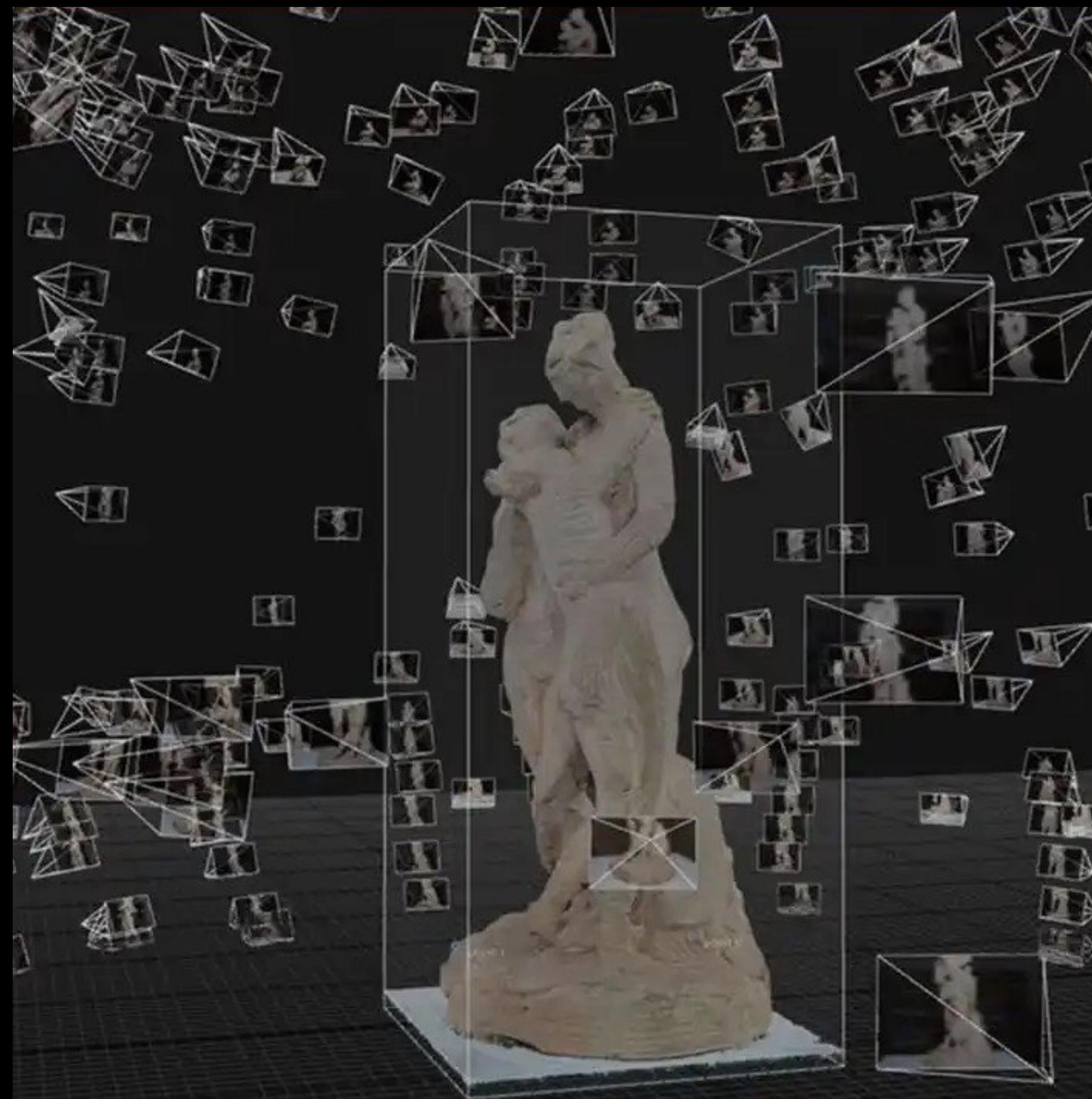
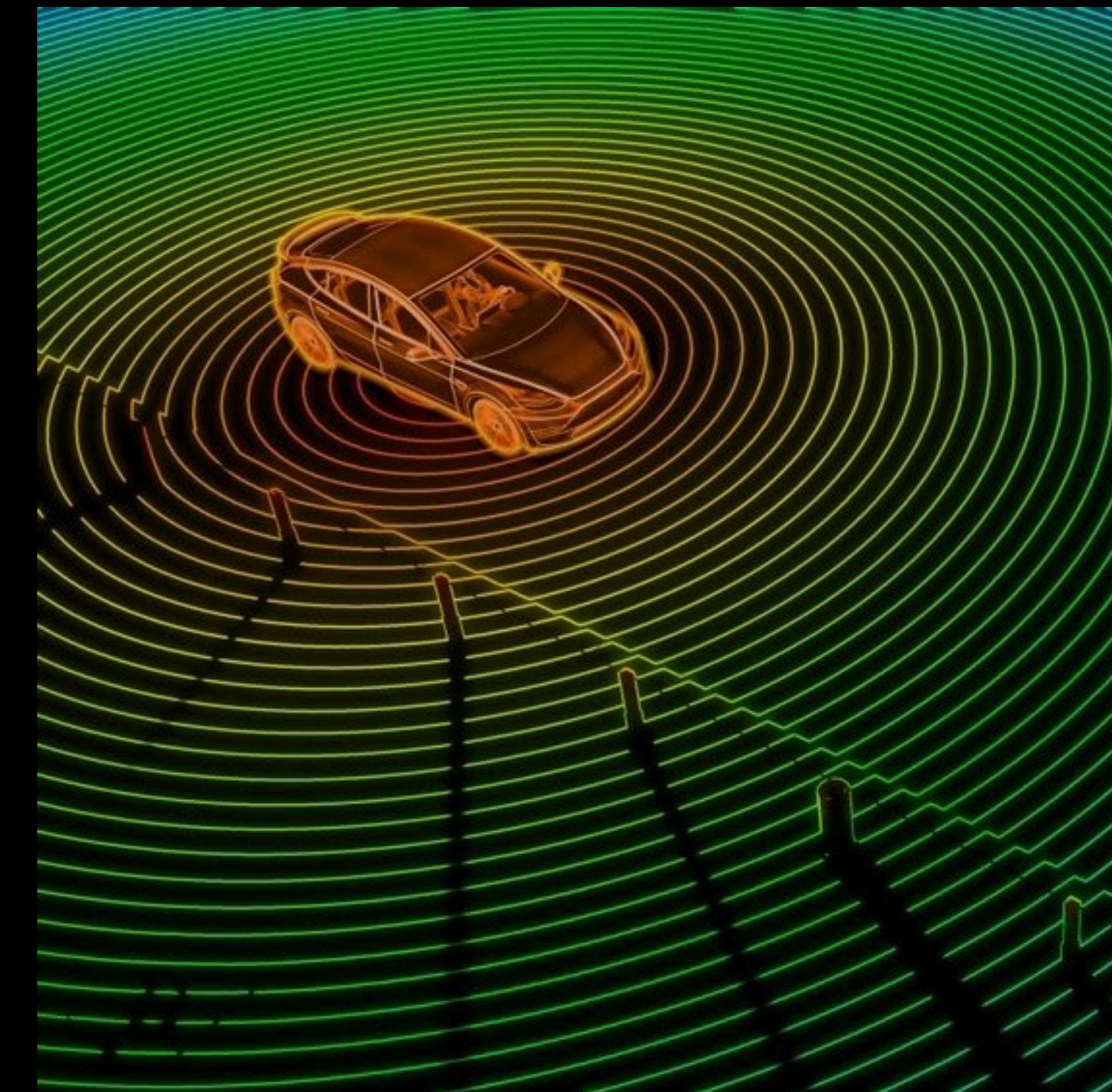


# Investigating 3D Model Reconstruction from 2D Stereoscopic Plant Images with Investigation between YOLOv8 and Detectron2 AI Architectures for Plant Class Segmentation

John Ivan Diaz, Craig Joseph Goc-ong, Kaye Louise Manilong  
Alvin Joseph Macapagal, Philip Virgil Astillo\*



Photogrammetry



LiDAR



+



+



+

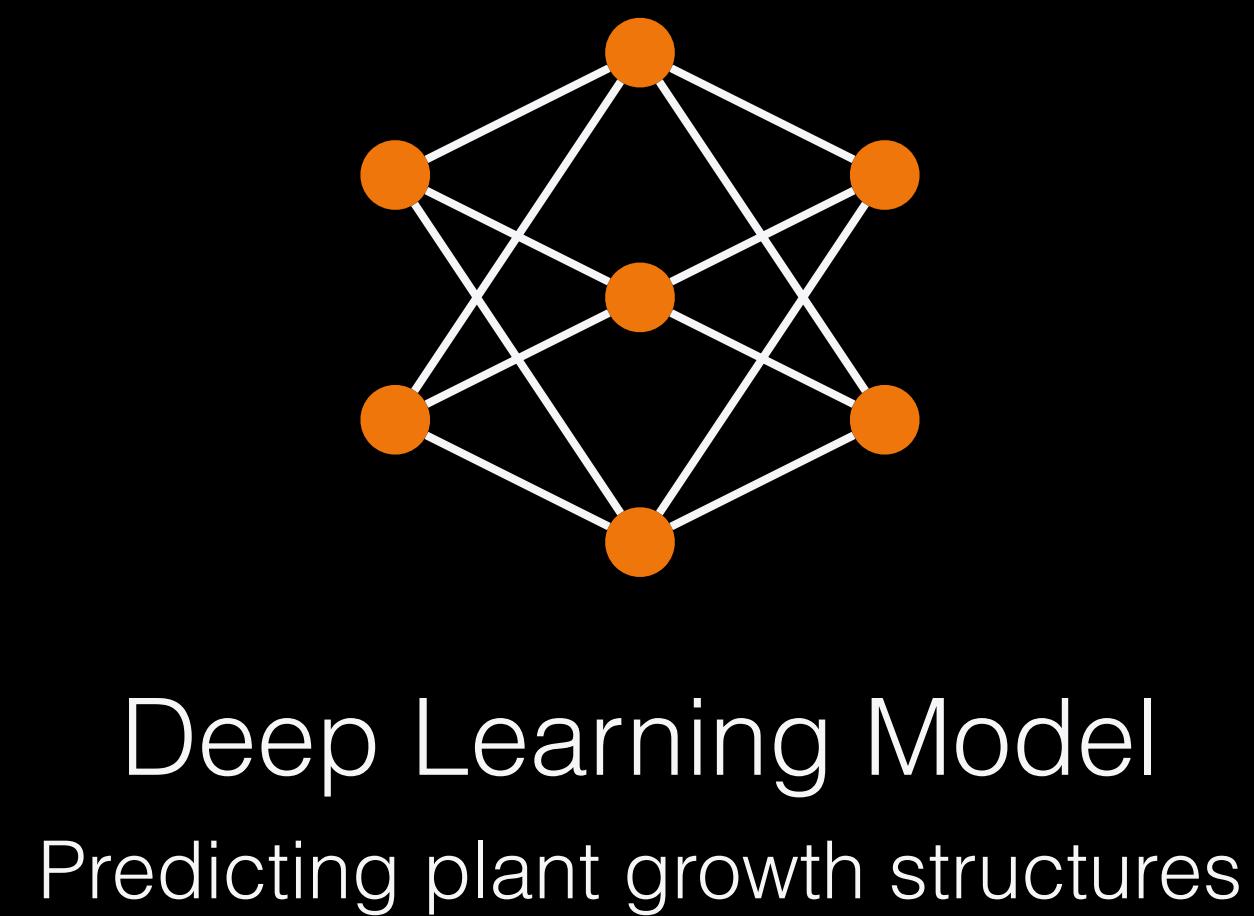
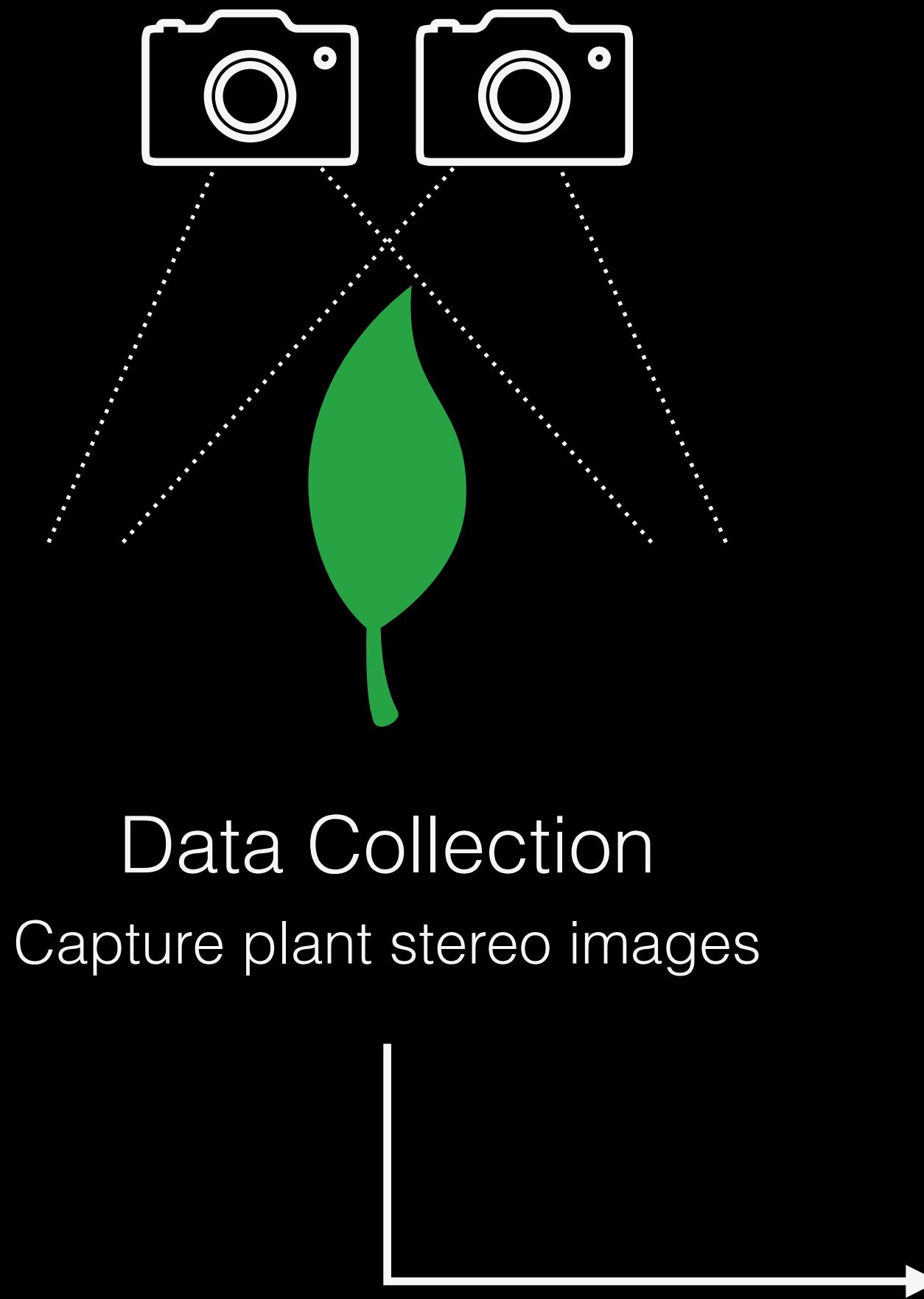


= ?

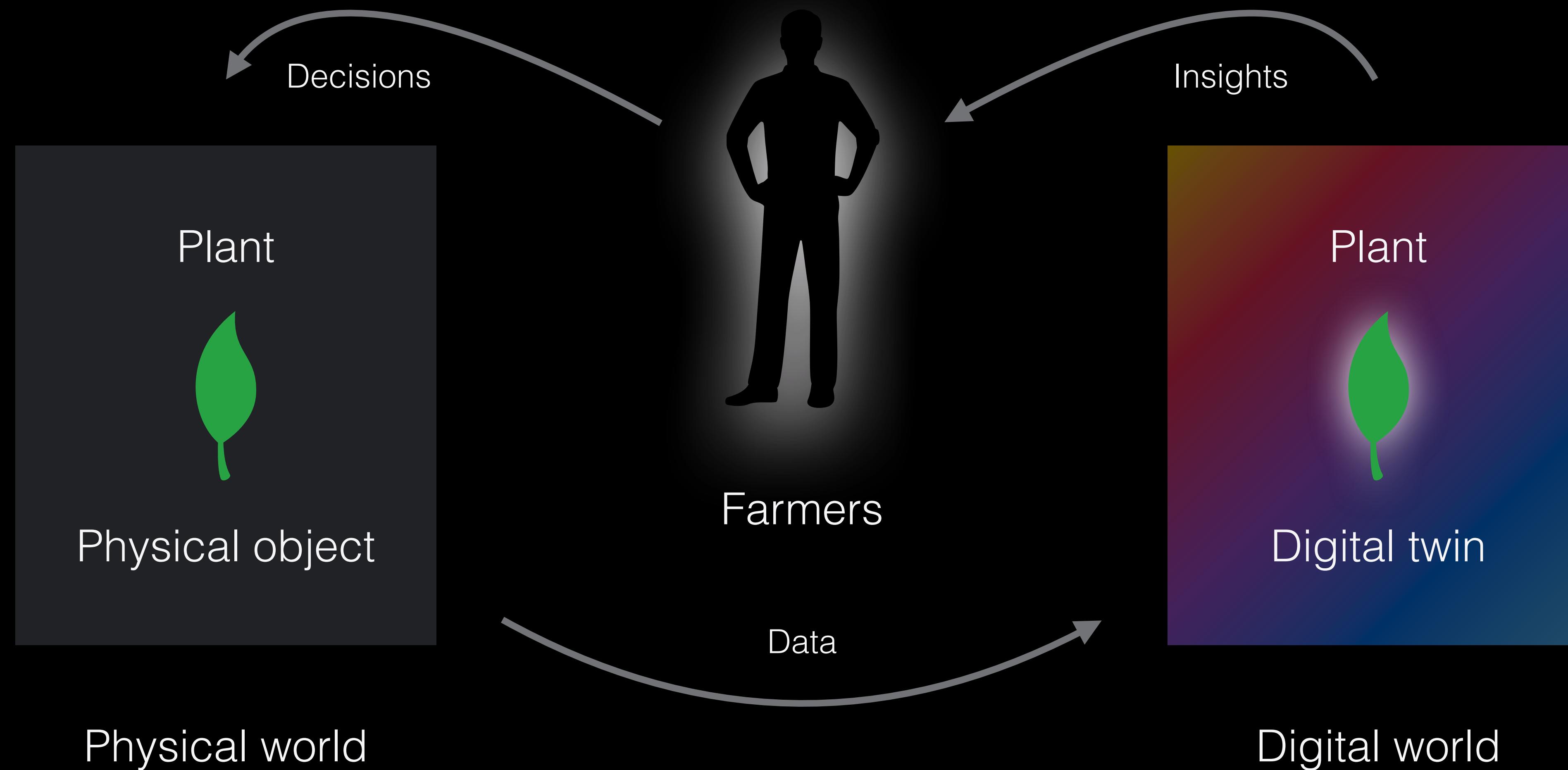
# Investigating 3D Model Reconstruction from 2D Stereoscopic Plant Images with Investigation between YOLOv8 and Detectron2 AI Architectures for Plant Class Segmentation

Diaz, Goc-ong, Manilong  
Macapagal, Astillo\*

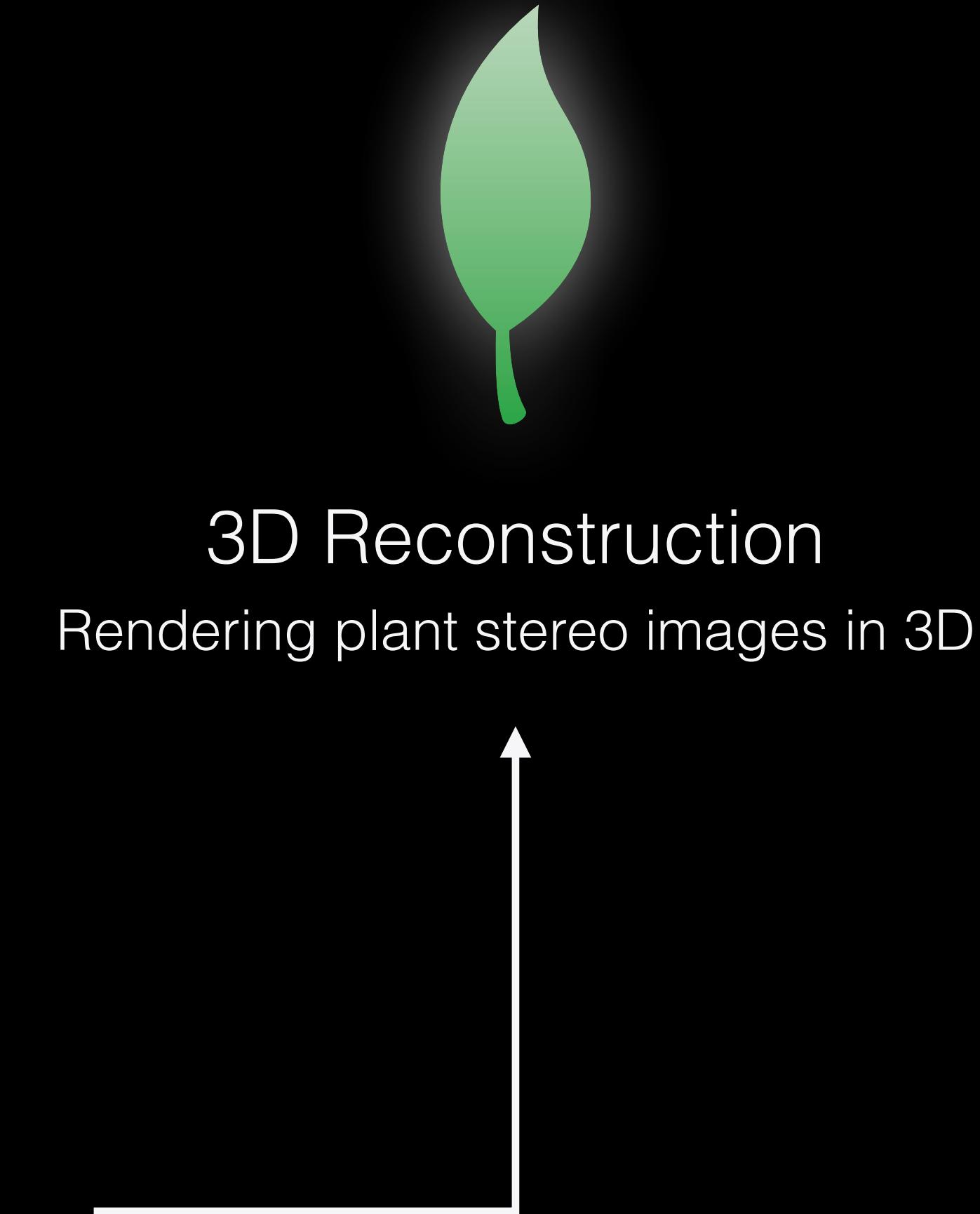
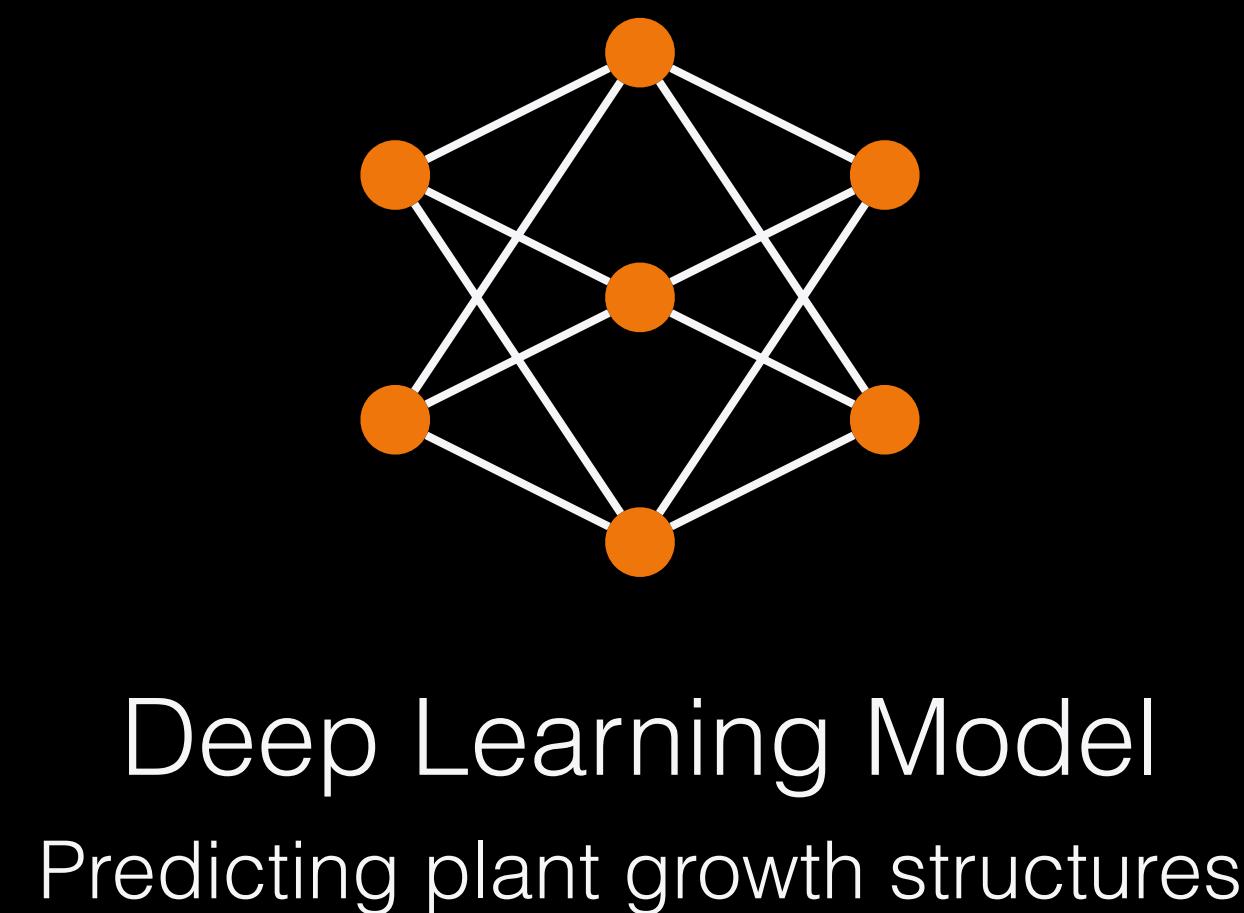
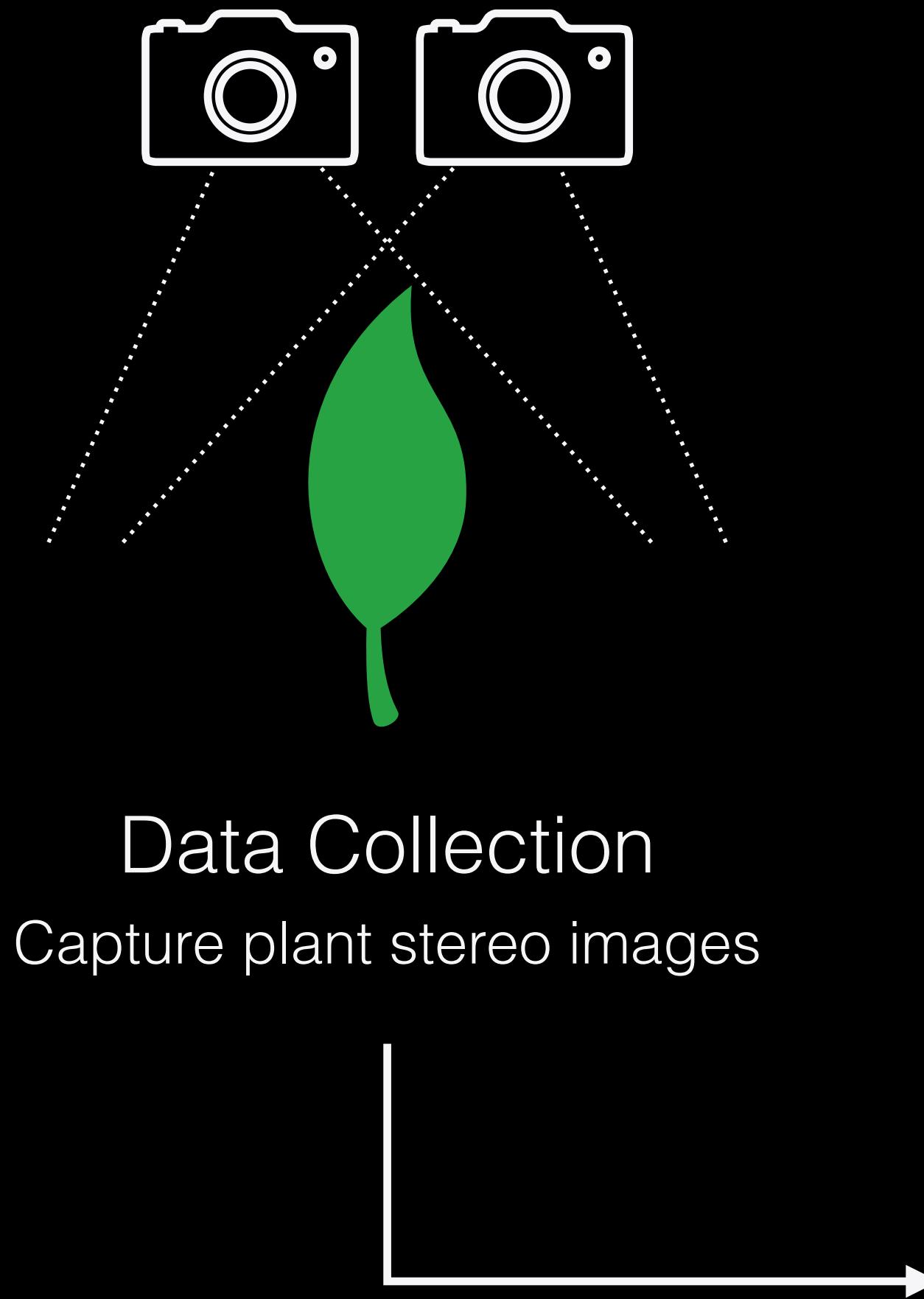
# ABOUT THE PARENT STUDY



