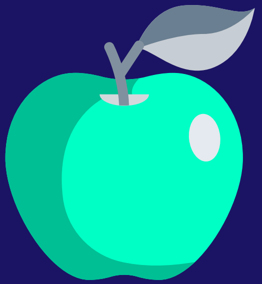
FRUIT DETECTION FROM 3D POINT CLOUDS OF TREES

SERVER KARAHAN SARIKAYA
YUSUF FURKAN DOĞAN
ALİHAN BOZKIR
EMRE KOÇAL



OUTLINE

Definition of Our Project

Introduction

Brief Introductory For The Project.

Methods and Products

Used Softwares, Methods of Project Management

Basics of Algorithm

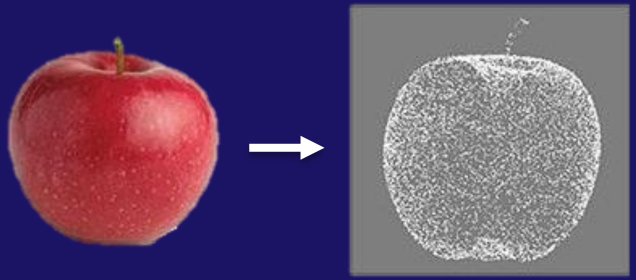
Topics Covered In Meetings Held Throughout The Project Process

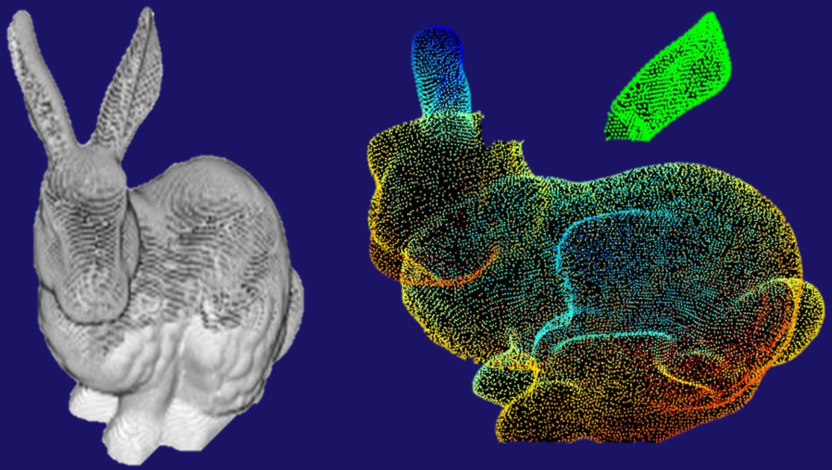
Gains and Results

Percentage Of Success And Suitability Of The Designed Algorithm

INTRODUCTION

Color, Dimension, Maturity are the spesifications those are used for detection of usage area of fruits. These spesifications, which are the main interest of the Pomology, are used in our Project to get implemented by artificial intelligence and image processing methods.





INTRODUCTION

CONCEPT

Forming a basis for future automated harvesting applications

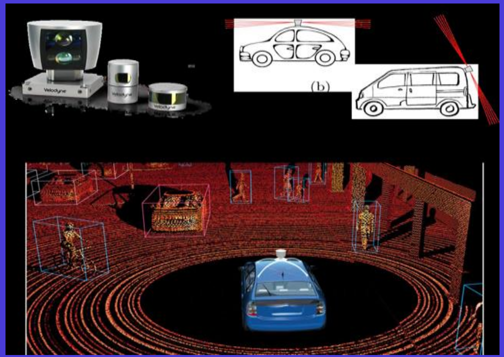
ADVANTAGE

TÜBİTAK

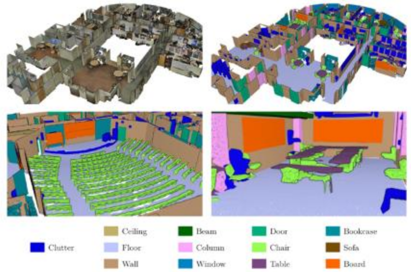
Providing opportunuties for industrial classifications

