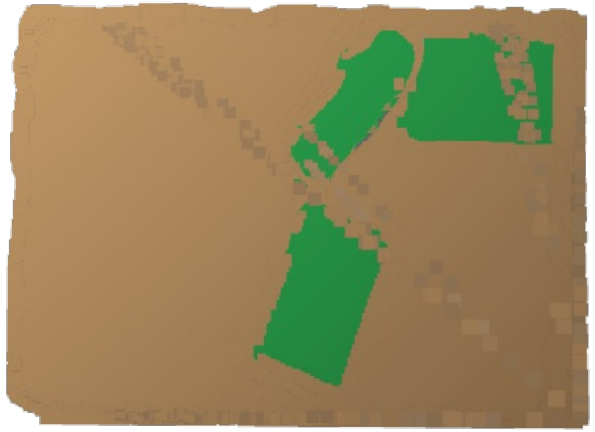
Colorful Boxes

– Inference Result (Uneven Tote)

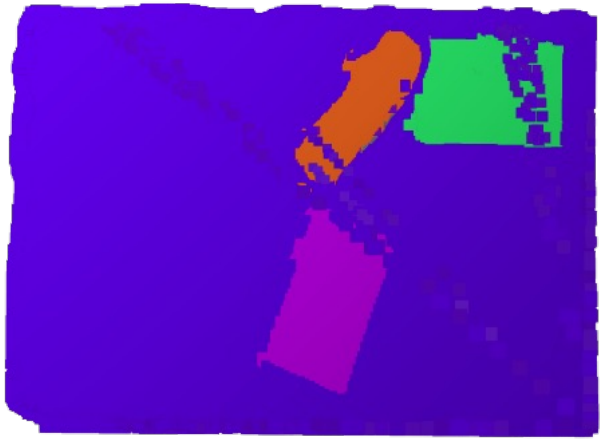
- **Scene** = Bandage Box x3

RGB Point Cloud

2-3s (AMAX)



Semantic Segmentation



Instance Segmentation

- **Scene** = Aspercreme Box x3

RGB Point Cloud

2-3s (AMAX)

Semantic Segmentation

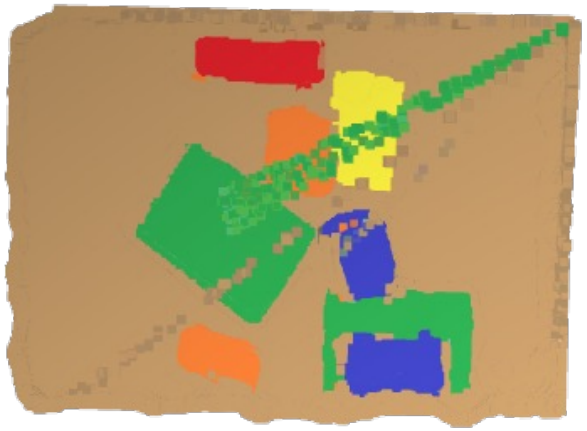
Instance Segmentation

– Inference Result (Uneven Tote)

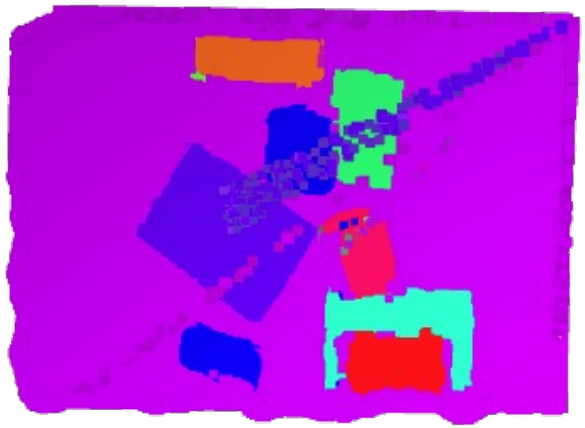
- **Scene** = Cluttered 1

RGB Point Cloud

~3s (AMAX)



Semantic Segmentation

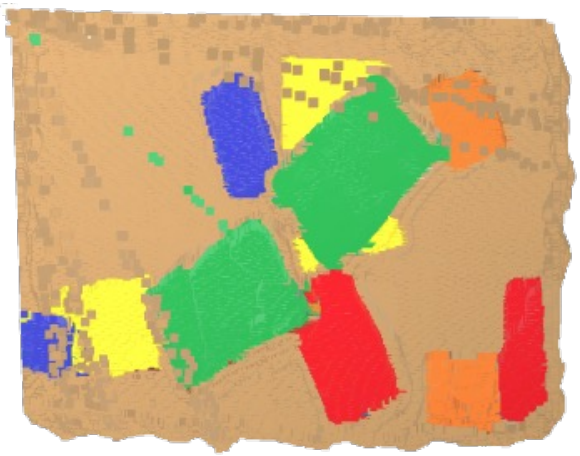


Instance Segmentation

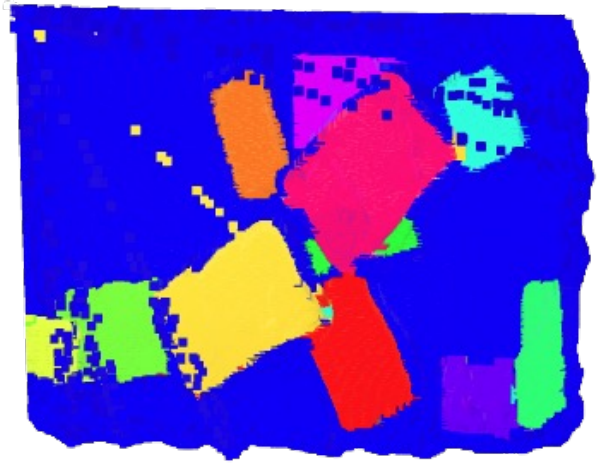
- **Scene** = Cluttered 2

RGB Point Cloud

~3s (AMAX)



