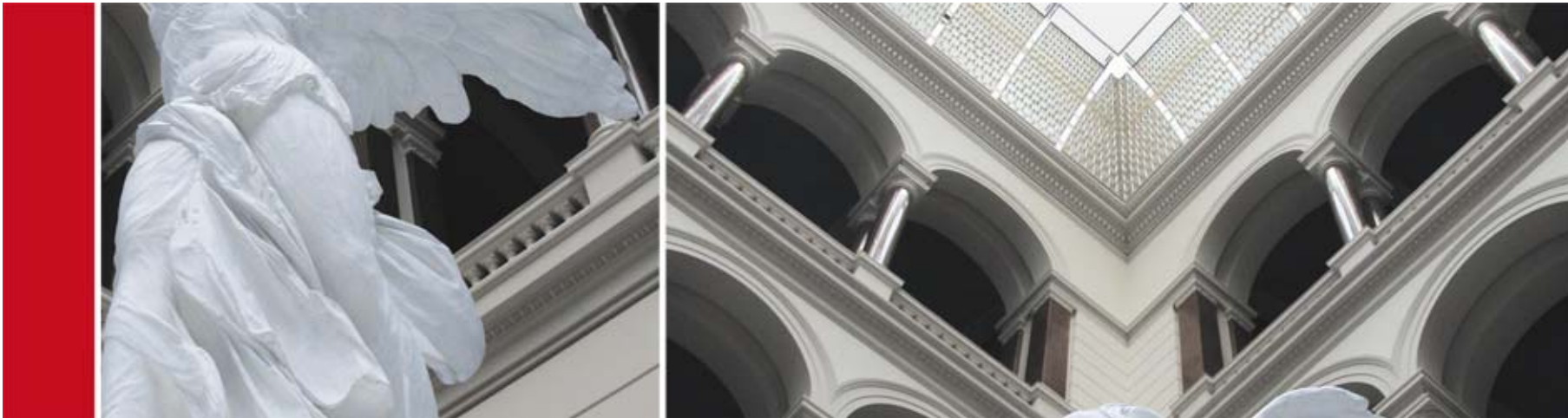
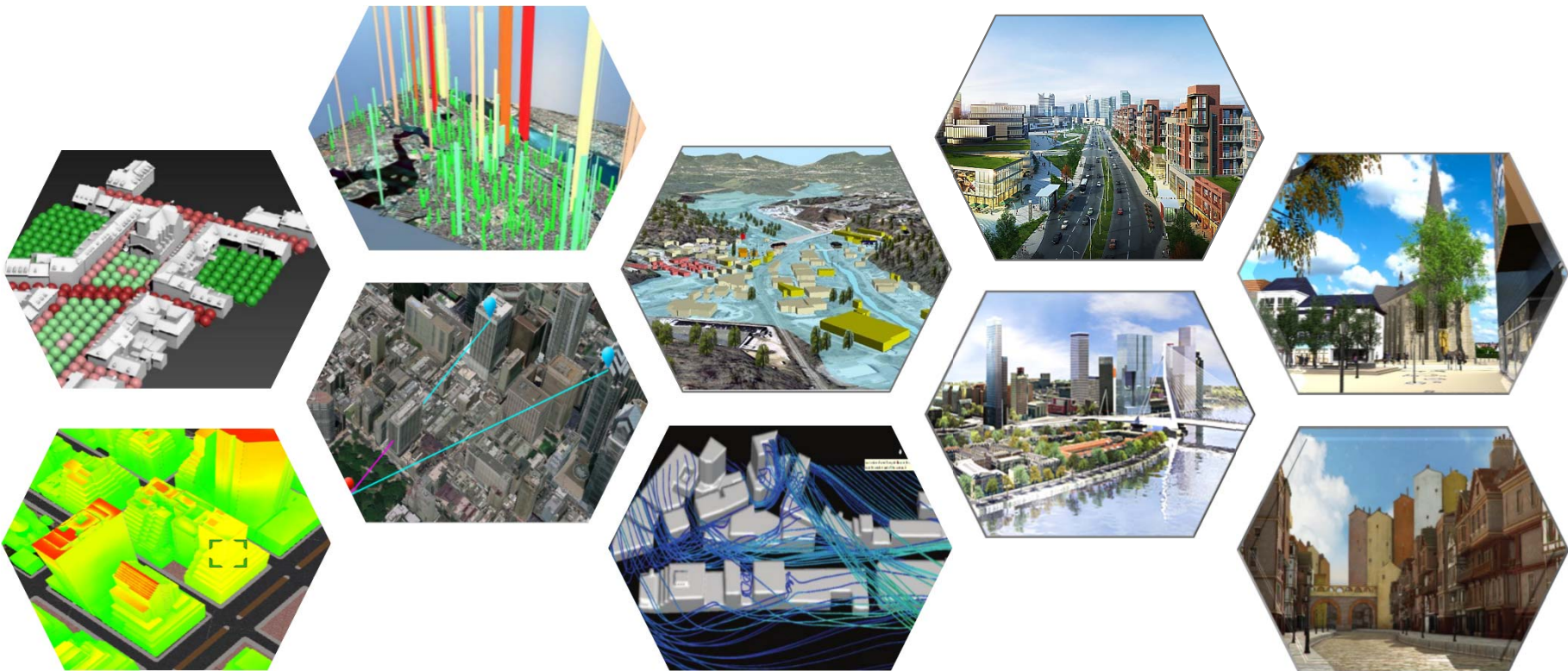




