
esri
THE SCIENCE OF WHERE
Object Extraction from LiDAR with Deep Learning

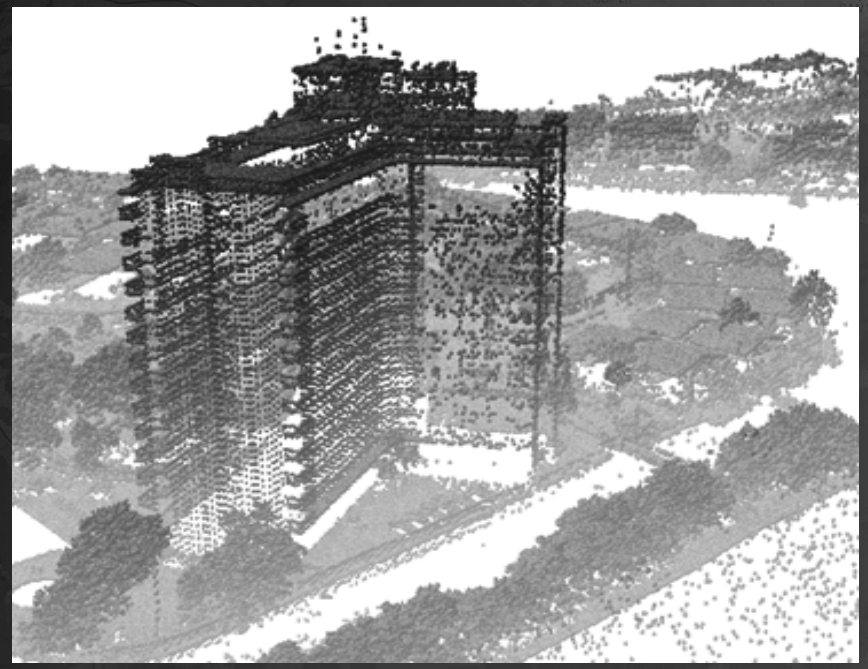
Digital Twin: valuable and expensive

- Digital Twins are important for efficient local government operations, urban planning and design, insurance & safety, etc.
- Creation and maintenance of Digital Twin is expensive and requires constant inflow of human resources and funds to maintain.
- Traditional raw data sources:
  - LiDAR Point clouds
  - Photogrammetry and resulting triangulated meshes

Unclassified point clouds and **continuous** triangulated meshes

- Both sources require segmentation before they can be used in analysis

- In its raw form, are easily interpretable by humans, but not by algorithms



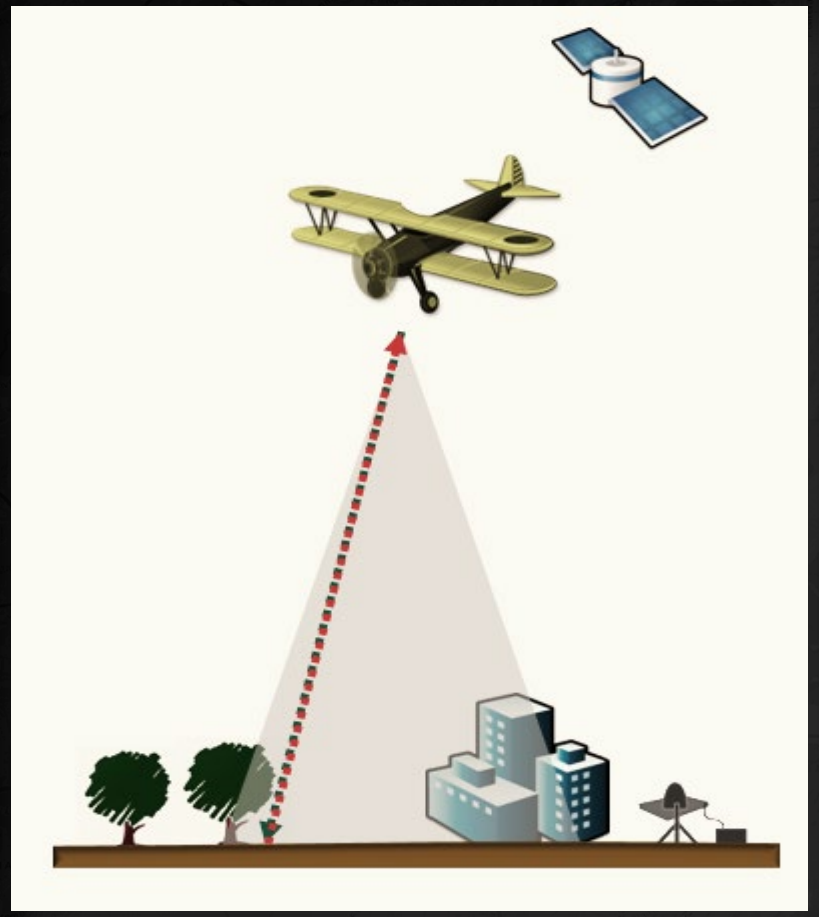
- *How many buildings are in the block?*

- *Estimate # of people in the buildings which are 5+ stores-high on Wednesday from 9-10AM*

- *What are the windows looking at this part of the park?*

What is LiDAR data?

Massive 3D point collections with additional attributes like:

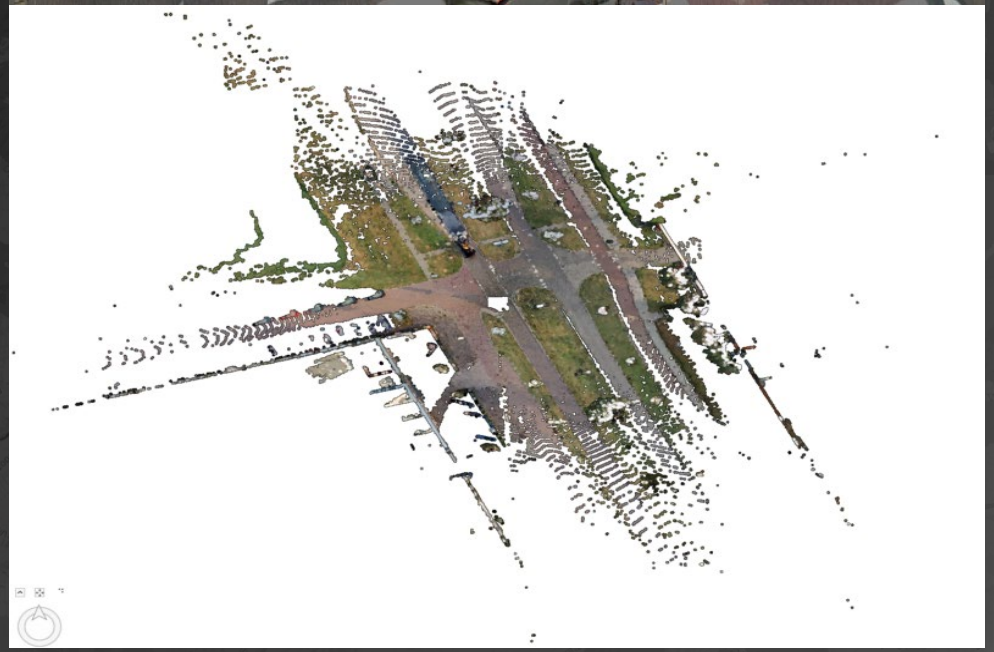


- Intensity,
- Number of Returns,
- Scan Angle,
- RGB,…

What is LiDAR data?

Massive 3D point collections with additional attributes like:

- Intensity,
- Number of Returns,
- Scan Angle,
- RGB,…



What can be extracted from point clouds?

- 3D building models
  - schematic, or
  - detailed (up to RGB+D textures)
- Electric grid
  - overhead conductors, poles, stay wires, transformers, etc.
- Trees
  - canopy polygons, precise height
- Street furniture
  - Signs, traffic lights, dividers, bus stops, fire hydrants, etc.

3D Building models

from airborne LiDAR

esri

THE SCIENCE OF WHERE

Digital Twins & 3D Building Models: Realism vs Cubism

Two directions / types of models:

1. High-fidelity models of historical or culturally-significant buildings, which look is frozen or changes very little in time:
   - Manual model construction
   - Significant initial investments, require little to maintain

2. Schematic models for commercial, industrial, residential zones – zones which are subject to constant changes:
   - Largest in area, number of buildings, number of people hosted
   - Periodic updates are required
   - Data collection and models update should be quick, accurate, cost-efficient

PoC: 3D Building model reconstruction from aerial LiDAR

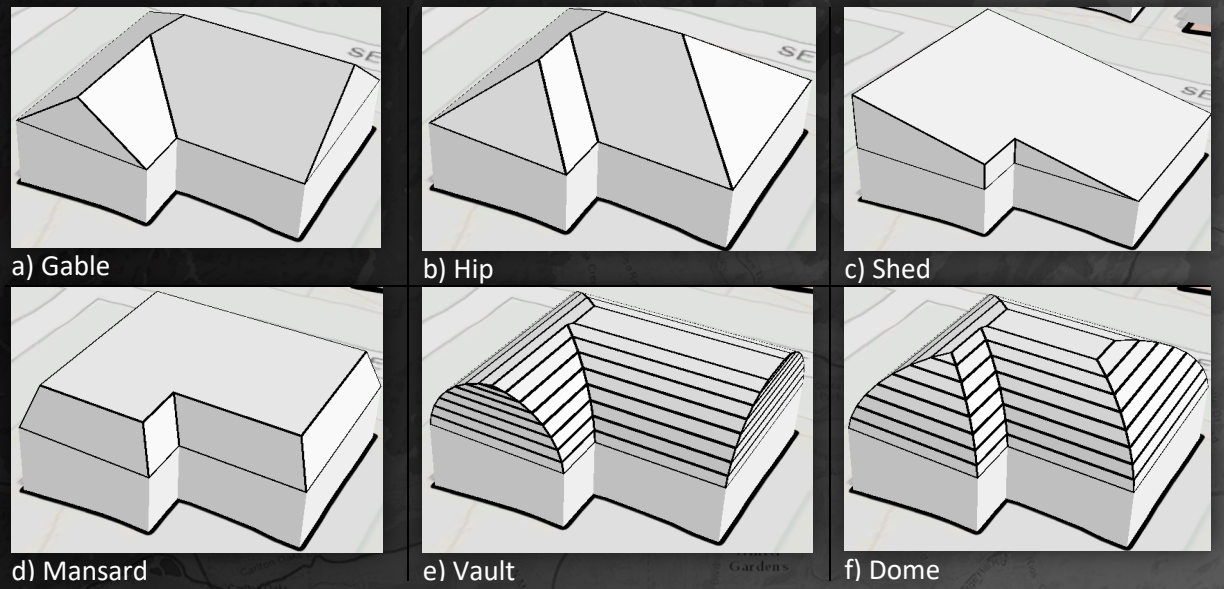
Rasterized Aerial LiDAR

Manually digitized Hip (purple) and Gable (orange) segments

3D reconstruction of building using manually digitized segments

PoC: 3D Building model reconstruction from aerial LiDAR

- Manually digitizing roof segments:
  - Over 3,000 man hours were spent on digitizing about 213,000 polygons covering the area of 200 square miles.
  - ~71 polygons / man hour.



PoC: 3D Building model reconstruction from aerial LiDAR

- Using Mask R-CNN to digitize roof segments
- Not as accurate as humans,
  but much faster: 60,000 polygons / hour.
- Regularize Building Footprints helps with accuracy.

Manually digitized “ground truth” data from the Test set

Prediction produced by the neural network


Point Cloud Layer
Miami_Demo - Scene - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Map
Scene
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
388 ft
944,466.04E 556,528.78N ftUS
3.453 ft
Selected Features: 0
PoC: Schematic 3D building models with Mask R-CNN

But can we work in true 3D (without rasterization)?

There is a traditional workflow with deterministic GP tools:

1. **ClassifyLASGround**
2. **ClassifyLASBuilding**

To get building footprints:

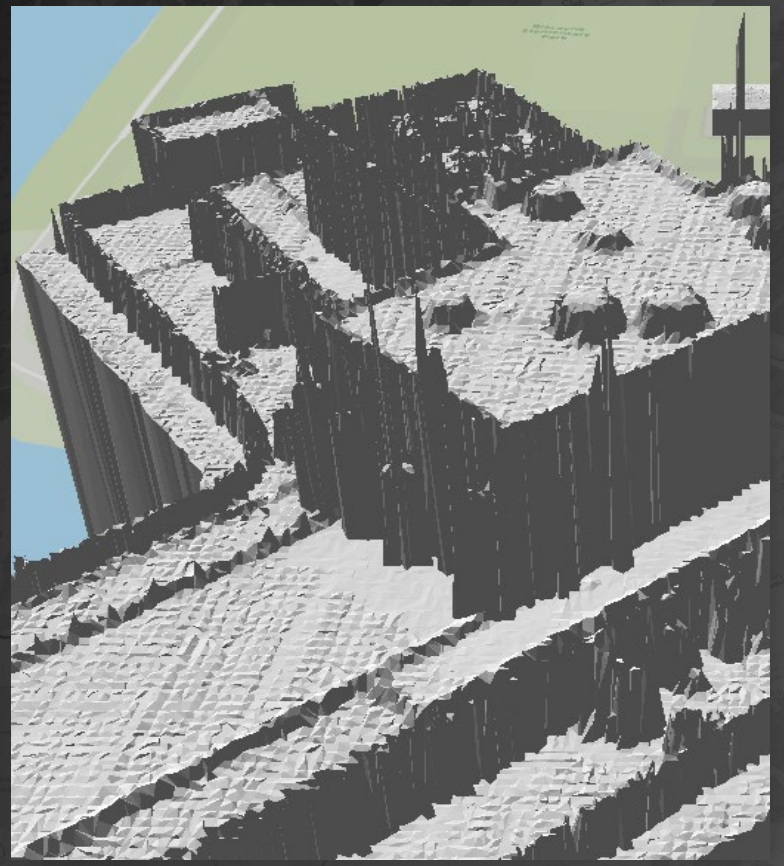
3. **LASPointStatisticsAsRaster**
   - with LAS layer filtered on class 6 (building)
   - using the 'Most Frequent Class Code' option
4. **RasterToPolygon**
   - Turn off the Simplify polygons option
5. **EliminatePolygonPart** to remove small holes (could alternately have performed some manipulation on the raster side for this)
6. **RegularizeBuildingFootprint** to straighten things out.

To extract shells:

7. **LASDatasetToRaster** with input LAS layer filtered on class 2 points to make DEM
8. **LASBuildingMultipatch**

Working in true 3D point cloud

- Resulting models are not suitable for manual editing because of largen ## of vertices
- Important to keep the noise level down

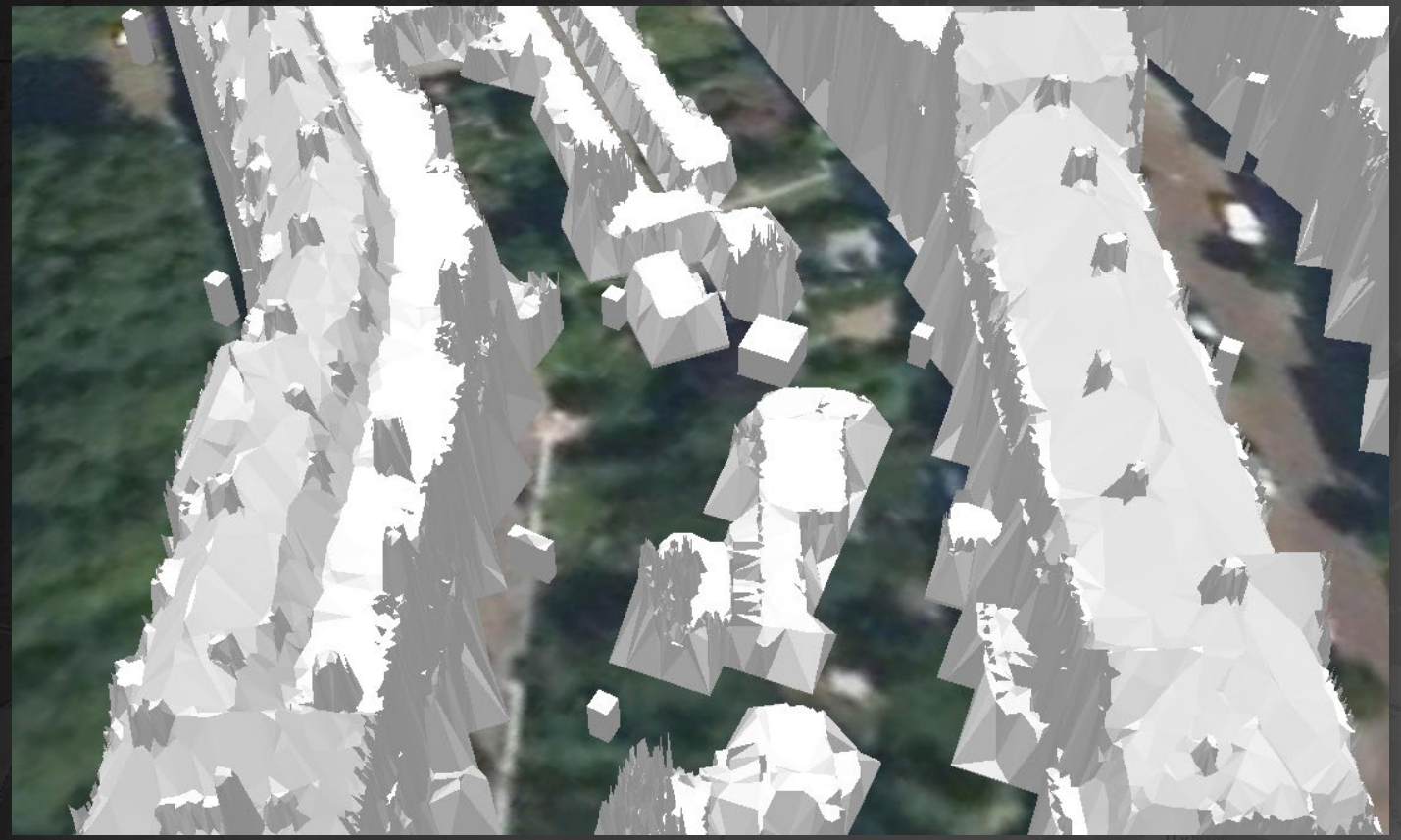


Working in true 3D point cloud

Main sources of noise in resulting models come from misclassification of

- Ground
- Buildings

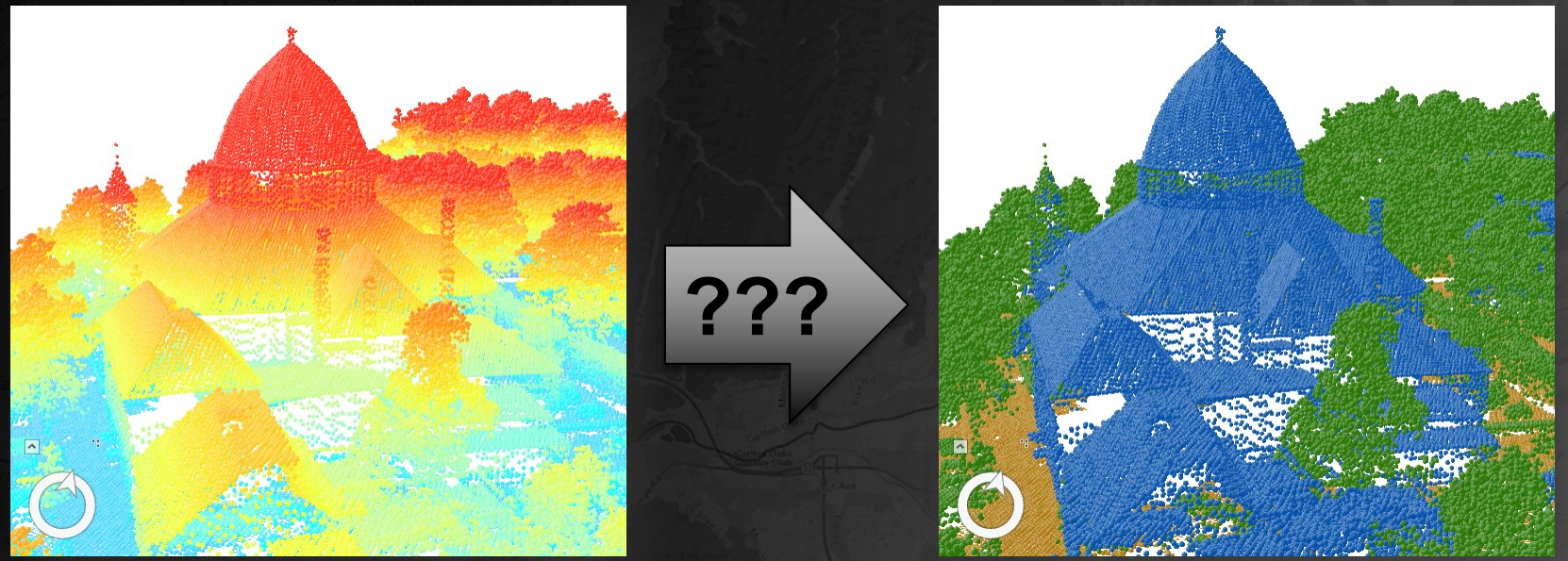
- **ClassifyLASBuilding** does not produce reliable results when there is a vegetation in direct proximity to the building



Can we use DL to classify Point Clouds?

- Point clouds are unordered, sparce data collections where direct application of convolutional kernels does not work efficiently.

- Good news in recent years: PointNet, PointNet++, Graph Convolutional networks, Deep Sets, PointCNN, etc.



PointCNN

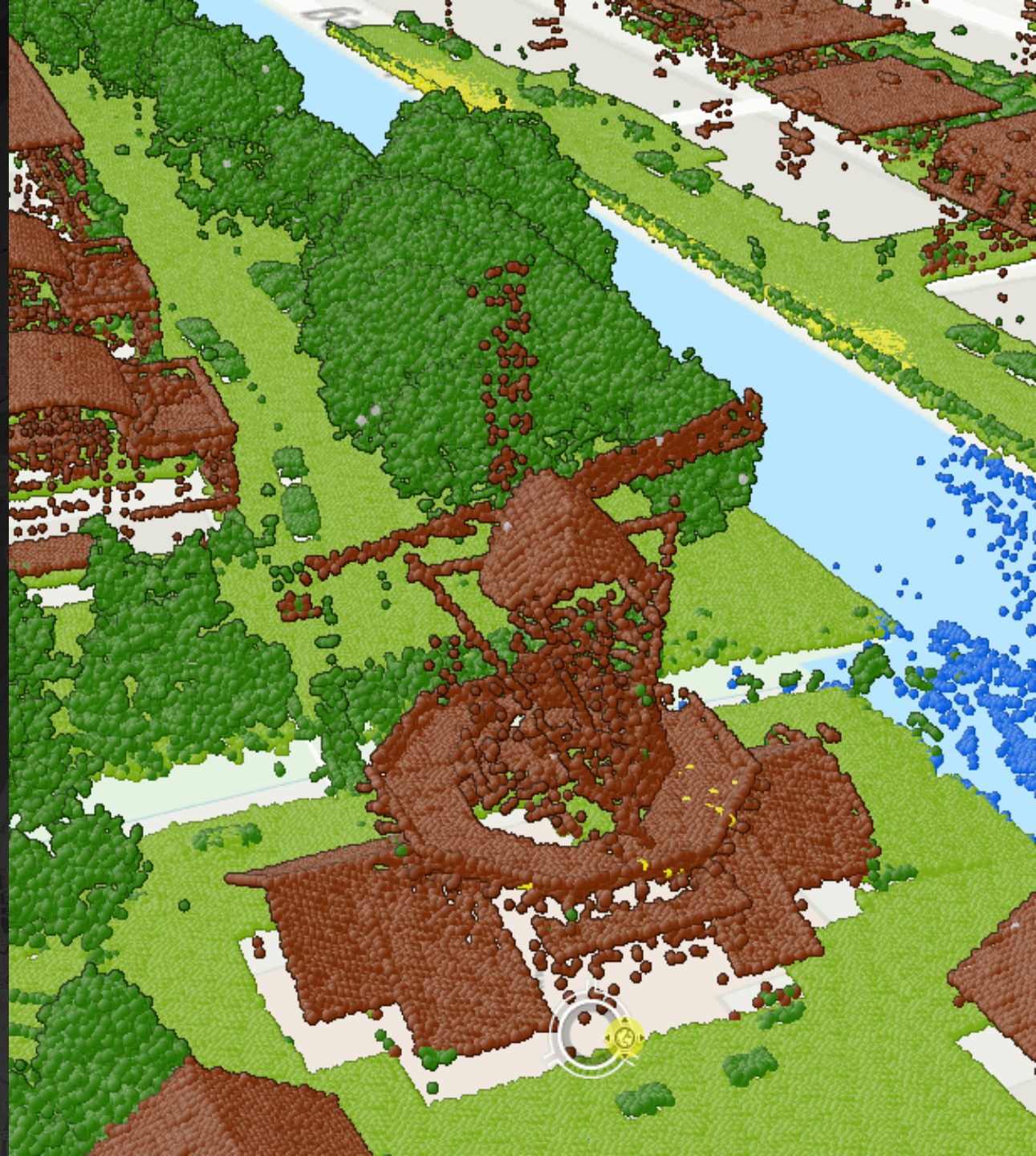
- Trained on 1.8B X-Y-Z points of airborne point cloud collected over Amsterdam, Netherlands.
- 0.97 accuracy just after 6.5h of training on Nvidia QUADRO V100.
- Testing on city of Utrecht, Netherlands

- Classify Building Points

- PointCNN

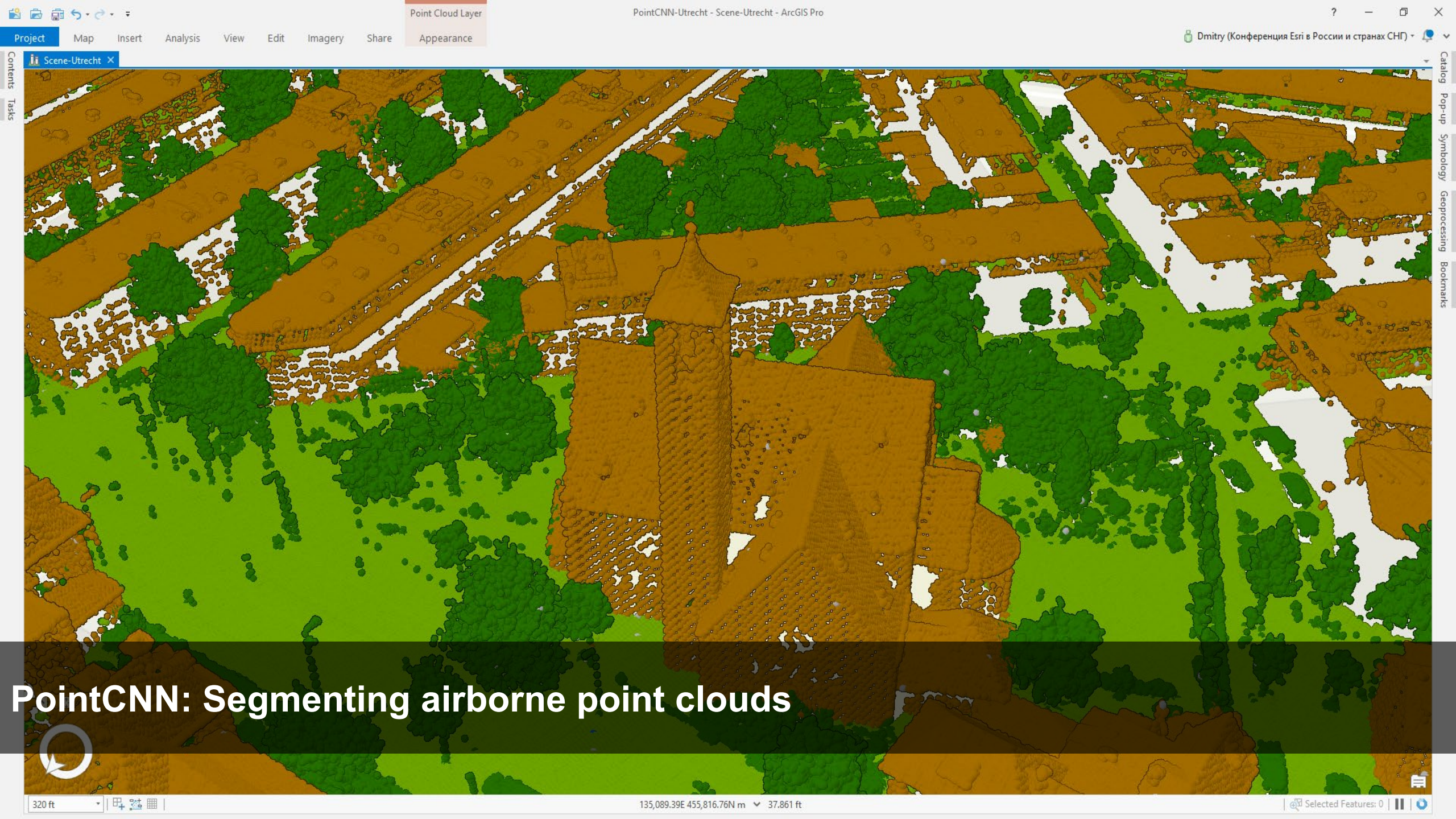
PointCNN

- Trained on 1.8B X-Y-Z points of airborne point cloud collected over Amsterdam, Netherlands.
- 0.97 accuracy just after 6.5h of training on Nvidia QUADRO V100.
- Testing on city of Utrecht, Netherlands



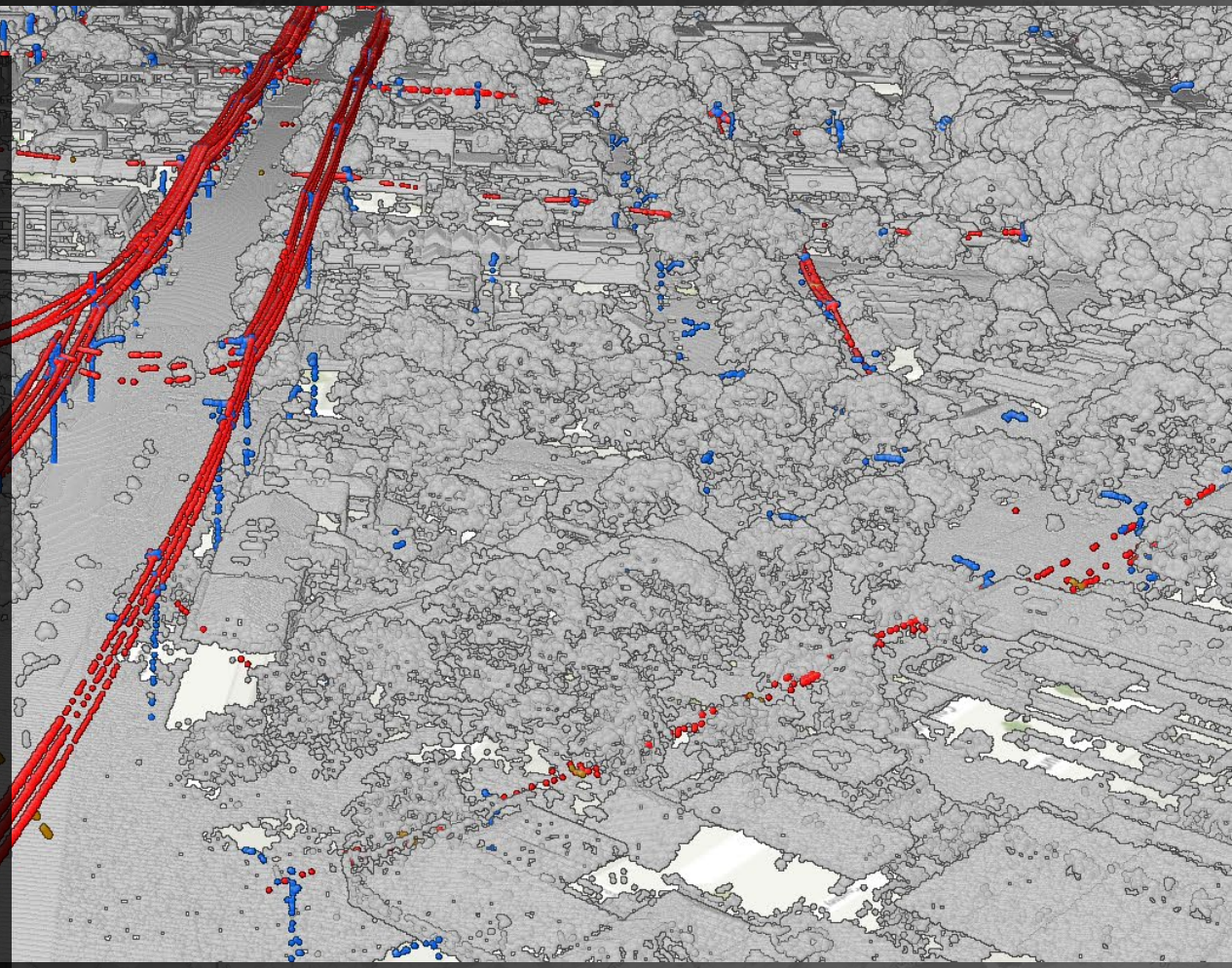
PointCNN
vs
ClassifyBuildingPoints

- Significantly lower noise level in the models reconstructed from point cloud classified by a PointCNN model.


Point Cloud Layer
PointCNN-Utrecht - Scene-Utrecht - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Scene-Utrecht
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
PointCNN: Segmenting airborne point clouds
320 ft
135,089.39E 455,816.76N m
37.861 ft
Selected Features: 0

PointCNN: Wires & Poles

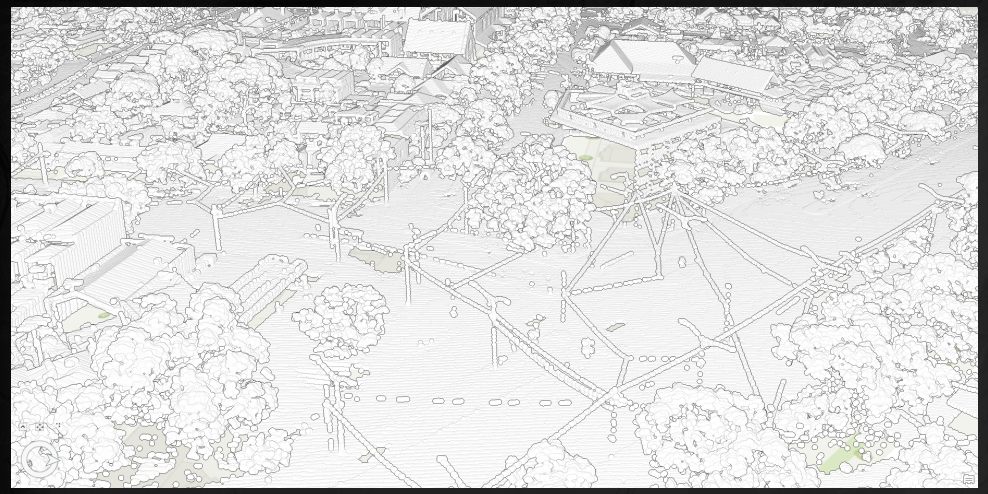
from airborne LiDAR



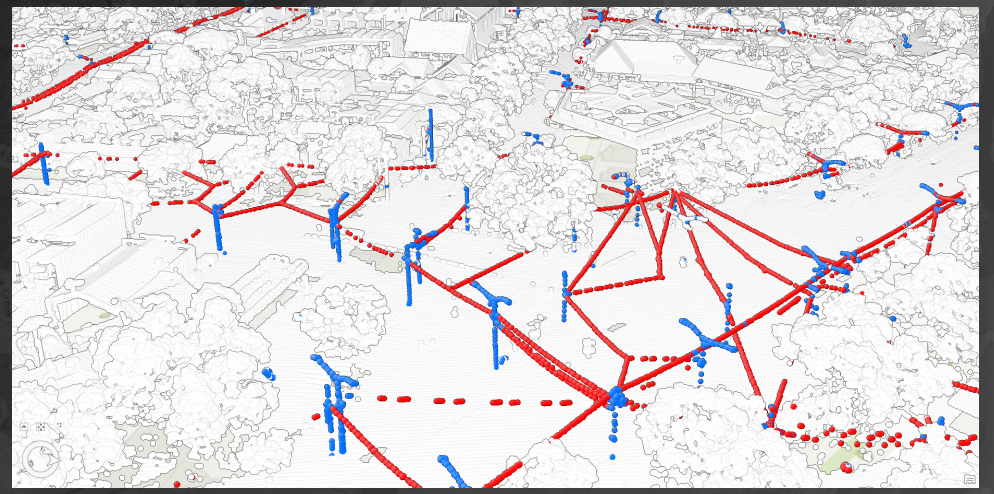
esri | THE SCIENCE OF WHERE

PoC: Wires detection in airborne LiDAR

AAM Group, Australia: collecting airborne LiDAR point clouds to detect the power lines and any easement encroachment.