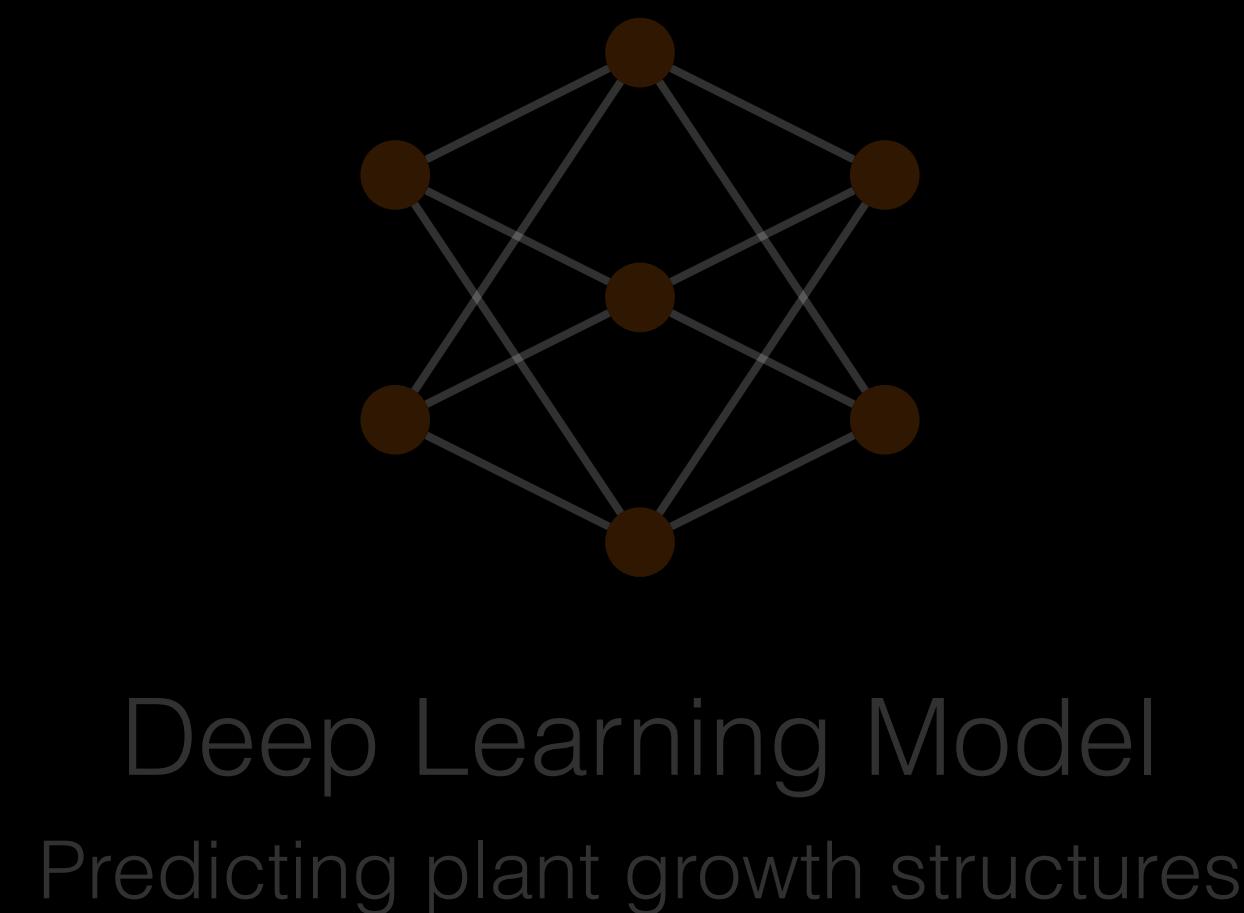
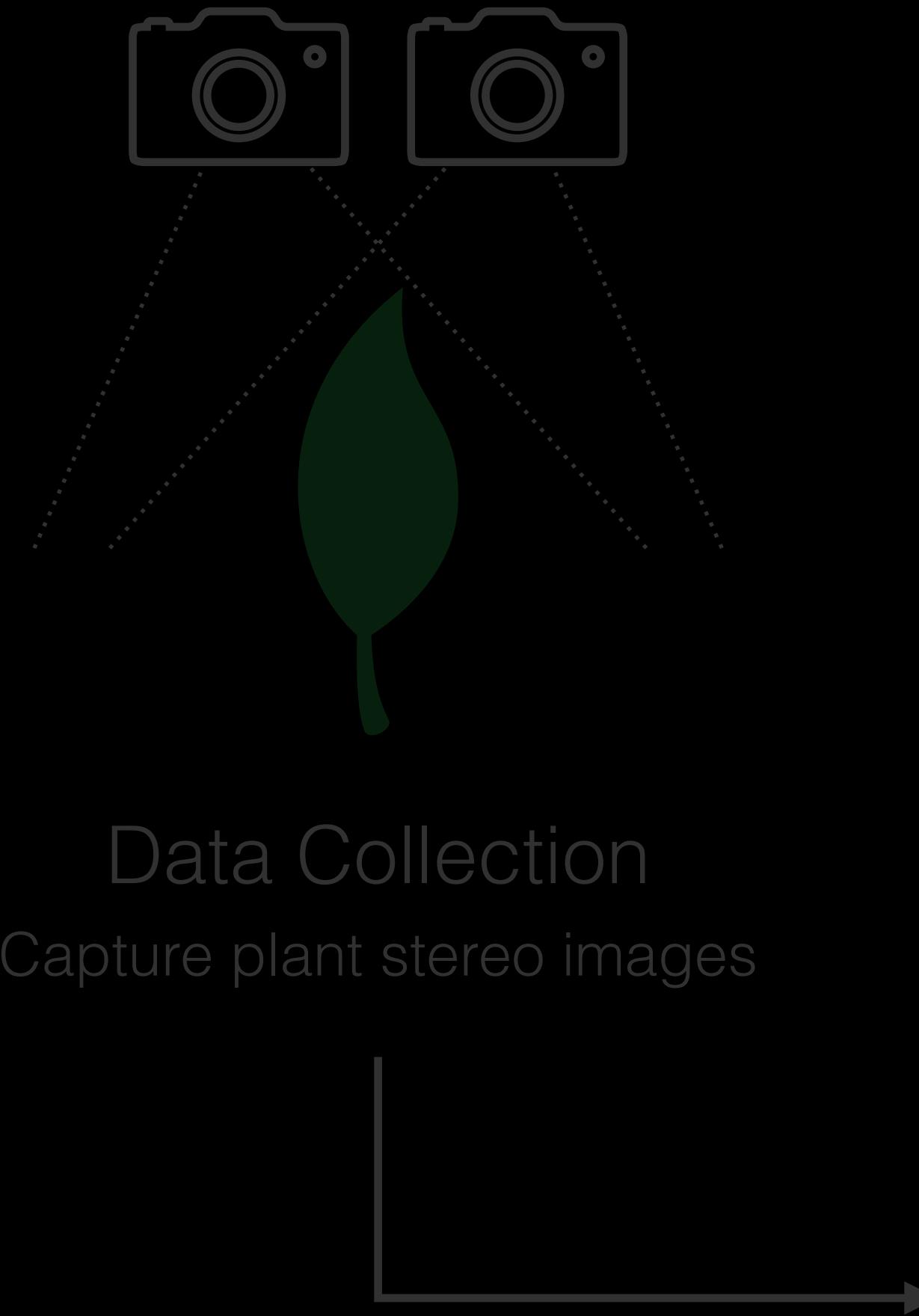
# ABOUT THE PARENT STUDY



# THIS CONFERENCE

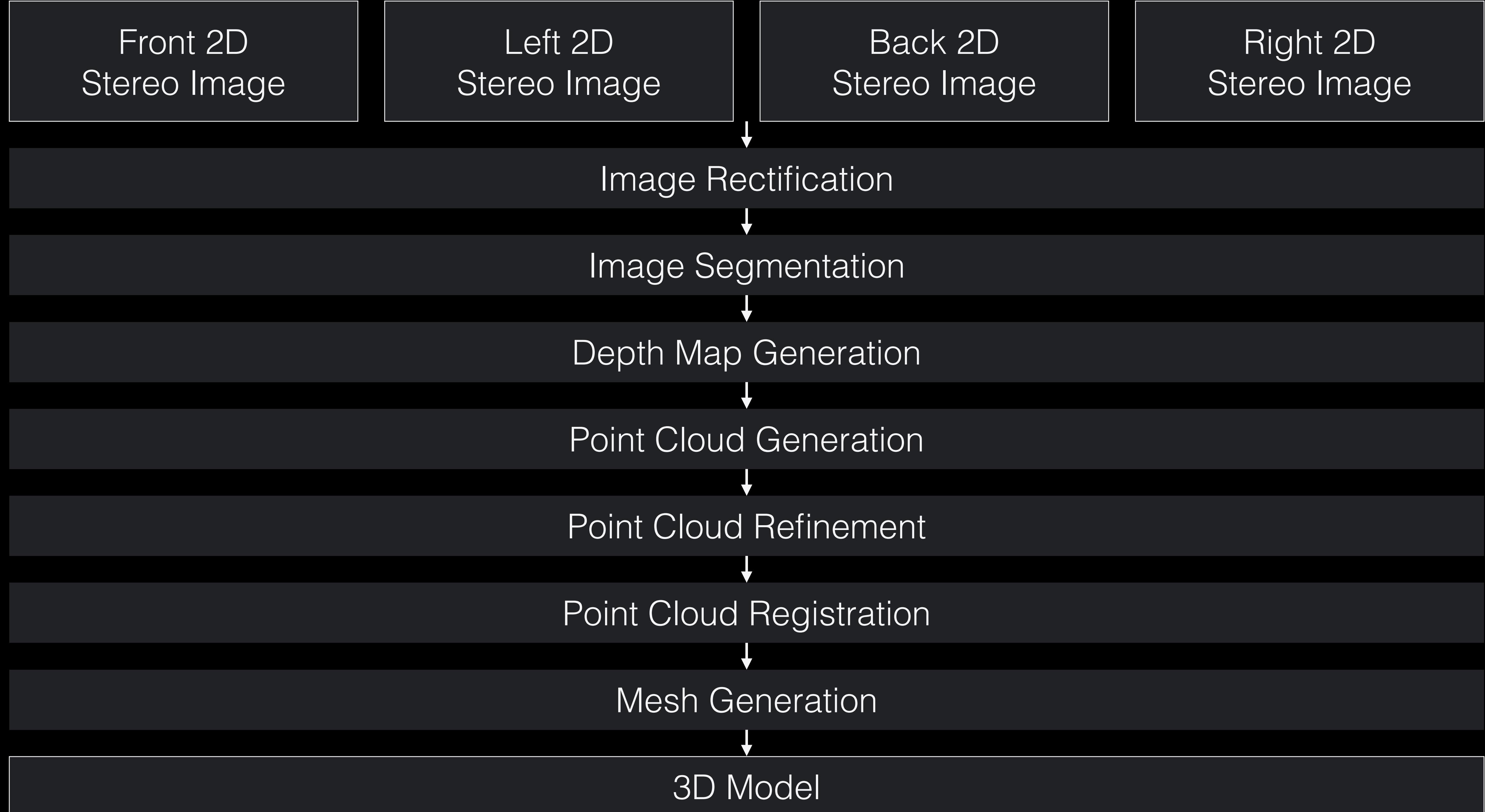


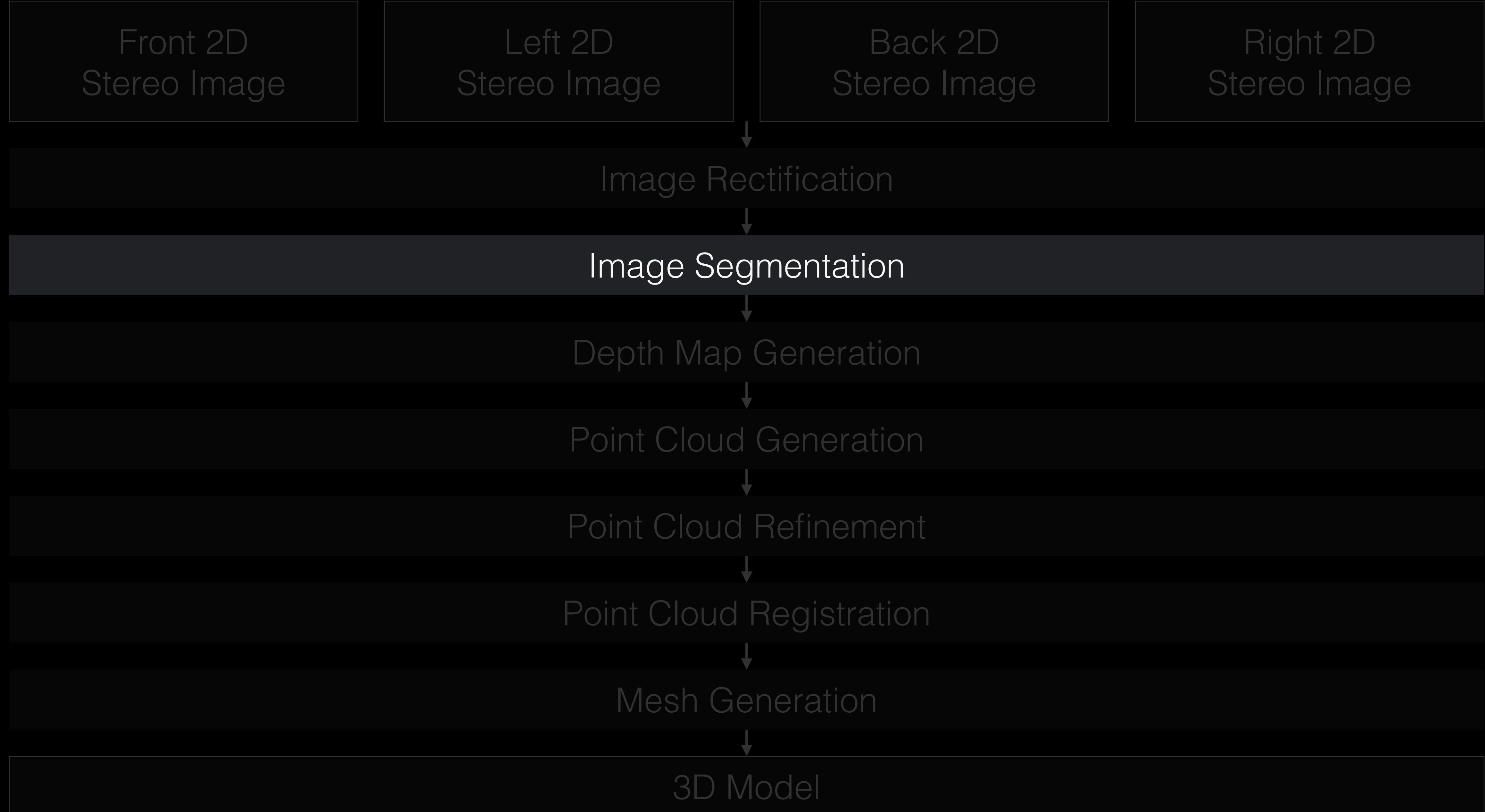
# THIS CONFERENCE

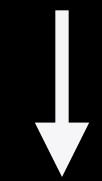


# OBJECTIVES

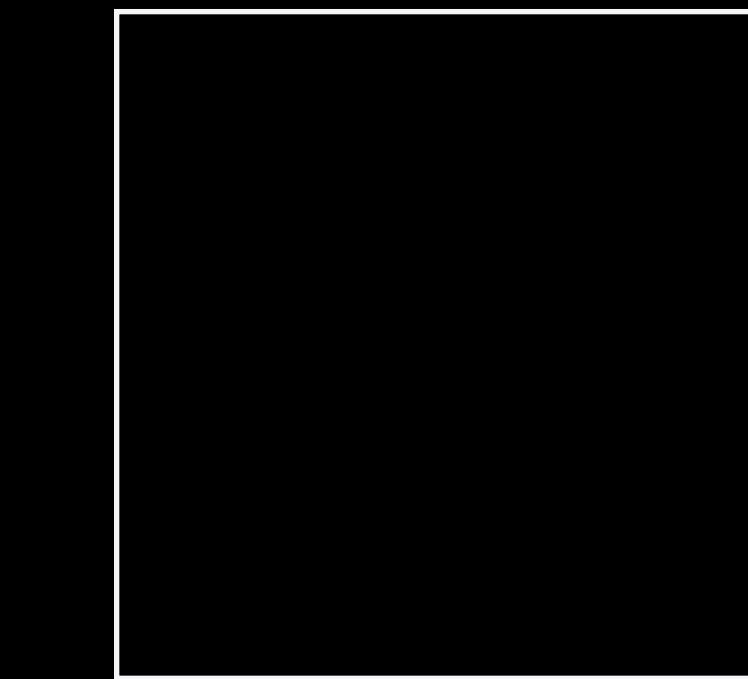
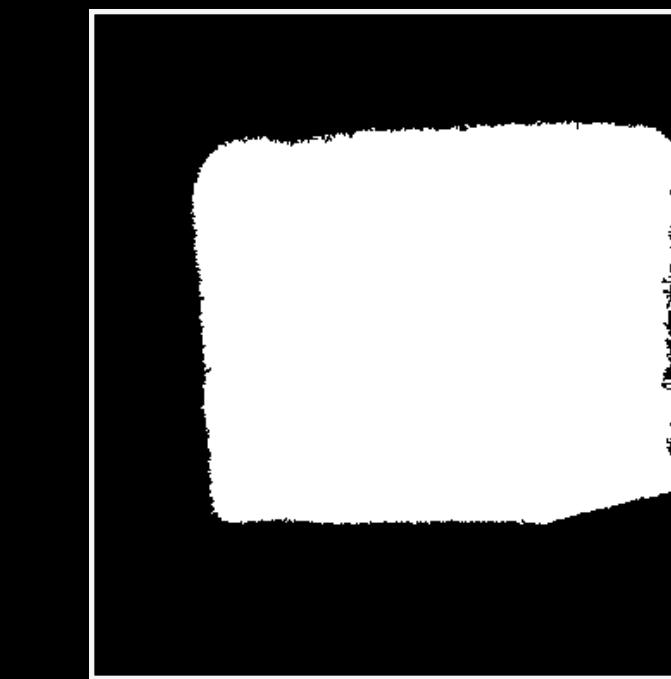
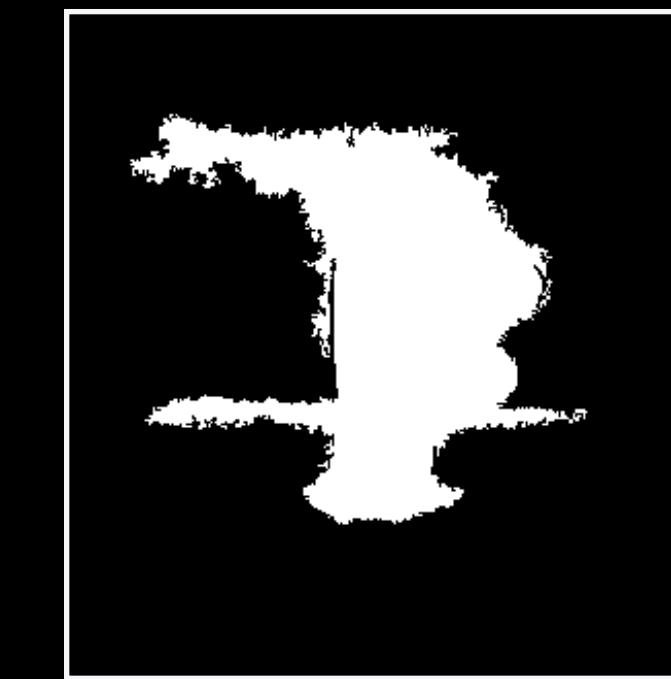
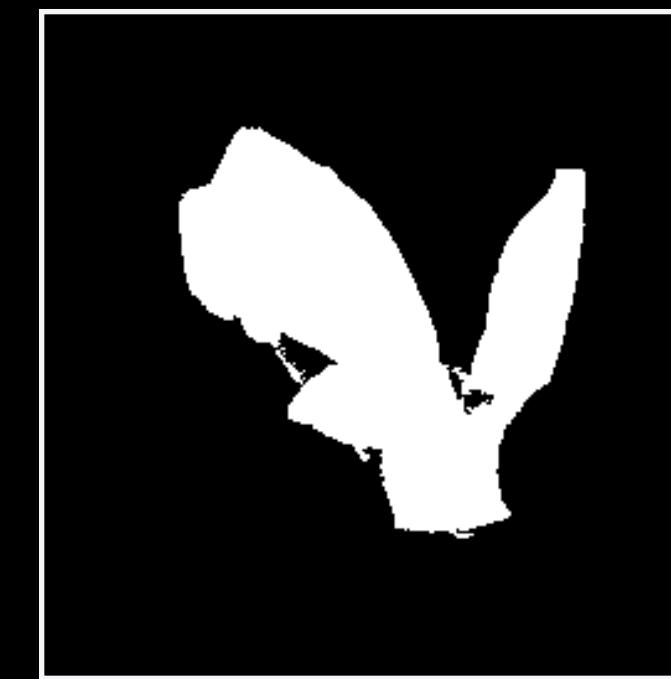
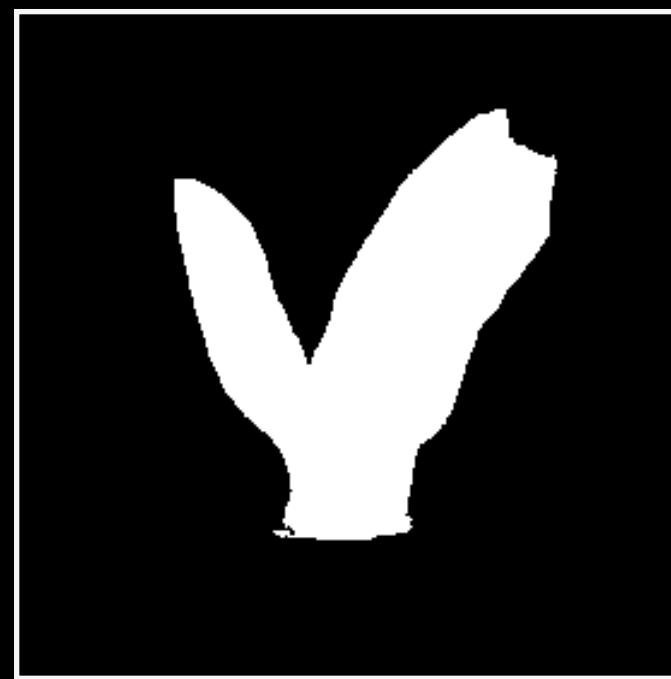
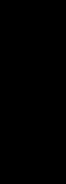
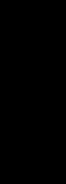
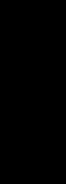
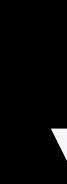
1. To collect 2D stereo images that capture a plant's front, back, left, and right views with 90-degree horizontal rotations
2. To investigate AI architectures, YOLOv8 and Detectron2, and determine which performs better in image segmentation for plant class.
3. To investigate a process that transforms the 2D stereo images of plants into 3D models.



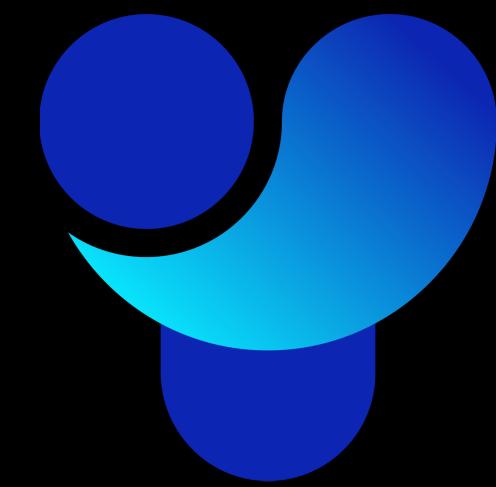




Threshold-based masking



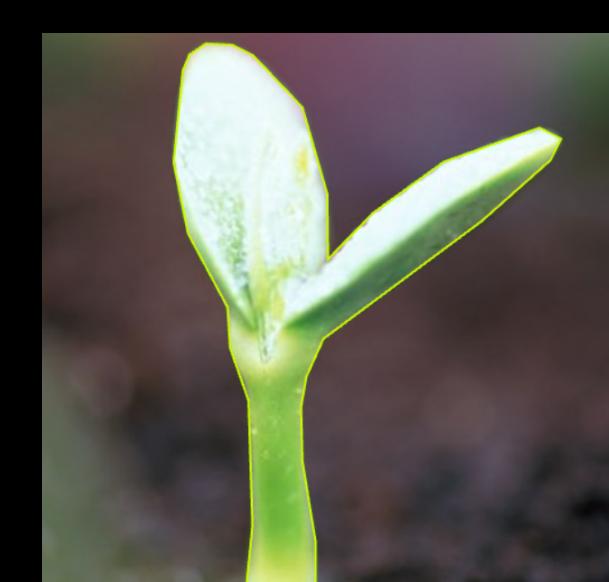
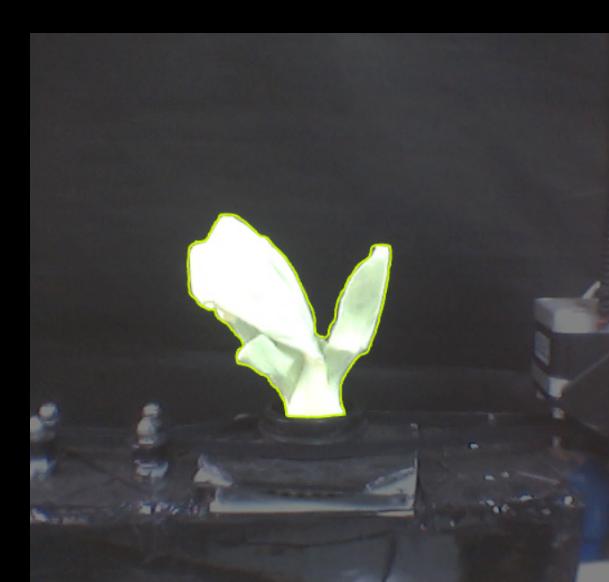
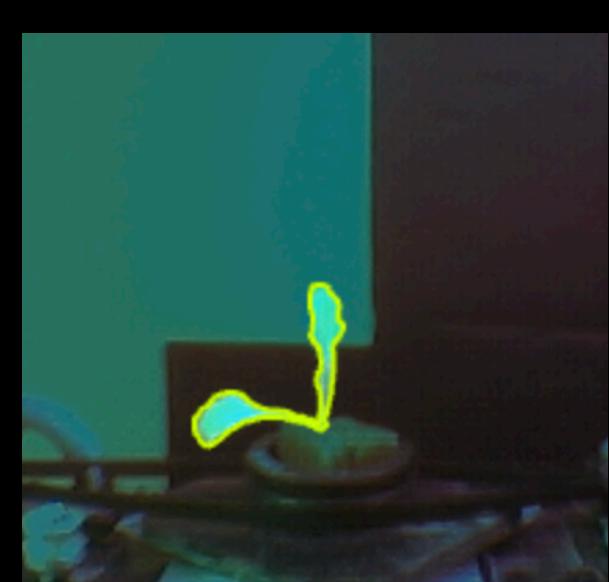
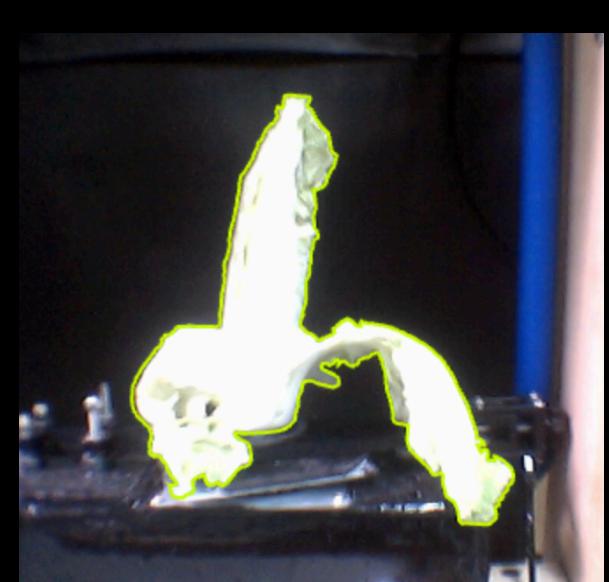
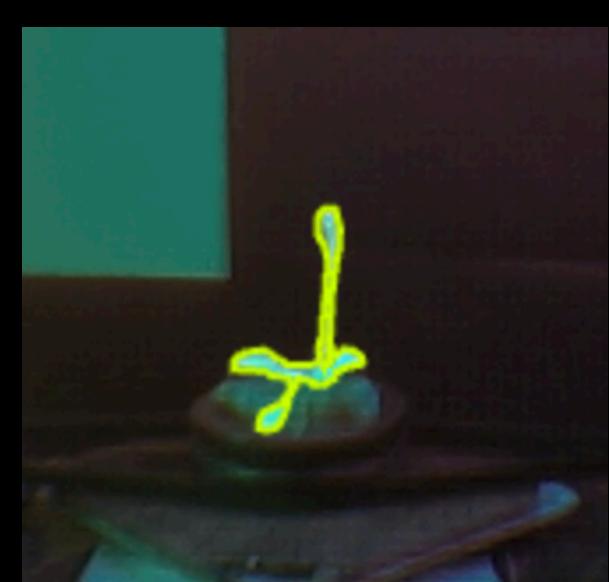
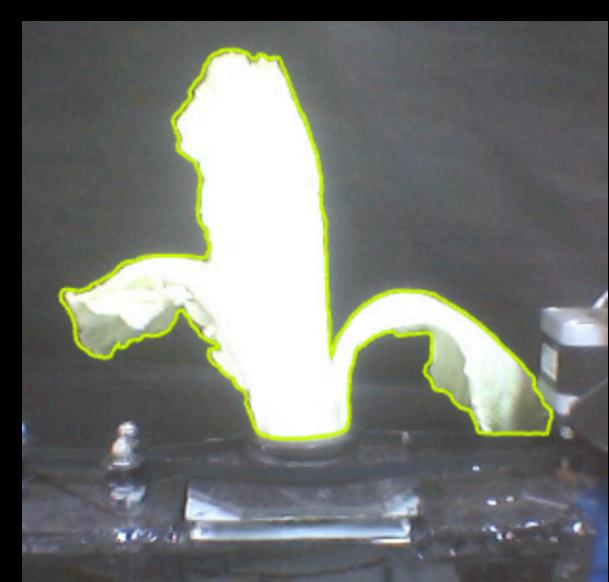
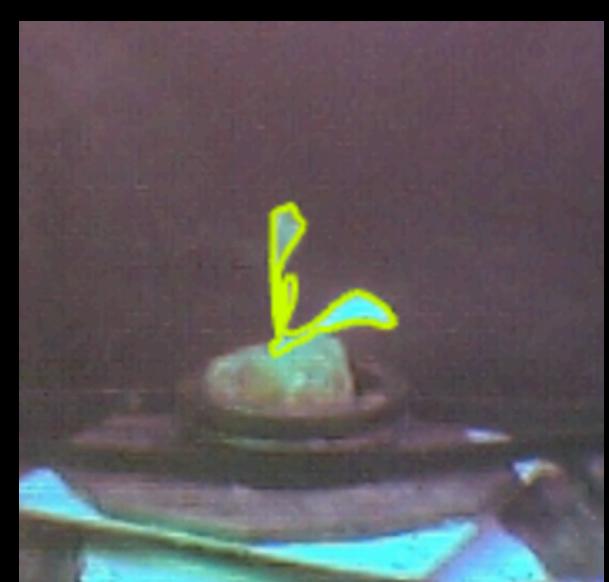
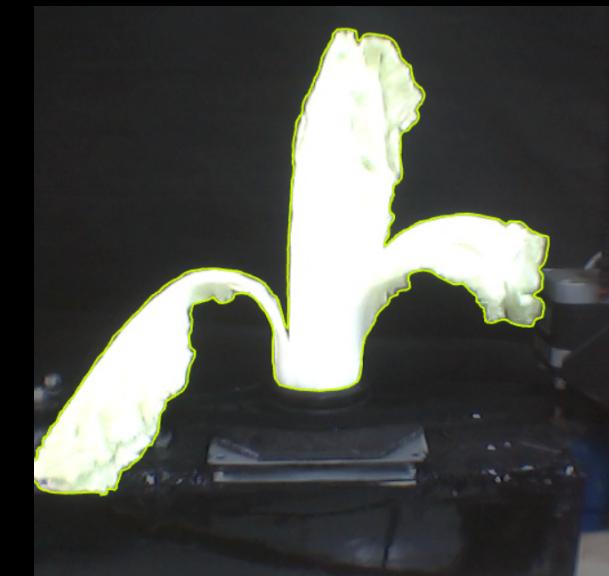
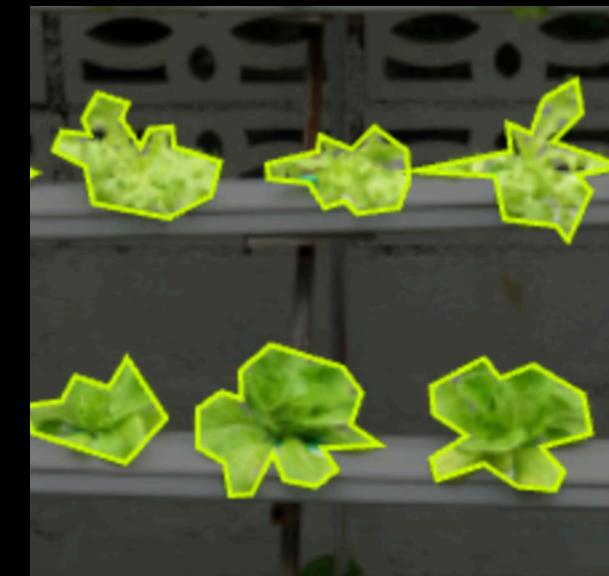
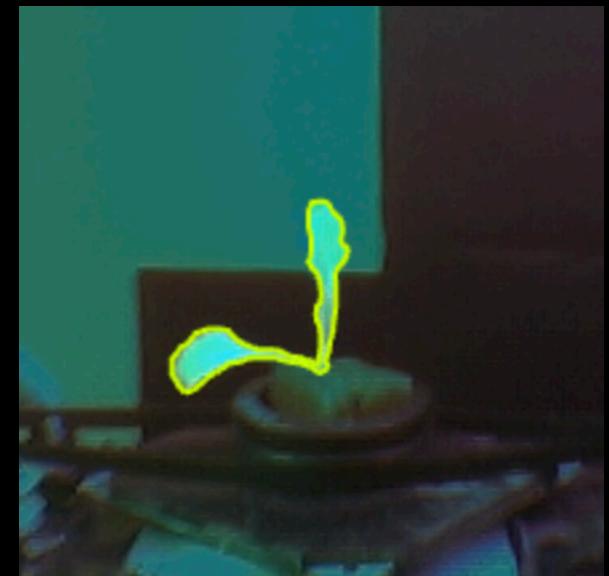
Threshold-based masking



YOLOv8



Detectron2



# EVALUATION

Segmentation  
Accuracy

Training  
Speed

Memory  
Consumption

# EVALUATION

Segmentation  
Accuracy

Box AP/50

Mask AP/50

Box Loss

Mask Loss

Training  
Speed

Train Time

Memory  
Consumption

System RAM

GPU RAM

Disk Usage

# EVALUATION

Hyperparameter Set A

Epoch size = 100, Batch size = 16,  
Worker size = 8, ...