Semantic Segmentation



Instance Segmentation

Fashion Clothes

– Inference (Clutter)

- **Scene** = Fashion Cluttered 1

RGB Point Cloud

2~3s (AMAX)

Semantic Segmentation

Instance Segmentation

- **Scene** = Fashion Cluttered 2

RGB Point Cloud

2~3s (AMAX)

Semantic Segmentation

Instance Segmentation

Model Analysis

– Q&A Analysis

- **Why we keep getting decent result on various objects so far?**
  Mask3D works well on giant 3D dataset (Scannet/S3DIS/STPLS3D). Our data is small and of high quality for the model to converge (overfitting)

- **Why result of reflective materials (84%) is lower than others (>90%)? (figure)**
  Reflectiveness causes loss of point cloud data, which is the biggest issues for 3D segmentation (point cloud data) from hardware

- **Can we further optimize the training & inference process (time)?**
  Test on various hardware platforms (GeForce RTX 1060 series vs GeForce RTX 3070 Ti) shows that hardware makes difference for training process. And inference can be optimized by deploying TRT model on Jetson

- **How we feed data to train a great model?**
  1). Manual label should be good to generate PCD with high quality
  2). Collect data in different/complex scenarios as much as possible

– Dataset Collection (Fashion Clothes)

- **Scene** = Combination of objects of 5 types (right next to each other)
- **Data Scale** = Combination (9) x Number (15) x Data Augment (4) = 540





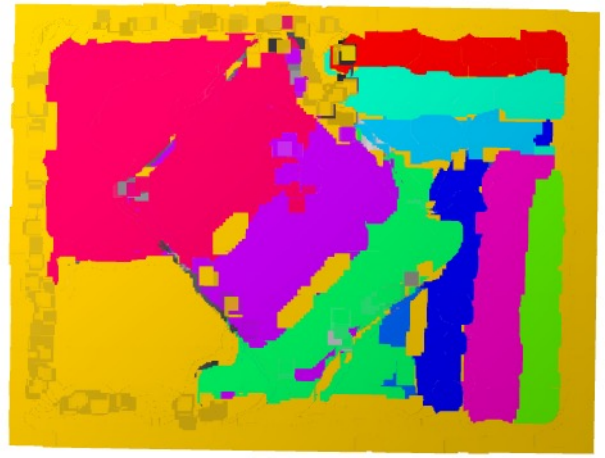

– Inference (Unexpected for overlap scenes)

- **Scene** = Fashion Overlap 1

RGB Point Cloud

2~3s (AMAX)

Semantic Segmentation



Instance Segmentation

- **Scene** = Fashion Overlap 2

RGB Point Cloud

2~3s (AMAX)

Semantic Segmentation

Instance Segmentation

– Dataset Collection (Fashion Clothes with overlap)

- **Scene** = overlap scenes of hanes and socks
- **Data Scale** = Number (30) x Data Augment (4) = 120





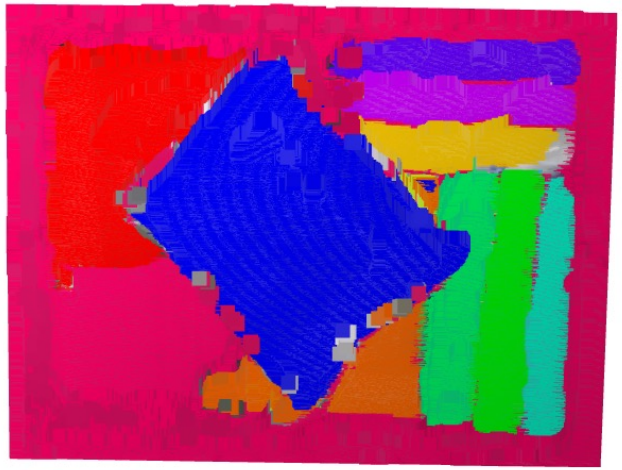

– Inference (Expected for overlap scenes)

- **Scene** = Fashion Overlap 1

RGB Point Cloud

2~3s (AMAX)

Semantic Segmentation

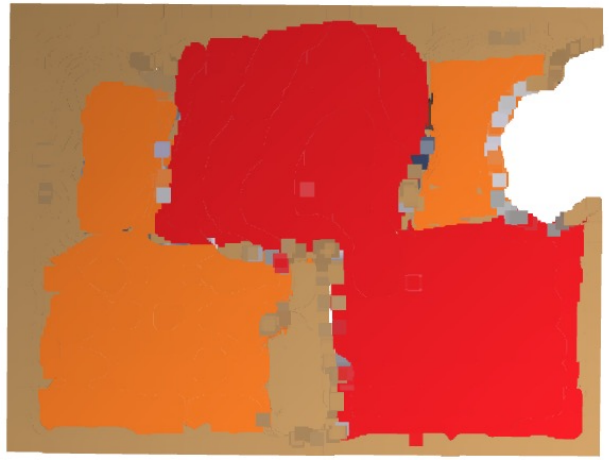


Instance Segmentation

- **Scene** = Fashion Overlap 2

RGB Point Cloud

2~3s (AMAX)