50,000 man hours per year of manual labeling

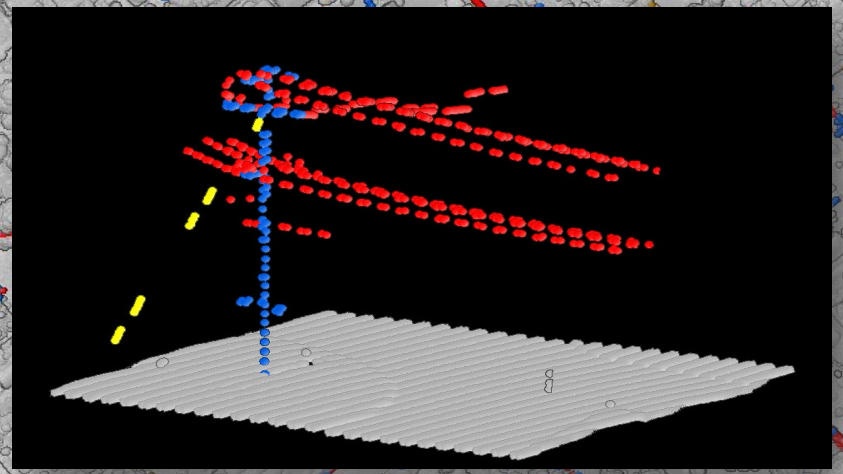


PoC: Wires detection in airborne LiDAR

- AAM Group shared 10B+ of manually labeled points to train a neural network.
- Thousands of miles of transmission lines.

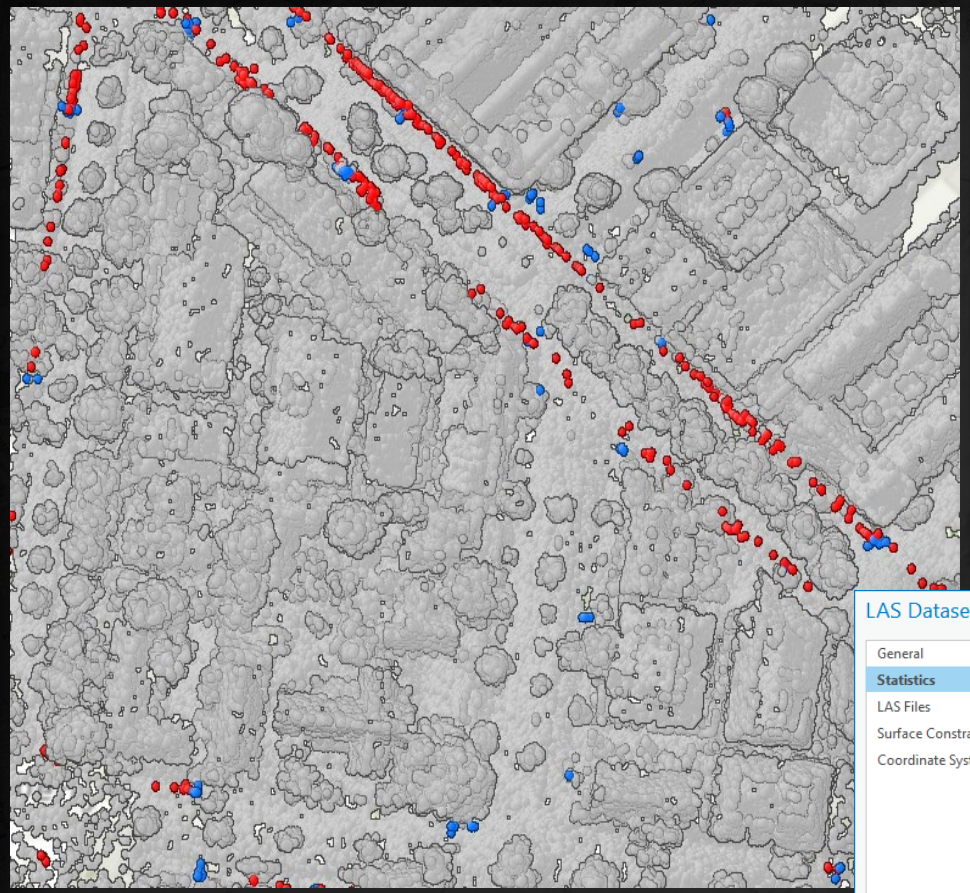
Four object classes:

1. Wire
2. Stay Wire
3. Pole
4. Other



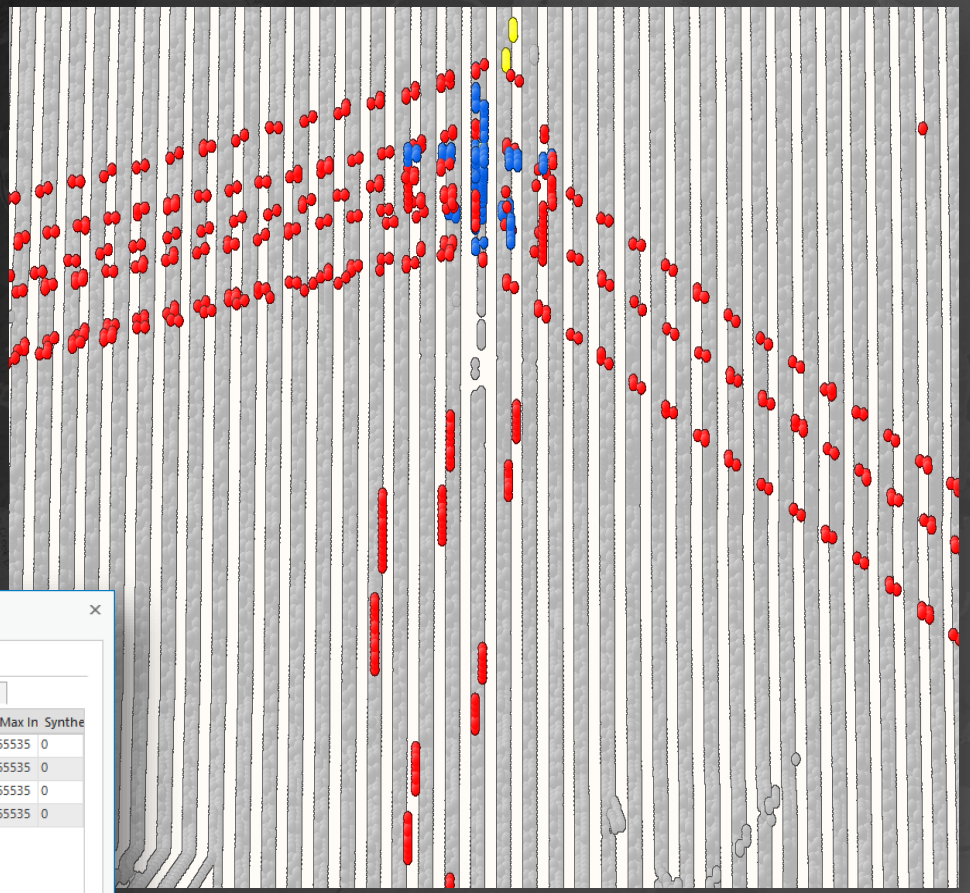
PoC: Wires detection in airborne LiDAR

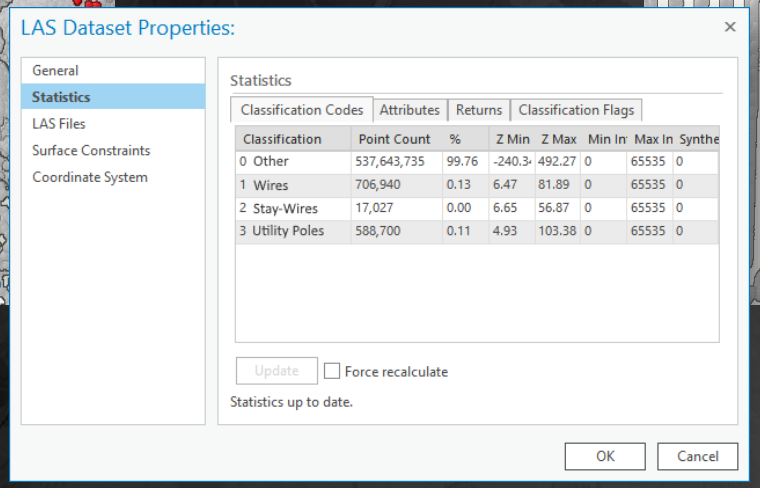
With wires, there challenges though -



Significant class imbalance:
99.76% points are of the Other class.

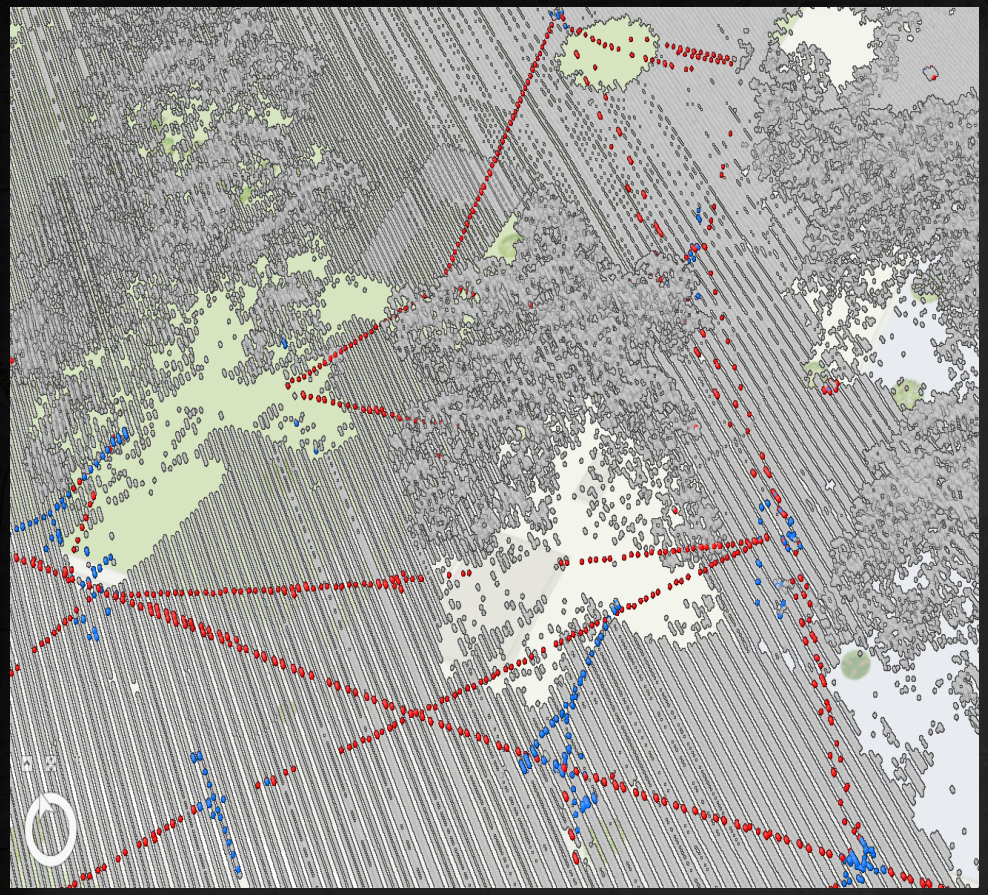
Scan lines angle causes the Wire point density fluctuation



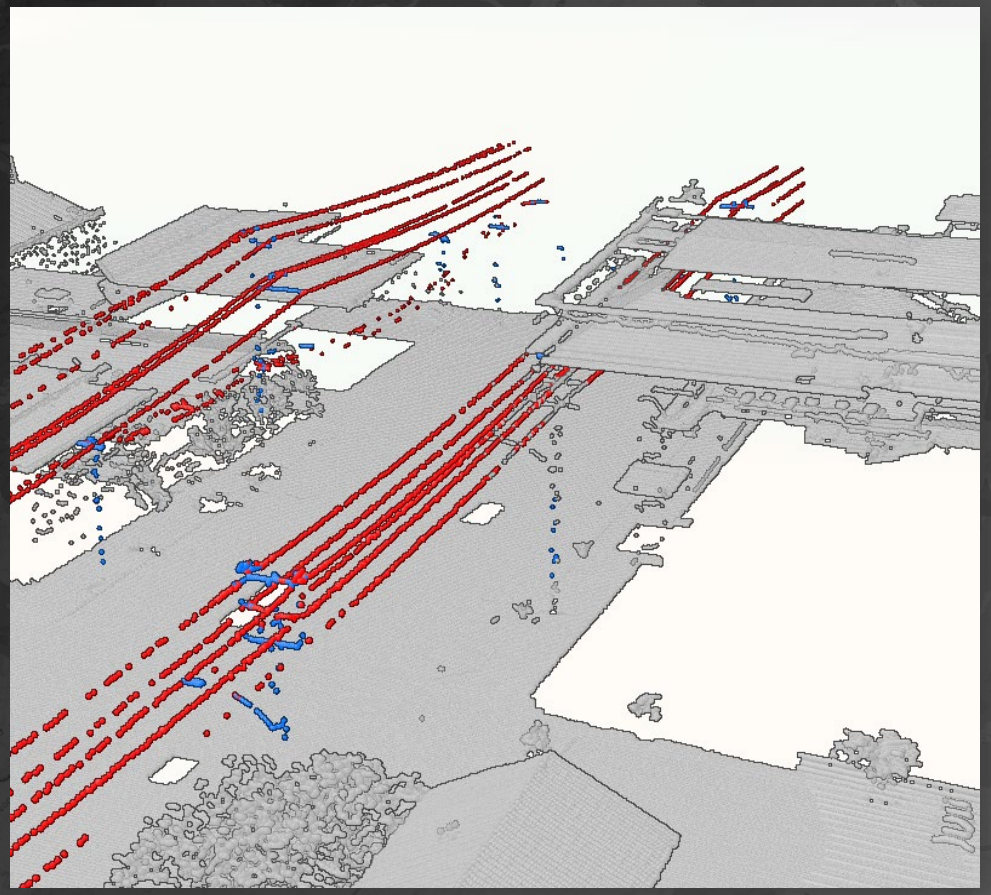


PoC: Overhead conductor inspection

With wires, there challenges though -



Wires are zero-area point neighborhoods which easily blend into tree canopies and buildings

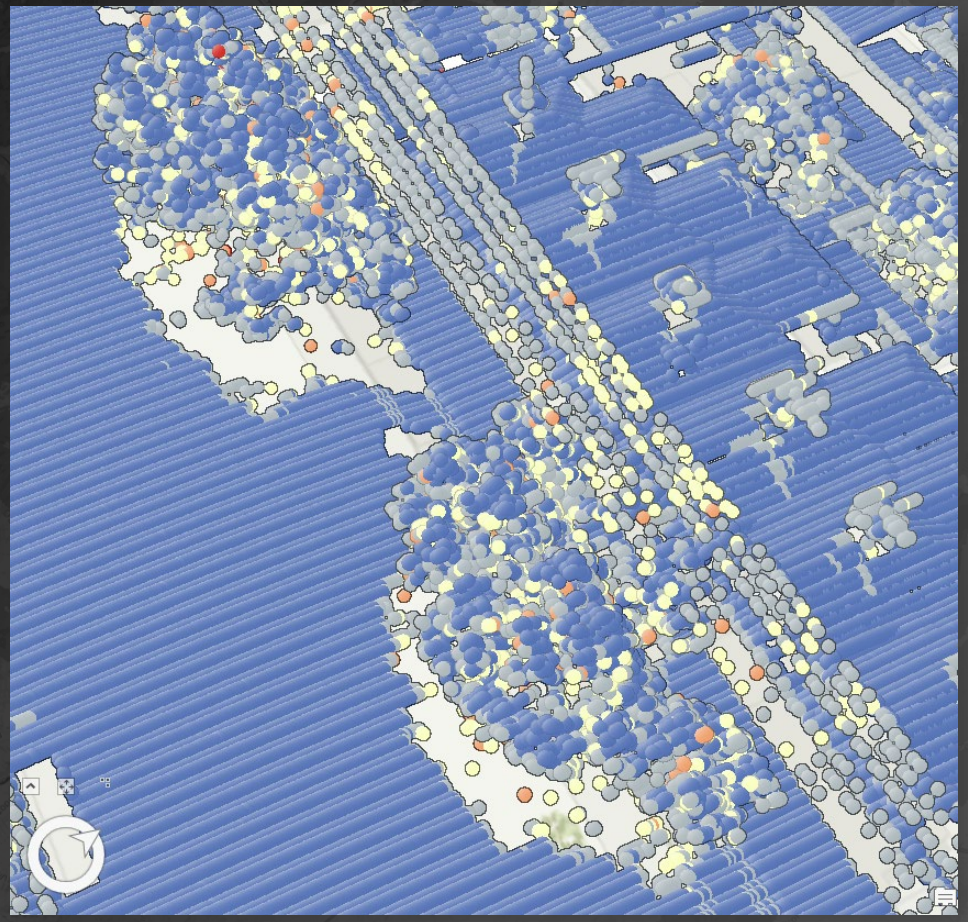


PoC: Overhead conductor inspection

…but there are some good news too -

Intensity and the Number of Returns on the Wire points is often different than of the surroundings

This allows for training PointCNN on
XYZ +
Intensity +
Number of Returns

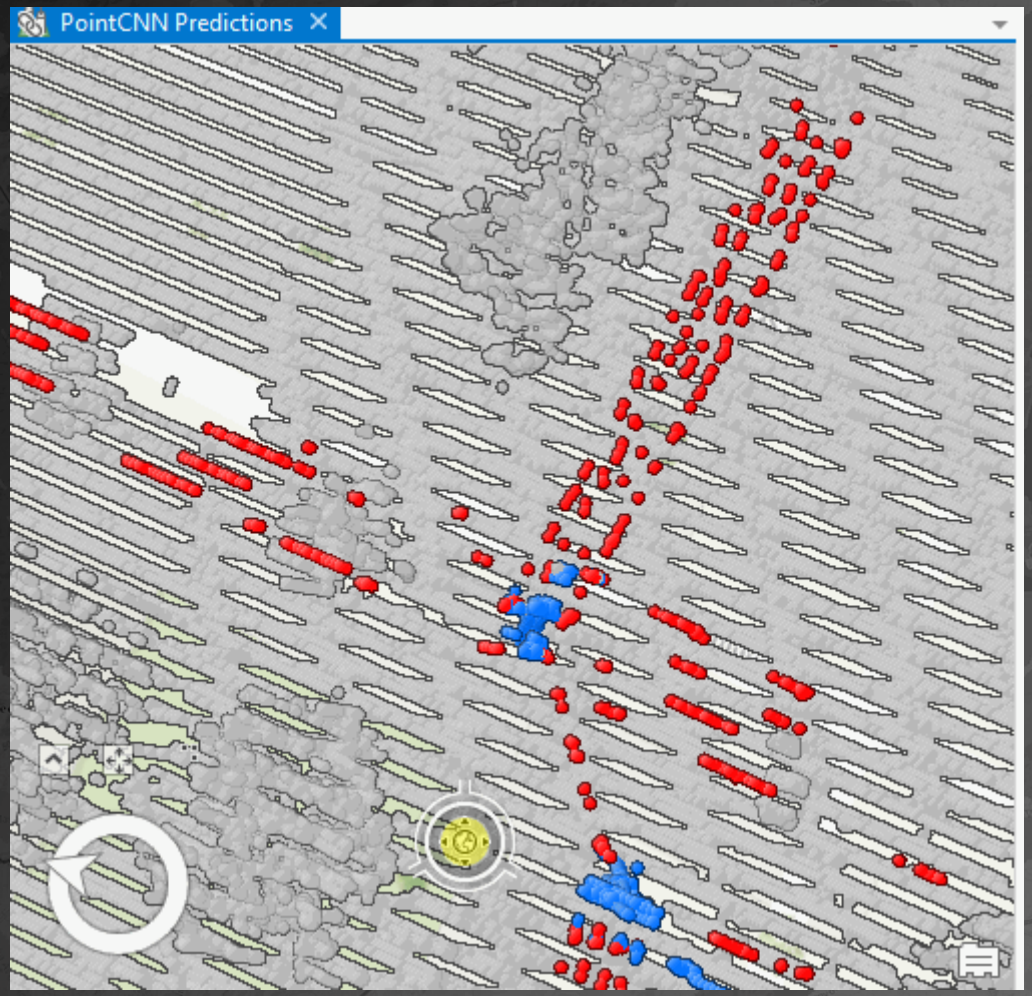


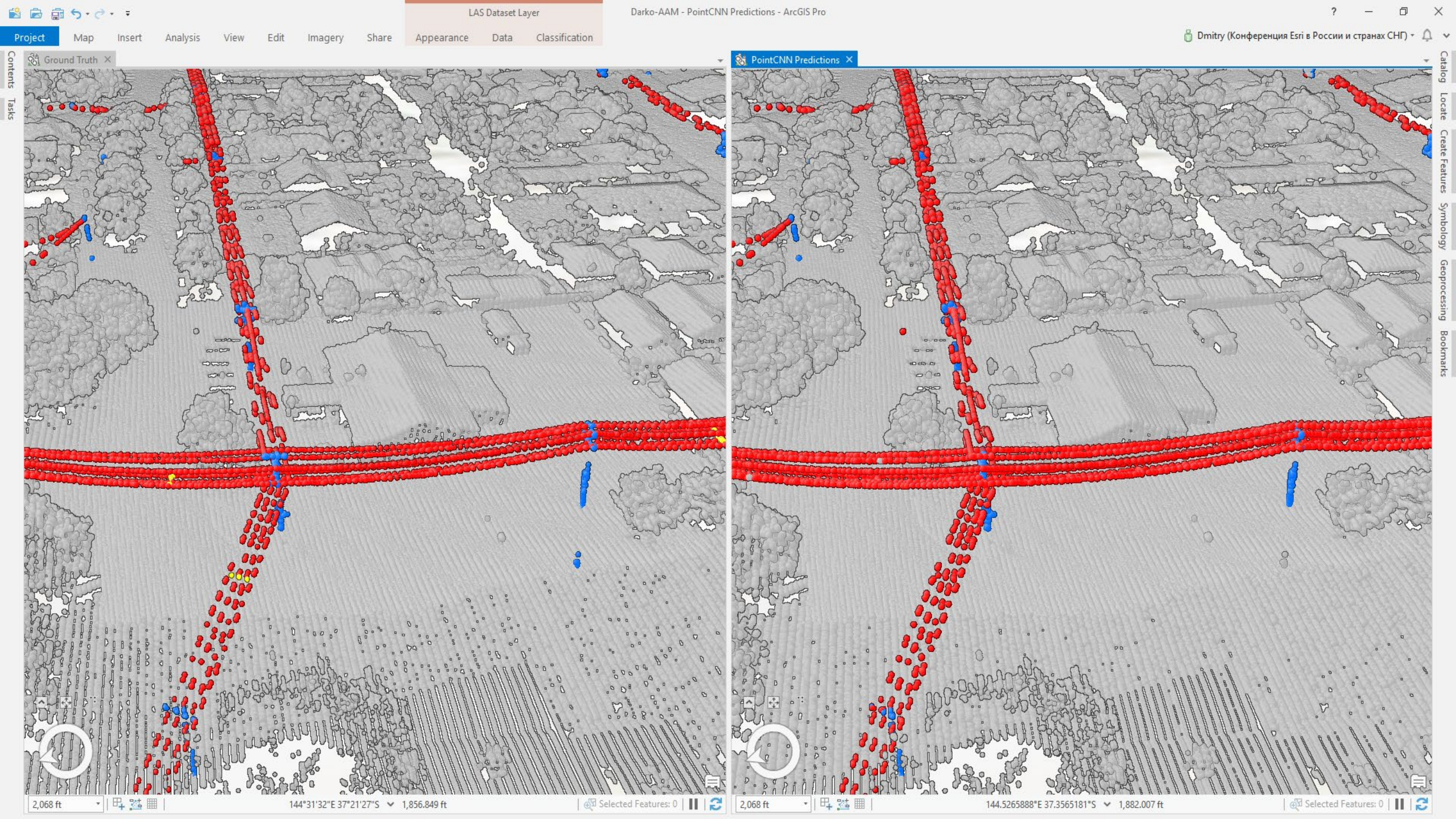
PoC: Overhead conductor inspection

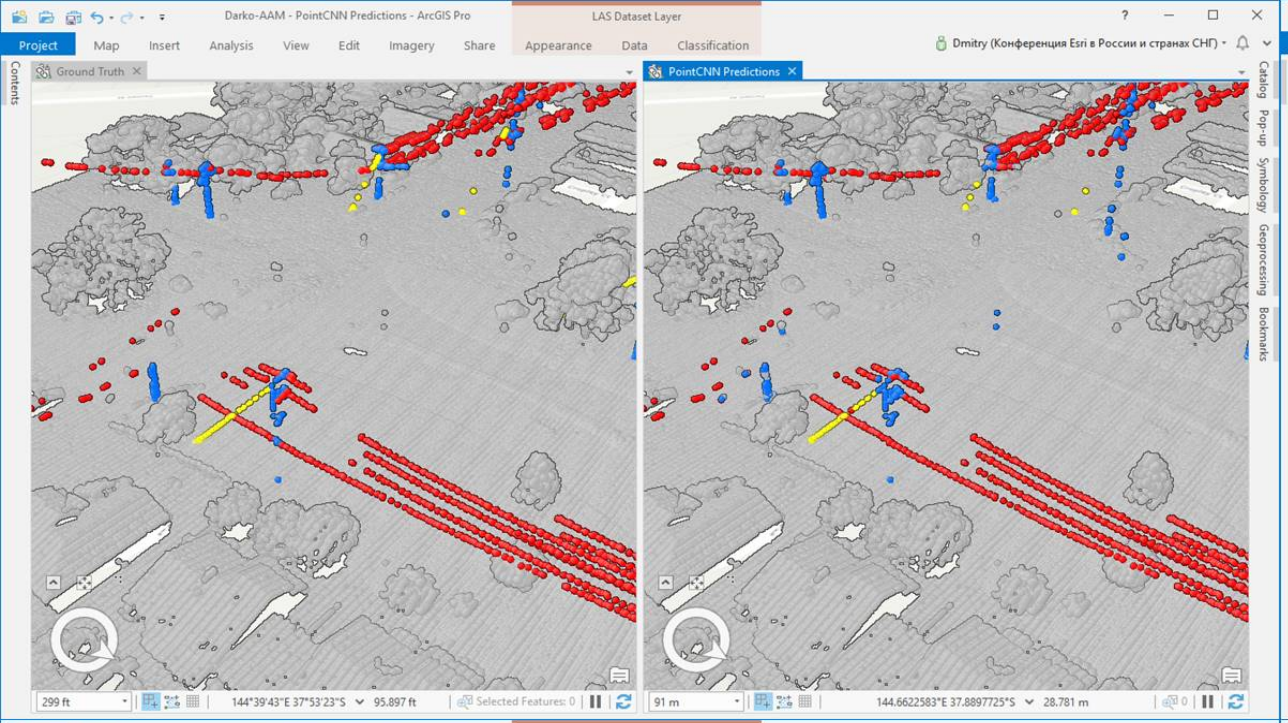
Training & Results:

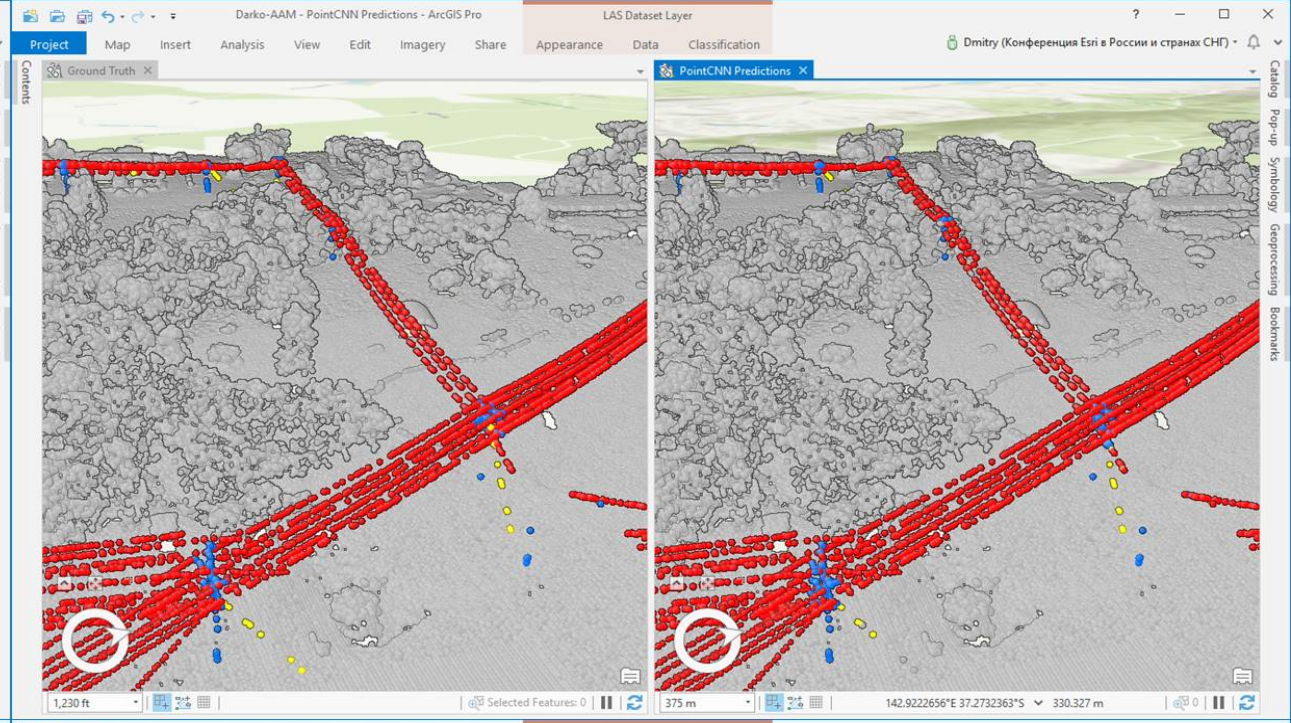
after training on a single GV100 for ~20 hours

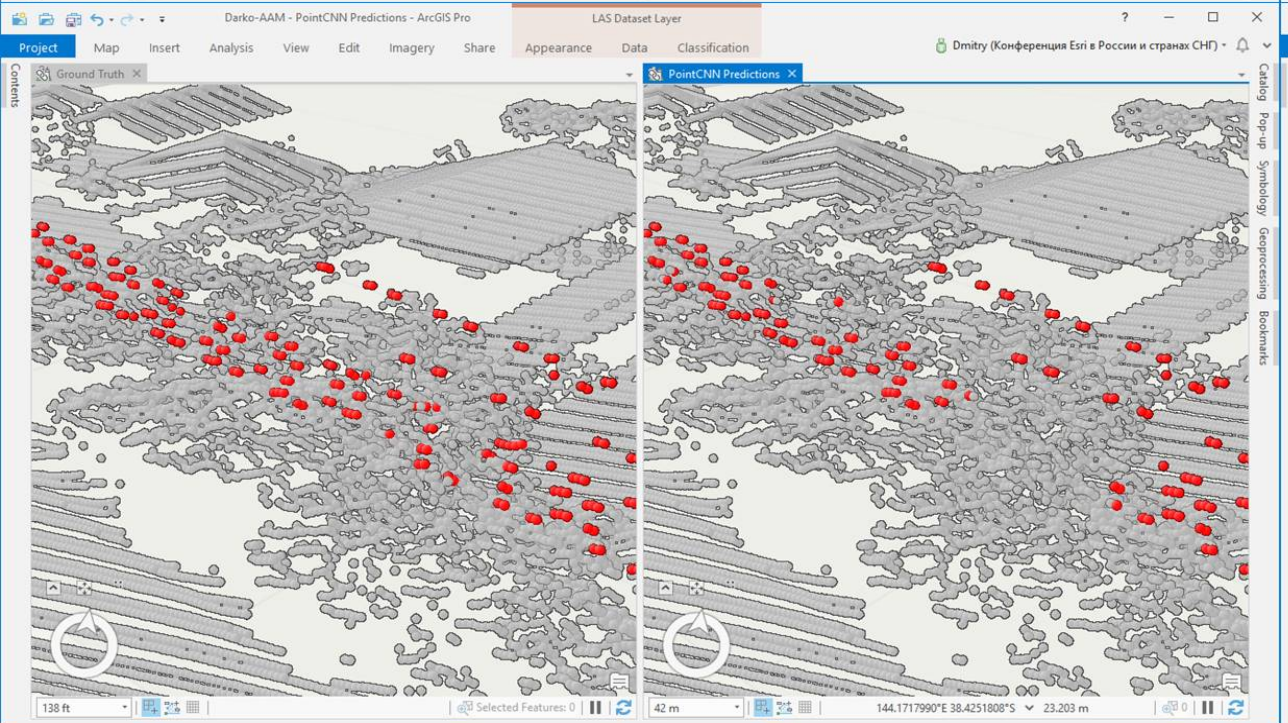
```
@E232K best RECALL
            OTHER       WIRE        STAY-WIRE   POLE
Precision:[0.99988538 0.96672749 0.83674406 0.80313546],
Recall:   [0.99987221 0.98060249 0.21455632 0.77497643],
F1 score: [0.9998788  0.97361556 0.3415365  0.78880471]
```

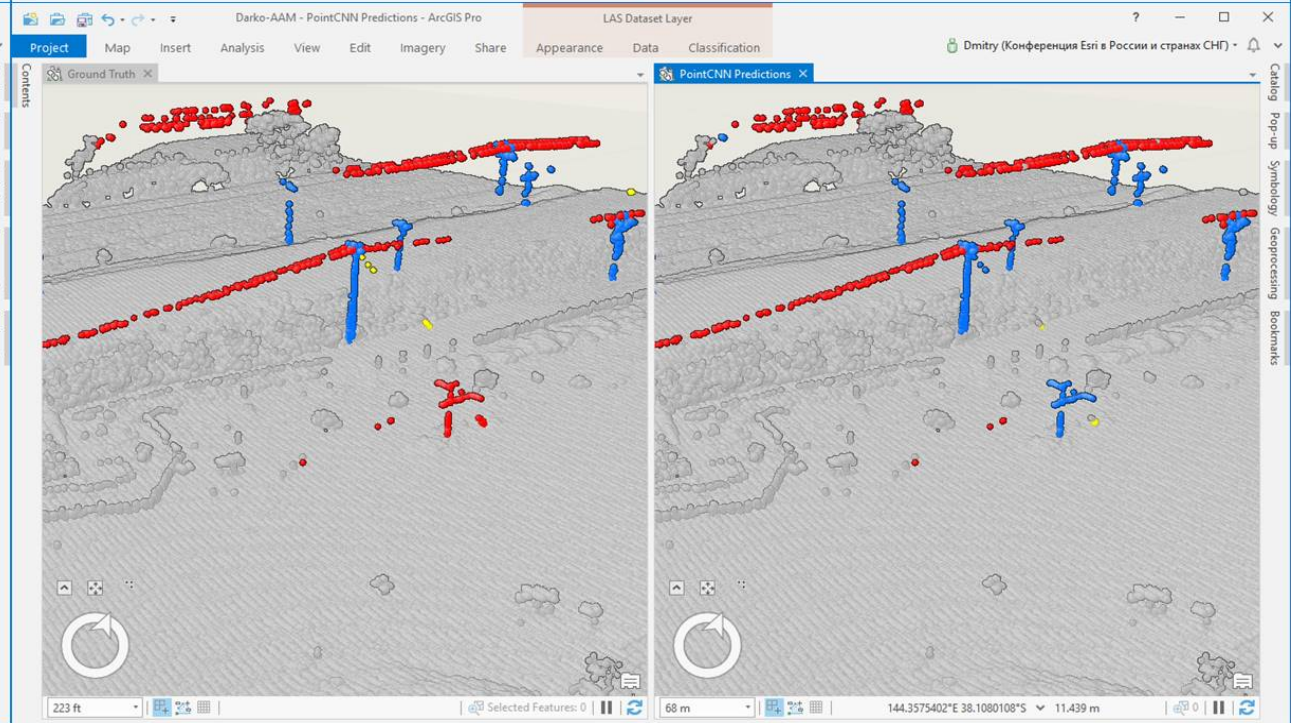


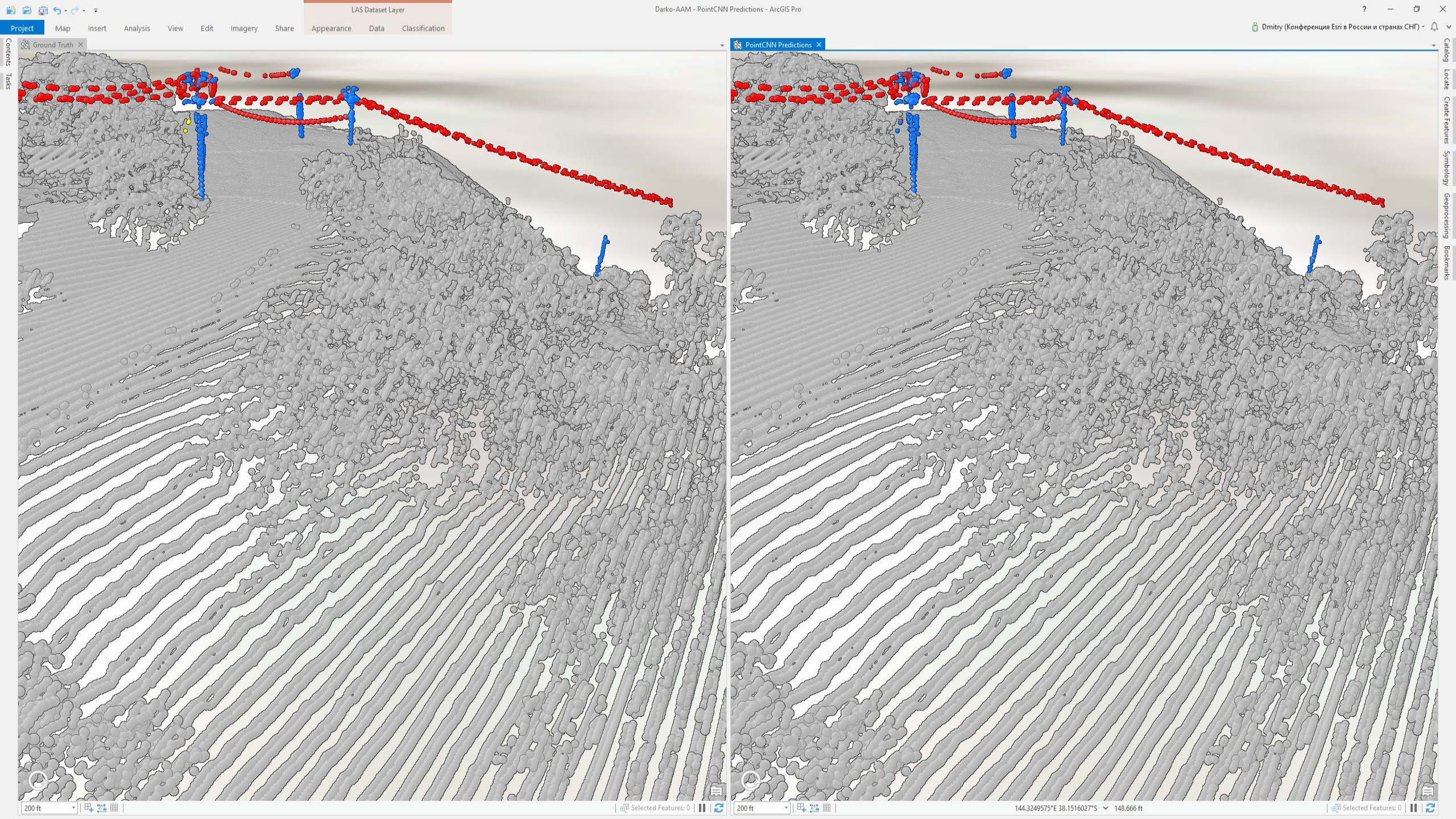

LAS Dataset Layer
Darko-AAM - PointCNN Predictions - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Contents
Tasks
Ground Truth
PointCNN Predictions
Catalog
Locate
Create Features
Symbology
Geoprocessing
Bookmarks
2,068 ft
144°31'32"E 37°21'27"S
1,856.849 ft
Selected Features: 0
2,068 ft
144.5265888°E 37.3565181°S
1,882.007 ft
Selected Features: 0