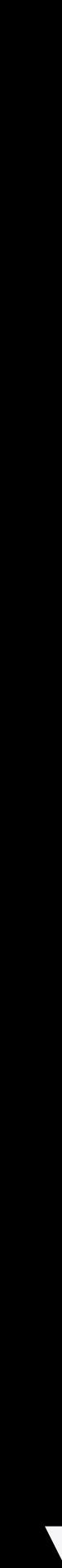
Hyperparameter Set B

Epoch size = 75, Batch size = 8,  
Worker size = 6, ...

Hyperparameter Set C

Epoch size = 50, Batch size = 4,  
Worker size = 4, ...

Decreasing values



# EVALUATION

Hyperparameter Set A

Epoch size = 100, Batch size = 16,  
Worker size = 8, ...

Hyperparameter Set B

Epoch size = 75, Batch size = 8,  
Worker size = 6, ...

Hyperparameter Set C

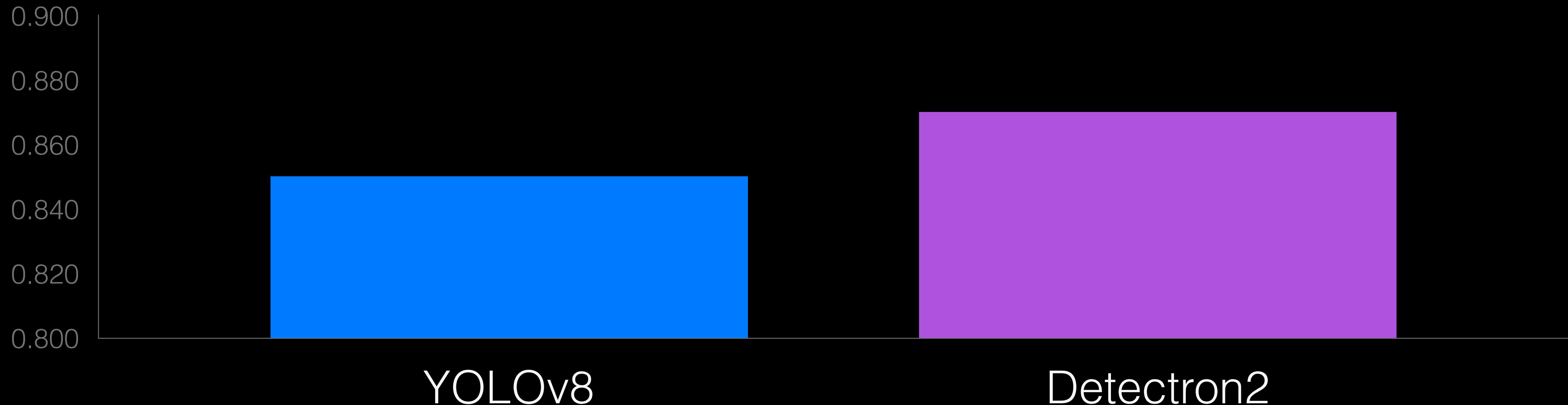
Epoch size = 50, Batch size = 4,  
Worker size = 4, ...

Decreasing values



# EVALUATION

## Segmentation Accuracy

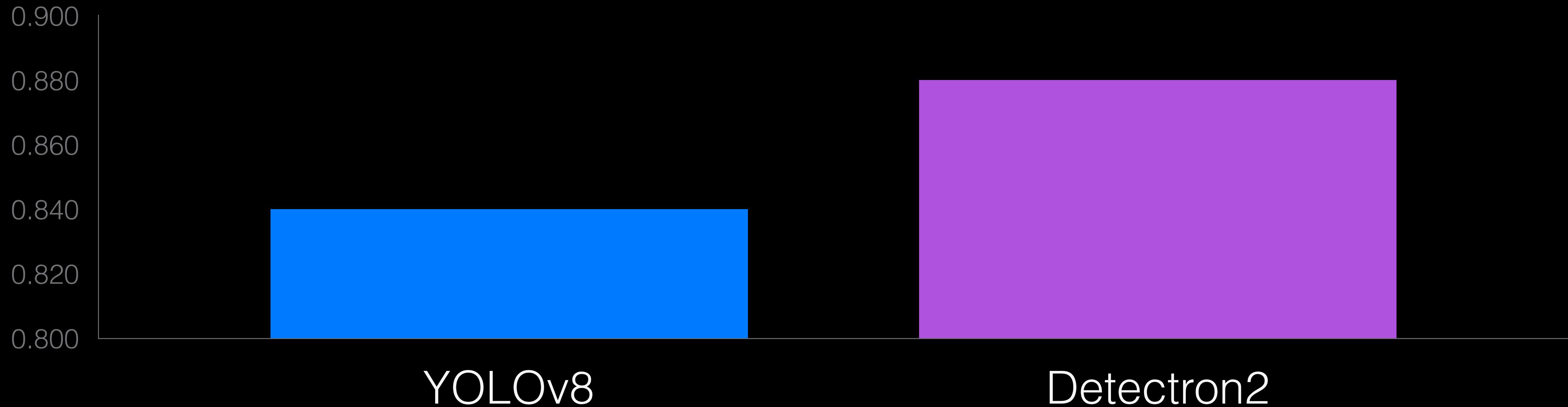


Box Average Precision at 50% IoU threshold

Higher means better in detecting and localizing objects.

# EVALUATION

## Segmentation Accuracy

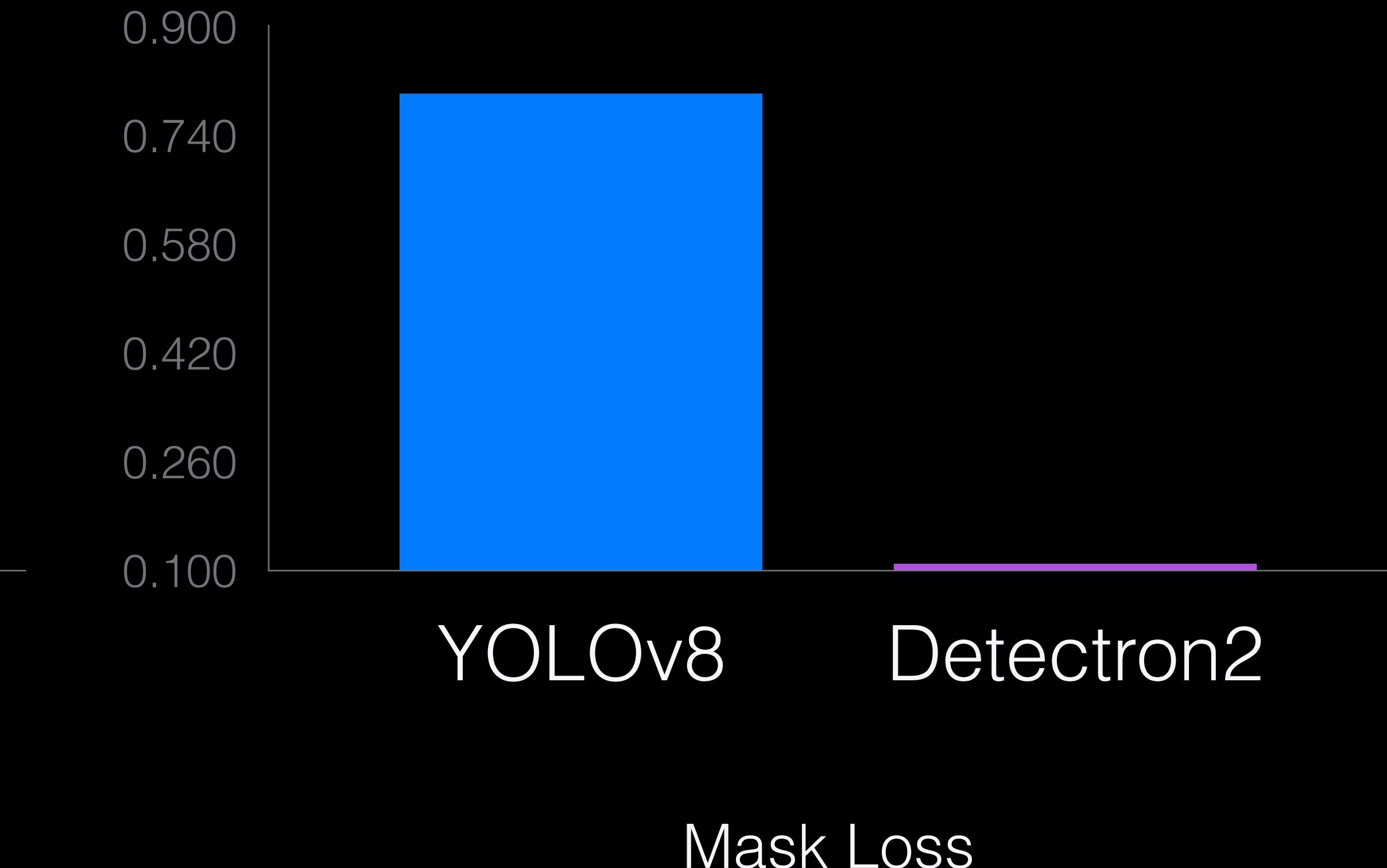
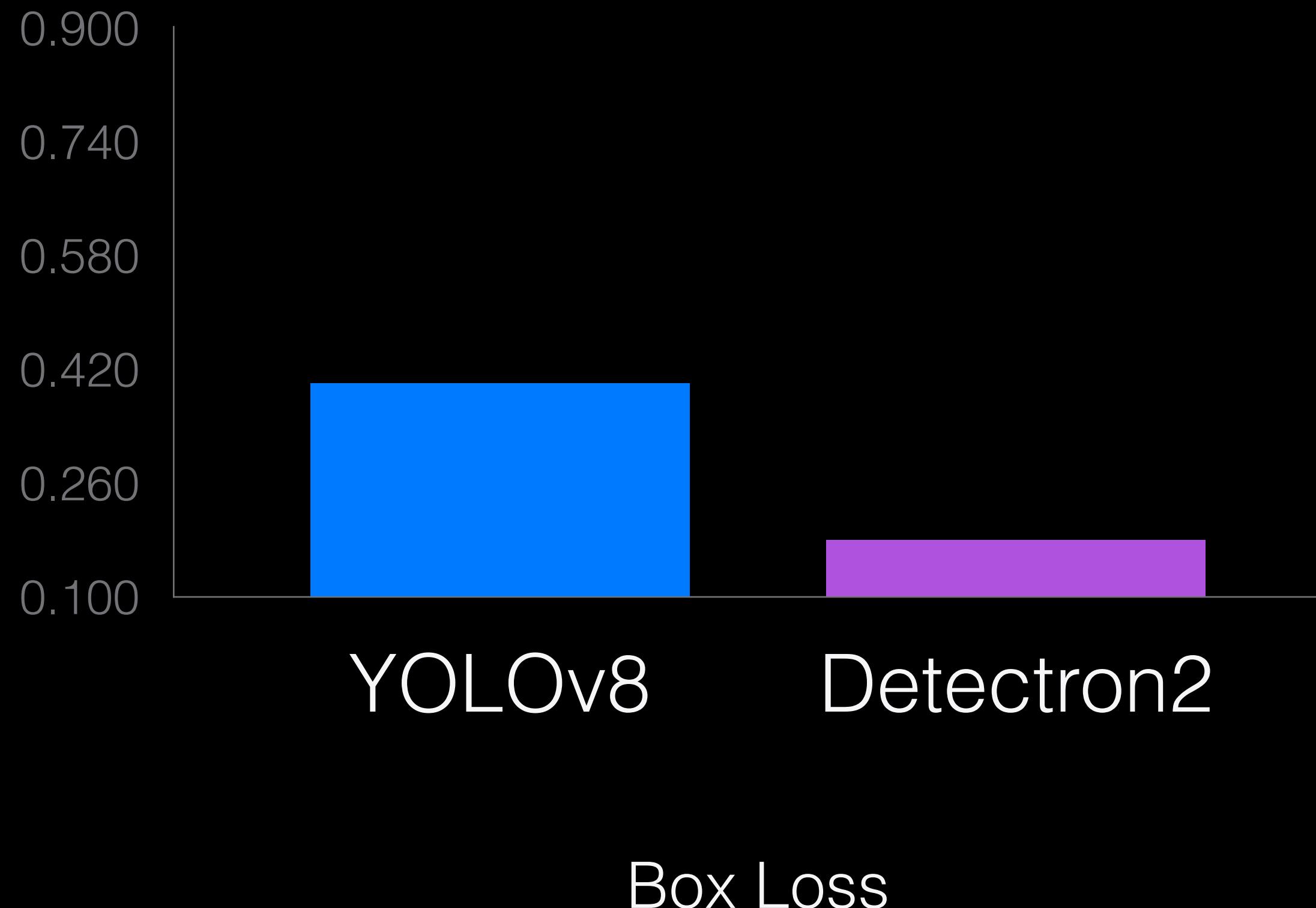


Mask Average Precision at 50% IoU threshold

Higher means better in detecting and outlining objects.

# EVALUATION

## Segmentation Accuracy



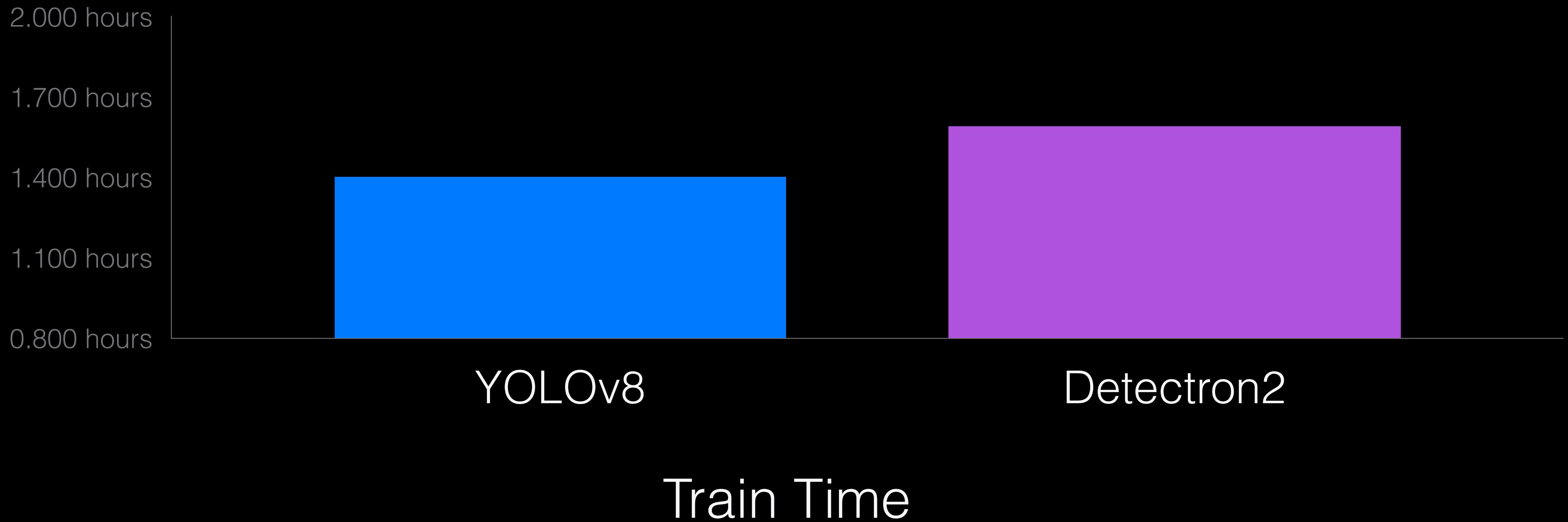
Lower is better.

Detectron2 segments better  
with higher Box and Mask Average Precisions, and  
lower Box and Mask Losses.

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

# EVALUATION

## Training Speed



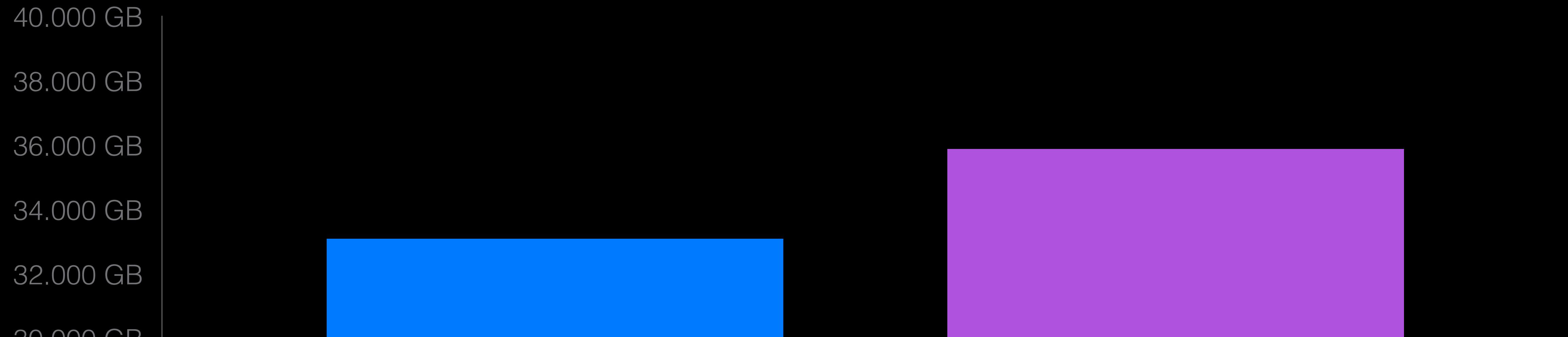
Lower means faster model training and quicker iteration.

# YOLOv8 trains faster with lower Train Time.

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

# EVALUATION

## Memory Consumption



YOLOv8

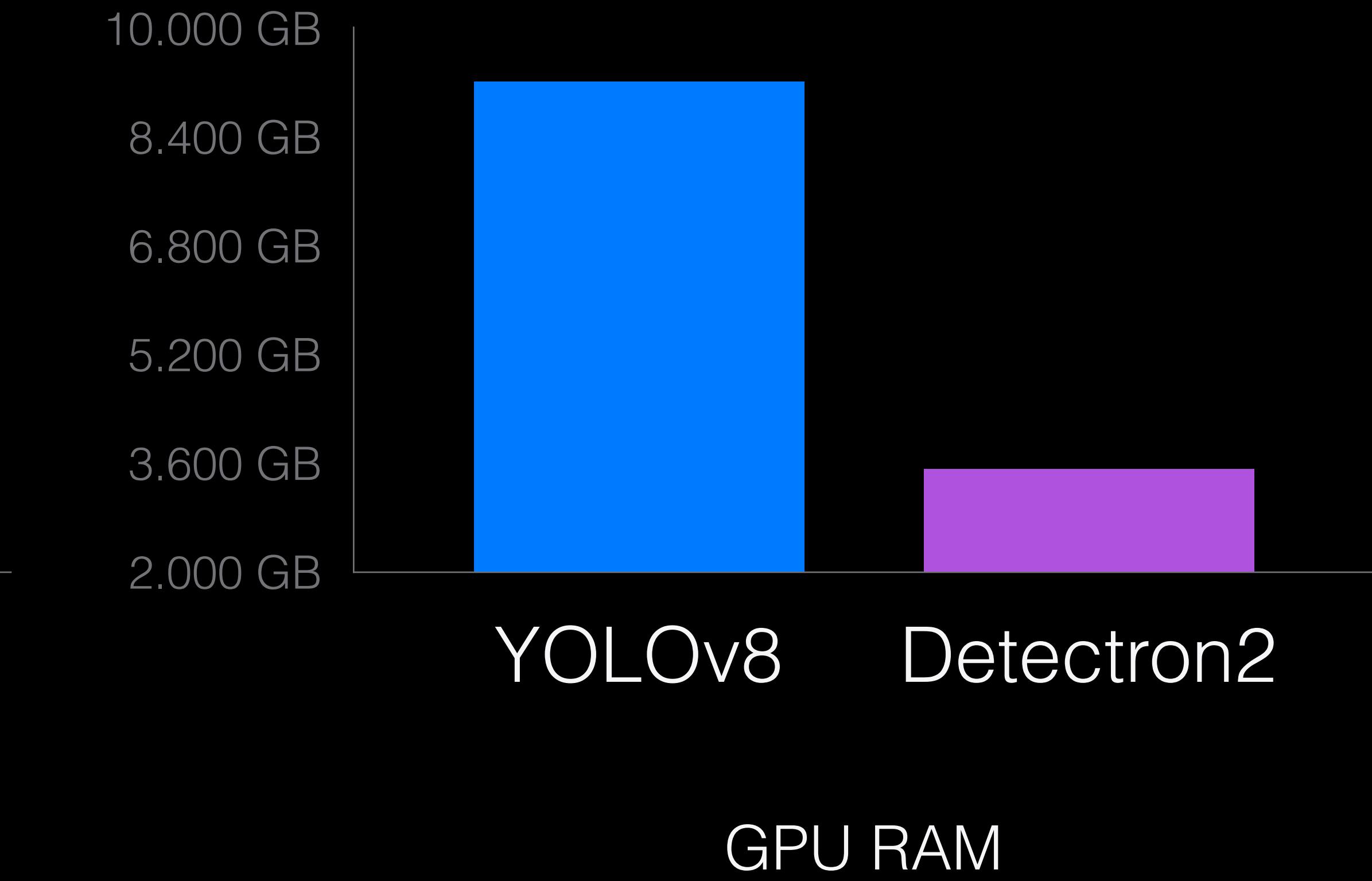
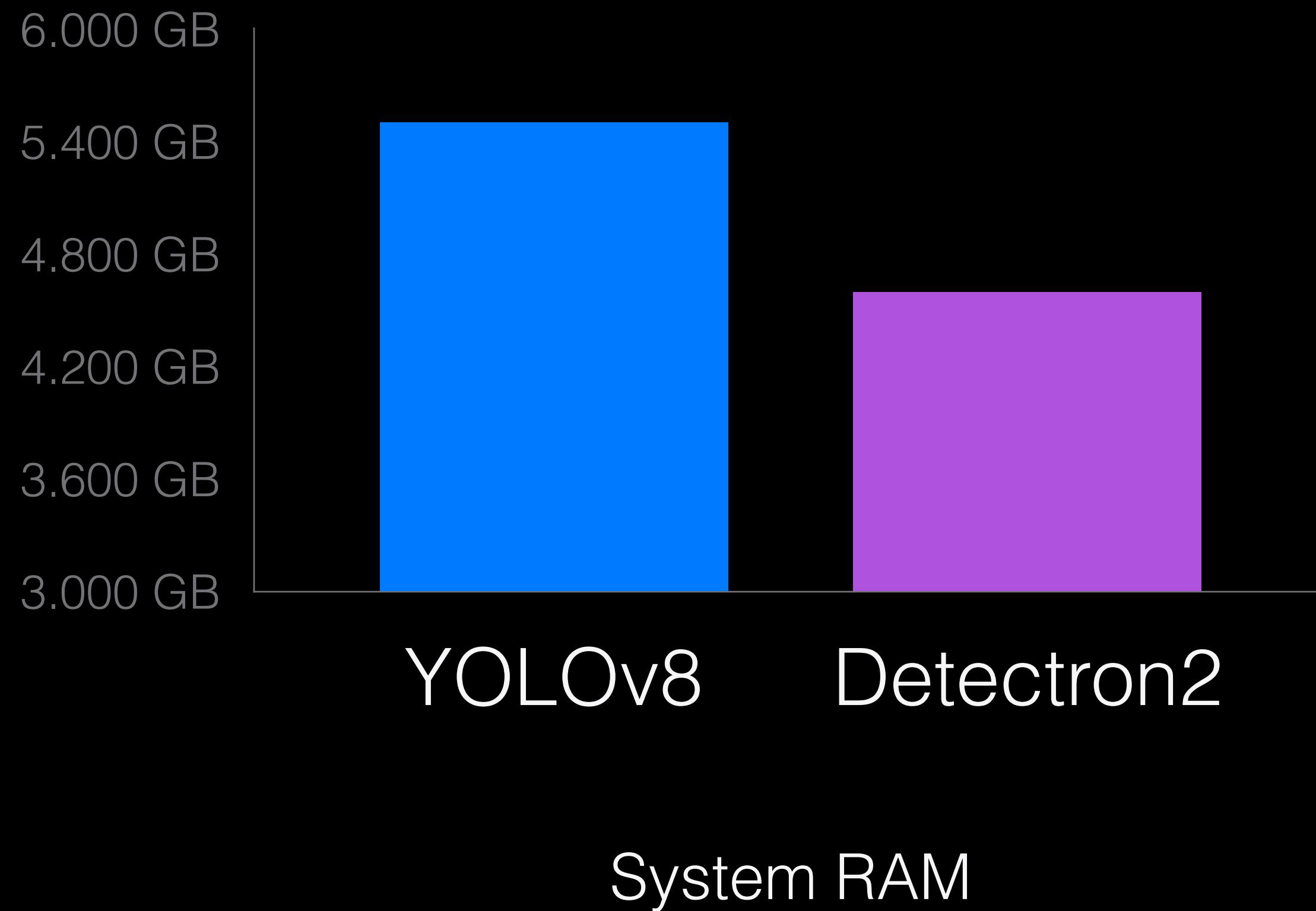
Detectron2

Disk Usage

Lower is better.

# EVALUATION

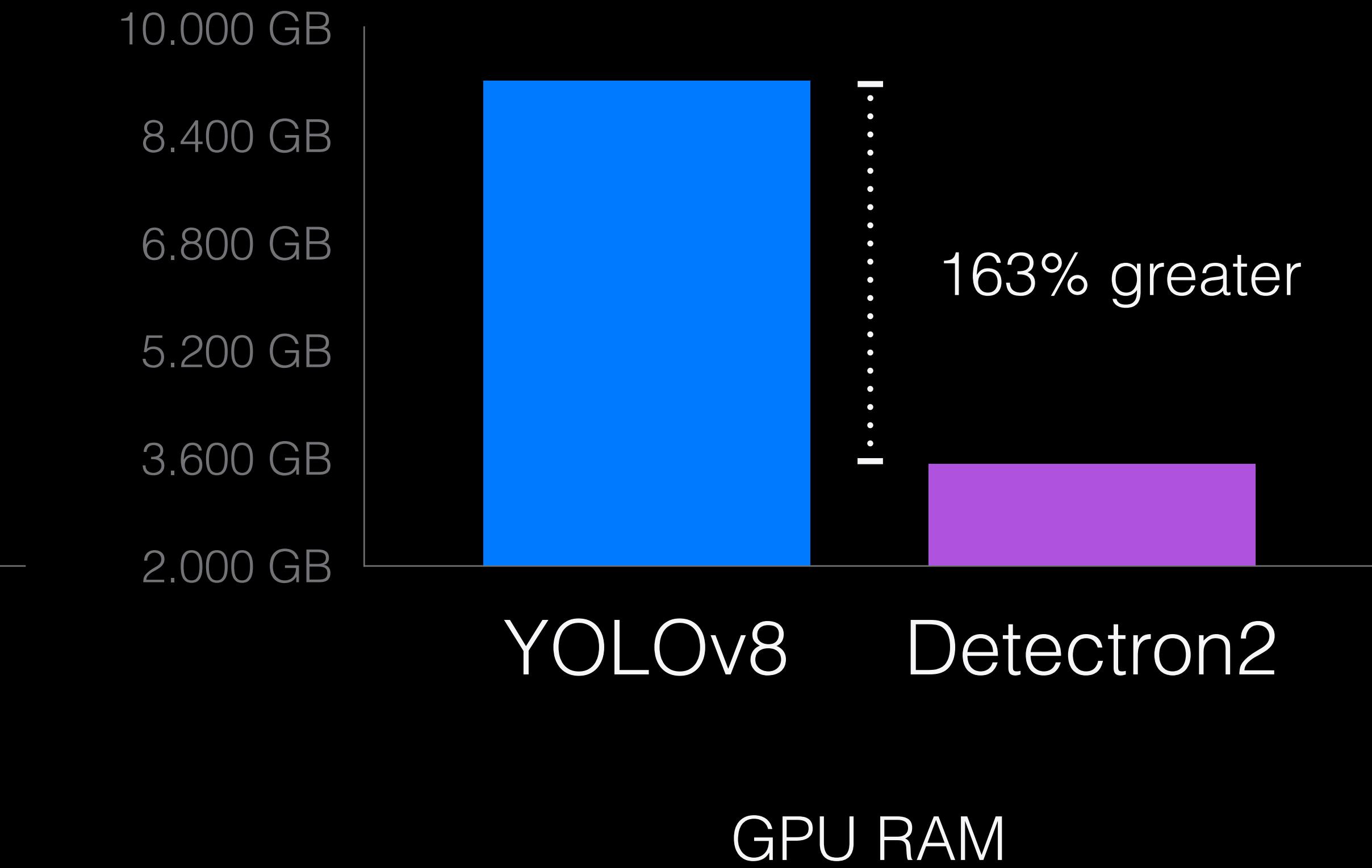
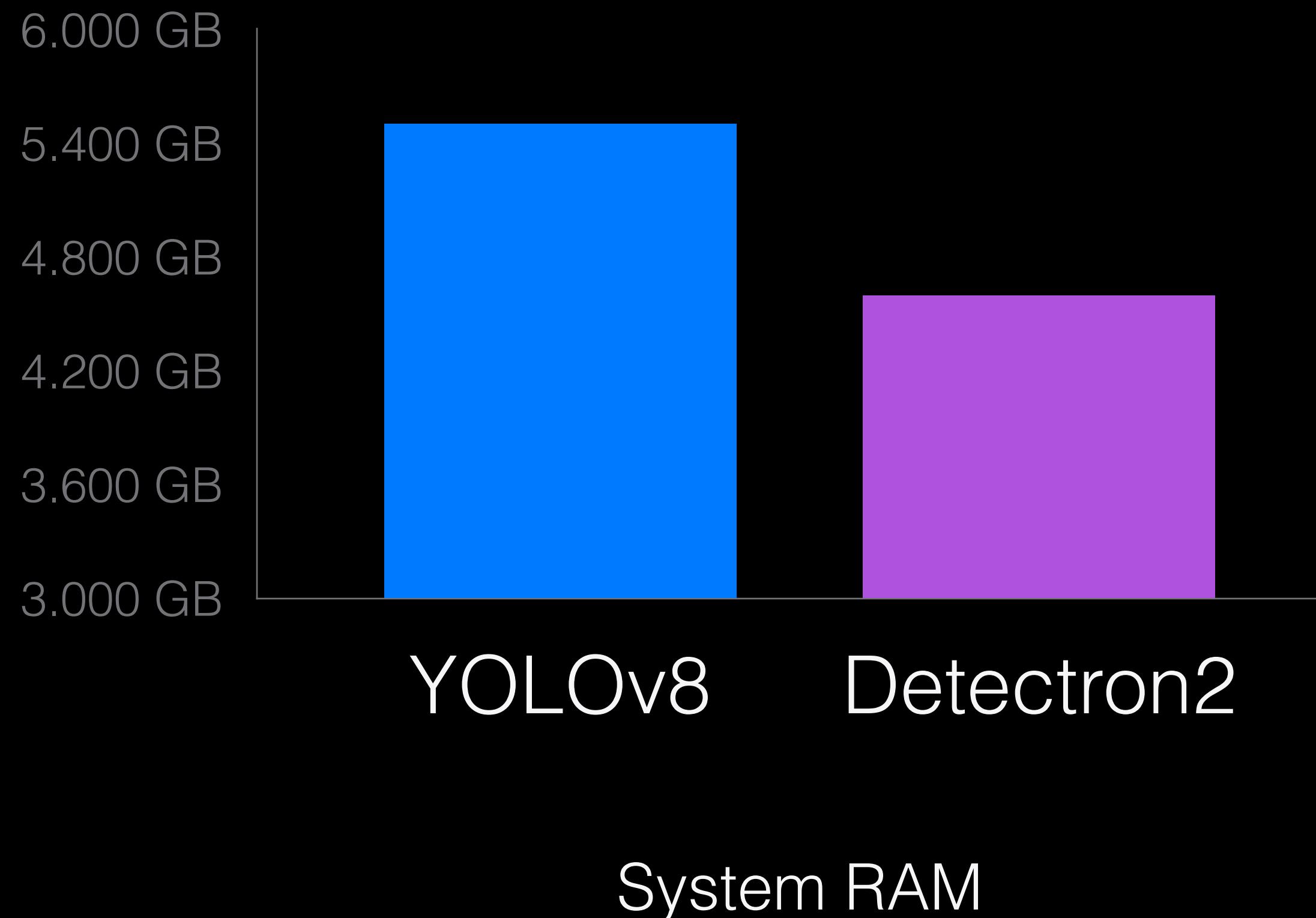
## Memory Consumption



Lower is better.

