Semantic segmentation of point clouds using deep learning

Use of AI (machine learning) in Geospatial Domain

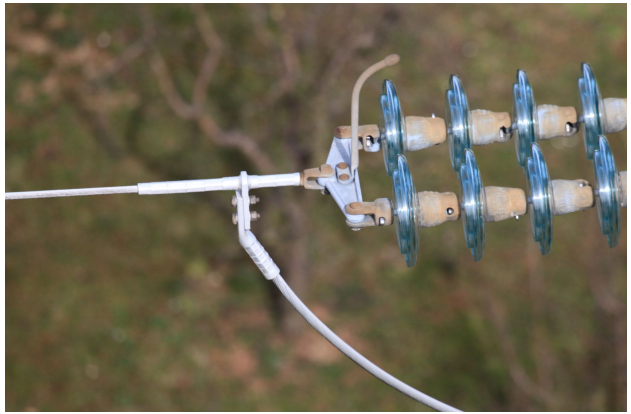
Nejc Dougan

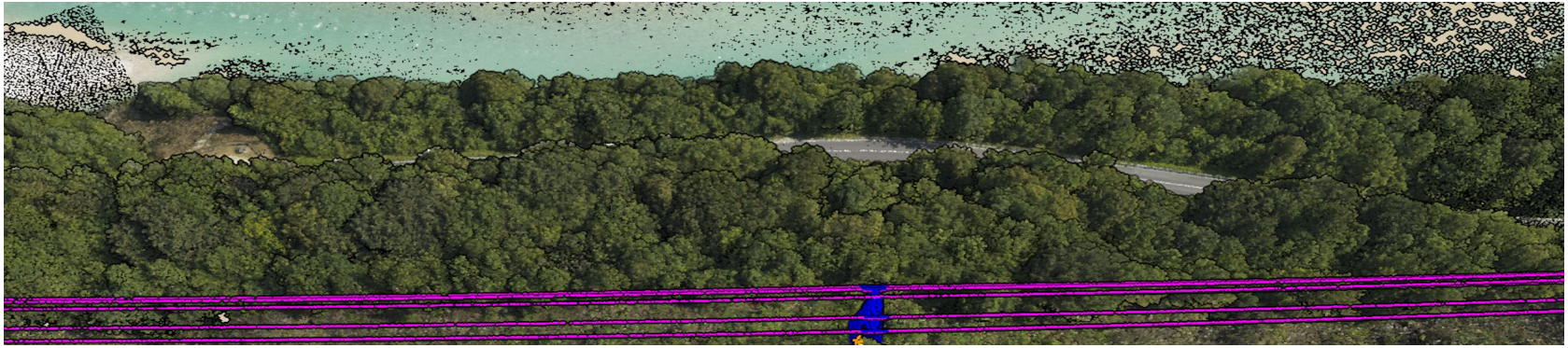
Ljubljana, 22.10.2019

About what we will be talking today

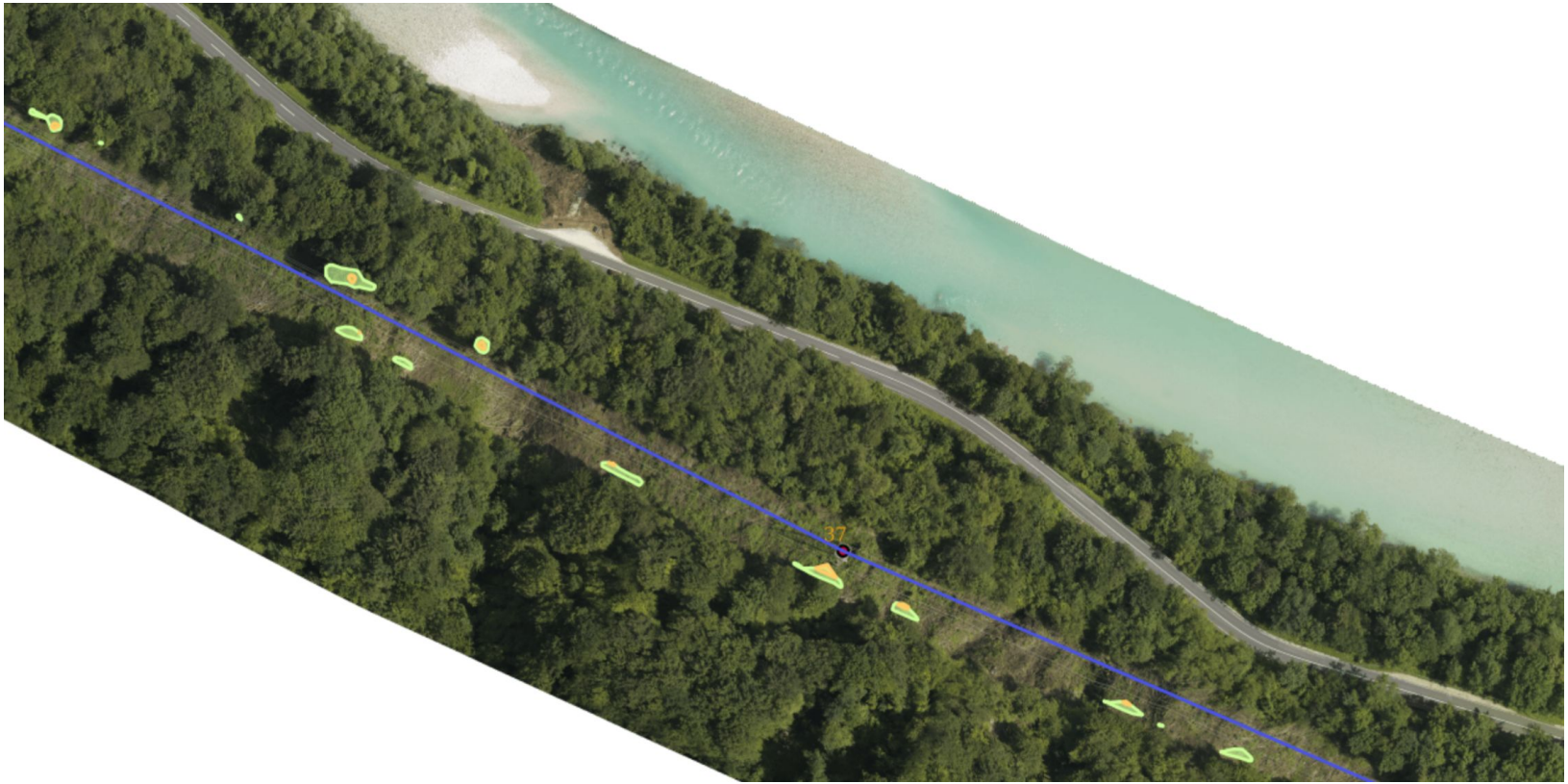
- Why we even bother
- What are point clouds
- How it was done in past
- How it will be done in future :)
- Mini demo

Why





Structured data



Users love spreadsheets and DB

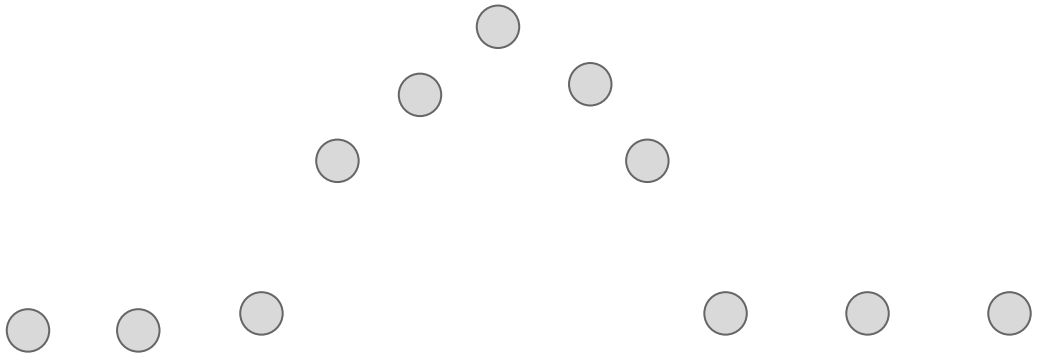
Vegetation high risk (0 - 3m) 58

#	Powerline ID	Date of inspection	Area [m2]	Zone
17	165	08.07.2016	0.85	3
18	165	08.07.2016	2.33	3
19	165	08.07.2016	0.44	3
20	165	08.07.2016	26.79	3
21	165	08.07.2016	18.79	3
22	165	08.07.2016	2.00	3
23	165	08.07.2016	8.18	3
24	165	08.07.2016	1.86	3
25	165	08.07.2016	2.40	3
26	165	08.07.2016	15.03	3
27	165	08.07.2016	0.14	3
28	165	08.07.2016	9.60	3
29	165	08.07.2016	0.08	3
30	165	08.07.2016	15.86	3
31	165	08.07.2016	0.03	3
32	165	08.07.2016	0.12	3
33	165	08.07.2016	2.33	3

