Development of Carbs Estimation Model using ML/DL

Date: 09-07-2024

Anuj Attri
Under the Supervision of: Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Work Progress

Phase	Task	Status
I	Image Data Collection	In Progress
	Data Preprocessing	In Progress
	Data Annotation and Labelling	In Progress
	Classification/Segmentation Model Selection	Completed
	Classification Model Training	Awaiting action
	Image Segmentation	Awaiting action
II	Volume Estimation using Images	In Exploratory stage
	Carbohydrates Data correspond to images	In Progress
	Estimation of Carbohydrates	Not Started
III	Android Development	In Progress
	Android Application Integration	In Exploratory Stage
	Cloud Deployment	Not Started

Data Collection

Image Data Collection

- Source 1: Indian Food Images Dataset (kaggle.com)
- Source 2: food20 dataset(Indian food) (kaggle.com)
- Source 3: From Web and Restaurants

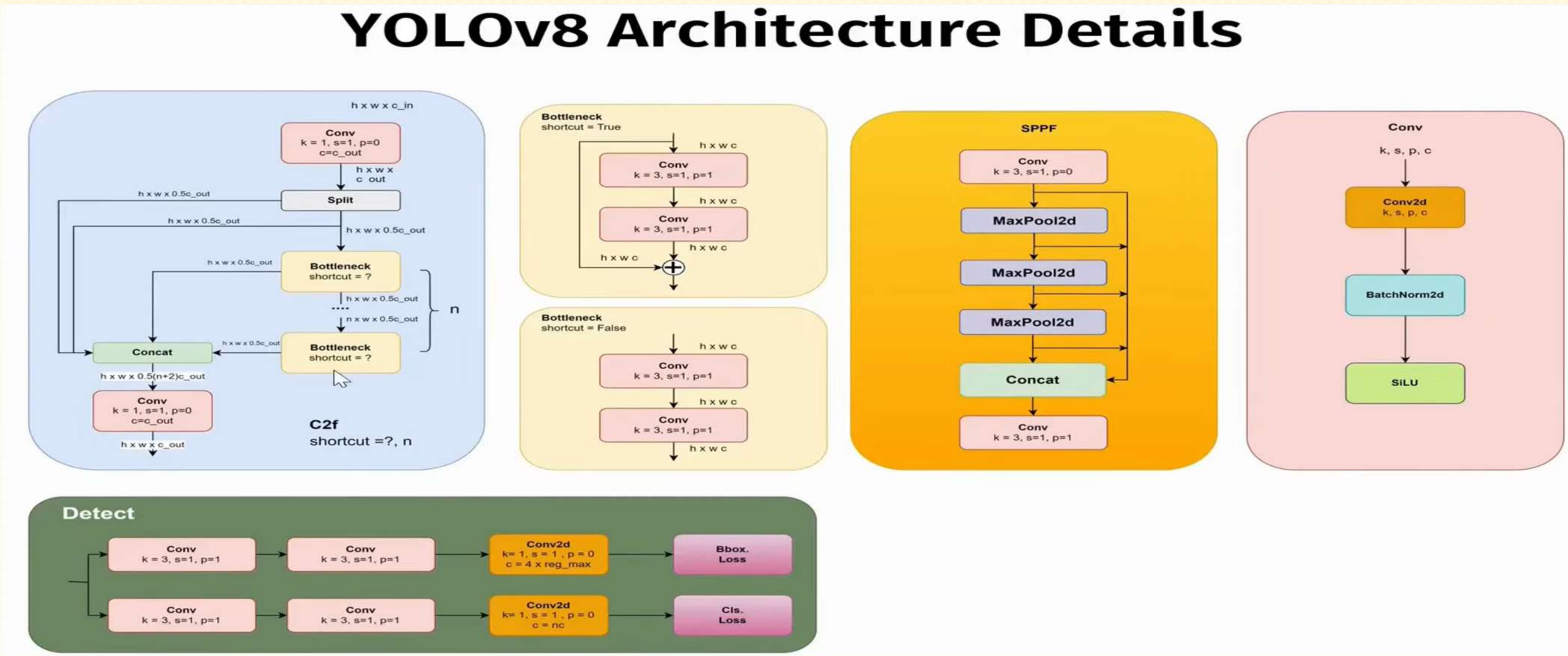
Carbohydrates Data

- Source 1: Home - mylife Diabetescare – India (mylife-diabetescare.com)
- Soruce 2: FNDR

