Deep learning on graph for semantic segmentation of point cloud

Alexandre Cherqui

Master in Electrical and Electronics Engineering
Master Thesis
LTS2, EPFL
Picterra

Supervisors:

Michaël Defferrard (LTS2)
Frank De Morsier (Picterra)

July 9th, 2018

Table of contents

1. Introduction
   - Motivation
   - Semantic segmentation
   - Prior art on images
   - From images to graphs
2. Model
   - Build a graph
   - Graph convolutions
   - Coarsening and pooling
   - Model architecture
3. Results
   - Available data
   - Data preprocessing
   - Performances of our model
4. Conclusion

Table of contents

1. Introduction
   - Motivation
   - Semantic segmentation
   - Prior art on images
   - From images to graphs
2. Model
   - Build a graph
   - Graph convolutions
   - Coarsening and pooling
   - Model architecture
3. Results
   - Available data
   - Data preprocessing
   - Performances of our model
4. Conclusion

Origins of the project

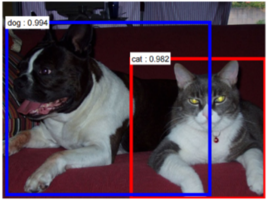
- Need for surveying the territory.
- Aerial images taken from satellites or drones.
- Can be combined to get a 3D representation and thus better recognize objects.
- But manually labeled so far.
- Collaboration with startup Picterra to automatize the task.

Aerial images from a drone.

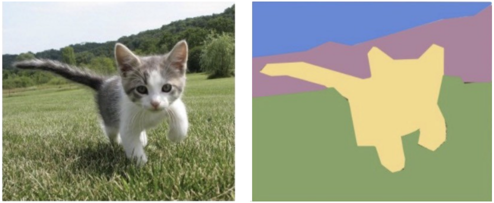
The problem of semantic segmentation

Deep learning can be used for different tasks:

- Images classification: very coarse level
- Objects detection: coarse level
- Semantic segmentation: fine level



(a) Illustration of detection



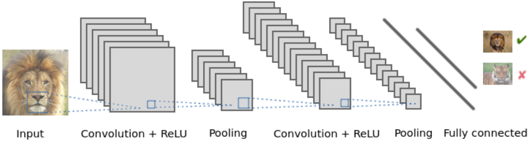
(b) Illustration of semantic segmentation

Illustrations of two problems which can be tackled with deep learning methods.

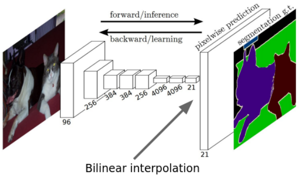
Semantic segmentation : perform a dense labelling.

Prior art on images

- Patch based parallelized: from CNN[1] to FCN [2]



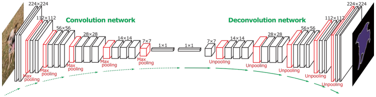
CNN architecture.



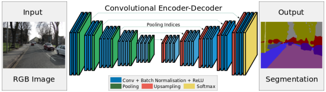
FCN architecture.

Prior art on images

- Learn the upsampling:

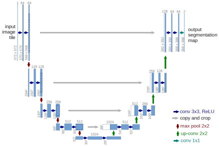


(a) DeconvNet [3]

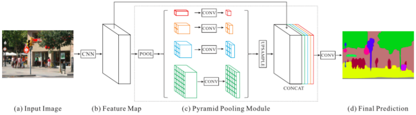


(b) Segnet [4]

- Learn at different scales:



(c) U-net [5]



(d) PSPNet [6]

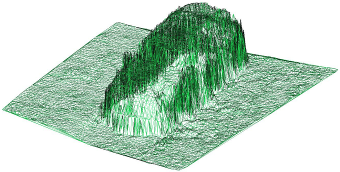
From images to graphs

- Our goal: semantic segmentation of 3D point clouds
- Some architectures directly extend what exist on images: 3D-CNN[7]
- But not well suited nor efficient (sparse data)
- $\rightarrow$ Graphs can efficiently represent these data
- $+$ Efficient computations
- $+$ Capture local neighborhood

Table of contents

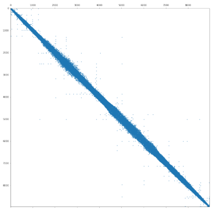
1. Introduction
   - Motivation
   - Semantic segmentation
   - Prior art on images
   - From images to graphs
2. Model
   - Build a graph
   - Graph convolutions
   - Coarsening and pooling
   - Model architecture
3. Results
   - Available data
   - Data preprocessing
   - Performances of our model
4. Conclusion

Build a graph from a cloud



Mesh generation on a car.

$$w_{i,j} = \exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)$$



Adjacency matrix of the car.