Basics of Our Methodology

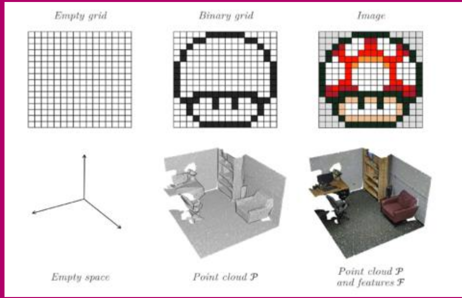
Scanning Techniques



Creating Datasets

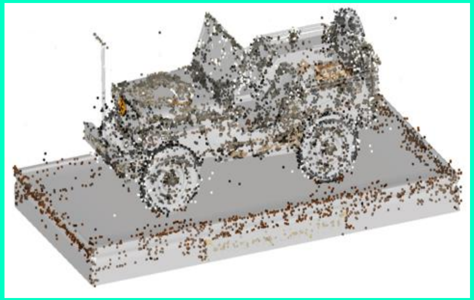


Point Clouds

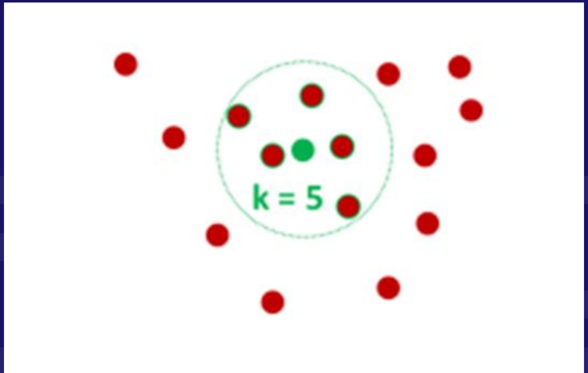


Basics of Our Methodology

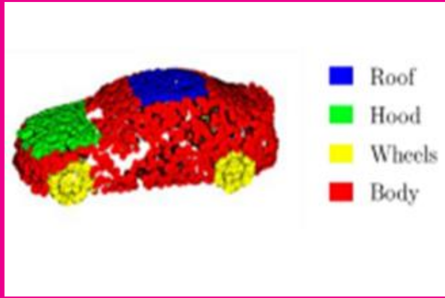
Properties of Point Clouds



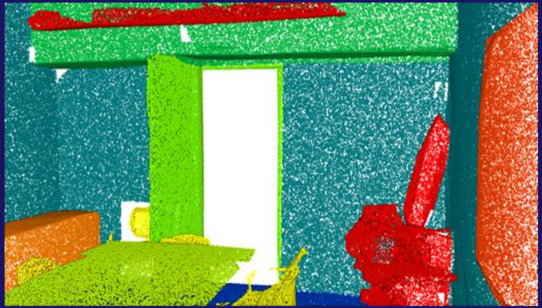
Neighborhood Implementations



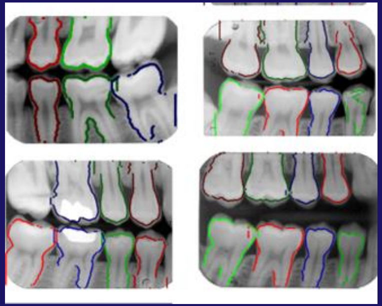
Segmentation in Point Clouds



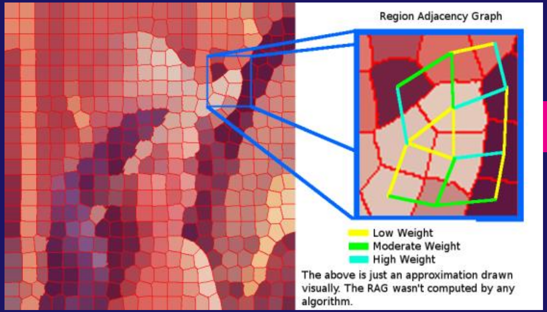
Segmentation Methods



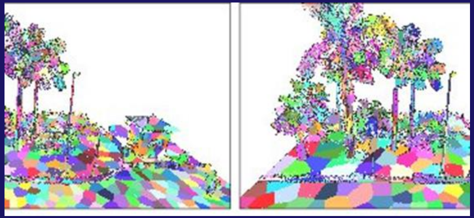
Primitive Based Segmentation



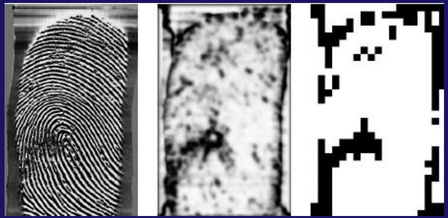
Contour Based Segmentation



Graph Based Segmentation

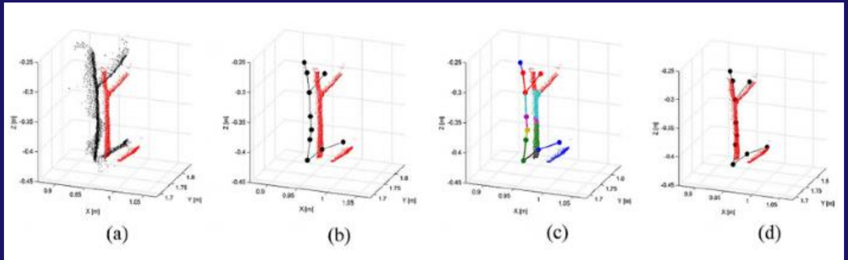


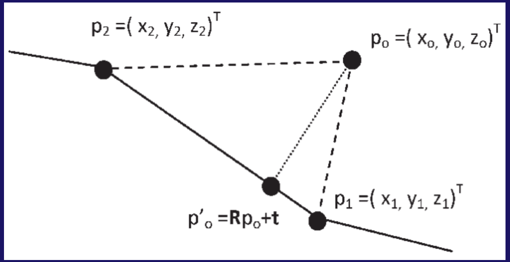
Super Voxel Based Segmentation



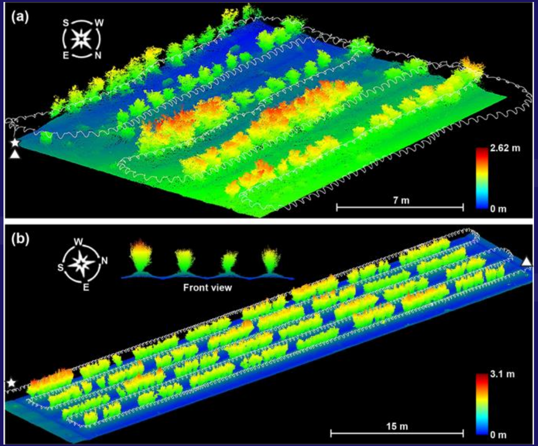
Morphology Based Segmentation

ACADEMIC AND REAL LIFE APPLICATIONS

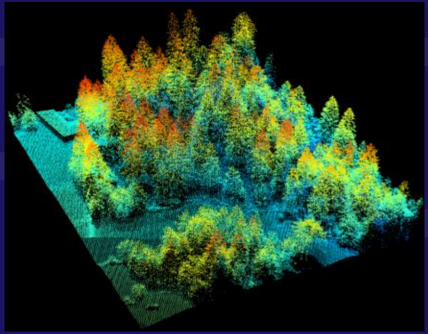




Localized Registration of Point Clouds of Botanic Trees



3D Point Cloud Data to Quantitatively Characterize Size and Shape of Shrub Crops



Forest and Road Tree Species LiDAR Classification Services

METHODS AND PRODUCTS

METASHAPE

Software program that combines raw datasets and converts them into 3D data.

