Classification d'objets urbains à partir de données LiDAR 3D terrestre par Deep-Learning

Younes Zegaoui[1, 2], **Marc Chaumont**[1, 3], **Gérard Subsol**[1], **Philippe Borianne**[4], **Mustafa Derras**[2]

1. *LIRMM, Univ. Montpellier, CNRS, France*
   zegaoui@lirmm.fr,subsol@lirmm.fr
2. *Berger-Levrault, Toulouse, France*
   mustafa.derras@berger-levrault.com
3. *Univ. Nîmes, France*
   marc.chaumont@lirmm.fr
4. *AMAP, Univ. Montpellier, CIRAD, CNRS, INRA, IRD, Montpellier, France*
   philippe.borianne@cirad.fr

RÉSUMÉ. *La détection automatique d'objets urbains reste un défi pour les gestionnaires de ville. Les approches existantes en télédétection comprennent l'utilisation d'imagerie aérienne ou de LiDAR pour cartographier une scène. Nous voulons mettre à l'épreuve les méthodes 3D Deep Learning pour aborder le problème de la détection d'objets urbains. Dans cet article, nous présentons les résultats de plusieurs expériences sur la classification des objets urbains avec le réseau PointNet.*

ABSTRACT. *Automatic urban object detection remains a challenge for city management. Existing approaches in remote sensing include using aerial images or LiDAR to map a scene. We then wanted to make use of the rise of 3D Deep-Learning methods to tackle the issue of urban object detection with the help of terrestrial LiDAR acquisition. In this paper we present result of several experiments on urban object classification with the PointNet network.*

MOTS-CLÉS : *LiDAR, deep-learning, nuages de points, objets urbains*

KEYWORDS: *LiDAR, deep-learning, 3D points clouds, urban objects*

1. Présentation et contexte

La reconnaissance de formes est entrée dans une nouvelle ère avec le développement des algorithmes d'apprentissage en profondeur, ou Deep-Learning, dans la communauté universitaire comme dans le monde industriel, grâce aux avancées majeures réalisées ces 5 dernières années essentiellement dans le traitement des images 2D. Les méthodes d'apprentissage en profondeur s'étendent maintenant à de très nombreux domaines scientifiques ou industriels,qui pourraient potentiellement tous bénéficier des progrès récents de cette technologie LeCun *et al.* (2015).

Actuellement, la plupart des recherches sur l'apprentissage en profondeur proviennent de l'imagerie 2D où elles ont permis des gains de performances énormes dans la classification d'images, visibles en particulier dans le challenge ILSVRC Russakovsky *et al.* (2014). Cette augmentation soudaine des performances a permis à des algorithmes de détection d'objets robustes en temps réel tel que Faster R-CNN Ren *et al.* (2015) d'émerger. Cependant, leur généralisation à des données 3D n'est pas une tâche simple. Cela est particulièrement vrai dans le cas des nuages de points 3D où les informations ne sont pas structurées comme dans les maillages 3D.

Les capteurs LiDAR terrestres, qui sont de nos jours généralement monté sur des appareils mobiles tels que des voitures, peuvent être utilisés pour effectuer une acquisition dynamique d'une scène entière telle qu'une ville ou une agglomération. Cela peut facilement être réalisé avec une voiture équipée de LiDAR traversant le réseau routier d'une ville, générant un nuage de points 3D correspondant à la ville dans son entièreté Anguelov *et al.* (2010). Par rapport à un simple enregistrement vidéo, une telle acquisition donne une information contextuelle 3D et fournit des mesures précises de profondeur.

A partir d'une acquisition LiDAR, il serait possible de détecter les objets d'une scène 3D et d'utiliser cette information dans la gestion d'une agglomération. Par exemple, connaître précisément le nombre d'arbres ou de poteaux présents et leurs emplacements aiderait grandement à mettre à jour les bases de données d'objets urbains, les localisant précisément et en gardant une trace de leur statut. D'autre part, il est particulièrement intéressant de suivre des objets en constante évolution, l'exemple le plus notable étant les arbres. La plupart des appareils LiDAR mobiles sont aussi équipés d'émetteurs GNSS (Global Navigation Satellite System) qui permettent un géoréférencement des données lors de l'acquisition. Ainsi, tout objet urbain détecté dans le nuage de points peut être projeté sur un SIG (Système d'Information Géographique) existant. Néanmoins, pour que des algorithmes de localisation basé Deep-Learning fonctionnent, nous devons au préalable nous assurer que leur contreparties en classification fournissent des résultats acceptables.

