

Stage
EUROMETROPOLE
ECAM 2020



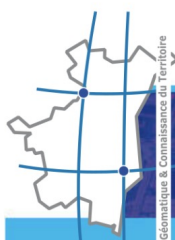
*Application du Machine Learning
à l'identification d'éléments
de mobilier urbain et de végétation
Instructions d'utilisation*

Pierre LEISY

6 août 2020

Service Géomatique et Connaissance du Territoire <https://sig.strasbourg.eu>

Données en Open Data <https://data.strasbourg.eu>



Géomatique et Connaissance du Territoire - Strasbourg Eurométropole

Strasbourg.eu
eurométropole

Le service GCT

Portail Carto

Cartothèque PDF

Données en ligne

1. Instruction d'utilisation :

Développements effectués sur un IMAC posix /Darwin /18.0.0 avec l'OS 10.14 (Mojave) et 24Go de RAM ou sur un MacBook-Pro aussi avec l'OS 10.14 et 16Go de RAM.

Installation de PYTHON avec ANACONDA

2. Les librairies PYTHON utilisées :

2.1 Meilleures pratiques :

Figurer la configuration dans un environnement spécifique pour maîtriser toutes les étapes et rester reproductible.

- **pip freeze > requirements_EMS_RN.txt**
ou
- **conda list -e > requirements_EMS_RN.txt**

On peut aussi créer un fichier **requirements_EMS_RN.in** avec les librairies nécessaires et utiliser la commande **pip-compile** pour créer effectivement le fichier **requirements_EMS_RN.txt** avec toutes les dépendances

- 1) Créer un environnement : **conda creat -n ENV python=3.7 ou 3.8**
- 2) Activation du nouvel environnement : **conda activate ENV**
- 3) Installer toutes les librairies : **conda/pip install -r requirements.txt**

2.2 Les librairies "indispensables/fonctionnant" avec PYTHON 3.7 ou 3.8

- 1) PANDAS >= 1.0
- 2) NUMPY >= 1.18
- 3) KERAS >= 2.3
- 4) MATPLOTLIB >= 3.0 et SEABORN >= 0.10
- 5) SCIKIT LEARN >= 0.22 et SCIKIT PLOT >= 0.3
- 6) LASPY version : 1.6.0
- 7) PDAL et python-pdal
- 8) plus quelques autres
- 9) GEOPANDAS >= 0.

2.3 les fonctions :

Quelques fonctions "maison" créées afin de les réutiliser dans les différentes cellules ... (ToDo : standardiser et nettoyer ces fonctions)

Pour le moment, surtout pour créer les dataframes et visualiser les résultats.

3. Les données

3.1 LASPY

Permet de lire un fichier **.LAS** ou **.LAZ** et de créer un tableau **NUMPY** en 3D (X,Y,Z) ou plus de dimensions.

3.2 Fichiers Mobiliers urbains

Pour récupérer les informations concernant les mobiliers urbains et corréler les informations avec les données LIDAR 3D.

3.2.1 Bancs Publics

Essais de caractérisation et statistiques concernant les bancs public :

Pas encore concluant, ni fonctionnel :

- 1) objets à 2 dimensions
- 2) longueur variable des entrées (ligne ou multi-ligne - nombre de points les définissant)
- 3) occupant plus de 1 sous dalle en majorité
- 4)

3.2.2 Lampadaires

3.2.2.1 Fichiers .ASC ou table .XLSX .

Lecture et écriture dans une DataFrame PANDAS par la fonction **pandas.read_csv**.

Fichier créé en même temps que les dalles (par FME) avec les 5 valeurs utiles dans le header.

3.2.2.2 Fichiers CSV

Lecture et écriture dans une DataFrame PANDAS par la fonction **pandas.read_csv**.

Problèmes de correspondances : 358 - 357 ou 355 objets ... à partir du **.SHP** il y en a bien 358 !

3.2.2.3 Fichiers SHP

Pour récupérer toutes les informations des mobiliers urbains y compris le positionnement. Lecture et écriture dans une DataFrame PANDAS par la fonction **geopandas.read_file**.

Permet de créer directement la DataFrame des lampadaire pour chaque sans étape intermédiaire de FME.

4. KERAS 1D : vectorisation

Approche classique mais qui ne permet pas d'obtenir la précision recherchée.

La reconnaissance 2D et la recherche par des images est absolument nécessaire

5. KERAS 2D : images bi-dimensionnelles

Approche plus gourmande en temps de calcul, mais permettant d'obtenir des résultats prometteurs

5.1 Réseaux Convolutionels :

5.2 Conclusions

Table des matières

1	Instruction d'utilisation :	2
2	Les librairies PYTHON utilisées :	2
2.1	Meilleures pratiques :	2
2.2	Les librairies "indispensables/fonctionnant" avec PYTHON 3.7 ou 3.8	2
2.3	les fonctions :	2
3	Les données	3
3.1	LASPY	3
3.2	Fichiers Mobiliers urbains	3
3.2.1	Bancs Publics	3
3.2.2	Lampadaires	3
3.2.2.1	Fichiers .ASC ou table .XLSX	3
3.2.2.2	Fichiers CSV	3
3.2.2.3	Fichiers SHP	3
4	KERAS 1D : vectorisation	3
5	KERAS 2D : images bi-dimensionnelles	4
5.1	Réseaux Convolutionels :	4
5.2	Conclusions	4