MATLAB

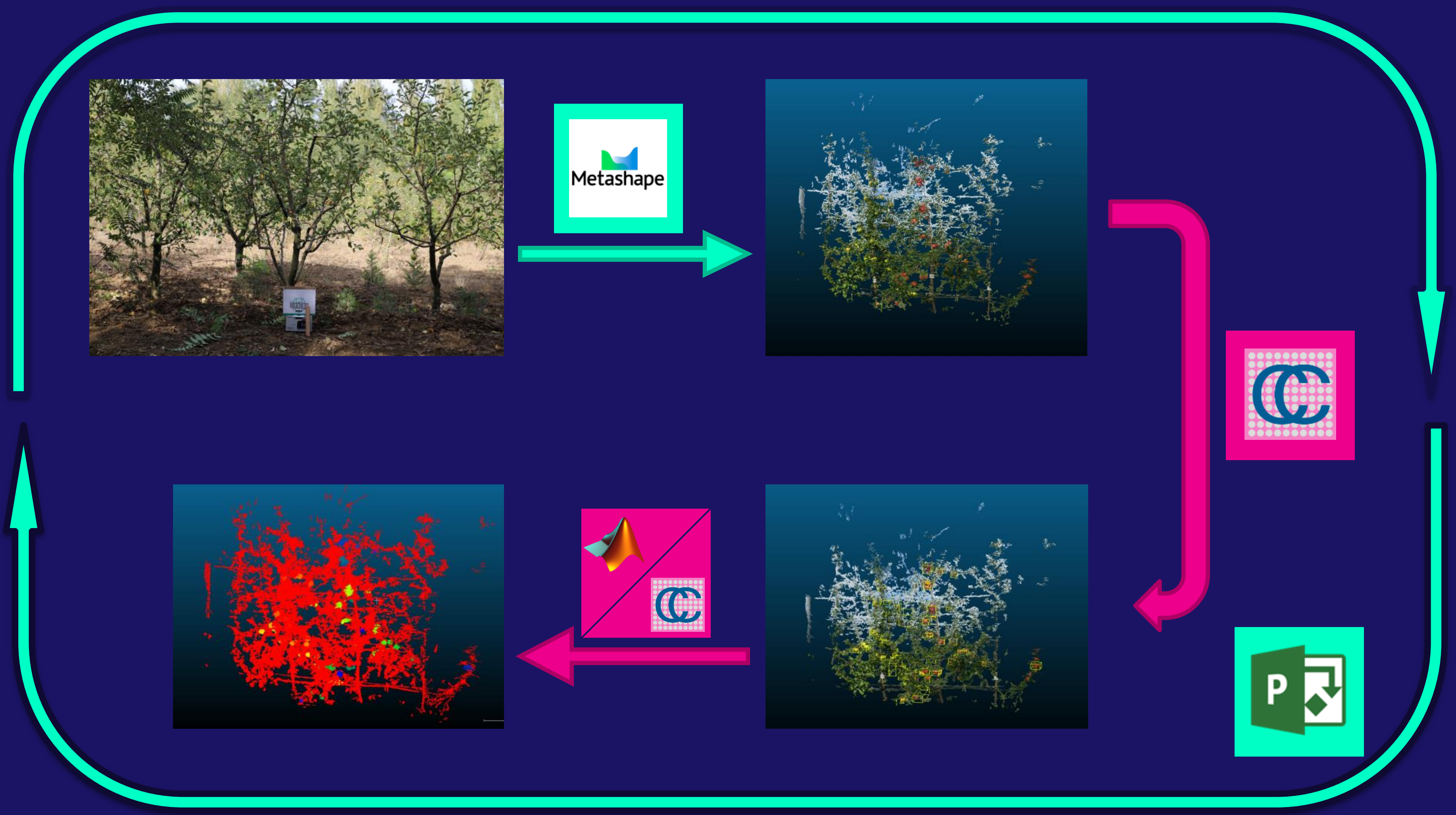
Software program for image processing and mathematical operations.

CLOUD COMPARE

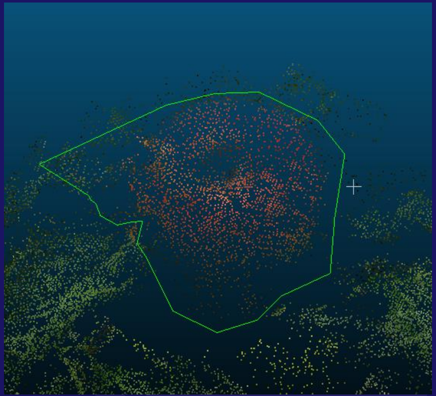
The software where point clouds are created and processed.

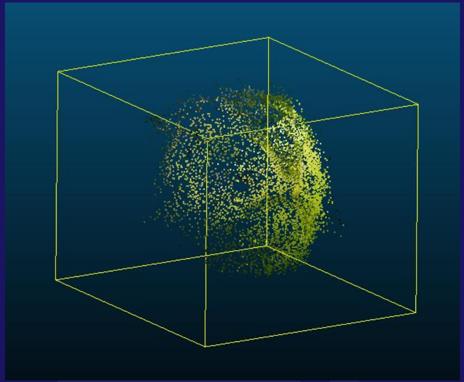
MS Project

Software for making the project management process efficient and to control the planned processes.