# EVALUATION

## Memory Consumption



Lower is better.

**Detectron2 is RAM-efficient  
with lower System RAM and GPU RAM consumptions.**

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

# EVALUATION

Hyperparameter Set A

Epoch size = 100, Batch size = 16,  
Worker size = 8, ...

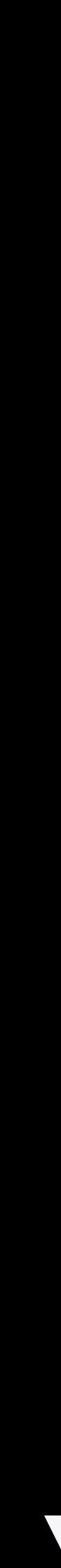
Hyperparameter Set B

Epoch size = 75, Batch size = 8,  
Worker size = 6, ...

Hyperparameter Set C

Epoch size = 50, Batch size = 4,  
Worker size = 4, ...

Decreasing values



# EVALUATION

Hyperparameter Set A

Epoch size = 100, Batch size = 16,  
Worker size = 8, ...

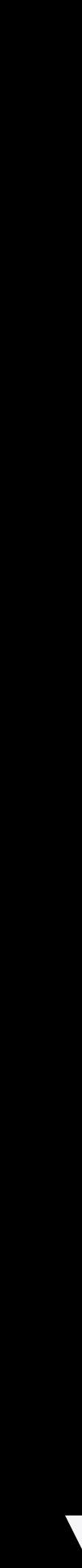
Hyperparameter Set B

Epoch size = 75, Batch size = 8,  
Worker size = 6, ...

Hyperparameter Set C

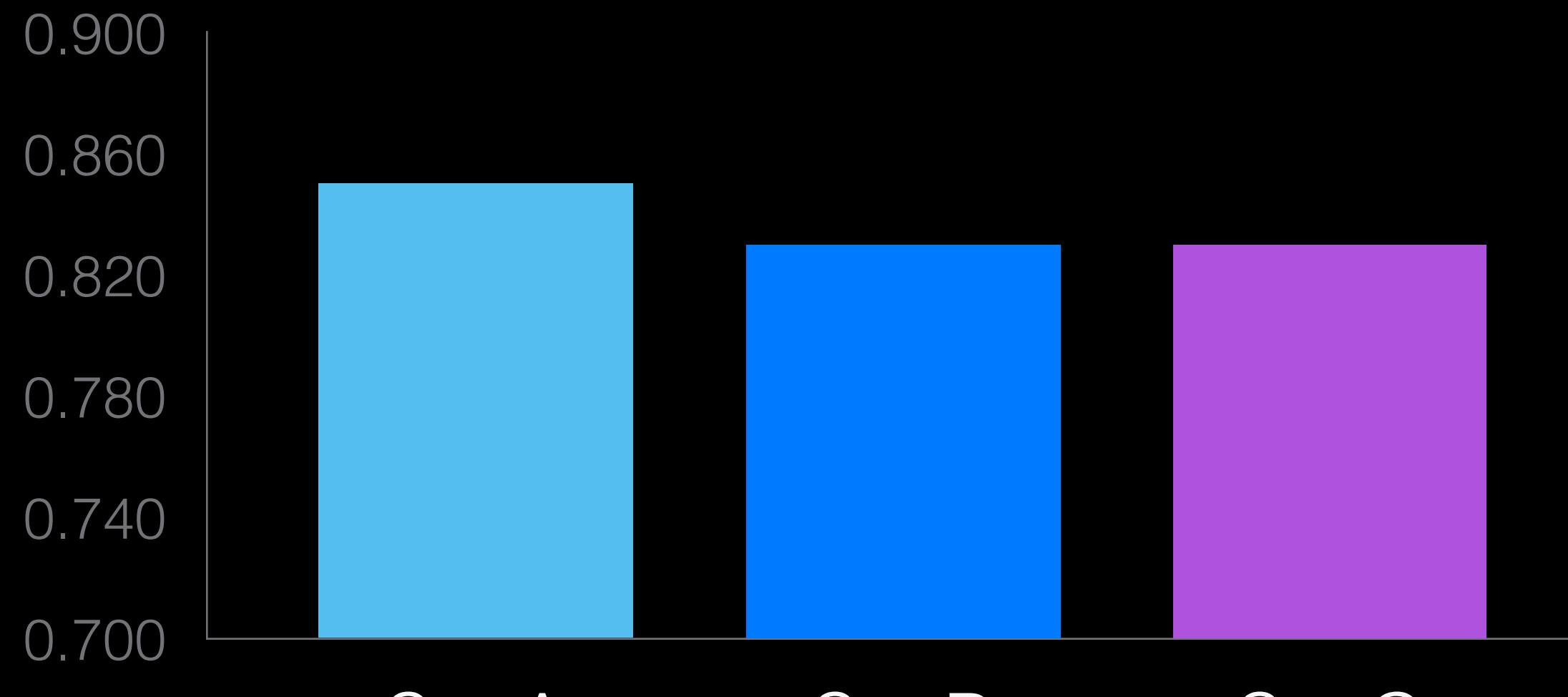
Epoch size = 50, Batch size = 4,  
Worker size = 4, ...

Decreasing values



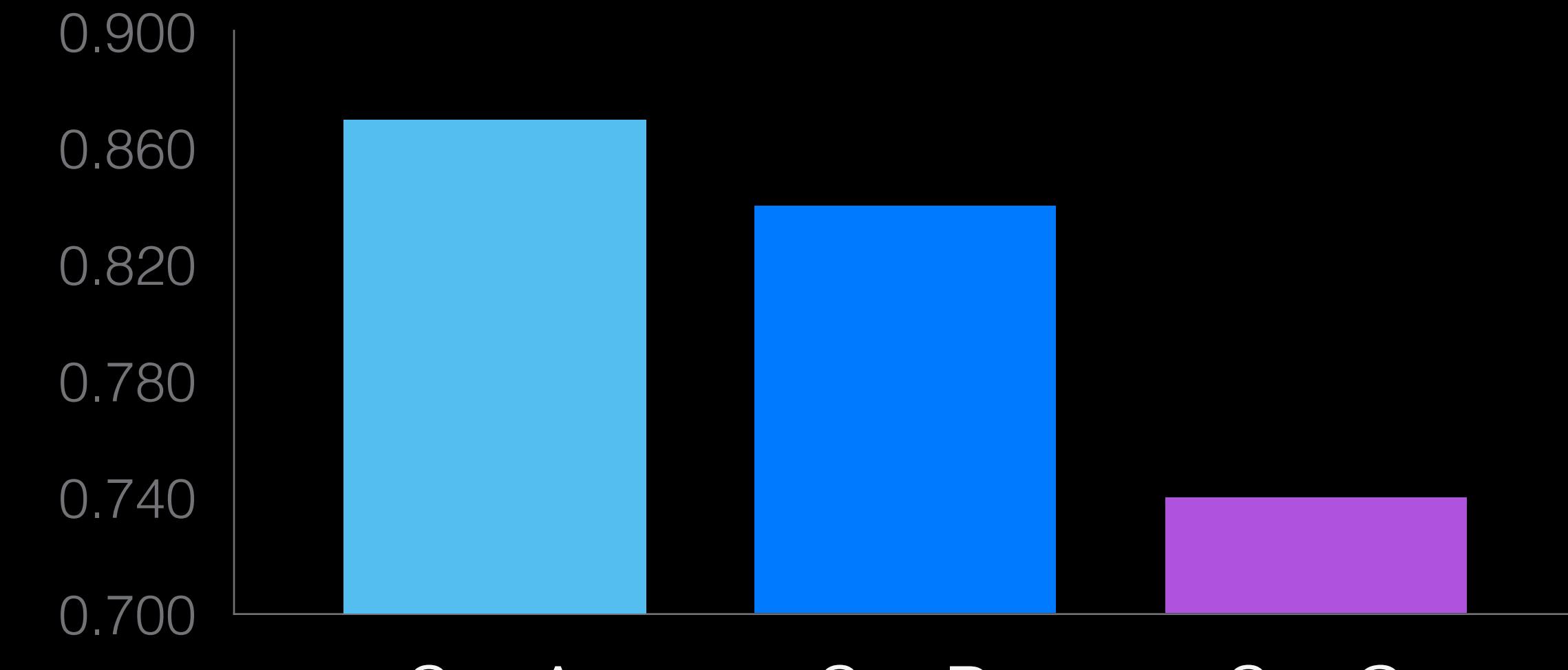
# EVALUATION

## Segmentation Accuracy



Hyperparameters

YOLOv8



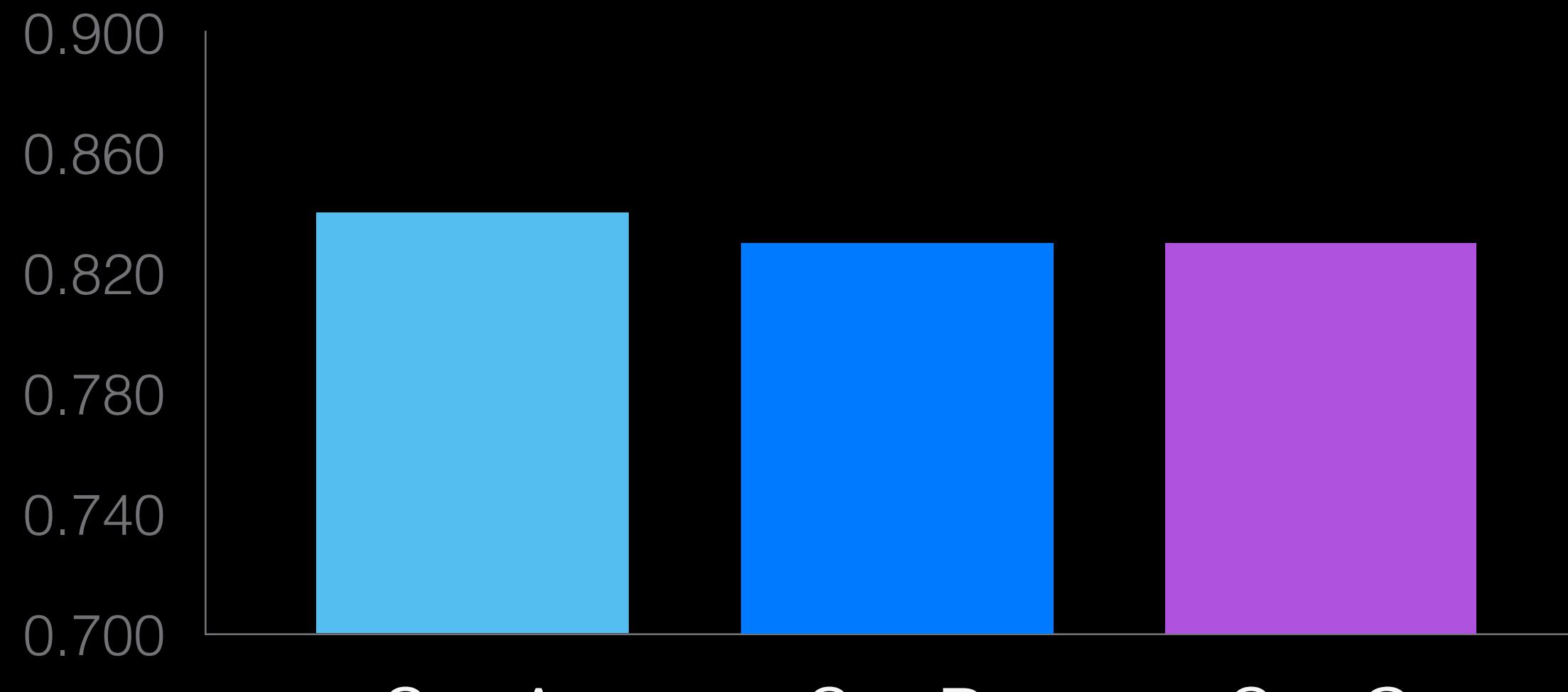
Hyperparameters

Detectron2

Box Average Precision at 50% IoU threshold

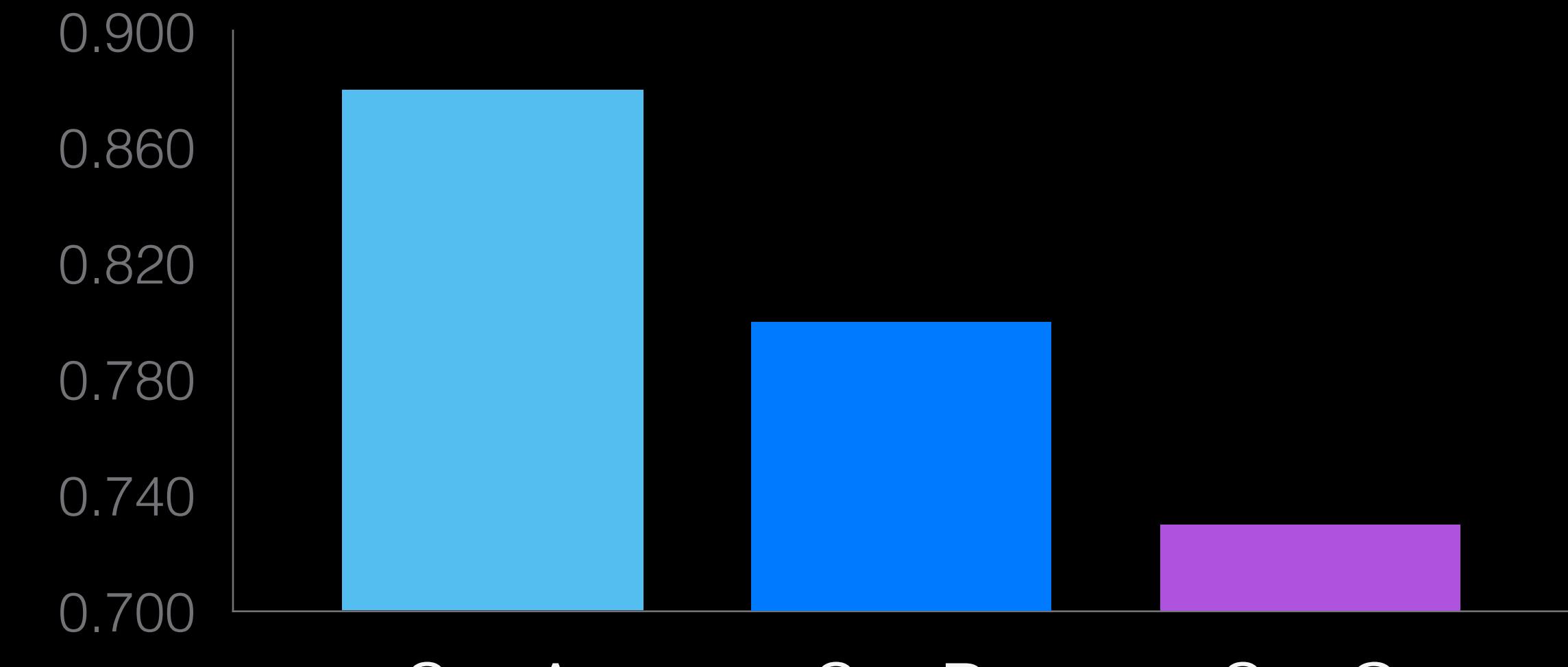
# EVALUATION

## Segmentation Accuracy



Hyperparameters

YOLOv8



Hyperparameters

Detectron2

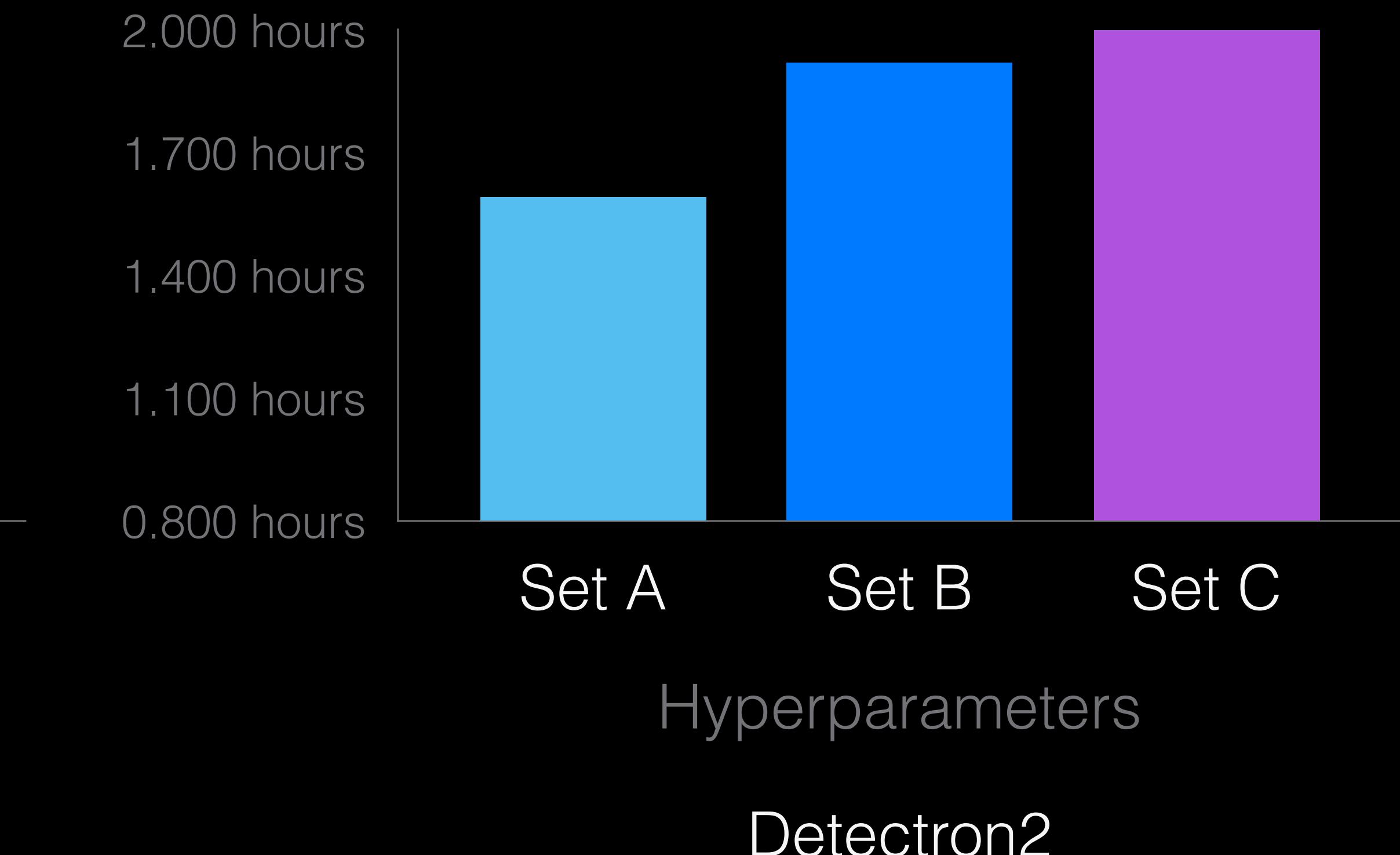
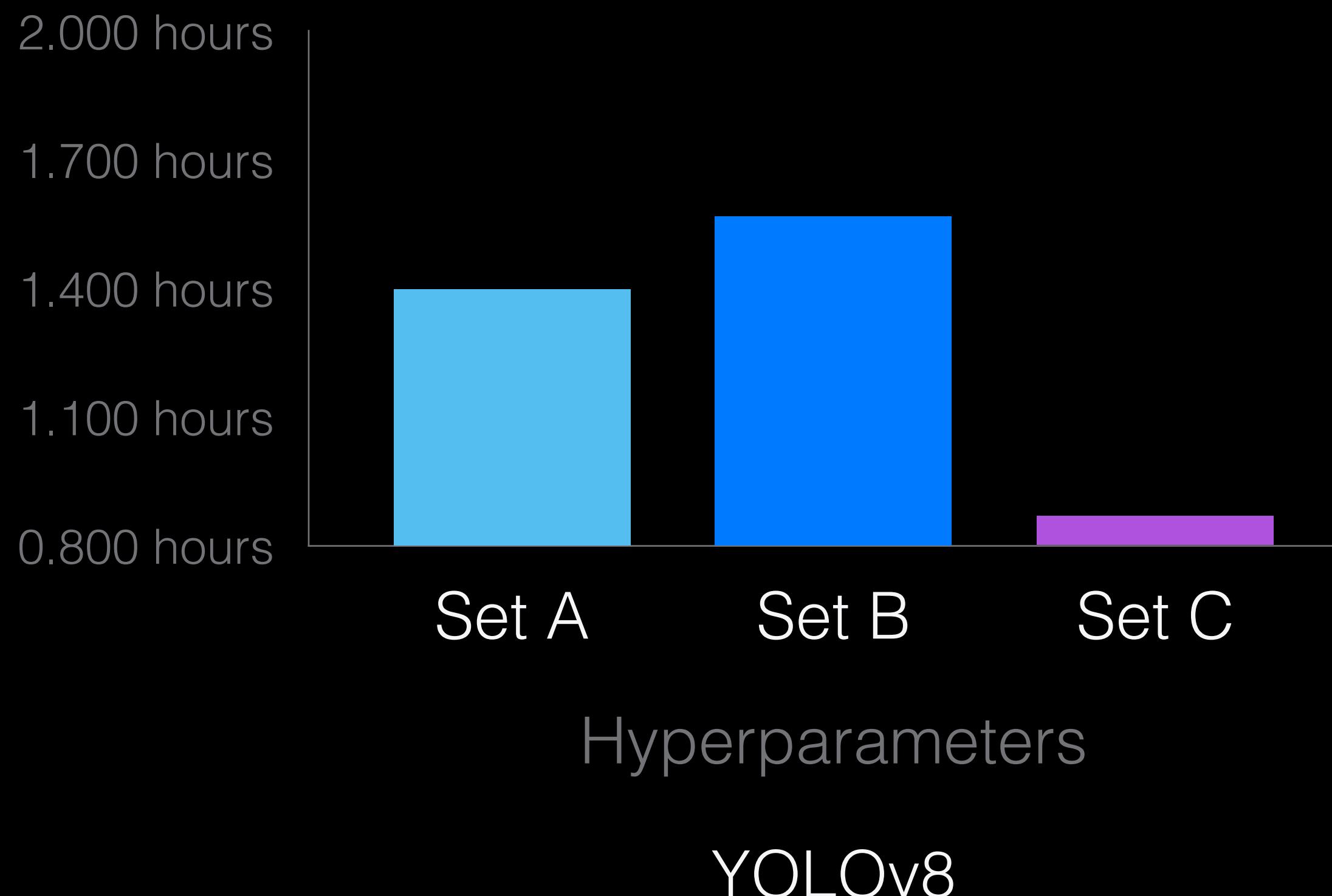
Mask Average Precision at 50% IoU threshold

Increasing hyperparameter values  
increase segmentation accuracy on both  
models.

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

# EVALUATION

## Training Speed



Train Time

Decreasing hyperparameter values train  
faster if using YOLOv8.

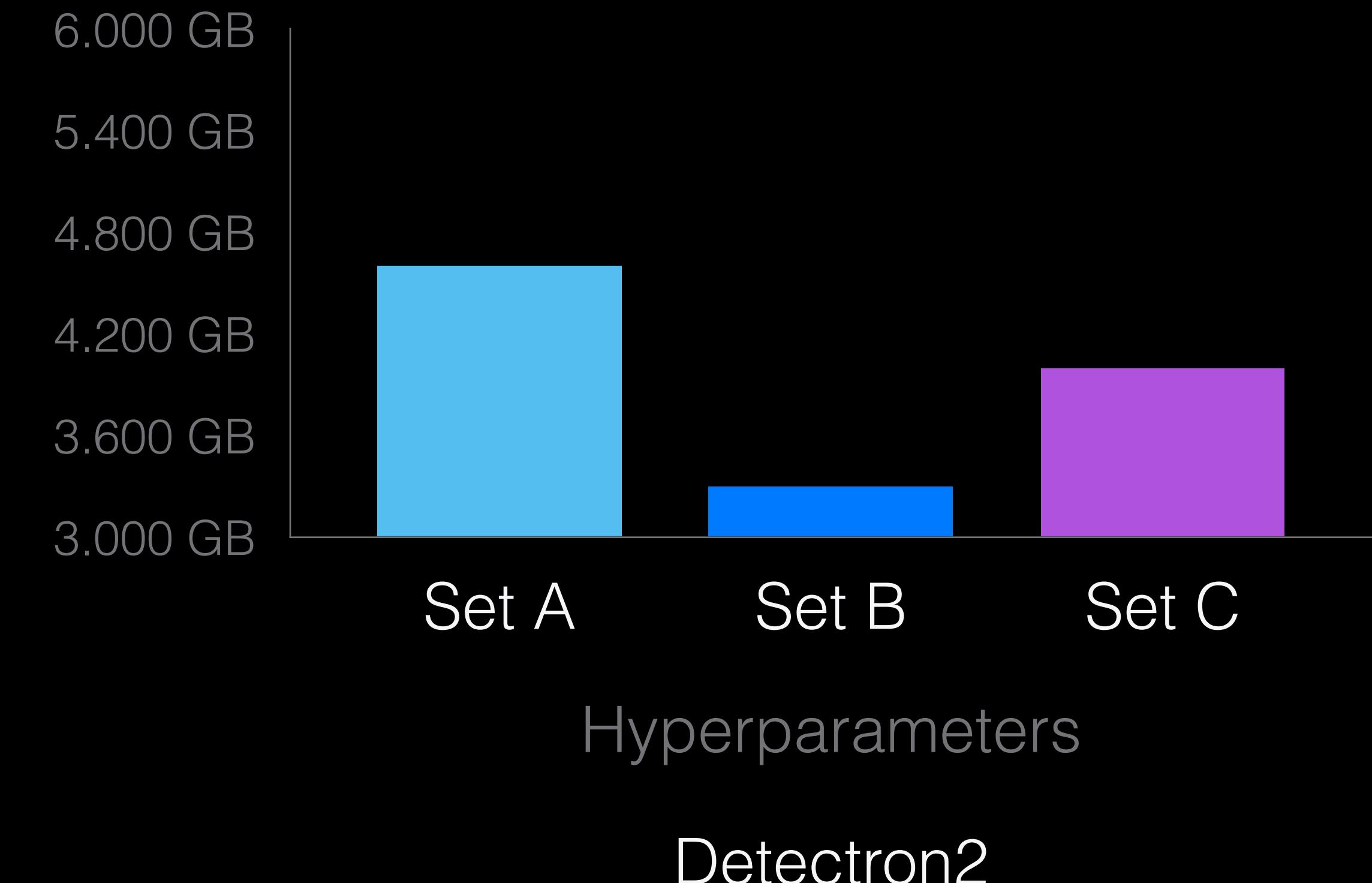
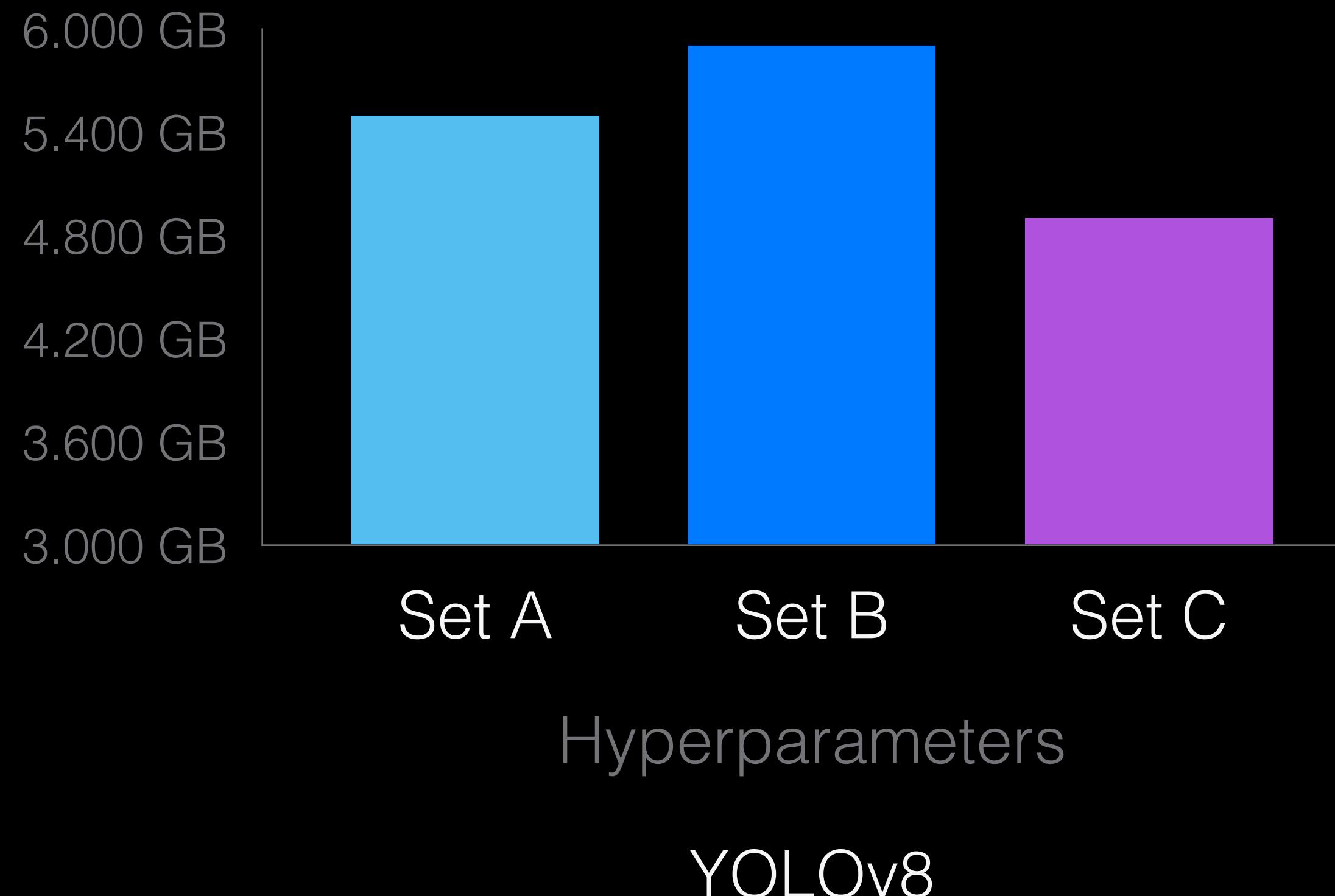
Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

If using Detectron2, it is important to note that decreasing hyperparameter values makes training slower instead.