Semantic Segmentation

Instance Segmentation

Analysis with Other State-of-the-art Models

– SAM Model

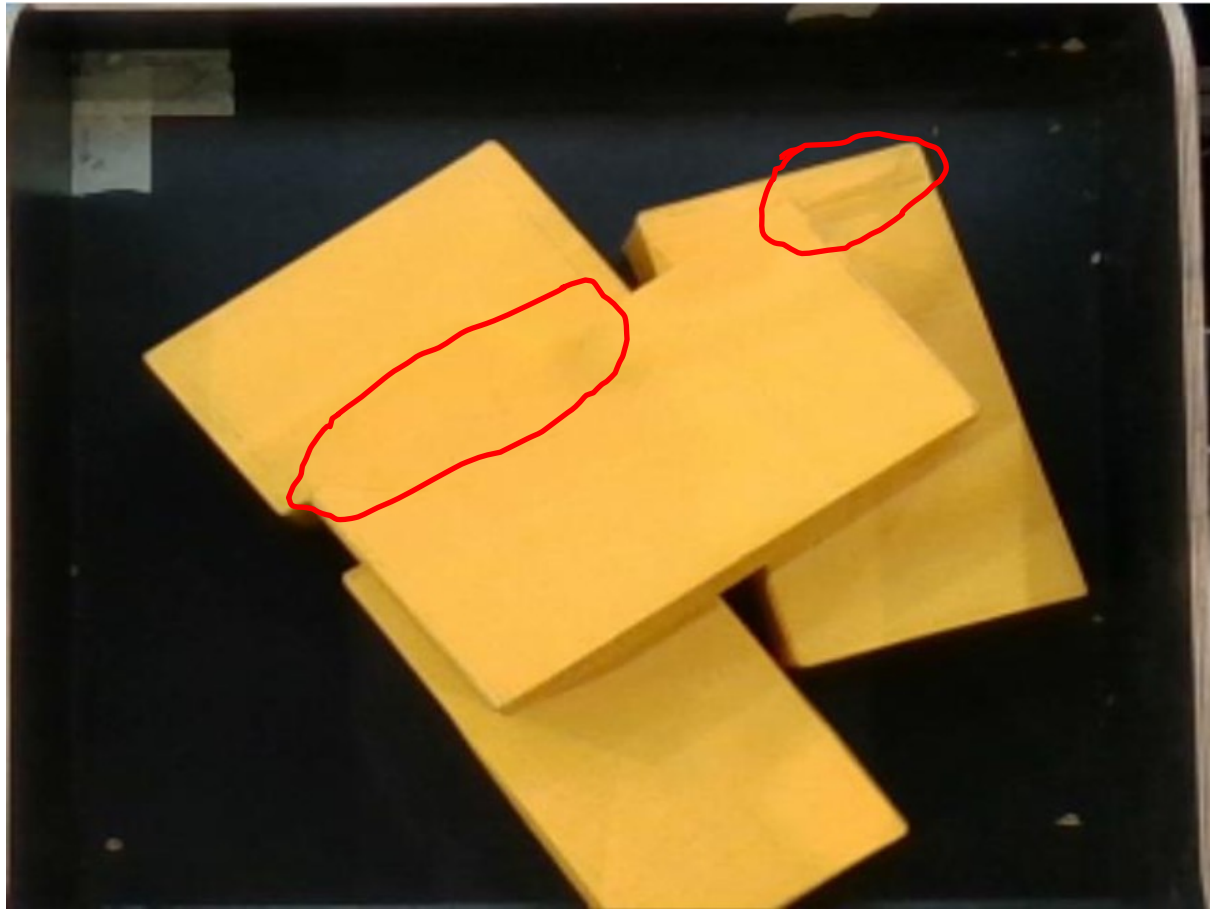
- **Open Source** = Segment Anything Model (SAM) Best model on 2D Segment
- **Composition** = Big Dataset Library (11 million) + Promptable Model
- **Functionality** = Efficient Labeling Tool for 3D Segmentation