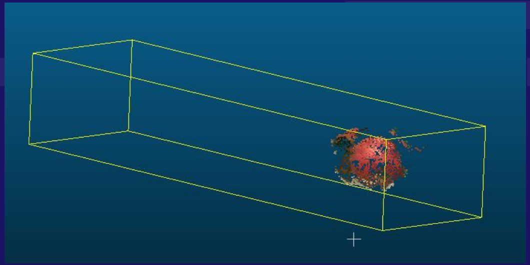

Metashape

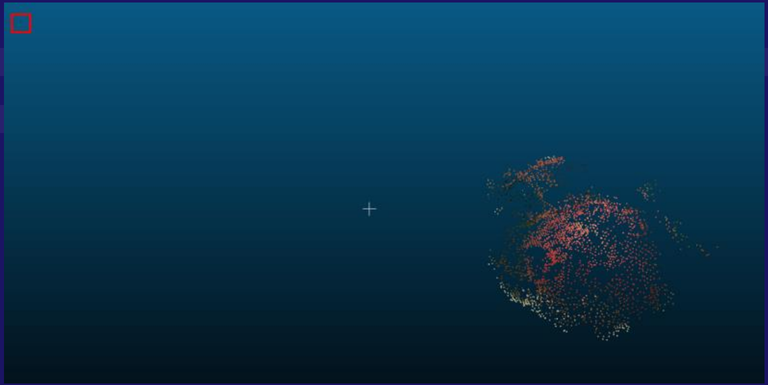
CREATING OUR DATASET



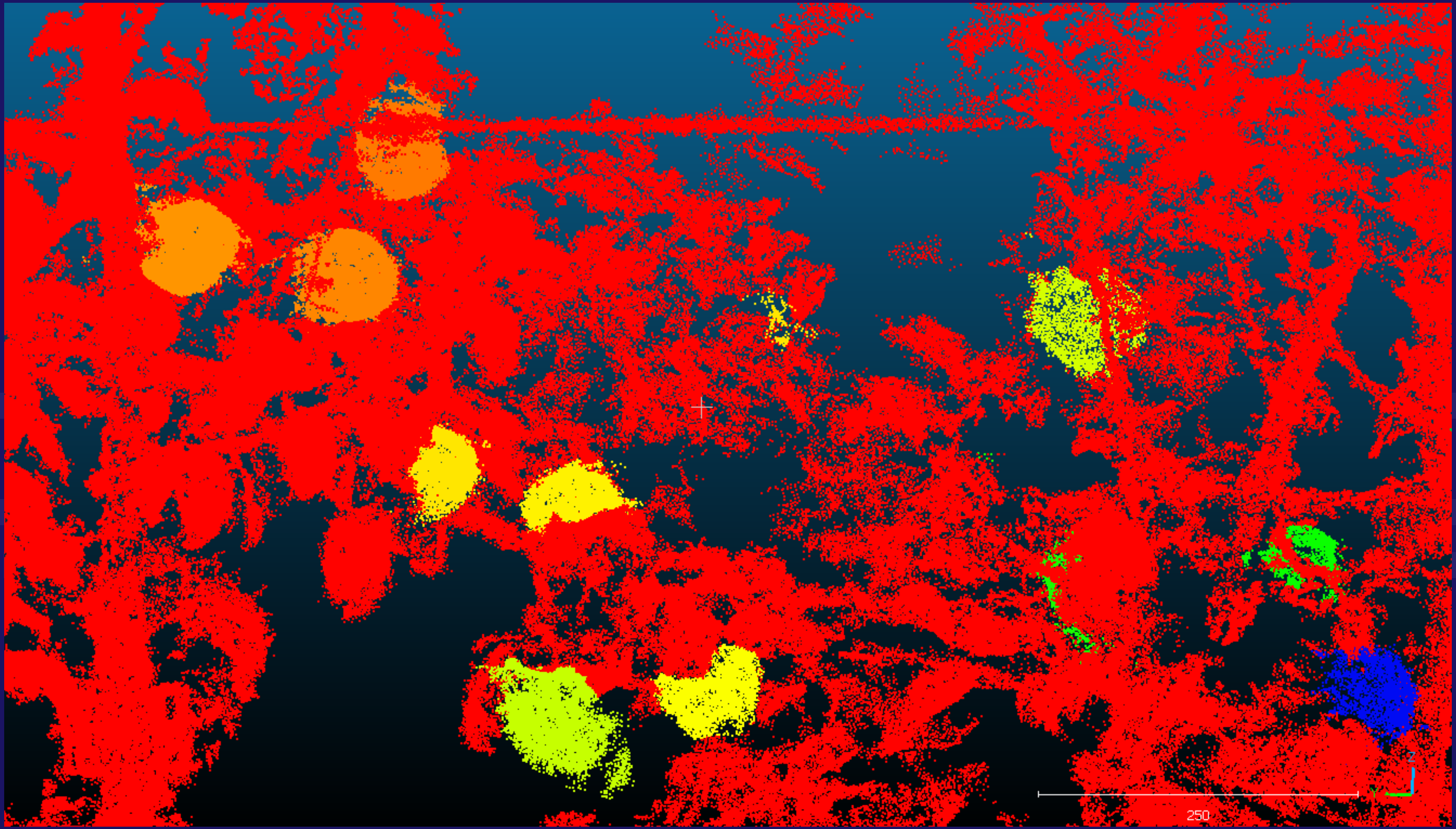


Labeling Apples and Fixing Label Issues





CREATING OUR DATASET



CREATING OUR DATASET

Our files have to be:

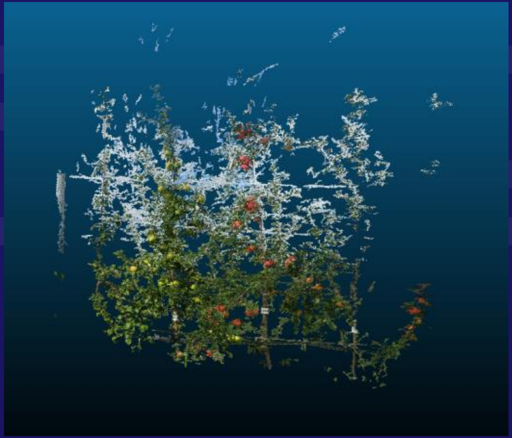
.ply

Polygon File Format or
Stanford Triangle Format
Color, Transparency,
Surface Normals,
Coordinates