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.

# EVALUATION

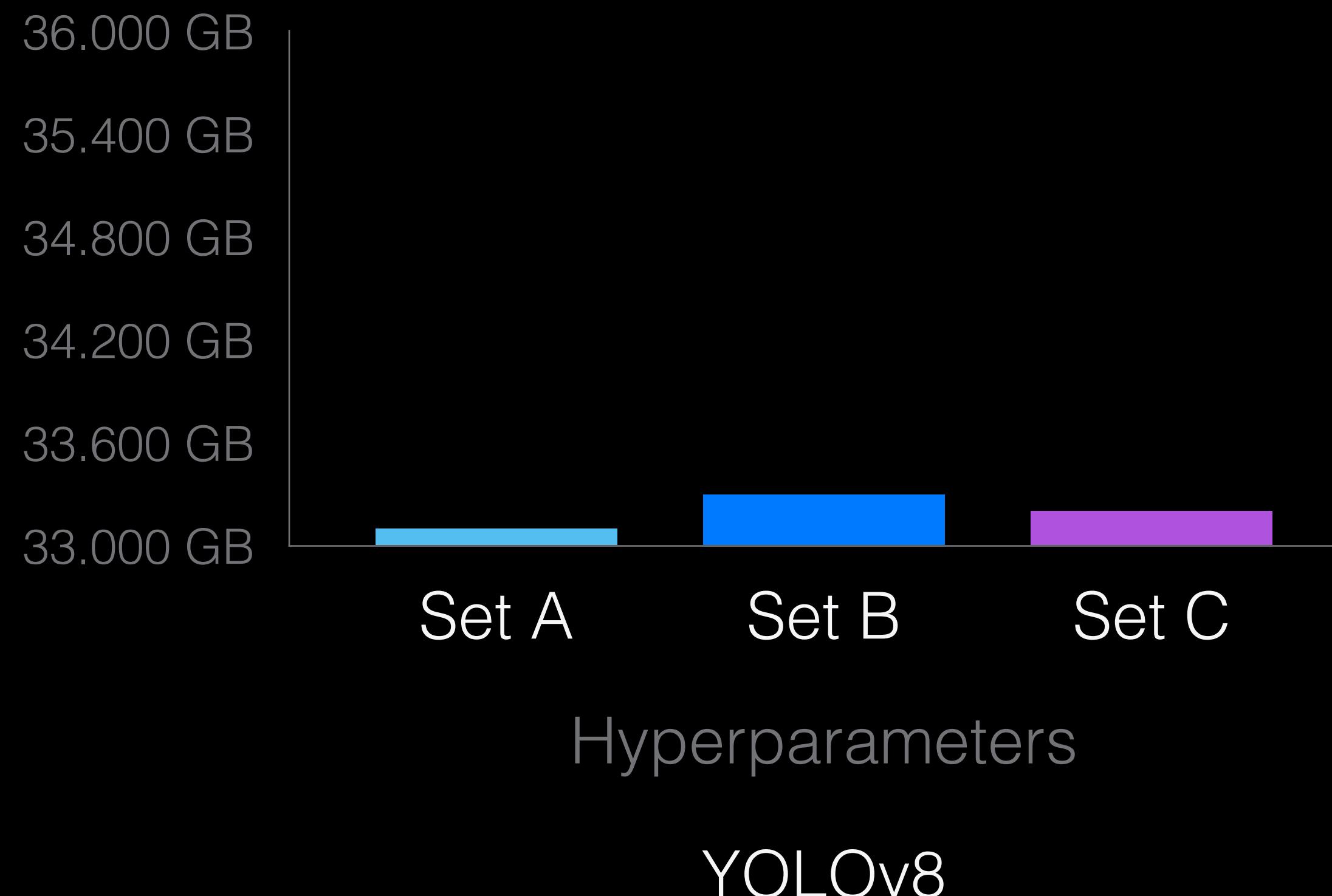
## Memory Consumption



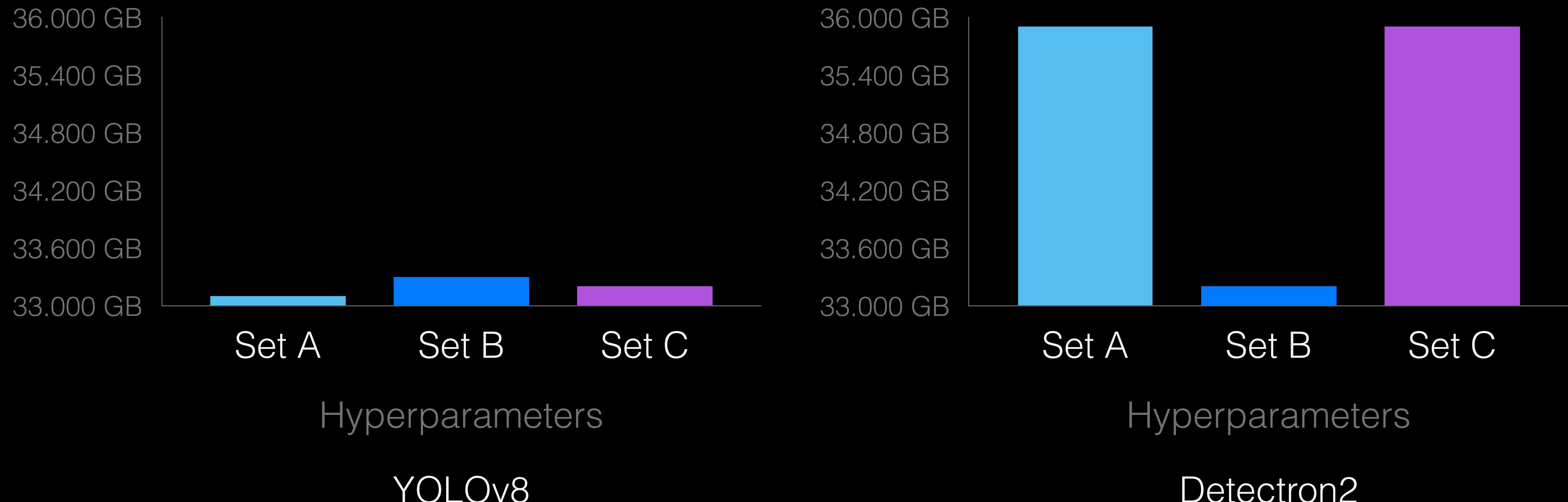
System RAM consumption

# EVALUATION

## Memory Consumption

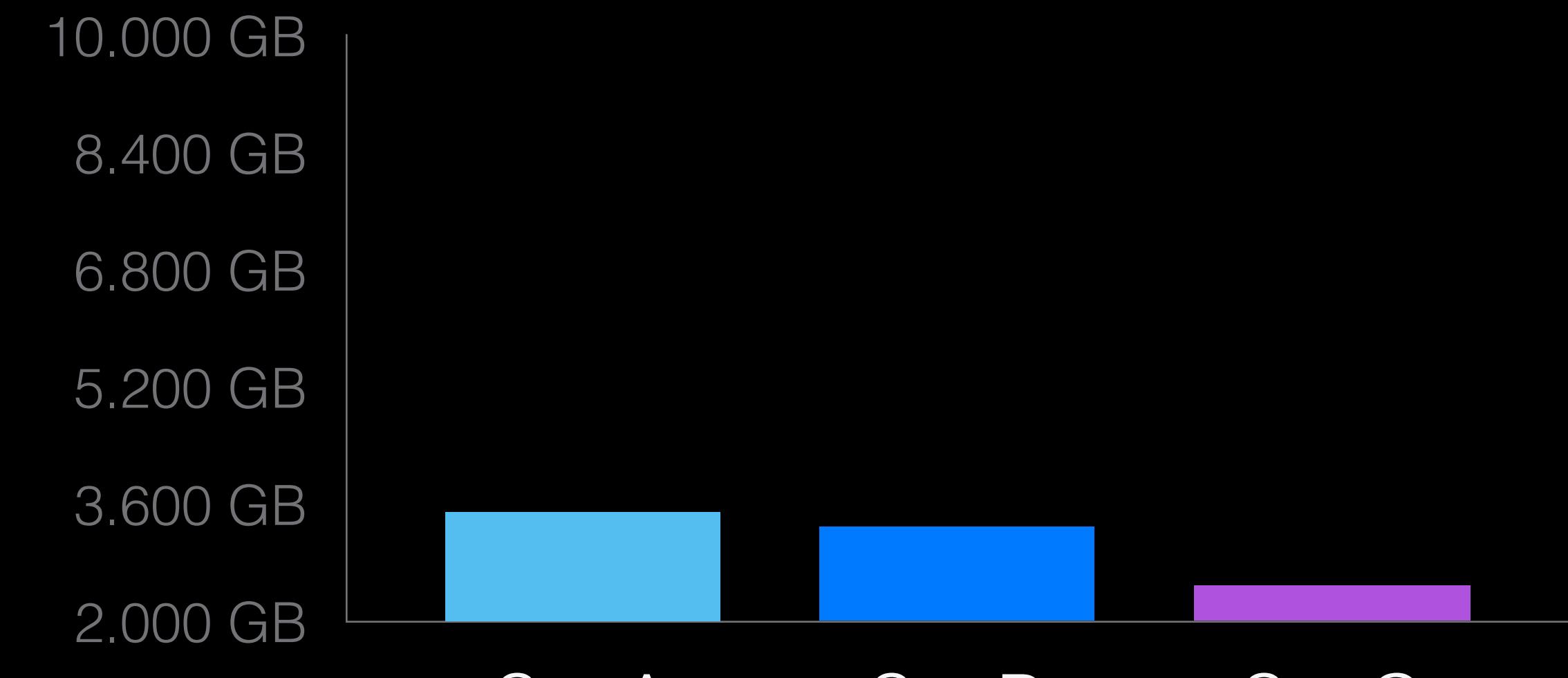
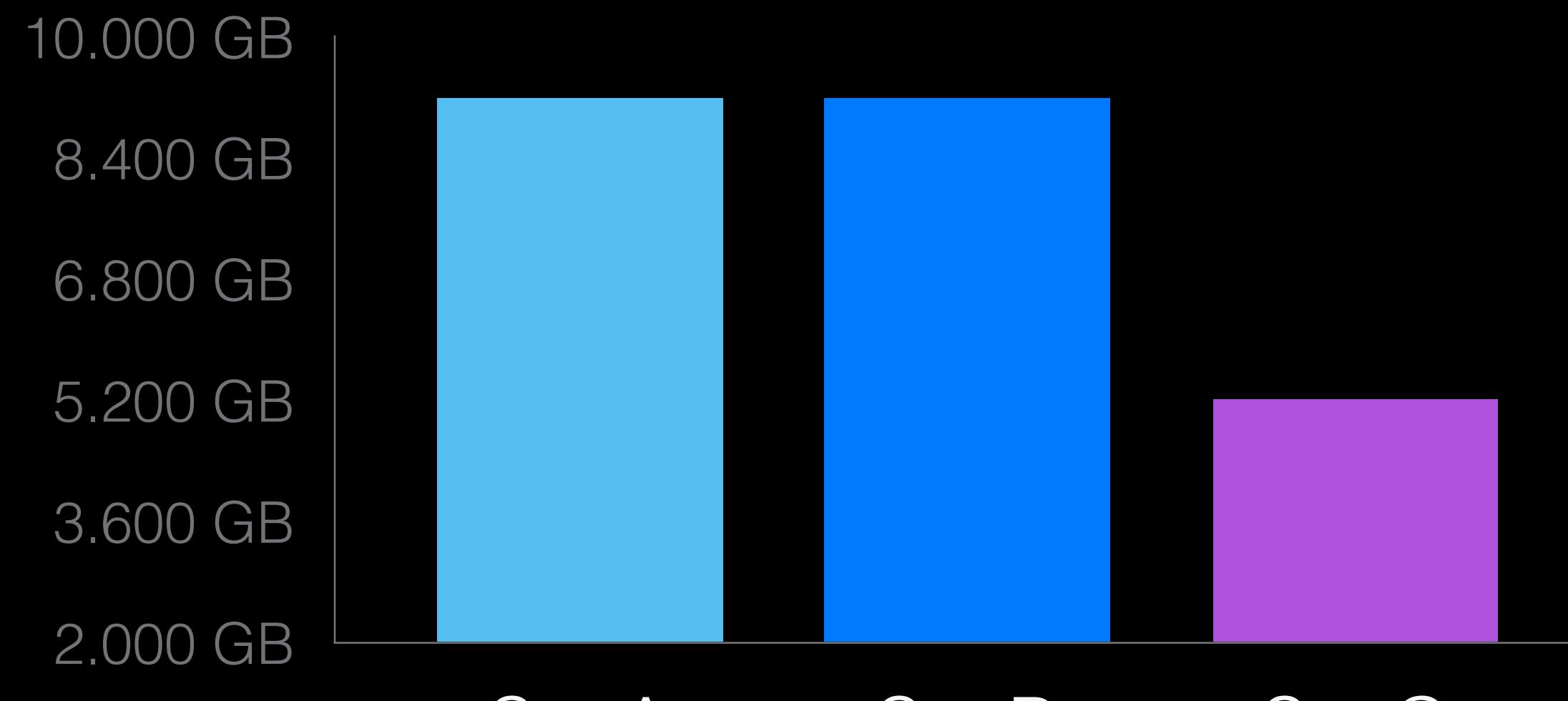


Disk Usage



# EVALUATION

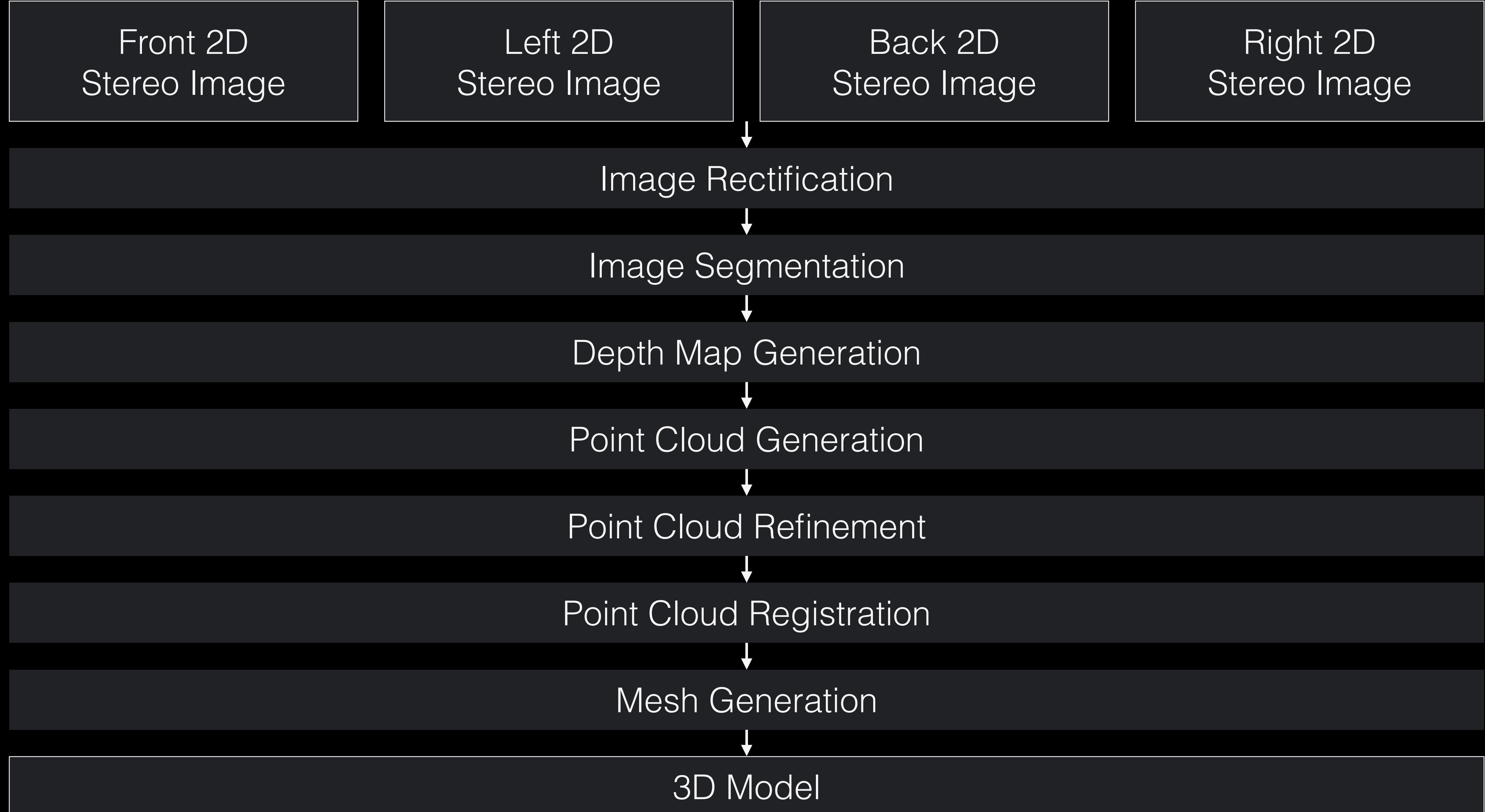
## Memory Consumption



GPU RAM consumption

Decreasing hyperparameter values lower  
GPU requirements on both models.

Results are based on the plant class, with specified hyperparameter values, and the images and size of the dataset used for training.