Point

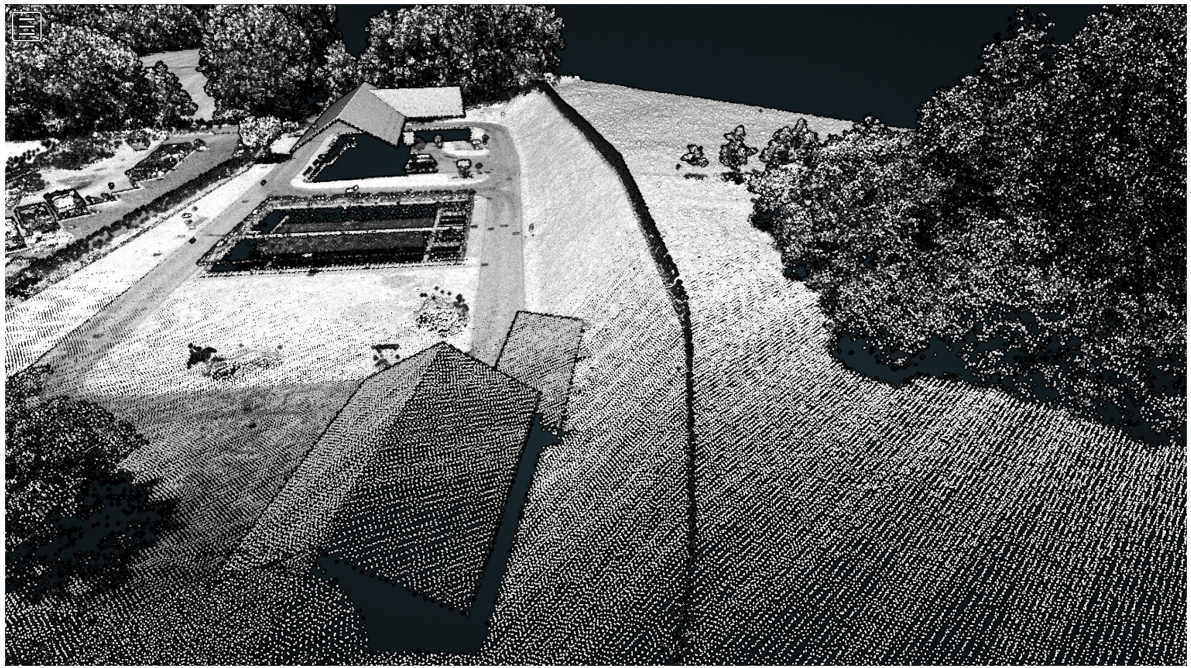
$P_i = (x, y, z)$

Point Cloud

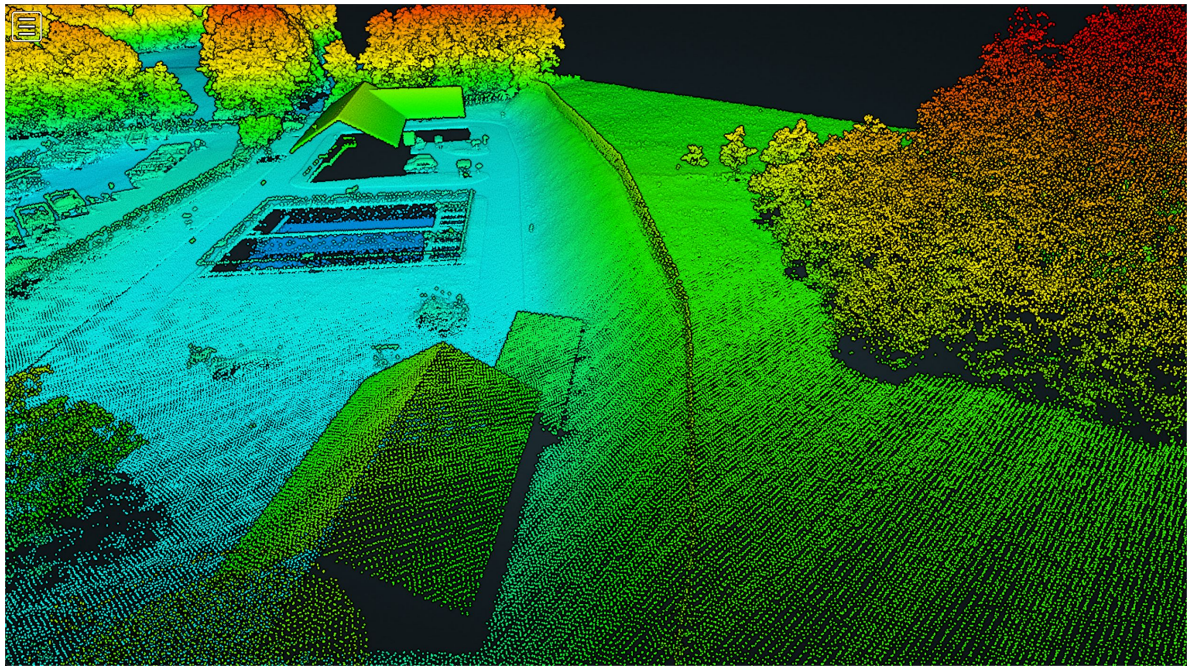


Point Cloud = $[P_1, P_2, \dots, P_n]$

What we know about each point



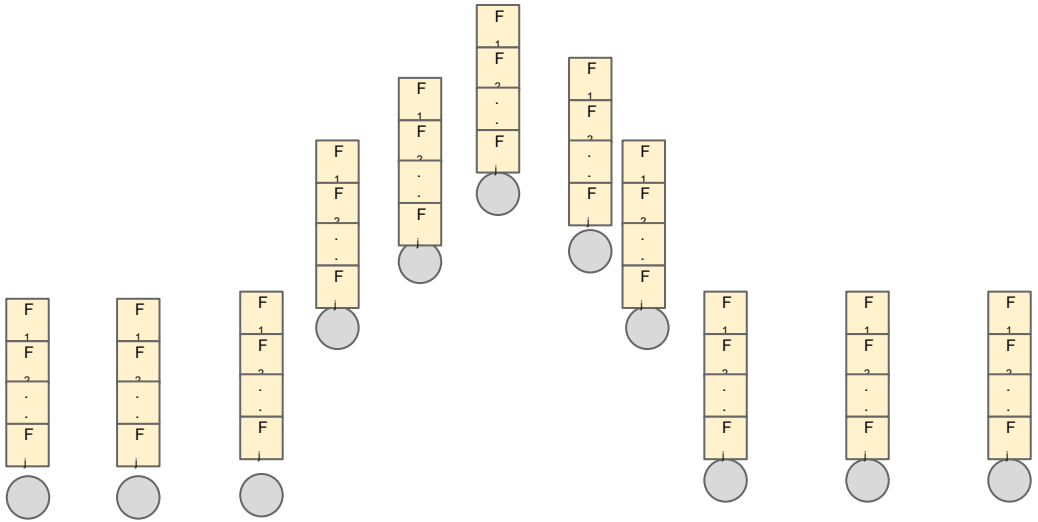
What we know about each point



What we know about each point

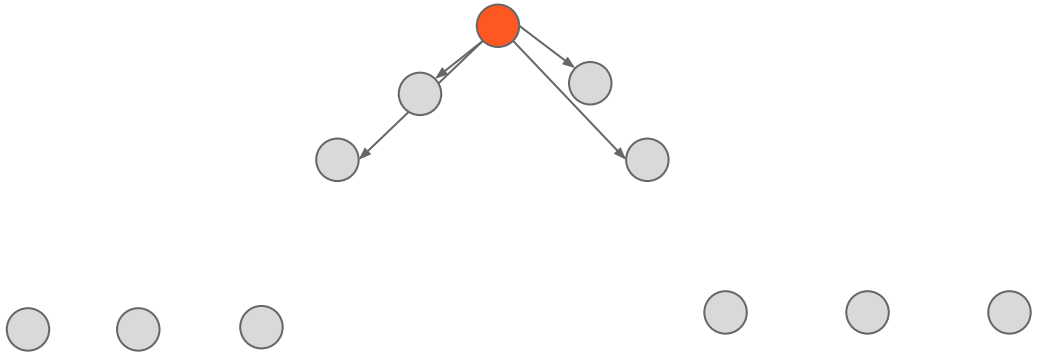
Features = $[F_1, F_2, \ldots, F_j]$

Features



Features = [F1, F2, … , Fj]