Graph convolutions: from spectral to spatial domain

$$\begin{cases} L = D - W \\ L = U \Lambda U^T \end{cases}$$

$$\begin{cases} \hat{x} = \mathcal{F}_{\mathcal{G}}\{x\} = U^T x \\ \tilde{x} = {\mathcal{F}_{\mathcal{G}}}^{-1}\{\hat{x}\} = U \hat{x} = x \end{cases}$$

For $s \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$:

$$s *_{\mathcal{G}} x = {\mathcal{F}_{\mathcal{G}}}^{-1}\{\mathcal{F}_{\mathcal{G}}\{x\} \odot \mathcal{F}_{\mathcal{G}}\{s\}\}$$

$$s *_{\mathcal{G}} x = U(U^T x \odot U^T s) = U(\mathrm{diag}(\hat{x}) U^T s)$$

$$s *_{\mathcal{G}} x = U \begin{bmatrix} \hat{x}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \hat{x}(\lambda_n) \end{bmatrix} U^T s \qquad [8]$$

Graph convolutions: from spectral to spatial domain

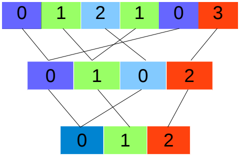
$$\forall i, \hat{x}(\lambda_i) = \sum_{j=0}^{K-1} \theta_j T_j(\lambda_i) \qquad [9]$$

$$s *_{\mathcal{G}} x = U\left(\sum_{j=0}^{K-1} \theta_j T_j(\Lambda)\right) U^T s = \sum_{j=0}^{K-1} \theta_j T_j(L) s$$

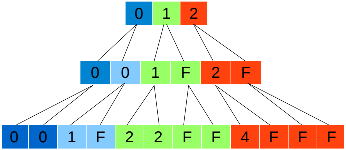
$$Ls = \begin{bmatrix} \sum_{j \in \mathcal{N}(1)} l_{1j} s_j \\ \vdots \\ \sum_{j \in \mathcal{N}(n)} l_{nj} s_j \end{bmatrix}, \qquad L^2 s = \begin{bmatrix} \sum_{k \in \mathcal{N}(1)} l_{1k} \sum_{j \in \mathcal{N}(k)} l_{kj} s_j \\ \vdots \\ \sum_{k \in \mathcal{N}(n)} l_{nk} \sum_{j \in \mathcal{N}(k)} l_{kj} s_j \end{bmatrix}$$

$$\forall p \in [\![1; n]\!], \forall k \in [\![1; N_{out}]\!], S_{out}(p, k) = \sum_{i=1}^{N_{in}} \sum_{j=0}^{K-1} \theta_{i,j}^k (T_j(L) s_i)(p)$$

Form a binary tree to ease the pooling operation



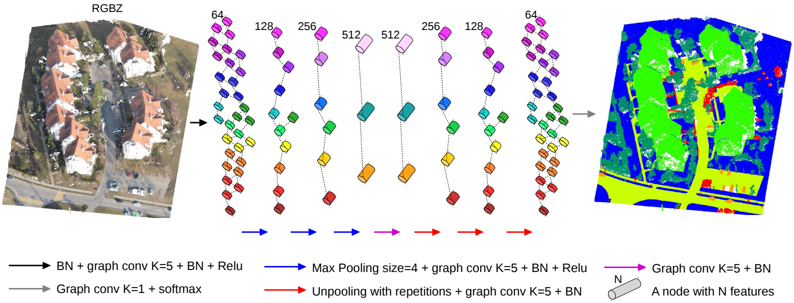
(a) Match nodes with respect to their edges weights for the different levels of coarsening



(b) Reorder the nodes so that the union of two matched neighbors from layer to layer forms a binary tree (add fake nodes F if needed)

Form a binary tree to ease the pooling operation.

Our architecture



Model architecture. Spectral distances between colors are related to spatial distances between intra- and inter-layers real nodes.

Table of contents

1 Introduction
- Motivation
- Semantic segmentation
- Prior art on images
- From images to graphs

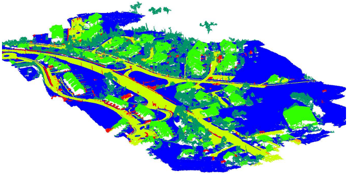
2 Model
- Build a graph
- Graph convolutions
- Coarsening and pooling
- Model architecture

3 Results
- Available data
- Data preprocessing
- Performances of our model

4 Conclusion

Available data

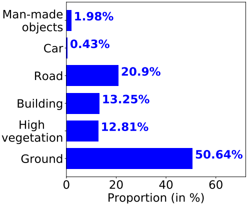
(a) Dataset (RGBZ)



(b) Dataset (labelled)

Cadastre: dataset provided by Pix4D.