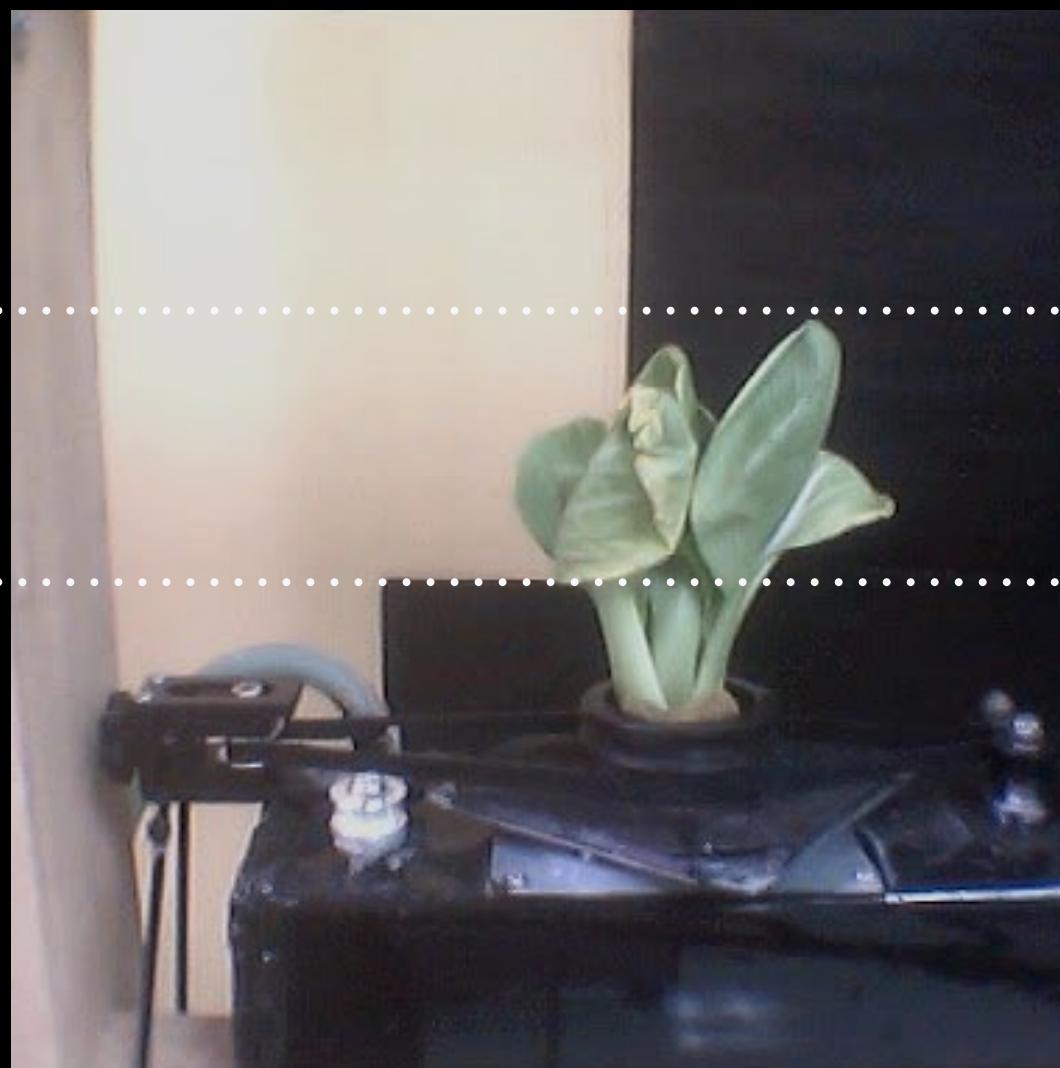


# DEPTH MAP GENERATION

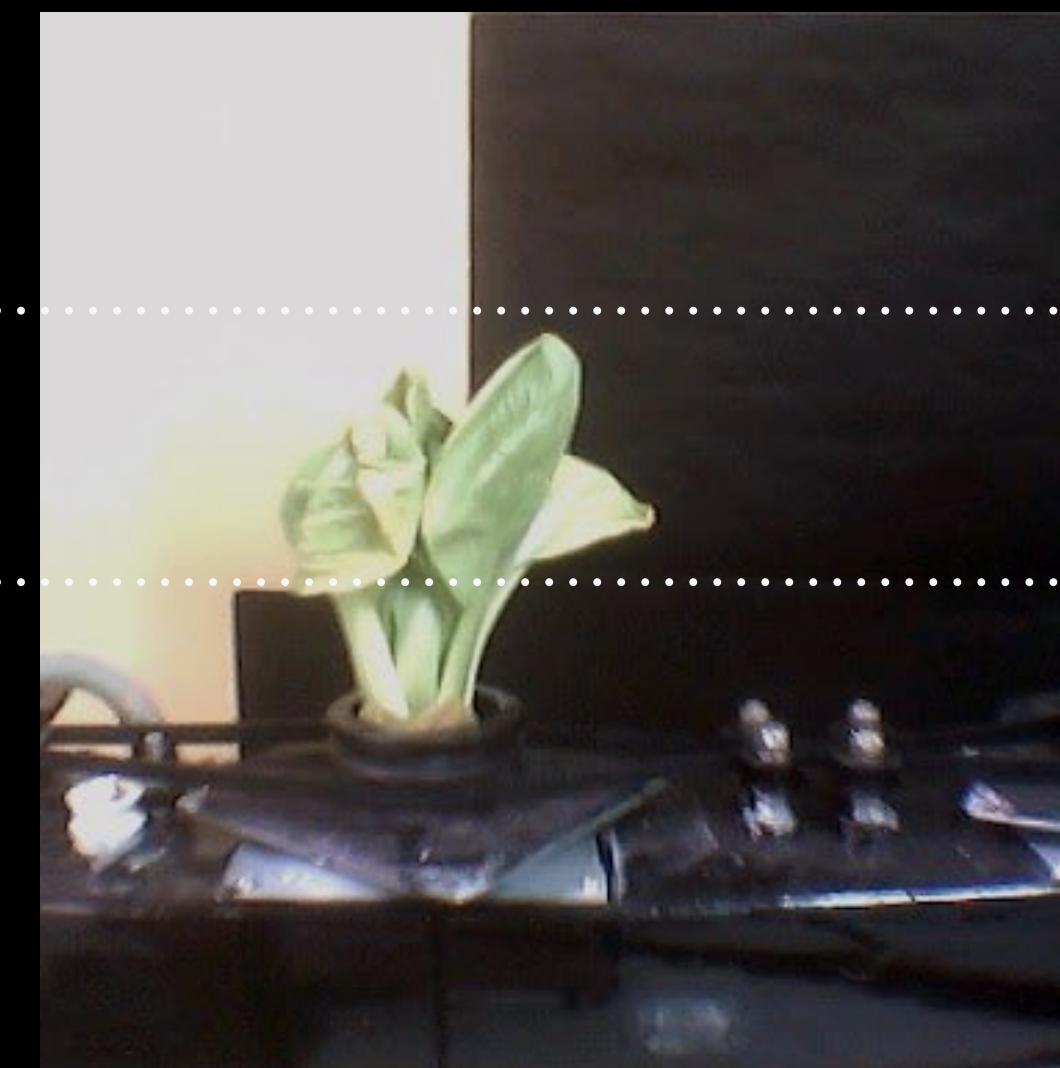


# DEPTH MAP GENERATION

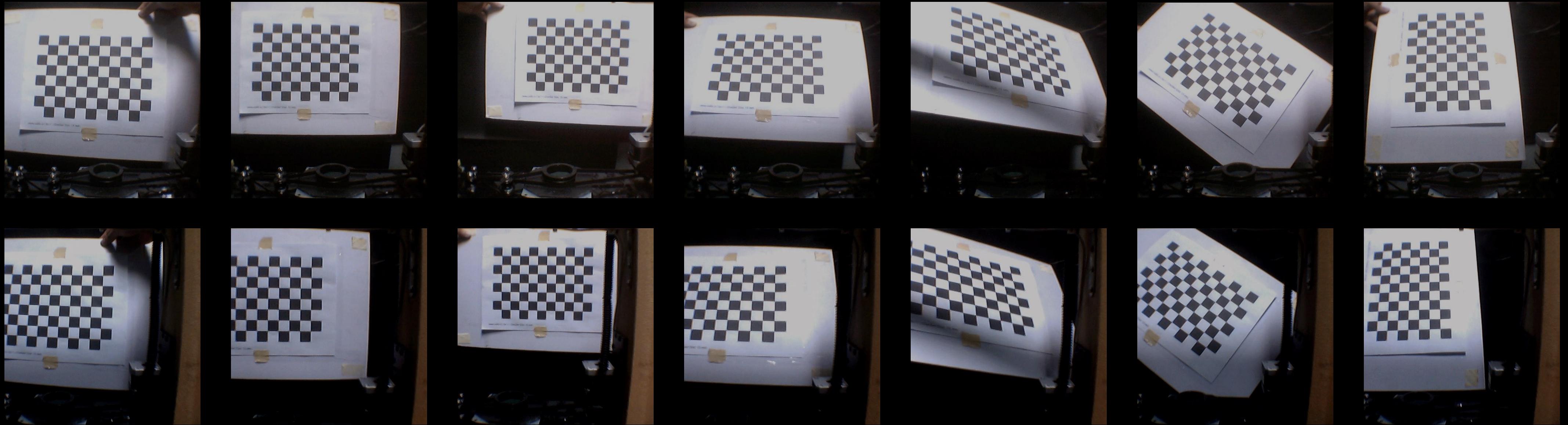
Distortions



Misalignments



# DEPTH MAP GENERATION



Checkerboard images captured by the stereo cameras

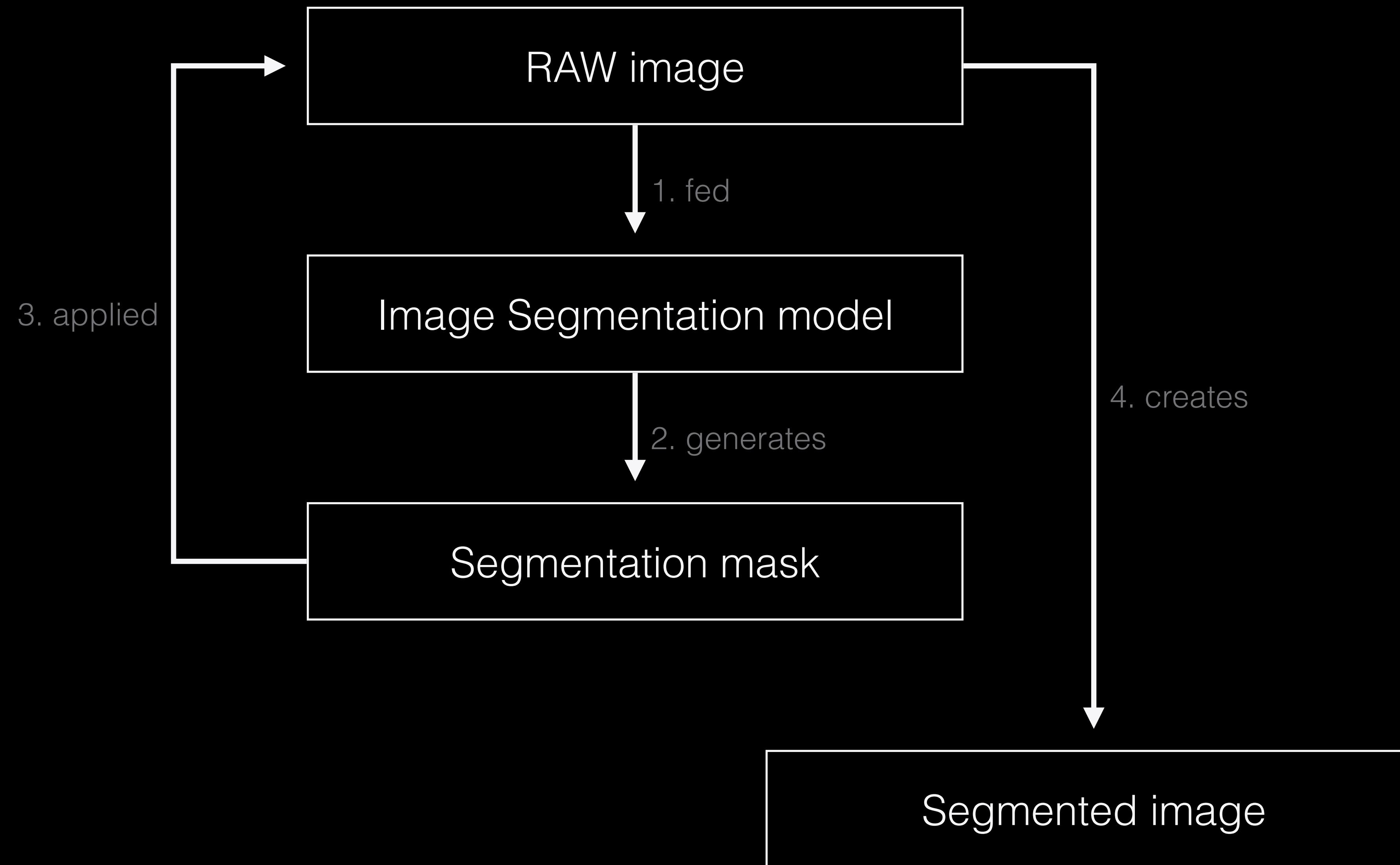


Calibration Map

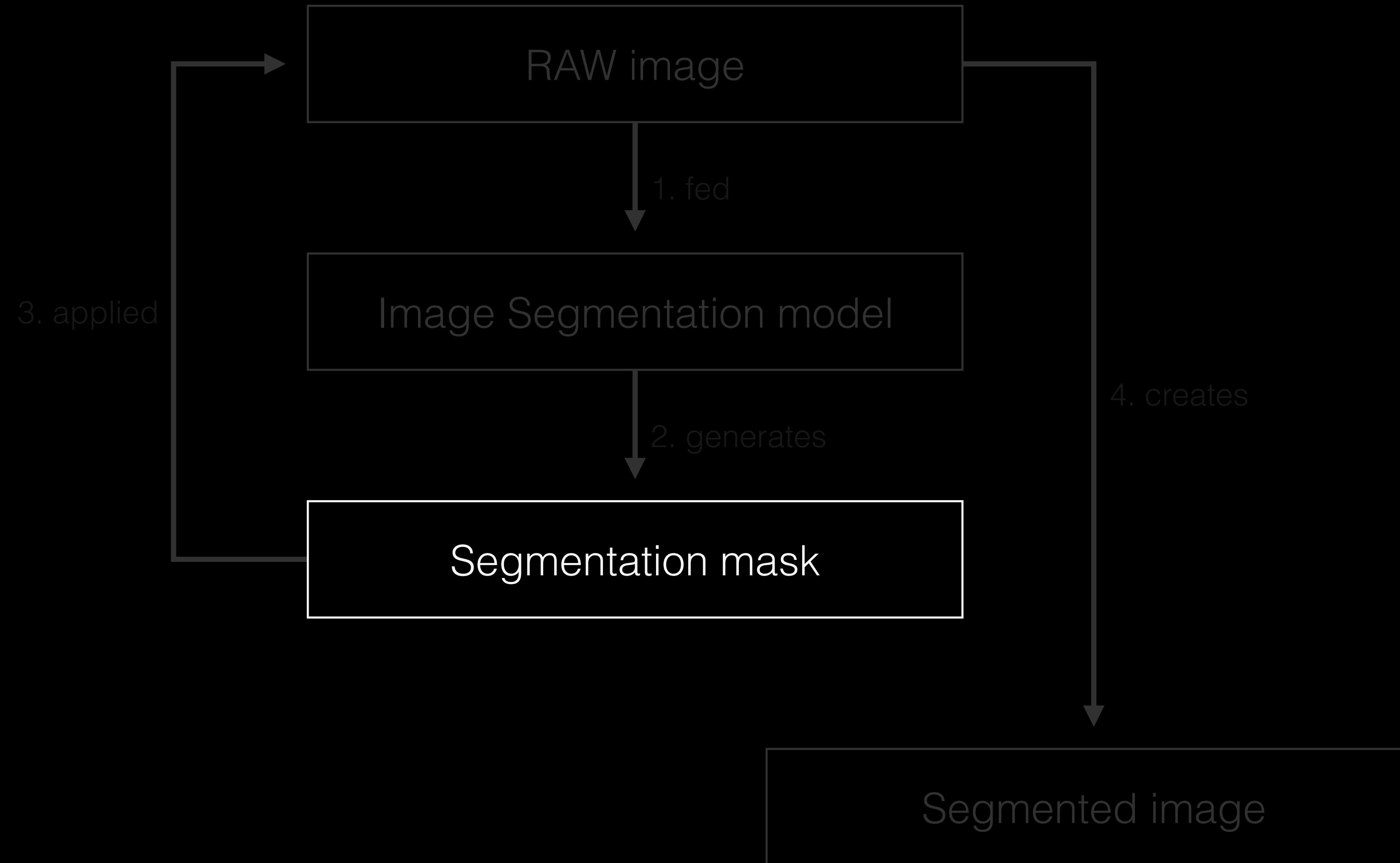


Distorted stereo images

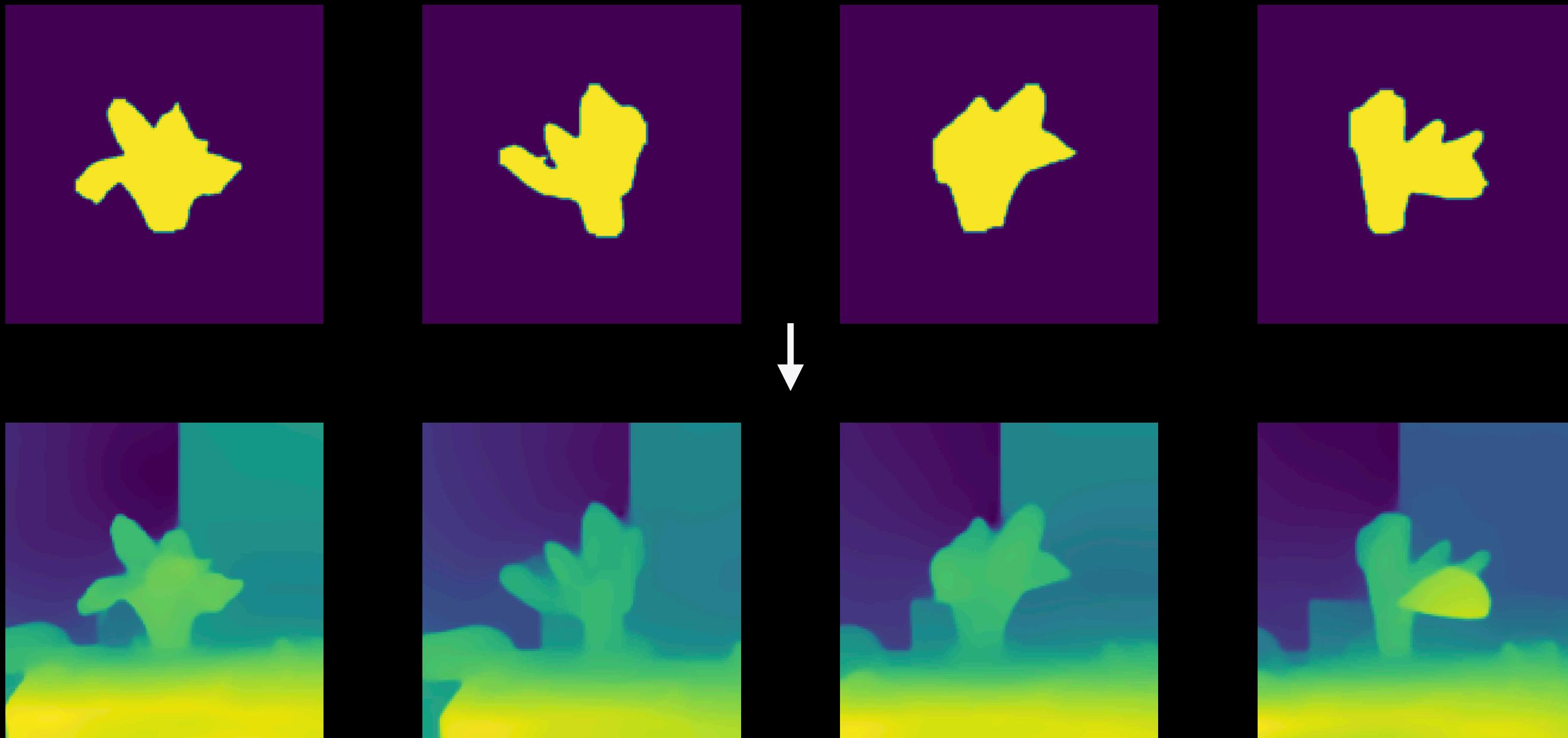
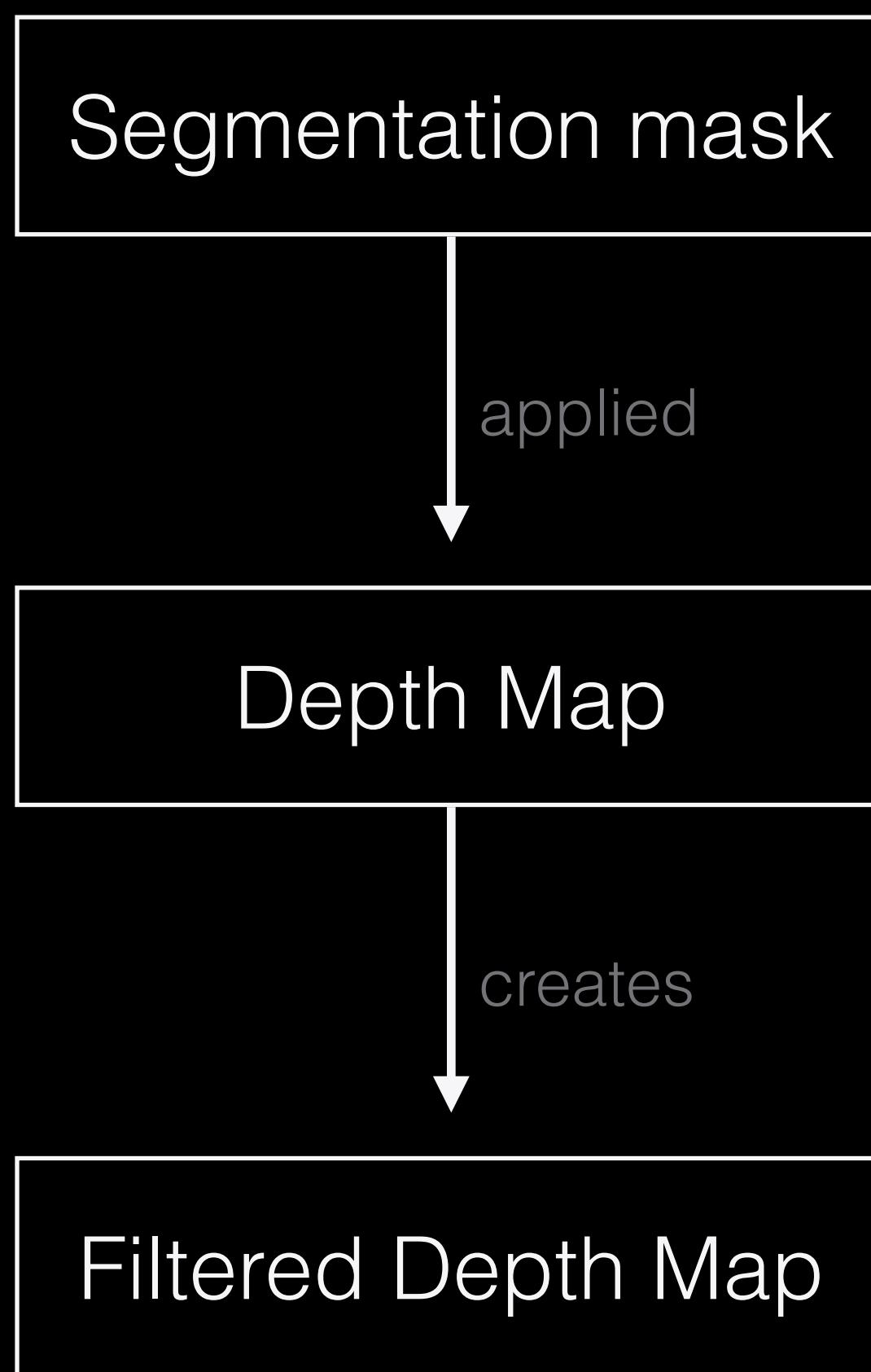
# DEPTH MAP GENERATION



# DEPTH MAP GENERATION

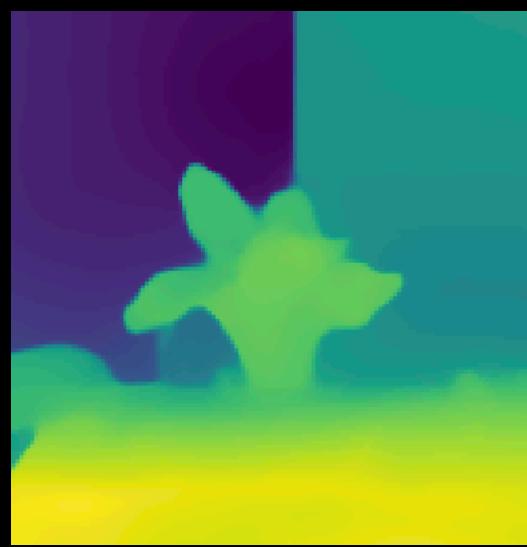


# DEPTH MAP GENERATION

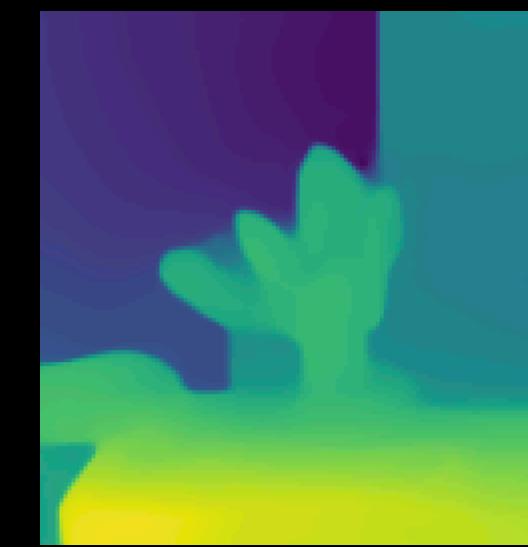


# POINT CLOUD GENERATION

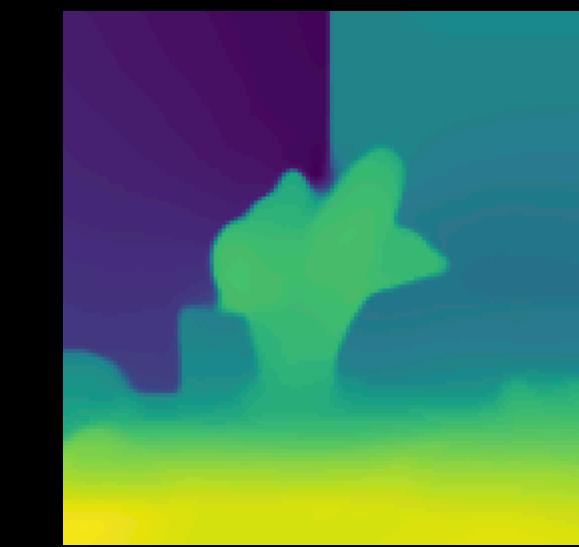
Depth Maps



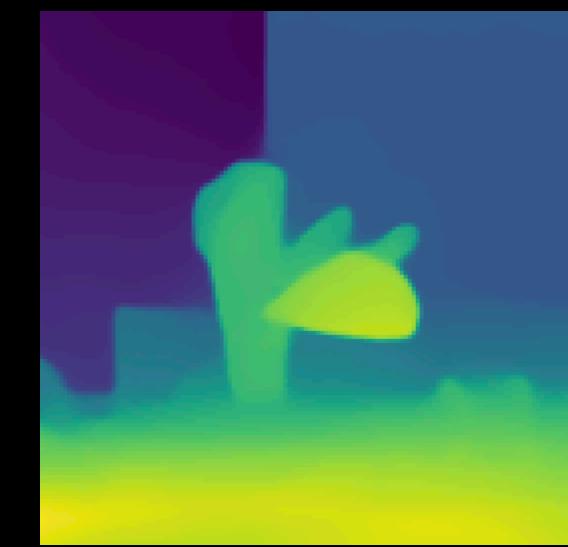
Front side



Left side



Back side



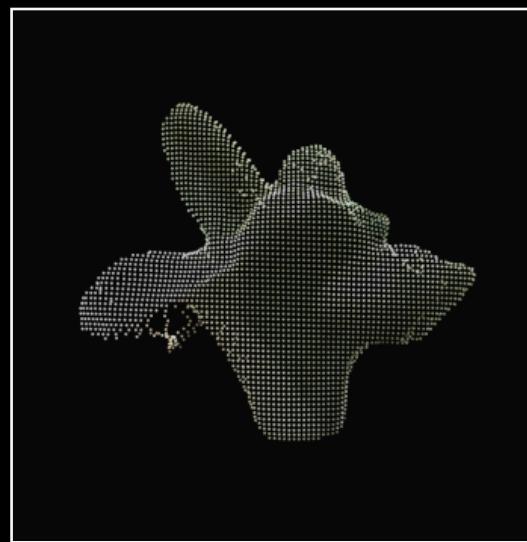
Right side



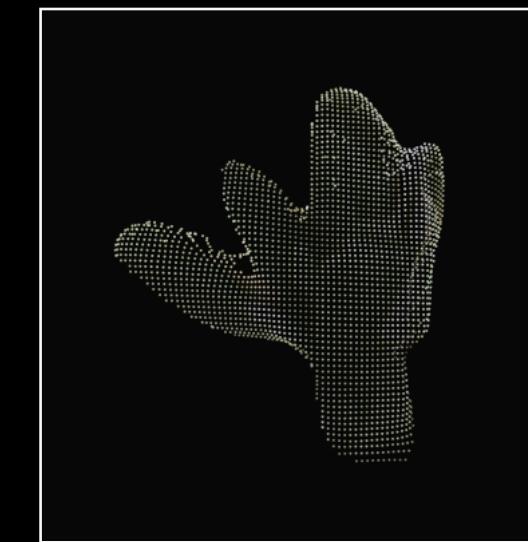
3D Reprojection



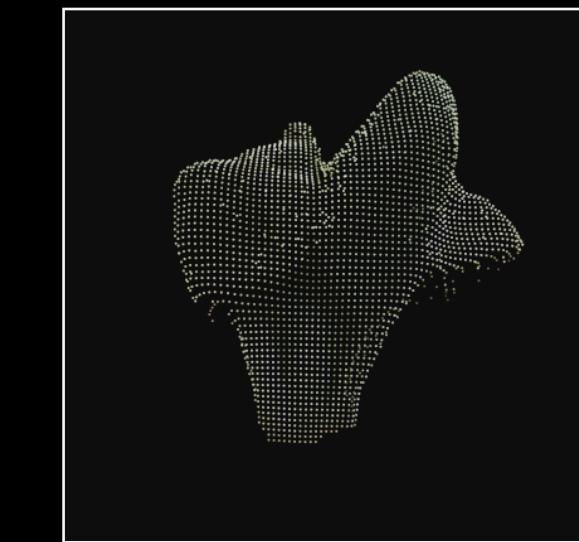
Point Clouds



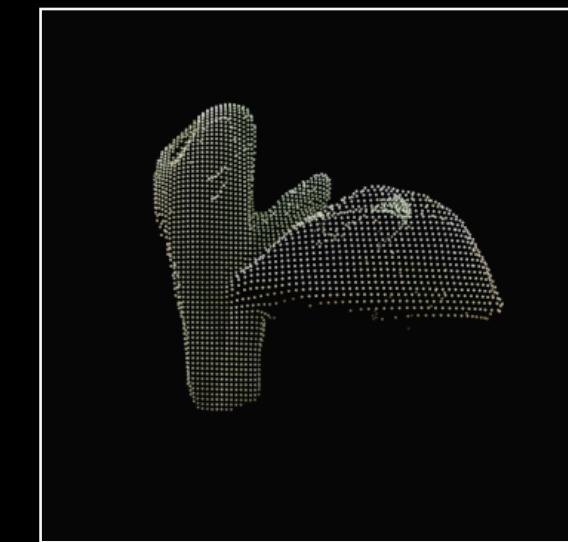
Front side



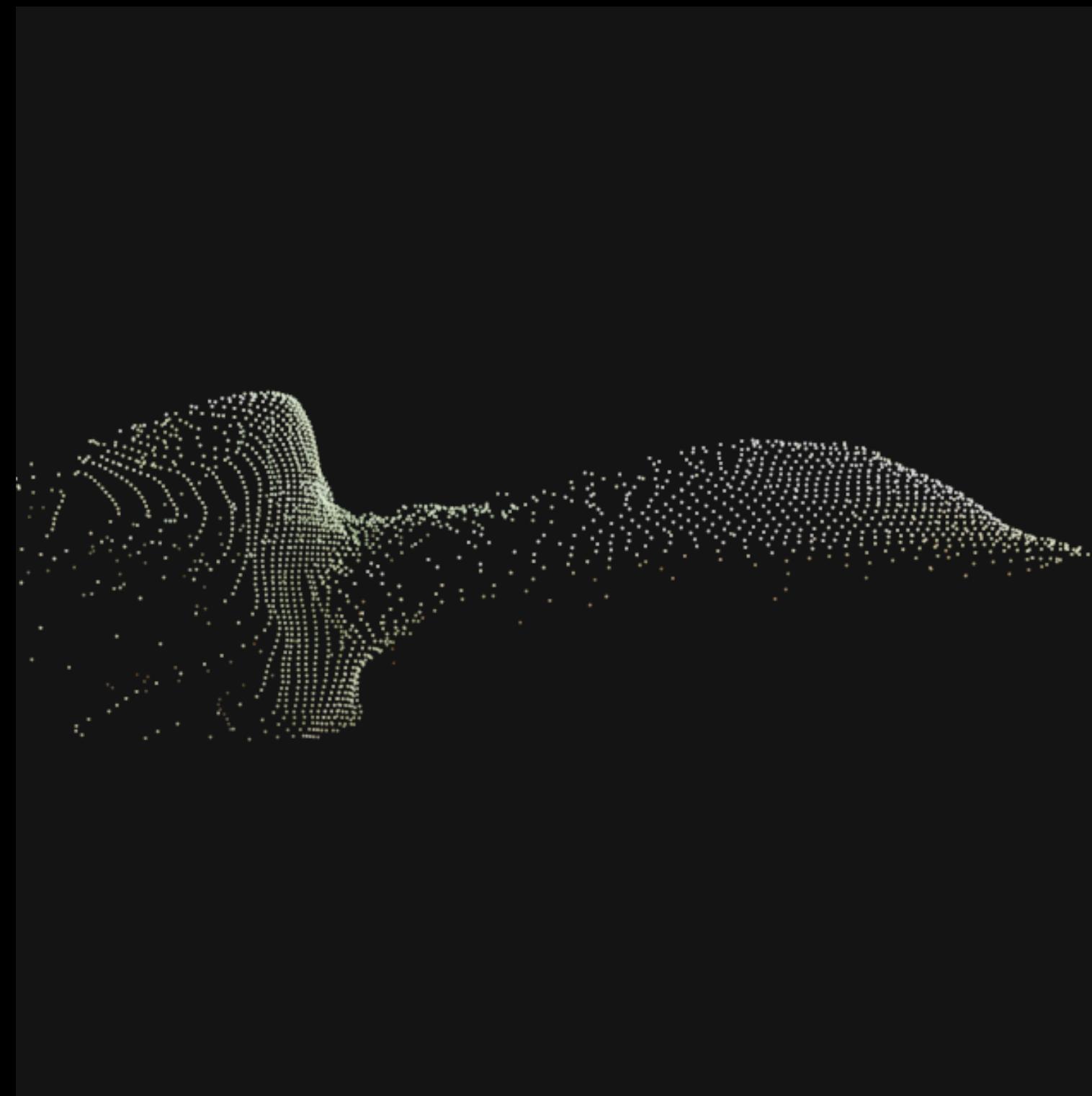
Left side



Back side



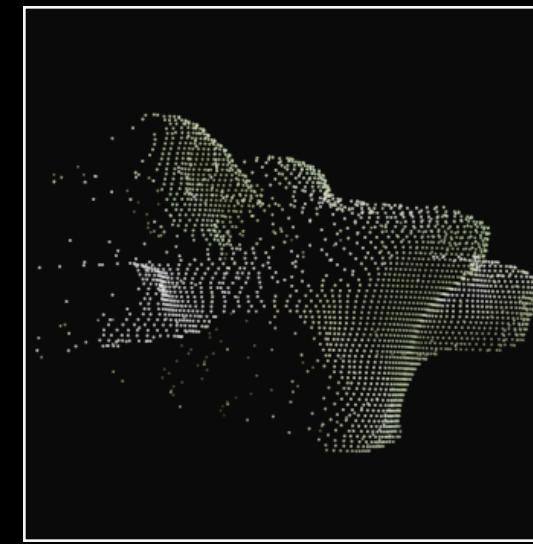
Right side



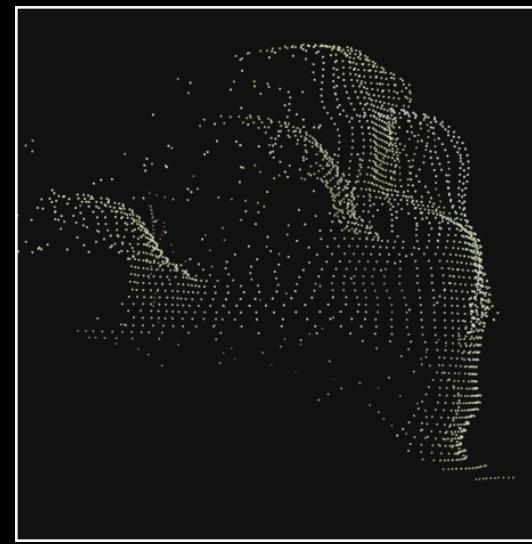
Warping problem

# POINT CLOUD REFINEMENT

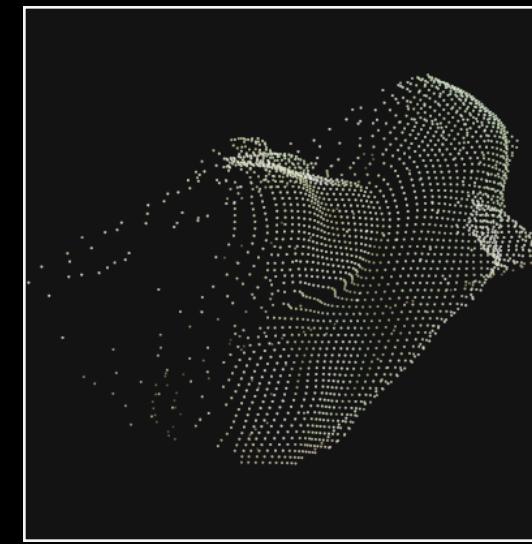
Fixing warping



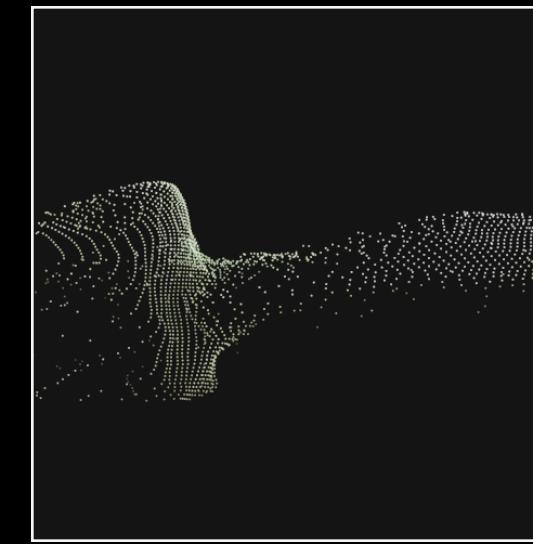
Front side



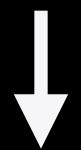
Left side



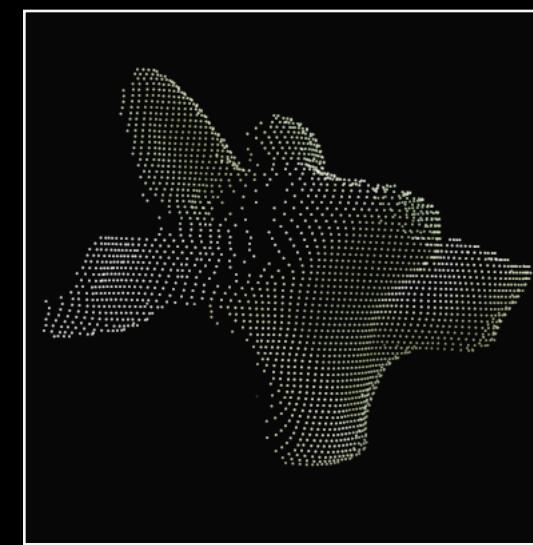
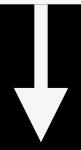
Back side