Show me who your friends are, and I'll tell you who you are

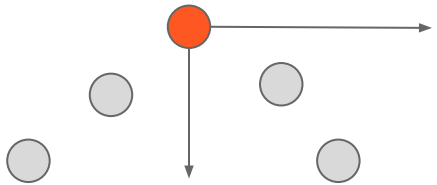


Normal, PCA, eigenvalue decomposition

Eigenvalues

$\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$

$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$



$$Linearity : L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$$

$$Planarity : P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$$

$$Scattering : S_\lambda = \frac{\lambda_3}{\lambda_1}$$

$$Omnivariance : O_\lambda = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$$

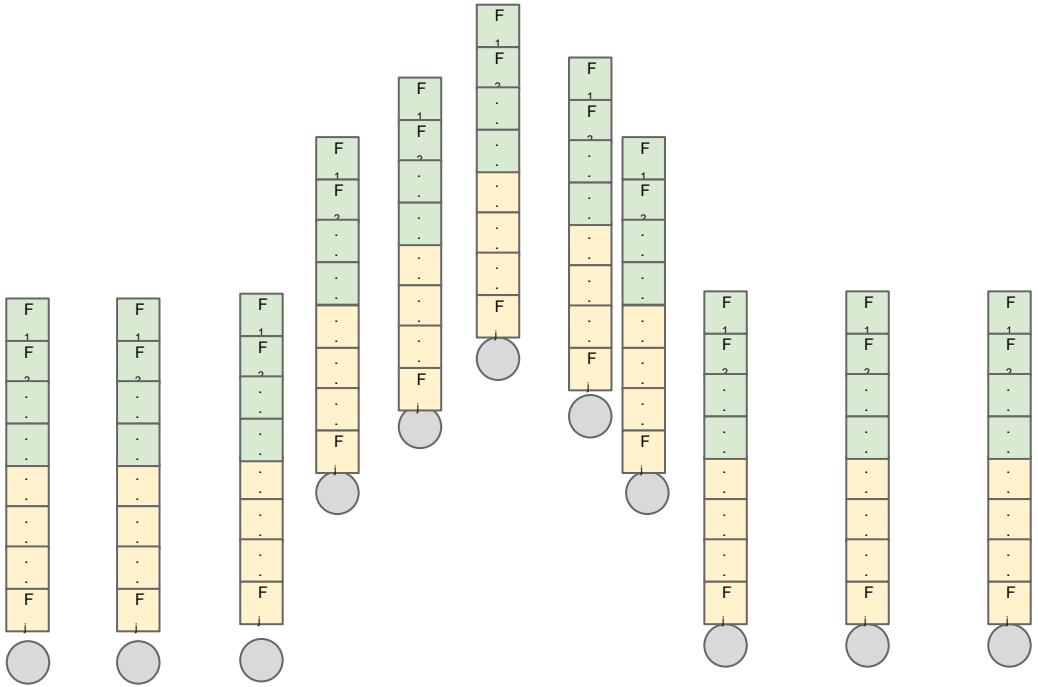
$$Anisotropy : A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$$

$$Eigentropy : E_\lambda = \sum_{i=n}^{3} \lambda_i \, ln(\lambda_i)$$

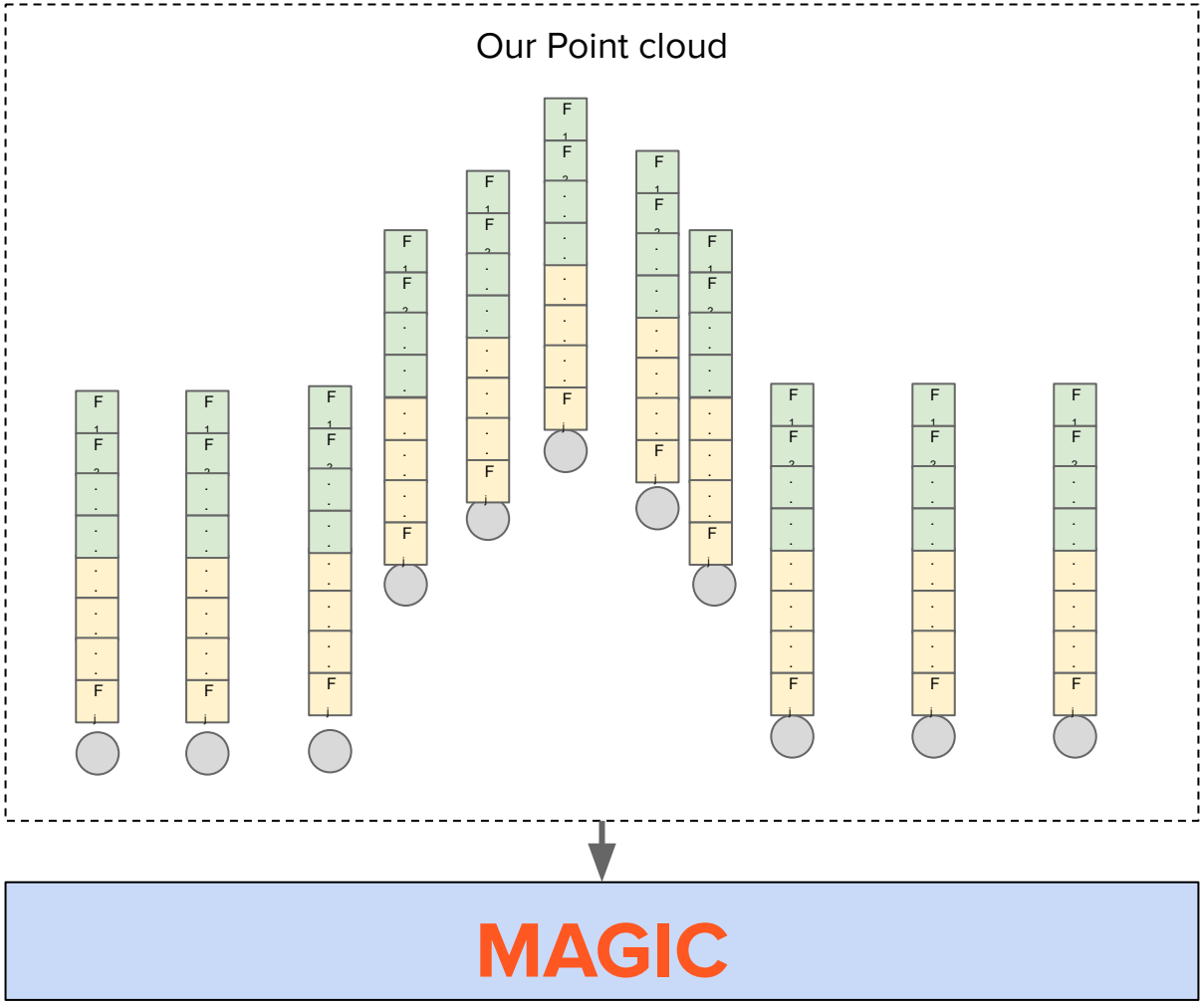
$$Sum\ of\ eigenvalues : \Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3$$