Towards a Deep Learning Framework for 3D Building Reconstruction

Valentina Schmidt, Martin Kada

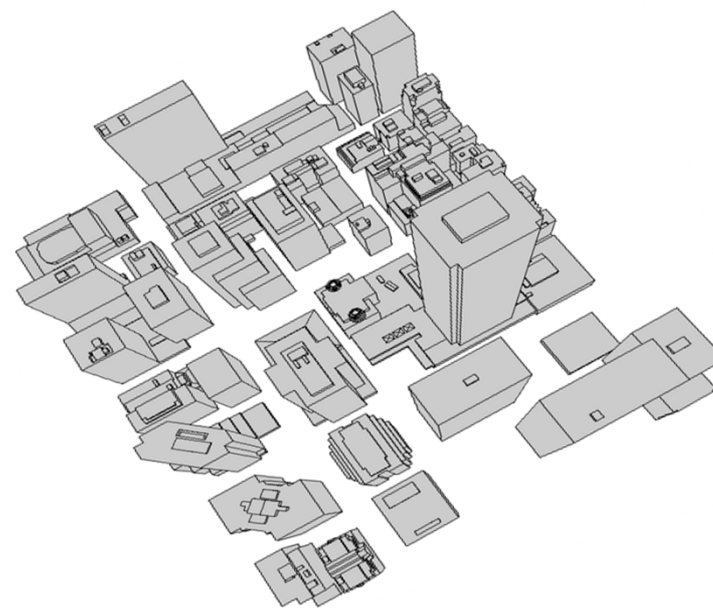
Applications for 3D Building Models



Motivation

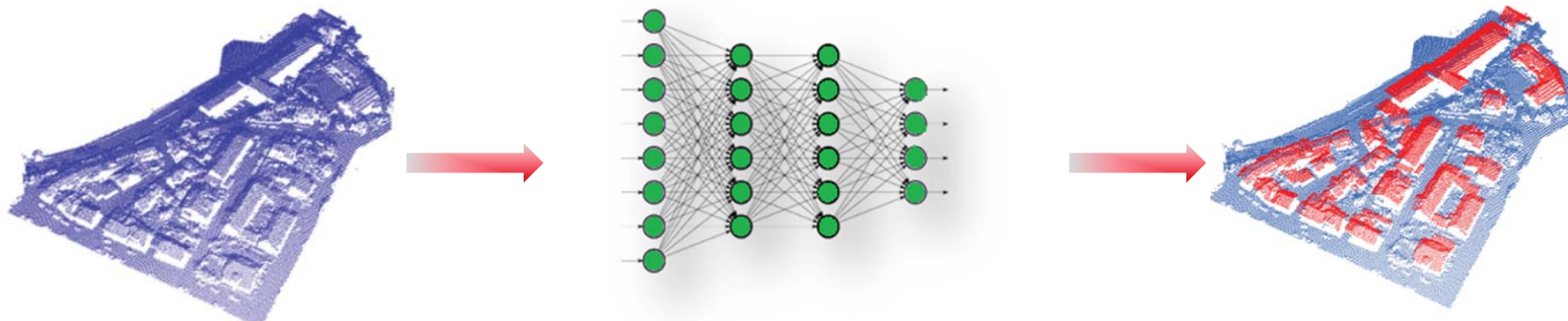
Generating up-to-date 3D city models at large scale raises the **need for automation**.

Due to the intensive parametrization they rely on, rule-based methods have challenges to generalize to new input data.

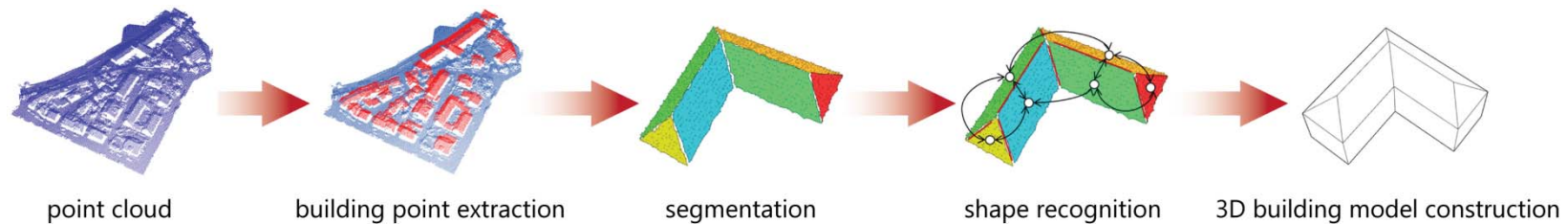


Advantages of a Deep Learning Approach for 3D Building Reconstruction

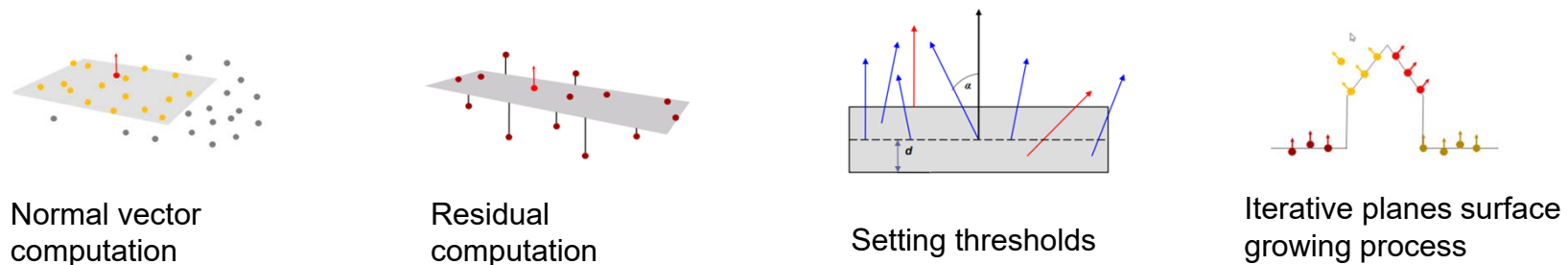
- ✓ Features are extracted directly from data
- ✓ Particularities of the data distribution are learned from data
- ✓ Strong generalization capabilities
- ✓ Scalability