From 2D to 3D thanks to photogrammetry.



Highly imbalanced class distribution.

Data preprocessing

Tiling of the dataset in tiles of $36m \times 36m$ ($48m \times 48m$ with the context):

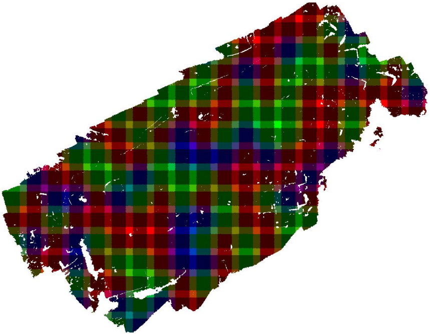
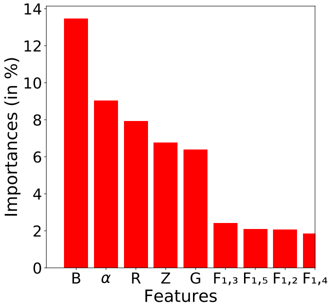


Illustration of the tiles split: the dark green tiles correspond to the training set (50%), the dark blue ones to the validation (16%) set and the dark red ones to the test set (34%). The other colors correspond to the area where the tiles overlap.

Baselines and extra features

- Random forest: 100 trees, max depth: 30, class weighted
- XGBoost: 100 trees, max depth: 5, learning rate: 0.2, weighted samples
- Extra features selected with random frorest: 3D aspect at scales 0.3m, 1.5m, 3m and 10m + angle between normals and xy plane.

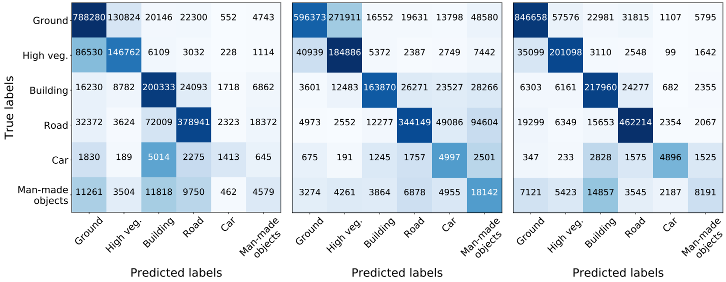


Features selection with respect to their importances for the random forest.

Performances on the cadastre with RGBZ

Performances	Overall accuracy (in %)	Mean accuracy (in %)
Random Forest	74.93	52.92
XGBoost	64.68	59.44
Our model	85.85	68.09
Majority class	47.65	16.67

Performances on the test set of the cadastre with RGBZ.



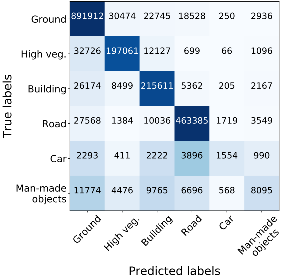
(a) Random Forest (b) XGBoost (c) Our model

Confusion matrices computed on the test set of the cadastre with RGBZ.

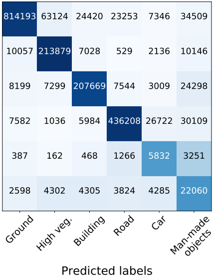
Performances on the cadastre with extra features

Performances	Overall accuracy (in %)	Mean accuracy (in %)
Random Forest	87.61	63.53
XGBoost	83.78	73.83
Our model	86.63	71.83
Majority class	47.65	16.67

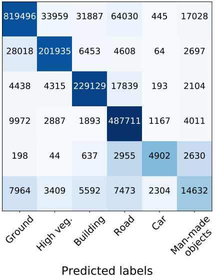
Performances on the test set of the cadastre with extra features.



(a) Random Forest



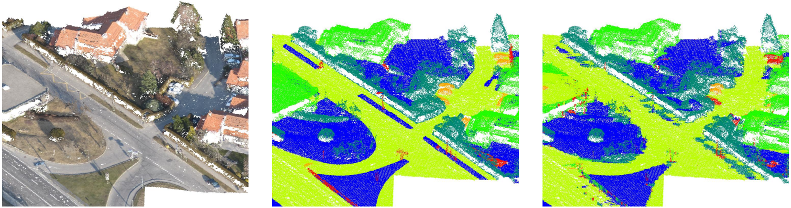
(b) XGBoost



(c) Our model

Confusion matrices computed on the test set (cadastre) with extra features.