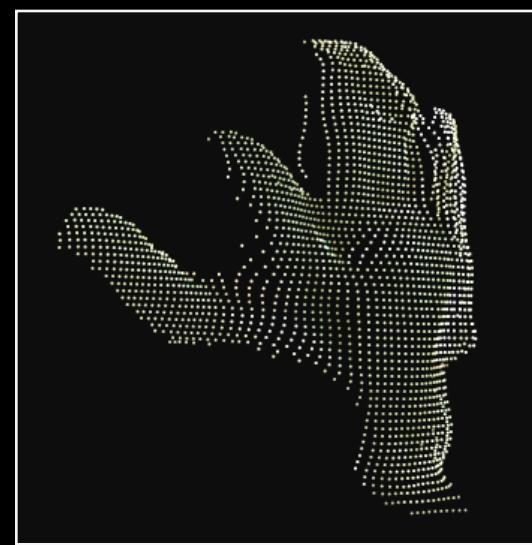
Right side



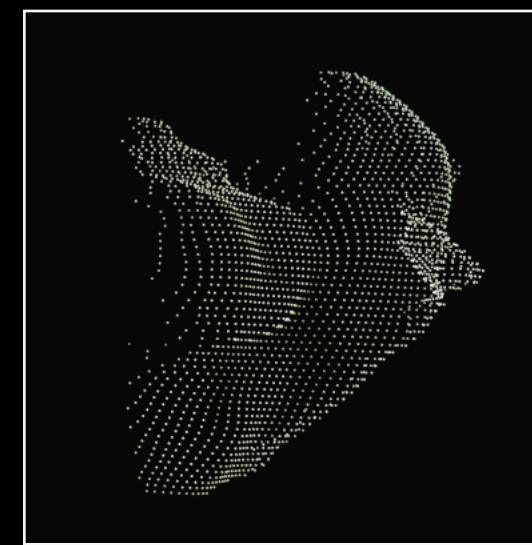
.transform



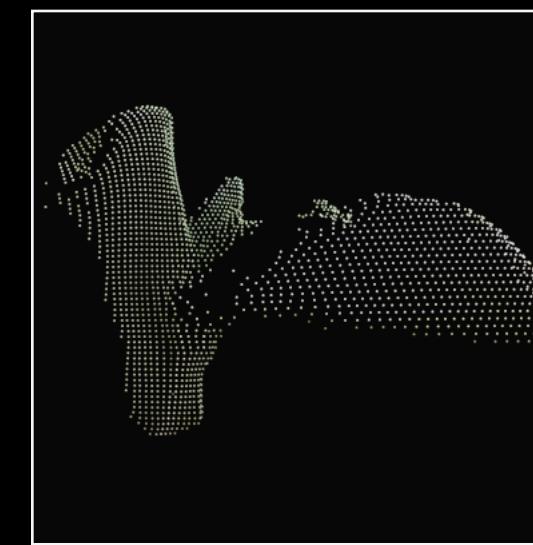
Front side



Left side

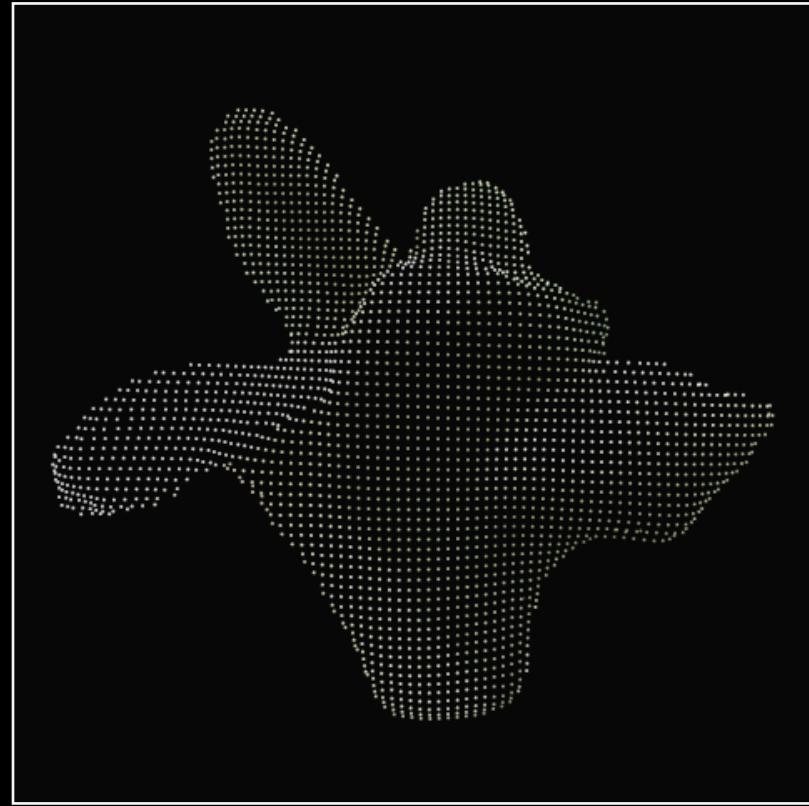


Back side

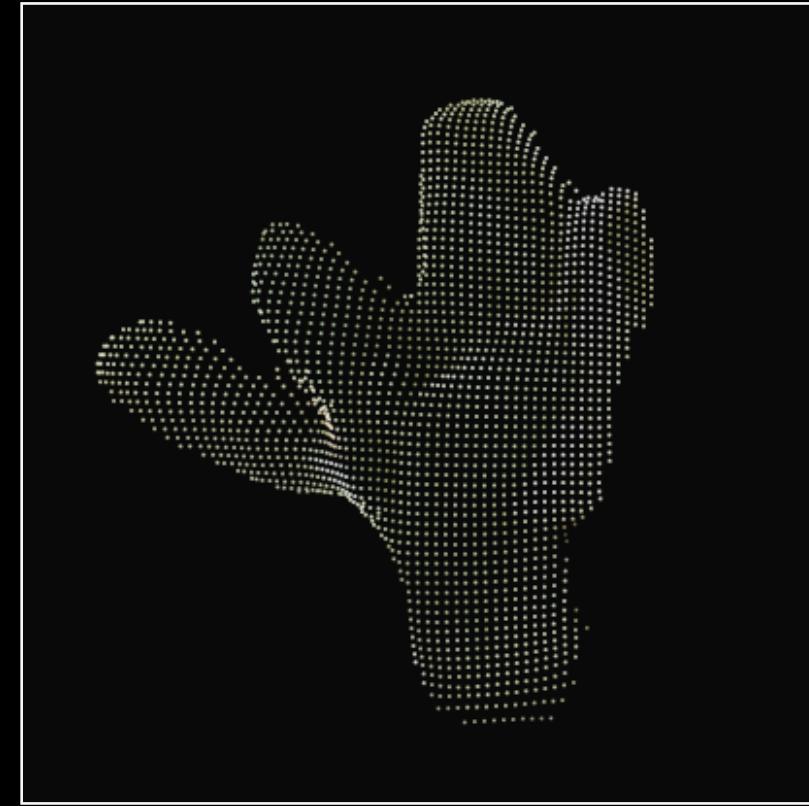


Right side

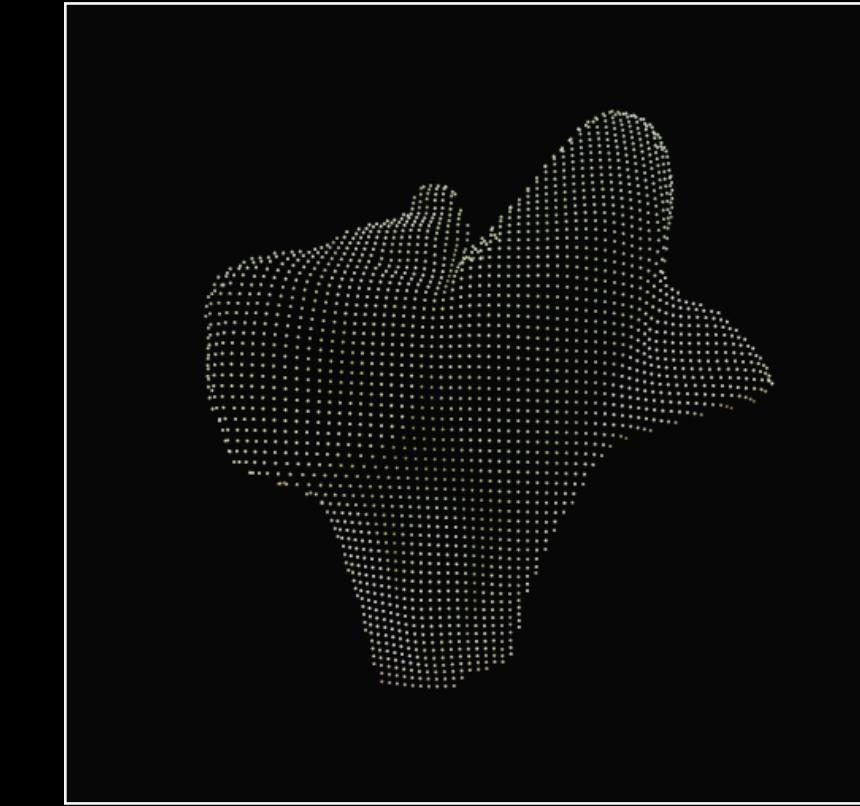
# POINT CLOUD REGISTRATION



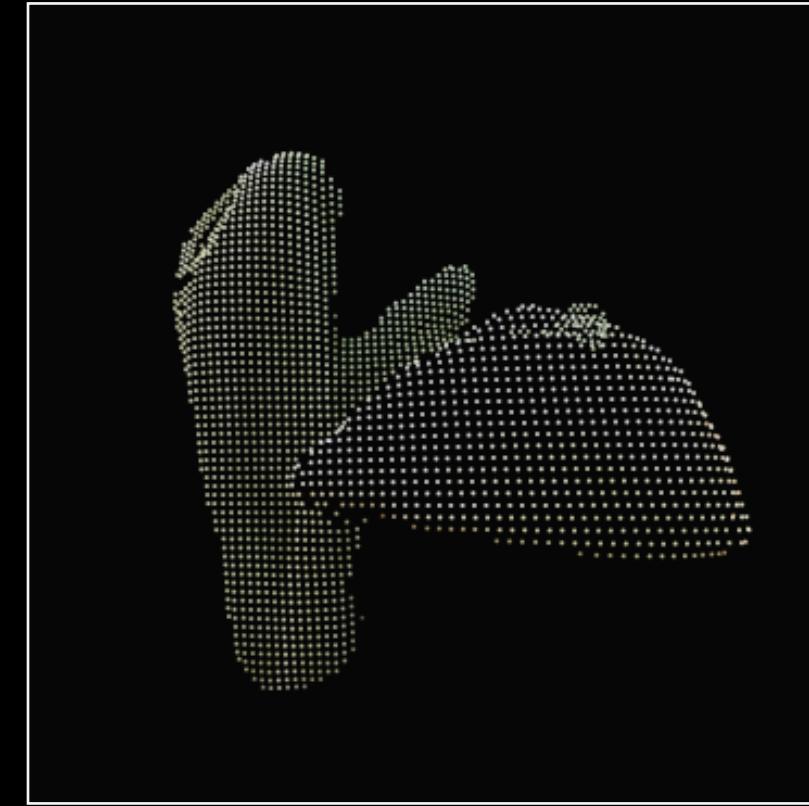
Front side



Left side



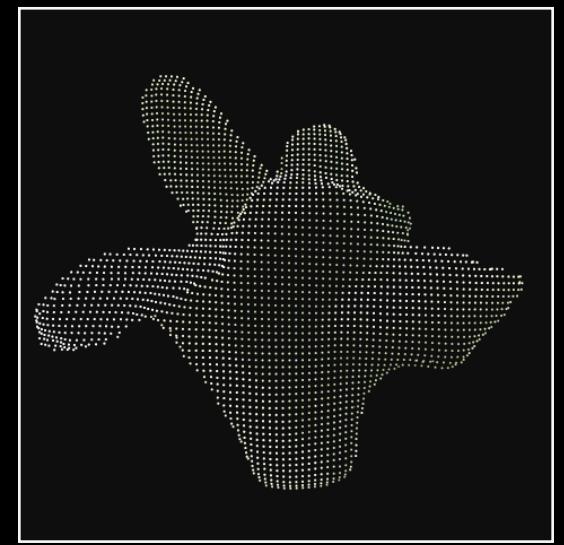
Back side



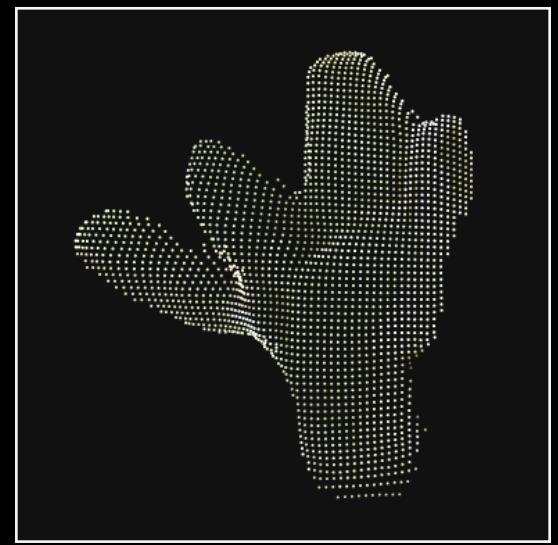
Right side

Disconnected point clouds

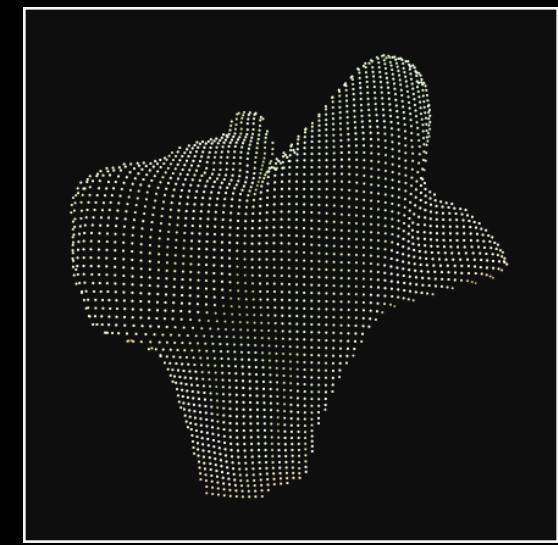
# POINT CLOUD REGISTRATION



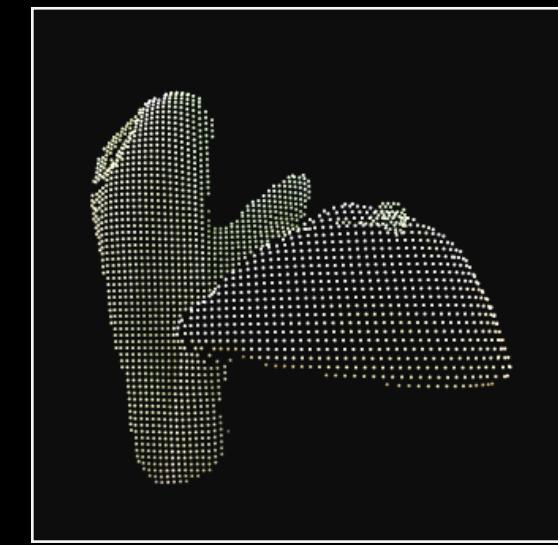
Front side



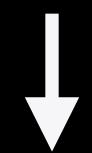
Left side



Back side



Right side

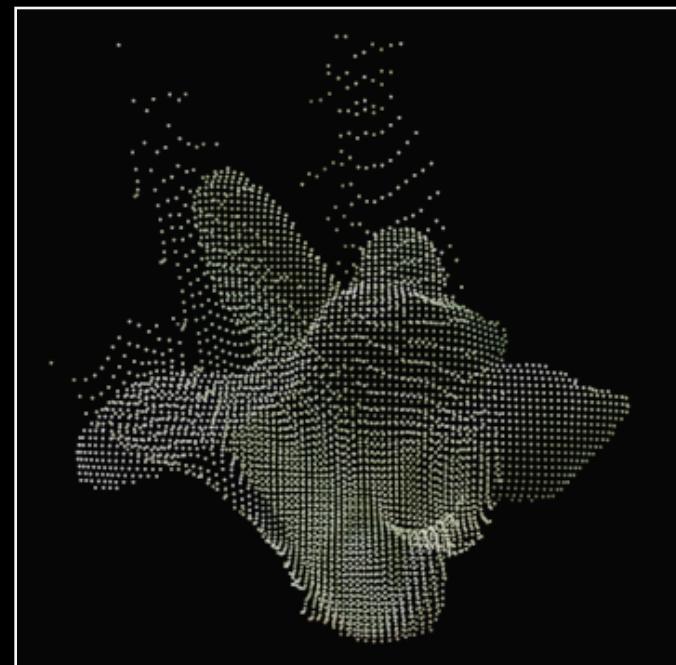


.translate

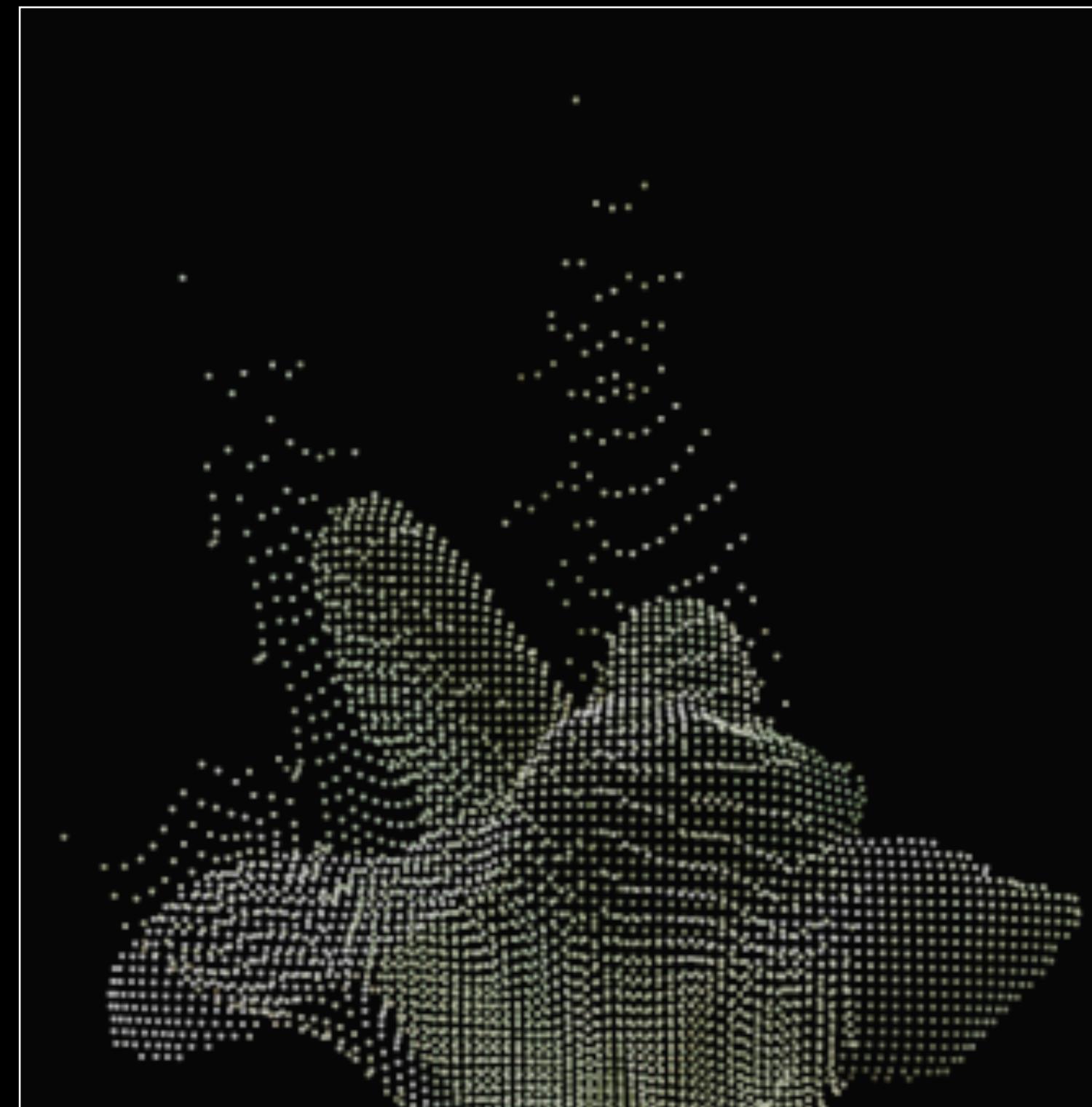
.rotate

.scale

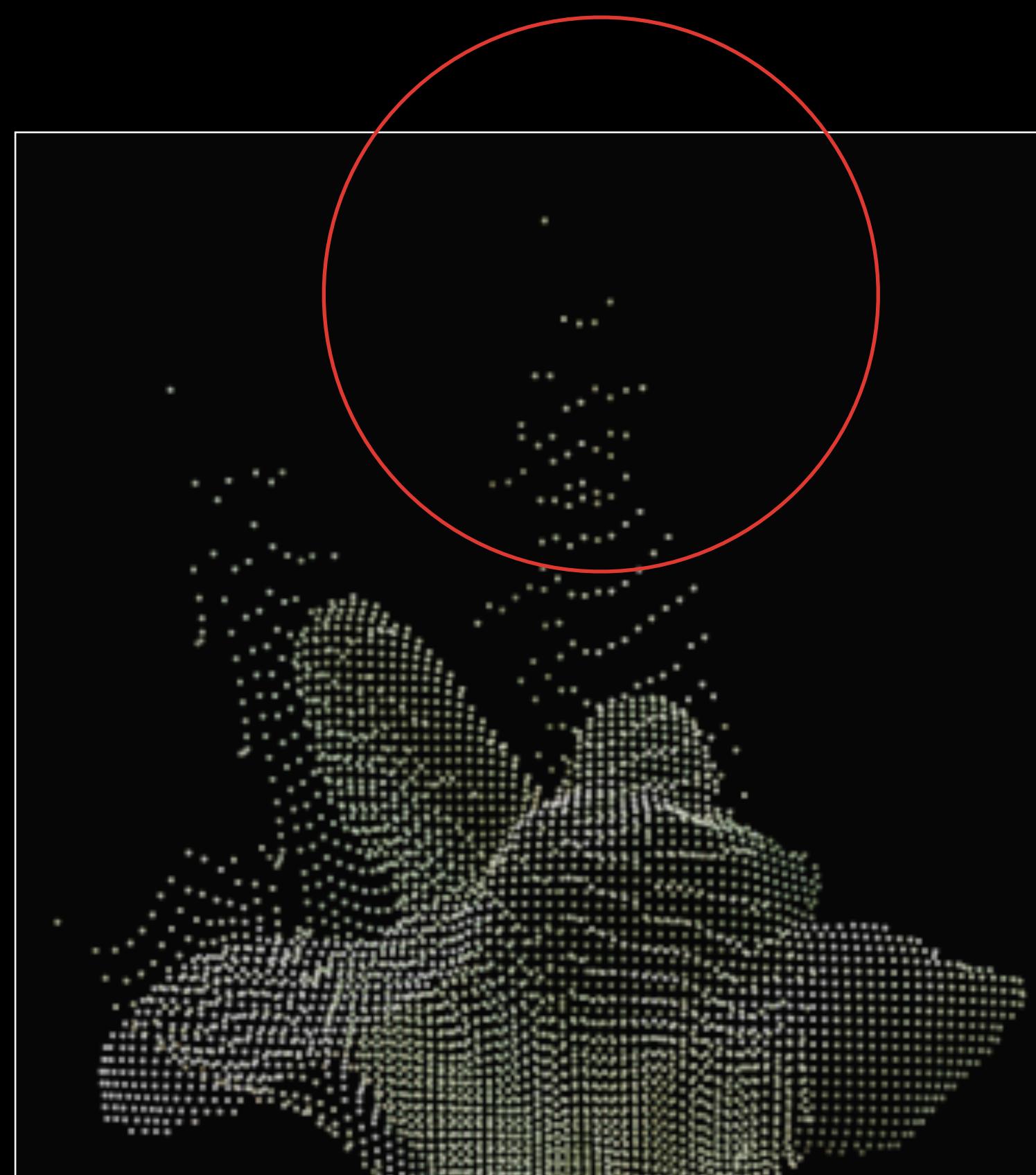
.transform



Unified point cloud



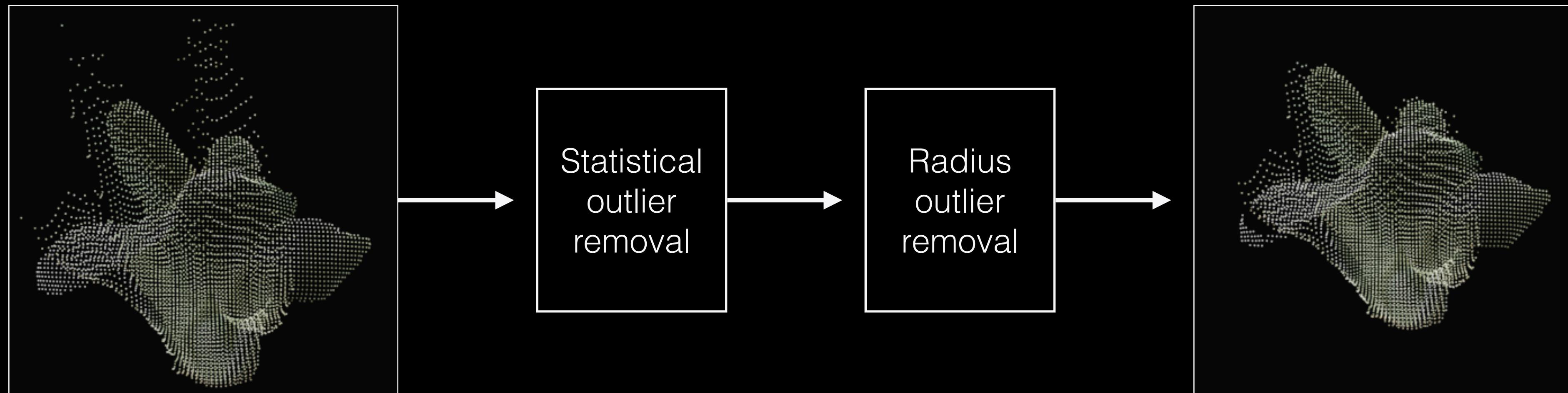
Outlier problem

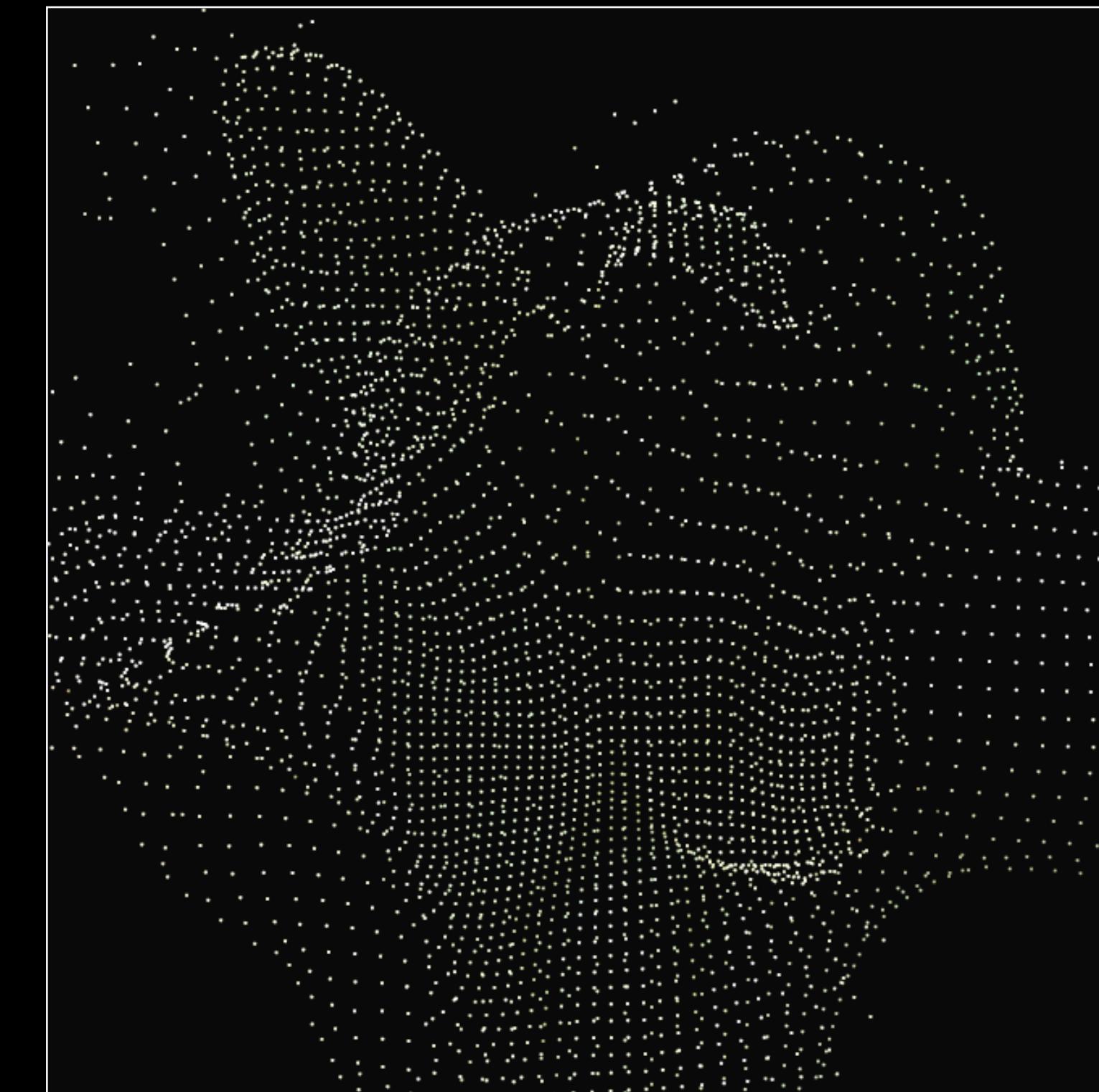
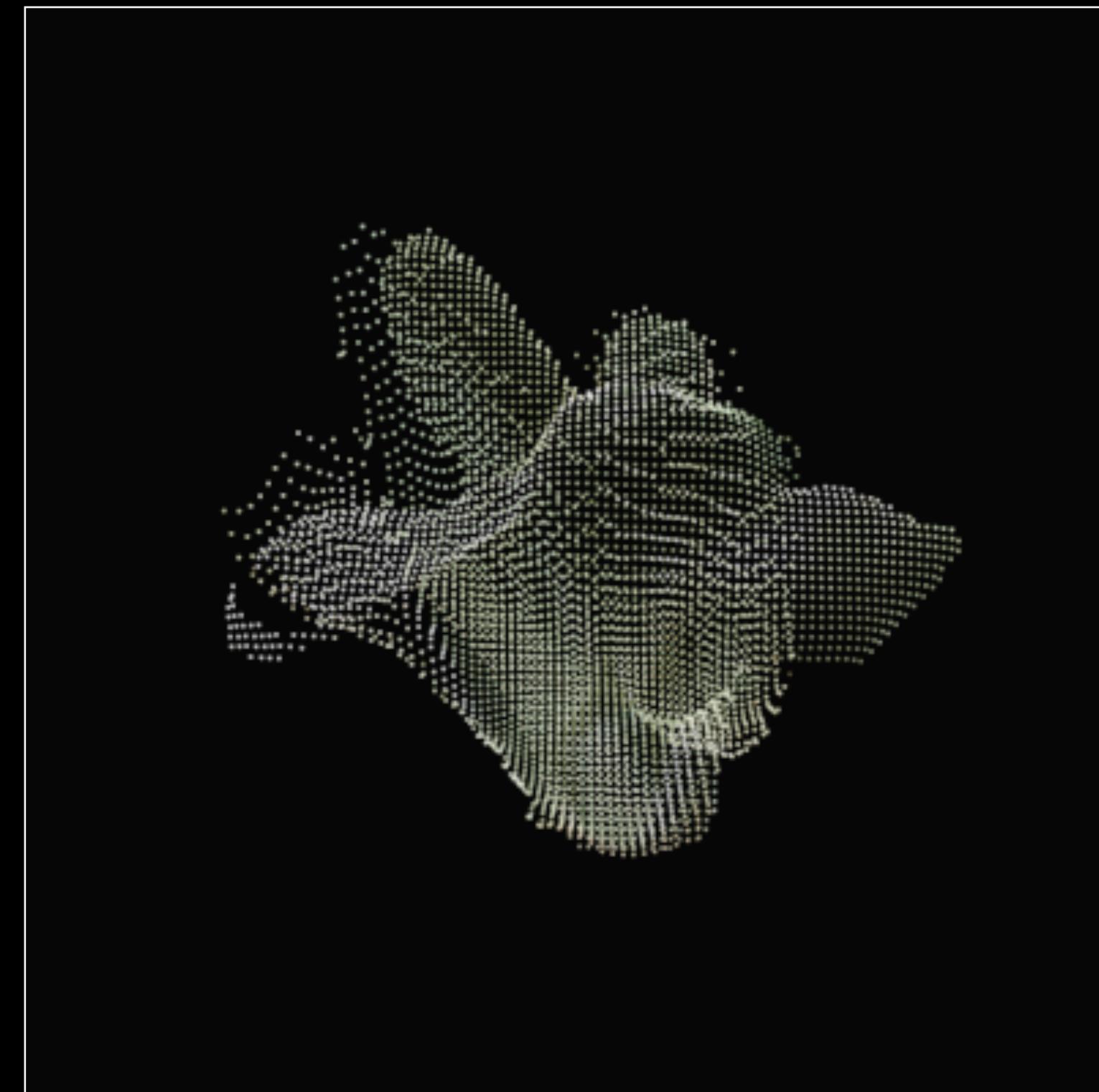