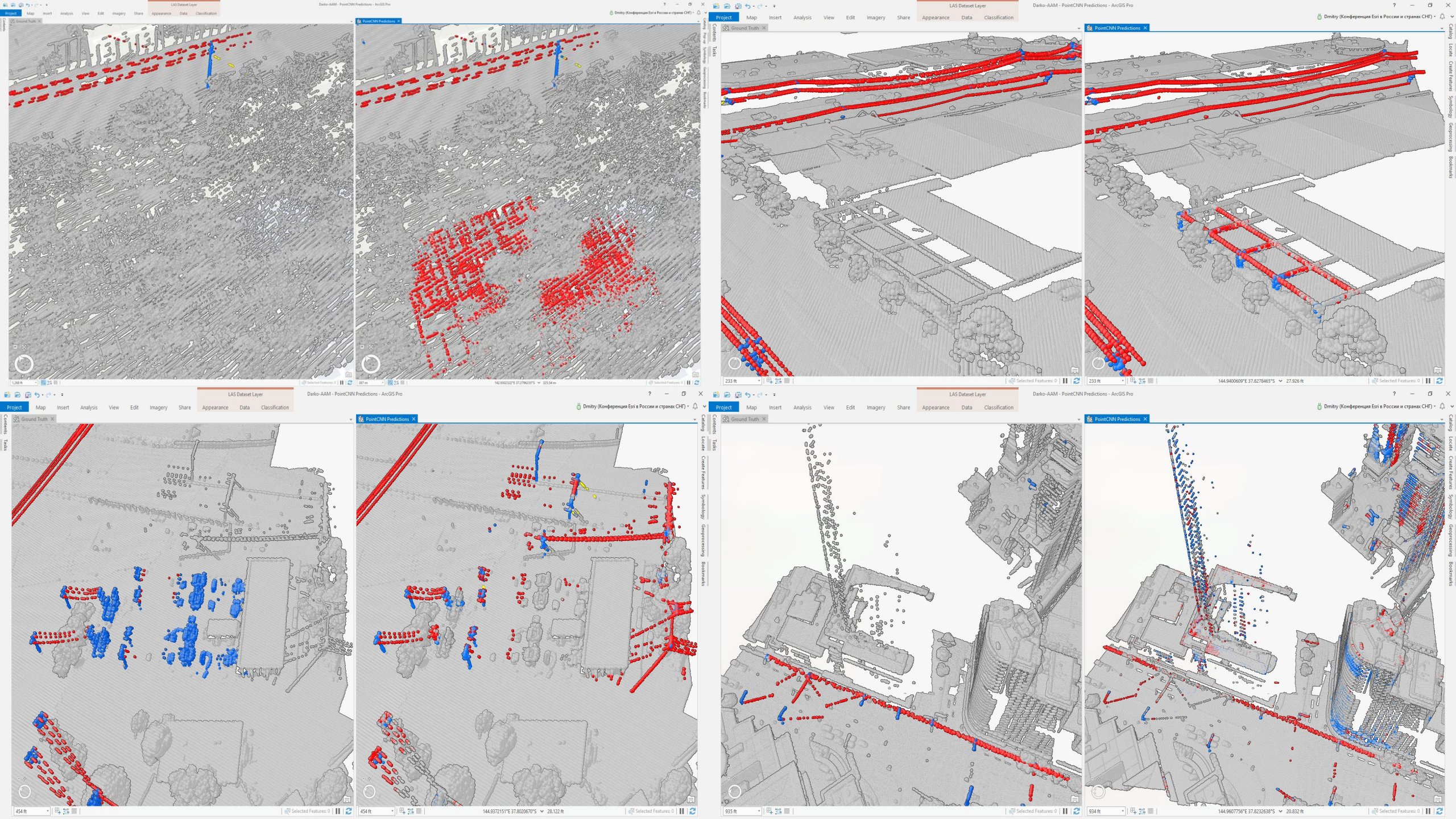

Darko-AAM - PointCNN Predictions - ArcGIS Pro
LAS Dataset Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Ground Truth
PointCNN Predictions
Contents
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
299 ft
144°39'43"E 37°53'23"S
95.897 ft
Selected Features: 0
91 m
144.6622583°E 37.8897725°S
28.781 m

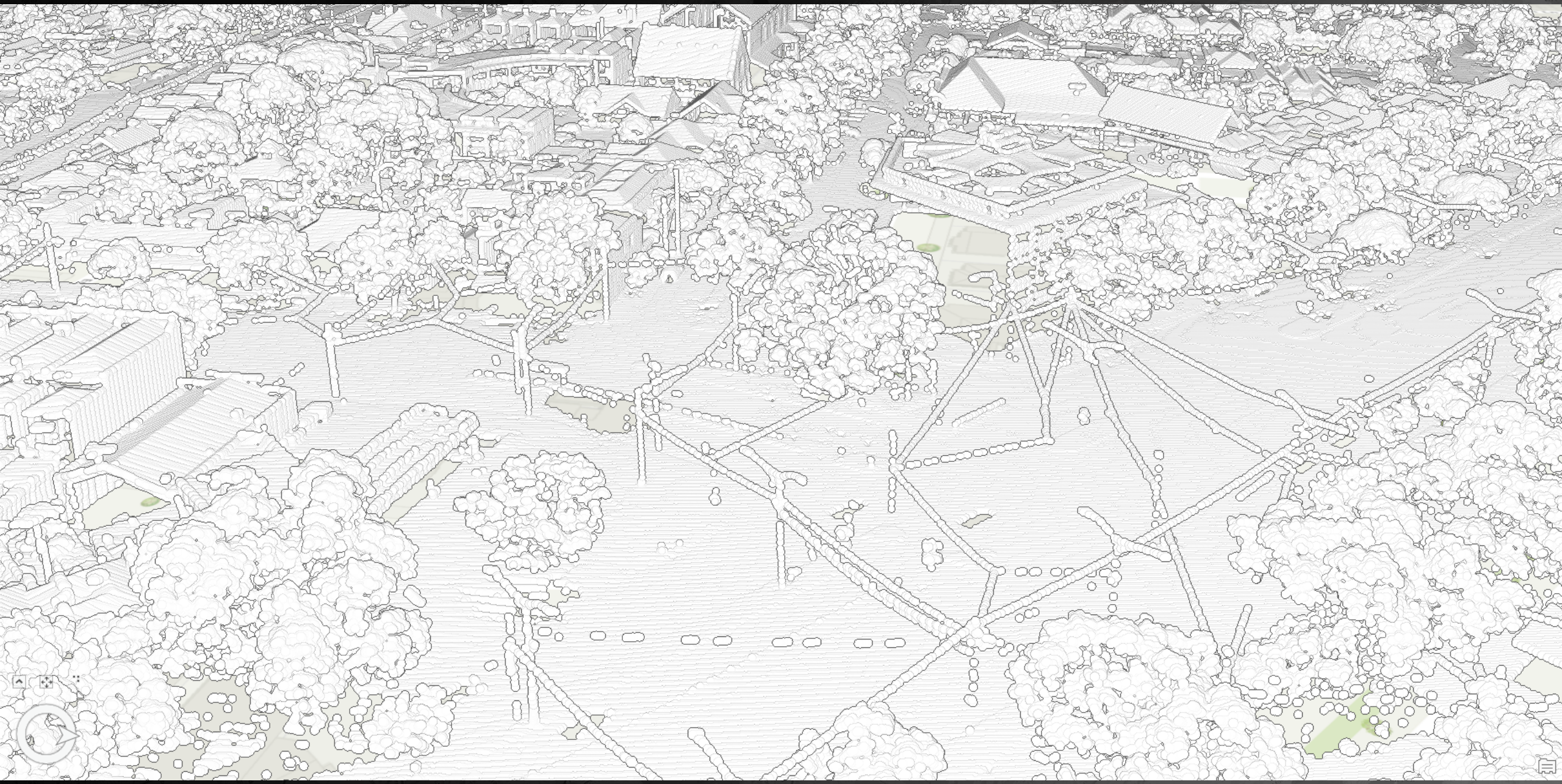

Darko-AAM - PointCNN Predictions - ArcGIS Pro
LAS Dataset Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Ground Truth
PointCNN Predictions
Contents
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
1,230 ft
Selected Features: 0
375 m
142.9222656°E 37.2732363°S
330.327 m

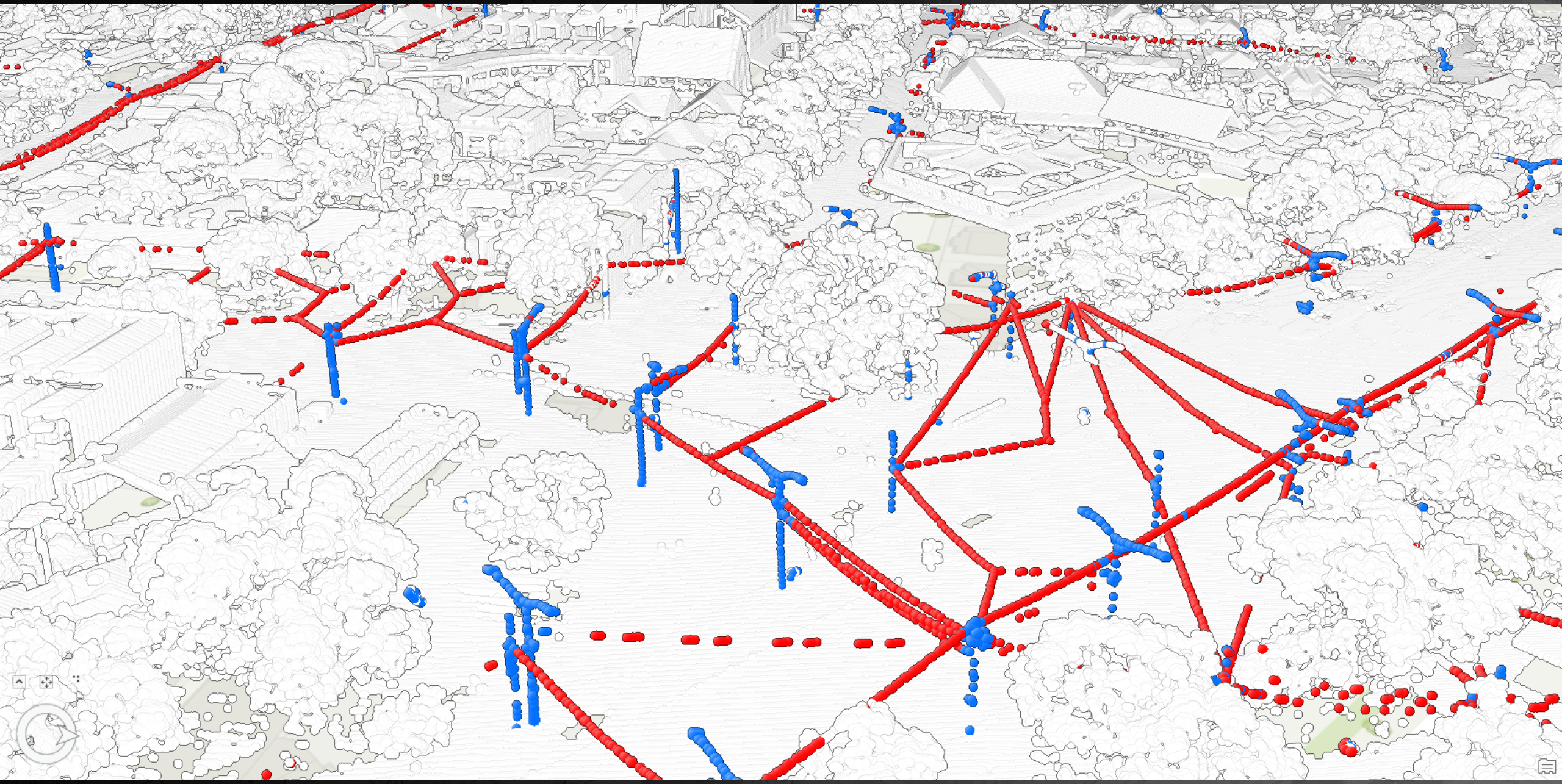

Darko-AAM - PointCNN Predictions - ArcGIS Pro
LAS Dataset Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Ground Truth
PointCNN Predictions
Contents
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
138 ft
Selected Features: 0
42 m
144.1717990°E 38.4251808°S
23.203 m

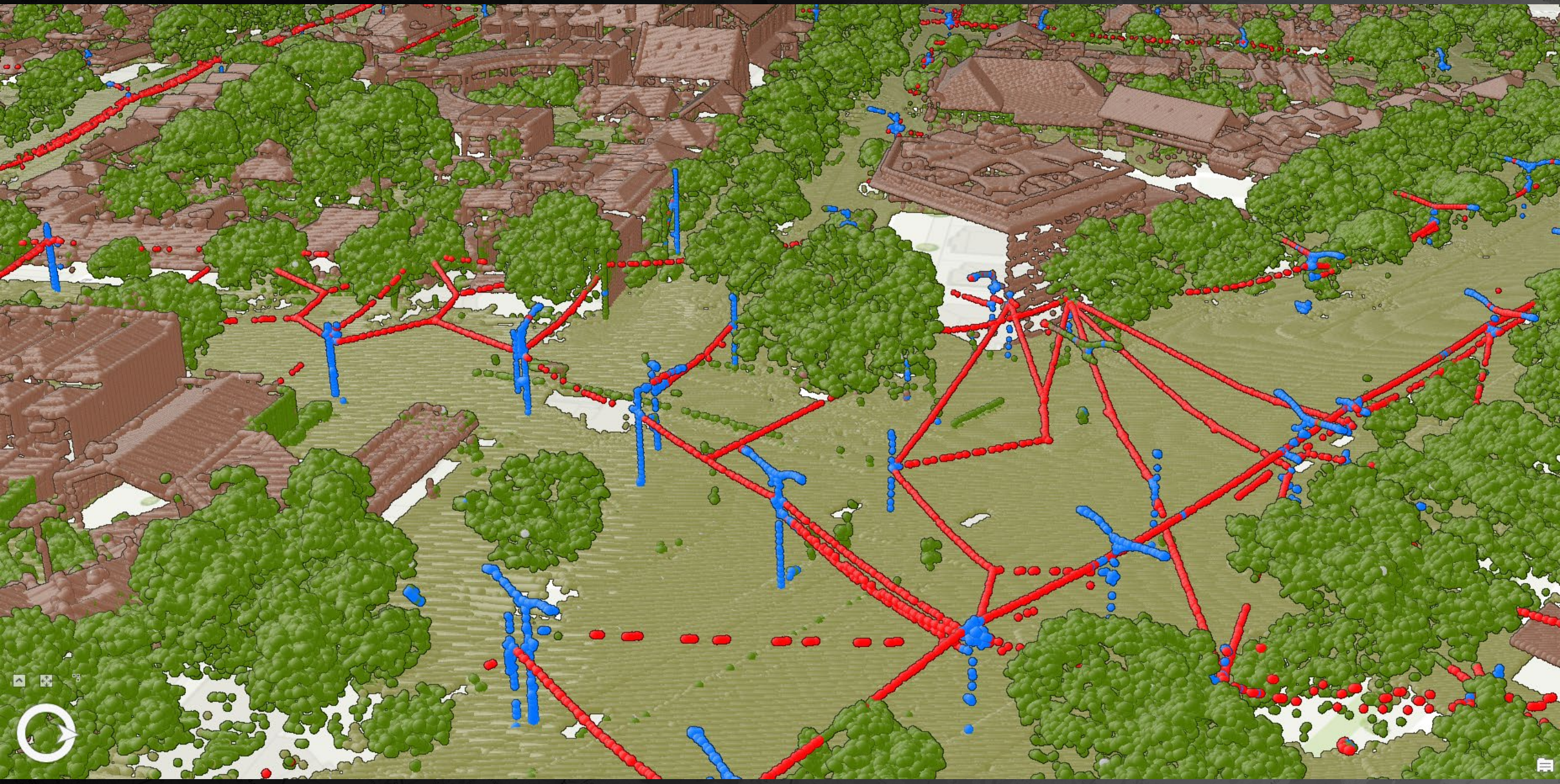

Darko-AAM - PointCNN Predictions - ArcGIS Pro
LAS Dataset Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Ground Truth
PointCNN Predictions
Contents
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
223 ft
Selected Features: 0
68 m
144.3575402°E 38.1080108°S
11.439 m

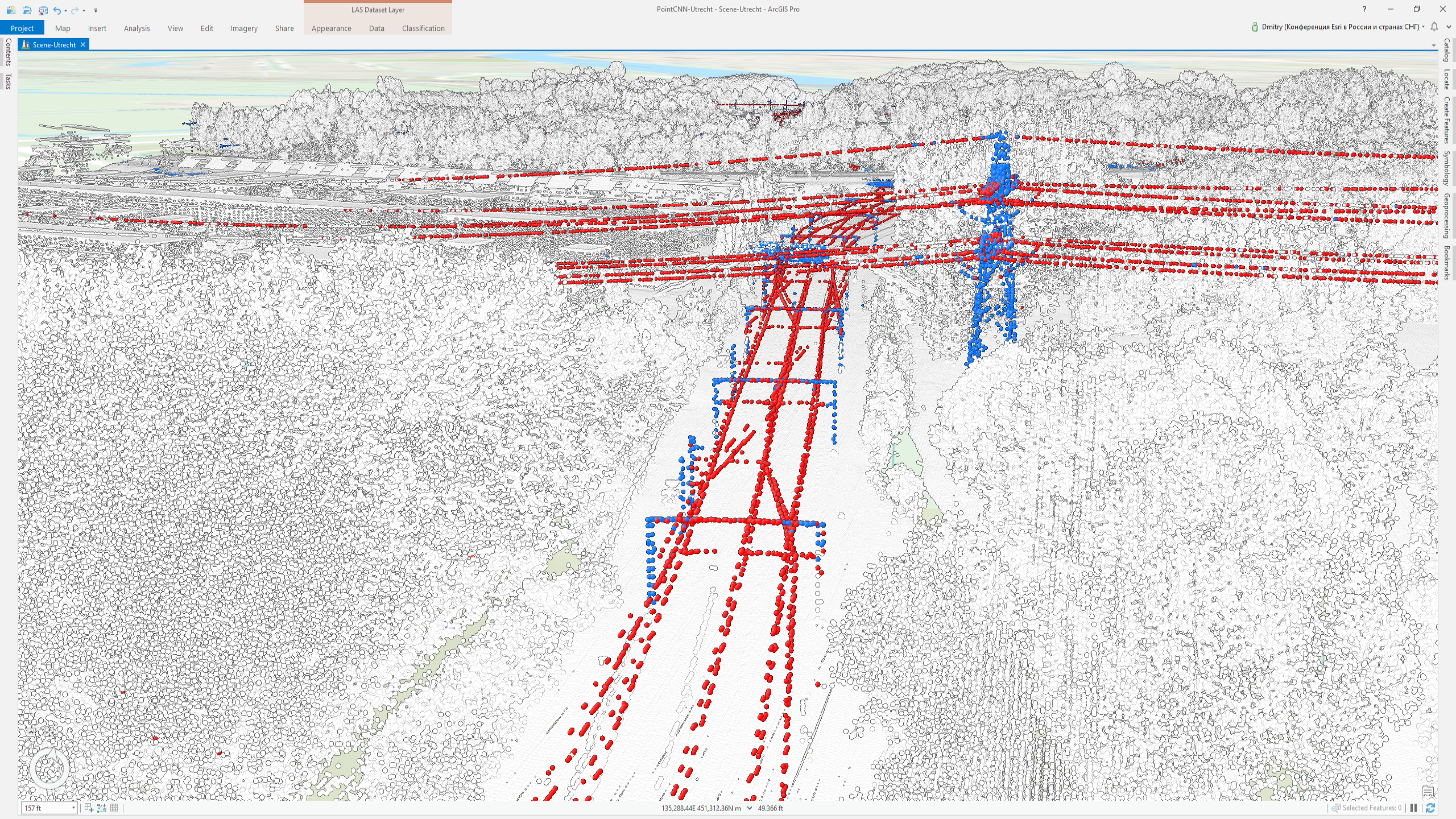

LAS Dataset Layer
Darko-AAM - PointCNN Predictions - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Contents
Tasks
Ground Truth
PointCNN Predictions
Catalog
Locate
Create Features
Symbology
Geoprocessing
Bookmarks
200 ft
Selected Features: 0
200 ft
144.3249575°E 38.1516027°S
148.666 ft
Selected Features: 0

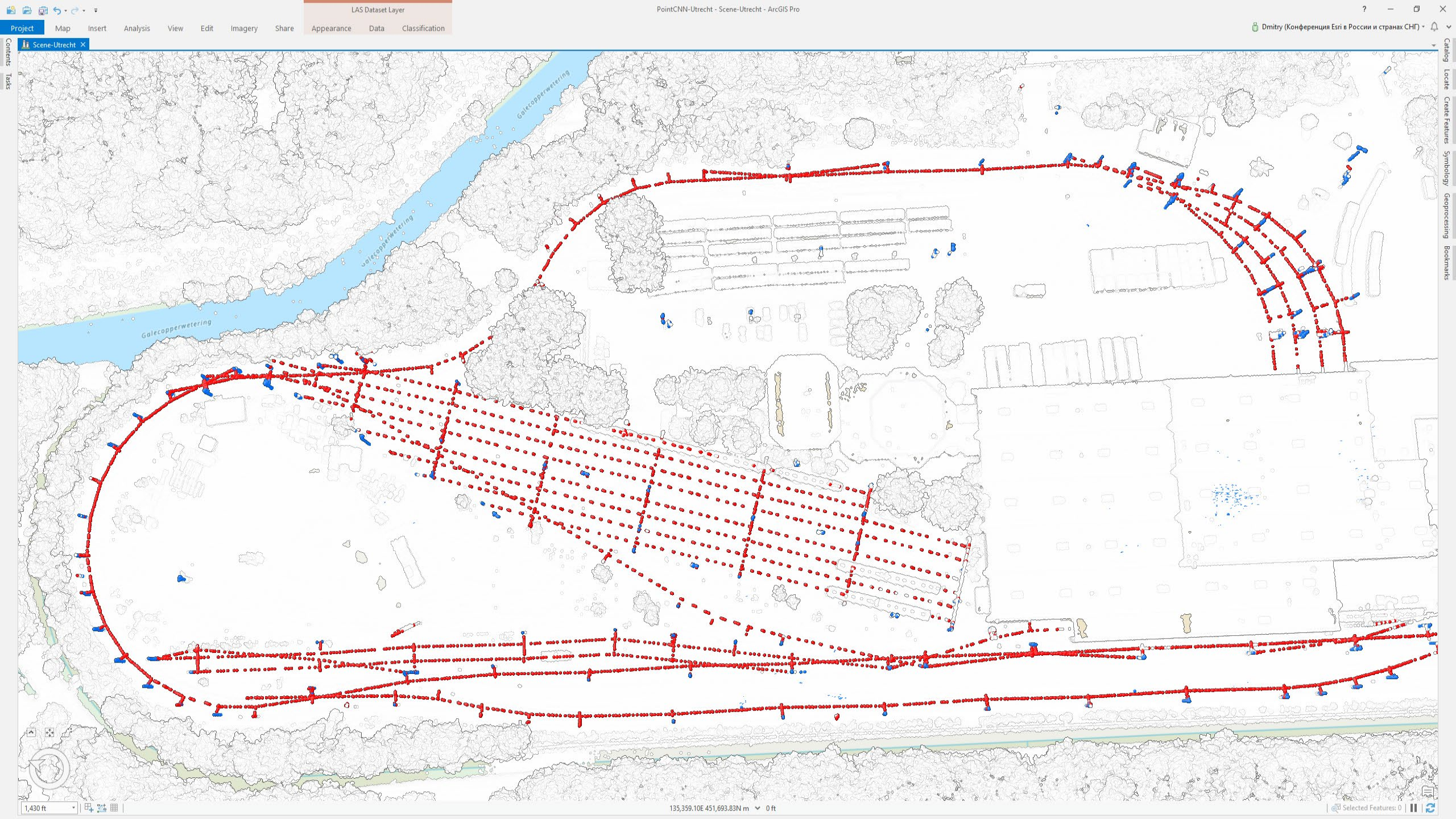

Darko-AAM - PointCNN Predictions - ArcGIS Pro
LAS Dataset Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Ground Truth
PointCNN Predictions
Dmitry (Конференция Esri в России и странах СНГ)
Contents
Tasks
Catalog
Locate
Create Features
Symbology
Geoprocessing
Bookmarks
Selected Features: 0
144.9400609°E 37.8278465°S
27.926 ft
144.9371511°E 37.8020670°S
28.122 ft
144.9607756°E 37.8232638°S
20.832 ft








LAS Dataset Layer
PointCNN-Utrecht - Scene-Utrecht - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Scene-Utrecht
Contents
Tasks
Catalog
Locate
Create Features
Symbology
Geoprocessing
Bookmarks
157 ft
135,288.44E 451,312.36N m
49.366 ft
Selected Features: 0


LAS Dataset Layer
PointCNN-Utrecht - Scene-Utrecht - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Scene-Utrecht
Contents
Tasks
Catalog
Locate
Create Features
Symbology
Geoprocessing
Bookmarks
Galecopperwetering
Galecopperwetering
Galecopperwetering
1,430 ft
135,359.10E 451,693.83N m
0 ft
Selected Features: 0

Railway Asset Detection

David Yu

esri | THE SCIENCE OF WHERE

Automating Rail Asset Detection

Using Deep Learning x GIS

David Yu

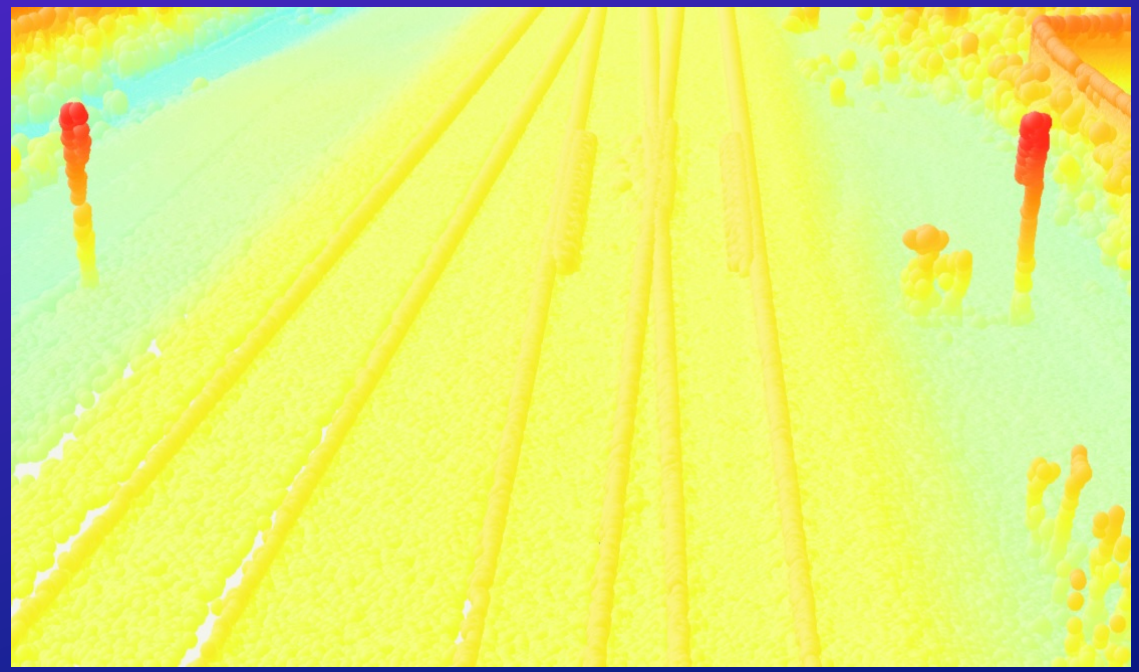
Challenge / Opportunity / Value

Challenge:

- Auditing, inventorying, managing change – at scale
- Asset collection & change detection

Value:

- PTC compliance
- Decreased cost
- Increased Accuracy
- Increased Frequency



Machine Learning / Input / Outcome

Conventional approach:

- Convert to image and apply CNN for feature extraction

Problem statement:

- Big data (e.g., over billion points)
- Train a deep learning model to detect objects directly from 3D point clouds?

Inputs:

- 3 subdivisions worth of labeled asset locations
  - Crossings, Mileposts, Switches, Signals
- 3D point clouds for study areas
- Track centerline

Outcomes:

- Geospatially enabled and trained deep learning model
- Found locations areas with previously unknown assets
- Identified mislabeled data

Study Area

Three regions in SoCal:

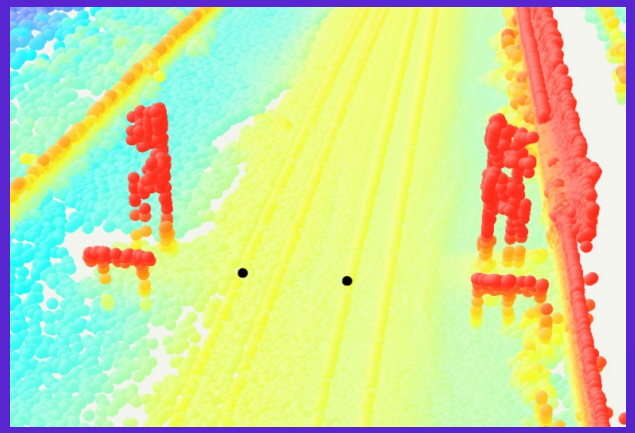
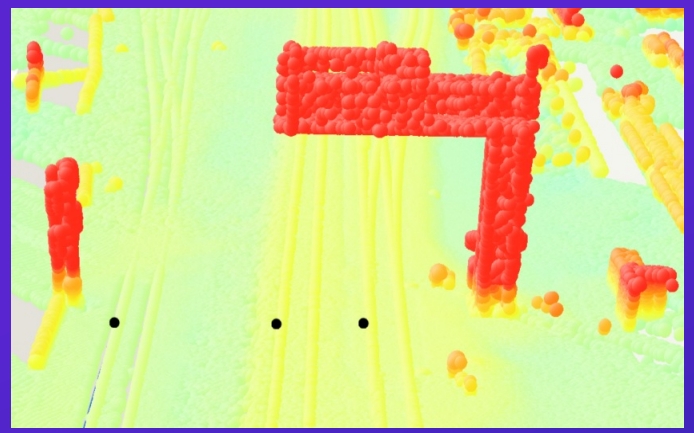
- Bakersfield
- Cajon
- San Bernardino

Data

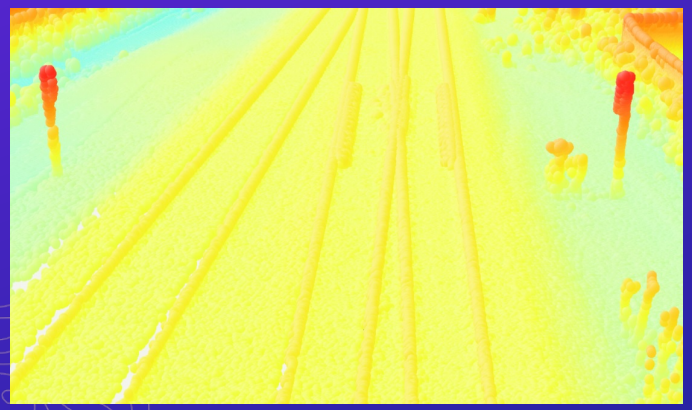
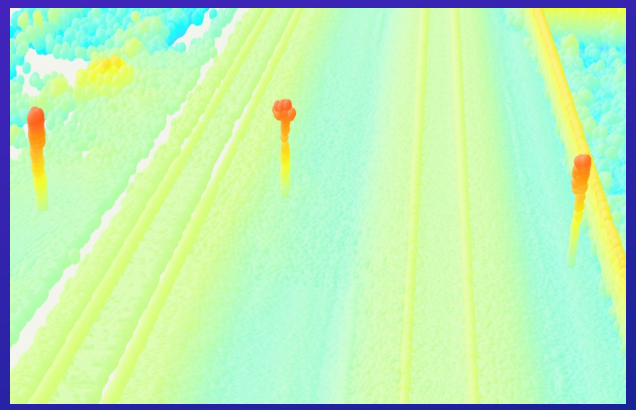
- 3D point clouds for study areas
- Ortho images in TIFF format
- Labelled asset data (point feature class)
  - 150 labels per class per region
  - ~1,800 labels (150 x 4 x 3)

Sample of Objects

- Signal

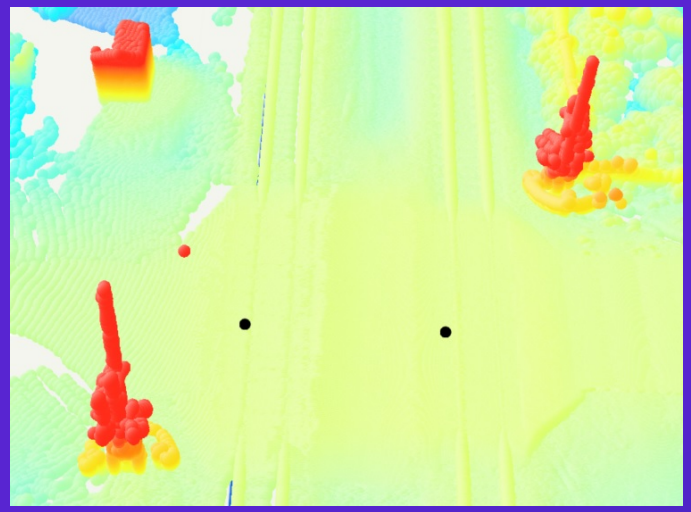
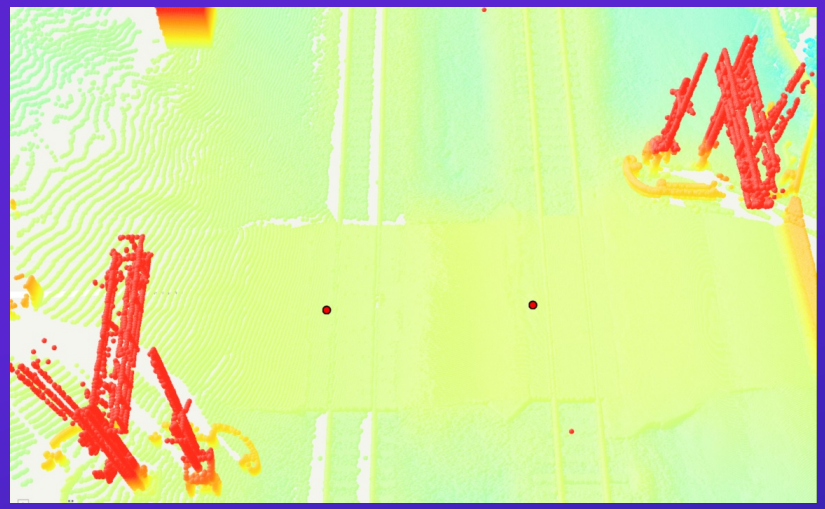