Qualitative results on the cadastre with RGBZ



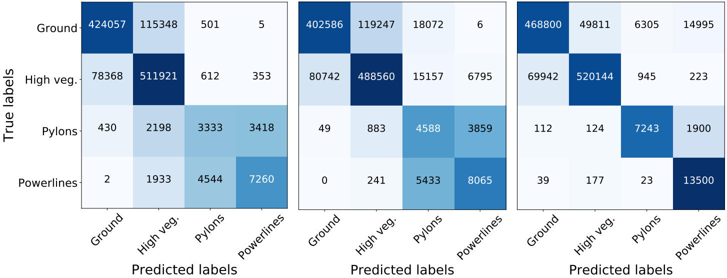
(a) Test set (b) Ground truth (c) Predictions

Qualitative results of our model on the test set.

Performances on another dataset

Performances	Overall accuracy (in %)	Mean accuracy (in %)
Random Forest	82.01	63.38
XGBoost	78.30	66.20
Our model	87.47	87.57
Majority class	51.22	25.00

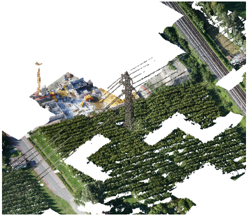
Performances on the test set from Picterra's dataset with RGBZ.



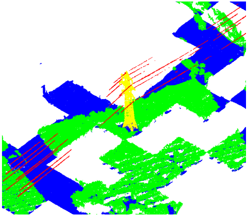
(a) Random Forest (b) XGBoost (c) Our model

Confusion matrices computed on the test set from Picterra with RGBZ.

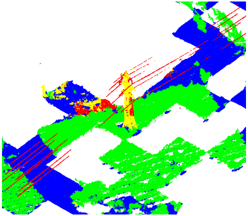
Qualitative results on test set from Picterra with RGBZ



(a) Test set



(b) Ground truth



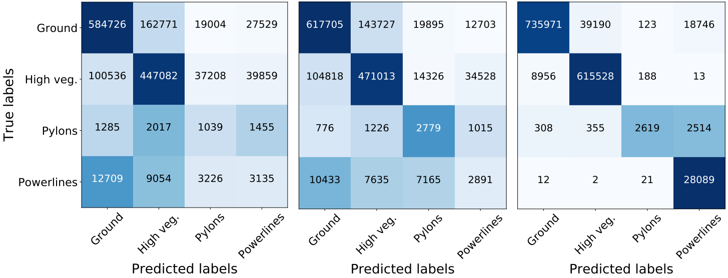
(c) Predictions

Qualitative results of our model on the test set from Picterra.

Performances inter-dataset

Performances	Overall accuracy (in %)	Mean accuracy (in %)
Random Forest	71.32	43.57
XGBoost	75.34	52.86
Our model	95.15	84.07
Majority class	54.66	25.00

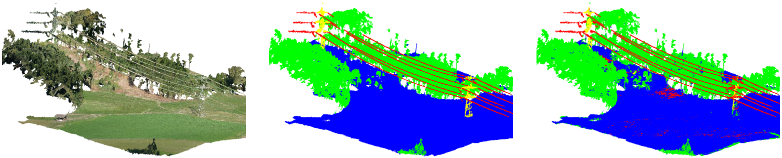
Performances on a dataset from Picterra with RGB.



(a) Random Forest (b) XGBoost (c) Our model

Confusion matrices computed on a dataset from Picterra with RGB.

Qualitative results on a dataset from Picterra with RGB



(a) Test set (b) Ground truth (c) Predictions

Qualitative results of our model on a dataset from Picterra.

Conclusion

Summing up:

- Model for semantic segmentation of aerial photogrammetry points clouds.
- Better results than random forest or XGBoost with a reduced number of features.

Future work:

- Dilated convolutions and skip connections.
- Learning on other graphs.

References

[1] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. *Object Recognition with Gradient-Based Learning*, pages 319–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[2] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE*, pages 1–12, May 2016.

[3] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

References (cont.)

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[6] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.

[7] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. pages 2670–2675, Dec 2016.

[8] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains. *CoRR*, abs/1211.0053, 2012.

References (cont.)

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375, 2016.