Deep learning pour l'extraction d'informations dans des bases de données géographiques

---

Auteur :
Massimo VENUTI

Encadrants :
Gabriel FREY Aurélie LEBORGNE

Travail d'étude et de recherche

Université de Strasbourg
UFR de Mathématique et d'Informatique
ICube

Mars 2022

Table des matières

Table des figures

Introduction

Dans un contexte où les données à analyser sont de plus en plus nombreuses et connectées, la représentation des données sous la forme de graphes pourrait faciliter l'extraction de liens et de structures communes. Les réseaux sociaux, la cybersécurité et la bio-informatique font partie des domaines où les données se modélisent naturellement sous la forme de graphes. Cependant, l'apprentissage d'une métrique de similarité entre les graphes est considéré comme un problème clé dans le cadre de tâches telles que la classification, le regroupement et la recherche de similitude.

Récemment, il y a eu un intérêt croissant pour l'utilisation de réseaux de neurones pour estimer la similarité entre graphes. L'objectif à travers leur utilisation est de plonger les graphes dans une espace cible, de telle sorte que la distance dans l'espace cible se rapproche de la distance structurelle dans l'espace d'entrée. Ce travail d'étude et de recherche vise alors à explorer et évaluer ces méthodes sur les données géospatiales de la base de données *Corine Land Cover*.

1 Travaux réalisés

1.1 Appréhension des données

La base de données de *Corine Land Cover* met à disposition deux types de données : des données raster et des données vector. Dans cette section, je décris l'exploration que j'ai réalisée sur ces deux types de données.

1.1.1 Données raster

Les données raster sont constituées d'une matrice de zones carrées, des pixels, dont la taille détermine le détail qui peut être conservé dans le jeu de données. La valeur d'un pixel peut être continue ou catégorielle représentant par exemple respectivement l'altitude ou l'utilisation des terres. Ce type de donnée est notamment utilisé pour décrire l'intérieur des entités cartographiques et pour stocker des données qui varient continuellement d'un endroit à l'autre comme l'altitude, la température ou le pH du sol.

Les données raster de *Corine Land Cover* sont constitués d'une unique bande représentant la catégorie de terrain de chaque pixel. Un code RGB est associé à chacune des catégories de terrain.

Une partie du raster complet, centrée sur Paris, peut être visualisée sur la figure 1.

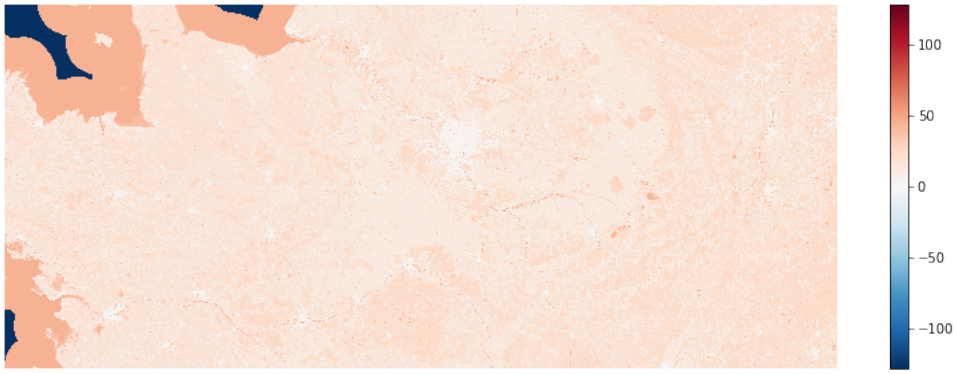

FIGURE 1 – Partie des données raster centrée sur le milieu de la France

1.1.2 Données vector

Les données vector sont constituées de coordonnées x et y pour définir la forme et l'emplacement des zones correspondant aux entités cartographiques. Un vecteur peut être un point, une ligne ou un polygone représentant par exemple respectivement un point d'intérêt, une route ou un lac. Ce type de

donnée est notamment utilisé pour décrire le centre et les bords des entités et pour stocker des détails spatiaux.