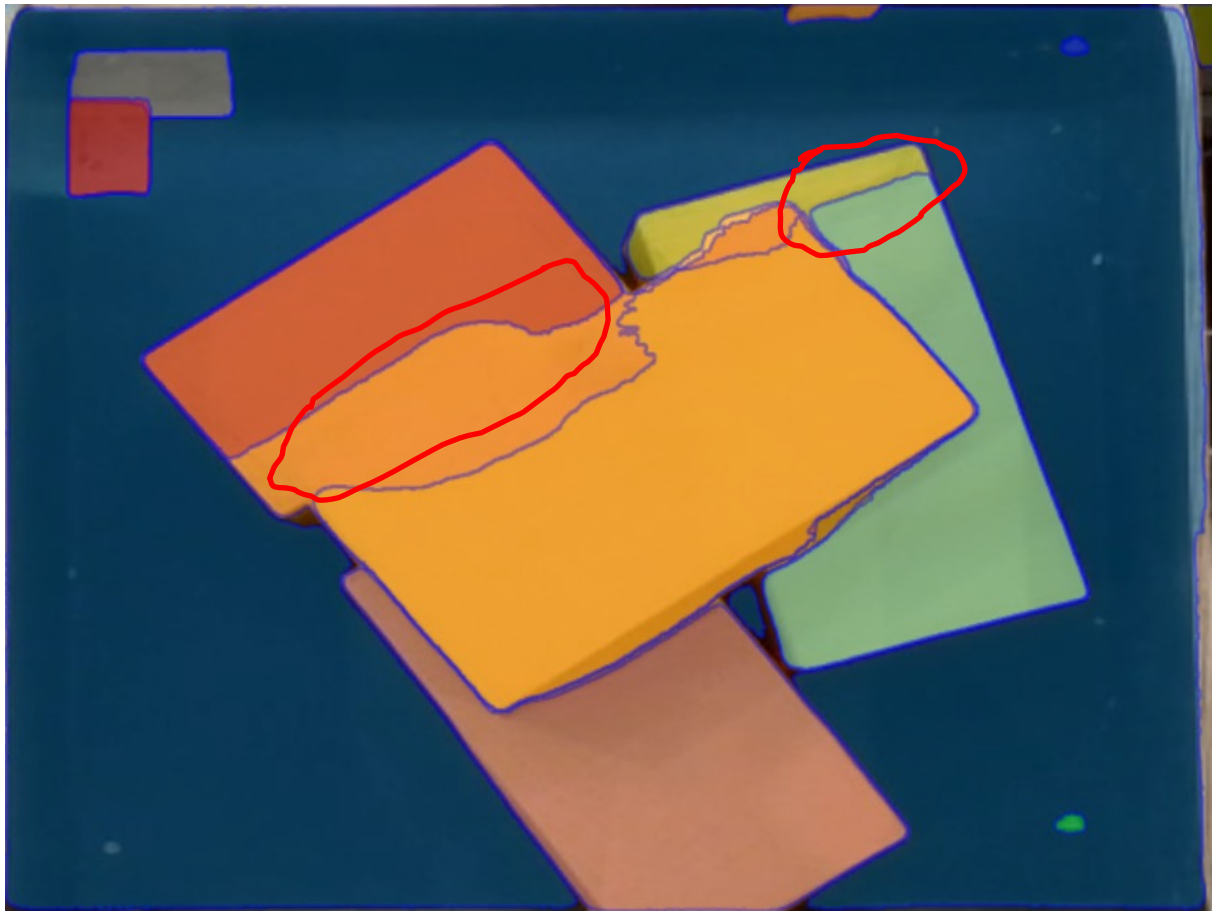
SAM Model

- **Advantages** = Can make accurate edge detection

– SAM Model

- **<u>Limitations</u> =** Inaccurate segment where edges are less obvious Need Depth