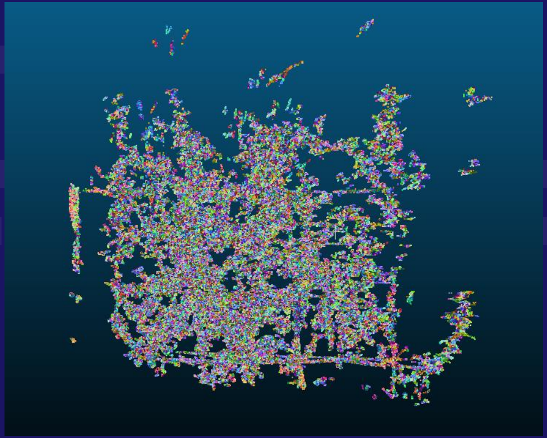
Labeled clouds have to be merged to one

merged

HSV

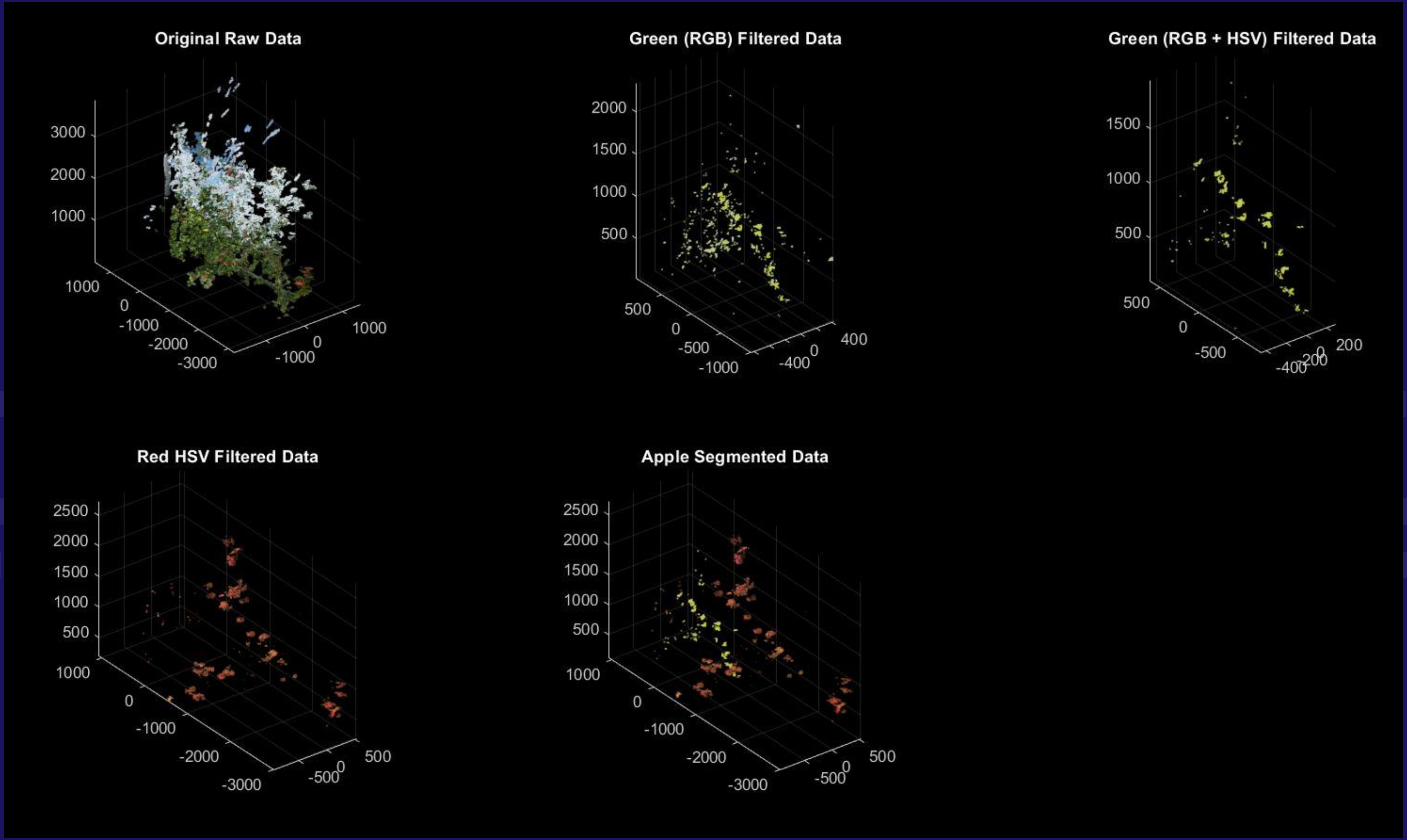


RGB

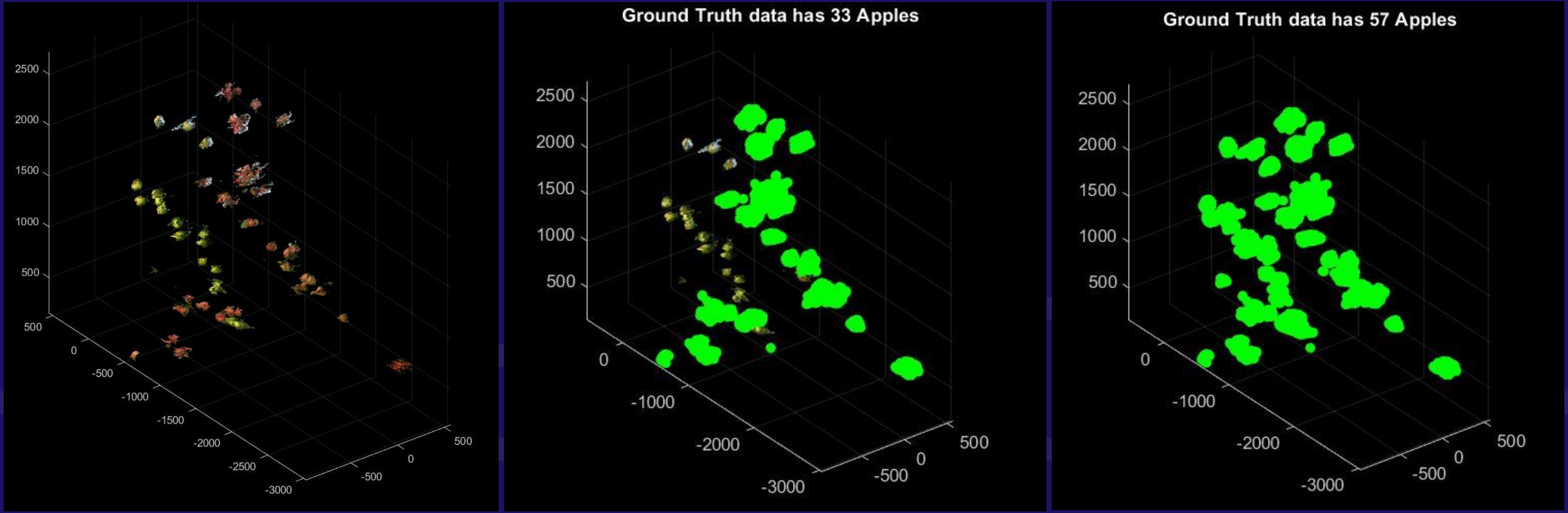


HSV

OUTPUTS BY OUR ALGORITHM

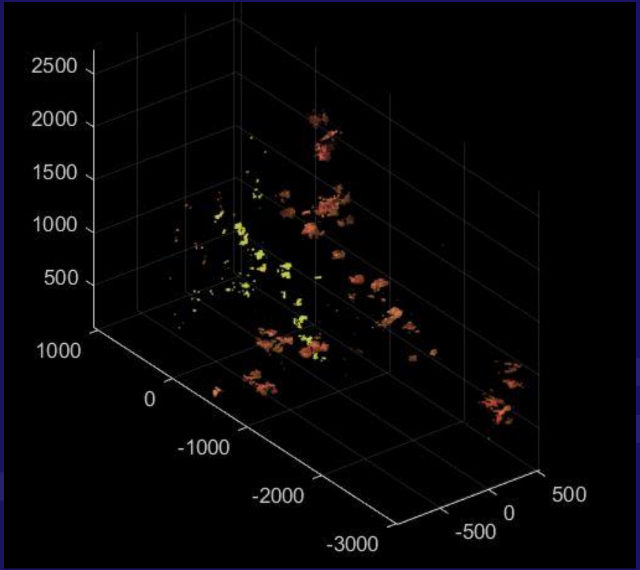


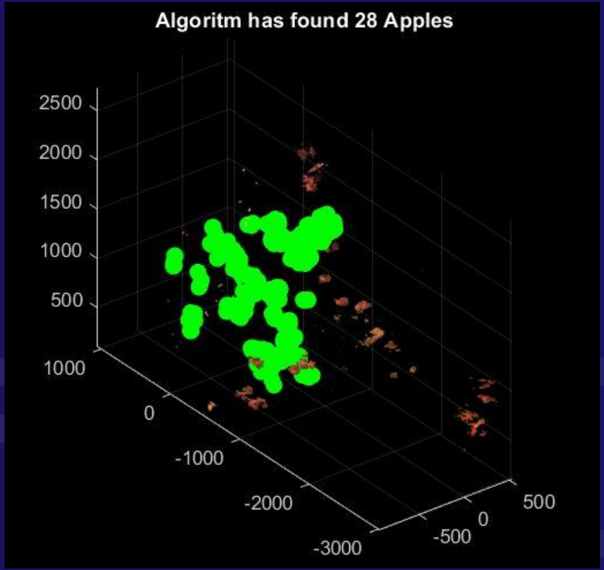
OUTPUTS BY OUR ALGORITHM

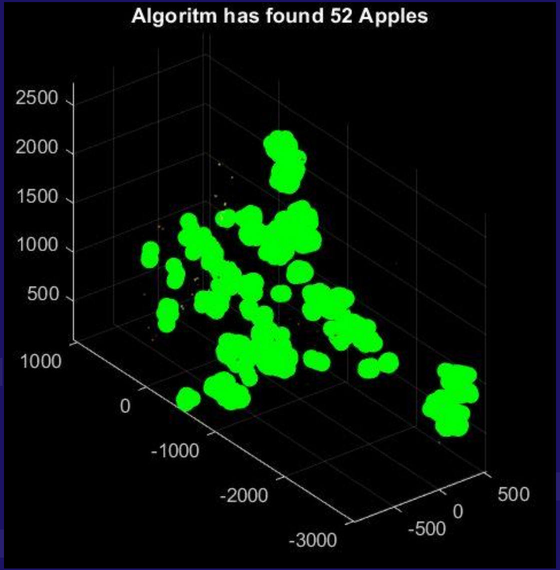


Apple Count of Ground Truth

OUTPUTS BY OUR ALGORITHM







Apple Count of Ground Truth

«find_apples» Script

```
% Load the point cloud data
load('ptCloud_ROI_123_s.mat')
point_cloud = ptCloud_ROI;

figure(1)
subplot(2,3,1);
pcshow(point_cloud);
title('Original Raw Data');
axis on;

green_channel = point_cloud.Color(:,2);

green_indices = find(green_channel > 200);
ptCloudGreen = select(point_cloud,green_indices);


%pcshow(ptCloudGreen);


% Extract the red channel of the point cloud
red_channel = ptCloudGreen.Color(:,1);

% Find the indices of the red points
red_indices = find(red_channel < 200);
ptCloudRed_Green = select(ptCloudGreen,red_indices);



%pcshow(ptCloudRed_Green);


blue_channel = ptCloudRed_Green.Color(:,3);
blue_indices = find(blue_channel < 150);
ptCloudRed_Green_Blue = select(ptCloudRed_Green,blue_indices);

ptCloudGreenSegmented = ptCloudRed_Green_Blue;

subplot(2,3,2);
pcshow(ptCloudGreenSegmented);
title('Green (RGB) Filtered Data');
axis on;

point_cloud2 = ptCloudGreenSegmented;
```

```
%
%orijinal data
hsv1 = rgb2hsv((double(point_cloud.Color))/255);
hsv1 = hsv1*255;

hue_rd = hsv1(:,1);
sat_rd = hsv1(:,2);

%yeşil rgbden çıkan data
hsv2 = rgb2hsv((double(point_cloud2.Color))/255);
hsv2 = hsv2*255;

hue_gr = hsv2(:,1);
sat_gr = hsv2(:,2);

green_apple_indices = find((hue_gr>=40 & hue_gr<=70) & ( sat_gr>=120 &
sat_gr<=220 )); %finding red and green apple indices