Comme pour les données raster, les données vector de *Corine Land Cover* sont constitués du type de terrain de chaque vecteur.

Une partie des vecteurs, centrée sur Paris, peut être visualisée sur la figure 2.

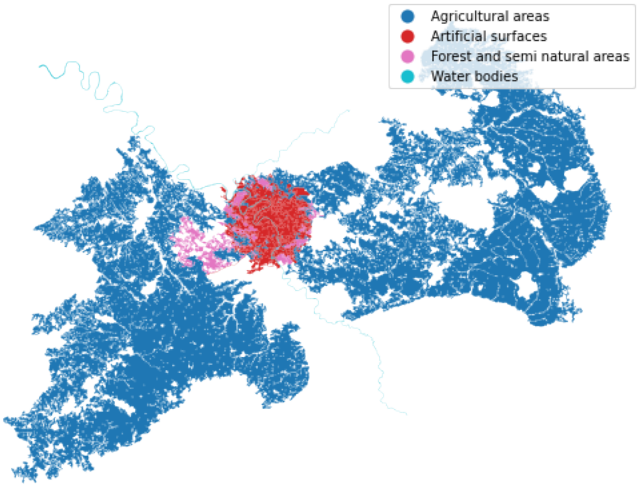

FIGURE 2 – Partie des données vectorielles centrée sur Paris

Prélèvement automatique de zones géographiques

L'extraction manuelle de zones de vecteurs à l'aide de logiciels tels que *QGIS* est fastidieuse et peu précise. C'est pourquoi il est nécessaire d'avoir un moyen d'automatiser le processus.

Le module *Python GeoPy*, intégré au sein du module *Geopandas*, permet de convertir un nom en coordonnées géographiques. À partir de ce point, nous pouvons créer une surface avec un certain rayon et extraire la partie du jeu de données complet qui lui correspond à l'aide de la fonctionnalité dédiée dans *Geopandas*. Cette dernière extrait toutes les surfaces en contact avec la zone d'extraction. Un exemple de construction d'une surface pour réaliser un prélèvement automatique est présenté sur la figure 3. Dans cet exemple, nous construisons une surface centrée sur Paris pour prélever une zone similaire à celle présentée sur la figure 2 .

Vérification des ordres de grandeur

Afin de définir si des informations relatives aux longueurs des vecteurs sont exploitables, il est nécessaire de vérifier si leur ordre de grandeur semble correcte.

Pour ce faire, nous vérifions les longueurs calculées des vecteurs sur une zone dont on connaît les dimensions. Par exemple, le lac d'Annecy. Grâce à l'outil de prélèvement automatique décrit précédemment, nous extrayons sa zone correspondante et nous calculons son périmètre : nous obtenons un écart de $2,6\%$. Cet écart est satisfaisant, c'est pourquoi nous considérons que les longueurs calculées sur les vecteurs sont correctes et peuvent être utilisées pour l'apprentissage.

1.1.3 Association raster - vector

Il aurait été intéressant d'associer les données raster aux données vector. En effet, les rasters auraient pu servir de complément d'information, telles des images, notamment grâce aux codes RGB présentes

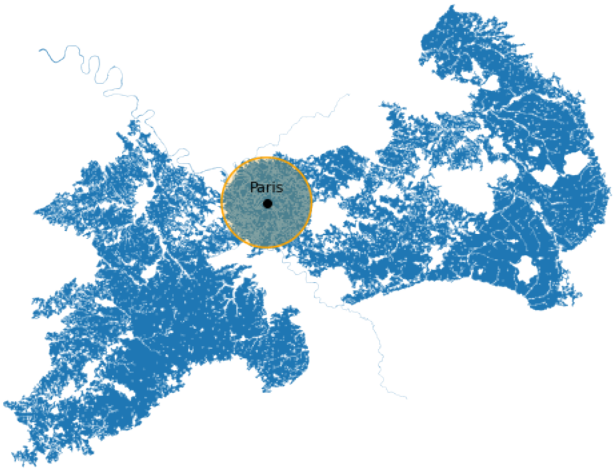

FIGURE 3 – Construction d'une surface pour le prélèvement automatique d'une zone

dans ces dernières. Cependant, une valeur RGB n'est pas associée à chaque pixel, mais à une catégorie de terrain. Cela rend donc leur intérêt moindre. Les données raster ne semblent donc finalement pas intéressante à exploiter.

