School of Science
Department of Physics and Astronomy
Master Degree in Physics

Deep Learning for Urban LiDAR Segmentation: Integrating AI into Bologna's Digital Twin Initiative

Submitted by:
Matteo Falcioni

Supervisor:
Prof. Mirko Degli Esposti

21 February 2025

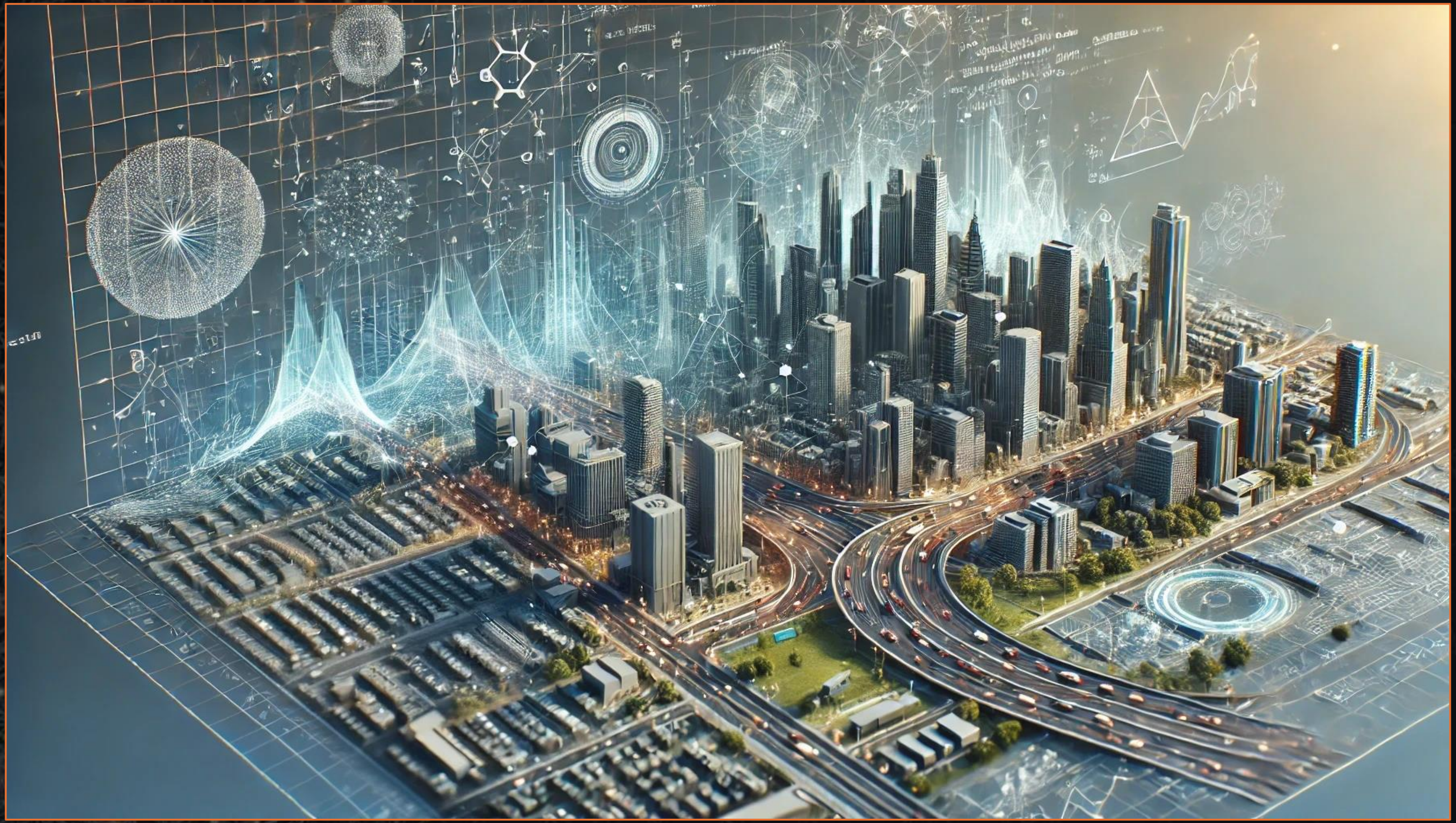
THE CITY AS A PHYSICAL SYSTEM

Patterns, interactions, and behaviors similar to complex physical systems

Geometrical – Physical modeling of cities

↓

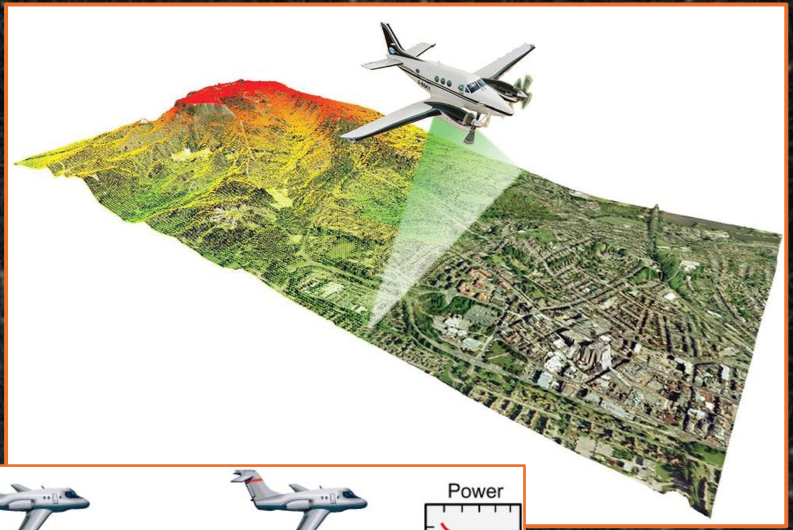
Urban Digital Twins

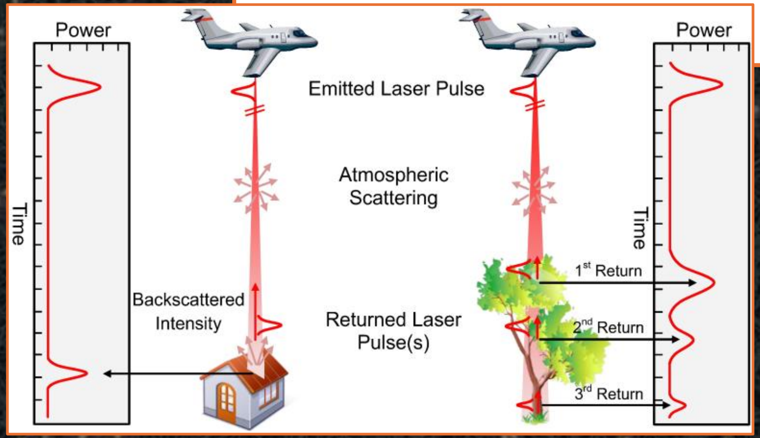


Statistical Physics → Urban scaling laws, self-organizing patterns
Fluid Dynamics → Traffic flow, traffic "shock waves"
Thermodynamics → Urban heat islands, energy consumption
Network Theory → Road networks, power grid connectivity