Dans cet article, nous proposons d'évaluer un réseau de Deep-Learning 3D récent et performant sur une tâche de classification de nuages de points. Après

un bref état de l'art du Deep-Learning en 3D dans la section 2, nous présentons notre méthodologie de classification dans la section 3. Dans la section 4, nous donnons les résultats expérimentaux et évaluons comment le réseau neuronal reconnaît un nuage de points 3D comme un objet urbain.

2. Deep-Learning en 3D

Comme précédemment indiqué, il n'existe pas de moyen simple de généraliser les méthodes de classification des images 2D aux nuages de points 3D. Ceci est dû au fait qu'un nuage de points 3D est une structure de données non ordonnée contrairement à une image où les pixels sont ordonnés. Il n'y a pas de corrélation entre l'ordre dans lequel les points sont classés dans le nuage, qui est simplement une liste de sommets (X, Y, Z), et les informations qu'ils encodent. Le nuage de points reste le même quelle que soient les permutations appliquées à la liste, alors que dans les images 2D, les valeurs des pixels voisins sont plus ou moins corrélées entre elles. Afin de contourner ces problèmes, les premiers algorithmes de classification 3D ont utilisé des structures de données intermédiaires pour représenter les nuages de points.

Les méthodes existantes peuvent être divisées en trois sous-catégories selon la structure de données intermédiaire qu'elles utilisent :

– Les méthodes basées sur les voxels ont été les premières à fournir des résultats significatifs. Elles utilisent des algorithmes de voxellisation pour transformer les nuages de points en images volumiques, comme celles utilisées dans l'imagerie médicale. Les convolutions spatiales peuvent alors être appliquées de la même manière que pour les images 2D avec des noyaux définis sur des voxels au lieu des pixels. Il est donc possible d'utiliser une même architecture de réseau neuronal convolutif (CNN) sur les données voxellisées en adaptant légèrement les couches afin qu'elles puissent traiter correctement la troisième dimension. VoxNet Maturana, Scherer (2015) utilise une grille d'occupation pour l'étape de voxellisation et un CNN à 3 couches pour la tâche de classification. Les auteurs de ORION "ORION" (s. d.) montrent que l'ajout d'une prédiction d'orientation à la tâche de classification aide le réseau à obtenir de meilleurs résultats. OctNet Riegler *et al.* (2017) propose une méthode de voxellisation des nuages en octree pour résoudre un des problèmes majeurs de ces méthodes, à savoir la consommation de mémoire graphique (VRAM) due à la taille en mémoire nécessaire pour avoir une résolution volumique suffisante. À compter de 2018, les performances des méthodes basées sur les voxels sont globalement inférieures aux deux autres.

– Les méthodes multi-vues génèrent plusieurs angles de vue autour de l'objet 3D avec une caméra virtuelle et synthétisent pour chaque angle une image 2D de l'objet. Les images 2D peuvent être en niveaux de gris, en RGB classique, une carte de profondeur ou une silhouette (image binaire). Les images 2D sont ensuite traitées par un CNN classique pour obtenir la classe. Le réseau

PairWise (Johns *et al.*, 2016) propose un réseau capable de prédire le meilleur point de vue suivant selon une vue initiale. En termes de précision, ces méthodes donnent les meilleurs résultats. Par exemple RotationNet Kanezaki *et al.* (2018) définit actuellement un nouvel état de l'art sur le jeu de données de Princeton ModelNet40. Cependant, elles supposent qu'il est possible de synthétiser des images 2D à partir de tous les angles possibles autour de l'objet. Cela est généralement vrai pour des maillages CAO qui sont triangulés et complets. Par contre les LiDAR fournissent des nuages de points qu'il faut trianguler, ce qui n'est pas facile du fait des fortes variations de densité des points et des nombreuses occultations.