1.2 Construction des jeux de données pour la classification de graphes

Dans l'objectif de faire de l'apprentissage pour la classification de graphes, il convient de construire et de préparer des jeux de données adéquats en amont. Dans cette section, je décris la construction des différents jeux de données qui seront utilisés pour l'apprentissage.

1.2.1 Capitales et campagnes

Dans ce premier jeu de données, nous voulons extraire de la base de données *Corine Land Cover* des capitales et des campagnes, dans le but de les classifier.

Sélection des capitales et des campagnes

J'ai alors recensé des capitales d'Europe de tailles similaires : Berlin, Madrid, Helsinki, Paris, Rome, Varsovie, Bucarest, Londres, Stockholm. Pour les campagnes, j'ai collecté manuellement 36 points dans des grandes zones rurales. Les campagnes étant de manière générale plus vastes et homogènes que les villes, ces points ont été collectés de manière aléatoire sur le territoire recouvert par la base de données *Corine Land Cover*.

Choix de la taille d'extraction

Le choix de la taille de la zone d'extraction est crucial pour construire un jeu de données en adéquation avec ce que l'on veut collecter. Dans le cas des capitales, l'objectif est d'extraire des zones qui représentent au mieux les espaces urbains. On ne veut donc pas inclure de terrains définissant des zones rurales. Il faut donc choisir une taille de zone assez grande pour avoir un nombre d'instances

suffisant pour l'apprentissage, mais pas trop grande pour ne pas récupérer la périphérie des plus petites villes.

C'est pourquoi j'ai relevé la répartition des catégories de terrain pour plusieurs tailles d'extraction. L'objectif est de choisir celle dont la répartition correspond le mieux au type de région qu'on souhaite extraire : urbain pour les capitales et rural pour les campagnes. Comme les campagnes sélectionnées sont plus vastes que les villes, la contrainte sera exclusivement imposée par les capitales. La figure 4 présente la répartition des catégories de terrain pour des tailles d'extraction de 10 à 100 km^2 pour les capitales.