```

```
green_ptCloud_seg = select(point_cloud2,green_apple_indices); % segmented
apples from original cloud
%|& ( sat>120 & sat<160 )

subplot(2,3,3);
pcshow(green_ptCloud_seg);
title('Green (RGB + HSV) Filtered Data');
axis on;
%
red_apple_indices = find((hue_rd >0 & hue_rd<20) & ( sat_rd>150 &
sat_rd<200 )); %finding red and green apple indices
red_ptCloud_seg = select(point_cloud,red_apple_indices); % segmented apples
from original cloud

subplot(2,3,4);
pcshow(red_ptCloud_seg);
title('Red HSV Filtered Data');
axis on;

merged = pcmerge(red_ptCloud_seg,green_ptCloud_seg,0.001)

subplot(2,3,5);
pcshow(merged)
title('Apple Segmented Data');
axis on;

pcwrite(merged,'ptCloud_ROI_123_s_SEGMENTED.ply','Encoding','ascii');
```

«data_labeling» Script

```
fig_apple = figure
ptCloudRedSegmented = pcread('ptCloud_ROI_123_s_SEGMENTED.ply');
pcshow(ptCloudRedSegmented);

distE = 25;
L = pcsegdist(ptCloudRedSegmented,distE);
counter = 0;

clusters_XYZ_Limits=[]

for i = min(L): max(L)
    apple = select(ptCloudRedSegmented,L==i);

    title_str = sprintf("Algoritm has found %d Apples", counter);
    title(title_str);
    figure(fig_apple)

    %hold on
    if apple.Count > 50
        counter = counter + 1;
        apple;


cluster_XYZ_Limits=[apple.XLimits(1),apple.XLimits(2),apple.YLimits(
1), apple.YLimits(2),apple.ZLimits(1),apple.ZLimits(2)];
```

```
        clusters_XYZ_Limits = [clusters_XYZ_Limits;
cluster_XYZ_Limits]; % Alt listeyi ana liste ile birleştir

        xmin = apple.XLimits(1);
        ymin = apple.YLimits(1);
        zmin = apple.ZLimits(1);
        xmax = apple.XLimits(2);
        ymax = apple.YLimits(2);
        zmax = apple.ZLimits(2);


        cuboid = images.roi.Cuboid(gca, 'Position', [xmin, ymin,