AIRBORNE LiDAR DATA

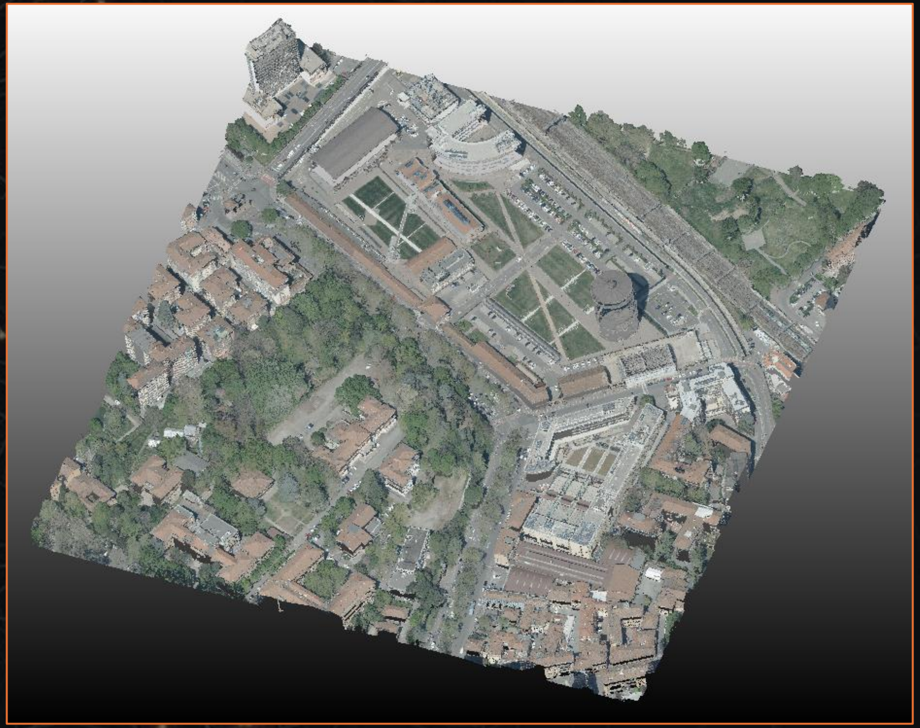
LiDAR uses mono-frequency laser pulses to measure distances based on time-of-flight and analyze optical properties of surfaces





Airborne LiDAR campaign

Point cloud data (*.las* files)



Result: 654 tiles, each one covering an area of 500x500 meters

DEEP NEURAL NETWORKS ON AIRBORNE LiDAR

sensors — MDPI

Article

Segmentation and Multi-Scale Convolutional Neural Network-Based Classification of Airborne Laser Scanner Data

Zhishuang Yang [1], Bo Tan [2], Huikun Pei [2] and Wanshou Jiang [1,*]

remote sensing — MDPI

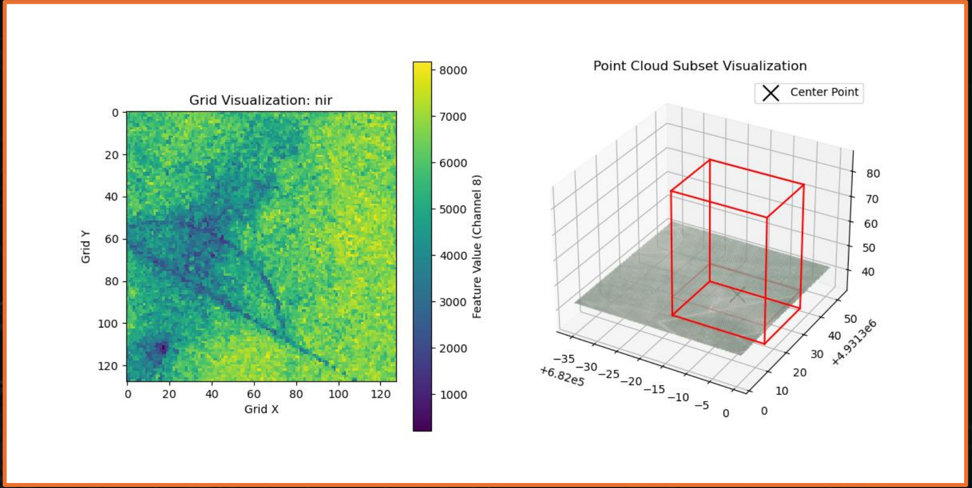
Article

A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds

Zhishuang Yang, Wanshou Jiang *, Bo Xu, Quansheng Zhu, San Jiang and Wei Huang

Idea: project LiDAR data on 2D in order to use pre-existing NN architectures for images

Projection: from LiDAR point cloud to feature image



+

Convolutional NN

Input: n-channel feature image

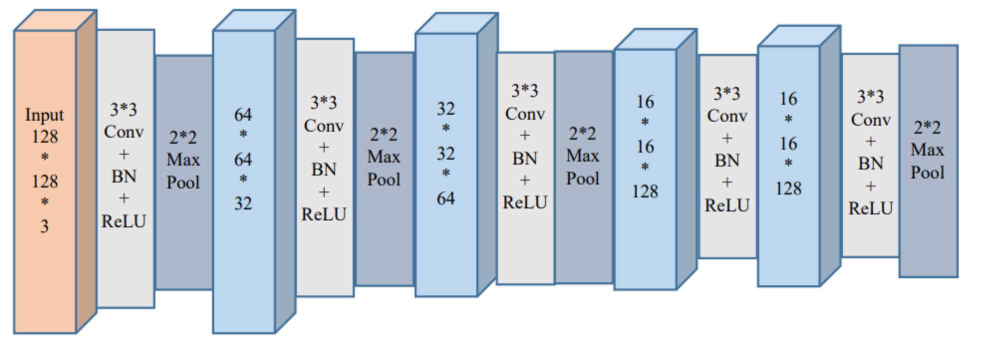
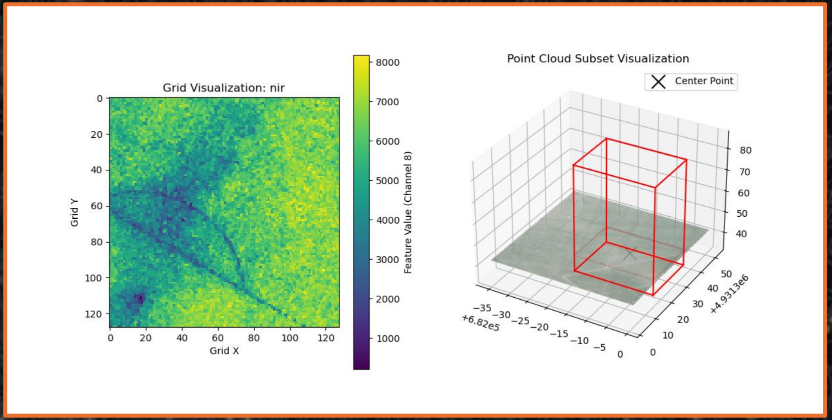