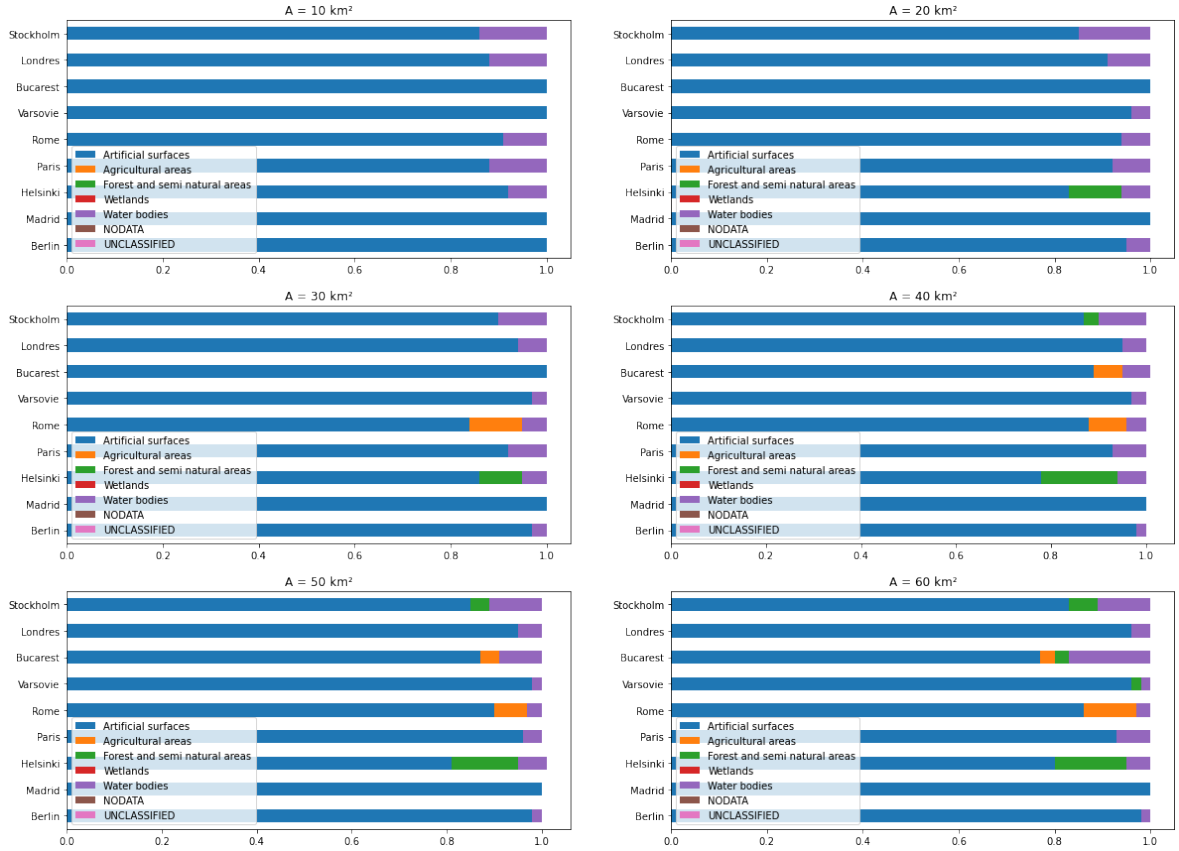

FIGURE 4 – Répartition des types de terrains en fonction de la taille d'extraction A pour la création du jeu de données des capitales

La taille d'extraction retenue est alors de 20 km^2. Cette dernière permet de maximiser le taux de parcelles urbaines dans le jeu de données tout en conservant un nombre important d'instances. Il sera peut-être nécessaire, en fonction des résultats de l'apprentissage, d'augmenter ce rayon ou d'utiliser plus de villes pour avoir plus d'instances.

1.2.2 Communes de France

Dans ce second jeu de données, nous souhaitons extraire de la base de données *Corine Land Cover* environs 5000 communes de tailles similaires, dans le but de les classifier de manière non supervisée. Pour ce faire, nous utilisons la base de données des communes de France.

Sélection des communes

L'objectif est de sélectionner environs 5000 communes avec des tailles similaires. Remarquons que 75% des exemples présents dans le jeu de données des communes de France ont une surface inférieure à 2000 hectares. La figure 5 montre la répartition des surfaces des communes, parmi celles ayant une surface inférieure à 3000 hectares. Nous sélectionnons des communes dont la taille n'est pas trop petite pour maximiser les chances d'y trouver des zones urbaines dans la base de données *Corine Land Cover*. Nous prenons donc les communes avec une surface comprise entre 1900 et 3000 hectares. Il s'agit donc de petites communes rurales.

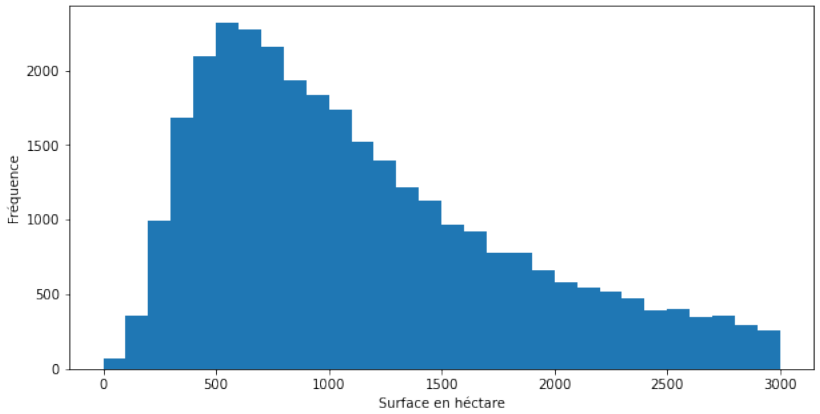

FIGURE 5 – Répartition de la surface des communes parmi celles ayant une surface inférieure à 3000 hectares

Extraction des zones géographiques

Afin d'extraire les zones géographiques correspondant aux communes, nous pouvons directement appliquer un masque dans la base de données *Corine Land Cover* à partir des communes de la base de données des communes. La figure 6 présente le découpage de la commune Eyne par ce procédé.