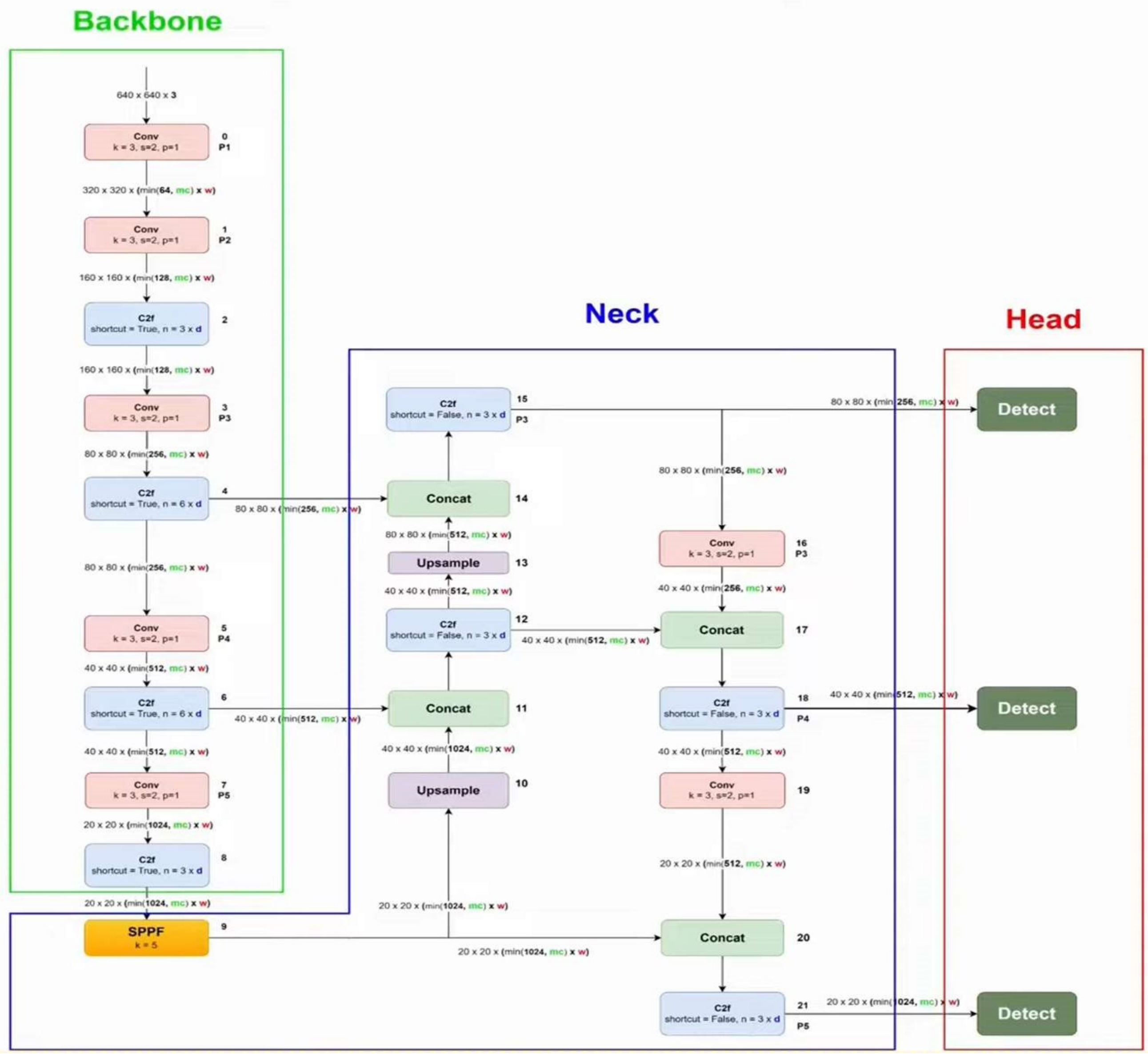
Model: YOLO v8

- YOLO is one of the most popular modules for real-time object detection and image segmentation currently considered as SOTA State-of-The-Art.
- YOLO is a convulsion neural network that predicts bounding boxes and class probabilities of an image in a single evaluation.