Simplified Pipeline for 3D building reconstruction



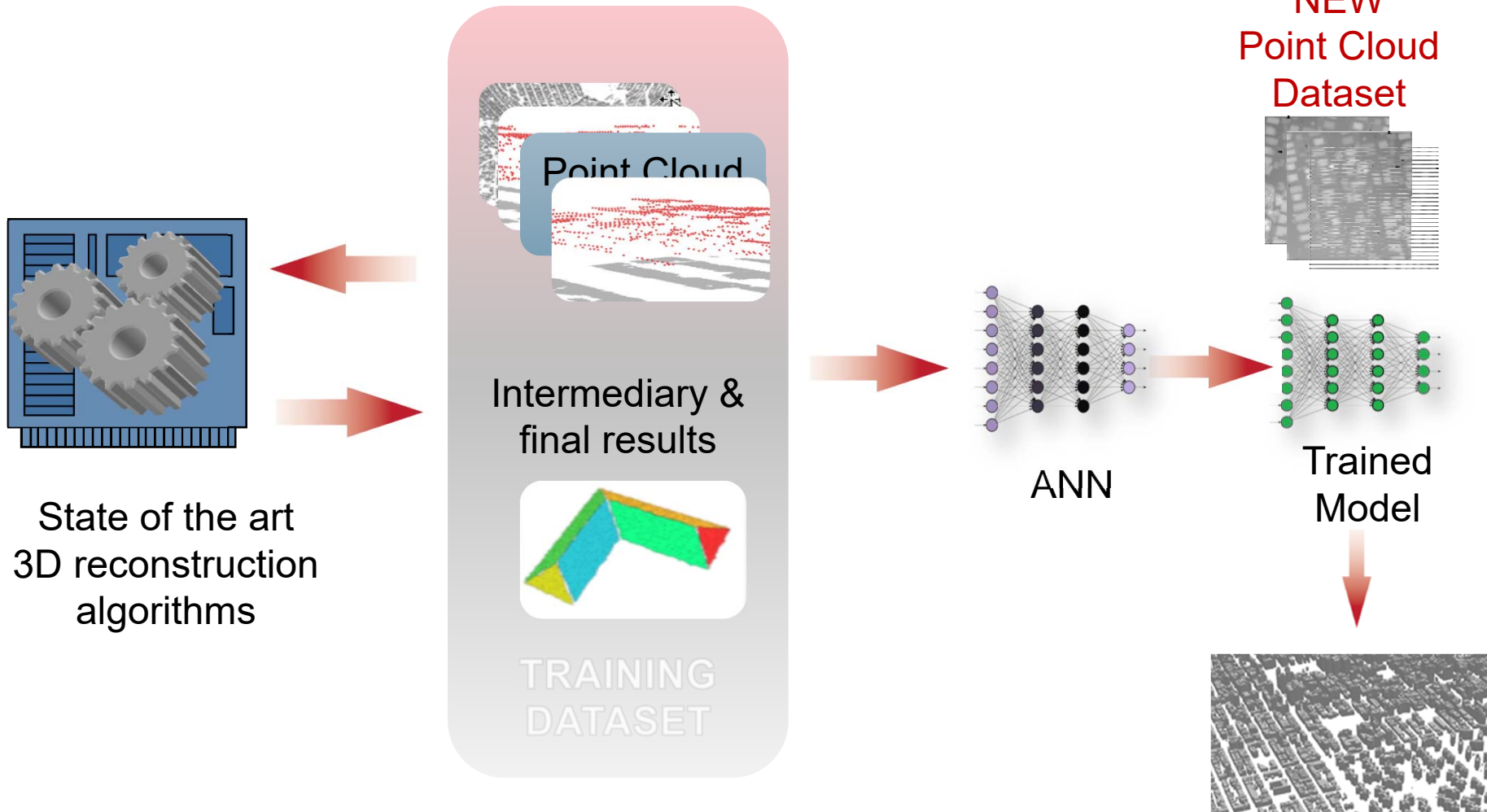
Example of processing workflow: Segmentation



Features extraction based on **hand designed parametrization** according to the particularities of the **current dataset**

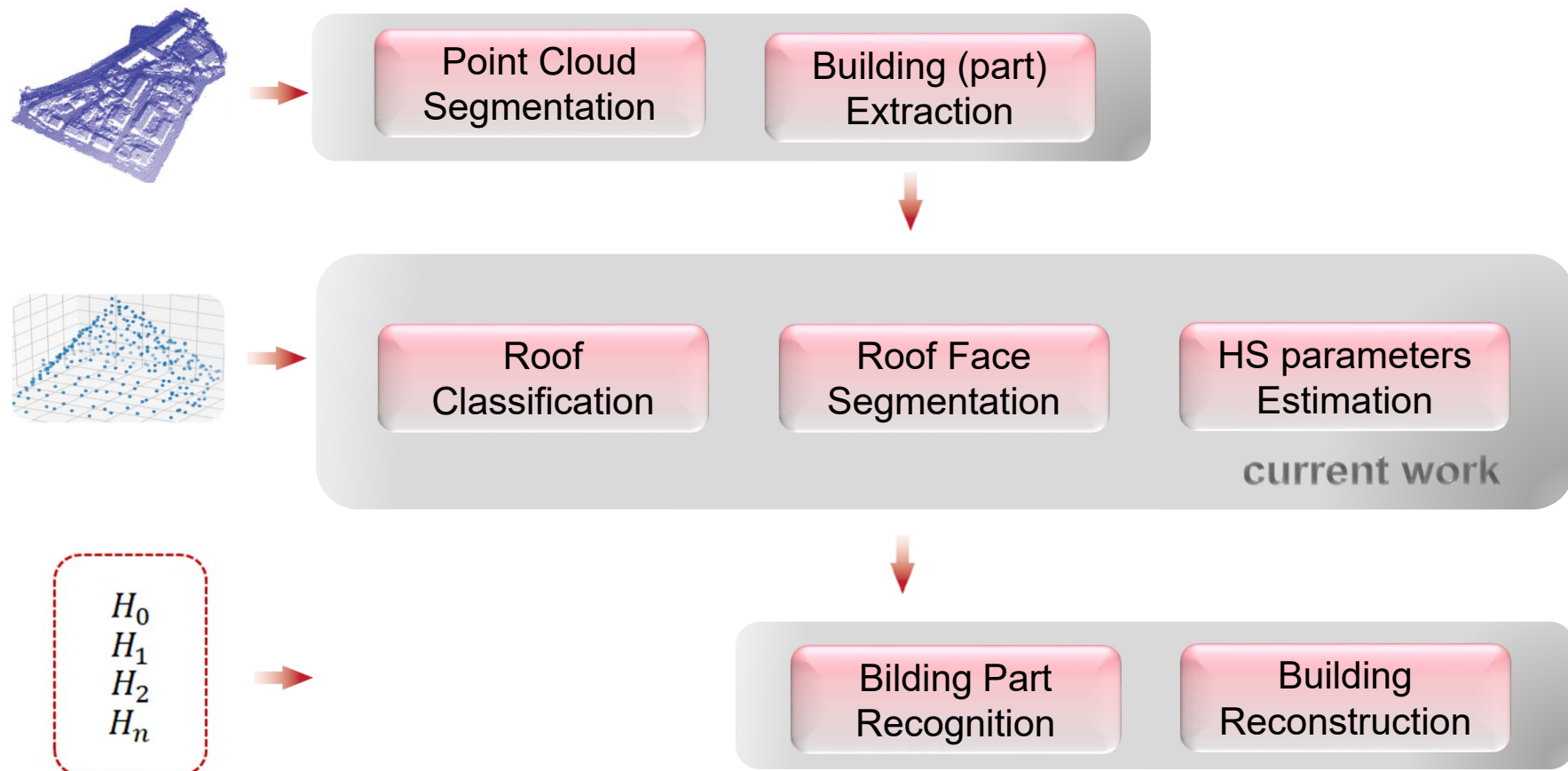
Proposed Deep Learning Framework for 3D Building Reconstruction