Cependant, nous pouvons observer qu'un tel découpage peut séparer des aires de *Corine Land Cover* en plusieurs parties distinctes. On souhaite alors les décomposer en instances séparées. La figure 7 décrit ce procédé pour la commune d'Eyne.

Enfin, le découpage fait apparaître des aires résiduelles aux extrémités de la carte. Ces dernières ne sont pas représentatives de la zone que l'on souhaite extraire et peuvent donc constituer un biais pour l'apprentissage. Nous souhaitons donc les supprimer. Nous calculons alors la proportion de la taille conservée des surfaces et la proportion de la taille totale des surfaces après découpage. Nous pourrons ainsi fixer par la suite un seuil pour ne conserver que les surfaces représentatives. Ces proportions sont présentées pour la commune d'Eyne sur la figure 8, et la figure 9 montre un exemple de suppression de ses aires résiduelles en fixant les seuils respectifs à 40% et 1% .

1.3 Construction des graphes

Un graphe représente une zone géographique de la manière suivante. Un noeud du graphe correspond à une surface dans le jeu de données de *Corine Land Cover*. Il contient à la fois des informations géométriques telles que l'aire et le périmètre de la surface, et des informations sémantiques telles que le type de terrain associé à la surface. Deux noeuds sont alors voisins si leurs surfaces correspondantes dans le jeu de données sont voisines. Il convient donc dans un premier temps de définir la notion de voisinage.

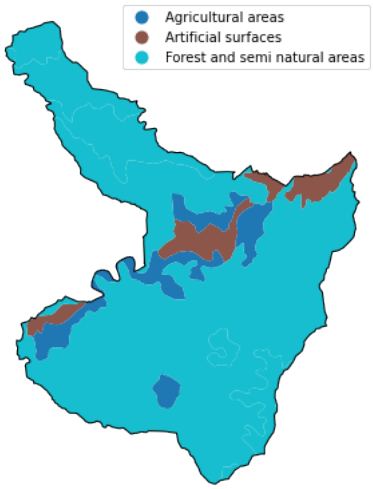

FIGURE 6 – Extraction de la commune d'Eyne dans les données de *Corine Land Cover* à partir de celles des communes de France

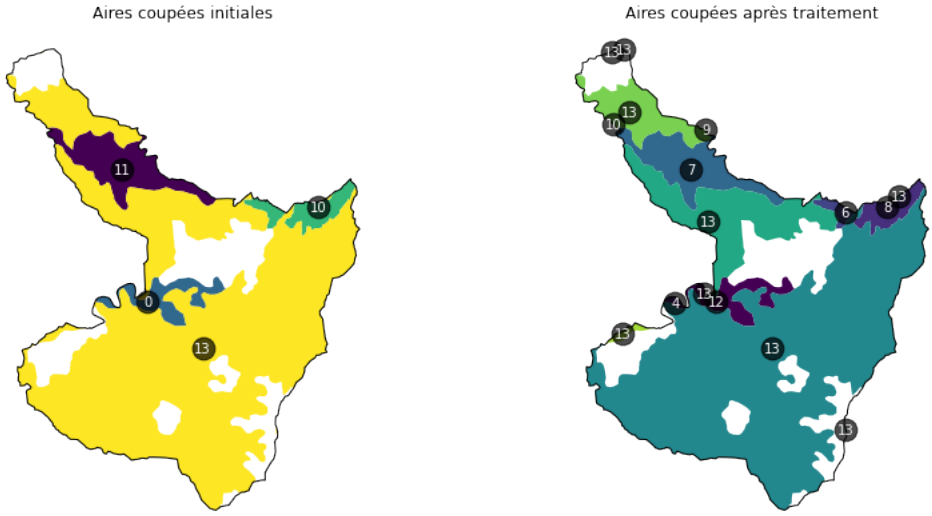

FIGURE 7 – Redéfinition des aires coupées après extraction pour la commune d'Eyne