$$Change\ of\ curvature : C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

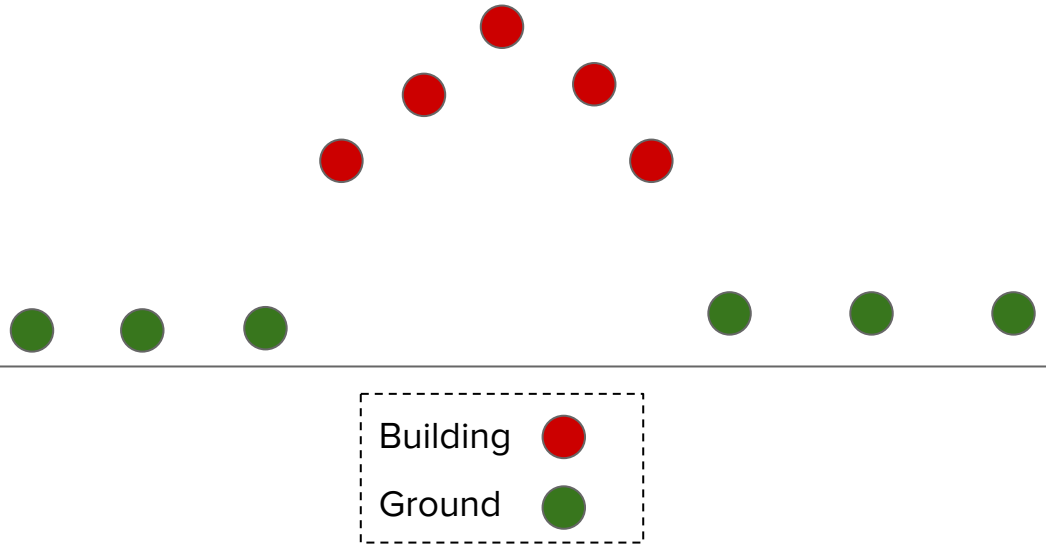
Features

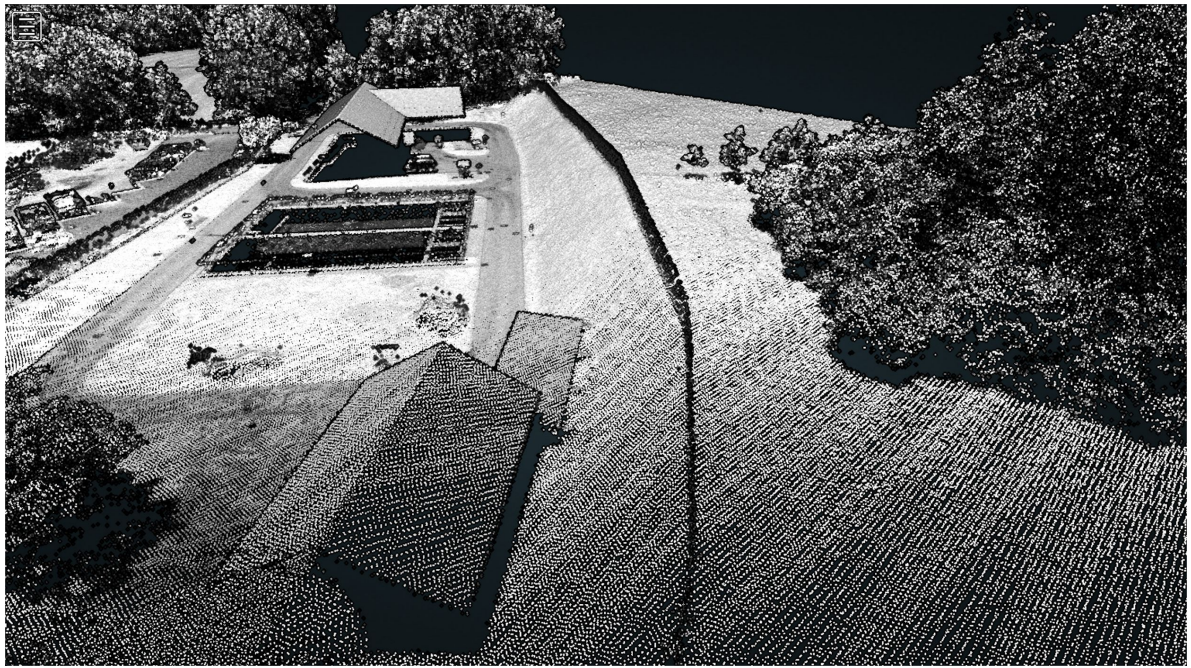


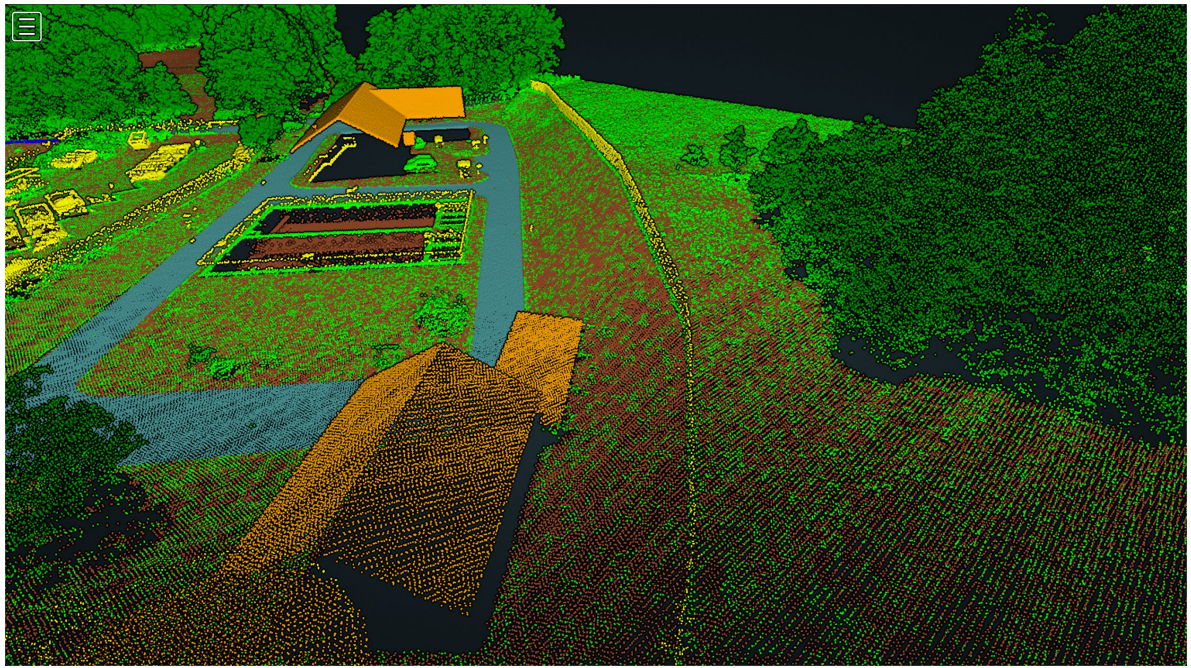
Features = [F1, F2, … , Fj]


Our Point cloud
MAGIC

Tada!



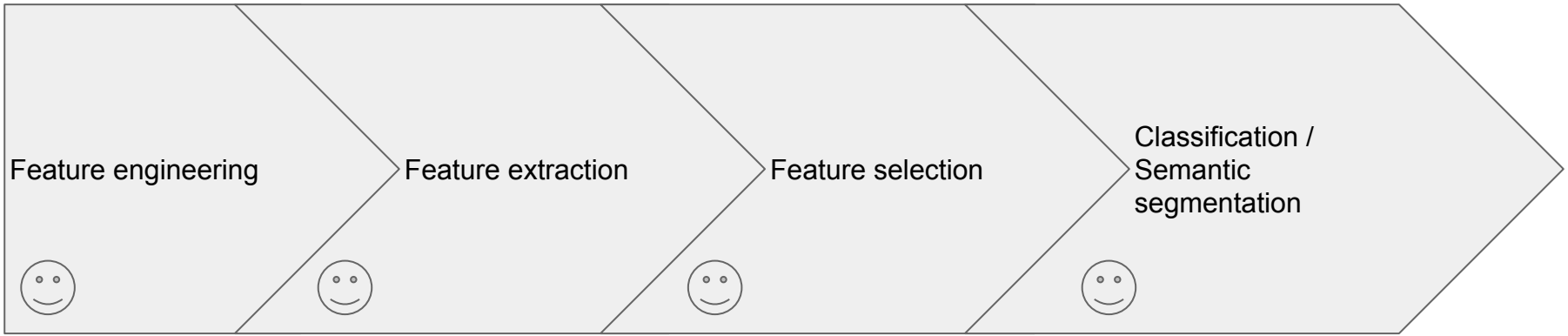




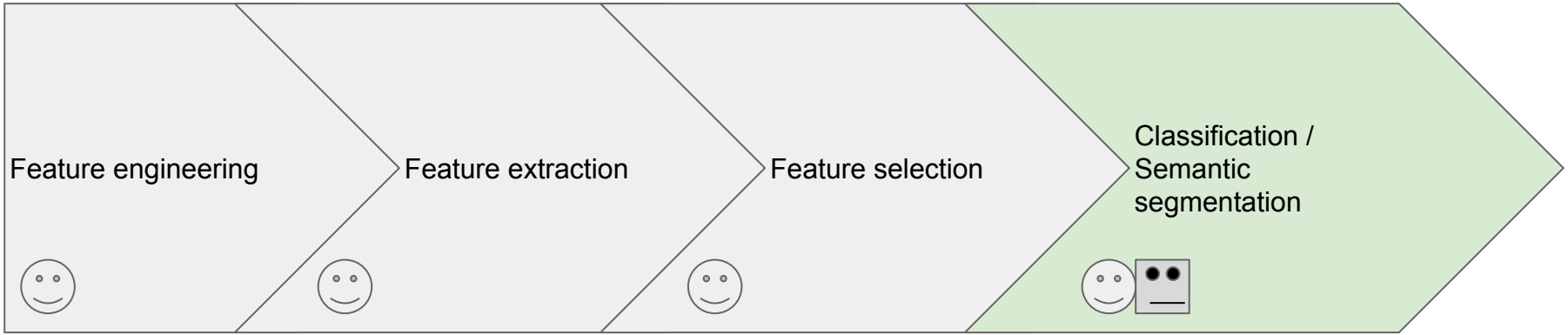
How can we do it

- Manually
- Classic algorithms for ground and building extractions
  - Morphological filtering
  - Progressive densification
  - Surface-based filtering
  - Segment-based filtering
- Classic machine learning:
  - Random forest
  - Support vector machine
  - Neural networks
- Convolutional deep neural networks