NEW
Point Cloud
Dataset



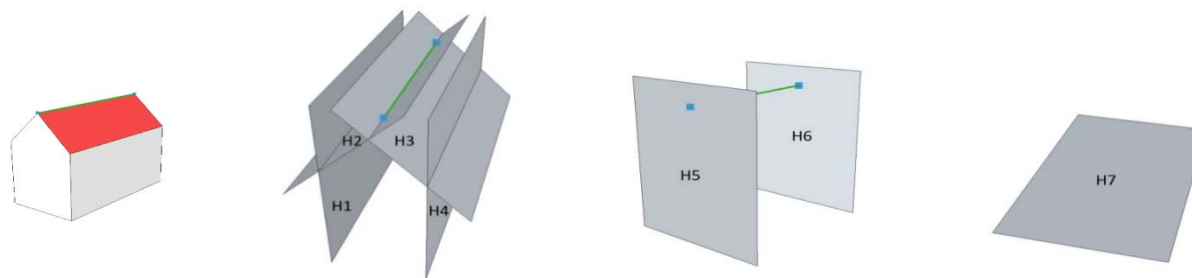
<http://www.roofn3d.de>

Deep Learning Approach to 3D Building Reconstruction with Half-Space Modeling



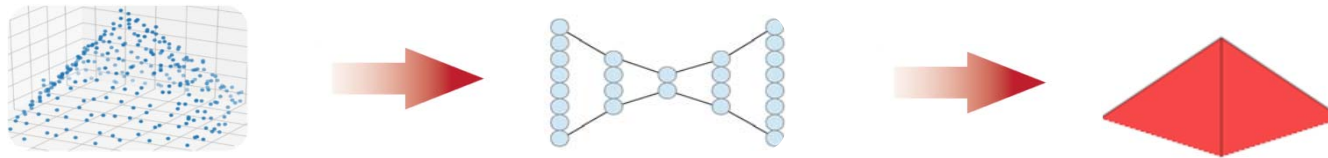
Half-space 3D Building modeling with Deep Learning

- ✓ A solid can be expressed as Boolean combination of a set of (planar) half-spaces
- ✓ Bijective mapping between (planar) segments and half-spaces
- ✓ Allows for abstracting away the shape/topology of the solid
- ✓ The half-space parameter values defining solids can be expressed as sequences



$$S = H_1 \cap H_2 \cap H_3 \cap H_4 \cap H_5 \cap H_6 \cap H_7$$

Feasibility study of a DL approach to 3D building reconstruction with HS modeling



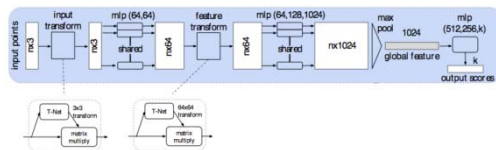
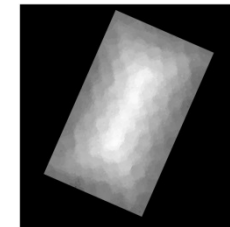
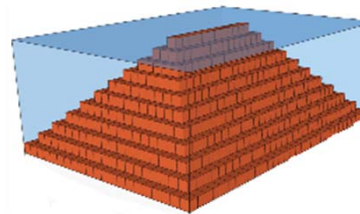
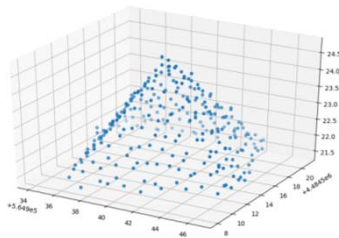
- ✓ Most often containing only geometric (implicit) information
- ✓ Irregular data structure (no explicit neighborhood, connectivity)
- ✓ 2.5D (2D manifold)
- ✓ Invariable to pose, illumination, texture
- ✓ Simple generating data distribution

Data
representation?

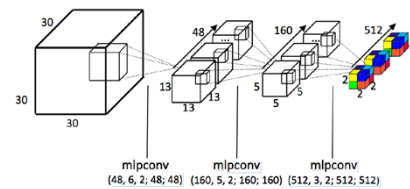
Architecture
type?