L'idée est de vérifier si des surfaces voisines dans les données de *Corine Land Cover* se touchent forcément, ou si elles peuvent être légèrement éloignées. Pour ce faire, on calcule la distance entre paire de surfaces et on vérifie si, en dessous d'un certain seuil, il existe ou non une autre surface entre les deux. Si tel est le cas pour l'intégralité des exemples tester, on peut considérer que les surfaces voisines se touchent toutes. Sinon, il faudra considérer une éventuelle distance entre paire de surfaces voisines. Sur 12 zones extraites du jeu de données initiale, avec en moyenne 600 surfaces par zone et en fixant un seuil à 100 mètres, il n'y a aucune paire de surfaces dans ce deuxième cas.

Nous définissons donc des surfaces voisines dans les données de *Corine Land Cover* comme des surfaces qui se touchent, et une arête du graphe met en évidence cette relation de voisinage. Le calcul du voisinage pour créer le graphe se fait alors naïvement en récupérant pour chaque surface des données, toutes les surfaces qui la touchent. Des informations supplémentaires peuvent ensuite être portées par

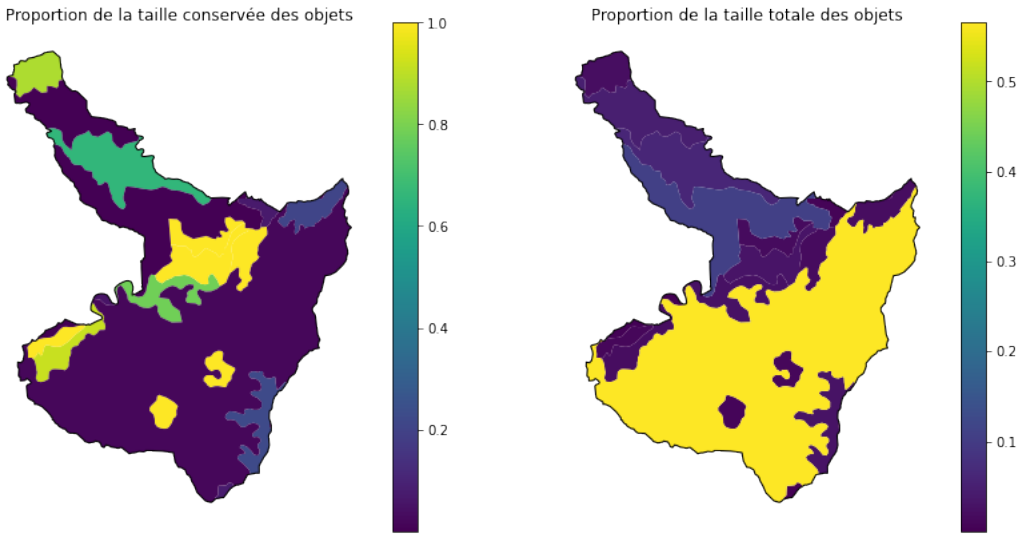

FIGURE 8 – Proportions de la taille conservée et de la taille totale des objets après extraction pour la commune d'Eyne