zmin, xmax-xmin, ymax-ymin, zmax-zmin]);
        cuboid.FaceAlpha = 0.3; % Küpün yüzeyine şeffaflık eklemek
için
        cuboid.EdgeColor = 'B'; % Küpün kenar rengini kırmızı yapmak
için

%          x = apple.Location;
%          plot3(x(:,1),x(:,2),x(:,3),'g.','MarkerSize',30)
       % pause



    else
        continue
    end
end
```

«gt_labeling» Script

```
figure_apple = figure

labeled_ply_filename = 'ptCloud_ROI_123_s.ply';
ptcloud = pcread(labeled_ply_filename);
pcshow(ptcloud)
C = read_label(labeled_ply_filename);
C = C+1;    % En küçük label 1 olsun
%figure
%colormap(hsv(max(C)))
%pcshow(ptcloud.Location,C)


counter = 0;


remove_tree = select(ptcloud,C~=max(C));

clusters_Limits= []
yedek_remove_tree = remove_tree
figureNew= figure
pcshow(remove_tree)
```

```
for i = min(C):max(C)-1
    apple = select(ptcloud,C==i);

    counter = counter + 1;
    title_str = sprintf('Ground Truth data has %d Apples', counter);
    title(title_str);
    figure(figureNew)



    apple;

    cluster_Limits = [apple.XLimits(1),apple.XLimits(2),apple.YLimits(1),
apple.YLimits(2),apple.ZLimits(1),apple.ZLimits(2)];

    clusters_Limits = [clusters_Limits; cluster_Limits]; % Alt listeyi ana
liste ile birleştir

    xmin = apple.XLim «find_apples» Script
    ymin = apple.YLimits(1);
    zmin = apple.ZLimits(1);
    xmax = apple.XLimits(2);
    ymax = apple.YLimits(2);
    zmax = apple.ZLimits(2);


    cuboid = images.roi.Cuboid(gca, 'Position', [xmin, ymin, zmin, xmax-
xmin, ymax-ymin, zmax-zmin]);
    cuboid.FaceAlpha = 0.3; % Küpün yüzeyine şeffaflık eklemek için
    cuboid.EdgeColor = 'b'; % Küpün kenar rengini kırmızı yapmak için

    %x = apple.Location;
    %plot3(x(:,1),x(:,2),x(:,3),'g.','MarkerSize',20)


end
```