- Mile Post

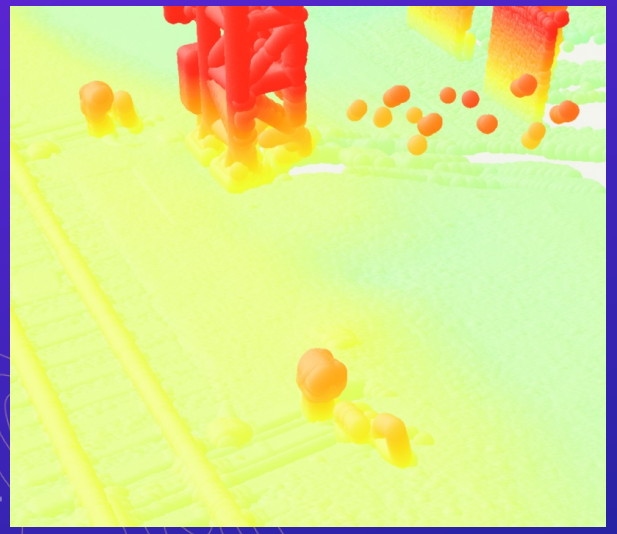




Sample of Objects

- Crossing

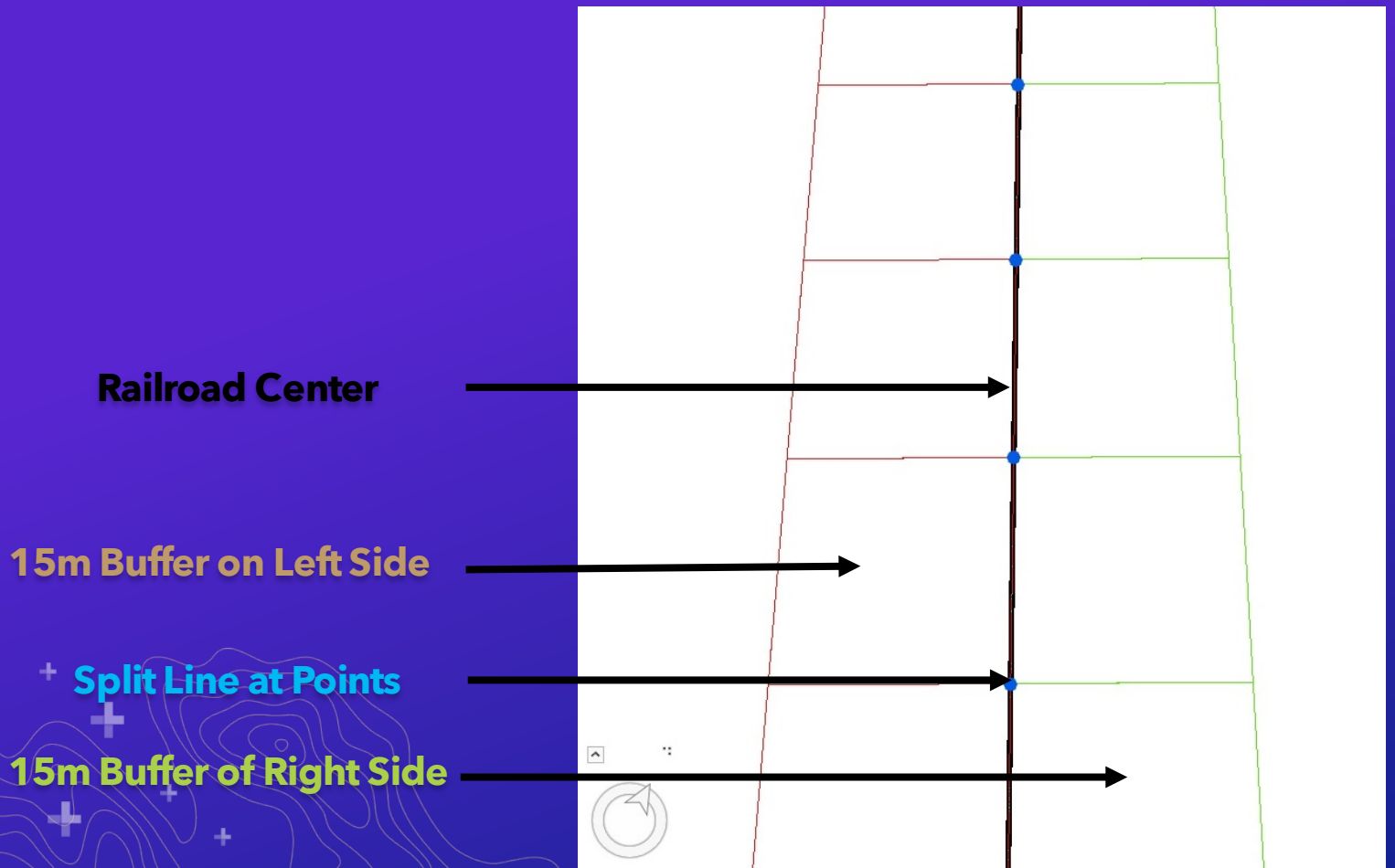

- Switch



Data Processing, Modeling & Accuracy Assessment

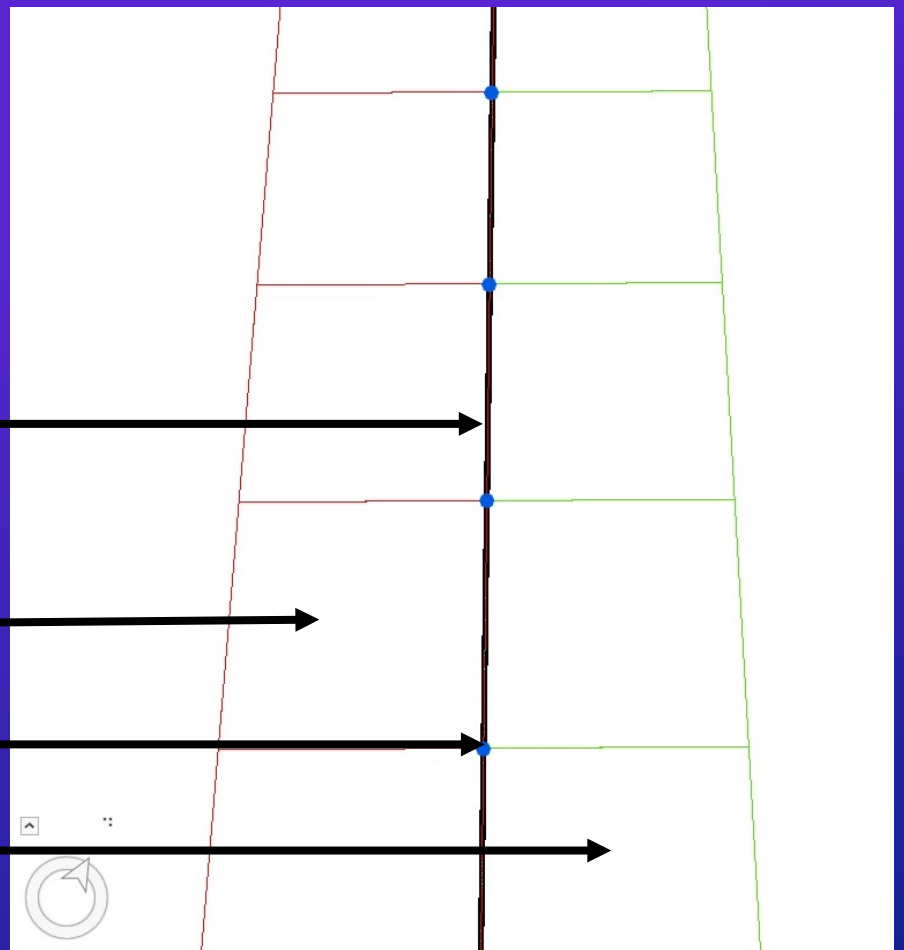
Data Processing

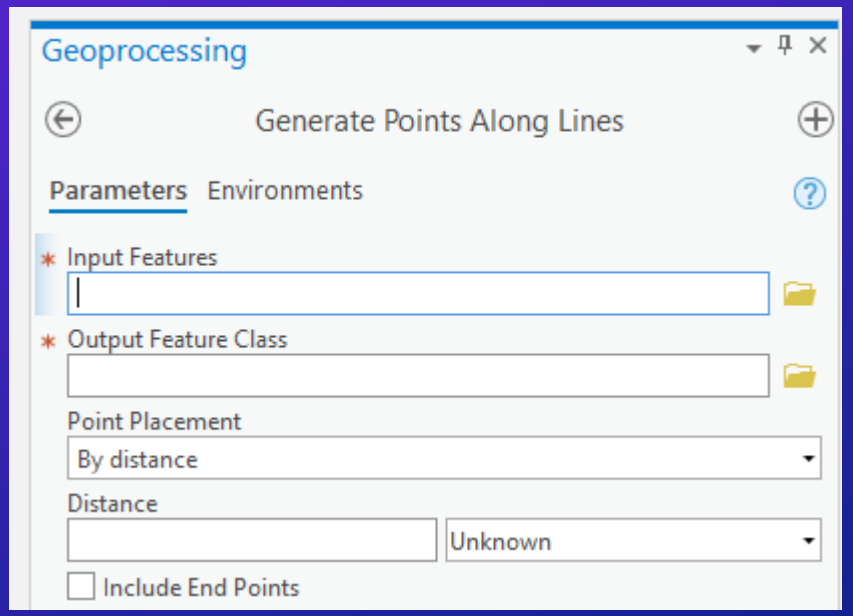
- Export Training Data for Deep Learning
- Create Unit of Analysis
- Identify Training Samples (Positive and Negative Samples)



Data Processing

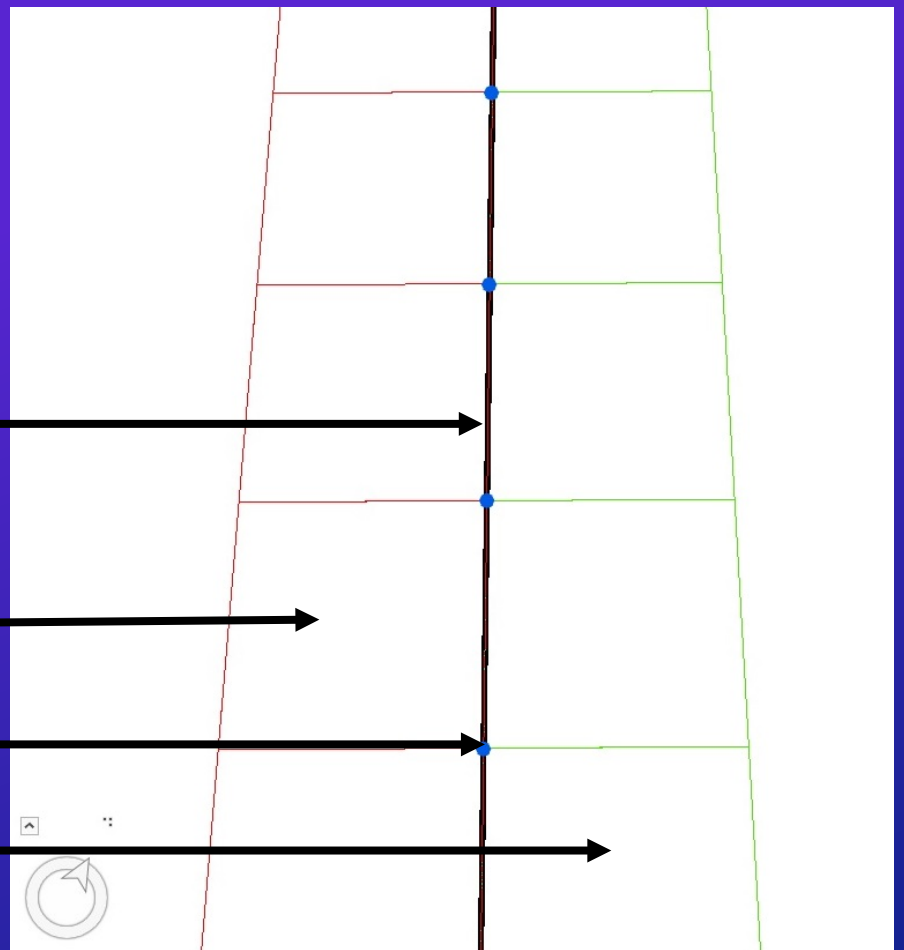
- Export Training Data for Deep Learning
- Create Unit of Analysis
- Identify Training Samples (Positive and Negative Samples)

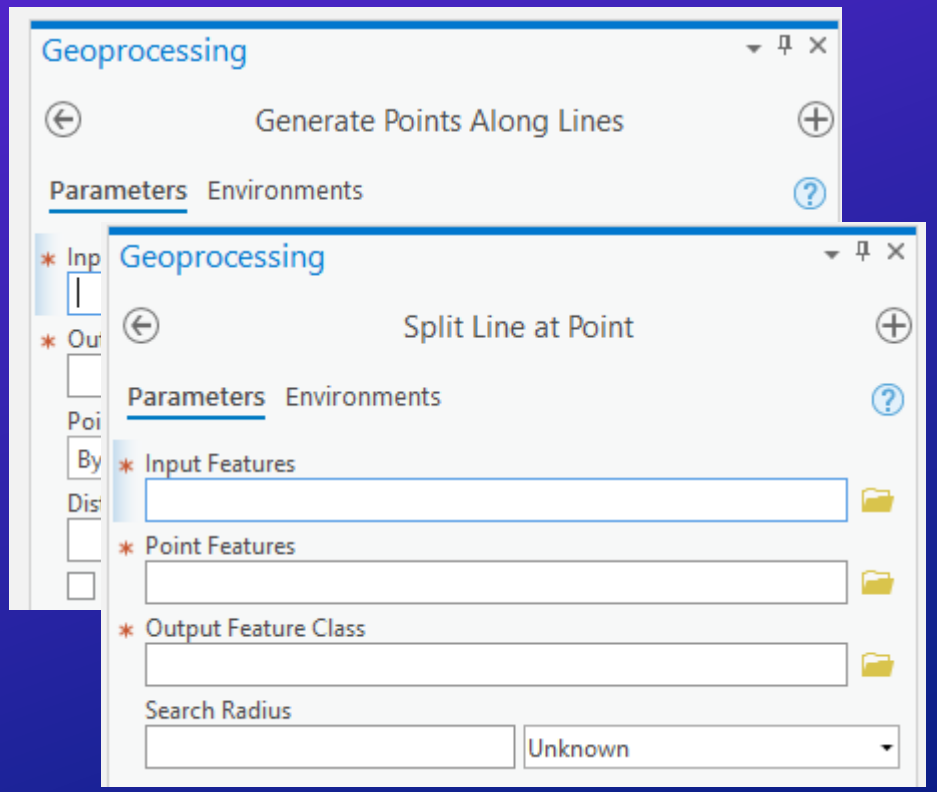




Data Processing

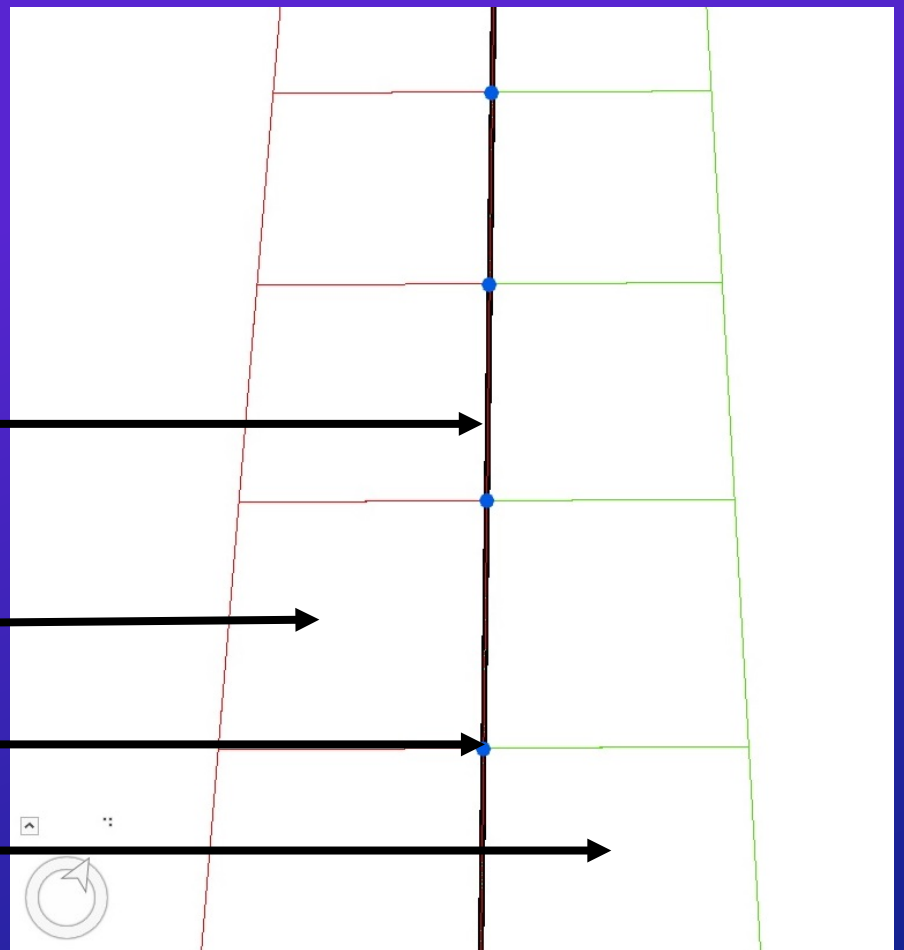
- Export Training Data for Deep Learning
- Create Unit of Analysis
- Identify Training Samples (Positive and Negative Samples)

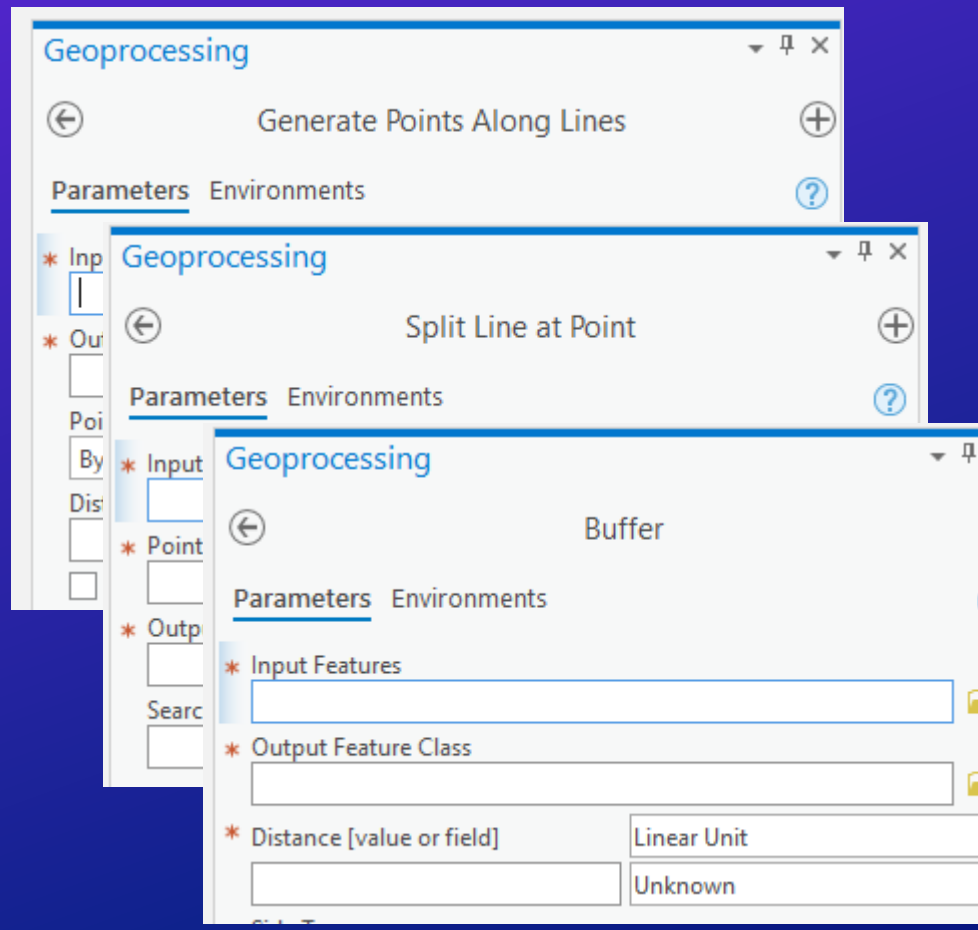




Data Processing

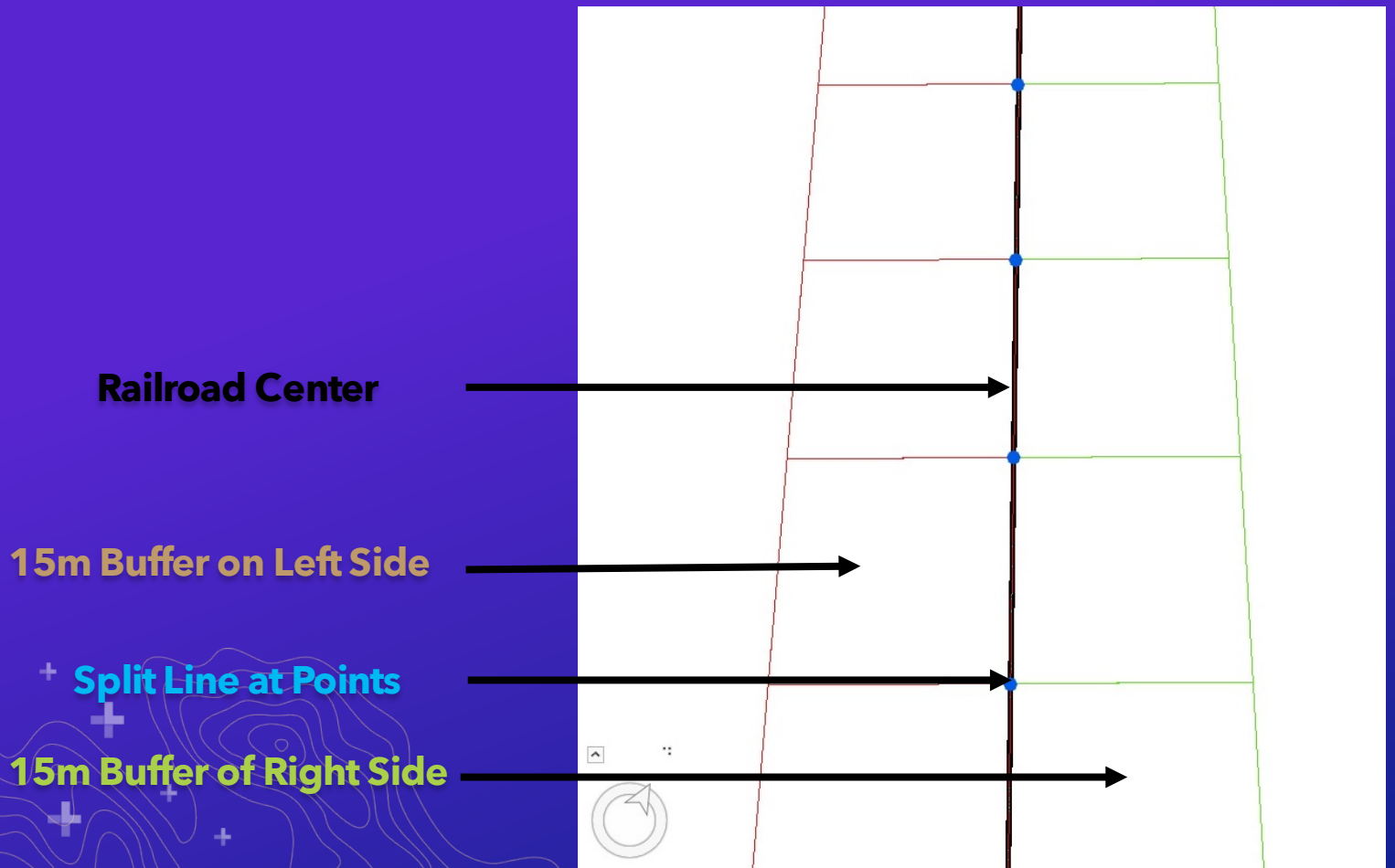
- Export Training Data for Deep Learning
- Create Unit of Analysis
- Identify Training Samples (Positive and Negative Samples)

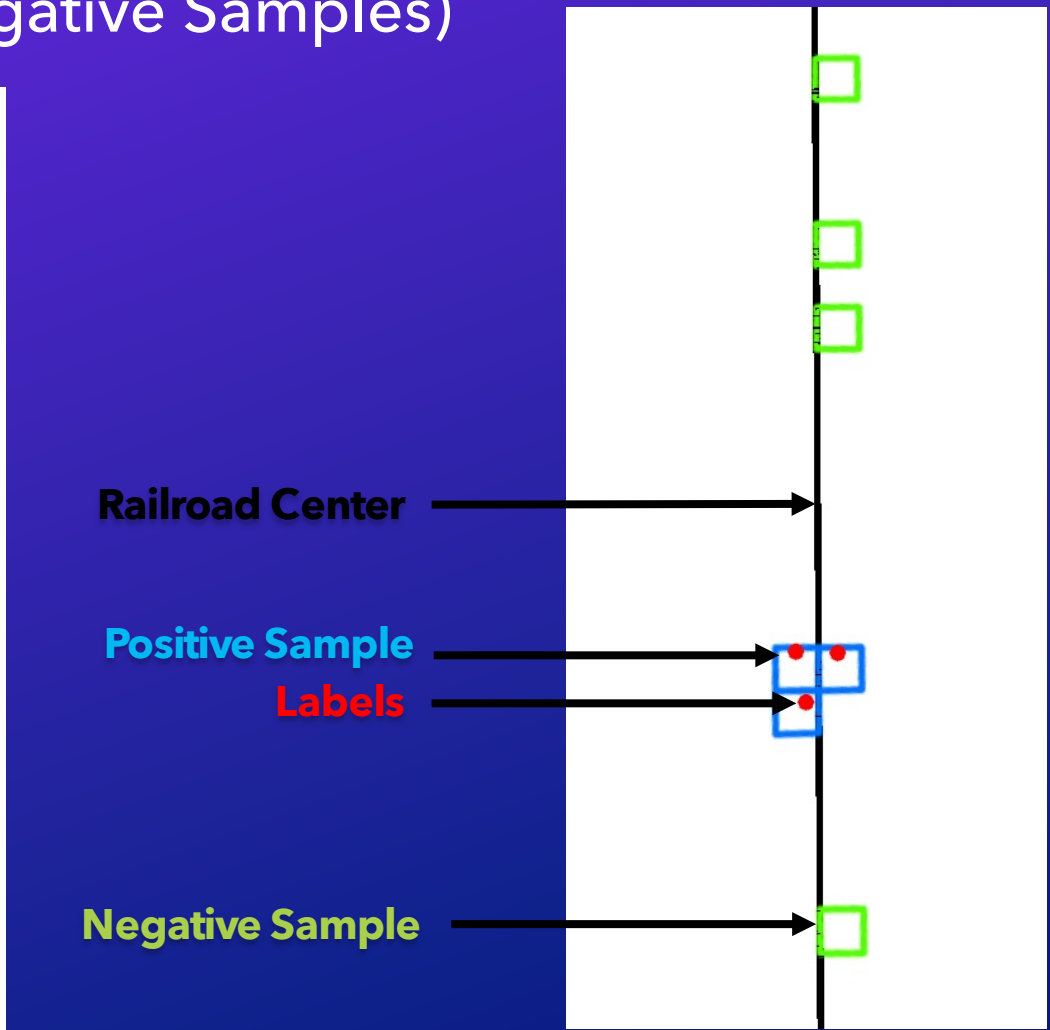




Data Processing

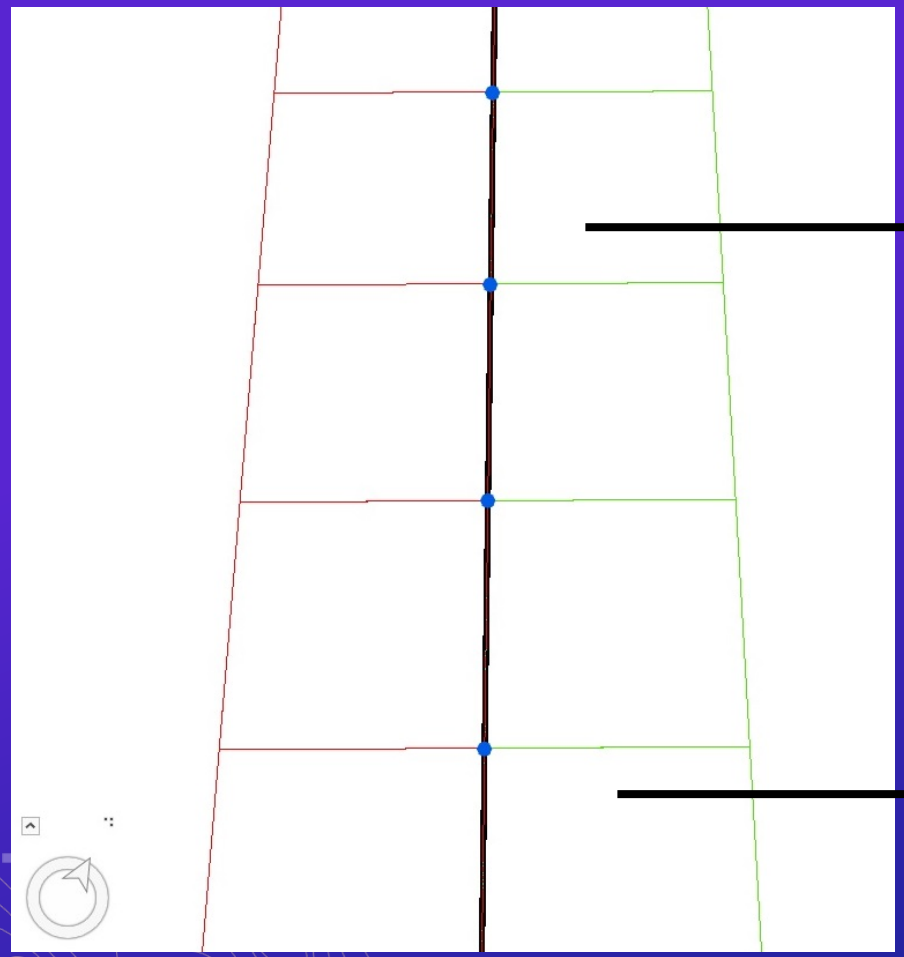
- Export Training Data for Deep Learning
- Create Unit of Analysis
- Identify Training Samples (Positive and Negative Samples)

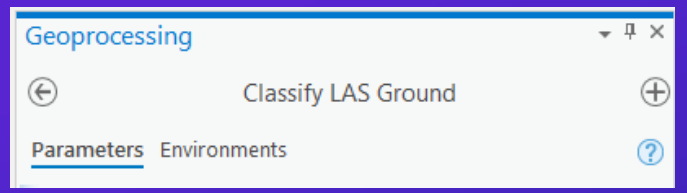




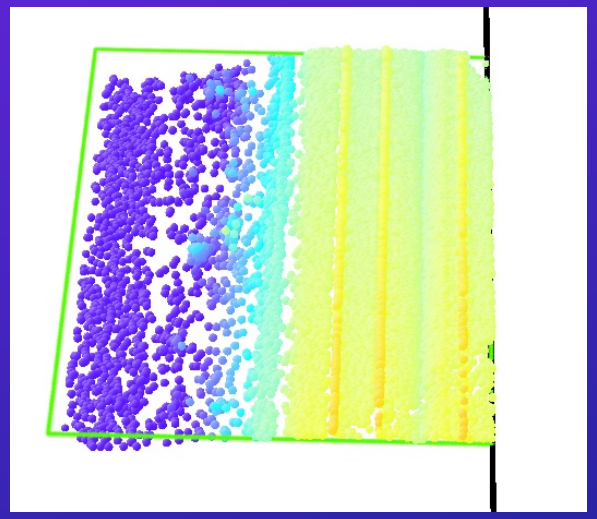
Data Processing

- Clip original LAS files to boxes

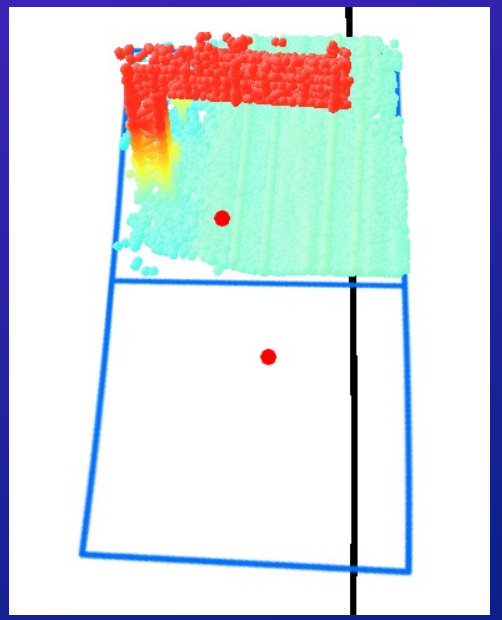




Negative Sample

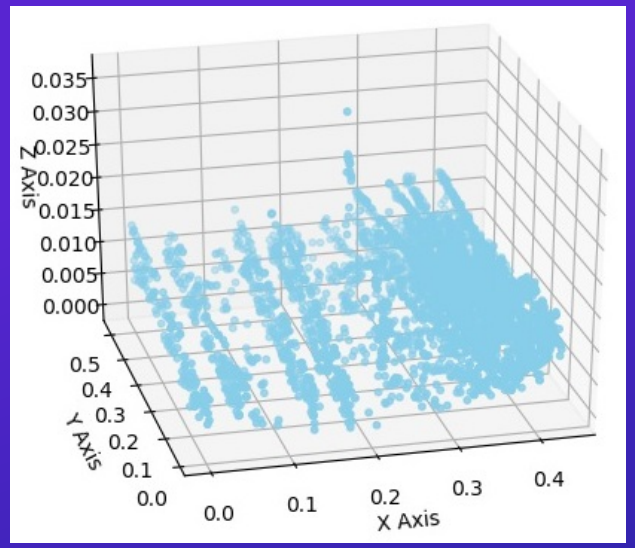


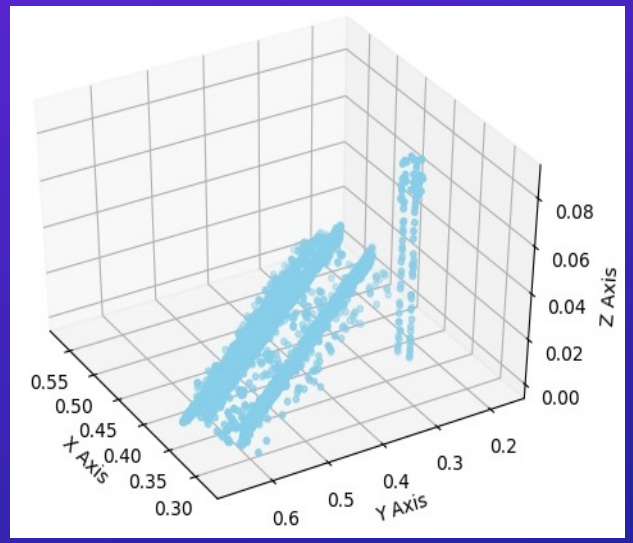
Positive Sample

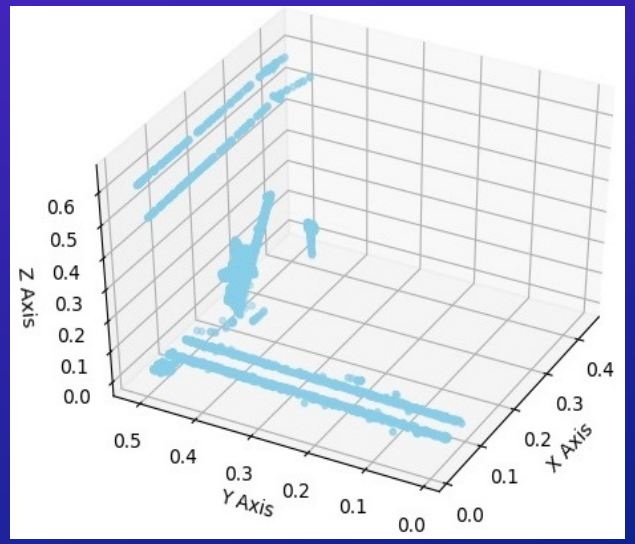


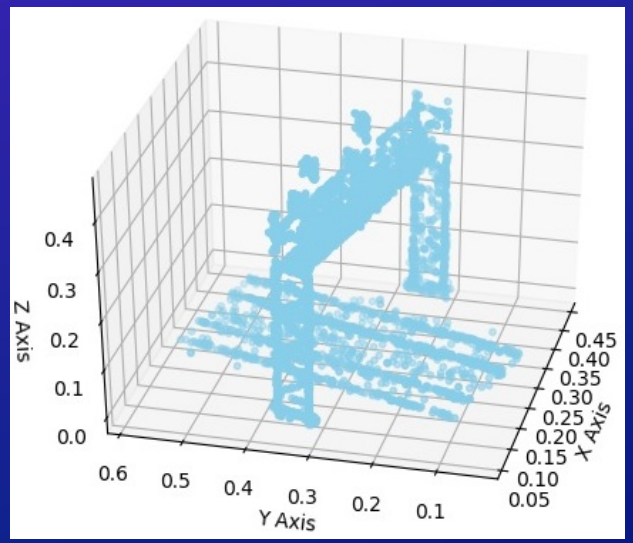
Data Processing

- Spatial Random Sampling
- Each 15m box had 50k – 200k points
- Optimal # of points for model ~4k points
- Train PointNet structure to use X, Y, relative Z, and Intensity





