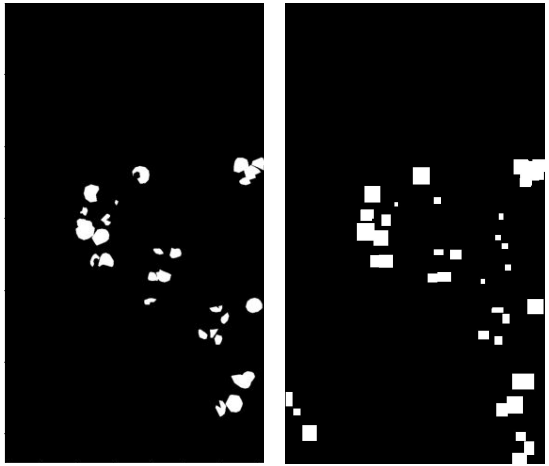
Apple Counting

Ekin Ercetin, Matthew Watkins, Hardik Gupta

Pipeline

1. Image Sequence
2. 3D Reconstruction
    a. Recover camera poses
    b. Locate ground
3. 2D Detection
    a. Locate apple clusters in each image
4. 3D Clustering
    a. Locate each cluster in 3D
5. Counting
    a. Count apple clusters from 2D view of clusters
6. Summation
    a. Merge counts for final estimate

2D Detection - Yolov8




```
{
    "annotations": [
        {
            "area": 542.0,
            "bbox": [
                282,
                915,
                38,
                41
            ],
            "category_id": 1,
            "id": 0,
            "image_id": 1,
            "iscrowd": 0,
            "segmentation": [
                [
                    295,
                    915,
                    295,
                    917,
```


`<object-class> <x> <y> <width> <height>`

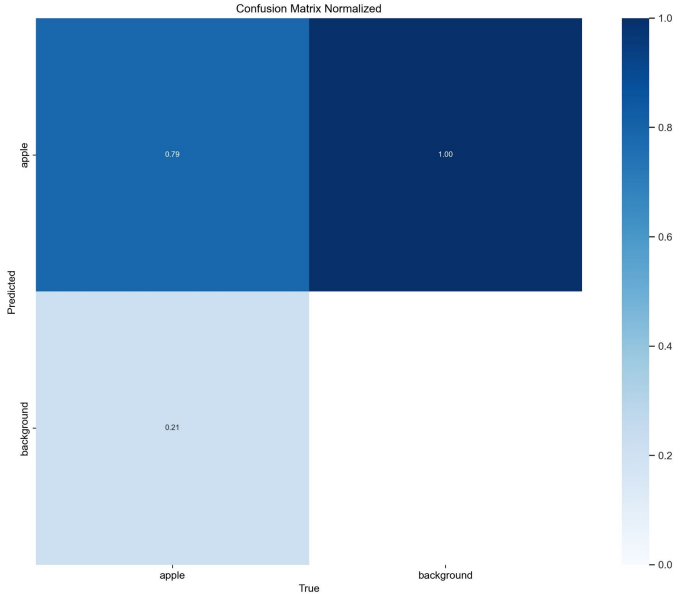

```
0 0.418056 0.730859 0.0527778 0.0320312
0 0.466667 0.714063 0.0444444 0.0234375
0 0.580556 0.661719 0.0277778 0.0125
0 0.91875 0.660547 0.0347222 0.0179687
0 0.358333 0.653906 0.025 0.015625
0 0.256944 0.635547 0.0138889 0.0164063
0 0.467361 0.625781 0.0430556 0.021875
0 0.220139 0.613281 0.0347222 0.0203125
0 0.682639 0.609375 0.0402778 0.0203125
0 0.63125 0.605469 0.0263889 0.0125
0 0.956944 0.604297 0.0361111 0.0179687
```

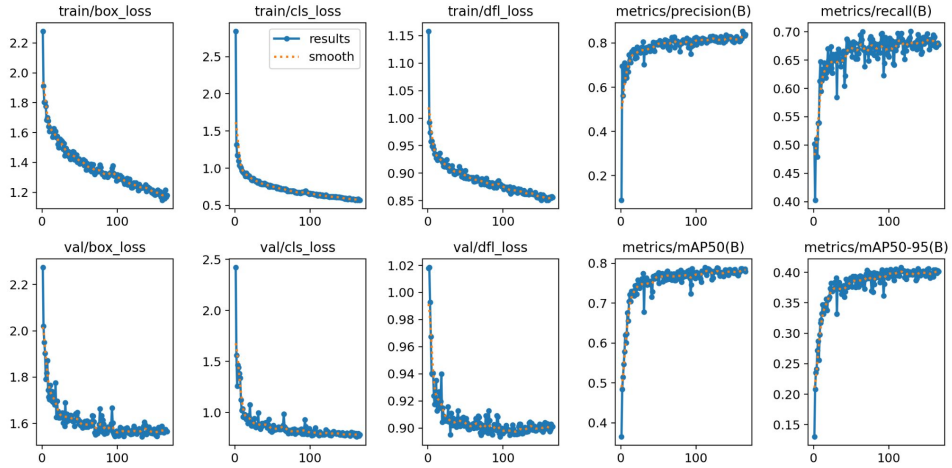
2D Detection - Yolov8 Results

Training Parameters:

epochs=200, batch=16,

learning_rate= 0.01





2D Detection - Yolov8 Results

Output:

- best.pt

```
model = YOLO('best.pt')
original_image = cv2.imread(image_path)
results = model(image_path, conf=0.4)

for r in results:
    for box_info in range(len(r.boxes)):
        a = r.boxes[box_info].xyxy[0].tolist()
        print("bounding box coordinates: ", a)
```

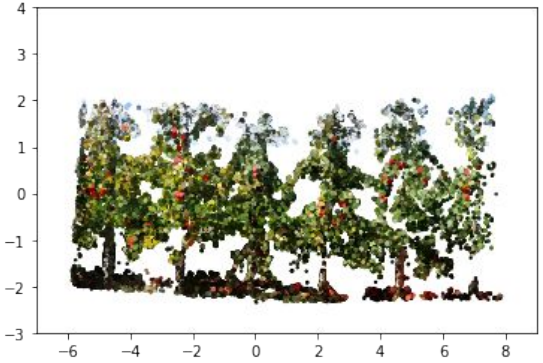
3D Reconstruction

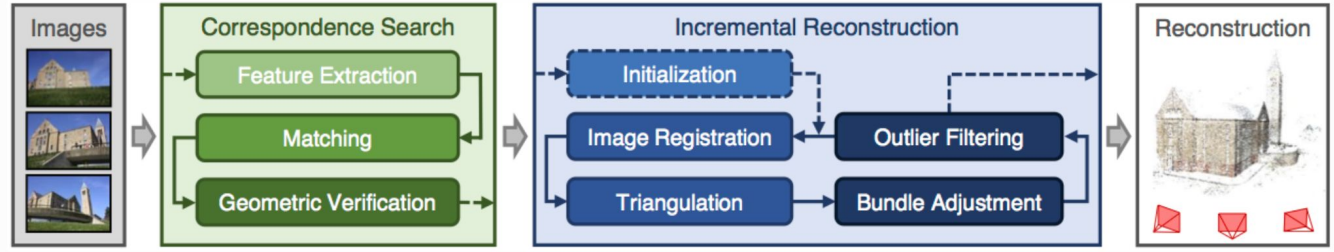
Using various 2D set of images from different view points, 3D reconstruction of the scene is done.

The following picture represents COLMAP Pipeline.

And one of the output of the COLMAP is provided.

Filtering of points are done which are on the ground

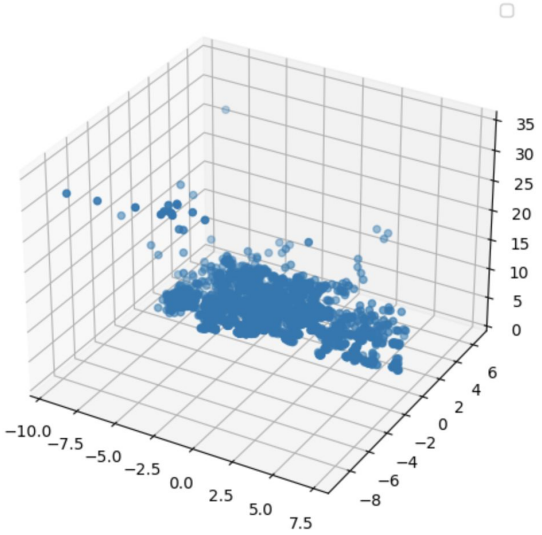




3D Projection

Following are the steps:

1. Calculation of **E**xtrinsic Matrix
2. Calculation of **K** matrix (Intrinsic Matrix)
3. Using **K** (Intrinsic Matrix), **E**xtrinsic, **M**ask → selective apple points are calculated from view of each camera, and it picks the point closest to camera for each masked pixel



3D Clustering

Cluster each 3D point to find apple clusters

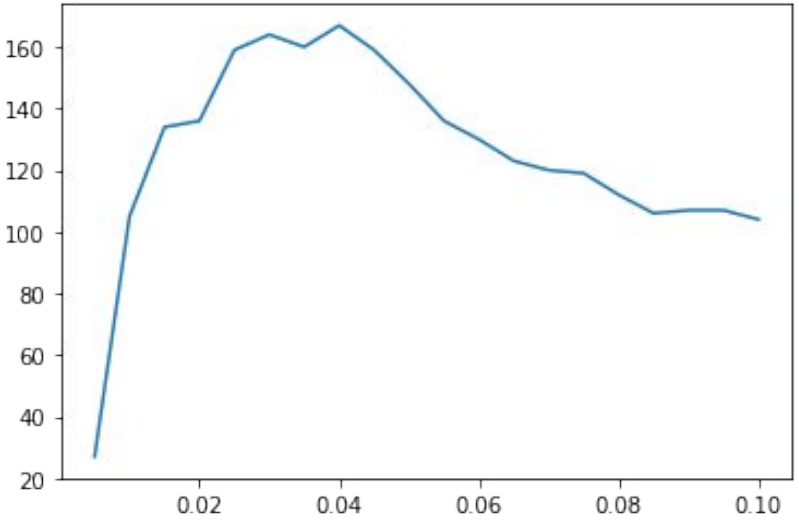
Using **DBSCAN** and **StandardScaler**,