Network
capacity?

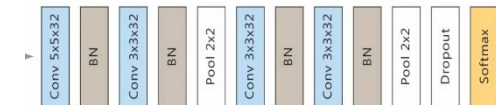
Roof Classification



PointNet



3D CNN



2D CNN

RoofN3D Database

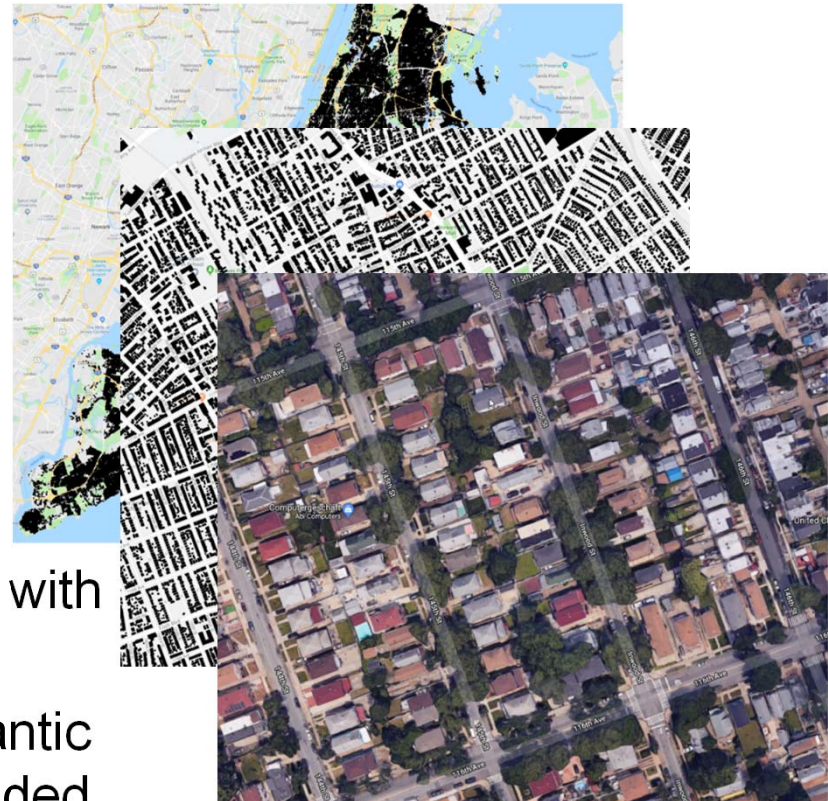
New York dataset

- ✓ Area of about 1,000 km²
- ✓ Average point density of about 5 points/m²
- ✓ > 1,000,000 buildings

RoofN3D

- ✓ Massive point cloud training data with focus on buildings
- ✓ Not only geometric but also semantic and structural information is provided

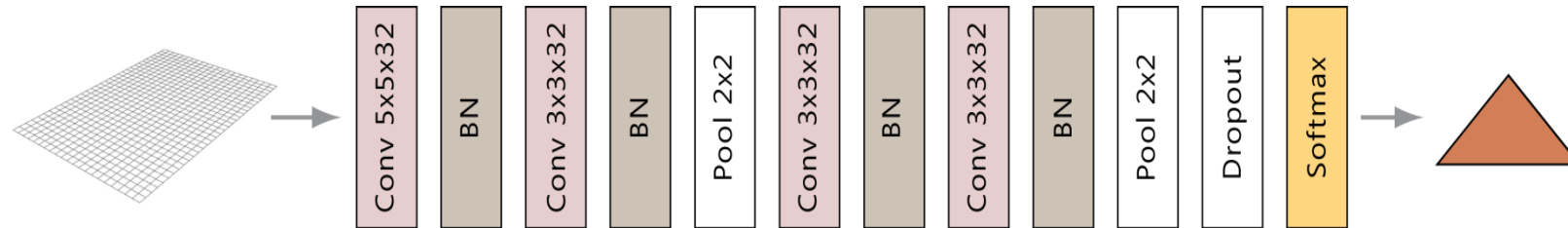
<http://www.roofn3d.de>



Dataset for Classification

Roof Type	#Training examples	#Validation examples	#Test examples	Total
Pyramid	1000	250	310	660
Two-sided Hip	11.610	2.900	3.630	18.140
Saddleback	49.457	12.360	15.450	77.267
Total	62.055	15.510	19.390	96.067
Ratio	63 %	16%	20%	