– Enfin, les approches basées uniquement sur les points ne nécessitent pas de structure de données intermédiaire. Apprendre directement sur les coordonnées des points est une option qui a réellement démarré en 2017 avec l'apparition de PointNet Charles *et al.* (2017). Sa particularité est de ne pas utiliser de structure intermédiaire entre le nuage de points et le réseau : les points sont directement passés au réseau. Il y a deux problèmes principaux à aborder lors de l'apprentissage direct à partir des coordonnées des points : le réseau doit être invariant par rapport à l'ordre des points et au repère géométrique des nuages. Vous trouverez plus de détails sur la manière dont PointNet gère ces problèmes dans la section suivante. PointNet ++ Qi *et al.* (2017) et Engelmann *et al.* (2017) ont proposé d'ajouter des couches supplémentaires pour permettre à PointNet d'utiliser les informations de contexte spatial. KD-net Klokov, Lempitsky (2017) propose d'utiliser un arbre KD qui est généralement utilisé pour réduire les coûts de calcul, pour ajouter une information de contexte tout en apprenant à partir des coordonnées des points.

Le fait que PointNet n'ait pas besoin d'une structure de données intermédiaire et qu'il fournisse de très bons résultats malgré une architecture relativement simple comparée aux autres réseaux basés sur VGG ou ResNet, en fait le candidat le plus approprié pour nos expériences de classification.

3. Description du réseau PointNet

Le but de l'architecture PointNet Charles *et al.* (2017) est de calculer une fonction symétrique afin que les résultats produits par le réseau soient invariants par rapport à l'ordre des points dans le nuage. Il comprend également un mini-réseau appelé T-Net pour gérer le problème de normalisation des coordonnées.

Le T-Net est essentiellement une version miniaturisée de PointNet dont l'objectif est de prédire les coefficients d'une matrice 3x3. Ces coefficients correspondent à une transformation affine qui est appliquée au nuage de points en entrée afin d'aligner les données sur un espace canonique.

Une fois que les points sont "alignés" par le T-Net, ils passent par le module d'extraction de caractéristiques. Ce module consiste en une série de couches

MLP (Multi Layer Perceptron) où des vecteurs de caractéristiques sont extraits de chaque point, suivis par une opération d'agrégation regroupant ces caractéristiques en un vecteur global. Ce qui correspond au calcul d'un espace d'entité invariant à l'ordre des points. Le vecteur global passe ensuite dans un classificateur qui permet de prédire la classe du nuage de points.

4. Expériences de classification

Nous avons décidé d'évaluer les performances de la *classification* d'objets urbains, numérisés par acquisition LiDAR, avec l'utilisation du réseau PointNet. Chaque nuage de points 3D, c.-à-d. une liste de coordonnées (X, Y, Z), contient un seul objet urbain. Le but de ces expériences est d'évaluer la capacité du réseau à prédire la classe d'un objet en traitant uniquement ses coordonnées. Nous allons décrire brièvement les expériences dans la section suivante.

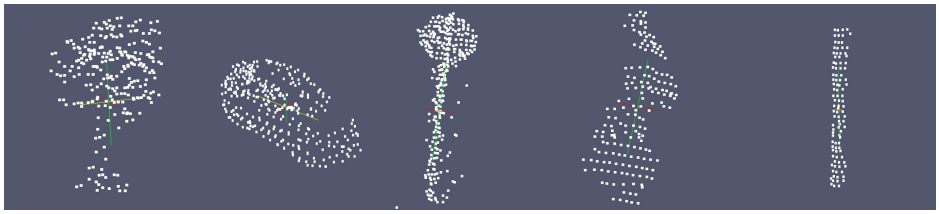

FIGURE 1 – BaseA : de gauche à droite, un arbre, une voiture, un panneau de signalisation, un piéton, un poteau.

4.1. Jeu de données utilisé

En raison du manque de base de données de référence dans le domaine de la classification de nuages de points 3D, nous avons décidé de rassembler des données à partir de plusieurs bases de données annotées contenant des nuages de points d'objets urbains. Nous répartissons les objets urbains en 8 classes différentes: arbre, voiture, feu et panneau de signalisation, poteau, piéton, bâtiment, bruit et artefact. Quelques exemples sont présentés dans la Fig. 1. Nous avons trouvé trois ensembles de données accessibles au public correspondaient à ces critère à notre connaissance : Lai, Fox (2010), Serna *et al.* (2014) et Quadros *et al.* (2012). Cela fait un total de 724 objets dont 80% utilisés pour l'apprentissage et 20% utilisés pour la validation. Dans le reste de l'article, nous ferons référence à cet ensemble de données sous le nom de BaseA.