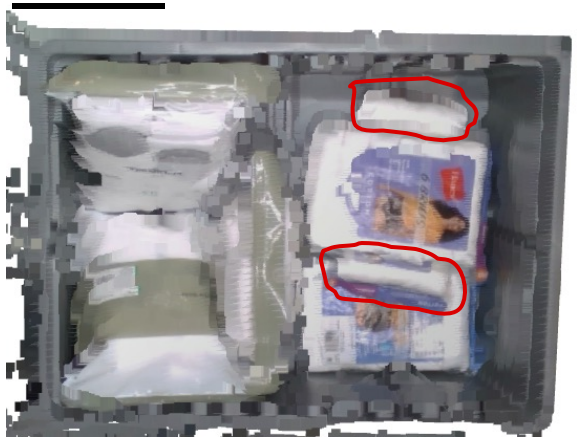


– SAM Model

- **<u>Limitations</u> =** Inaccurate segment for object under occlusion

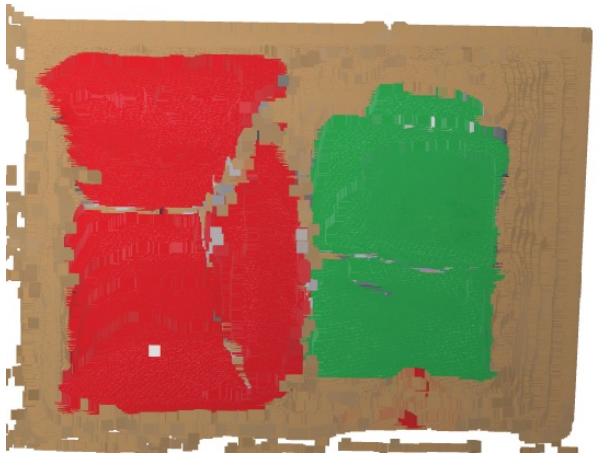
– Comparison with Mask3D

- **Merit** = Able to detect object in occlusion

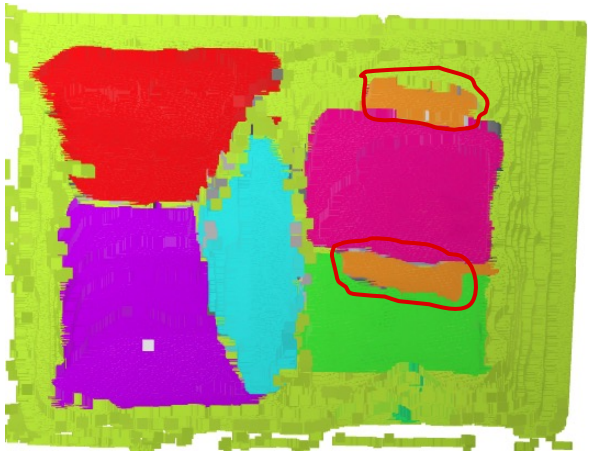


RGB Point Cloud

Inference

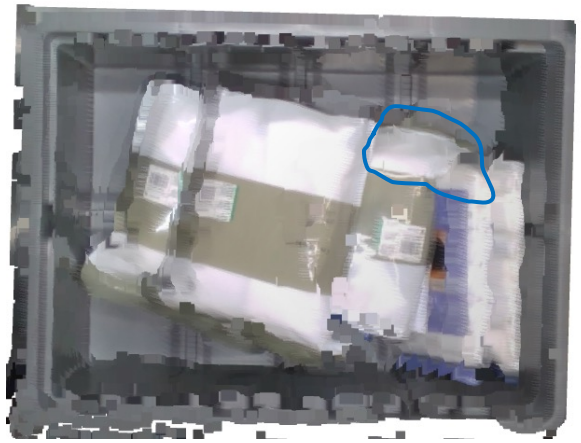


Semantic Segmentation



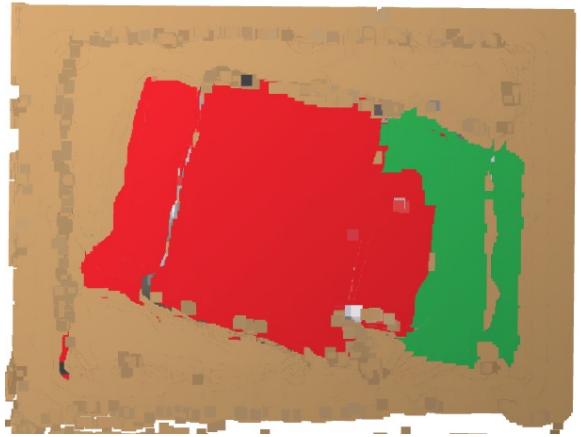
Instance Segmentation

- **Demerit** = not segment well with small dataset

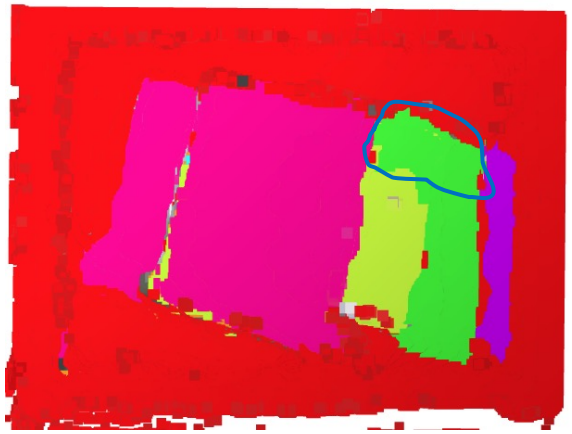


RGB Point Cloud

Inference



Semantic Segmentation



Instance Segmentation

ARMBench Dataset

- **Name** = Amazon Robotic Manipulation Benchmark (ARMBench)
- **Composition** = Big 2D Dataset (450000+ manual label) + Mask R-CNN Model
- **Limitations** = Not do well when degree of clutter increases (Mask3D outperforms it)

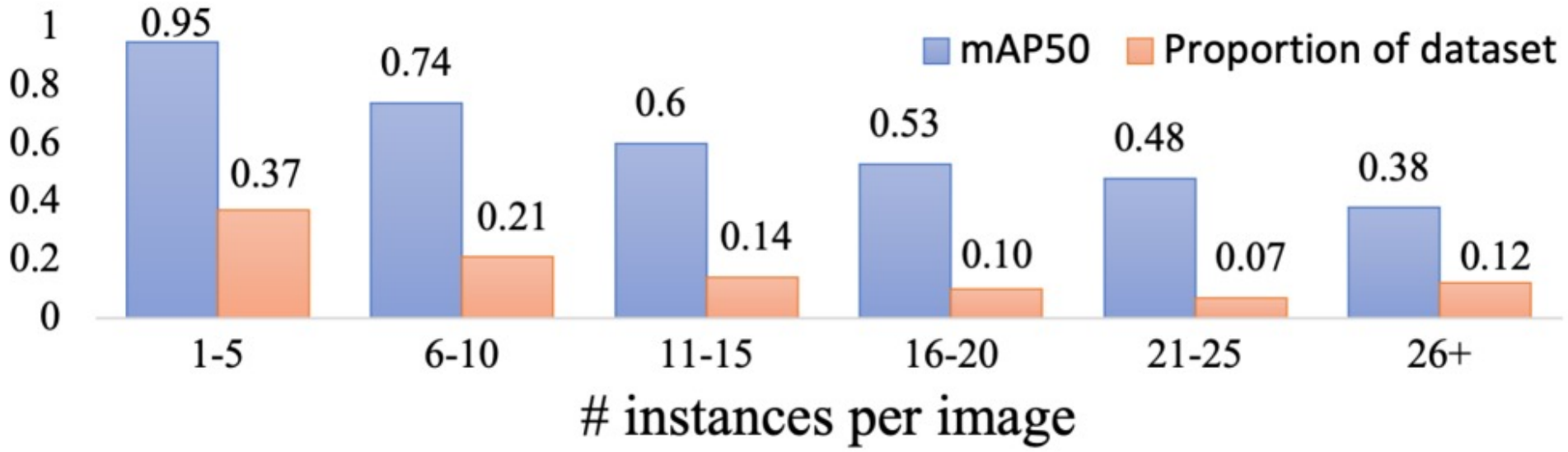


Fig. 4. Performance on *mix-object-tote* with varying degree of clutter.