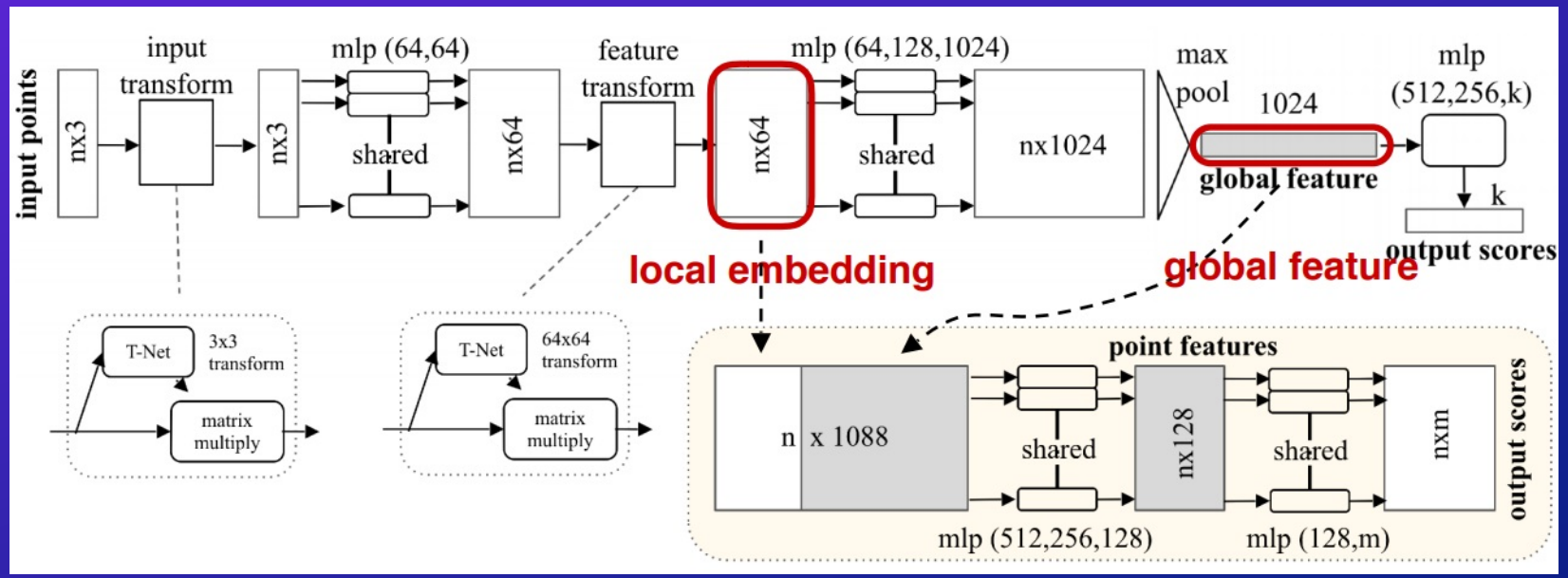




Model Training with GPU

- Training = Two Railways
- Testing = One Railway

Class	Training Run	Testing Run
Signal	242	115
Mile Post	302	103
Crossing	211	85
Switch	272	246
Negative	1667	654



Accuracy Assessment

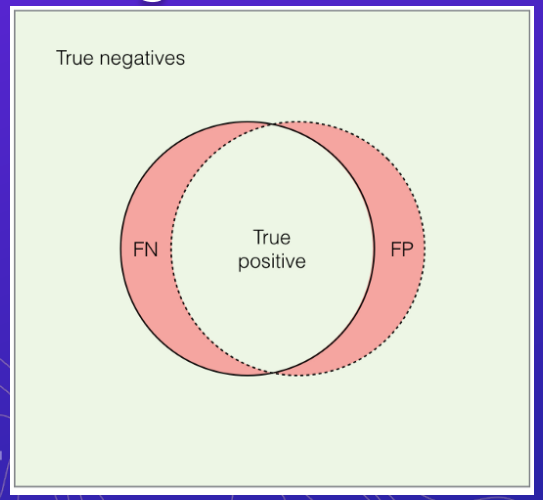
Training Data

Class	Precision	Recall
Negative	0.89	0.88
Post	0.72	0.76
Crossing	0.85	0.81
Signal	0.80	0.87
Switch	0.55	0.53

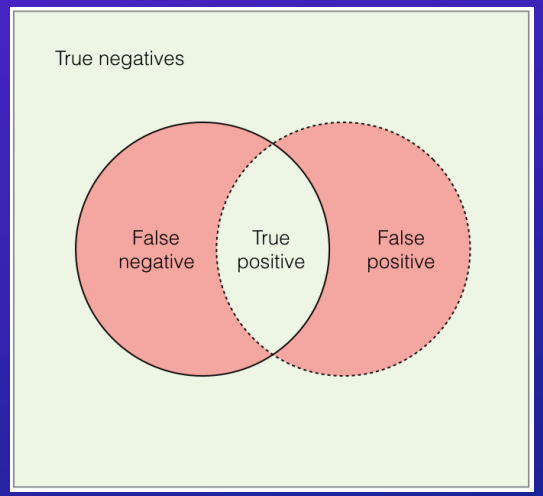
Testing Data

Class	Precision	Recall
Negative	0.86	0.85
Post	0.70	0.73
Crossing	0.74	0.72
Signal	0.76	0.75
Switch	0.48	0.51

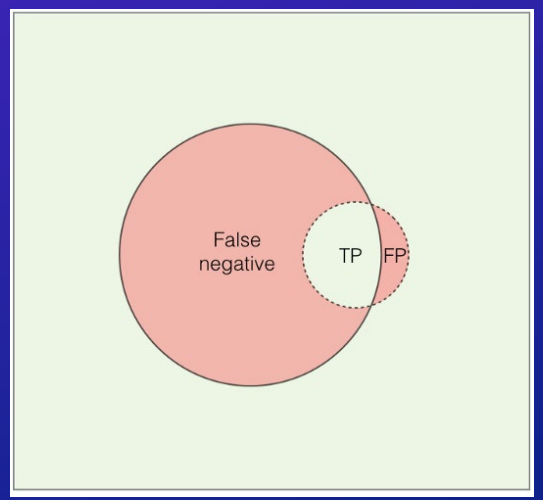
High Precision High Recall



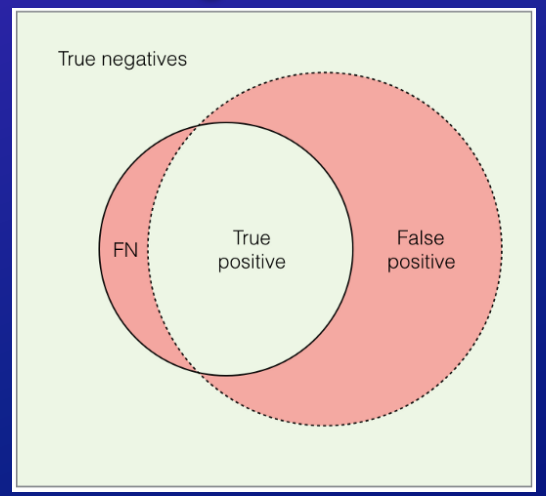
Low Precision Low Recall



High Precision Low Recall

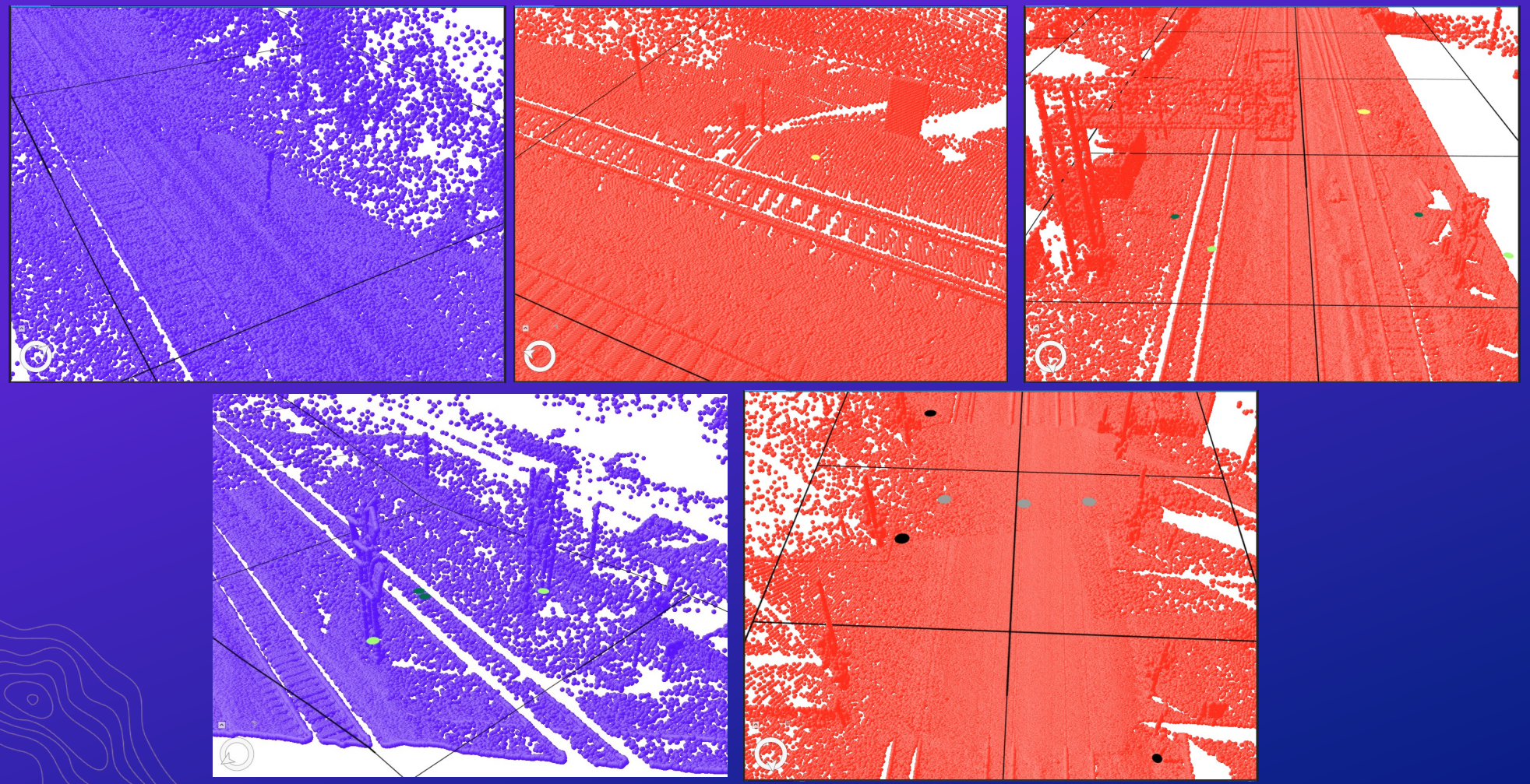


Low Precision High Recall



Output Results

- Point feature class of identified asset locations
- Attribution classifying each point as specific asset type

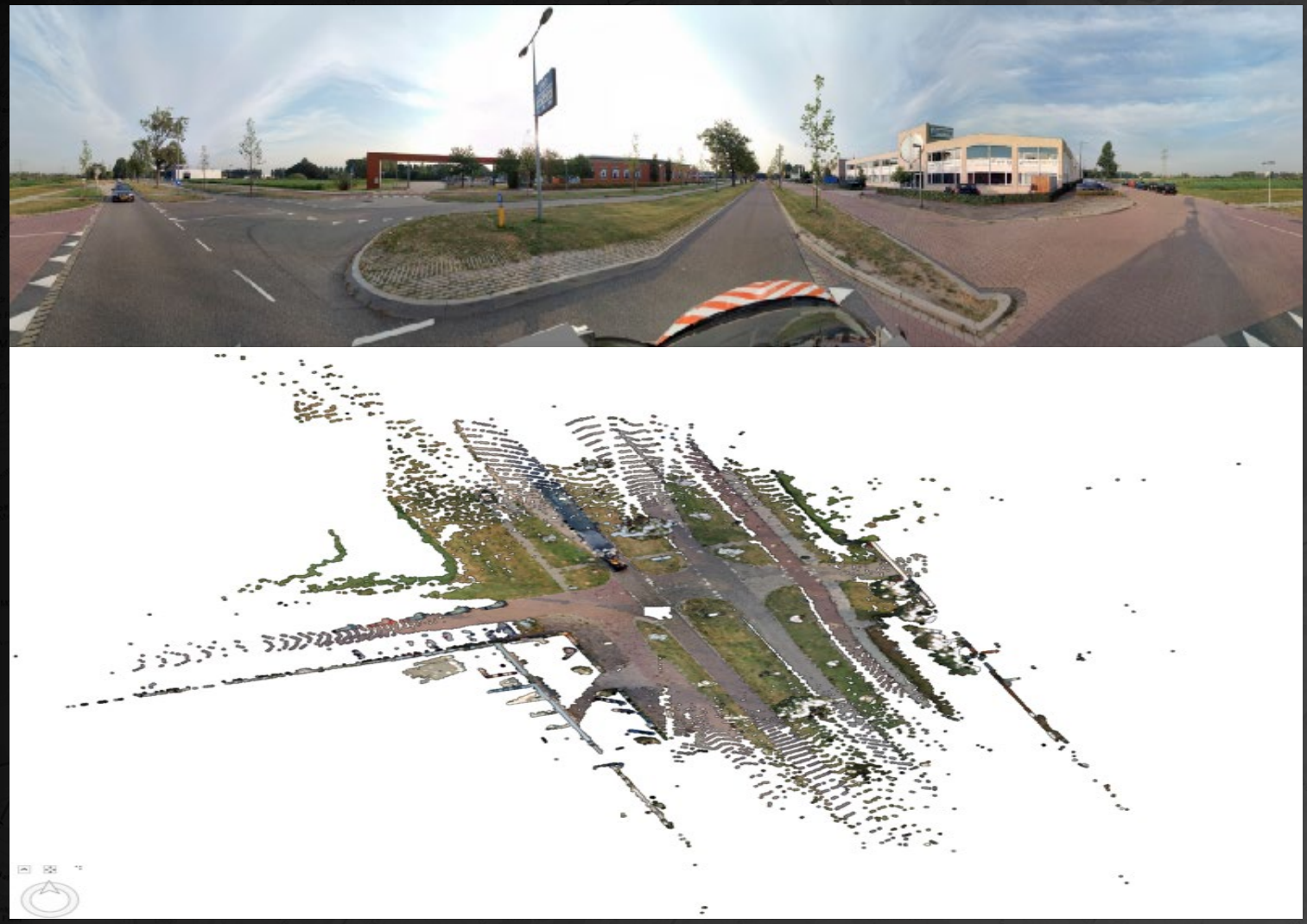


PointCNN & Feature Extraction: mobile point clouds

esri | THE SCIENCE OF WHERE

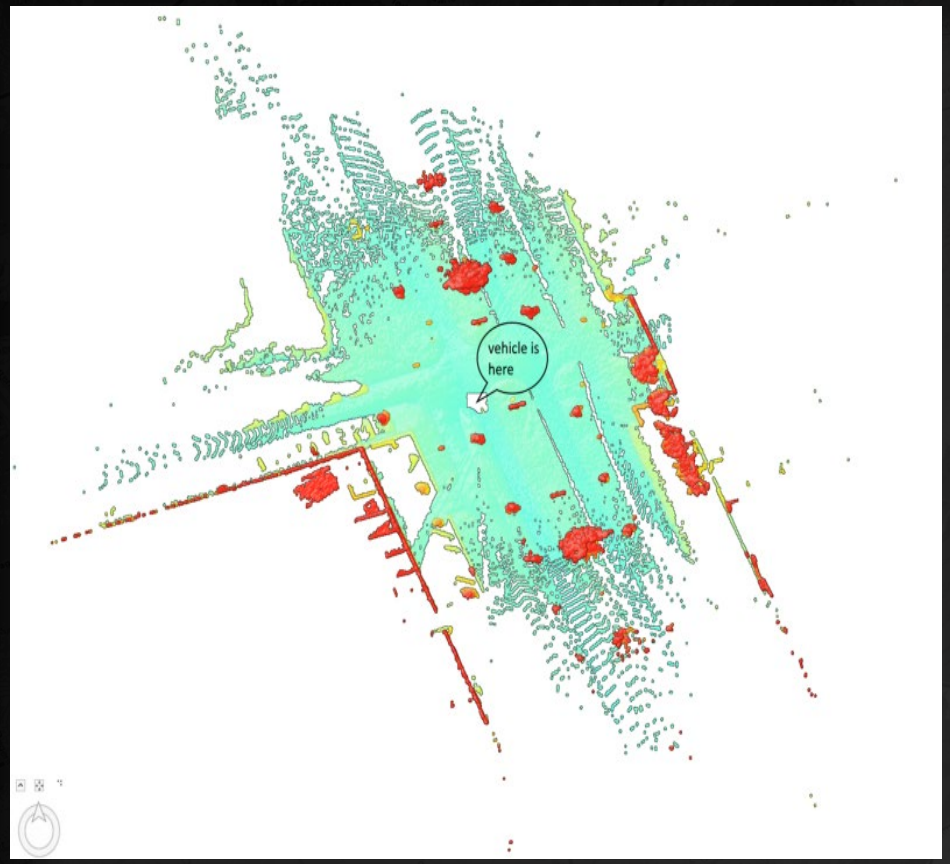
PoC: PointCNN in Mobile Point Clouds

CycloMedia's "CycloRama" sample point cloud with synced 360-imagery



PoC: PointCNN in Mobile Point Clouds

Mobile point clouds have two peculiarities making them harder to work with compared to airborne data:

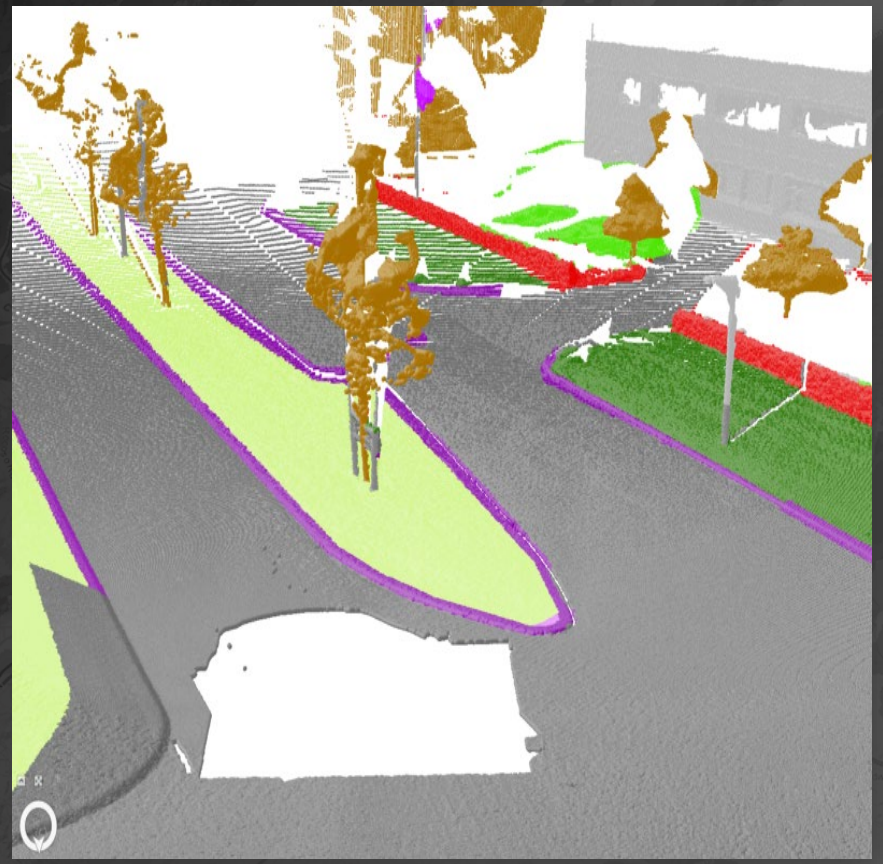


1. Point Density
2. Gaps in Data

PoC: PointCNN in Mobile Point Clouds

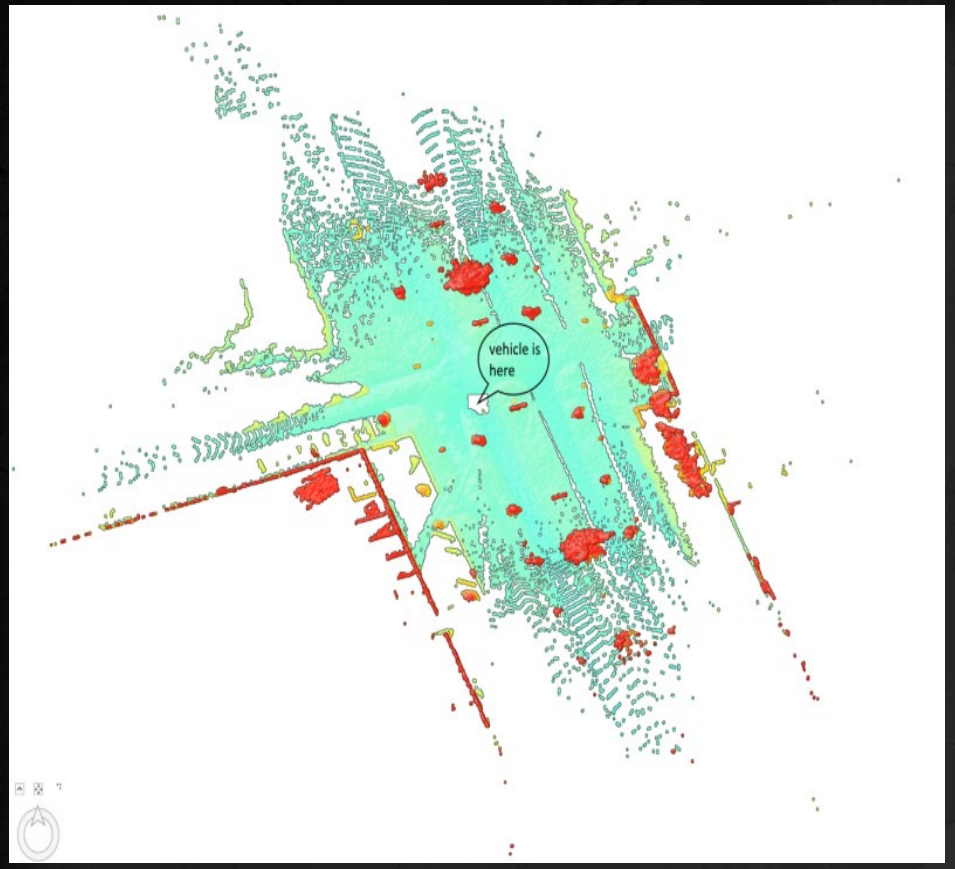
An advantage: simpler to label and create training data

Classifying by projecting 2D semantic masks onto the points

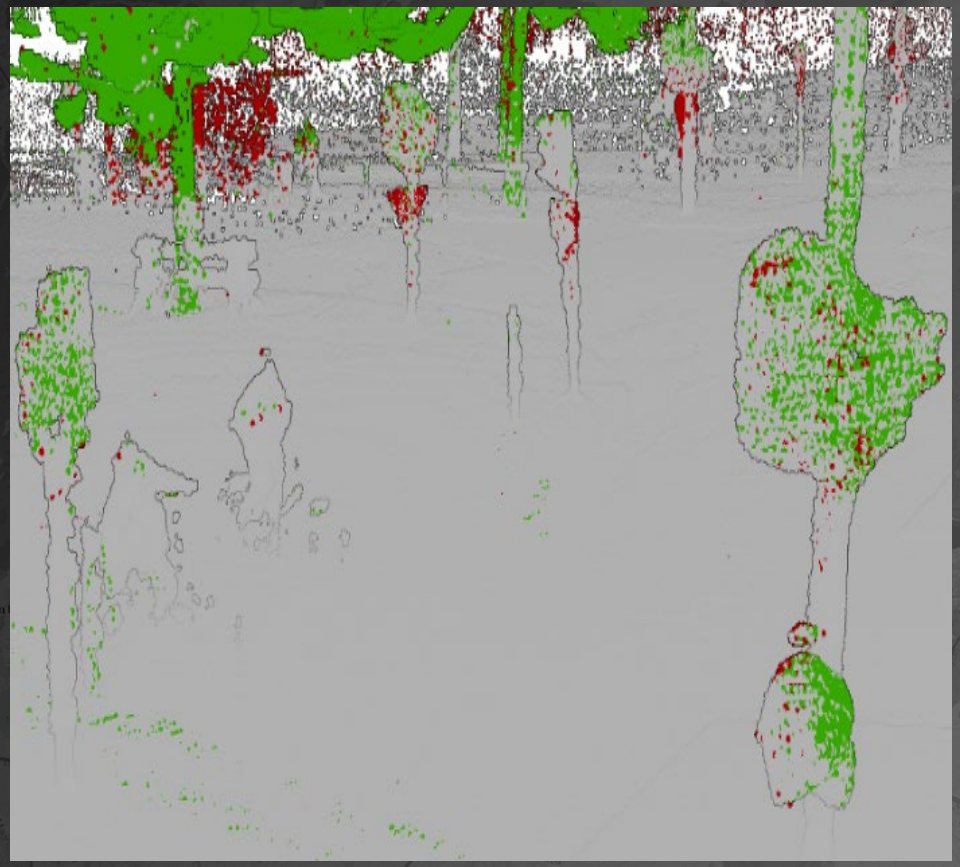


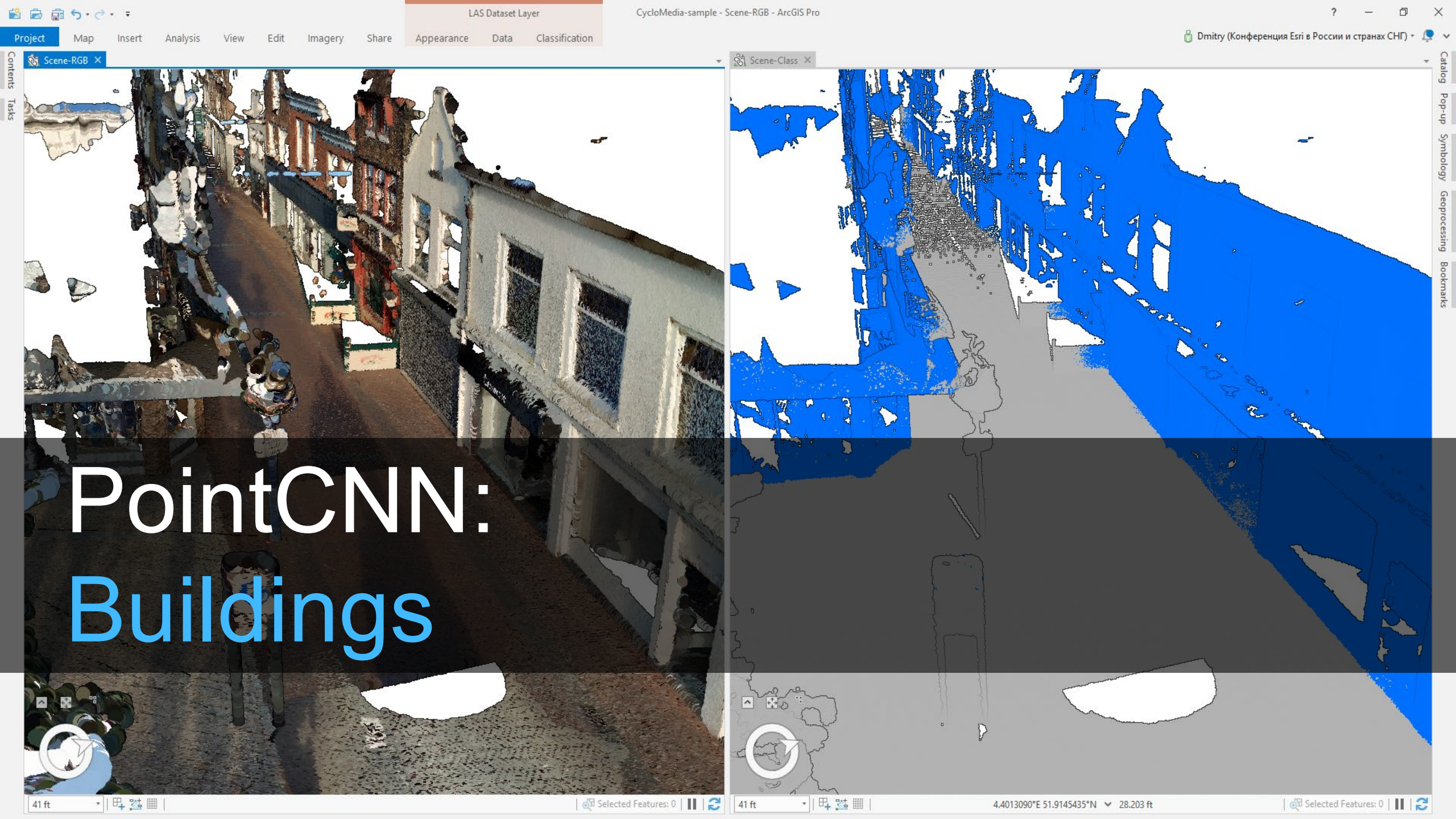
PoC: PointCNN in Mobile Point Clouds

Training on individual clouds or single consolidated cloud?



?

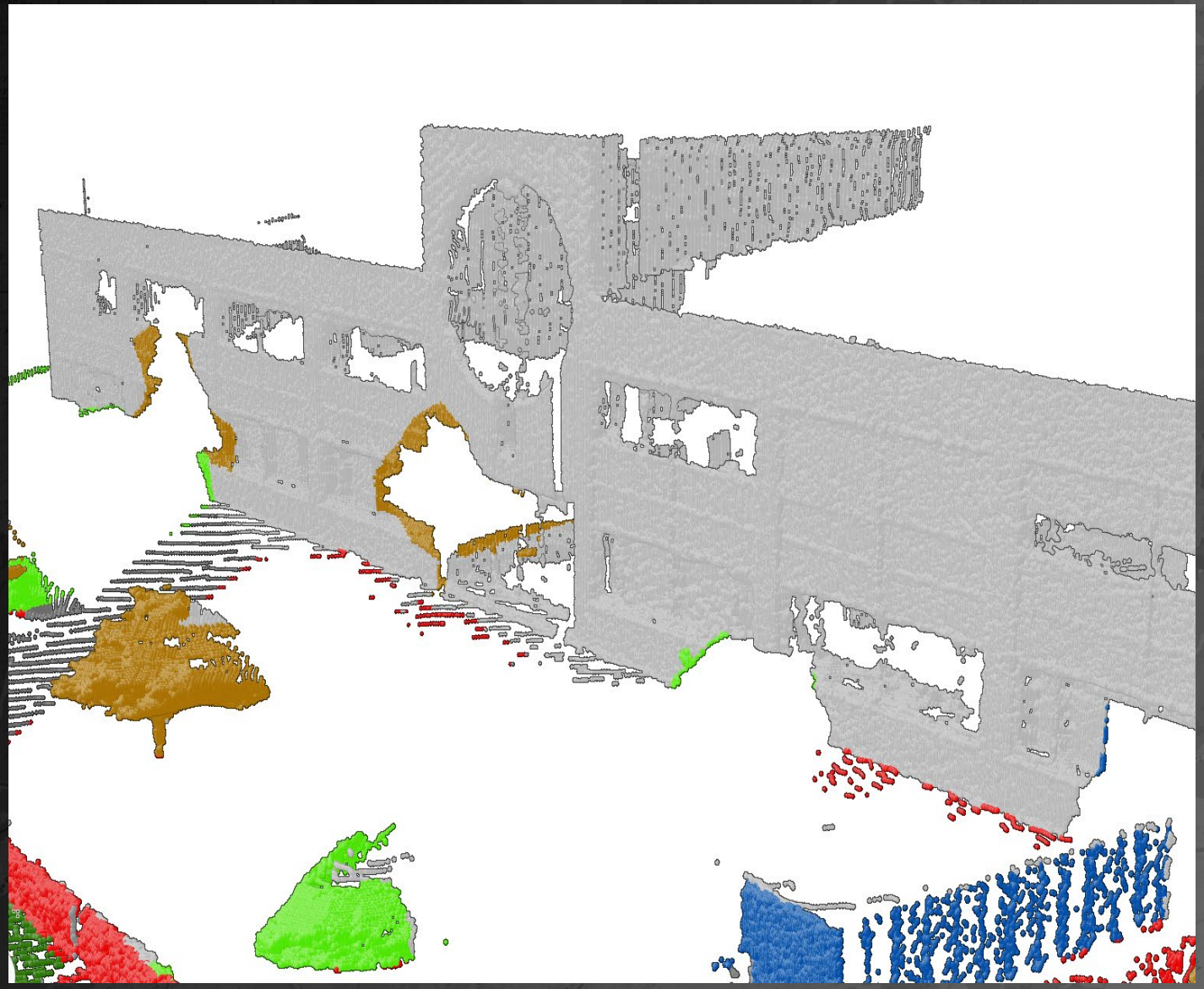



LAS Dataset Layer
CycloMedia-sample - Scene-RGB - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Data
Classification
Dmitry (Конференция Esri в России и странах СНГ)
Scene-RGB
Scene-Class
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
PointCNN:
Buildings
41 ft
Selected Features: 0
41 ft
4.4013090°E 51.9145435°N
28.203 ft
Selected Features: 0

PoC: PointCNN in Mobile Point Clouds

Buildings are well represented in the training set:

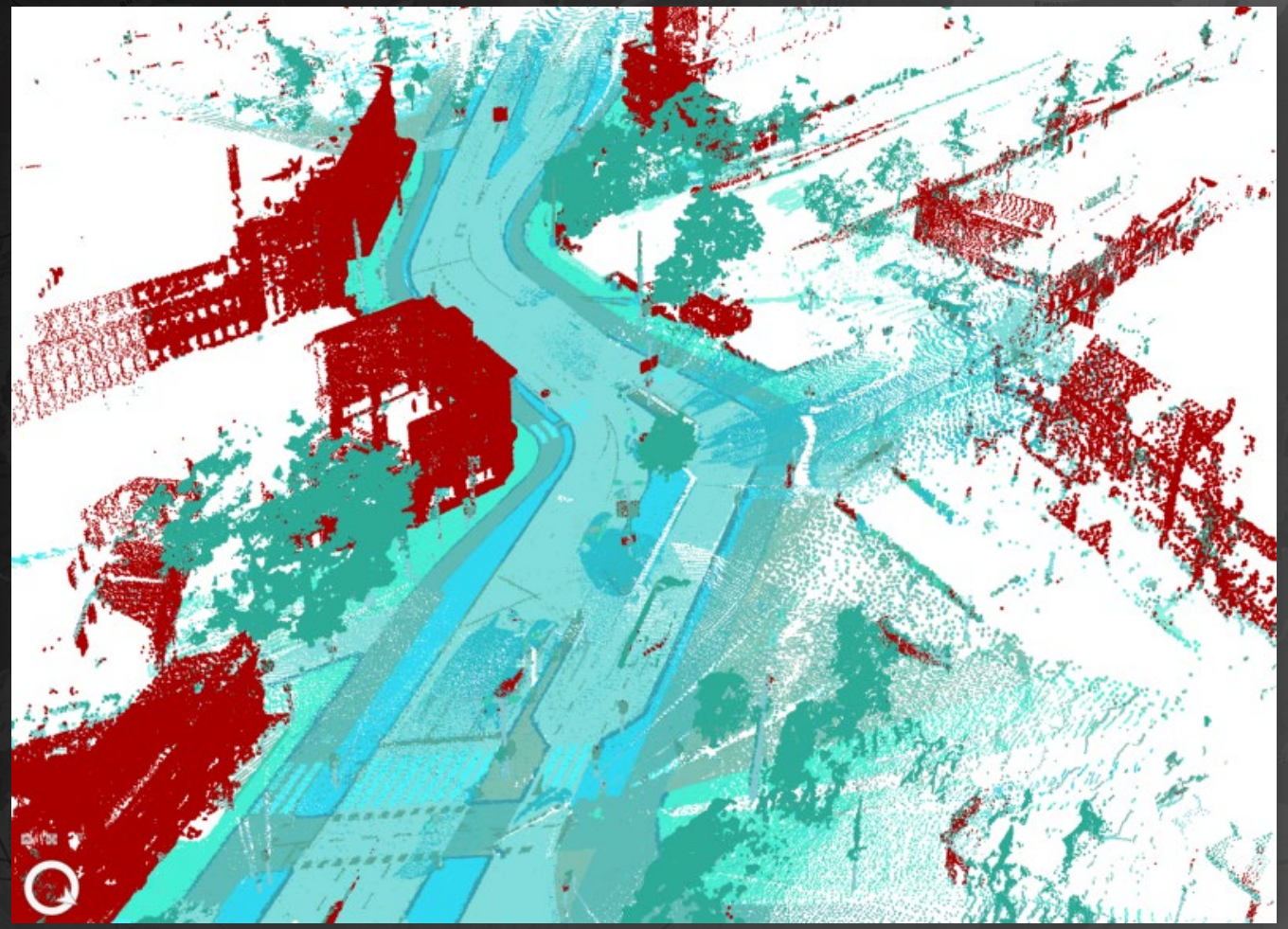
POINTS_COUNT	CLASS_NAME
539,065,936	road
361,835,379	building
284,781,152	vegetation_high
99,861,937	sidewalk
70,959,436	car
70,826,139	grass
55,858,858	parking
47,782,986	vegetation_low
43,682,759	barrier_wall
38,715,988	bike_lane
33,589,152	curb
32,879,547	barrier_separator
17,685,268	structure_bridge



PoC: PointCNN in Mobile Point Clouds

After ~36 hours of training on NVIDIA Quadro GV100:

```
@E150K
              OTHER       BUILDING
Precision:[0.99126286 0.96047395],
Recall:   [0.9923109  0.95528204],
F1-score: [0.9917866  0.95787096]
```

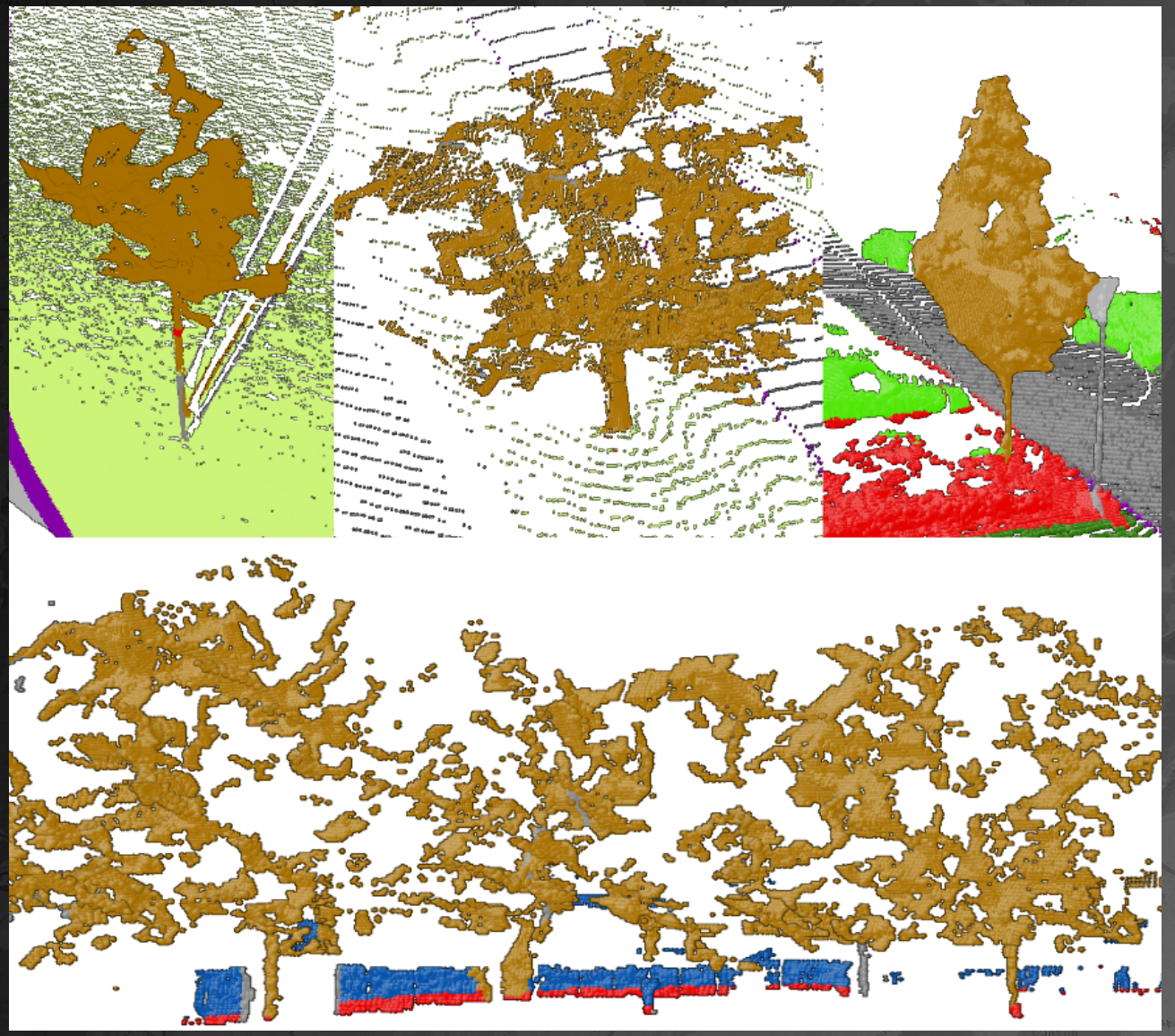



Point Cloud Layer
CycloMedia-sample - Scene-Class - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Scene-RGB
Scene-Class
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
PointCNN:
Trees
74 ft
Selected Features: 0
74 ft
4.3916330°E 51.9112411°N
2.453 ft
Selected Features: 0

PoC: PointCNN in Mobile Point Clouds

Trees are too well represented:

POINTS_COUNT	CLASS_NAME
539,065,936	road
361,835,379	building
284,781,152	vegetation_high
99,861,937	sidewalk
70,959,436	car
70,826,139	grass
55,858,858	parking
47,782,986	vegetation_low
43,682,759	barrier_wall
38,715,988	bike_lane
33,589,152	curb
32,879,547	barrier_separator
17,685,268	structure_bridge



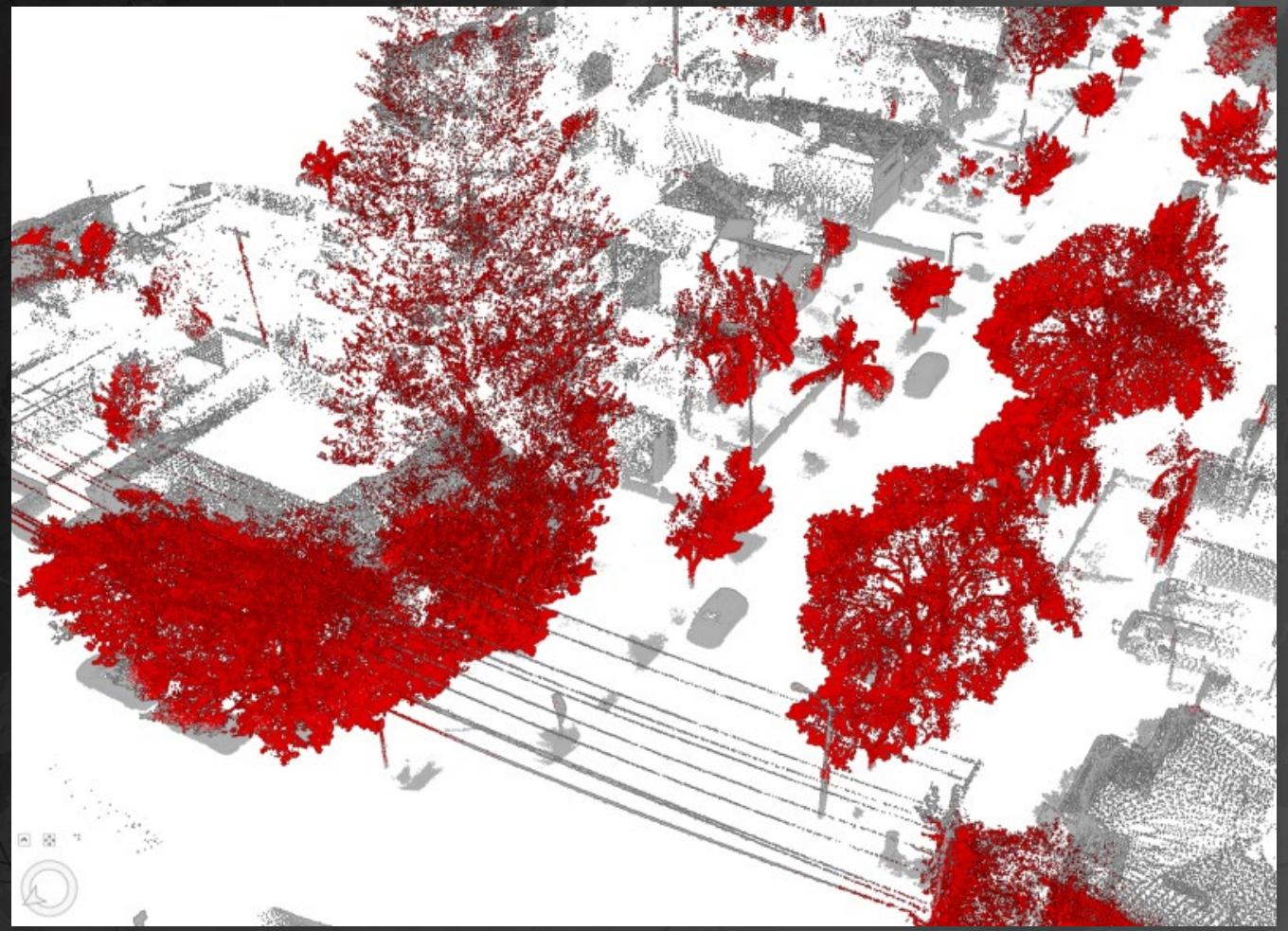
But harder to learn than buildings:

- Species
- Age
- Density
- RGB ranges
- Canopy type

PoC: PointCNN in Mobile Point Clouds

After ~24 hours of training on GV100:

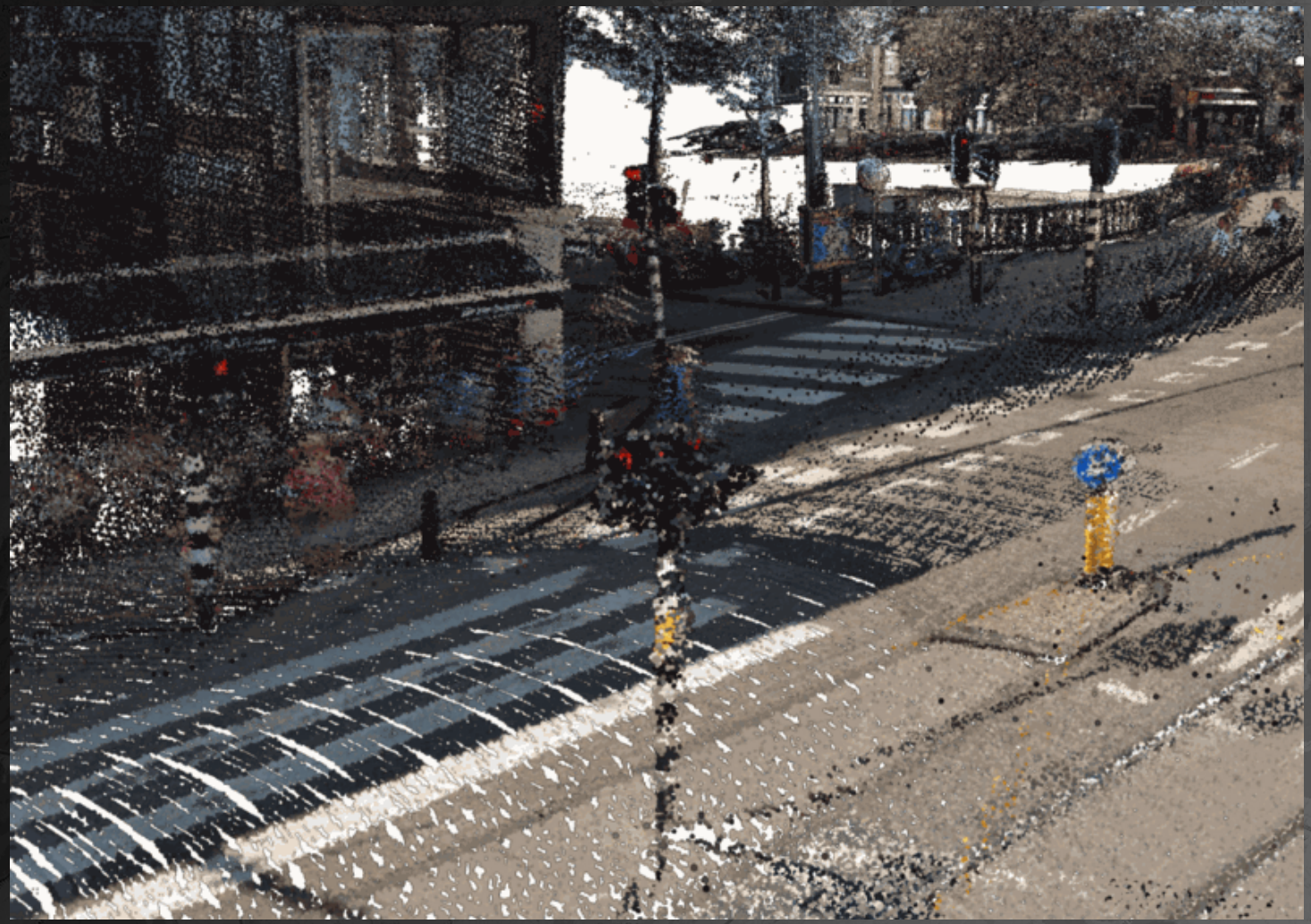
```
@E100K
              OTHER       TREE
Precision:[0.98007521 0.94283636],
Recall:   [0.98718658 0.91327329],
F1-score: [0.98361804 0.92781940]
```




Point Cloud Layer
CycloMedia-sample - Scene-Class - ArcGIS Pro
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Scene-RGB
Scene-Class
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
PointCNN:
Traffic Lights
64 ft
Selected Features: 0
4.4023086°E 51.9145364°N
10.018 ft

PoC: PointCNN in Mobile Point Clouds

- Traffic Lights are harder:
  - Small number of samples ~580,000 points (0.032%)
  - Different types, attachment options
  - Located at most noisy intersections



PoC: PointCNN in Mobile Point Clouds

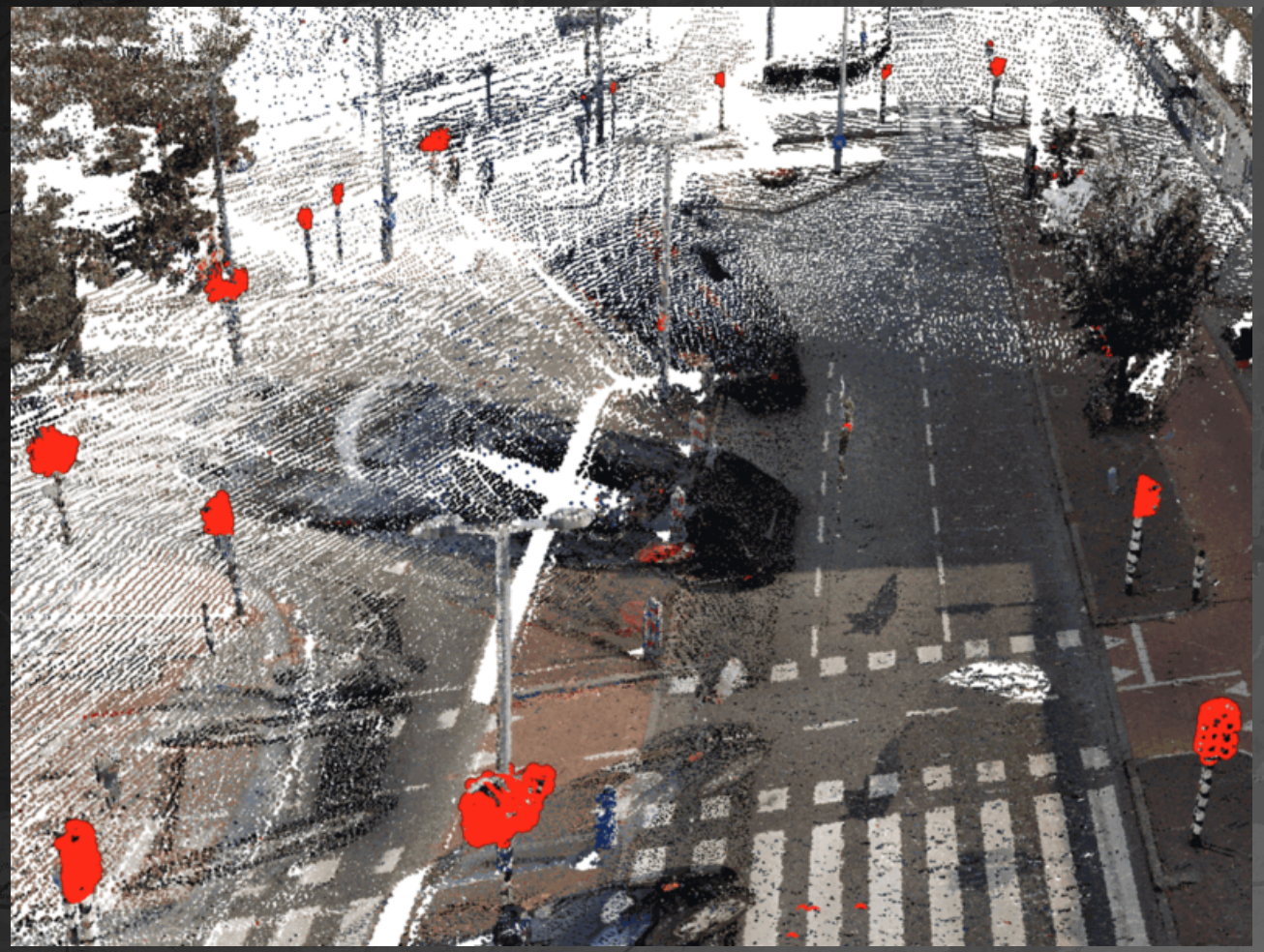
- Traffic Lights are harder:
  - Small number of samples ~580,000 points (0.032%)
  - Different types, attachment options
  - Located at most noisy intersections

PoC: PointCNN in Mobile Point Clouds

After ~72 hours of training on GV100:

```
@E300K
              OTHER      TRAFFIC LIGHT
Precision:[0.99974482 0.51987179],
Recall:   [0.99939181 0.72079467],
F1-score: [0.99956828 0.60406375]
```

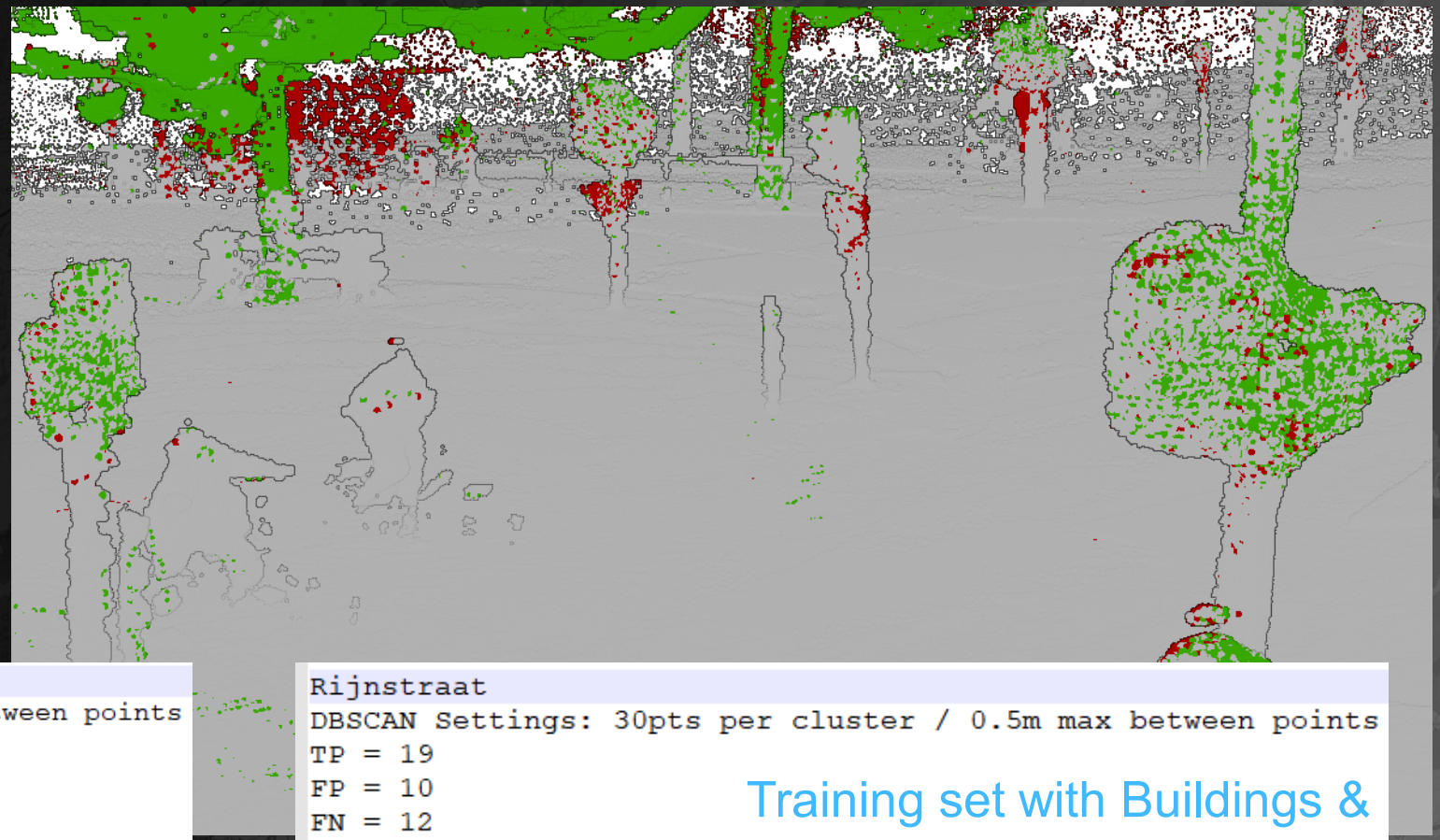
```
Extracted Objects, after DBScan:
DBSCAN 30pts/0.5m
Precision = .655
Recall = .704
F1-Score = .679
```



PoC: PointCNN in Mobile Point Clouds

Denoising trick with PointCNN:

- If the valuable semantics is consistent, but noise is random, a trained model can be used as a ”low-pass noise filter”

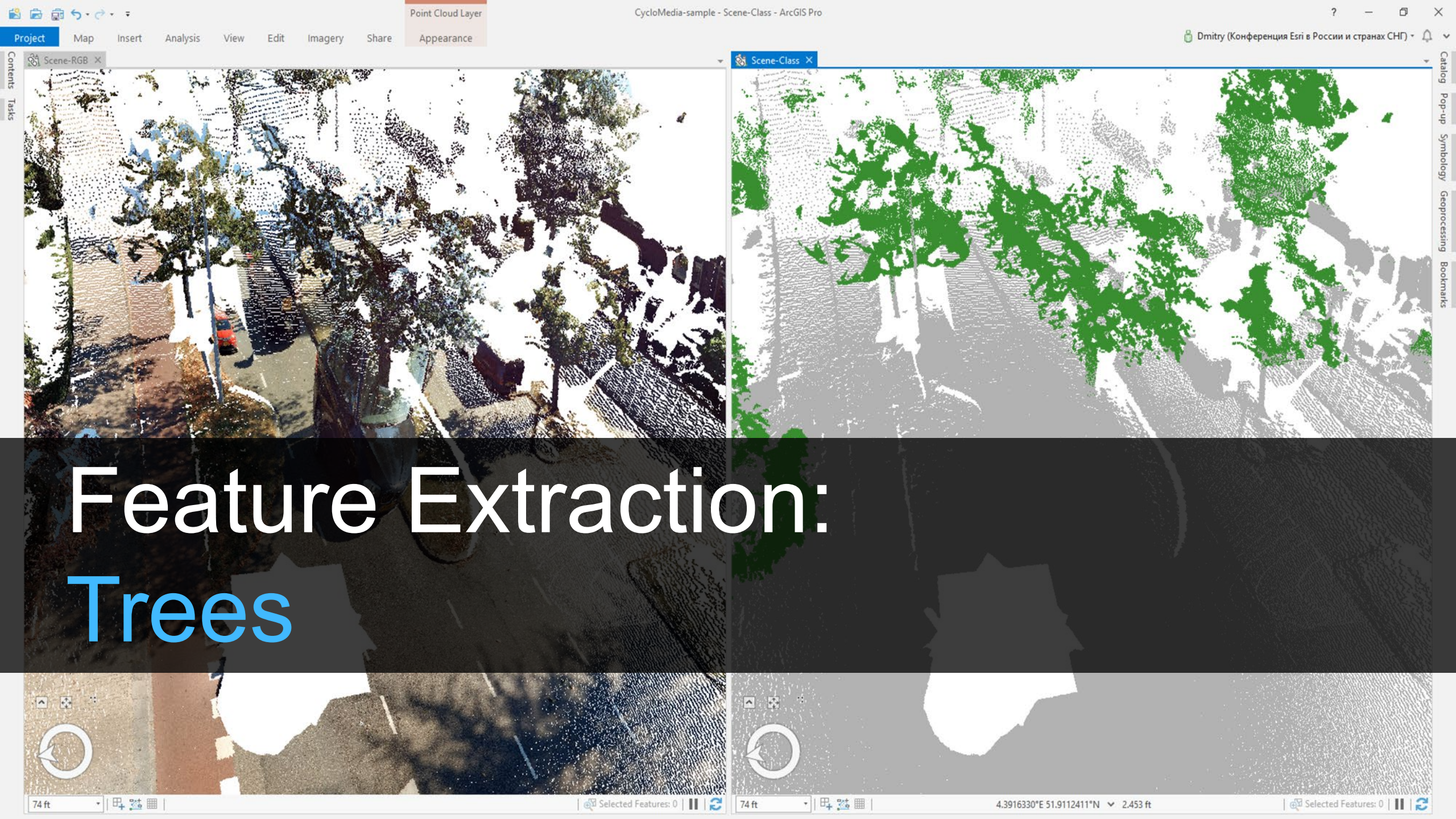


```
Rijnstraat
DBScan Settings: >=3 pts per cluster, .5m max between points
GroundTruth = 31 traffic lights
TruePositivies = 21
FalsePositives = 31
FalseNegatives = 10
Precision = .404
Recall = .677
F1-Score = .506
```

Original Training set

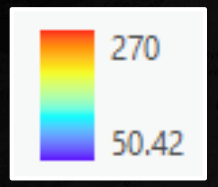
```
Rijnstraat
DBSCAN Settings: 30pts per cluster / 0.5m max between points
TP = 19
FP = 10
FN = 12
Precision = .655
Recall = .704
F1-Score = .679
```

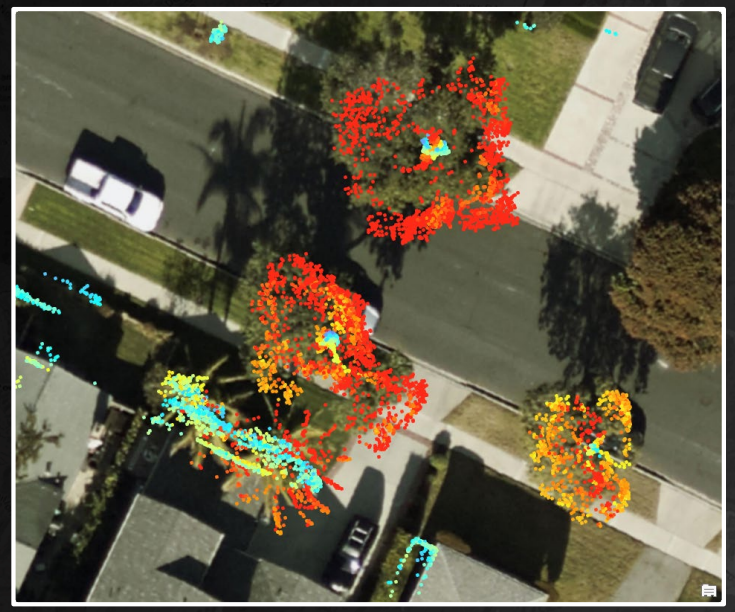
Training set with Buildings & Trees relabeled by PointCNN models

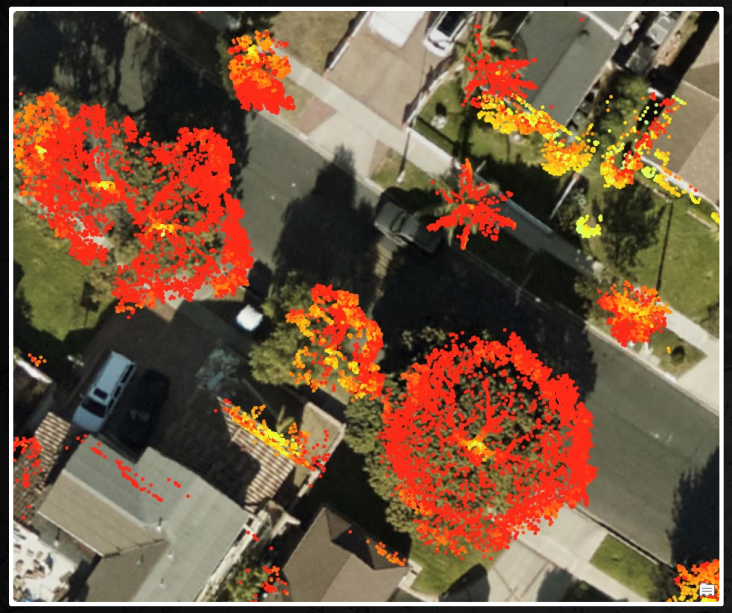

CycloMedia-sample - Scene-Class - ArcGIS Pro
Point Cloud Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Scene-RGB
Scene-Class
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
Feature Extraction:
Trees
74 ft
Selected Features: 0
74 ft
4.3916330°E 51.9112411°N
2.453 ft
Selected Features: 0

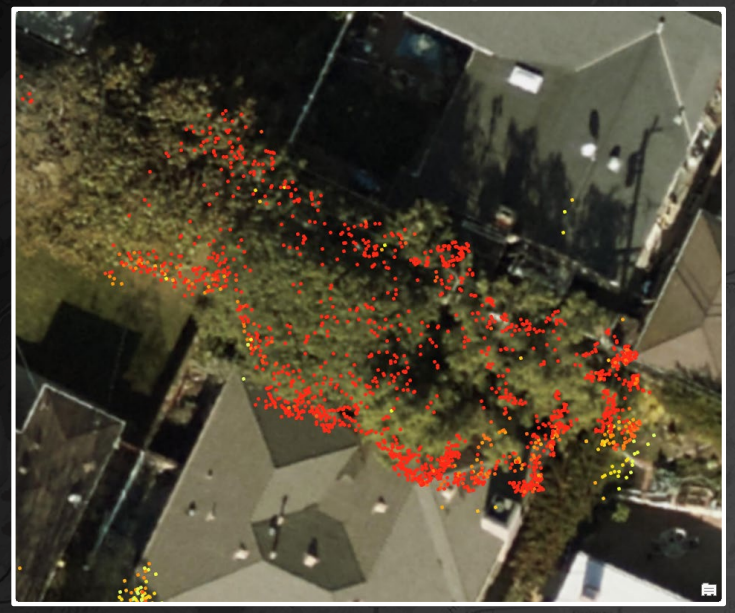
Los Angeles data - Trees

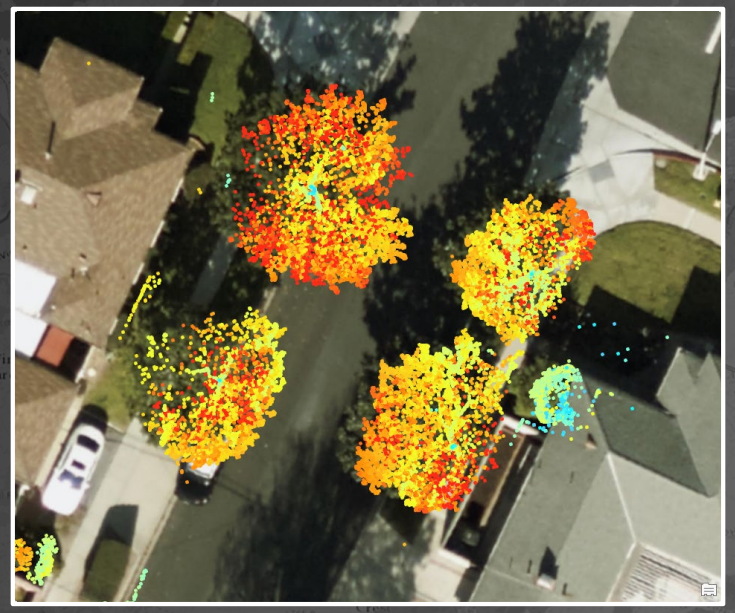
- Number of LAS files: 24
- Number of LAS points: 33.5 million
- XYZ units: US foot
- Average point spacing: 0.11 feet
- Classes
  - 0 = Other
  - 1 = Trees



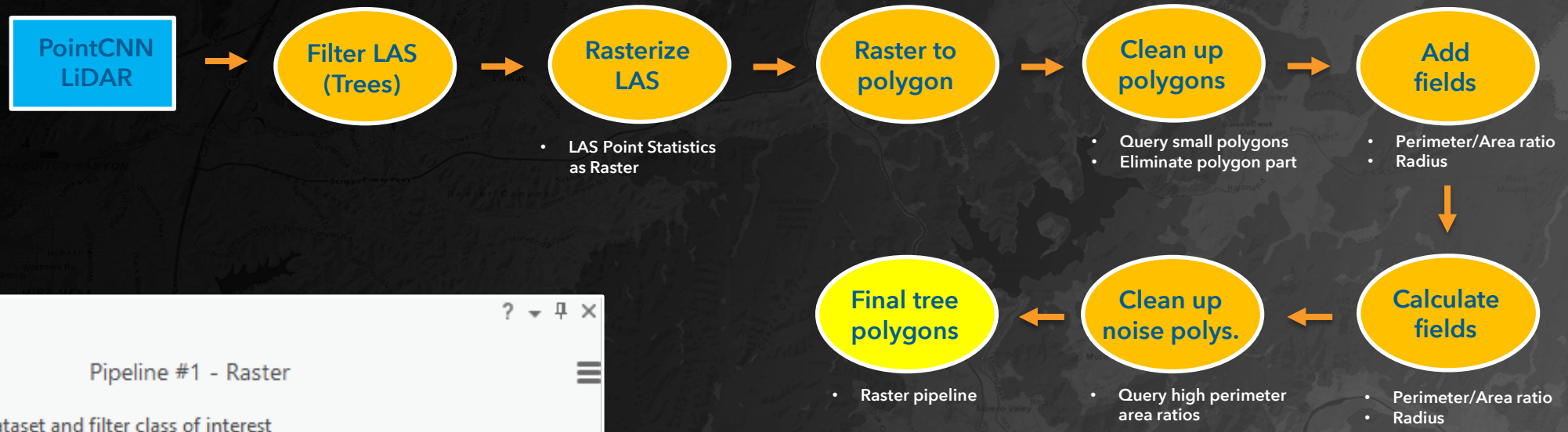


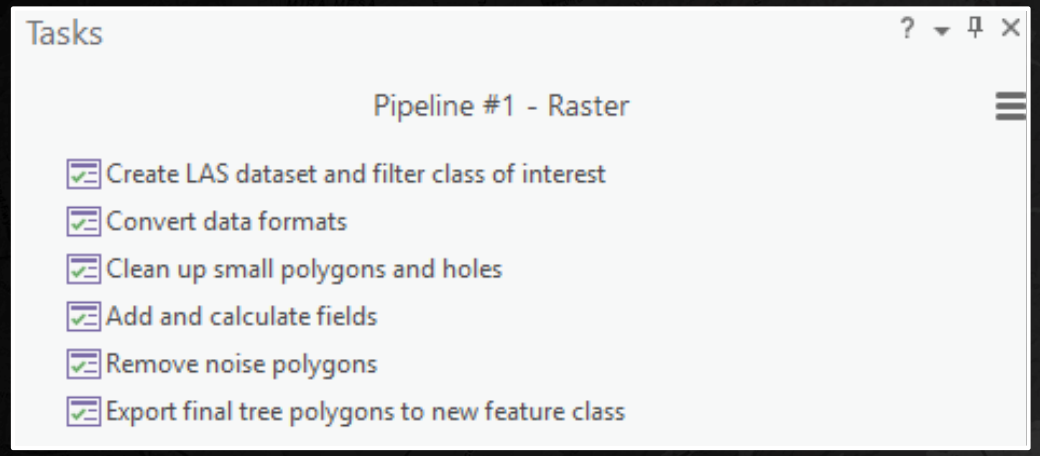






Feature extraction pipelines (#1 - Raster)