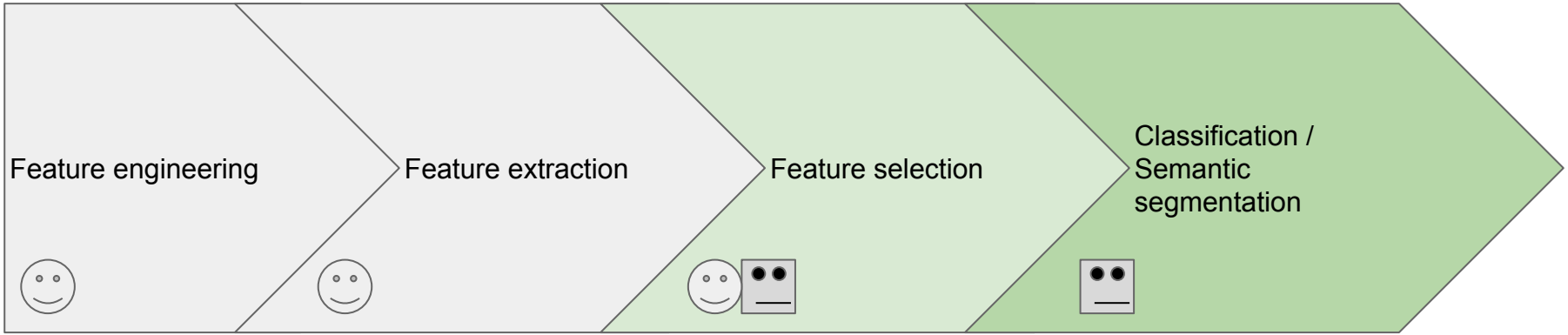
General processing pipeline (Manually)



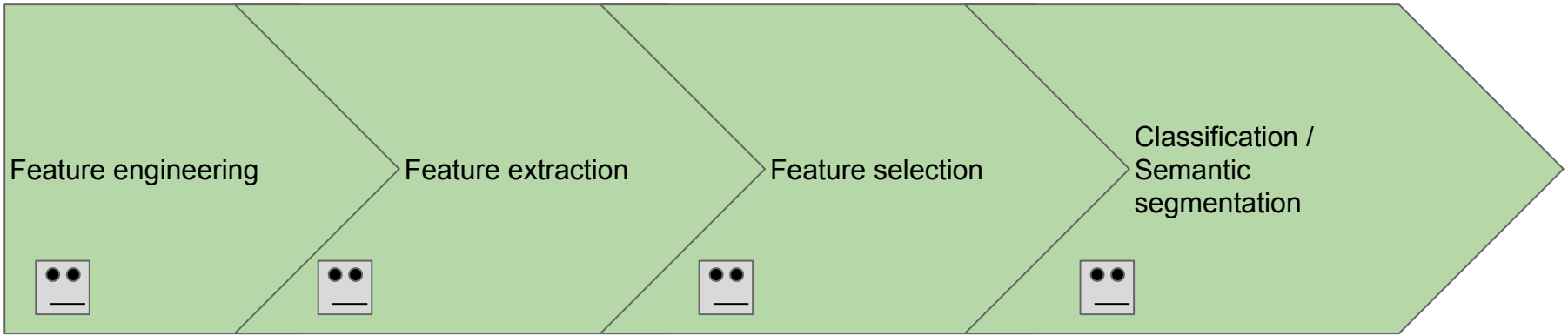
General processing pipeline (Classic filtering based)



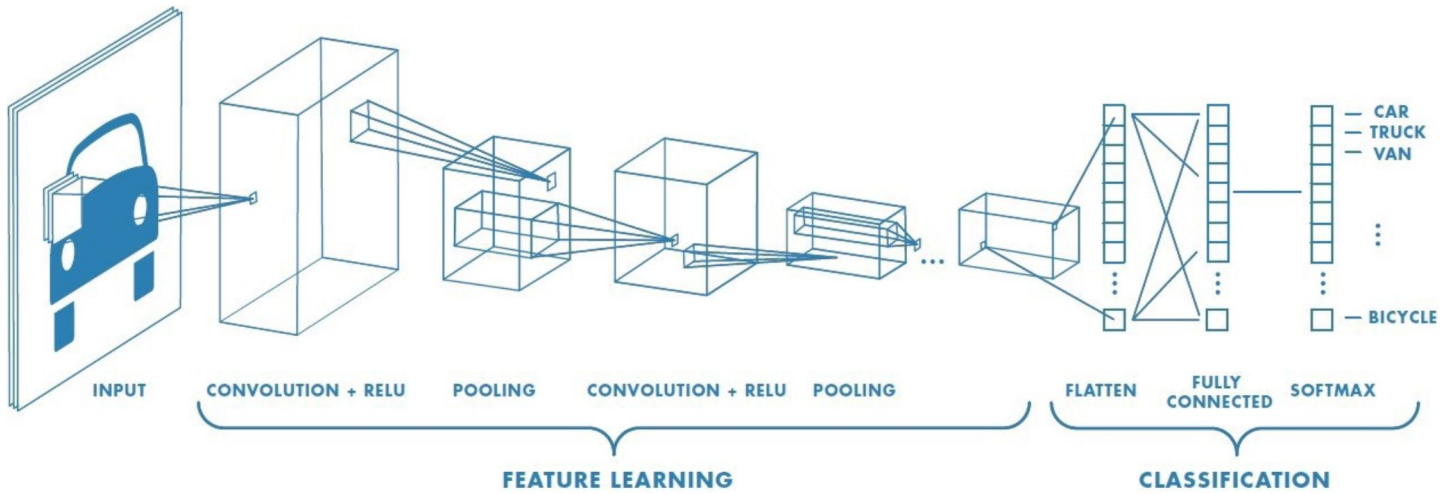
General processing pipeline (Classic ML)



General processing pipeline (Deep CNN)