We first perform normalisation of the data using StandardScaler and perform cluster analysis using DBSCAN

For hyperparameters,

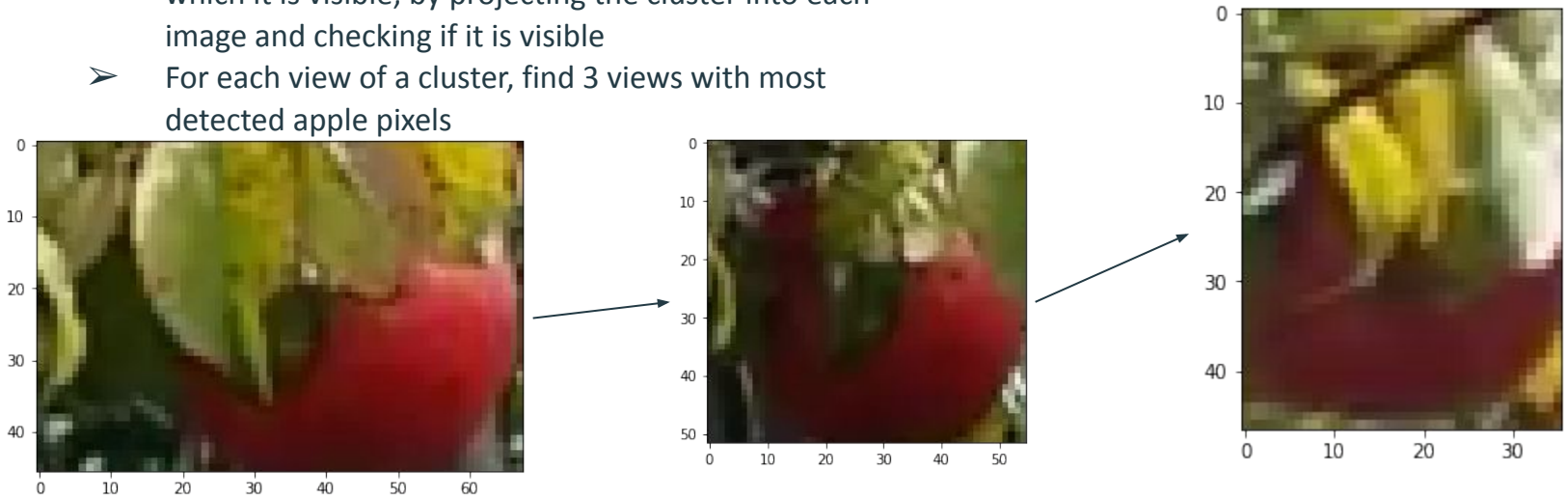
With eps = whichever eps gives highest number of clusters, (Following graph shows that calculation)

min_samples = 15



Cluster Views

- For each cluster in 3D space, we find each view from which it is visible, by projecting the cluster into each image and checking if it is visible
- For each view of a cluster, find 3 views with most detected apple pixels



Counting Model

We fine-tuned ResNet50, MobileNetV3 and ShuffleNet V2 x05, all pre-trained on ImageNet 1K. We replaced the 1K class prediction head with multiple linear layers with 7 classes, representing the number of apples in an image (0-6).

We see slight improvements with increasing model size.

For numbers of apples 3 and higher, it typically predicts lower than the truth, while for numbers 1 and lower it typically predicts higher than the truth.

	Test Set 1	Test Set 2	Test Set 3	Test Set 4	Number of Parameters (Millions)
ResNet50	84.10%	86.15%	92.86%	85.21%	23.9
MobileNet V3	82.64%	85.03%	92.69%	84.50%	5.5
ShuffleNet V2 x05	80.01%	82.64%	92.86%	83.50%	1.4

Summation

- Given the three best images for counting of a cluster, we use the counting model to predict the number of apples
- We use the median value of the predictions as our estimate for the total number of apples in each cluster
- We add up the number of apples per cluster for the total count

True Count	Front View Estimate	Back View Estimate
270	254	203

Future Work

- Implement method for merging front and back views of tree row
- Find better parameters for 3D clustering method or use a better algorithm
- Improve apple detection model and counting model, like Mask RCNN for improved masks