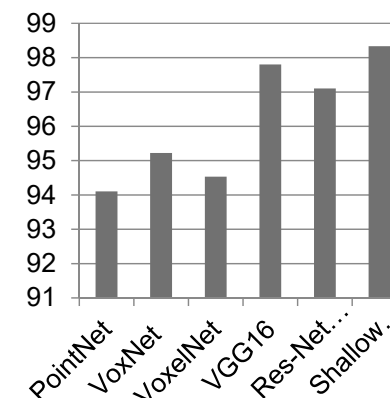
Shallow Convolutional Network



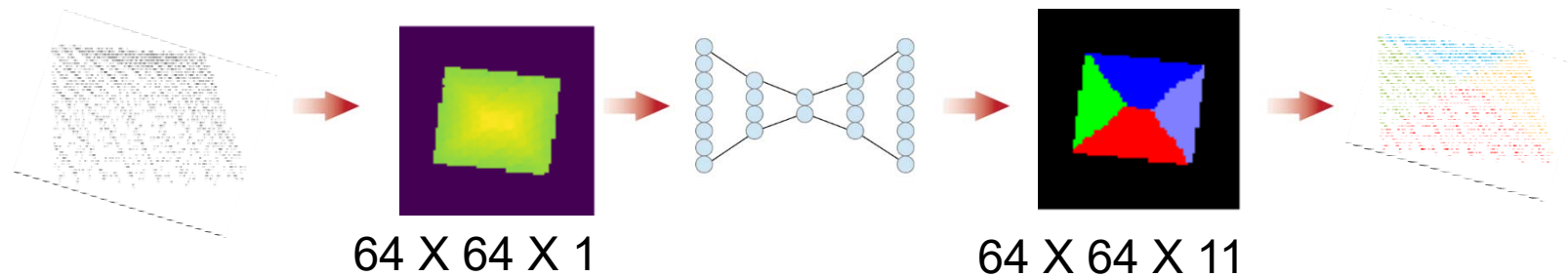
Roof Classification Results

Data Representation	Architecture	# Parameters	F1 Score
Point Cloud	PointNet ¹	3,5M	94.10
Volumetric (adaptive size, density grid)	VoxNet ²	0.9M	95.22
Volumetric	VoxelNet ³ (adapted)	0,46M	94.53
2D Raster	VGG16 ⁴	138 M	97,8
2D Raster	Res-Net Like Module	0.27M	97,10
2D Raster	Shallow Model	0.055M	98.33

F1 Score



Roof Face Segmentation

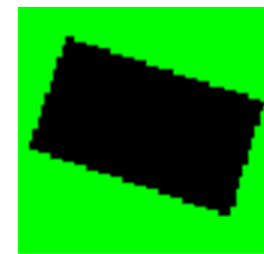
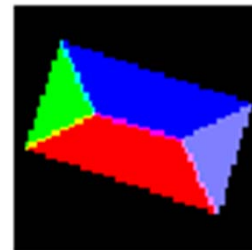


- ✓ deriving masks for the roof faces composing a building roof
- ✓ the roof segmentation problem is formulated as pixel-wise classification

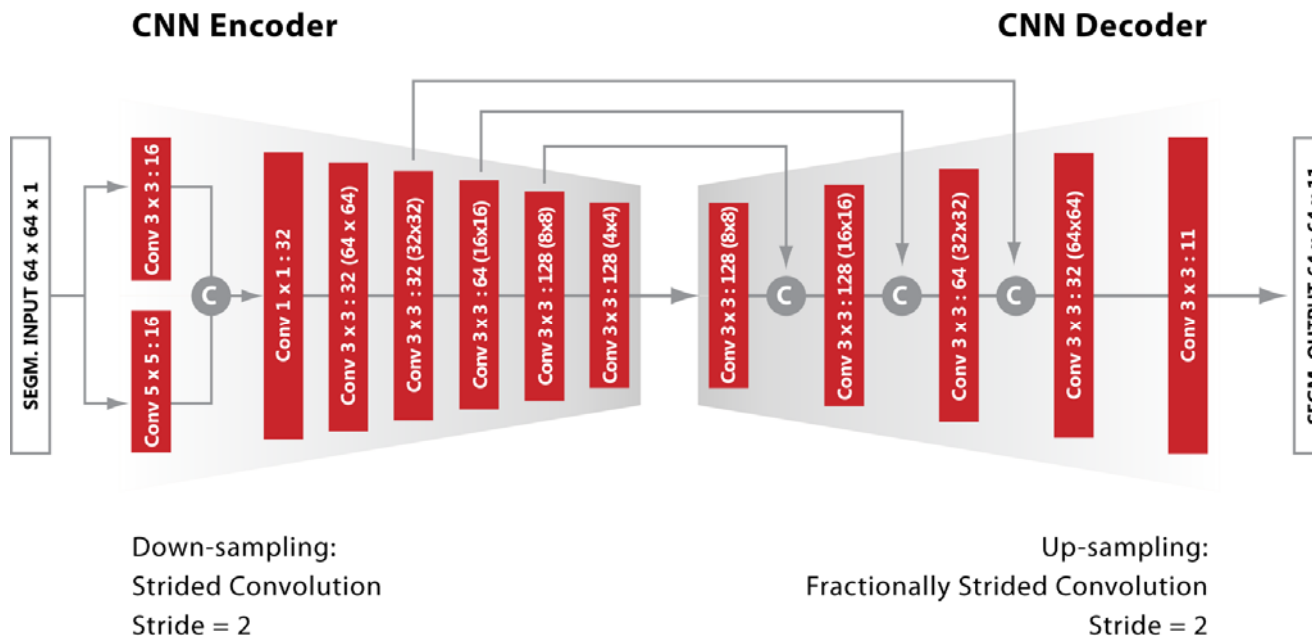
Training Data for Roof Segmentation

- 11 classes => label arrays shape: 64x64x11
 - background
 - 10 types of roof faces defined with respect to:
 - roof type
 - orientation

=> **Joint predictions** for segmentation and roof classification



CNN Encoder-Decoder Architecture



Objective
function:

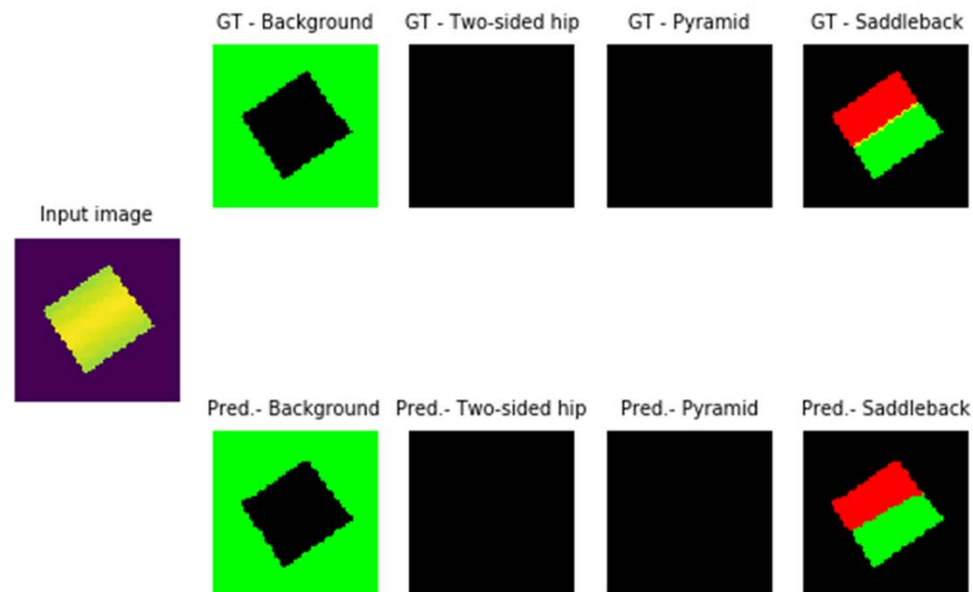
$$J(y, x) = - \sum_i \alpha_i p(y) \log(p(x_i))$$