De plus, nous avons réalisé une acquisition LiDAR terrestre à l'aide d'un sac à dos Leica Pegasus [1]. Cette acquisition a été réalisée en milieu urbain par une personne équipée du sac sur 200 mètres à pied. Une vue globale de la scène urbaine peut être trouvée dans la Fig. 2. Nous avons extrait manuellement 174

1. Nous aimerions remercier LeicaGeosystems d'avoir réalisé l'acquisition LiDAR.

objets urbains de cette acquisition et les avons annotés. La répartition de cette ensemble de données, appelé BaseB dans la suite de l'article, est la suivante: 75 arbres, 39 voitures, 8 feux et panneaux de signalisation, 23 poteaux et 29 piétons.

La taille des nuages de points est comprise entre 200 et 3 000 points pour la BaseA et entre 1000 et 300 000 pour la BaseB. Nous avons ensuite ré-échantillonnés chaque objets à 512 points avec un filtre "voxel-grid" afin qu'ils puissent être utilisés comme entrée pour le réseau PointNet. Dans la Fig. 2, nous présentons une illustration de cette étape pour un arbre de la BaseB. Les deux bases A et B sont toutes deux accessibles : http://www.lirmm.fr/~chaumont/DemoAndSources.html

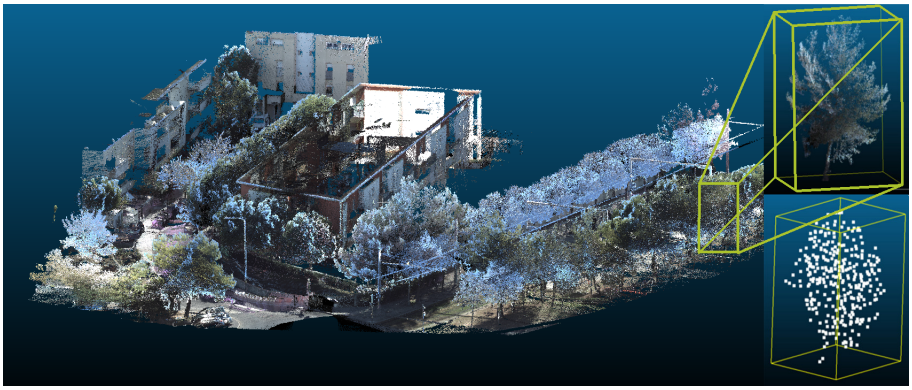

FIGURE 2 – Vue globale de notre acquisition LiDAR terrestre. On peut y distinguer des arbres, un arrêt de tramway ainsi qu'une petite résidence. A droite, un arbre est isolé puis sous échantillonné en un nuage de 512 points

4.2. Expérience témoin

Dans cette section, nous rappelons l'expérience de Zegaoui *et al.* (2018). Nous utilisons l'implémentation de PointNet [2] sous Tenserflow pour entraîner le réseau avec la BaseA et le tester sur la BaseB. Cette implémentation utilise l'augmentation des données avec des rotations aléatoires sur l'axe vertical, un bruit gaussien sur les coordonnées, l'optimisation Adam et la normalisation des lots (batch normalization). Les résultats sont reportés dans le tableau 1.

Nous avons constaté que la F mesure global était de 0.744. Les scores F1 pour la classe arbre : 0.896, voiture : 0.904 et piéton : 0.828, sont satisfaisants. Cependant, pour les feux et panneaux de signalisation ainsi que les poteaux, les scores sont beaucoup plus faibles : 0.267 pour les panneaux de signalisation et 0.074 pour les poteaux.

2. https://github.com/charlesq34/pointnet

Nous remarquons que le réseau confond ces deux dernières classes. Le réseau prédit que 12 des 23 poteaux sont des panneaux de signalisation et 3 des 4 panneaux de signalisation, des poteaux. Dans l'ensemble, les résultats sont vraiment encourageants compte tenu du peu de données utilisées.