Figure 3. The architecture of the SCNN.

FEATURE IMAGE GENERATION

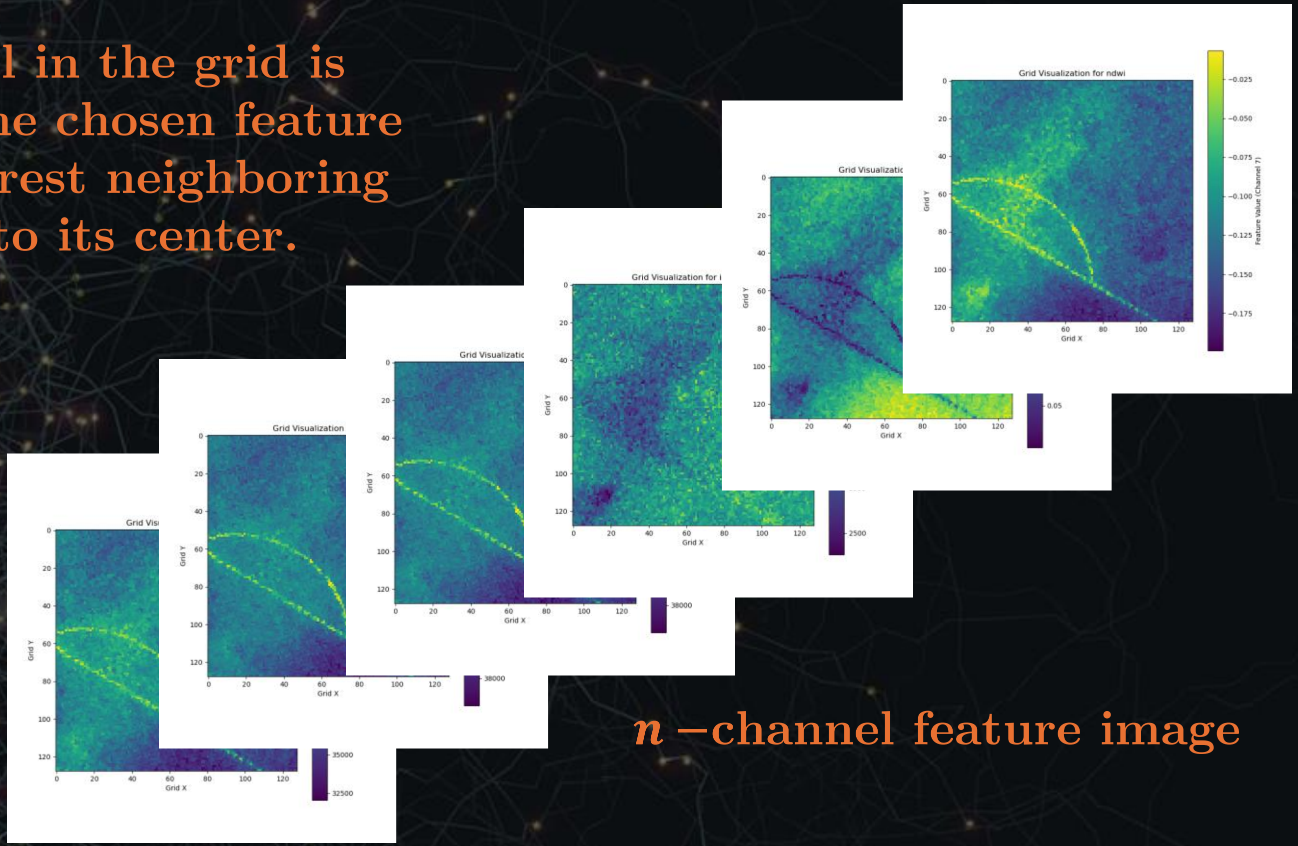


For each point in the point cloud, we generate a "feature image" - a 2D grid centered on the selected point.

Each cell in the grid is assigned the chosen feature of the nearest neighboring point to its center.

$$\begin{cases} X_{ij} = X_{p_k} - (64.5 - j) * w \\ Y_{ij} = Y_{p_k} - (64.5 - j) * w \\ Z_{ij} = Z_{p_k} \end{cases}$$