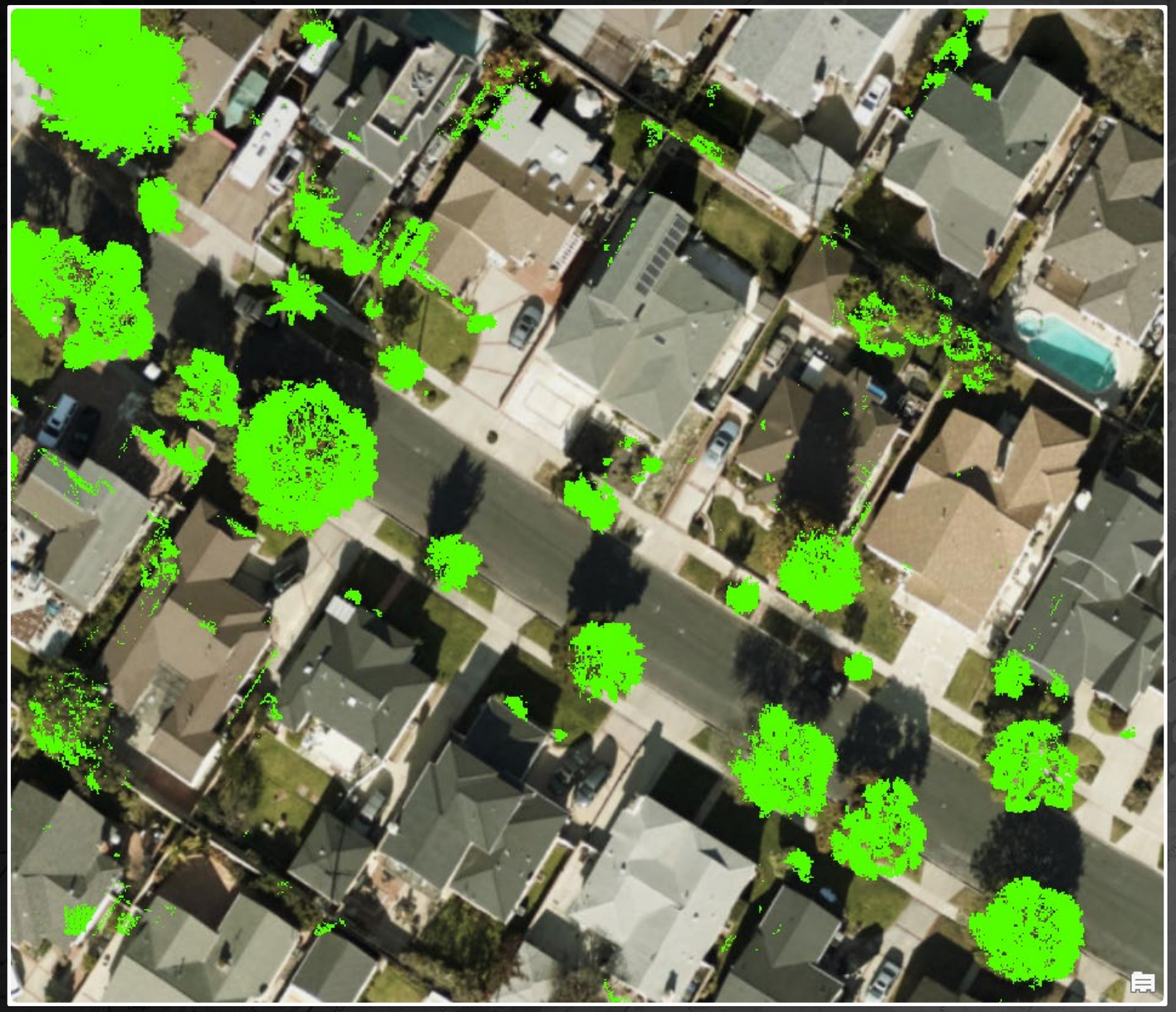


Feature extraction pipelines (#1 - Raster)

Suburban
Los Angeles

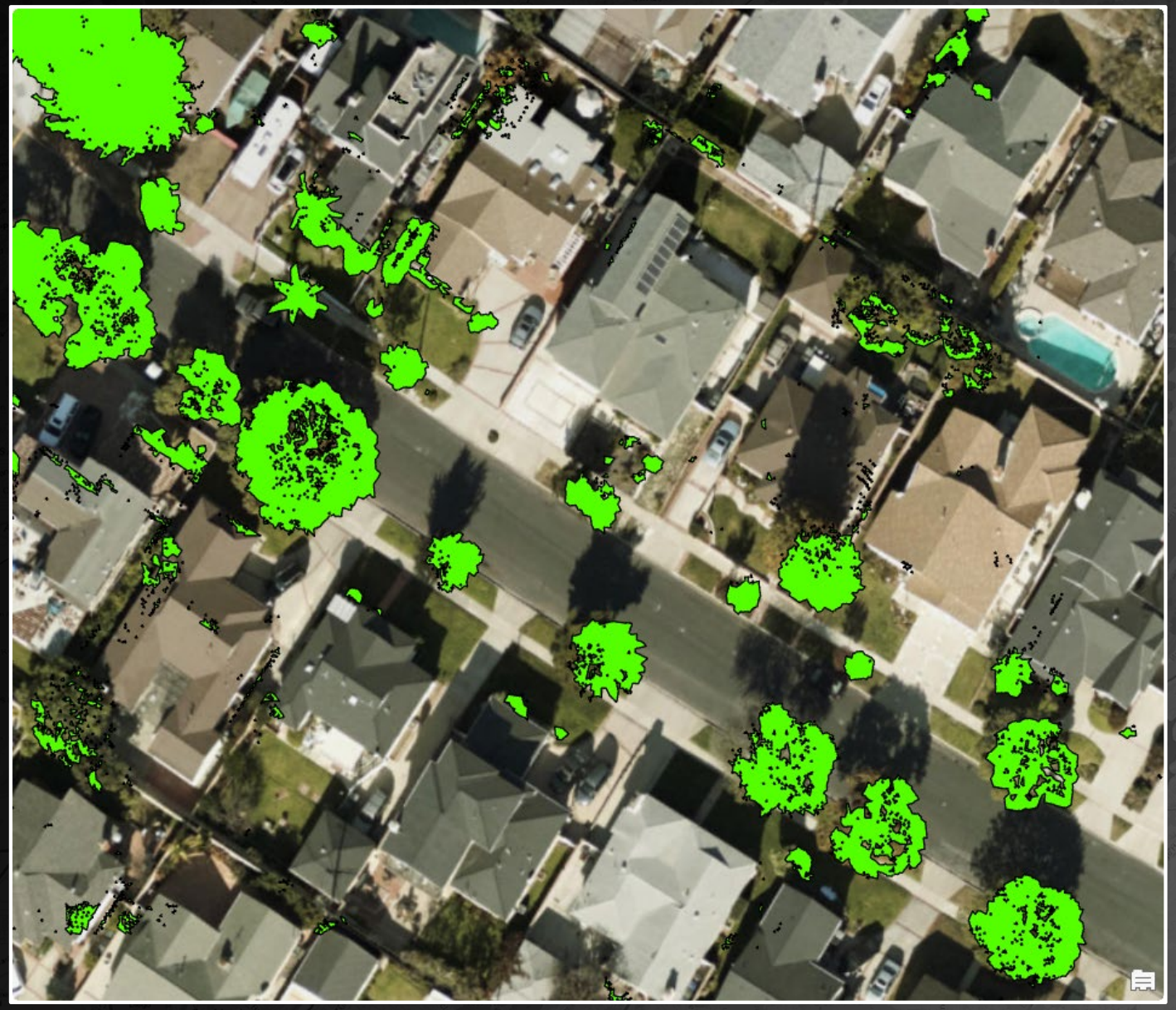
Feature extraction pipelines (#1 – Raster)

LiDAR to raster



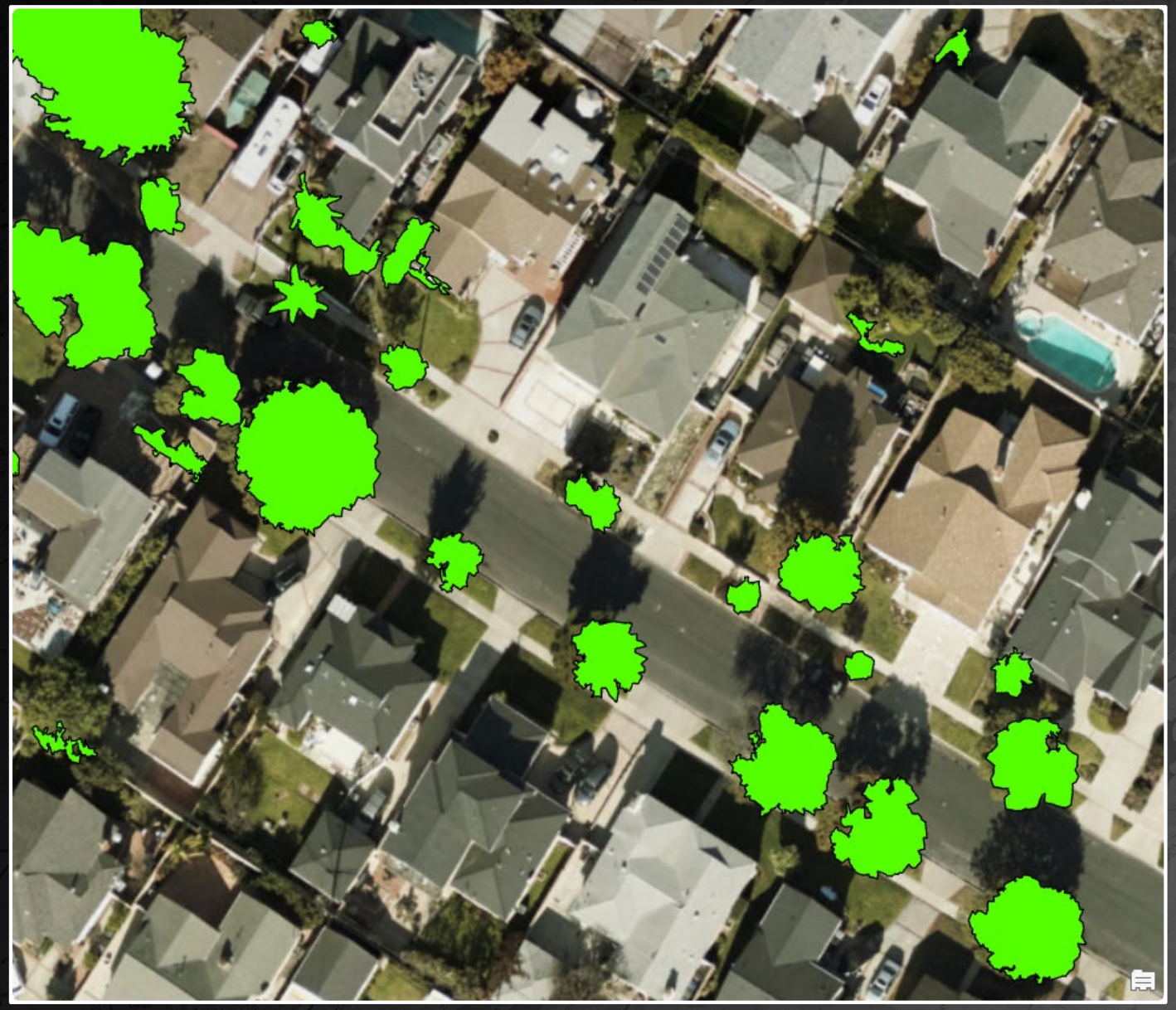
Feature extraction pipelines (#1 – Raster)

Raster to polygons



Feature extraction pipelines (#1 - Raster)

Remove small polygons, fill holes



Feature extraction pipelines (#1 - Raster)

Clean up

- Remove high perimeter/area ratios

Feature to point

Feature extraction pipelines (#2 – ML)

PointCNN LiDAR

LAS to Multipoint

Explode multipoints
- Multipart to Singlepart

Reduce dataset size
- Add field
- Calculate field (random)
- Query *n*%

DBSCAN
- 100 features minimum
- 5 foot search distance

Remove noise pts.
- Cluster ID ≠ -1

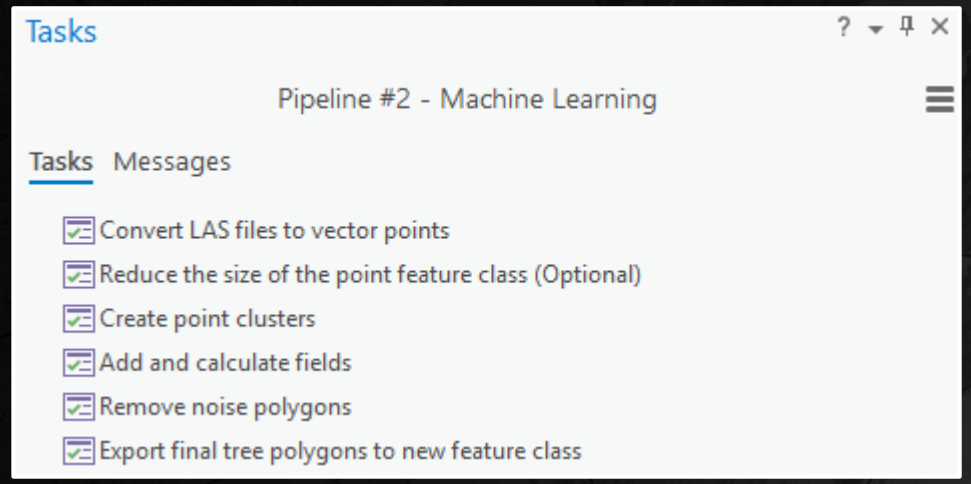
Cluster boundaries
- Minimum Bounding Geometry
- Convex hull

Add fields
- Perimeter/Area ratio
- Radius

Calculate fields
- Perimeter/Area ratio
- Radius

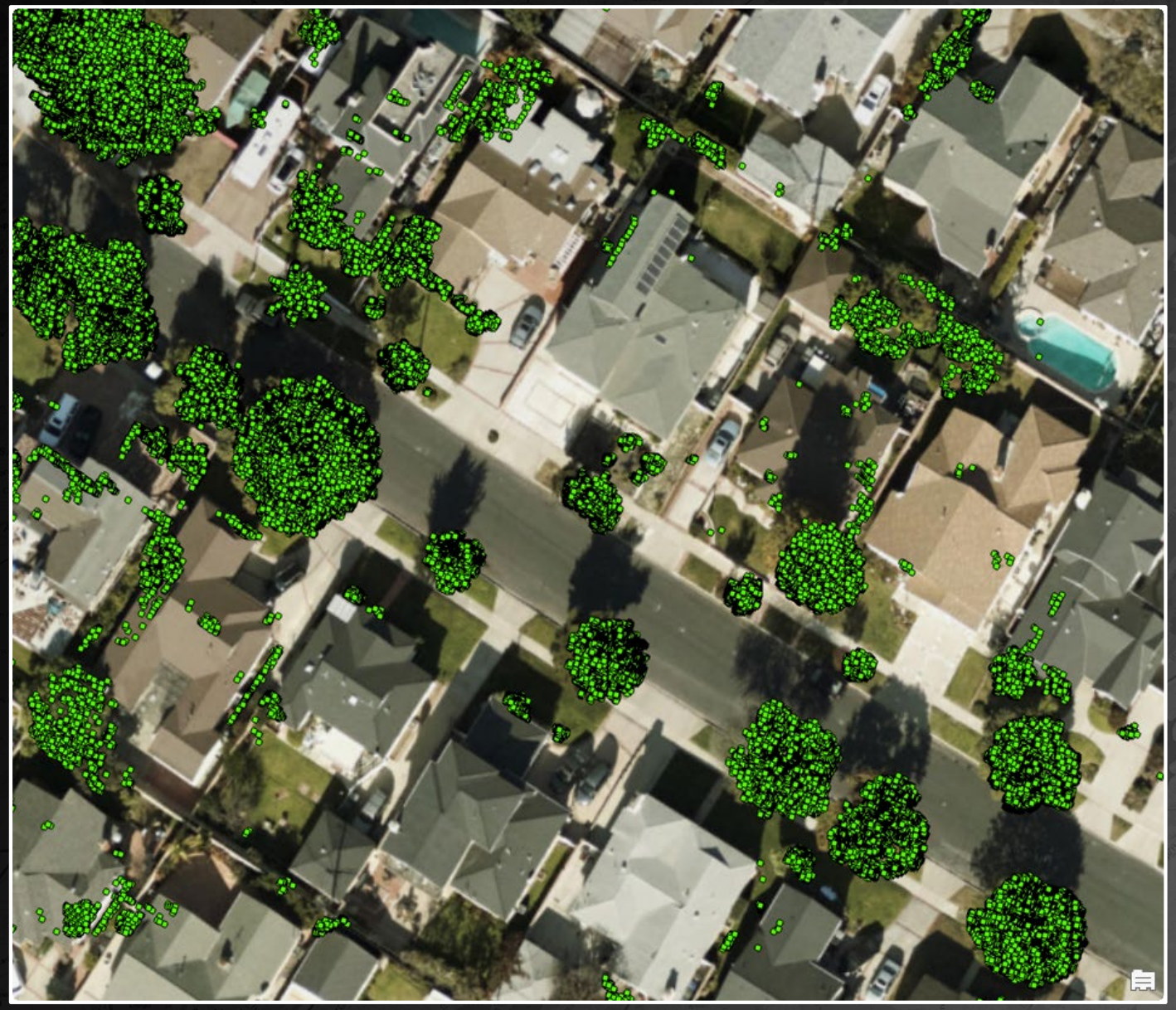
Clean up noise polys.
- Query high perimeter area ratios
- Query small polygons

Final tree polygons
- ML pipeline

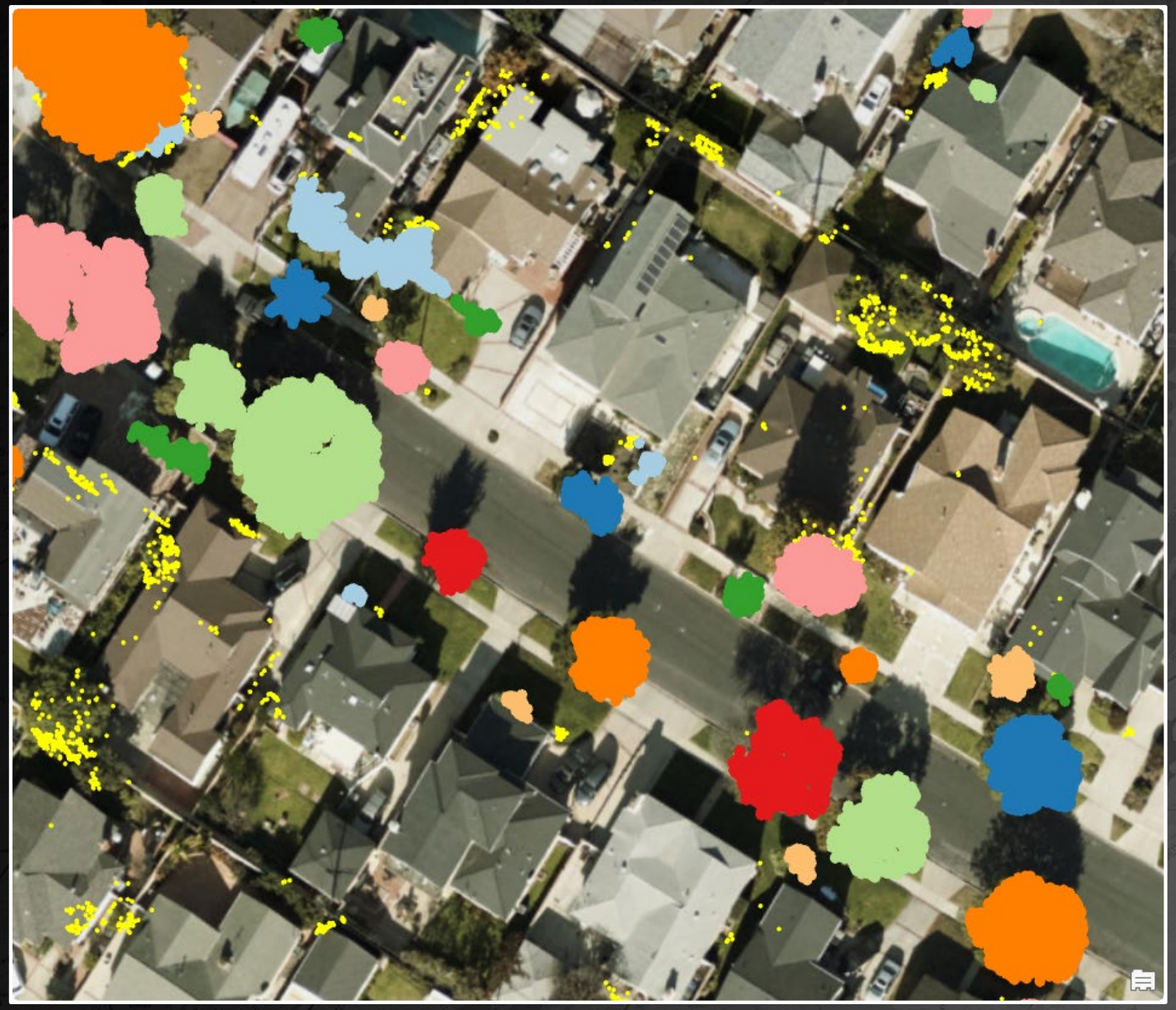


Feature extraction pipelines (#2 - ML)

LiDAR
to Multipoints



Feature extraction pipelines (#2 - ML)



DBSCAN

- Min features: 100
- Search distance: 5 ft

Feature extraction pipelines (#2 - ML)

Minimum bounding geometry

Feature extraction pipelines (#2 - ML)

Clean up

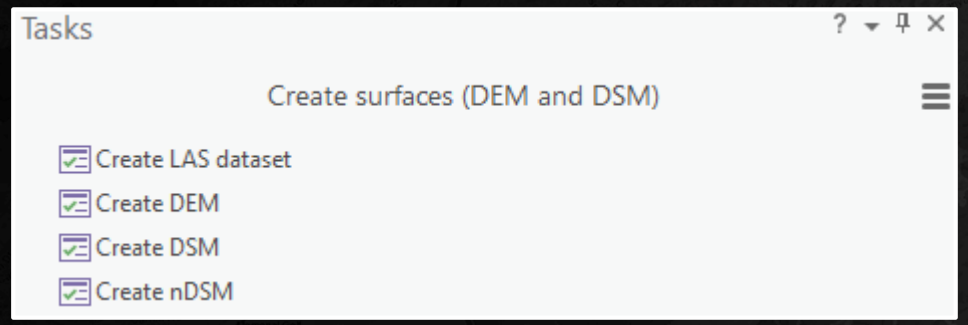
- Remove high perimeter/area ratios
- Remove small polygons

Feature to point

Feature extraction pipelines (Creating heights)

PointCNN LiDAR → Change LAS codes → Classify LAS ground → Filter LAS (ground) → Rasterize LAS → Filter LAS (Max) → Rasterize LAS → Minus → nDSM

- Change LAS codes: Class 1 (PointCNN trees) to class 5 (High vegetation)
- Classify LAS ground: Standard, Conservative
- Rasterize LAS: LAS Dataset to Raster, Digital Elevation Model (DEM)
- Rasterize LAS: LAS Dataset to Raster, Digital Surface Model (DSM)
- Minus: DSM - DEM
- nDSM: Normalized Digital Surface Model (nDSM)

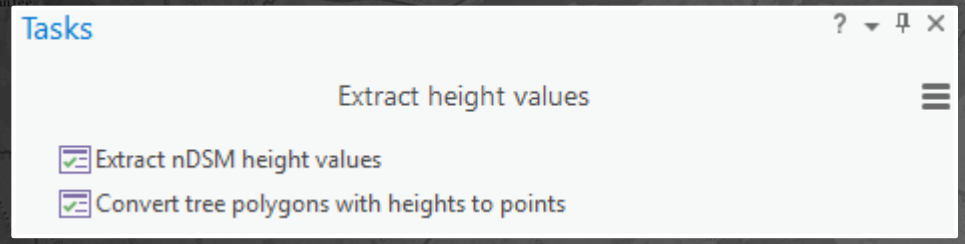


Final tree polygons → Zonal statistics → Join heights to pts. → Final tree points

- Final tree polygons: Raster pipeline
- Zonal statistics: PointCNN nDSM
- Join heights to pts.: nDSM Max
- Final tree points: Raster pipeline

Final tree polygons → Zonal statistics → Join heights to pts. → Final tree points

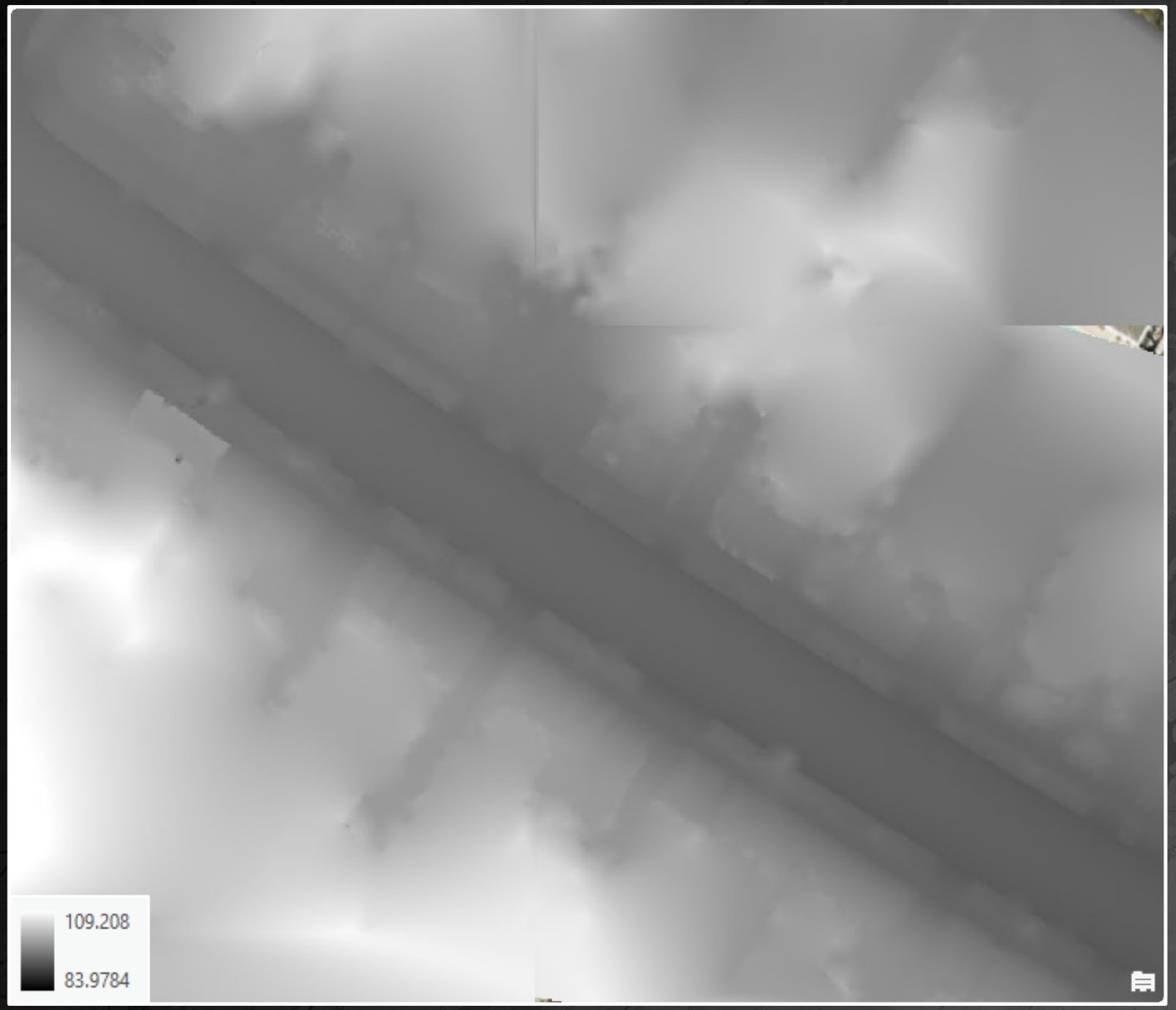
- Final tree polygons: ML pipeline
- Zonal statistics: PointCNN nDSM
- Join heights to pts.: nDSM Max
- Final tree points: ML pipeline



Feature extraction pipelines (Surfaces)

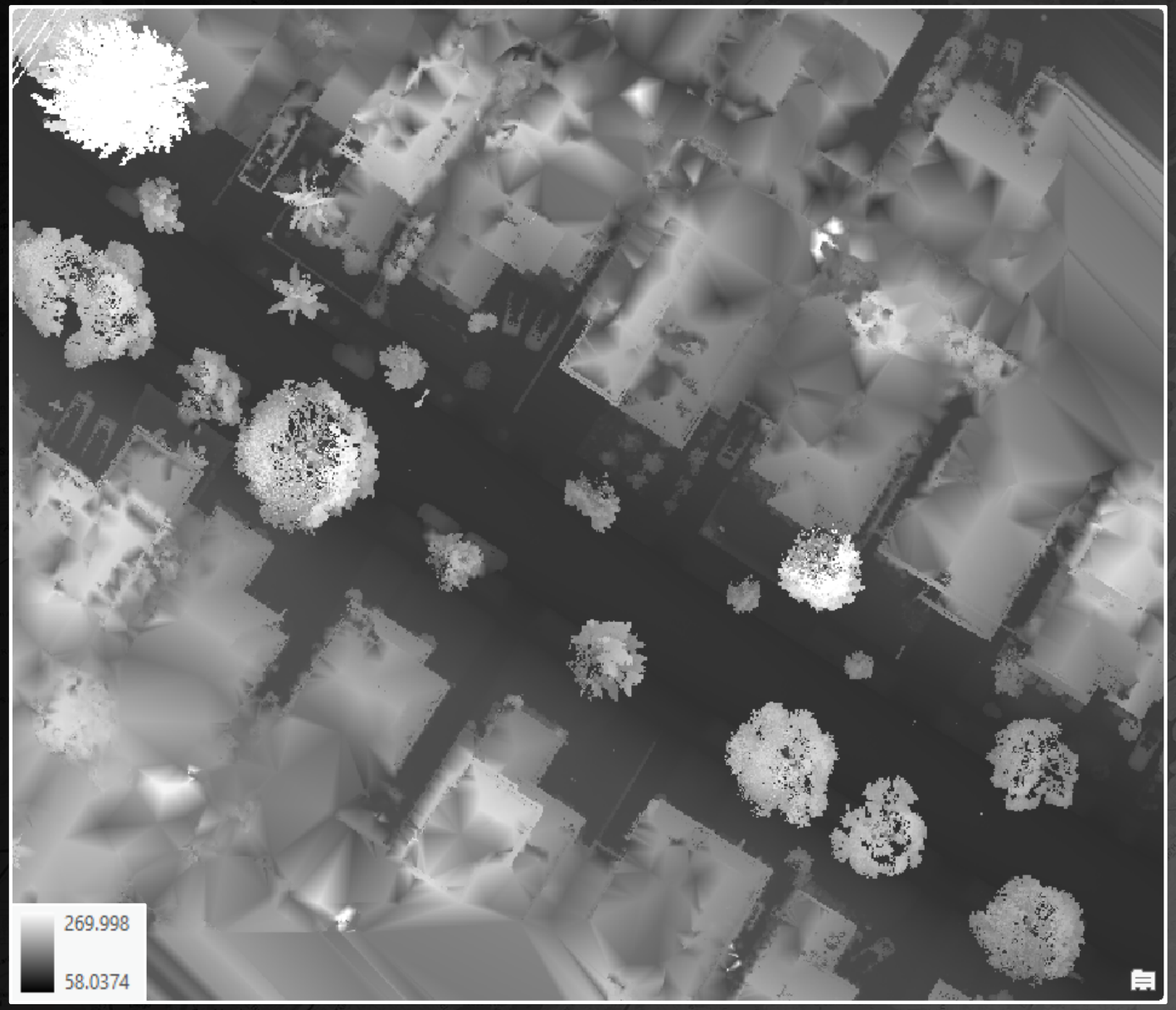
Feature extraction pipelines (Surfaces)

Digital Elevation Model (DEM)



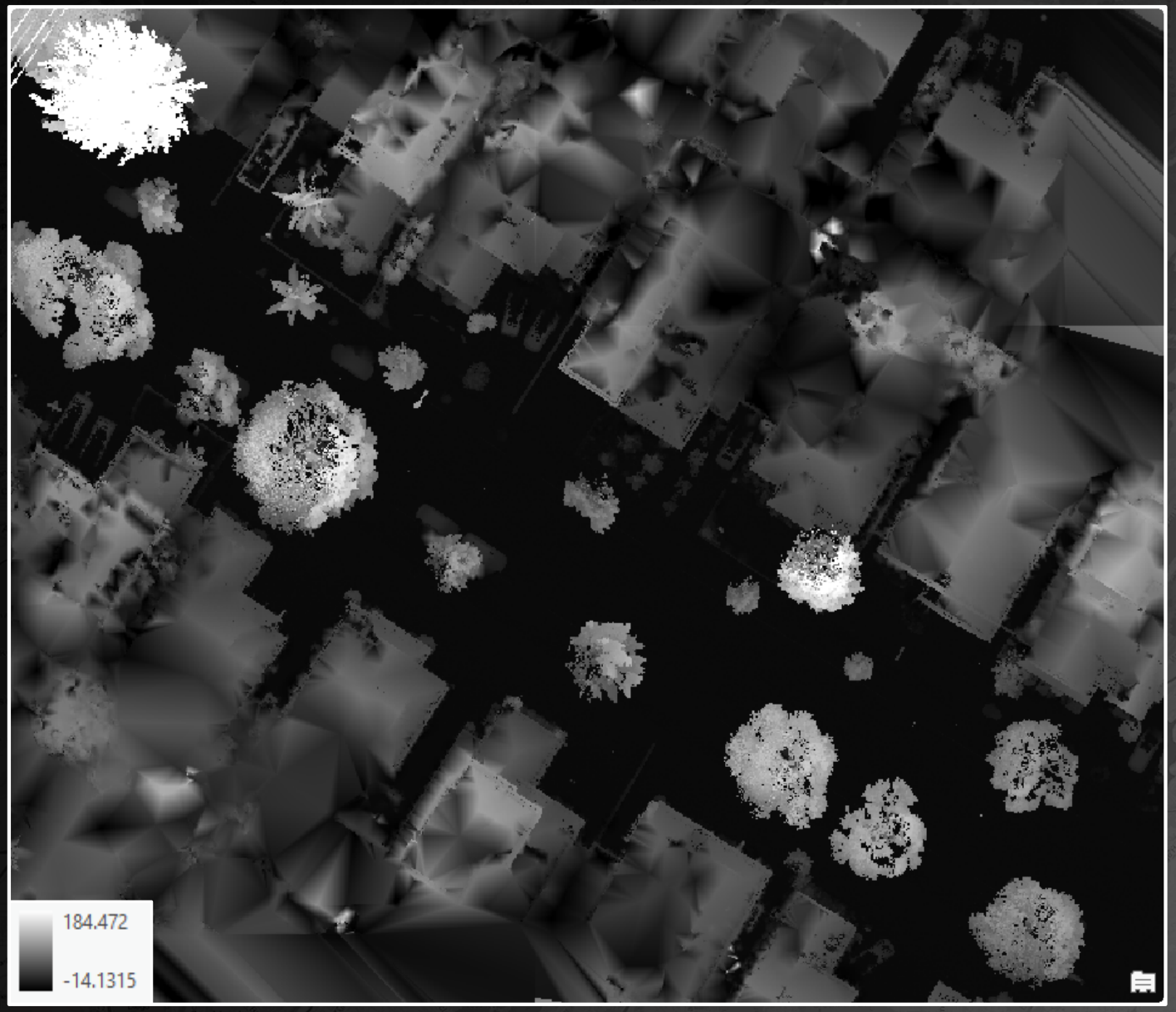
Feature extraction pipelines (Surfaces)