Summary

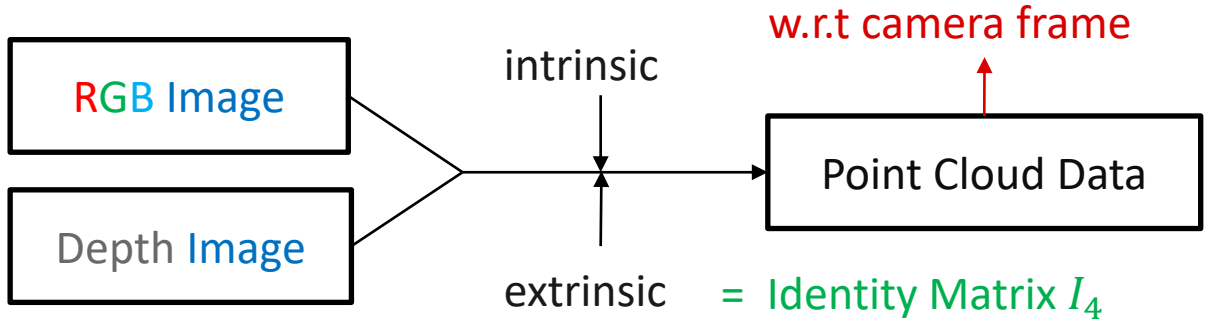
Summary

- **3D Segment for pick-and-place** =
  - **Advantage** = Mask3D performs semantic/instance segmentation well on 3D dataset
  - **Demerit** = labeling process is time-consuming and training takes long (transformer-based)
- **My Contribution** =
  - Setup Mask3D training pipeline on high-end computer locally
  - Propose 2D-based method for labeling point cloud data
  - Collect dataset, preprocessing and make analysis for experiment
  - Build inference test framework on Nvidia Jetson Orin platform
  - Keep documentation for experiment and code
- **Future work** =
  - Get post-processing done and generate pick points based on 3D point cloud data
  - Test feasibility of combination of Mask3D and SAM model
  - Deploy entire neural network on Jetson to reduce inference time