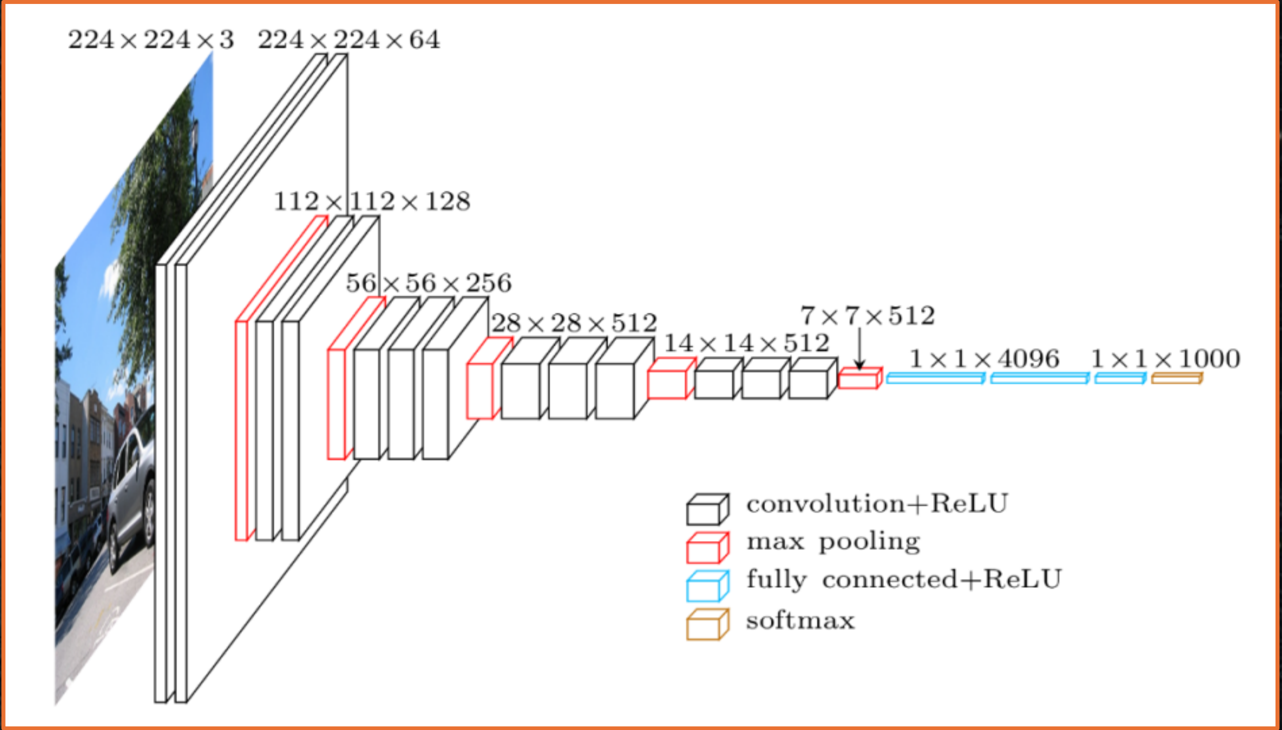
Coordinates at index i, j of the cell centers for each point cloud P_k of coordinates $(X_{P_k}, Y_{P_k}, Z_{P_k})$.
w is the width of the cell. The resolution is fixed at 128.



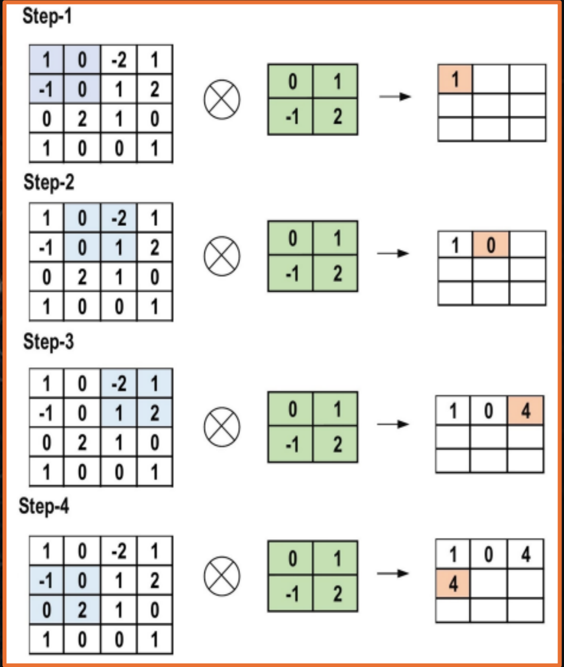
n −channel feature image

CONVOLUTIONAL NEURAL NETWORKS

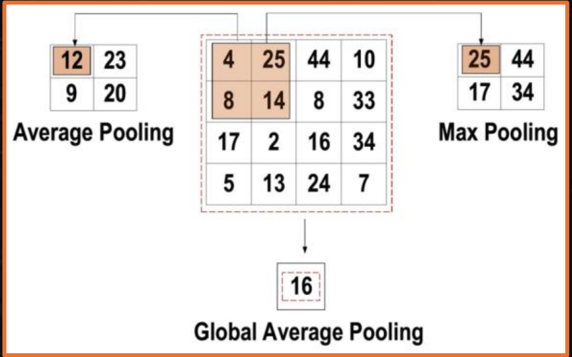
CNNs process input data using hierarchical layers of convolution, pooling, and activation functions. They downsample spatial dimensions while increasing depth to extract complex features, like a funnel.