Digital Surface Model (DSM)

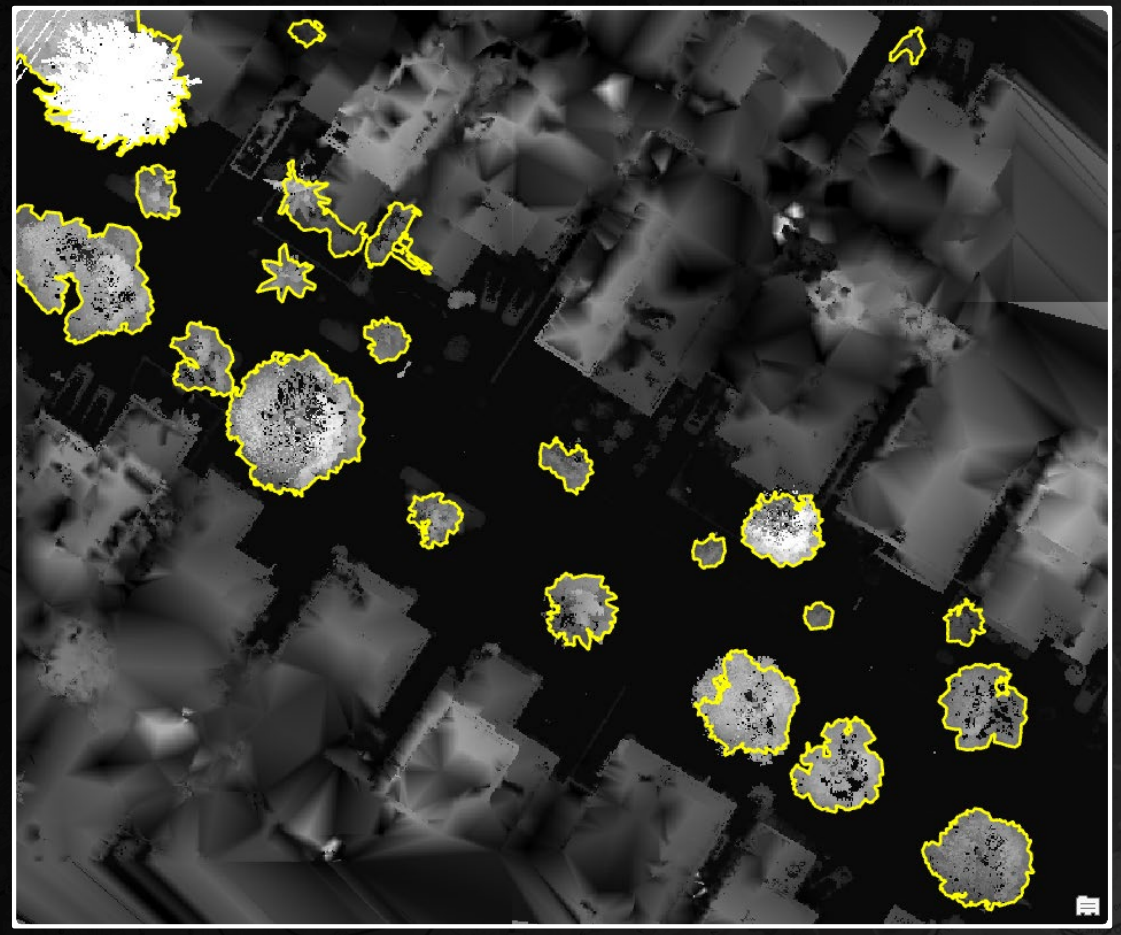


Feature extraction pipelines (Surfaces)

Normalized Digital Surface Model (nDSM)



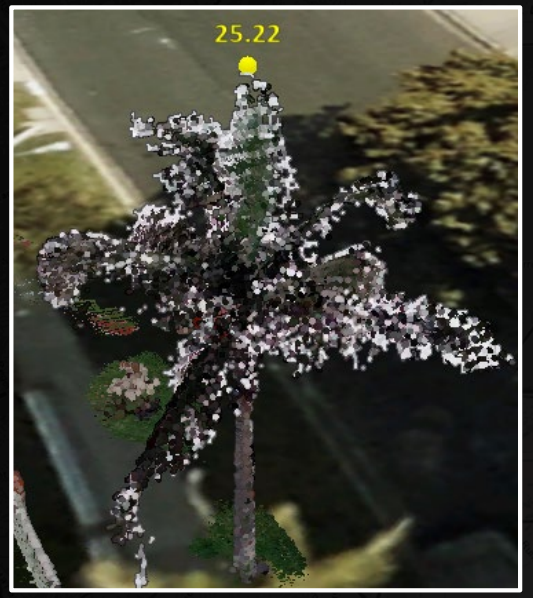
Feature extraction pipelines (Extracting heights)

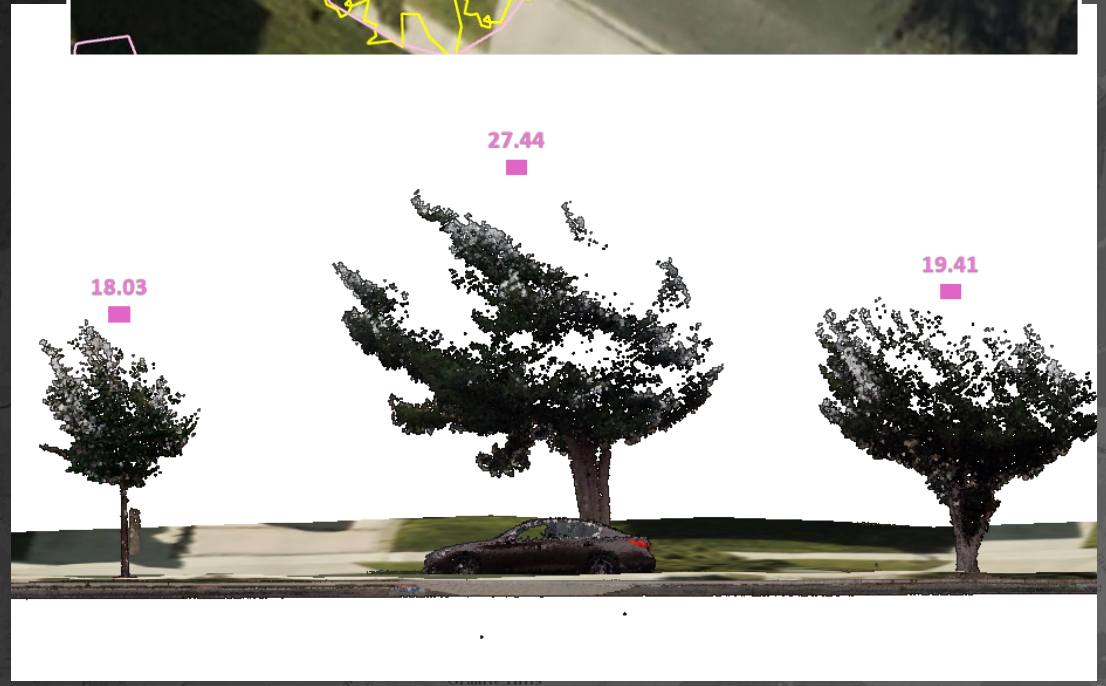


Field: Add Calculate | Selection: Zoom To Switch Clear Delete Copy

OBJECTID	Shape	Id	gridcode	ORIG_FID	PerimArea	Radius	Shape_Length	Shape_Area	MIN	MAX	MEAN
37	Polygon	2661	5	2661	0.410733	03.515994	650.41017	1583.536807	4.462326	154.2267	27.415965
9	Polygon	556	5	556	0.138295	86.932646	546.213924	3949.637901	0.013176	109.498703	39.667253
19	Polygon	1082	5	1082	0.693453	12.51784	78.65191	113.420674	0.003136	76.517021	66.666903
73	Polygon	5680	5	5680	0.747342	13.868811	87.140311	116.600382	-0.350594	47.808449	30.809626
61	Polygon	4796	5	4796	0.619071	63.564123	399.385163	645.13616	-0.002556	46.717812	30.278498
48	Polygon	4111	5	4111	0.288888	27.710872	802.431075	2777.651806	-0.396576	43.958366	16.804566
26	Polygon	1681	5	1681	0.251077	07.481045	675.323325	2689.701185	-0.33046	43.733139	21.36026
2	Polygon	115	5	115	0.229437	34.146386	214.548068	935.108598	-0.010002	43.346733	20.196509
69	Polygon	5540	5	5540	0.377664	49.883333	313.426225	829.906863	-0.047104	42.973679	26.045426
27	Polygon	1685	5	1685	0.277736	21.643755	135.991725	489.6445	-0.196877	42.370506	25.789876
23	Polygon	1505	5	1505	0.385087	45.064626	283.149397	735.287419	0.002182	40.509964	27.601931
68	Polygon	5159	5	5159	0.402617	73.061744	459.060475	1140.190739	-0.042221	39.826958	21.757074
43	Polygon	3526	5	3526	0.40172	64.971799	408.22985	1016.205348	-0.011345	38.565872	17.673984
4	Polygon	209	5	209	0.60037	48.334549	303.694931	505.846151	-1.052994	37.414551	17.958374

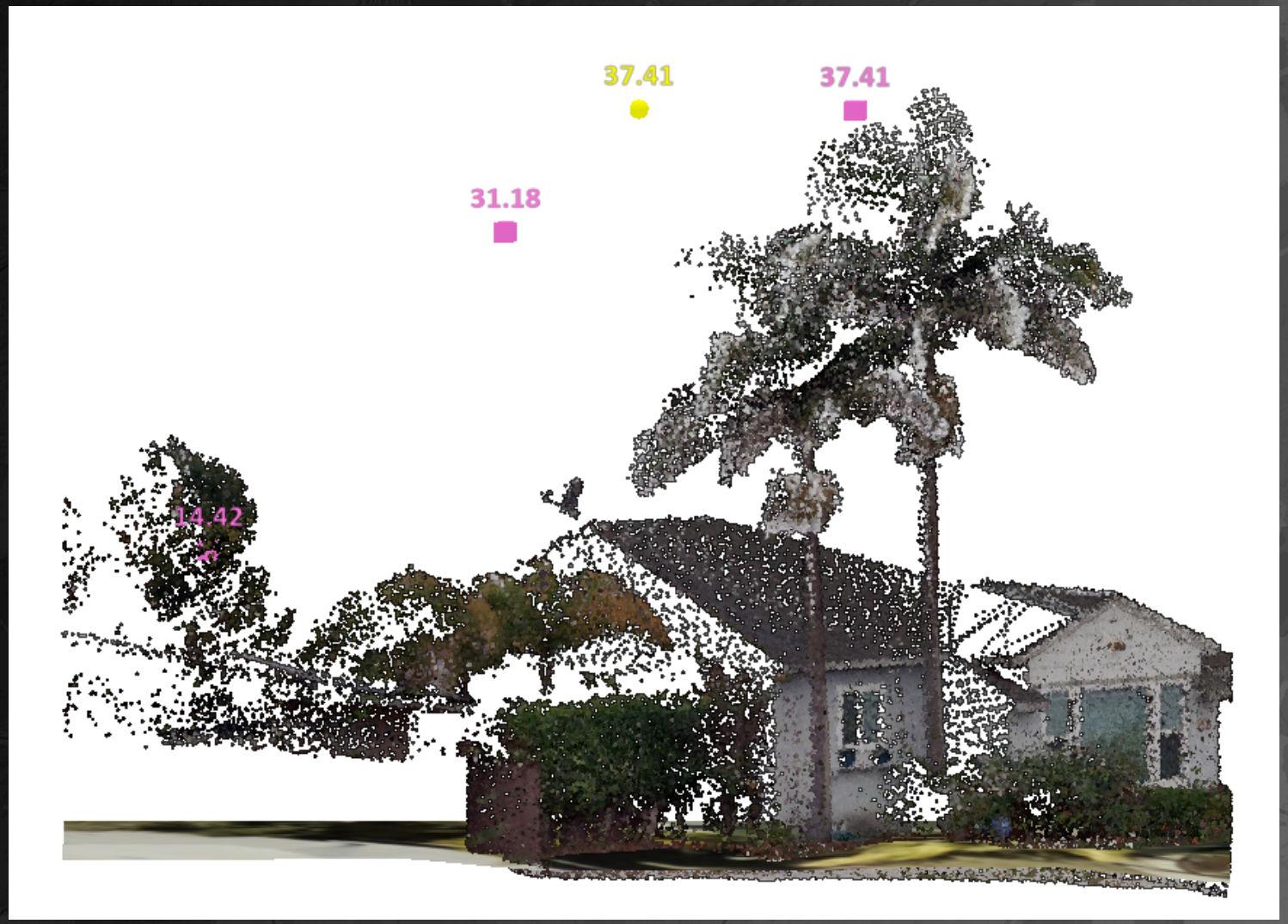
Results (2D and 3D)





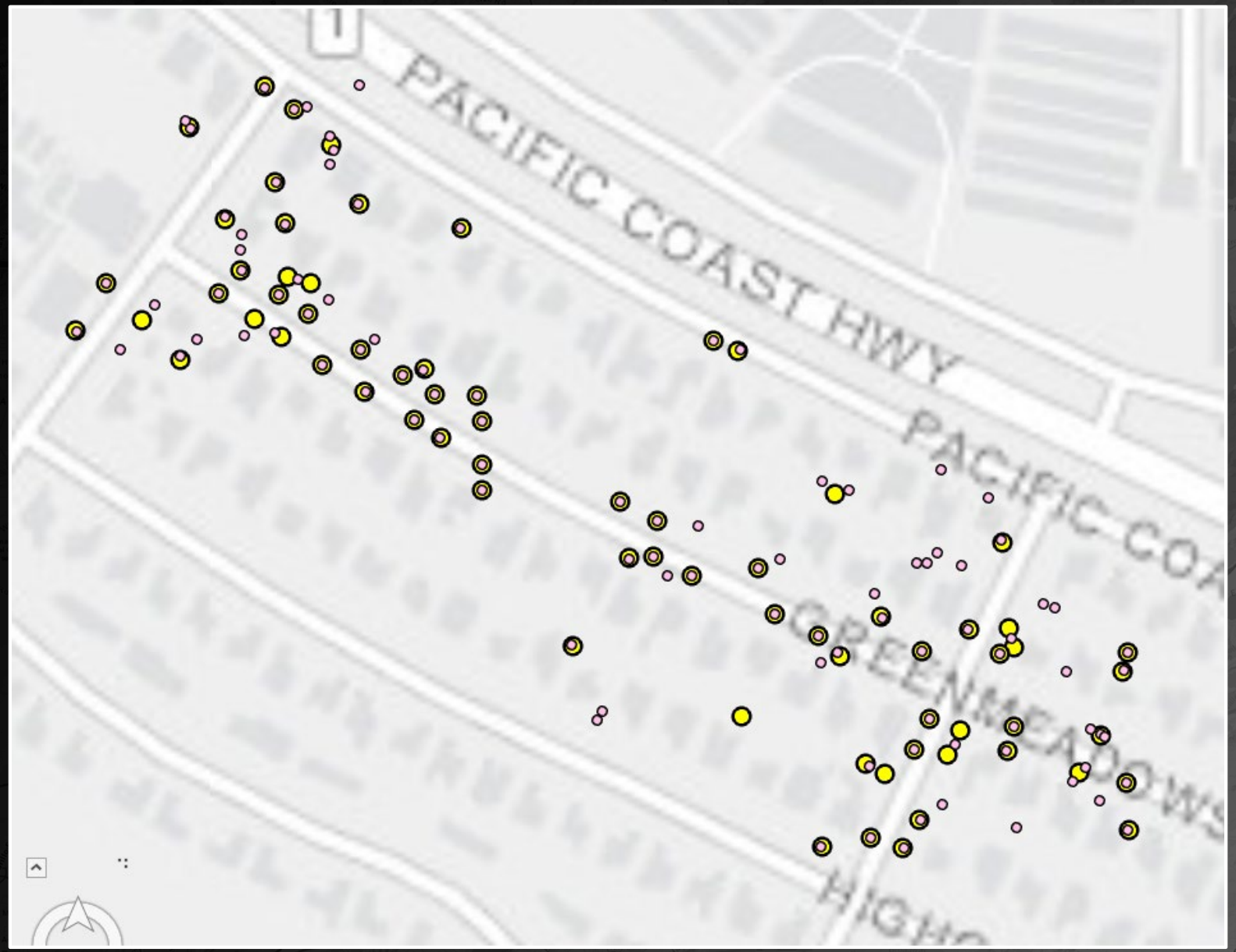
Results (2D and 3D)

Results (2D and 3D)



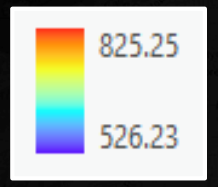
Results

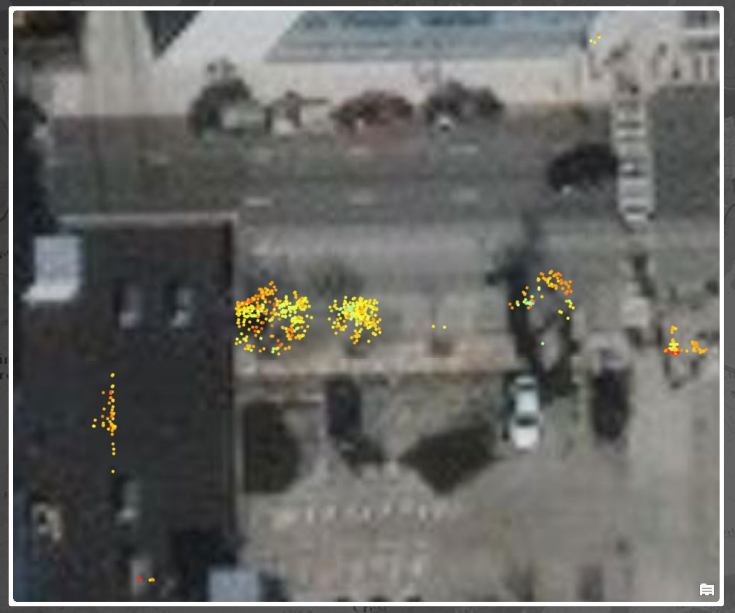
- #1 - Raster
- #2 - ML



Chicago data – Trees – overfitting for season

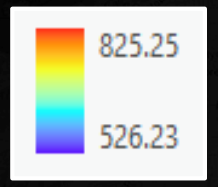
- Number of LAS files: 24
- Number of LAS points: 60 million
- XYZ units: US foot
- Average point spacing: 0.26 feet
- Classes
  - 0 = Other
  - 1 = Trees

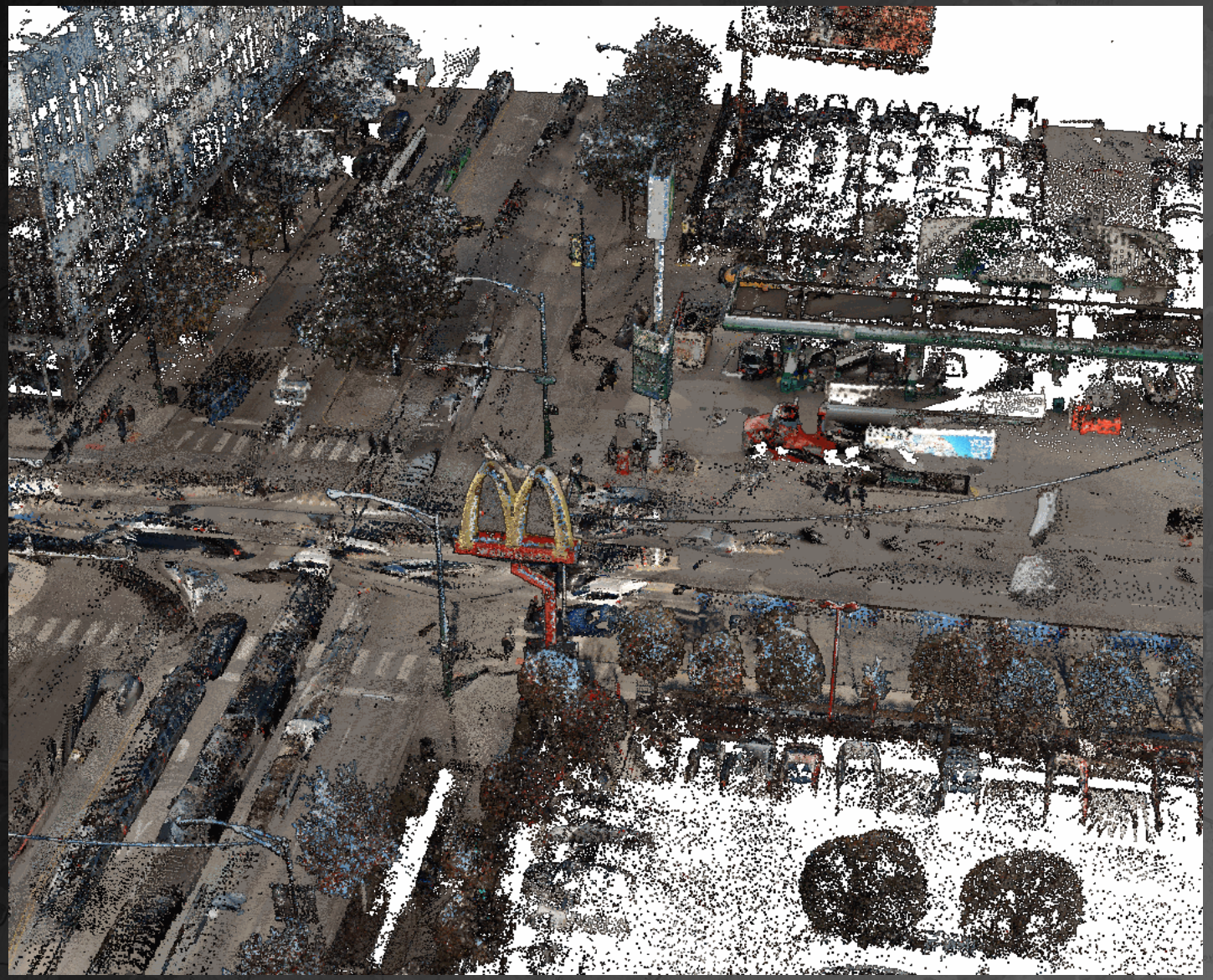




Chicago data – Trees – overfitting for season

- Number of LAS files: 24
- Number of LAS points: 60 million
- XYZ units: US foot
- Average point spacing: 0.26 feet
- Classes
  - 0 = Other
  - 1 = Trees



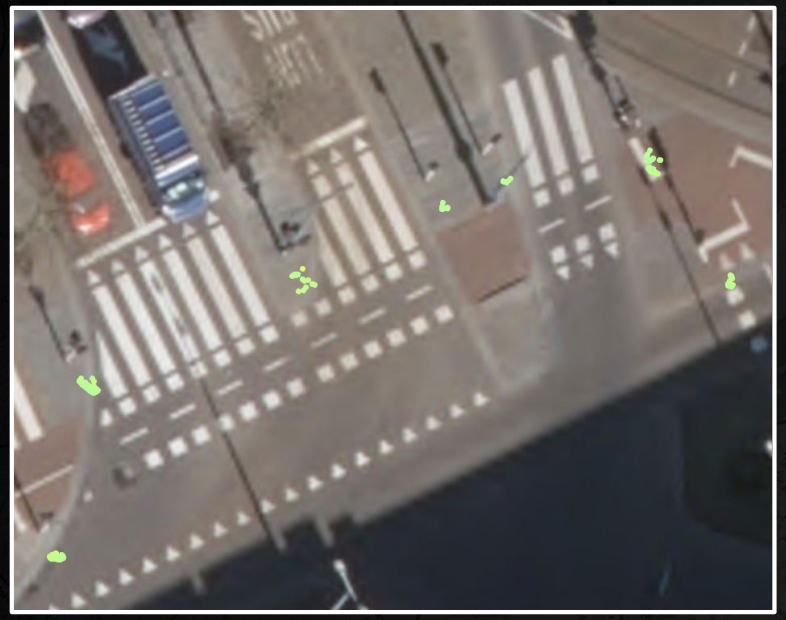



CycloMedia-sample - Scene-Class - ArcGIS Pro
Point Cloud Layer
Project
Map
Insert
Analysis
View
Edit
Imagery
Share
Appearance
Dmitry (Конференция Esri в России и странах СНГ)
Scene-RGB
Scene-Class
Contents
Tasks
Catalog
Pop-up
Symbology
Geoprocessing
Bookmarks
Feature Extraction:
Traffic Lights
64 ft
Selected Features: 0
4.4023086°E 51.9145364°N
10.018 ft

Netherlands data – Traffic lights

Schiedam

- Number of LAS files: 12
- Number of LAS points: 42.3 million
- XYZ units: Meters
- Average point spacing: 0.03 meters
- Classes
  - 0 = Other
  - 1 = Traffic lights



Amsterdam

- Number of LAS files: 35
- Number of LAS points: 136.6 million
- XYZ units: Meters
- Average point spacing: 0.04 meters
- Classes
  - 0 = Other
  - 1 = Traffic lights

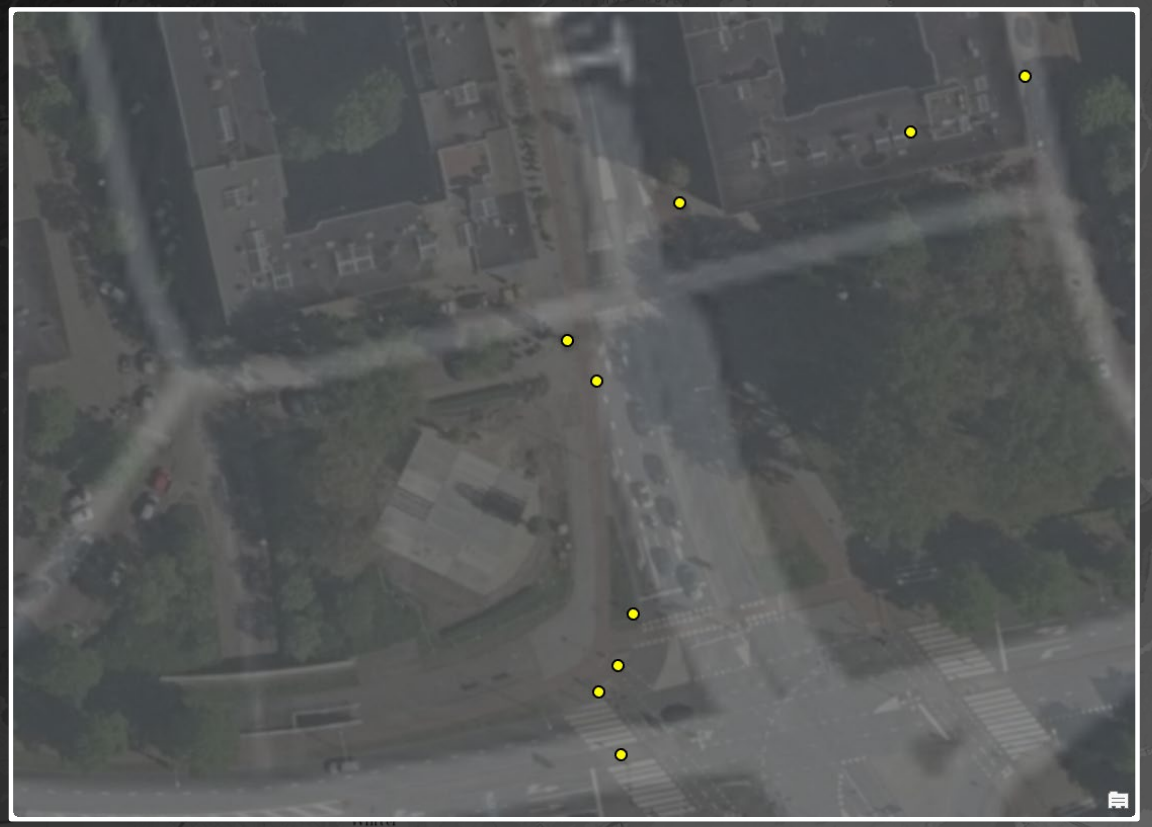
Rijnstraat, Amsterdam

- Number of LAS files: 45
- Number of LAS points: 188.6 million
- XYZ units: Meters
- Average point spacing: 0.03 meters
- Classes
  - 0 = Other
  - 1 = Traffic lights

Feature extraction pipelines (#2 - ML)

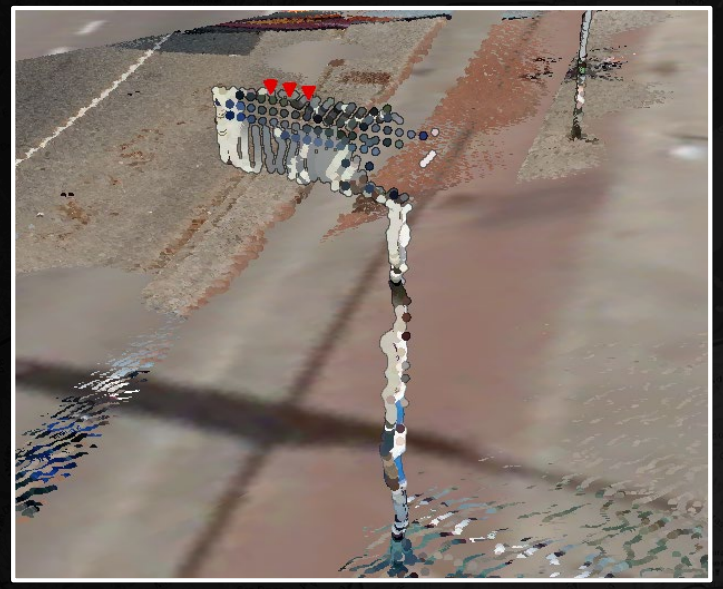
Schiedam

Amsterdam



Schiedam – Traffic lights (3D)

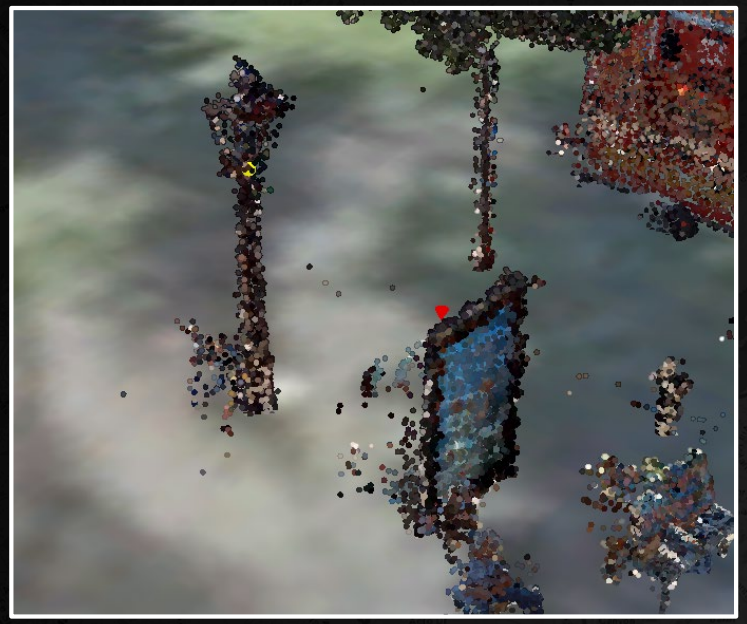
#1 - Traffic light
#2 - Noise



Amsterdam – Traffic lights (3D)

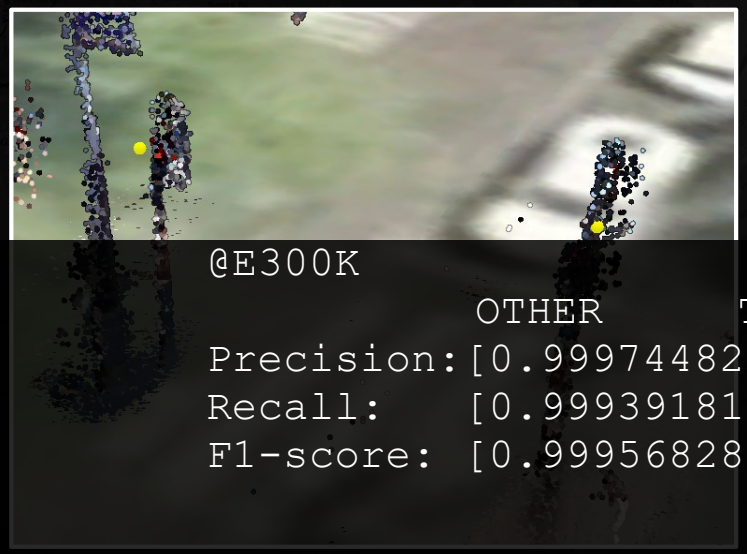
#1 - Traffic light
#2 - Noise



Amsterdam - Traffic lights (3D)

#1 - Traffic light
#2 - Noise



@E300K

	OTHER	TRAFFIC LIGHT
Precision:	[0.99974482	0.51987179],
Recall:	[0.99939181	0.72079467],
F1-score:	[0.99956828	0.60406375]

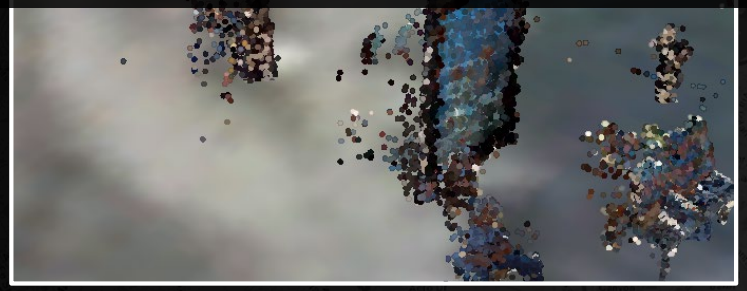
Extracted Objects, after DBScan:

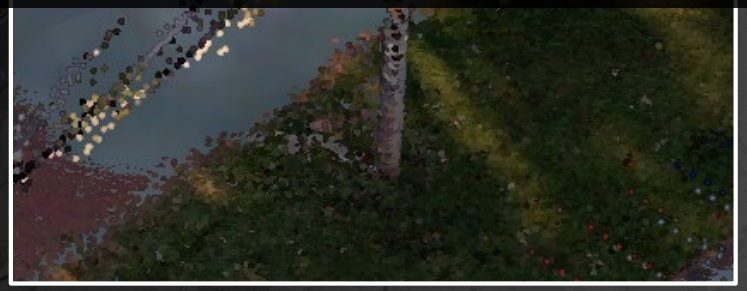
DBSCAN 30pts/0.5m

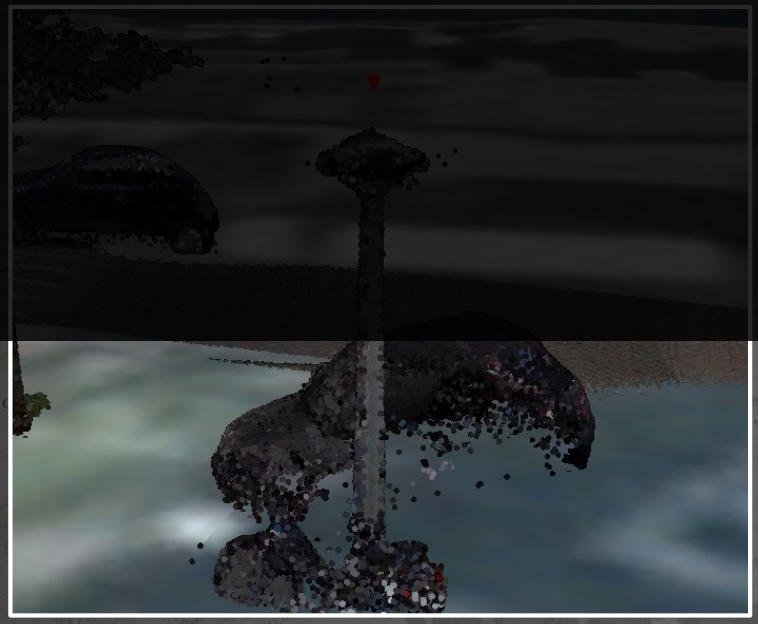
Precision = .655

Recall = .704

F1-Score = .679







Want to know more?

esri | THE SCIENCE OF WHERE

medium.com/geoai

- Reconstructing 3D buildings from aerial LiDAR with AI
- 3D cities: Deep Learning in three-dimensional space
- PointCNN: replacing 50,000 man hours with AI
- Object extraction from Mobile LiDAR point clouds with Machine Learning
- …and much more