Outlier problem

# POINT CLOUD REFINEMENT

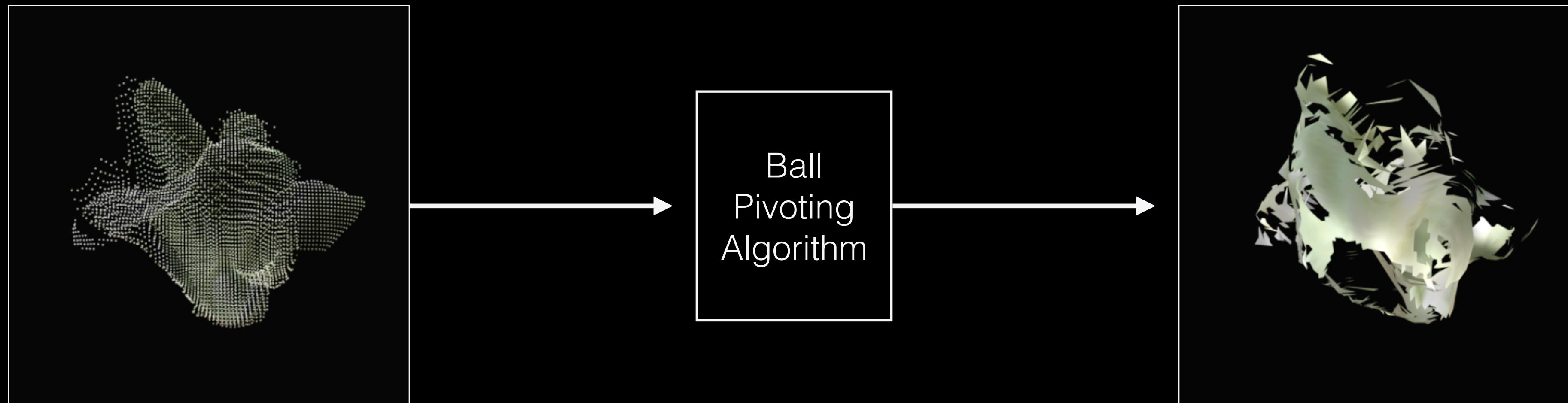
Fixing excess point clouds



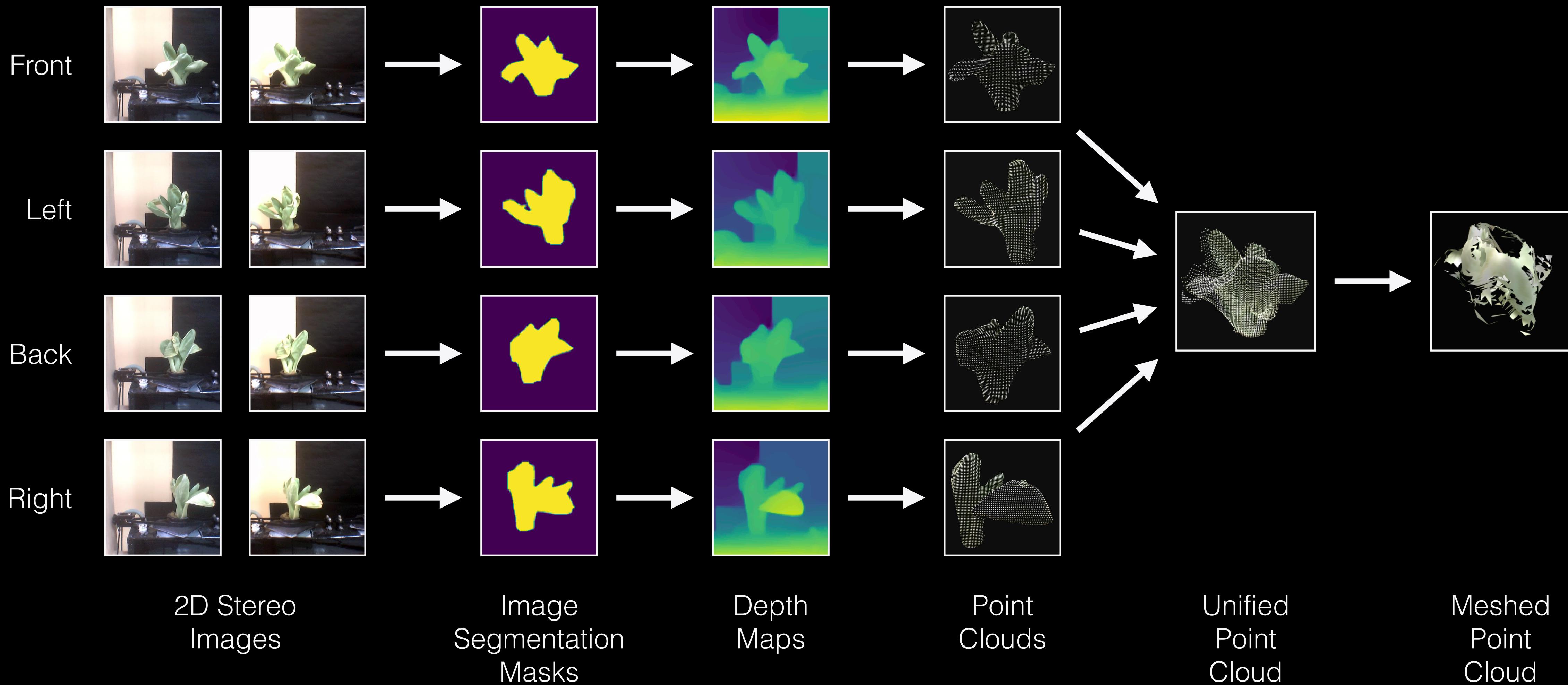


Discontinuous surface

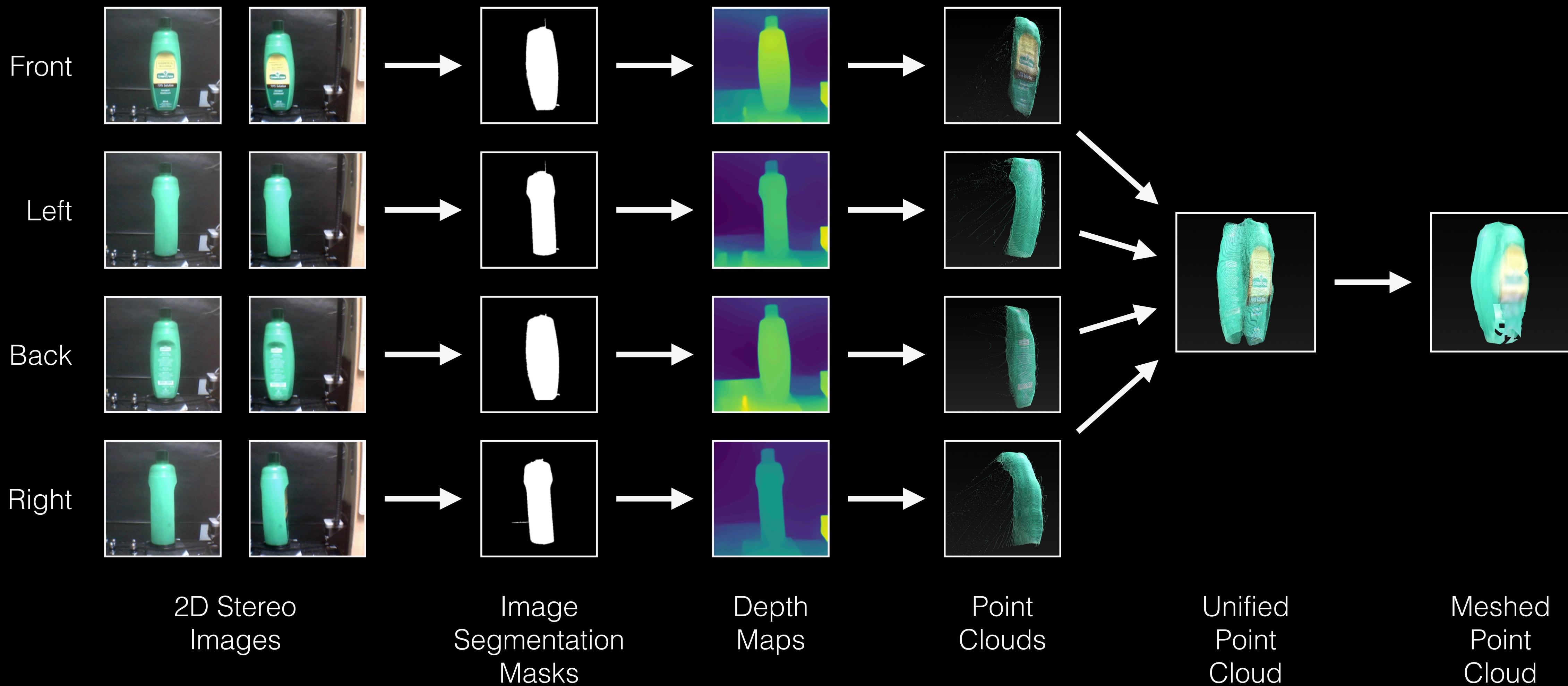
# MESH GENERATION



# Summary



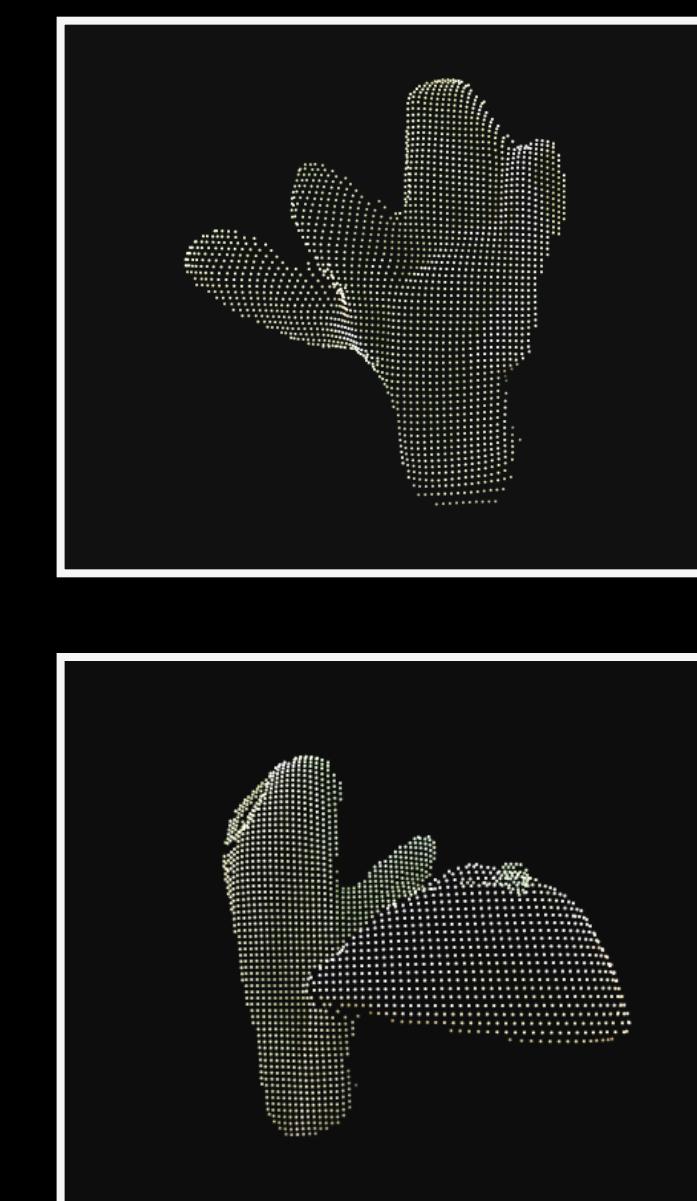
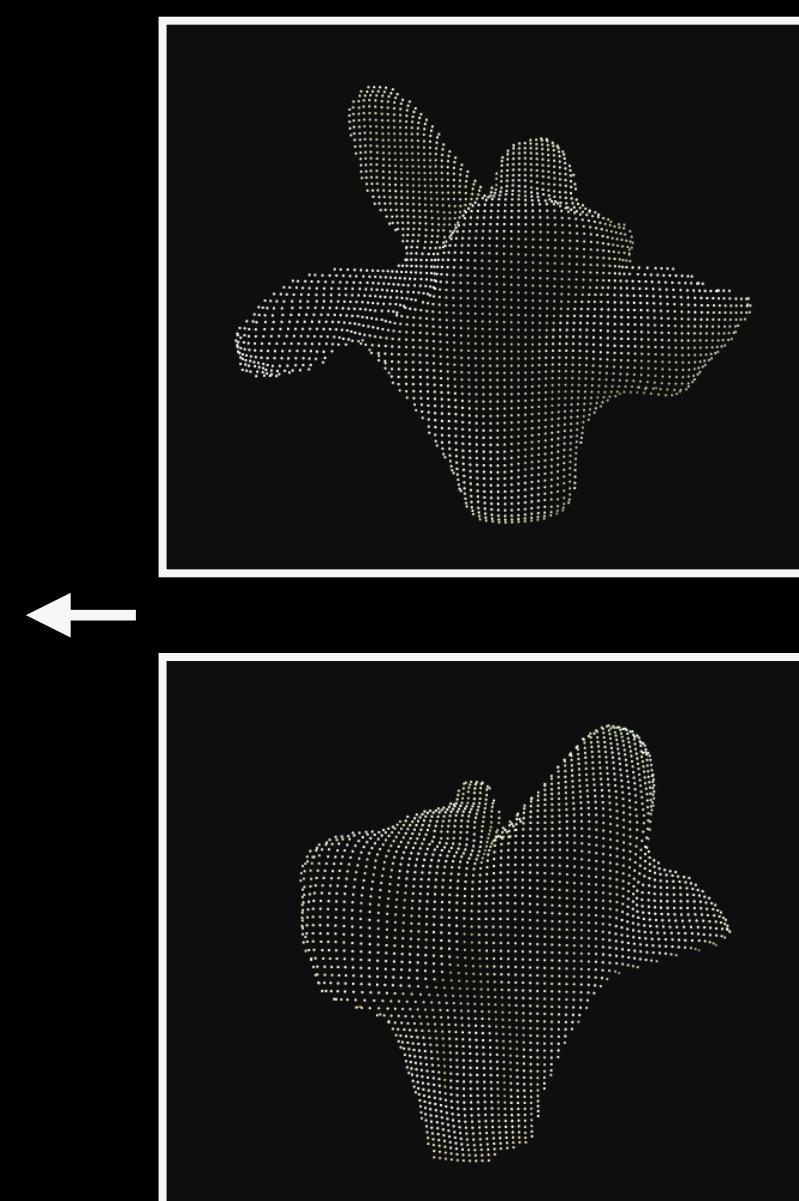
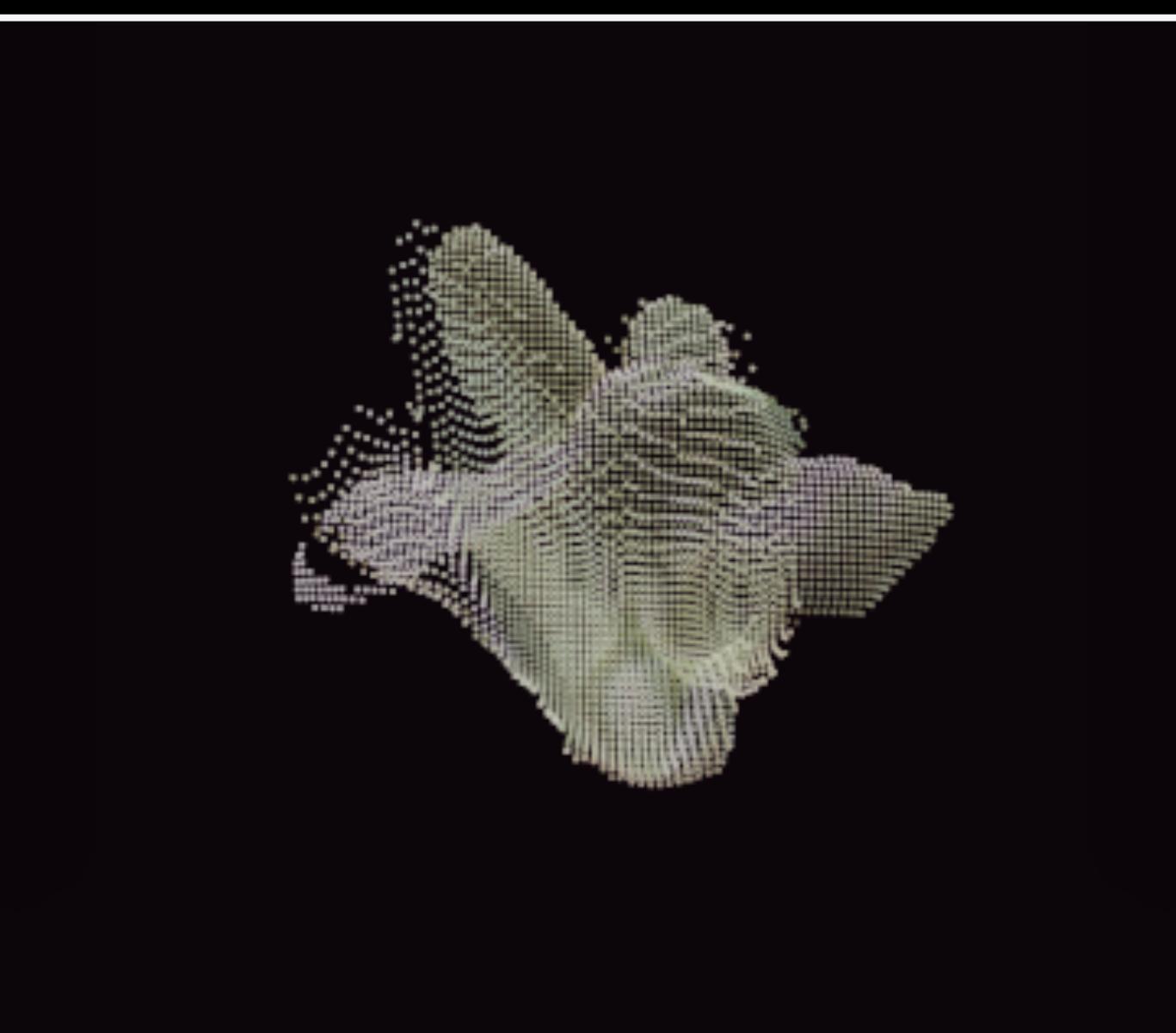
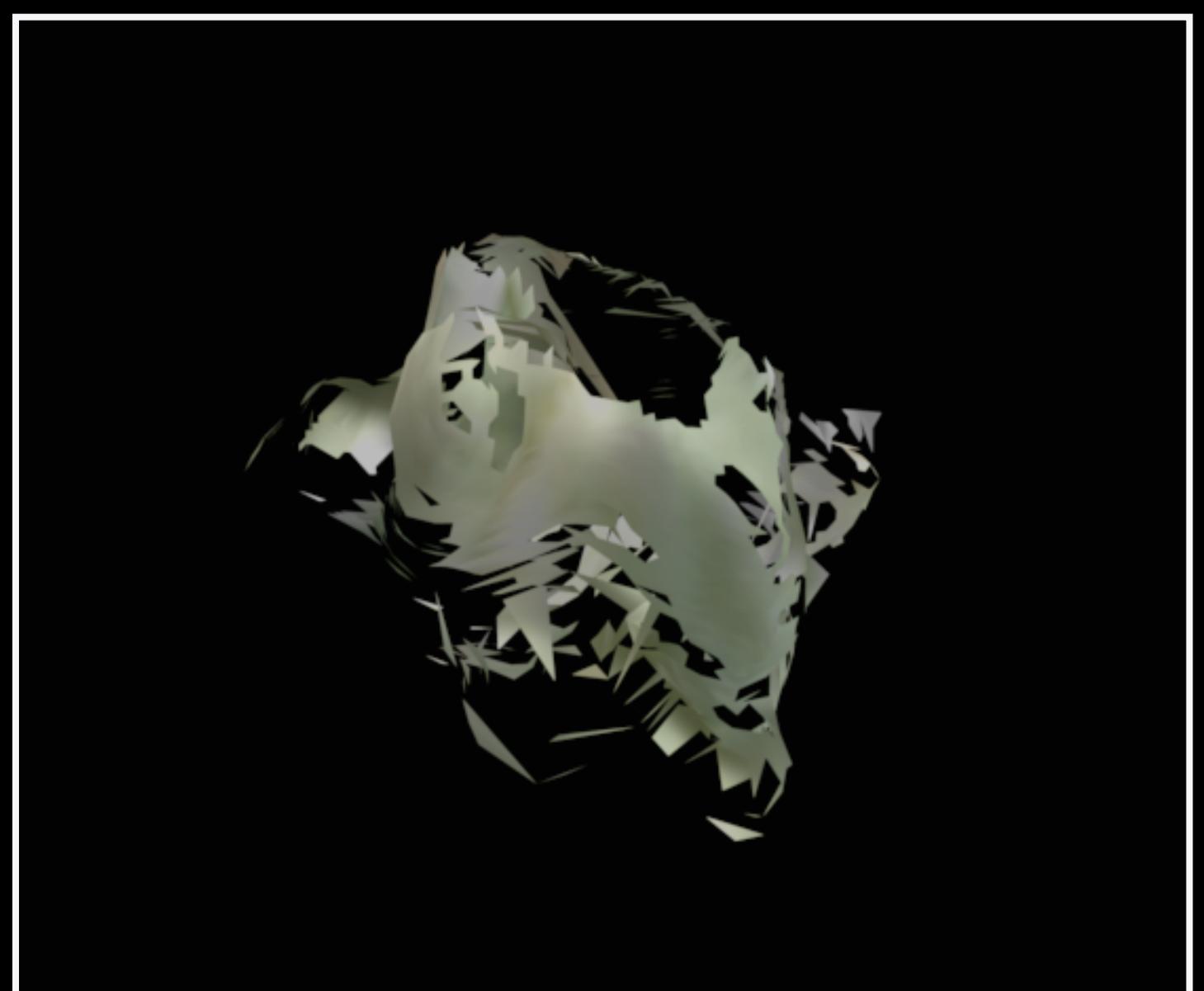
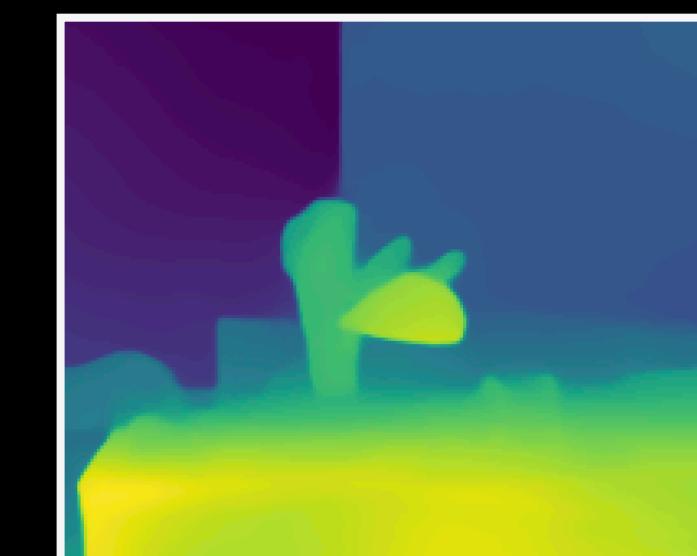
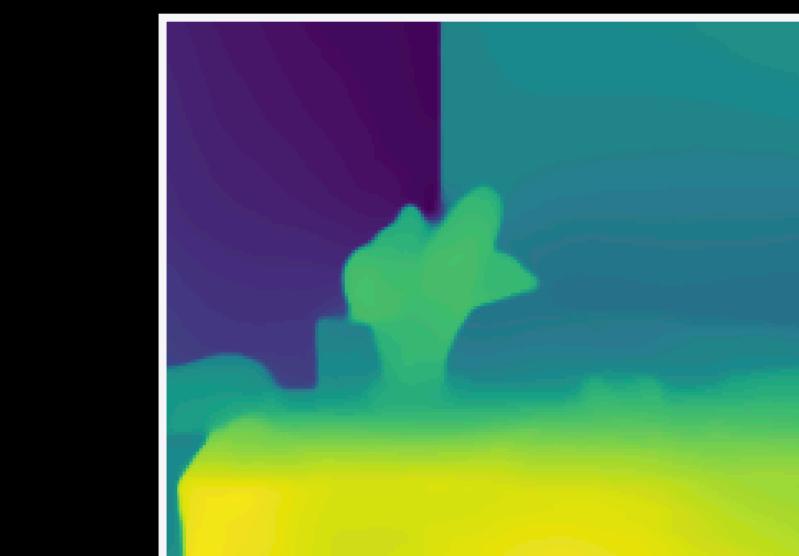
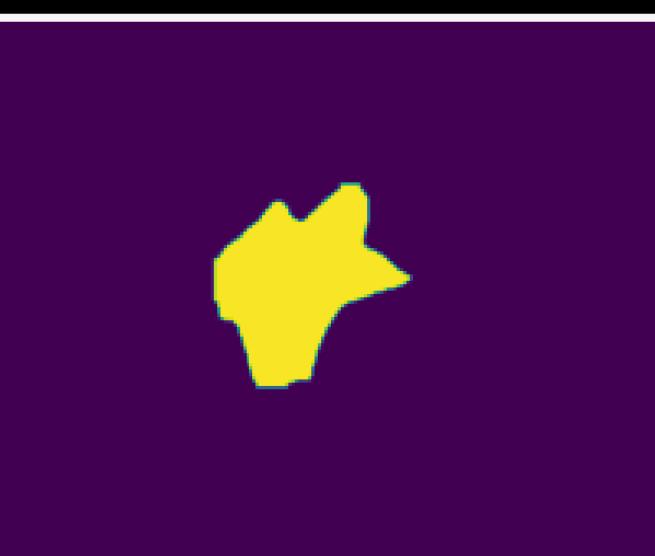
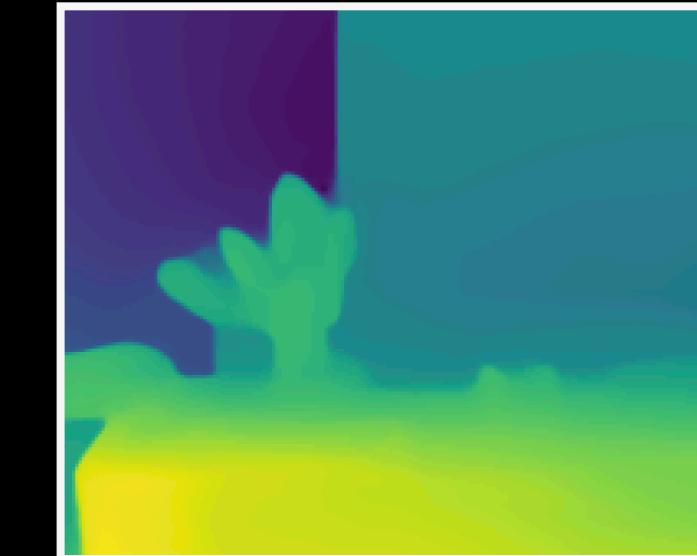
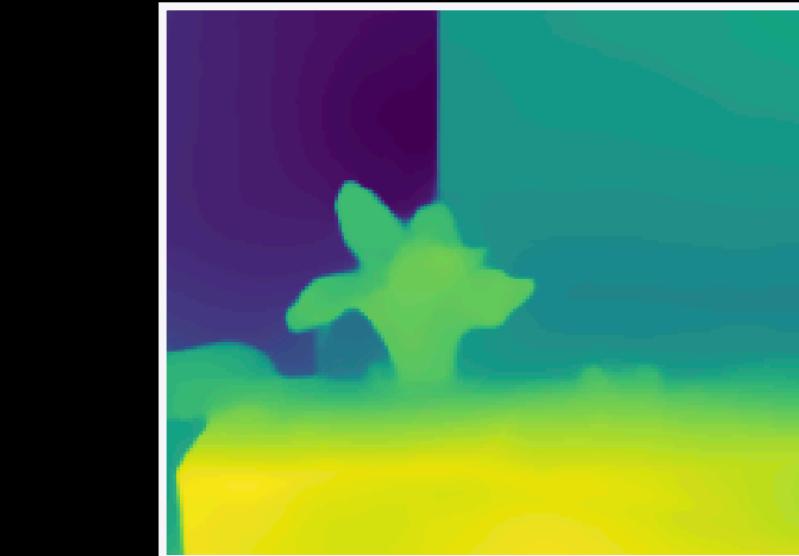
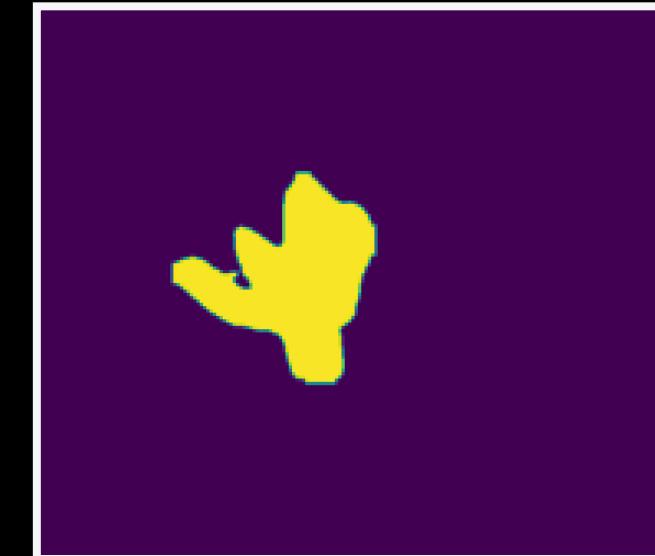
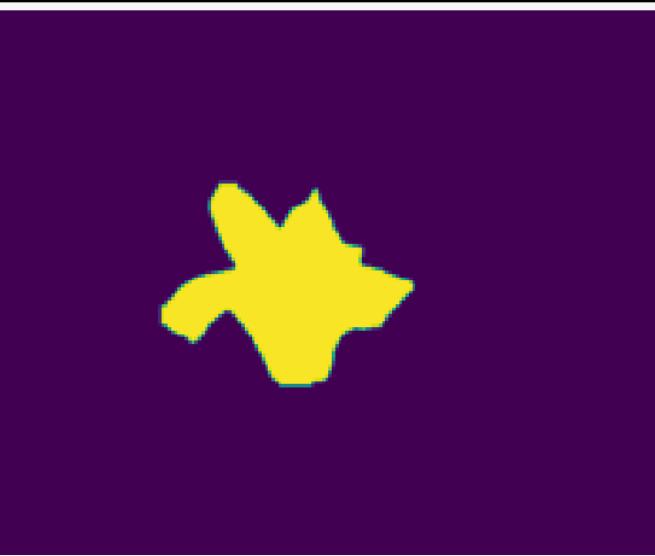
# Testing other objects



# LIMITATIONS

Evaluation in the 3D reconstruction process is limited to visual inspection, with no quantitative assessment.

The 3D reconstruction process focuses solely on the front, left, back, and right sides of the object, **excluding the top and bottom views**.



# CONCLUSIONS

# CONCLUSIONS

Comparing YOLOv8 and Detectron2 for Plant Class Segmentation

In plant class segmentation,

YOLOv8 trains faster with lower Train Times,

Detectron2 provides better segmentation with higher Box and Mask Average Precisions, and lower Box and Mask Losses,

Detectron2 is more RAM-efficient with lower System RAM and GPU RAM consumptions.

# CONCLUSIONS

## Comparing YOLOv8 and Detectron2 for Plant Class Segmentation

		Hyperparameters	
		Decreasing	Increasing
YOLOv8	Segmentation Accuracy	Worsens (in small decrements)	Improves (in small increments)
	Training Speed	Improves	Worsens
	Memory Consumption	Improves (with respect to GPU RAM)	Worsens (with respect to GPU RAM)
Detectron2	Segmentation Accuracy	Worsens (in large decrements)	Improves (in large increments)
	Training Speed	Worsens	Improves
	Memory Consumption	Improves (with respect to GPU RAM)	Worsens (with respect to GPU RAM)

# CONCLUSIONS

Comparing YOLOv8 and Detectron2 for Plant Class Segmentation

YOLOv8 is ideal

When you want faster training times and can accept slightly lower accuracy, with GPU RAM usage not being a concern.

Detectron2 is ideal

When you need better segmentation accuracy or have lower GPU RAM requirements, even if it means longer training times.

# CONCLUSIONS

Comparing YOLOv8 and Detectron2 for Plant Class Segmentation

For YOLOv8,

Use **lower hyperparameters** if you want faster training and lower GPU RAM usage, with a slight trade-off in segmentation accuracy.

Use **higher hyperparameters** if you aim to improve segmentation accuracy and are okay with slower training and higher GPU RAM usage.

# CONCLUSIONS

Comparing YOLOv8 and Detectron2 for Plant Class Segmentation

For Detectron2,

Use **lower hyperparameters** if you have limited GPU RAM and can accept slight compromises in segmentation accuracy and training speed.

Use **higher hyperparameters** if you want to maximize segmentation accuracy and training speed, and GPU RAM usage is not a concern.

# CONCLUSIONS

## 3D Reconstruction process from 2D Stereoscopic Plant Images

Combining existing computer vision technologies to reconstruct a 3D model from 2D stereoscopic plant images is theoretically possible. However, the visual inspection of the result does not meet expectations, as the plant's (or object's) structure or form is hardly distinguishable.



# CONCLUSIONS

## 3D Reconstruction process from 2D Stereoscopic Plant Images

Firstly, the point cloud density is very low, resulting in a blurry mesh with points that are spaced too far apart, preventing a smooth, continuous surface from forming.

Secondly, the process of merging the point clouds through manual transformations, while functional, is inefficient. It is time-consuming, and achieving good merging results often requires countless trials and errors.

# CONCLUSIONS

## 3D Reconstruction process from 2D Stereoscopic Plant Images

To improve point cloud quality, we recommend using high-resolution stereo cameras to produce higher-resolution point clouds, leading to more accurate disparity and depth mapping.

Another suggestion is to explore alternative techniques for capturing images with stereo cameras that may result in denser point clouds.

Lastly, we recommend implementing an automatic point cloud merging technique to replace the current manual process.

# Investigating 3D Model Reconstruction from 2D Stereoscopic Plant Images with Investigation between YOLOv8 and Detectron2 AI Architectures for Plant Class Segmentation

John Ivan Diaz, Craig Joseph Goc-ong, Kaye Louise Manilong  
Alvin Joseph Macapagal, Philip Virgil Astillo\*