$$\alpha_i = \text{median_freq} / f_i$$

$$f_i = \# \text{ points_class_i}$$

Roof Face Segmentation Results

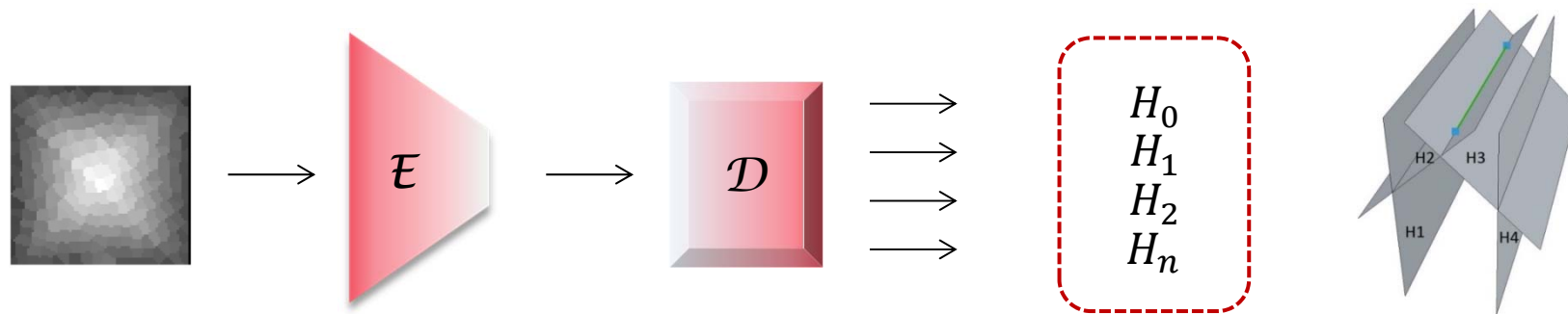
Image id: C:\00 - Presentation\data-2\surfacegrowing\testing\Saddleback roof\46988



Results Roof Face Segmentation

CNN Encode-Decoder Architecture	MIoU (raster)	Categorical Accuracy (per pixel)	Categorical Accuracy (per point)
CNN E-D 4 Conv + 4 Deconv	.93	.95	.78
CNN E-D Inception Module 4 Conv 4 Deconv	.94	.96	.78
CNN E-D Inception Module 4 Conv 4 Deconv +Skip conn.	.9475	.9760	.80

Learning Orientation Parameters with Regression



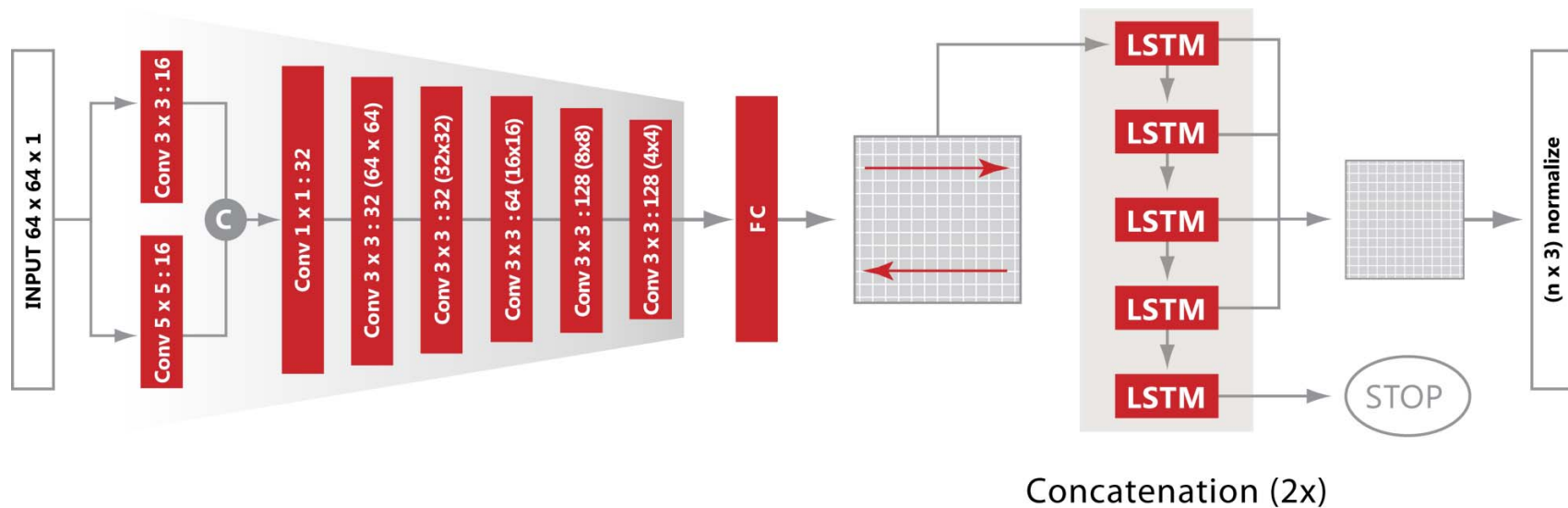
Input: height maps

Outputs:

a set of orientation parameters per **roof segment**

a confidence score per orientation parameters set candidate

DL Architecture for Orientation Parameters Inference (variable size output)