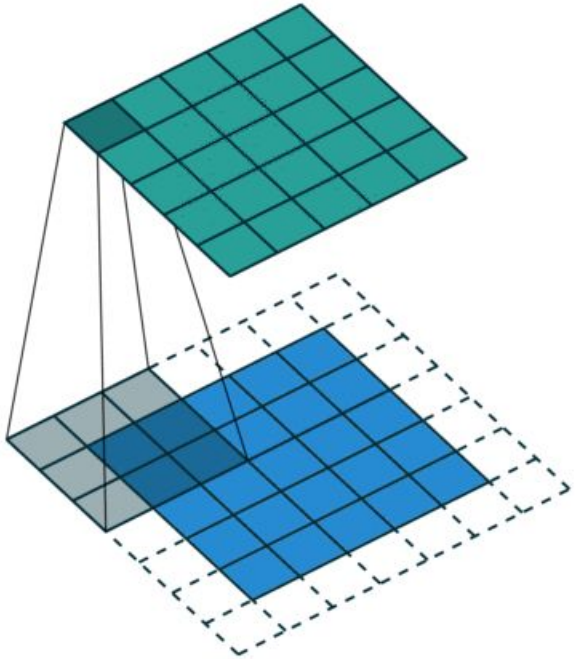


Deep learning and Convolutional neural networks for images



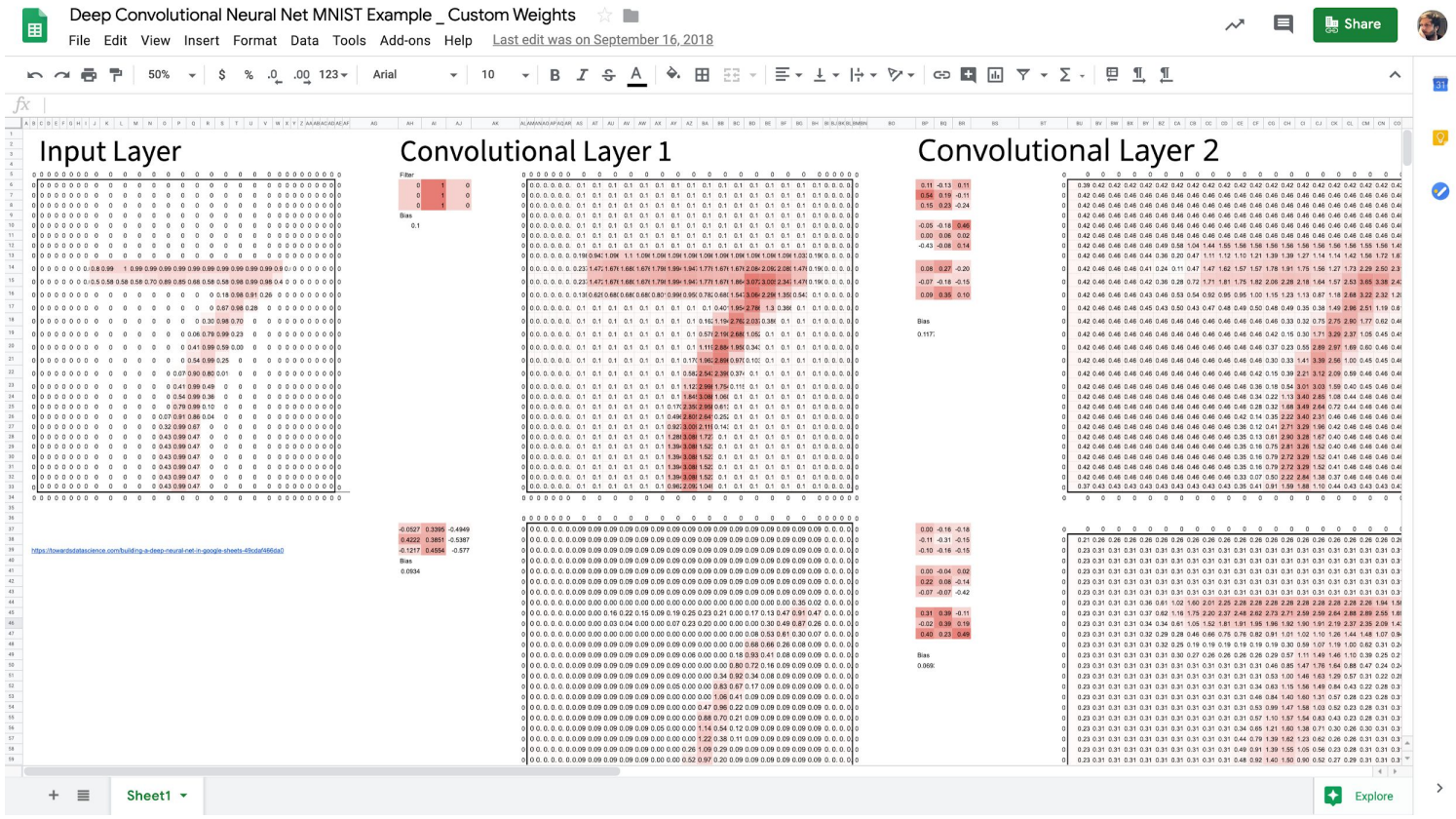
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolution

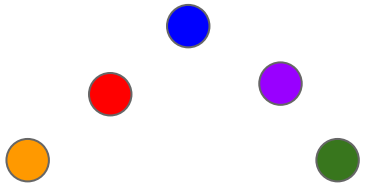


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

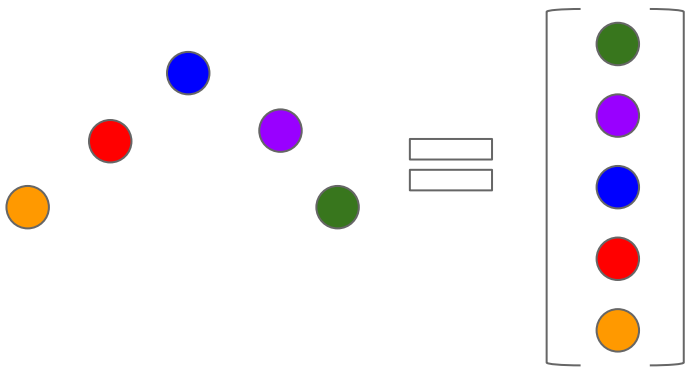
CNNs


Deep Convolutional Neural Net MNIST Example _ Custom Weights
File Edit View Insert Format Data Tools Add-ons Help
Last edit was on September 16, 2018
Share
Input Layer
Convolutional Layer 1
Convolutional Layer 2
Filter
Bias
0.1
Bias
0.0934
Bias
0.117
Bias
0.069
https://towardsdatascience.com/building-a-deep-neural-net-in-google-sheets-49dabf466da0
Sheet1
Explore

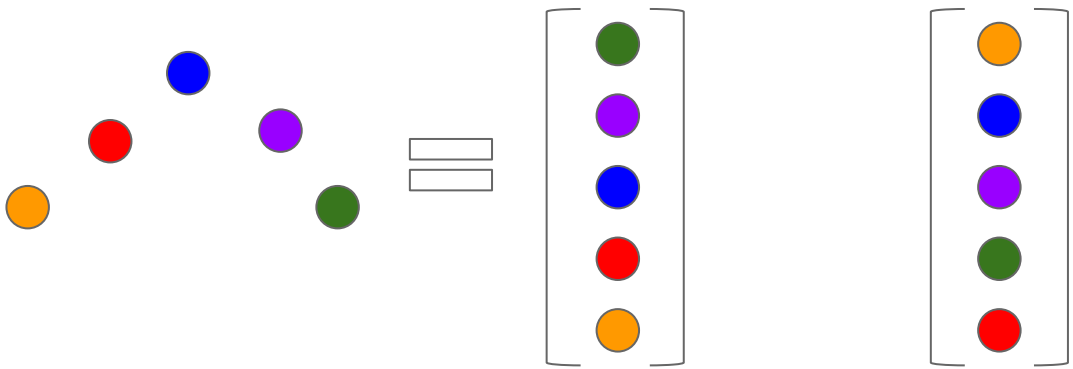
How to apply it to point clouds ?



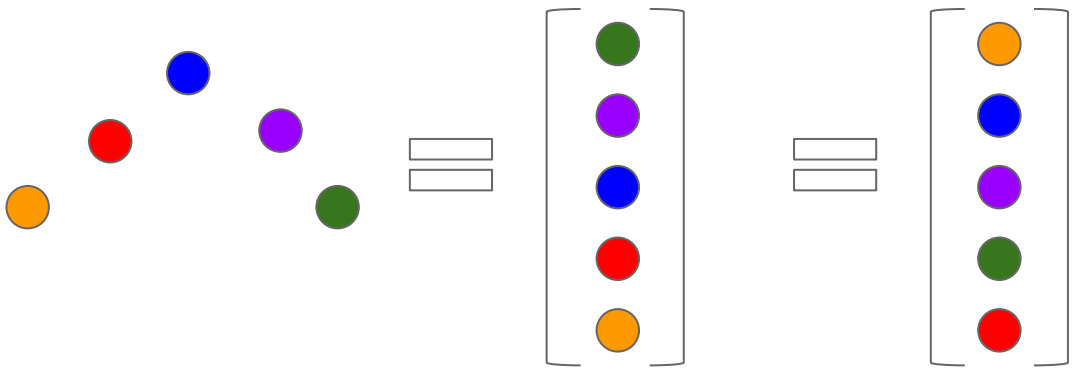
How to apply it to point clouds ?



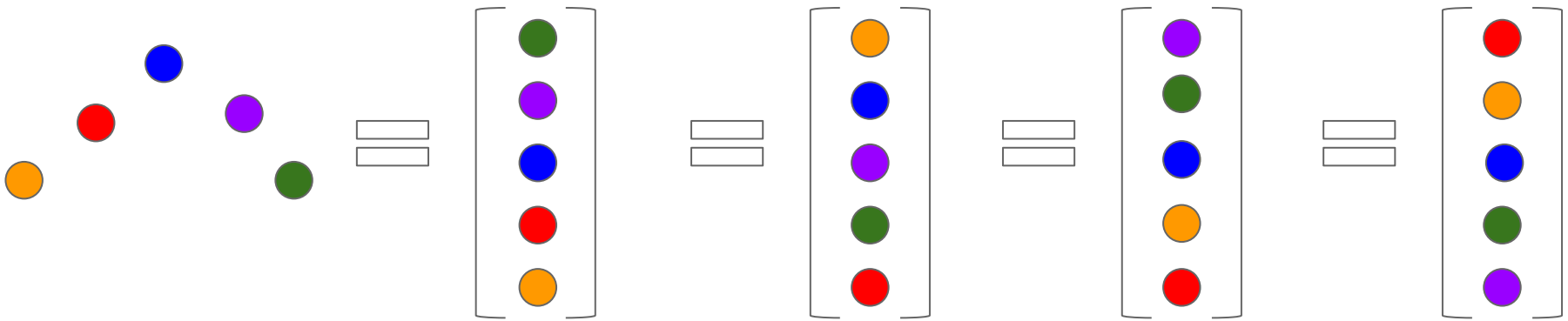
How to apply it to point clouds ?



How to apply it to point clouds ?