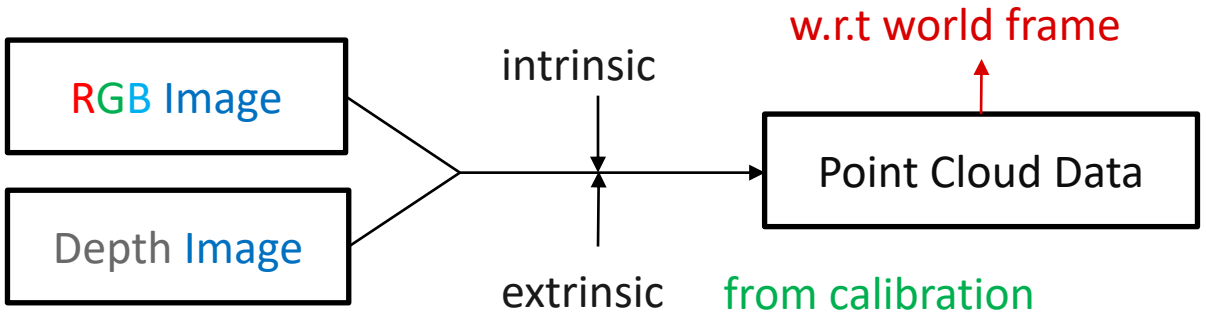
Post Processing

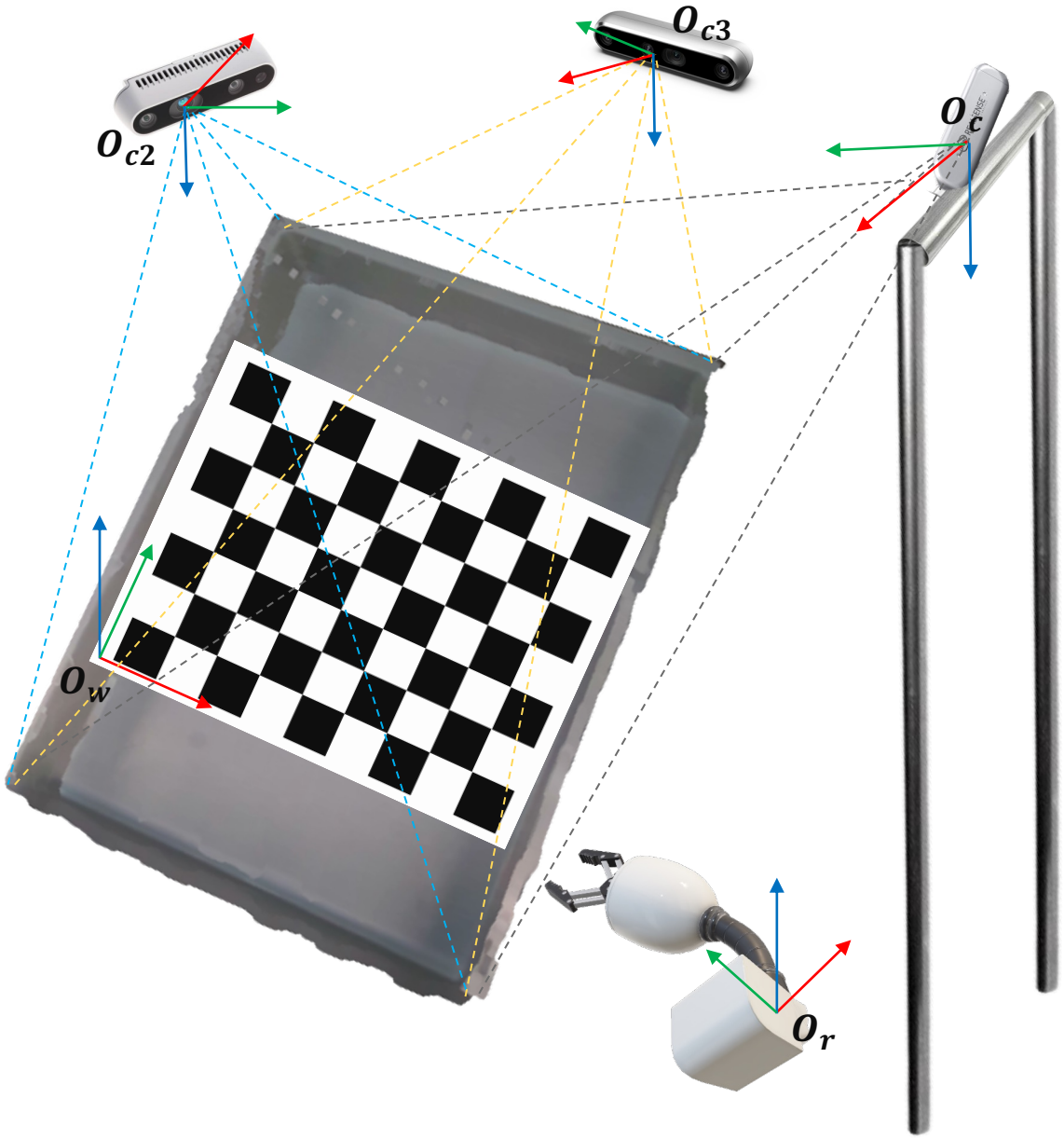
3D Coordinate System

- **Camera Frame** = generate point cloud data for training (must be still)



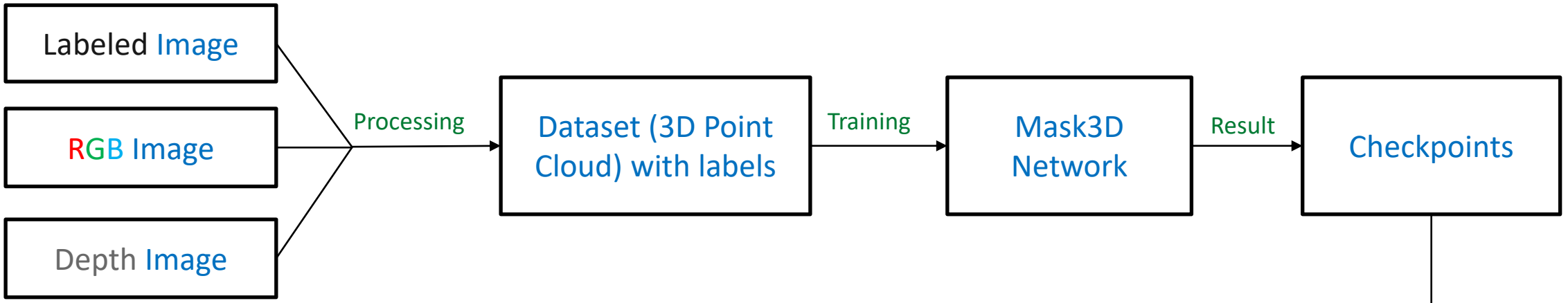
- **World Frame** = generate pick points