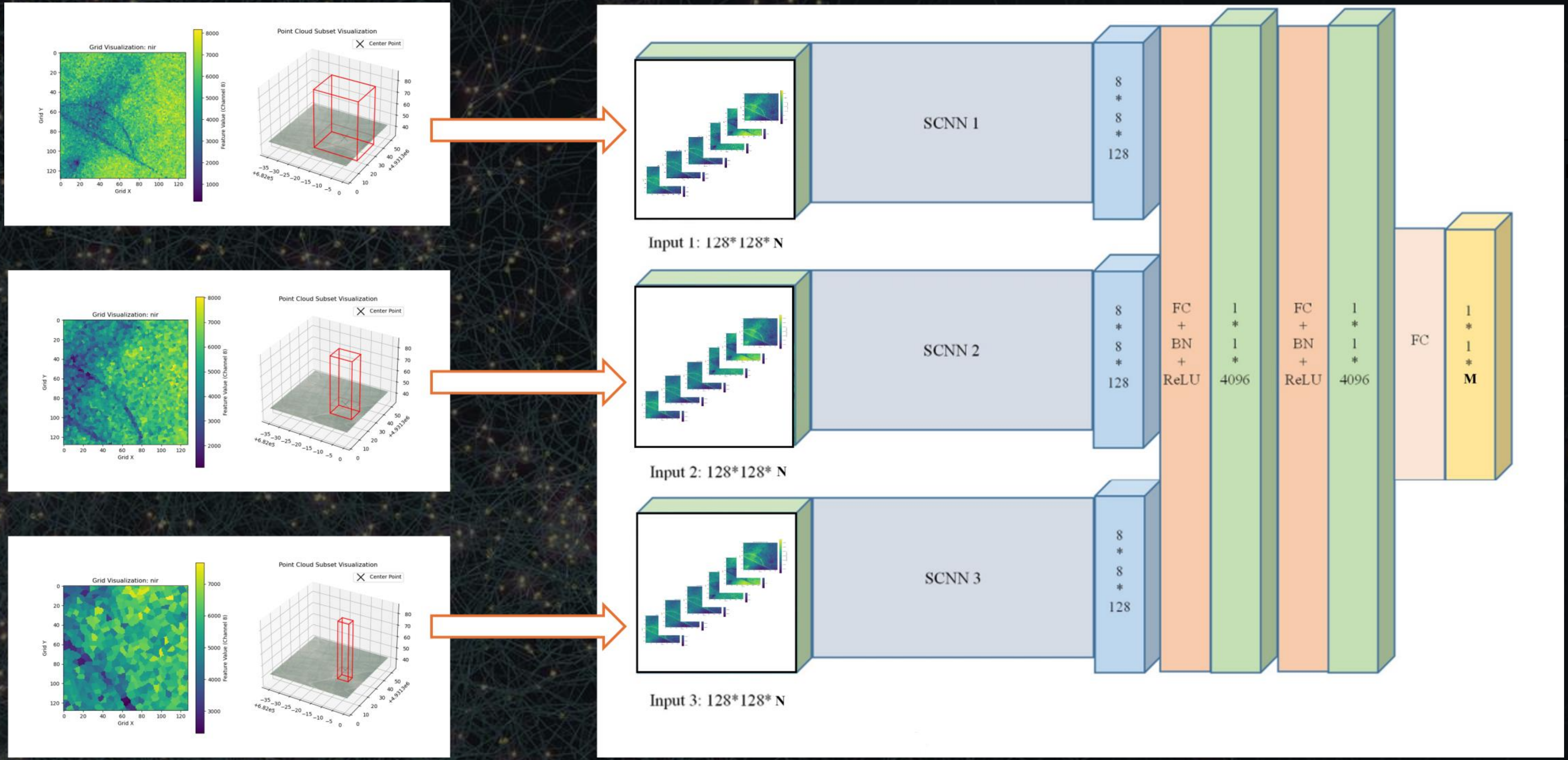
Convolutional operation



Pooling operation

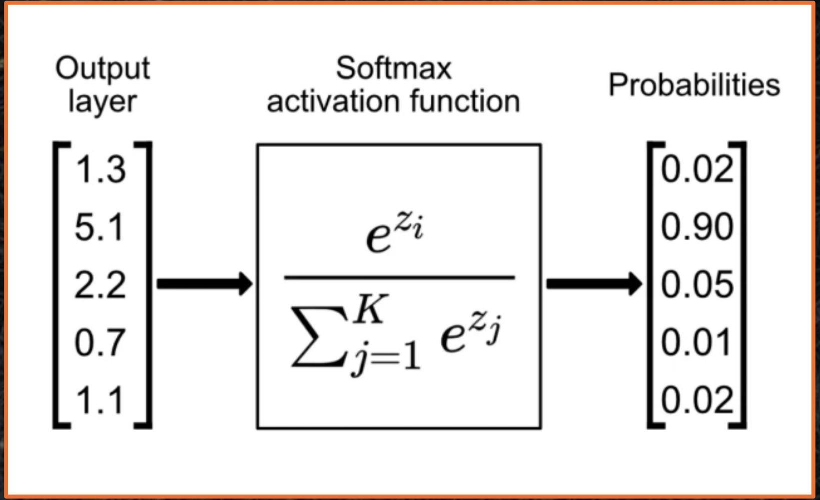


To extract information from the data at different scales, we can fuse the output of various CNNs with different window sizes



LEARNING BY OPTIMIZATION

(1) Softmax Layer



Are our predictions good?

(2) Cross Entropy Loss

$$\text{Loss} = - \sum_{i=1}^{\substack{\text{output} \\ \text{size}}} y_i \cdot \log \hat{y}_i$$

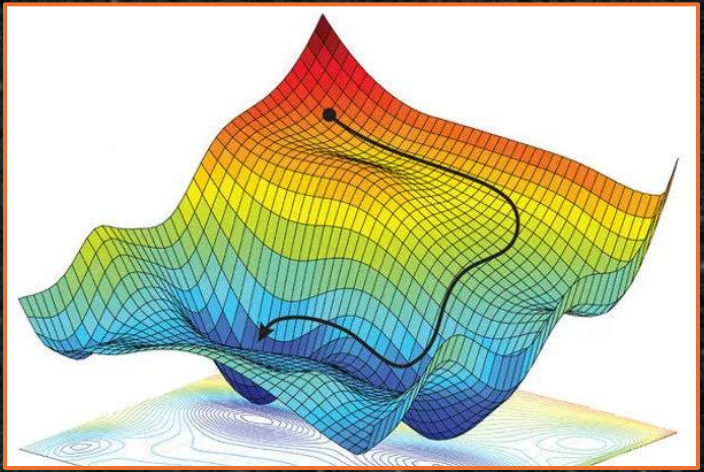
Measures the «distance» between true and predicted labels

(3) Error Backpropagation

= Computing and propagating the gradient of L backwards through the network using the chain rule

(4) Stochastic Gradient Descent

Optimizer: updates the model's parameters in the (opposite) direction of ∇L



EXPERIMENTAL SETUP

5 classes:

- Grass
- High Vegetation
- Building
- Railway
- Road
- Car

Table 3.1: Training and validation data distribution, after class rebalancing.

Class	**Training Samples**	**Validation Samples**
Grass	500,000	124,999
High Vegetation	500,000	124,999
Building	500,000	124,999
Railway	420,322	105,081
Road	500,000	124,999
Car	260,032	65,008

Dataset: subsample of approximately 2.5million labeled points from the Bologna LiDAR Dataset. Rebalanced and divided in 80% training set & 20% validation set.

Training and inference were conducted on the Leonardo HPC (Access granted by CINECA)

Code mainly written in Python with PyTorch (leveraging torch.nn module)

HYPERPARAMETERS EXPLORATION

(1) Window Sizes exploration

Table 4.3: Classification scores (averaged over all classes) for window sizes exploration. The best results are highlighted in green and correspond to the window sizes of values 10, 20, and 30 meters.

Window Sizes (m)	Accuracy	Precision	Recall	F1-Score
2.5, 5.0, 10.0	0.74	0.78	0.74	0.75
5.0, 10.0, 20.0	0.81	0.86	0.81	0.82
10.0, 20.0, 30.0	**0.82**	**0.88**	**0.81**	**0.82**
20.0, 30.0, 40.0	0.56	0.71	0.61	0.57

* WS exploration was conducted with features $I, \Delta z$, NIR, *plan*, *spher*.

(2) Selected Features exploration

Table 3.3: Feature subsets performance comparison. Additional features that led to performance improvements are shown in Green, while features that didn't optimize classification scores - and therefore were not included in the final subset - are shown in red.