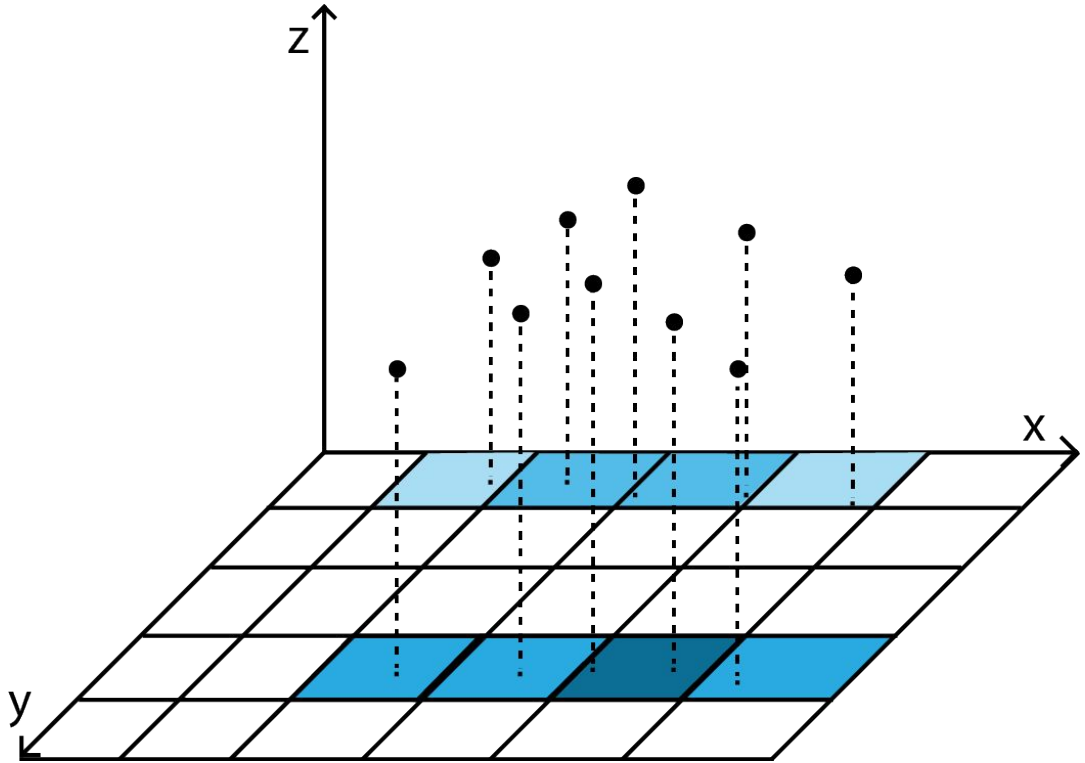


How to apply it to point clouds ?

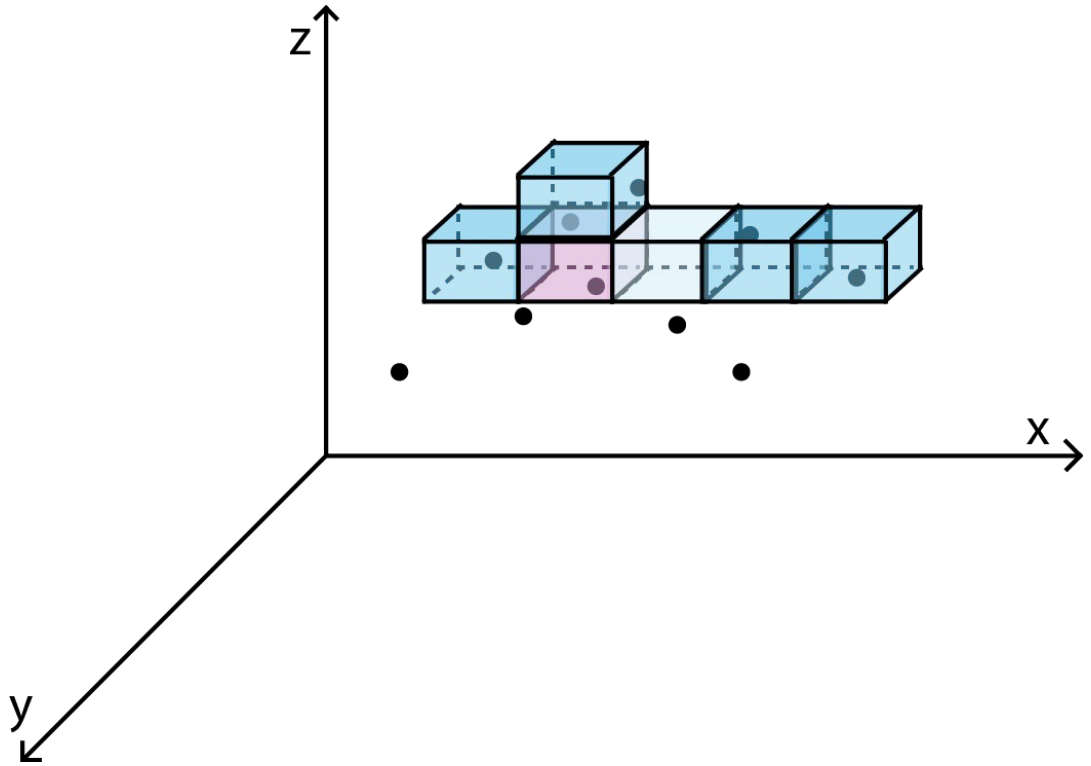


Point clouds are unordered datasets, meaning algoritme have to be permutation invariant and convolution as such is not.