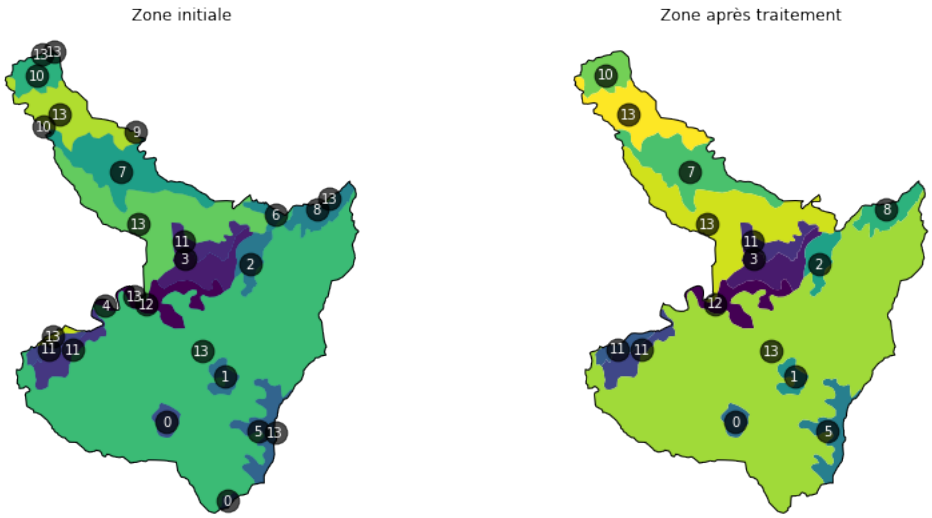

FIGURE 9 – Exemple de suppression des aires résiduelles après extraction pour la commune d'Eyne

les arrêtes du graphe comme la taille de la frontière entre deux surfaces voisines. Puisque ces frontières peuvent être discontinues, le nombre de frontières, la taille de la plus grande frontière et la somme des tailles des frontières sont également des informations portées par les arrêtes. La figure 10 montre un exemple de graphe construit avec cette méthode sur la commune d'Eyne.

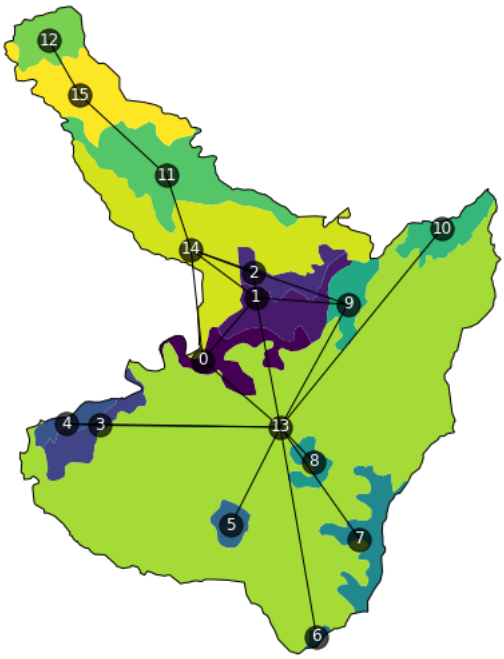

FIGURE 10 – Graphe associé à la commune d'Eyne

2 Travaux futurs

La première partie de ce projet s'est principalement articulé autour de l'appréhension et la préparation des données pour construire les graphes. La seconde partie consistera en la mise en oeuvre de l'apprentissage automatique pour la classification des graphes à partir de ce travail.

La prochaine étape sera alors de s'intéresser aux méthodes de plongement de graphes basées sur l'utilisation de réseaux de neurones. Pour ce faire, un état de l'art sera nécessaire pour découvrir, comprendre et choisir les méthodes les plus adaptées à notre problématique. Il s'agira ensuite de les implémenter, les évaluer et les comparer.

On s'intéressera en priorité aux données des communes de France, l'objectif étant d'appliquer les méthodes retenues à des échelles différentes, pour classifier des zones de plus en plus grandes et recouvrir finalement l'entièreté du territoire français. Cela conduira donc à la création de nouveaux jeux de données, dont la taille des zones géographiques, et donc des graphes, sera très variable.

Bibliographie

[1] Hongyun Cai, Vincent Wenchen Zheng et Kevin Chen-Chuan Chang. "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications". In : *IEEE Transactions on Knowledge and Data Engineering* 30 (2018), p. 1616-1637.

[2] *CORINE Land Cover, User Manual*. Copernicus Land Monitoring Service. 2021.

[3] Guixiang Ma et al. "Deep Graph Similarity Learning: A Survey". In : *ArXiv* abs/1912.11615 (2021).

[4] Annamalai Narayanan et al. "graph2vec: Learning Distributed Representations of Graphs". In : *ArXiv* abs/1707.05005 (2017).