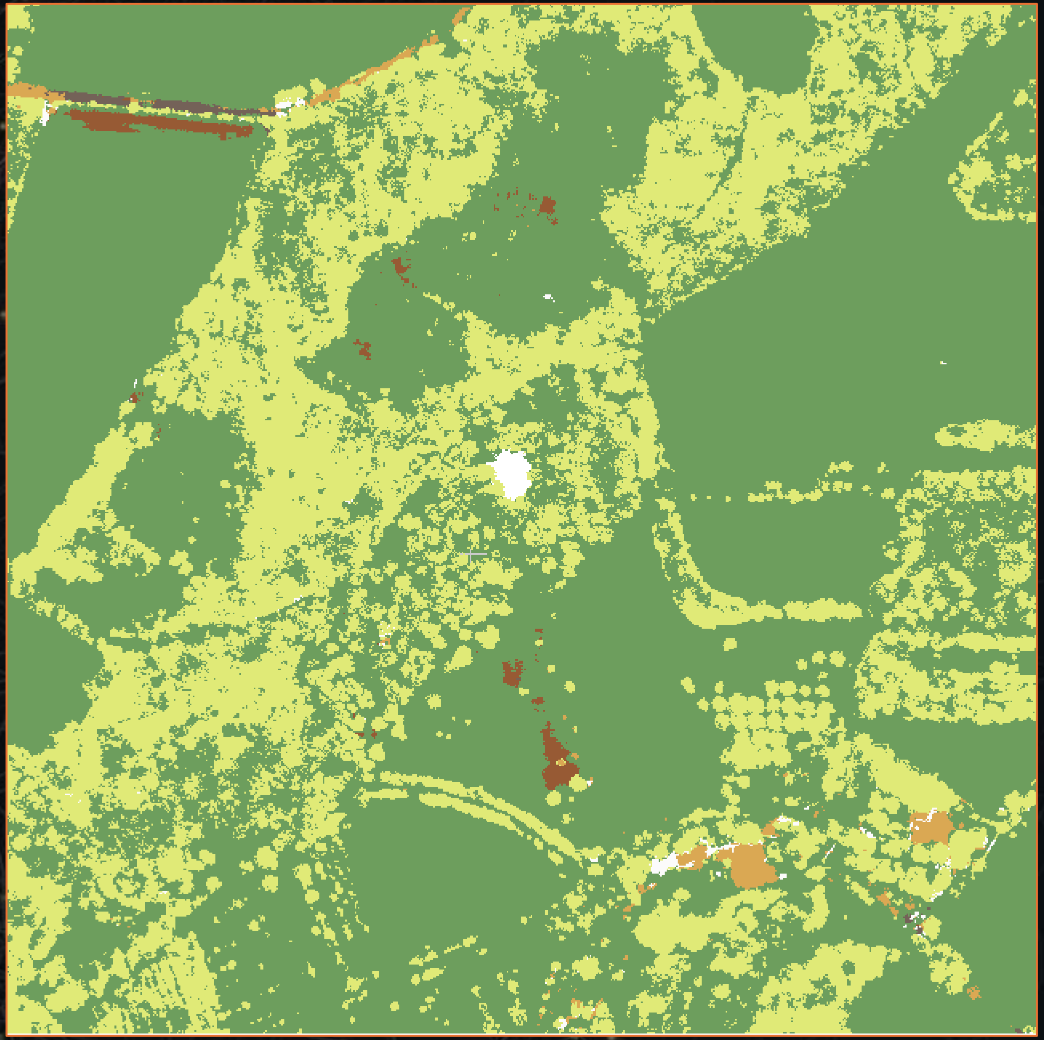
Feature Subset	Accuracy	Precision	Recall	F1-Score
Intensity	0.59	0.67	0.77	0.68
+ RGB	0.88	0.92	0.88	0.90
+ NIR	0.91	0.94	0.91	0.92
+ NDVI, SSI	0.90	0.93	0.90	0.91
+ ΔZ	0.86	0.92	0.87	0.88
+ $\boldsymbol{\lambda_1, \lambda_2, \lambda_3}$	**0.94**	**0.92**	**0.89**	**0.90**
+ θ, σ_θ^2	0.89	0.92	0.89	0.90

Confusion Matrix for the best model

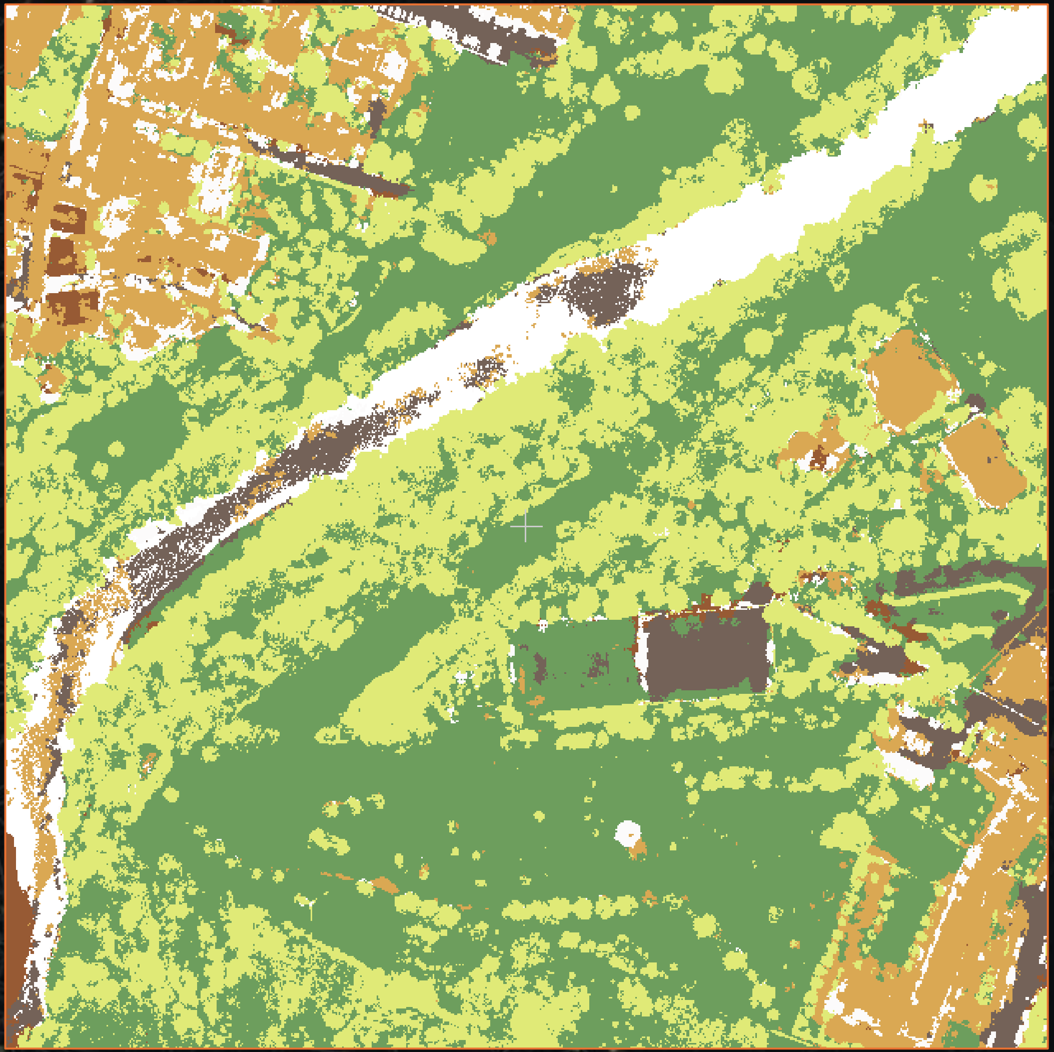
The best model was found to be the one that included the feature set: I, RGB, NIR, $\lambda_1, \lambda_2, \lambda_3$ with window sizes $10m, 20m, 30m$