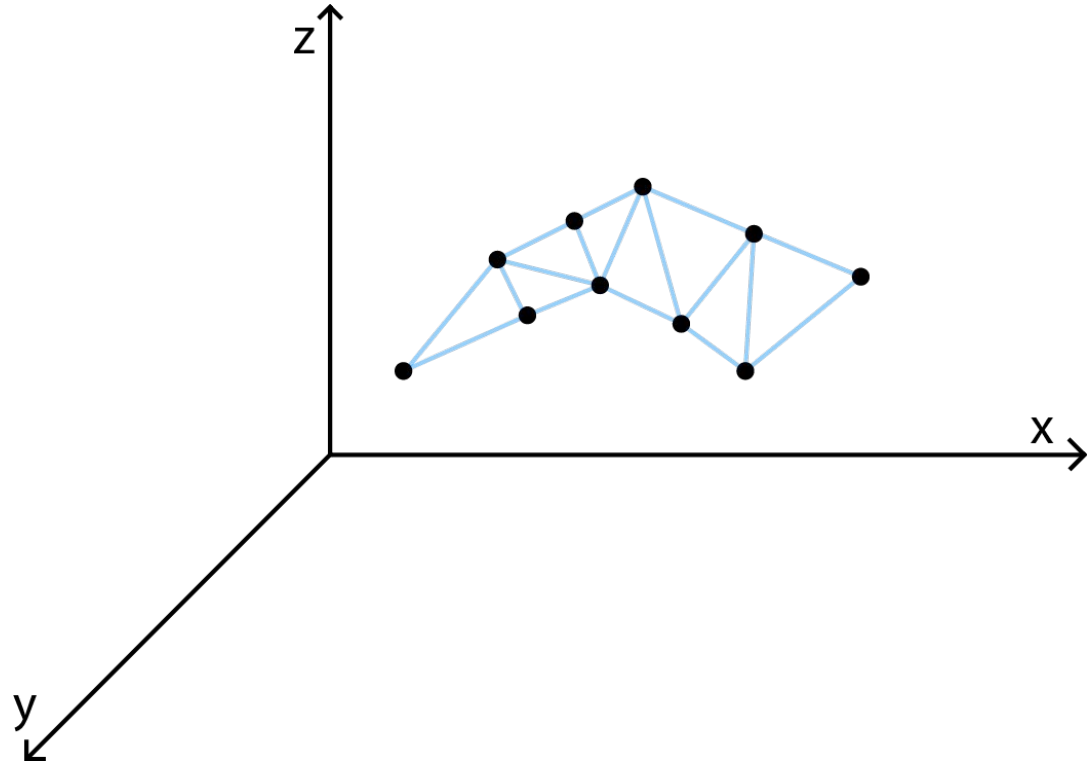
Rasterization



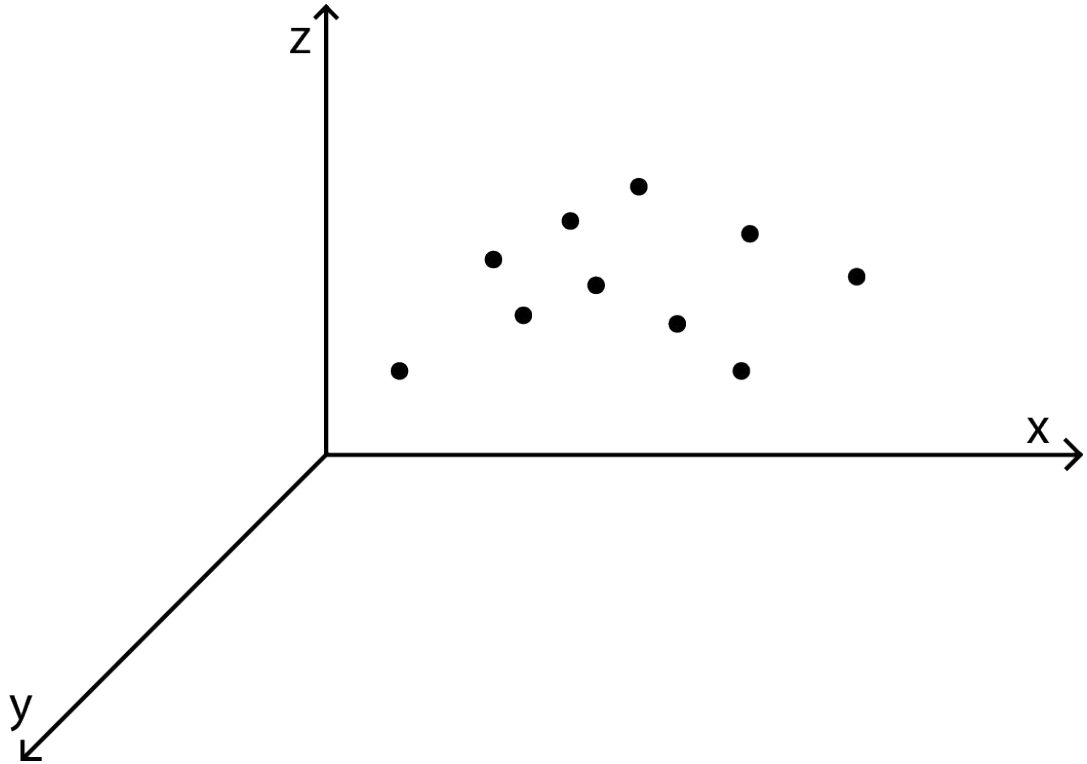
Voxelgrid



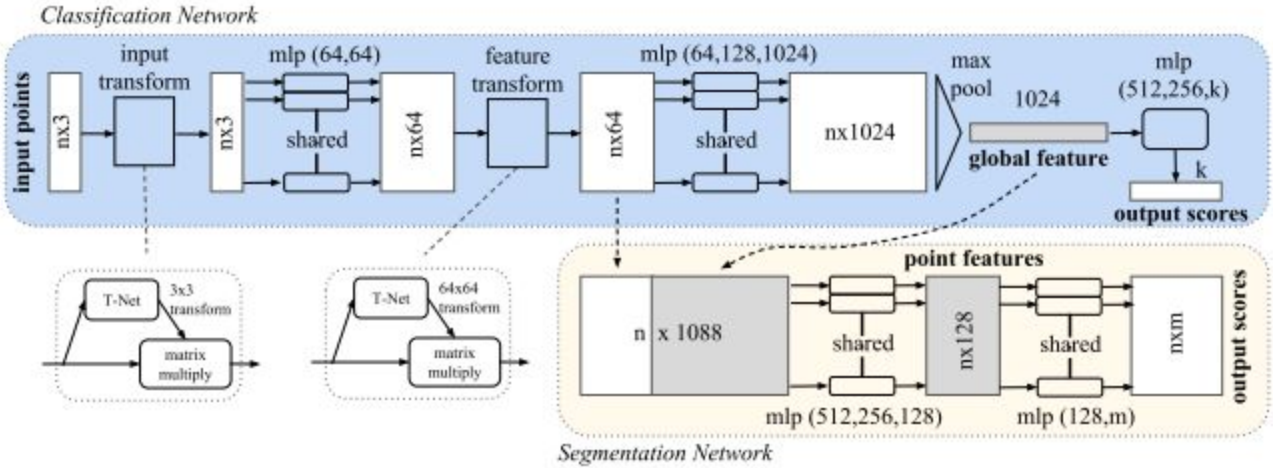
Graph based (Mesh)



Directly on points

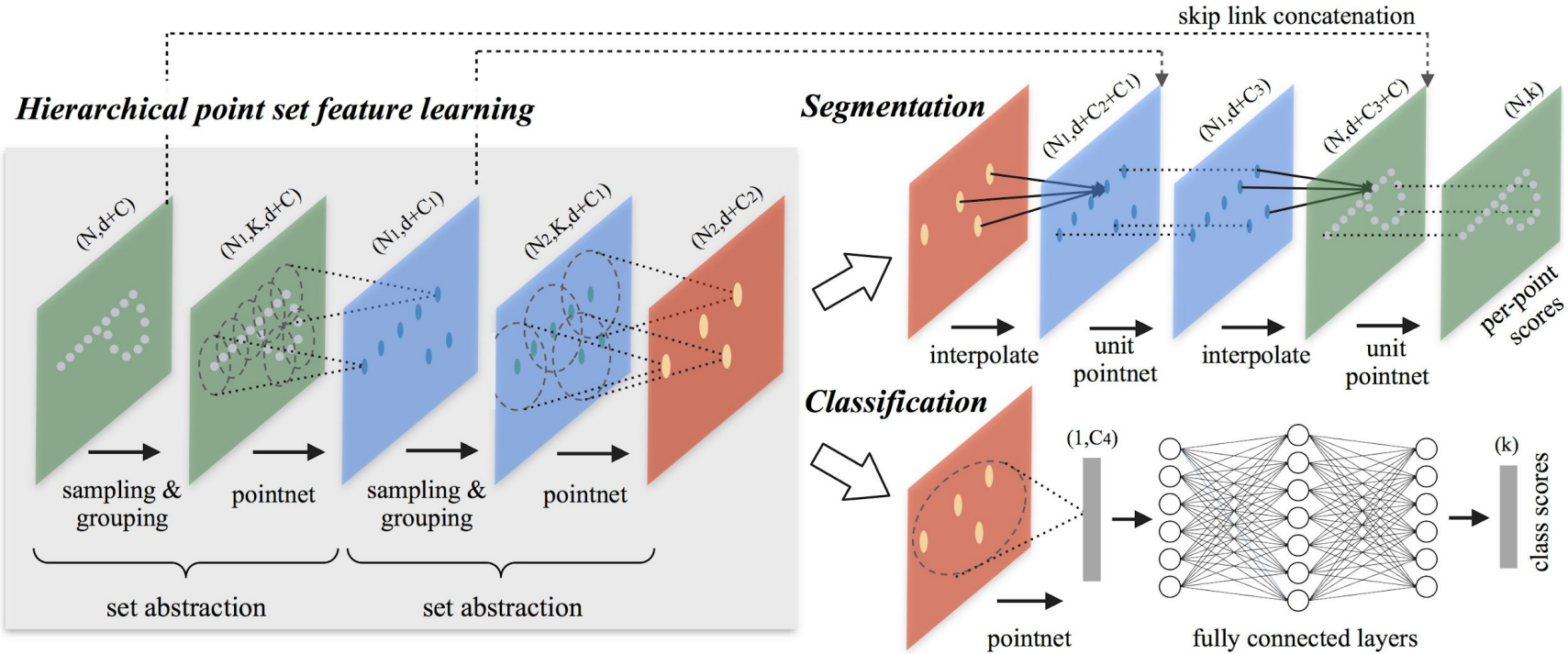


PointNet



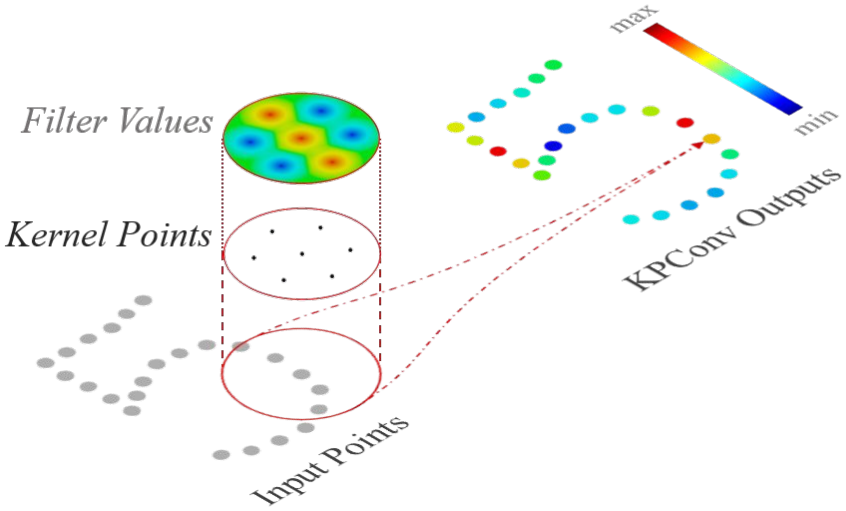
Qi et. al, 2017 [Github]

PointNet++



Qi et. al, 2017 [Github]

KPconv



KPConv (Hugues et. al, 2019) [Github]

Example CoLab

CoLab example - MNIST Pointcloud CNN Demo

Python PointCloud Package

We are hiring

- Geoinformatik / Programer
- (2x) Operater v zračnem plovilu / Task Specialist
- (2x) Študentsko delo: obdelava prostorskih podatkov
- QA/QC

http://www.flycom.si/zaposlitev/