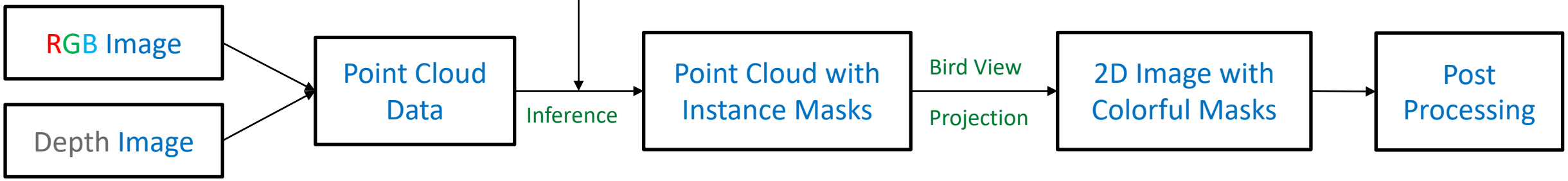


Mask3D Pipeline

- **Training** =

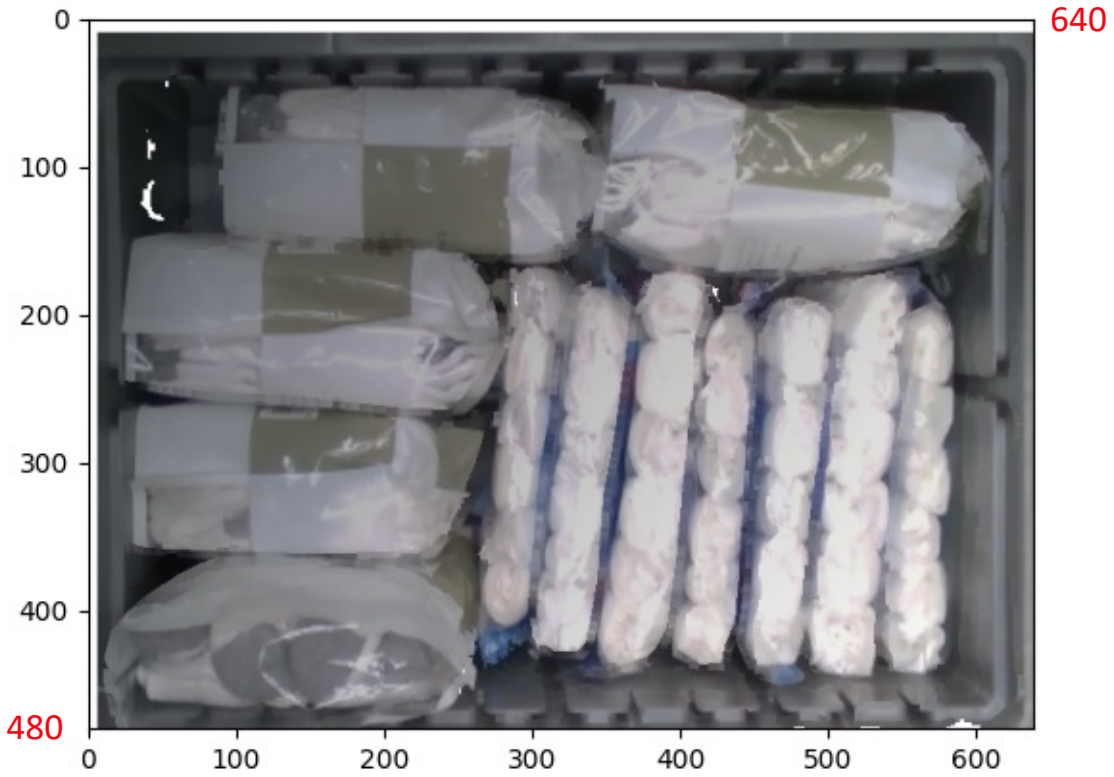


- **Inference** =

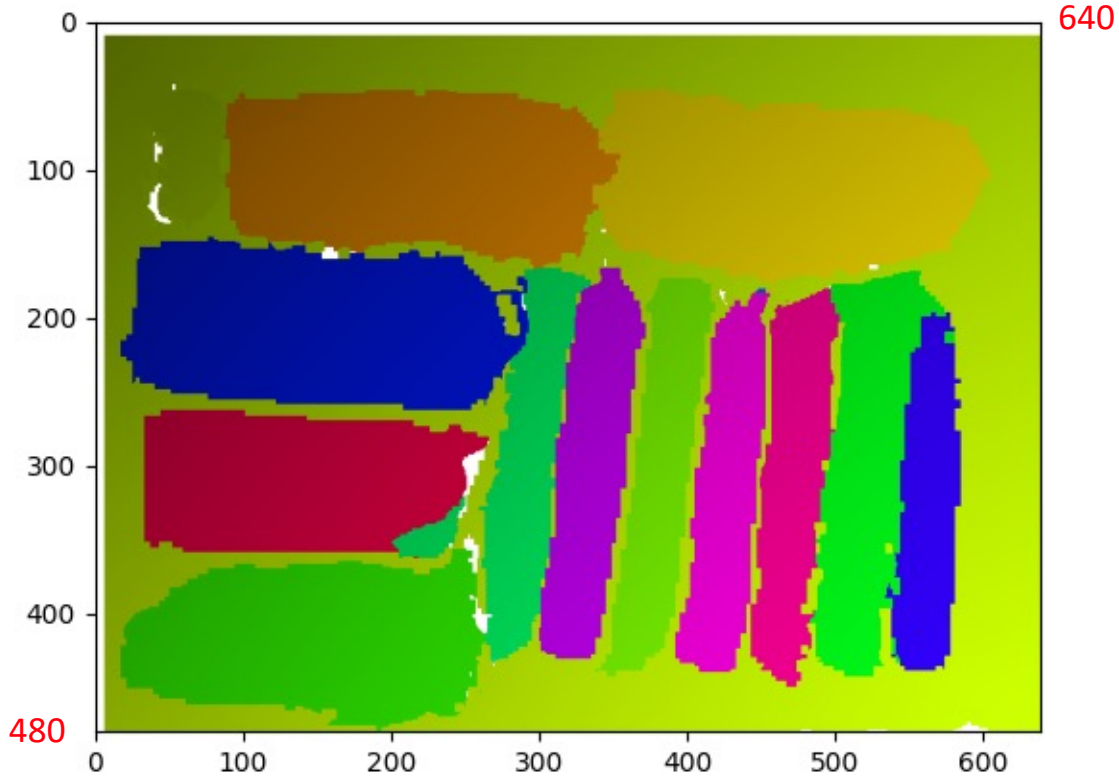


2D Bird View Projection (Top-down)

- **Capture** = point cloud + camera intrinsic param + extrinsic param
- **Result** = flat image (640 x 480)



2D RGB Image Projection



2D Instance Mask Projection

Results

Total Inference Time on AMAX = 1-1.5 sec

Novel Objects in Reflective Dataset:

```
| 5/6 [00:06<00:01,  1.21s/it]
```

3D Semantic Segmentation

```
4 0.9951390027999878
3 0.9337957501411438
3 0.9159947633743286
3 0.8607059717178345
1 0.07758796215057373
```

Confidence

```
4 0.9913750290870667
3 0.8828017115592957
3 0.8612247109413147
3 0.8286336660385132
3 0.11074884980916977
3 0.10346849262714386
```

Labels – 0: thumb_stacks 1: tablet_strip 2: orange_cylinder 3: white_cylinder 4: bin