EXPERIMENTAL RESULTS

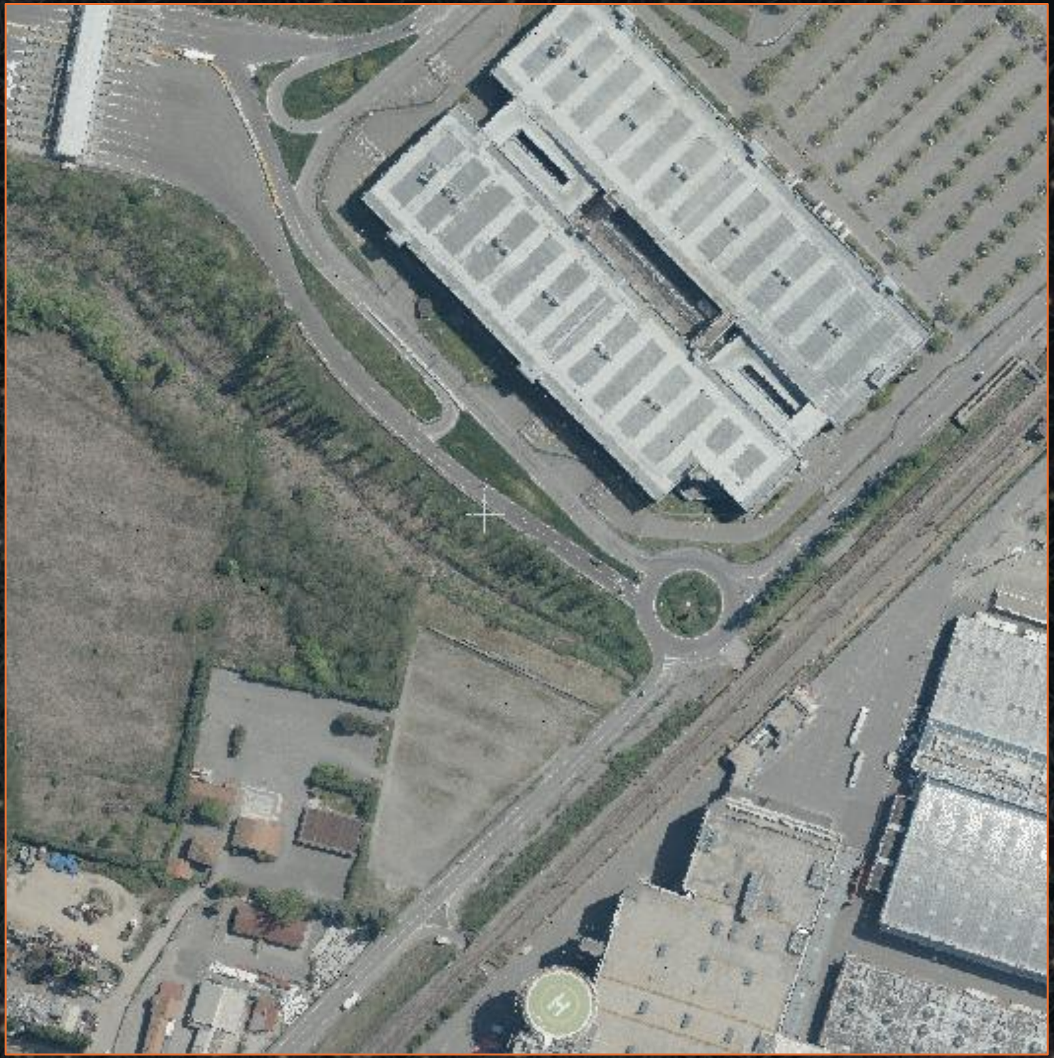
On the left, the image shows the point cloud in RGB values. On the right, the predicted labels. Buildings are orange, Grass is green, High Vegetation is yellow, Railway is brown, Roads are grey and Cars are white.