«label_from_gt» Script

```
function C = read_label(filename)

fid = fopen(filename);
s = fgetl(fid);
ss = textscan(s,'%s');
ss = ss{1};

num_points = 0;

while 1

    if length(ss) == 3
        if isequal('element',ss{1}) &&  isequal('vertex',ss{2})
            num_points = str2num(ss{3});
            break
        end
    end

    s = fgetl(fid);
    ss = textscan(s,'%s');
    ss = ss{1};

end
```

```
C = zeros(num_points,1);


while not(isequal('end_header',s))
    s = fgetl(fid);
end

formatSpec = '%f %f %f %d %d %d %f';
sizeA = [7,Inf];
C = fscanf(fid,formatSpec,sizeA);
C = C(end,:);
C = C';

% for i = 1:num_points
%     s = fgetl(fid);
%     ss = textscan(s,'%s');
%     ss = ss{1};
%     C(i) = str2num(ss{end});
%     disp([i num_points])
% end

fclose(fid);
```

«final_calculations» Script

```
clusters_Limits; % gt

gt_apple_number = size(clusters_Limits);
gt_apple_number = gt_apple_number(1);

algo_apple_number = size(clusters_XYZ_Limits);
algo_apple_number = algo_apple_number(1);

TP = 0; % elma tespit ettim ve elma çıktı
TN = 0; % elma tespit ettim ama elma çıkmadı

indices_TP=[];
indices_TN=[];
templist_true = 1;
templist_neg = 0;

indice_exact_True = []
indice_exact_False = []

indice_founded_apples_inOurAlgo = []
```

```
for  i = 1 : gt_apple_number
    a= [];

    gt_Xmin = clusters_Limits(i,1);
    gt_Xmax = clusters_Limits(i,2);
    gt_Ymin = clusters_Limits(i,3);
    gt_Ymax = clusters_Limits(i,4);
    gt_Zmin = clusters_Limits(i,5);
    gt_Zmax = clusters_Limits(i,6);

    for j = 1 : algo_apple_number

        algo_Xmin = clusters_XYZ_Limits(j,1);
        algo_Xmax = clusters_XYZ_Limits(j,2);
        algo_Ymin = clusters_XYZ_Limits(j,3);
        algo_Ymax = clusters_XYZ_Limits(j,4);
        algo_Zmin = clusters_XYZ_Limits(j,5);
        algo_Zmax = clusters_XYZ_Limits(j,6);

        iou =
calculateIOU3D([algo_Xmin,algo_Ymin,algo_Zmin,algo_Xmax,algo_Ymax,al
go_Zmax], [gt_Xmin,gt_Ymin,gt_Zmin,gt_Xmax,gt_Ymax,gt_Zmax]);
        list = [iou];
        a = [a;list];

    end

    if max(a)>0

        fprintf('apple %d- iout = %f\n',i,max(a))
        indices_TP = [indices_TP,templist_true];
        indices_TN = [indices_TN,templist_neg];
        %fprintf('GT_apple_%d: cluster:%d \n\n',i,find(a == max(a)))
        indice_exact_True = [indice_exact_True,i];
```

«IOU» Script

```
function iou = calculateIOU3D(bbox1, bbox2)
    % bbox1 ve bbox2 formatı: [xmin ymin zzmin xmax ymax zmax]

    % Bbox1 koordinatları
    x1 = bbox1(1);
    y1 = bbox1(2);
    z1 = bbox1(3);
    x1max = bbox1(4);
    y1max = bbox1(5);
    z1max = bbox1(6);

    % Bbox2 koordinatları
    x2 = bbox2(1);
    y2 = bbox2(2);
    z2 = bbox2(3);
    x2max = bbox2(4);
    y2max = bbox2(5);
    z2max = bbox2(6);

    % Bbox1'in koordinatlarına göre köşe noktalarını hesapla
    bbox1_top_left = [x1, y1, z1];
    bbox1_bottom_right = [x1max , y1max , z1max];

    % Bbox2'nin koordinatlarına göre köşe noktalarını hesapla
    bbox2_top_left = [x2, y2, z2];
    bbox2_bottom_right = [x2max, y2max , z2max];

    % Kesişim alanının köşe noktalarını hesapla
    intersection_top_left = max(bbox1_top_left, bbox2_top_left);
    intersection_bottom_right = min(bbox1_bottom_right, bbox2_bottom_right);
```

```
    % Kesişim alanının genişlik, yükseklik ve derinliğini hesapla
    intersection_width = intersection_bottom_right(1) - intersection_top_left(1);
    intersection_height = intersection_bottom_right(2) - intersection_top_left(2);
    intersection_depth = intersection_bottom_right(3) - intersection_top_left(3);

    % Kesişim alanının negatif değerleri kontrol et
    if intersection_width <= 0 || intersection_height <= 0 || intersection_depth <= 0
        iou = 0; % Kesişim alanı yok
        return;
    end

    % Kesişim alanını hesapla
    intersection_volume = intersection_width * intersection_height *
intersection_depth;

    % Bbox1 ve bbox2 hacimlerini hesapla
    bbox1_volume = (x1max-x1) * (y1max-y1) * (z1max-z1);
    bbox2_volume = (x2max-x2) * (y2max-y2) * (z2max-z2);

    % Birleşim hacmini hesapla
    union_volume = bbox1_volume + bbox2_volume - intersection_volume;

    % IOU'yu hesapla
    iou = (intersection_volume / union_volume)*100;
end
```

RESOURCES

- Dutagaci H, Rasti P, Galopin G, Rousseau D. ROSE-X: an annotated data set for evaluation of 3D plant organ segmentation methods. Plant methods. 2020; 16(1):1–14. https://doi.org/10.1186/s13007-020- 00573-w
- Chebrolu N, Lottes P, Schaefer A, Winterhalter W, Burgard W, Stachniss C. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. Intl Journal of Robotics Research (IJRR). 2017. https://doi.org/10.1177/0278364917720510
- Roynard X, Deschaud JE, Goulette F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. The International Journal of Robotics Research. 2018; 37(6):545–557. https://doi.org/10.1177/0278364918767506

THANKS!