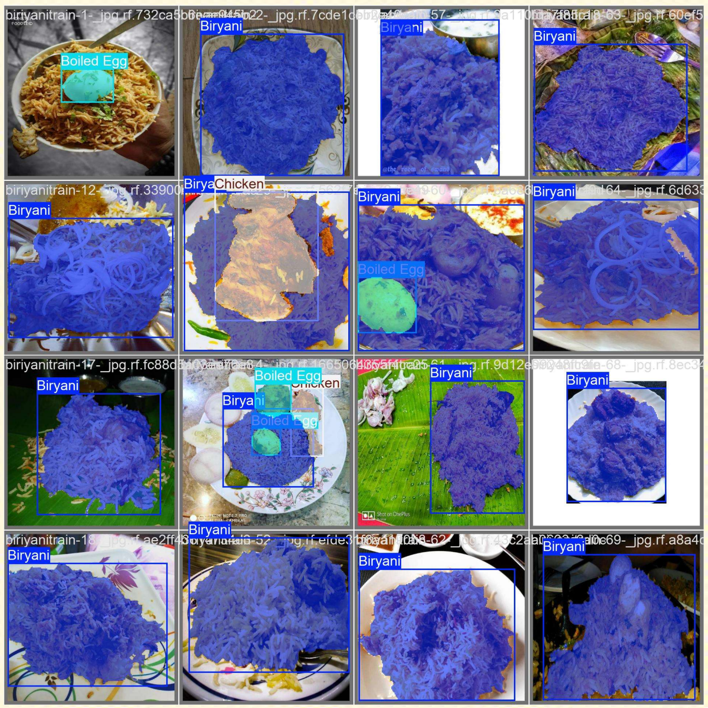
Architecture Breakdown

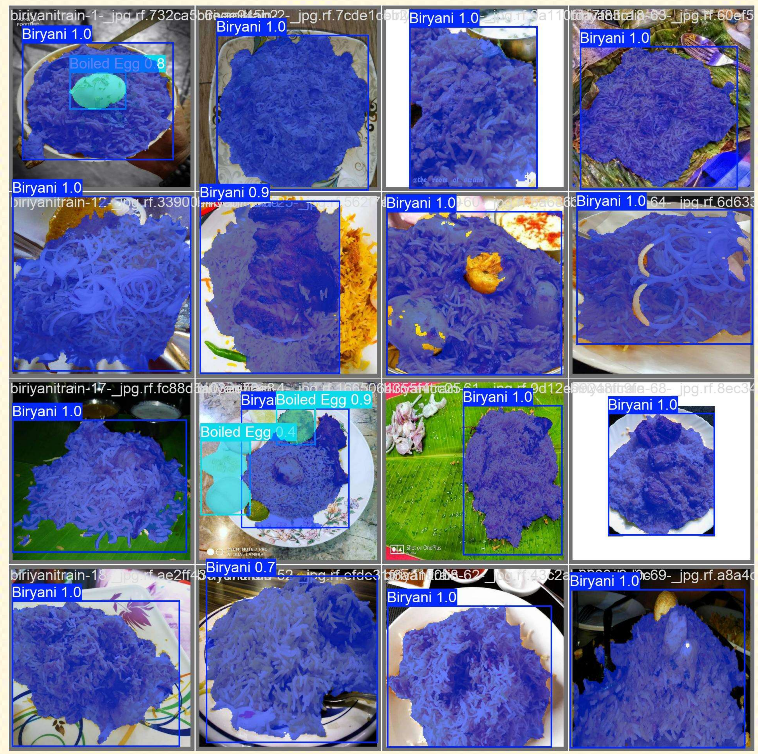


YOLOv8 Architecture

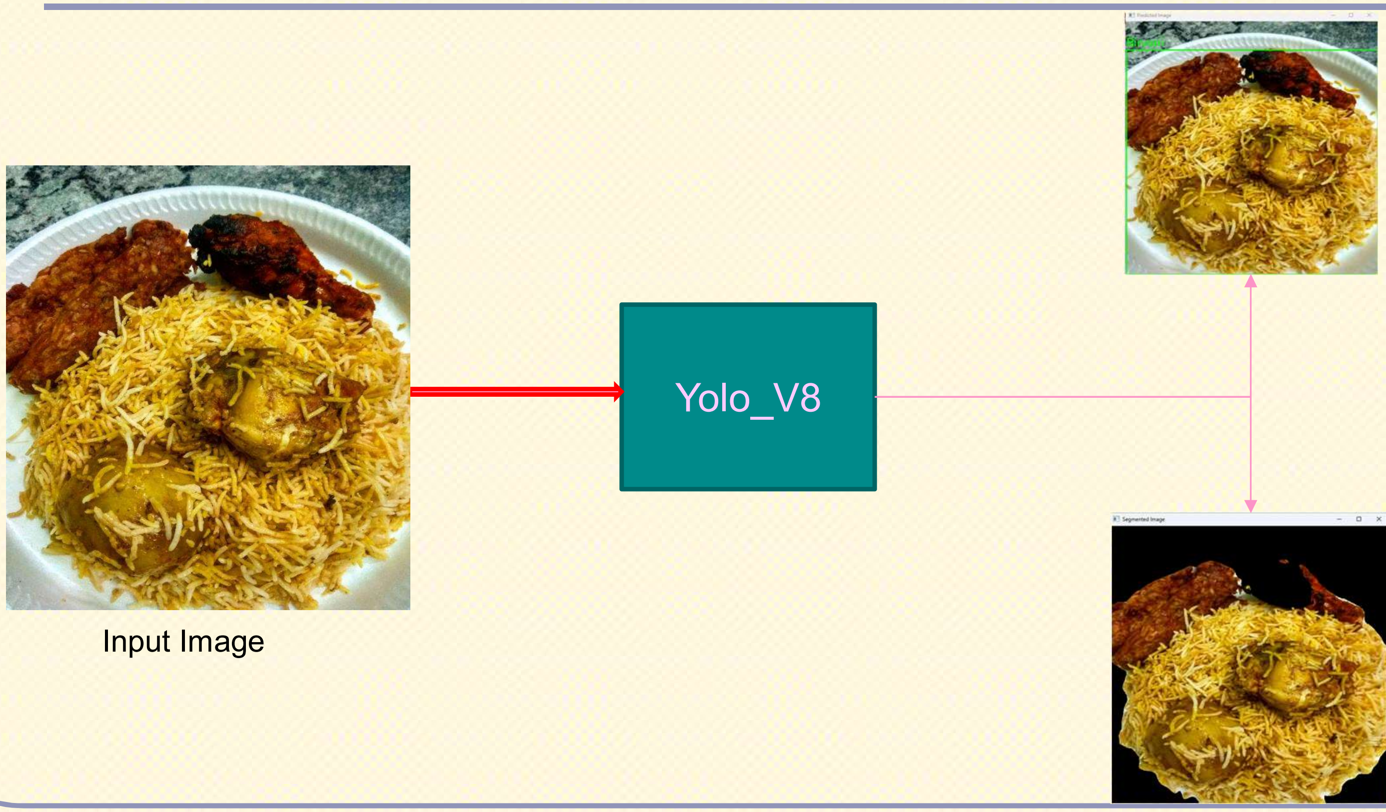


Validation Batch Results





Detection and Segmentation

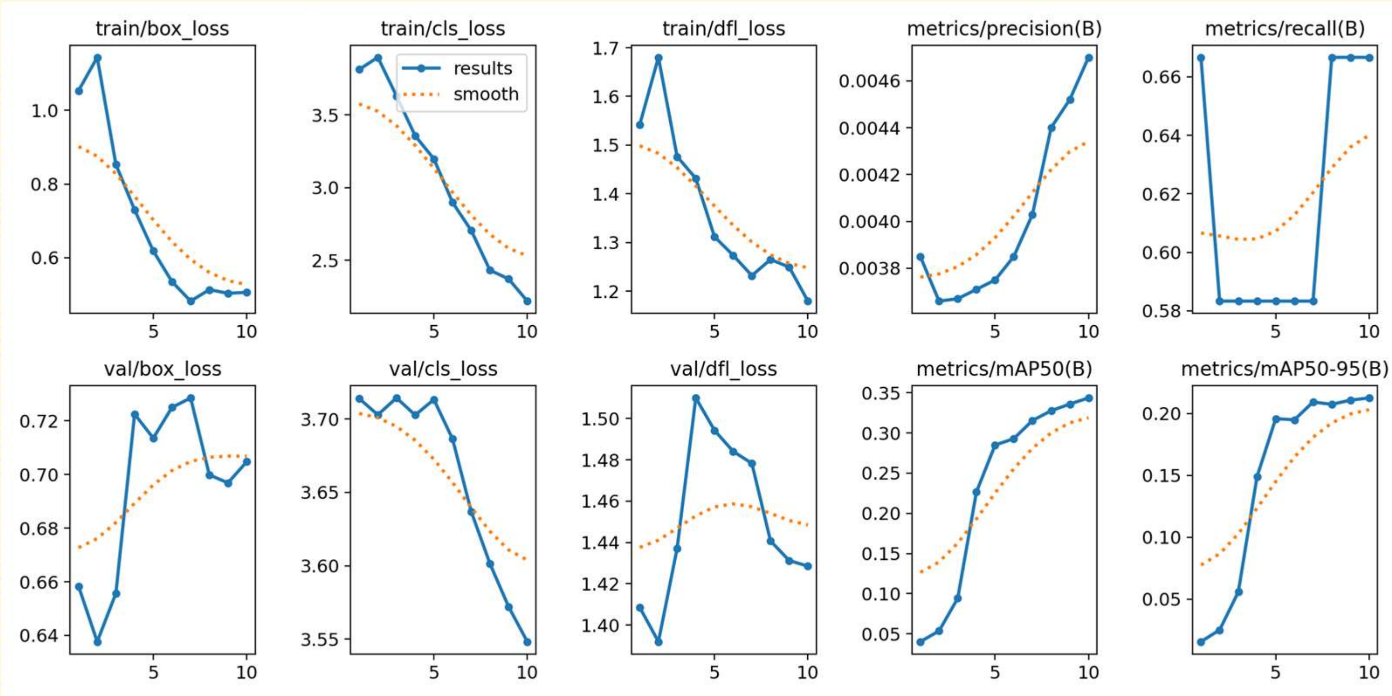


Input Image

Image Classification

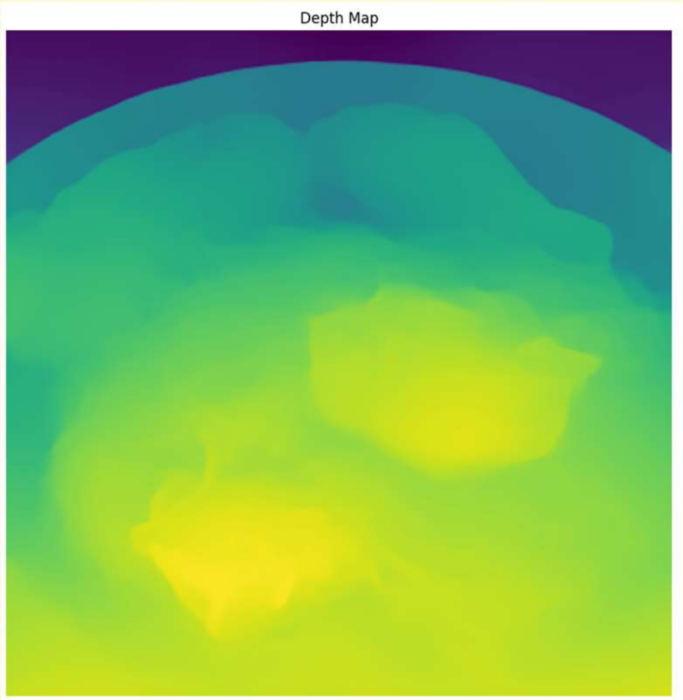
Image Segmentation

Results



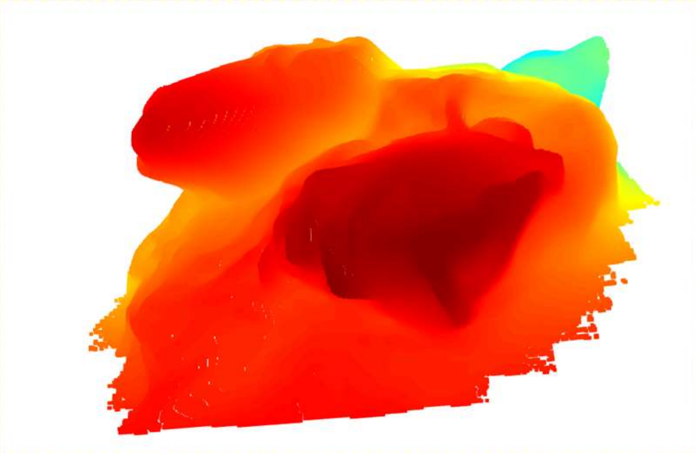
train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Segmentation Mask and depth Map



Segmentation Part