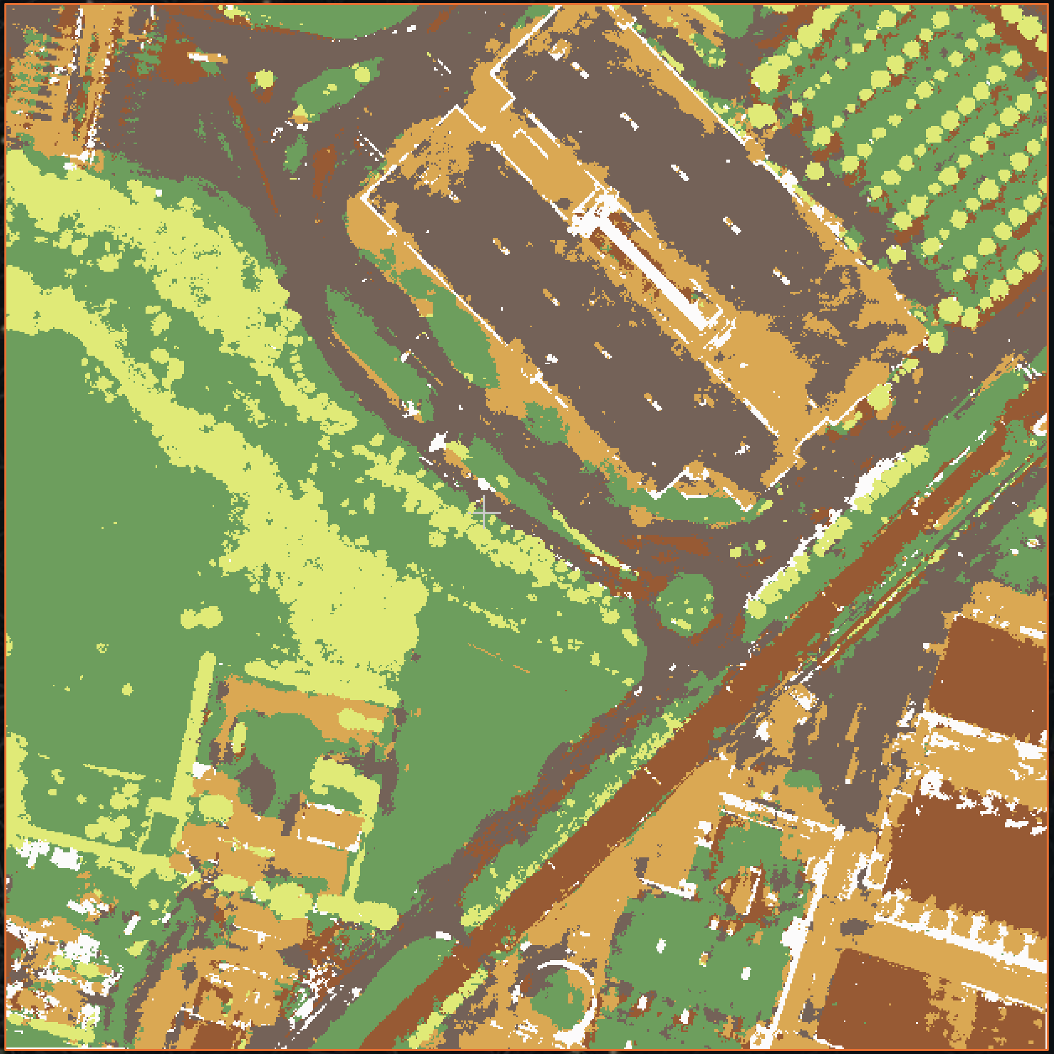


On the left, the image shows the point cloud in RGB values. On the right, the predicted labels. Buildings are orange, Grass is green, High Vegetation is yellow, Railway is brown, Roads are grey and Cars are white.



On the left, the image shows the point cloud in RGB values. On the right, the predicted labels. Buildings are orange, Grass is green, High Vegetation is yellow, Railway is brown, Roads are grey and Cars are white.





On the left, the image shows the point cloud in RGB values. On the right, the predicted labels. Buildings are orange, Grass is green, High Vegetation is yellow, Railway is brown, Roads are grey and Cars are white.

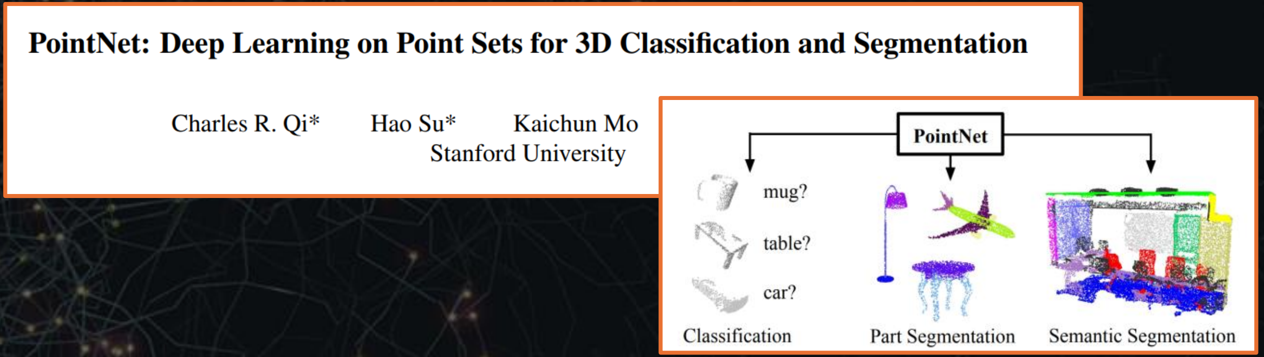
On the left, the image shows the point cloud in RGB values. On the right, the predicted labels. Buildings are orange, Grass is green, High Vegetation is yellow, Railway is brown, Roads are grey and Cars are white.

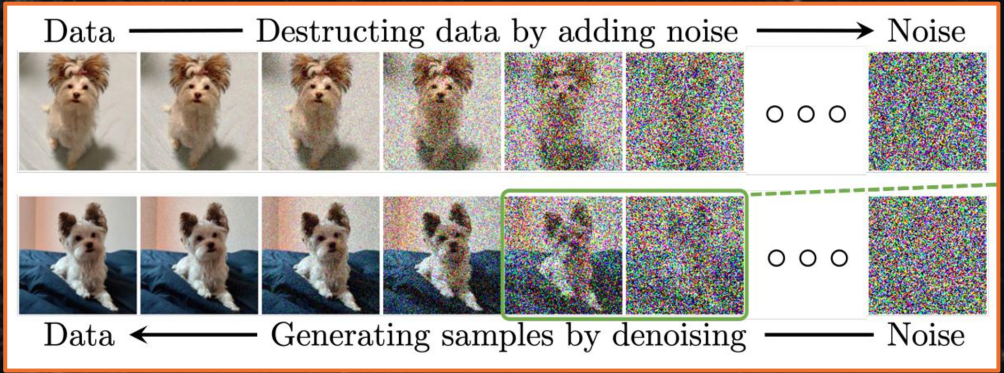
CONCLUSIONS

- Introduced the use of Artificial Intelligence for LiDAR data applications within Bologna's Digital Twin project;
- Demonstrated the effectiveness of deep learning for classification tasks, achieving high accuracy;
- Validated the projection of point cloud data onto 2D images as a viable approach to leverage pre-existing neural network architectures designed for images;
- Identified areas for further improvement and optimization.

FUTURE DIRECTIONS

- **Segmentation-based preprocessing** to enhance computational efficiency and improve classification accuracy;
- **Direct point cloud processing** using deep learning models that take raw point cloud data as input;
- **Generative AI** on Airborne LiDAR data.





Instead of the dog, we can use a 2D projection of the point cloud data

THANK YOU FOR YOUR ATTENTION