Objective function:

$$L_{total} = \alpha L_{normals} + L_{conf_score}$$

$$L_{normals} = -\frac{1}{n} \sum_i N \cdot \frac{\hat{N}}{\|\hat{N}\|}$$

$$L_{conf_score} = -\frac{1}{n} \sum_c y_c (\log \hat{y}_c)$$

Orientation Parameters Regression Results

Encoder	Decoder	Output	Mean Error azimuth angle (deg)	Mean Error Vertical angle (deg)	Cosine distance	Score accuracy
	Fully Connected	Spherical coordinates	29.66	5.03		
Downsamp. MaxPool	LSTM, (fix size output)	Orientation parameters	8.80	4.72	0.975	
Downsamp. Strided Conv	LSTM, (fix size output)	Orientation parameters	5.82	4.1	0.978	
Downsamp. Strided Conv	LSTM (variab. size output)	Orientation parameters	5.41	4.00	0.997	0.93

LSTM output	Angles err. ≤ 1 deg. (%)	Angles err. ≤ 3 deg. (%)	Angles err. ≤ 5 deg. (%)	Angles err. ≤ 7 deg. (%)	Angles err. ≤ 10 deg. (%)	Angles err. ≤ 15 deg. (%)	Angles err. ≤ 30 deg. (%)	Angles err. ≤ 60 deg. (%)	Angles err. ≤ 180 deg. (%)
Fix size	26.25	63.44	79,11	87,16	93,56	96,38	98,04	98,98	100
Var. size	27.67	66.57	81,59	89,26	94,33	97,54	98.49	98,92	100

Conclusions and Outlook

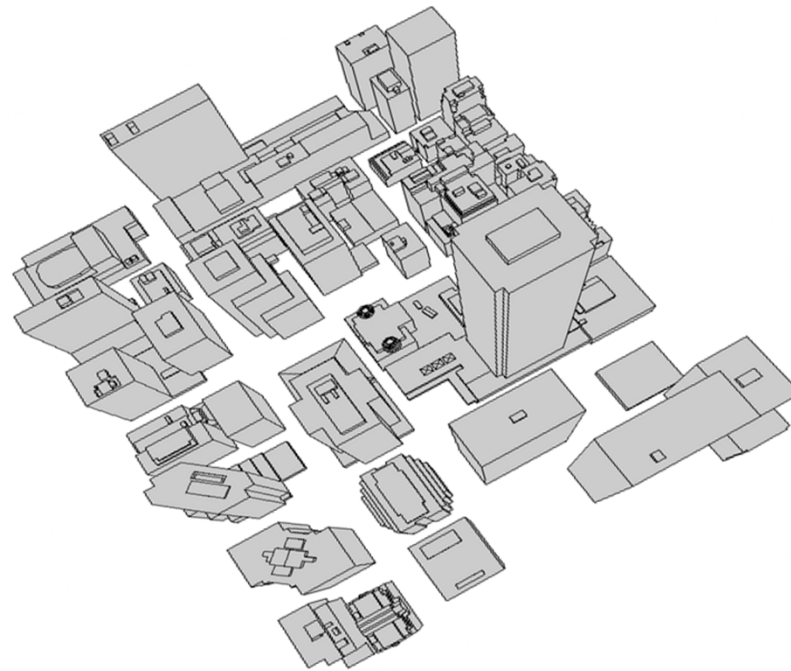
- ✓ The design and capacity of a 2D CNN for processing PC data in 2D representation (DSM) was determined
- ✓ A CNN encoder-decoder architecture was proposed for segmenting basic roof types (with fixed number of roof faces and no superstructures)
- ✓ An encoder (CNN) –decoder (LSTM) architecture for orientation parameters inference per roof segment was proposed

Open issues:

- Roof segmentation for complex roof types, including superstructures
- Joint estimation of the orientation the 4.th parameter- plane distance to origin



Thank you!



References

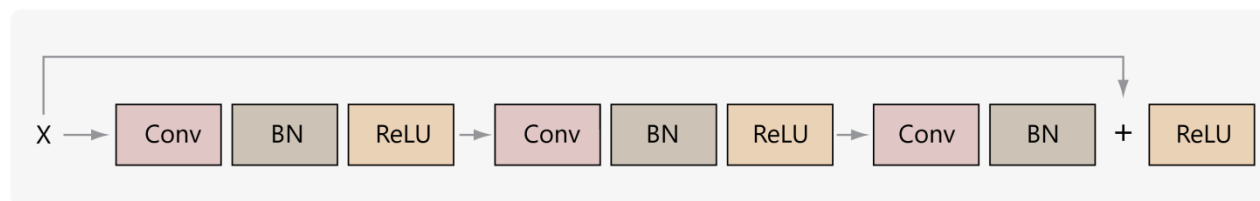
- 1 PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Qi et. al, 2017
- 2 VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition, Maturana and Scherer, 2015
- 3 VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, Zhou et. al, 2017
- 4 Very Deep Convolutional Networks for Large-Scale Image Recognition, Simonyan and Zisserman, Zisserman 2015

Supplementary material

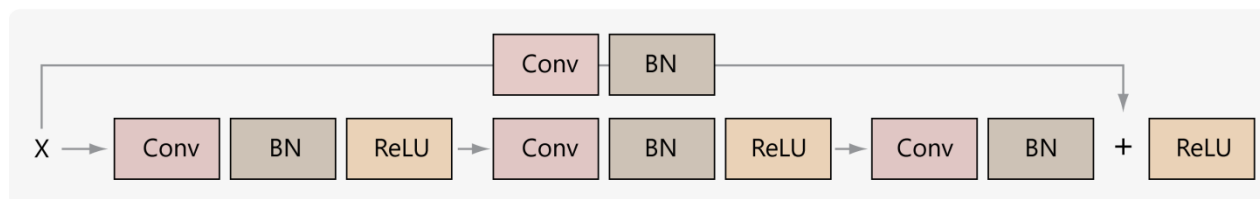
Residual Module Architecture



Id Block



Conv Block



Training Data set