Volume Estimation Results



Point Cloud

Voxel grid

Volume predicted: 634.38 cubic cm

Mathematical Intuition

Given:
f (focal length) = 1000 (example value)
cx = width of the image / 2
cy = height of the image / 2

Depth Map to 3D Point Cloud Conversion

The goal is to convert the depth map into a 3D point cloud.

1. **Depth Map Normalization:**

$$z = \frac{\text{depth\_map}}{255.0}$$

This normalizes the depth values from 0-255 to 0-1.

2. **Indexing the Pixels:**

$$i, j = \text{np.indices}((h, w))$$

Here, i and j represent the row and column indices of each pixel in the image, respectively.

3. **3D Coordinates Calculation:**

$$x = \frac{(j - cx) \cdot z}{f}$$

$$y = \frac{(i - cy) \cdot z}{f}$$

$$\text{points} = \text{np.stack}((x, y, z), \text{axis} = -1).reshape(-1, 3)$$

These equations transform the 2D pixel coordinates into 3D world coordinates using the depth value and intrinsic parameters.

Mathematics Intuition

Voxelization and Volume Estimation

Voxelization involves converting the point cloud into a voxel grid, which is a 3D grid of small cubes (voxels).

Voxel Grid Creation:

```
voxel_grid=o3d.geometry.VoxelGrid.create_from_point_cloud(pcd, voxel_size=voxel_size)
```

2. **Volume Estimation:**

$$\text{num\_voxels} = \text{len}(\text{voxel\_grid.get\_voxels}())$$

$$\text{volume} = \text{num\_voxels} \cdot (\text{voxel\_size}^3)$$

The total volume is calculated by multiplying the number of voxels by the volume of a single voxel. Each voxel's volume is the cube of the voxel size.

$$\text{volume\_cm3} = \text{volume} \cdot 10^6$$

This converts the volume from cubic meters to cubic centimeters.

ThankYou