		Vérité terrain (annotations)				
		arbre (75)	***voiture (39)***	***signalisation (8)***	***poteau (23)***	***piéton (15)***
Classification	***"arbre"***	69	2	0	8	0
	"voiture"	1	33	0	0	0
	"signalisation"	4	0	4	12	2
	"poteau"	0	0	3	1	0
	"piéton"	1	0	1	0	12
	"bâtiment"	0	0	0	2	0
	"bruit"	0	4	0	0	1
	F mesure	0.896	0.904	0.267	0.074	0.828

TABLE 1 – Matrice de confusion pour l'expérience témoin. F mesure globale : 0.744

4.3. Expériences supplémentaires

Dans cette section, toutes les expériences reposent sur la même méthodologie que l'expérience témoin, sauf si précisé autrement.

4.3.1. Rajout de base de données

Dans cette expérience, nous enrichissons la BaseA avec certains des objets disponibles dans le très récent jeu de données Roynard *et al.* (2018).

À partir de ce jeu de données, nous extrayons plus de 900 objets qui correspondent à l'une de nos 8 classes et nous les ajoutons à notre BaseA. Nous avons ensuite effectué une autre expérience de classification avec la BaseA nouvellement augmenté en tant qu'ensemble d'entraînement et avec la BaseB comme ensemble de test. La taille de notre base d'entraînement augmente ainsi de 230% (1668 objets au total), avec la répartition suivante 349 arbres, 382 voitures, 279 feux et panneaux de signalisation, 292 poteaux, 164 piétons, 61 exemples de bruits et 141 bâtiments.

Les résultats, dans le tableau 2 montrent une augmentation du score F1 global : de 0.744 à 0.857 (+11.3%). L'augmentation de la F mesure affecte toutes les classes à l'exception de la classe piéton qui passe de 0.828 à 0.815 (-1.3 %). L'augmentation est la plus prononcée pour la classe poteau avec un gain de 0.074 à 0.629 (+55.5%). La classe voiture atteint également un score parfait de 1.000 (+9.6 %).

Nous observons une amélioration notable des performances provenant de l'enrichissement de notre ensemble d'entraînement. Cela n'est pas surprenant étant donné qu'il est généralement admis dans le domaine de l'apprentissage en profondeur que le moyen le plus efficace d'améliorer la précision d'un réseau est de l'alimenter avec davantage de données.

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	70	0	0	2	1
	"voiture"	2	39	0	0	1
	"signalisation"	2	0	7	10	2
	"poteau"	0	0	1	11	0
	"piéton"	0	0	1	0	11
	"bâtiment"	1	0	0	0	0
	"bruit"	0	0	0	0	0
	F mesure	0.946	1.000	0.467	0.629	0.815

TABLE 2 – Matrice de confusion pour l'expérience avec ensemble d'entraînement enrichi. F mesure global : 0.857

4.3.2. Fusion de classes

Une des raisons des faibles performances à la fois pour les classes poteau et feu/panneau de signalisation est la confusion du réseau entre celles-ci. Ce qui est compréhensible étant donné qu'elles décrivent des objets géométriquement proches.

Nous les avons donc fusionnés en une seule classe TSLP (Traffic Sign/Light + Pole) et exécuter à nouveau l'expérience de classification avec les mêmes ensembles de données que dans notre expérience de base, mais avec seulement 6 classes au lieu de 7.

Les résultats dans le tableau 3 montrent un score F1 significativement plus élevé pour la classe TSLP, 0.849, que dans le tableau 1 pour la classe signalisation 0.267, et la classe poteau 0.074. Si nous combinons les résultats de l'expérience témoin pour ces deux classes, nous trouvons que 20 fois sur 31, le réseau a "correctement" prédit soit un poteau, soit un feu ou panneau de signalisation. Cela nous donne un score F1 de 0.702 qui reste inférieur à celui de la classe TSLP, 0.849(+14.7%). Il y a également une augmentation du score F1 global de 0.744 à 0.831 (+8.7%) ainsi qu'une légère augmentation pour la classe des arbres, de 0.896 à 0.920 (+ 2.4%). Cependant, nous pouvons également voir une diminution pour les classes voiture et piéton, respectivement de 0.904 à 0.806 (-9.8%) et de 0.828 à 0.750 (-7.8%).

Le réseau apprend correctement à différencier les feux et panneaux de signalisation ainsi que les poteaux du reste, mais il n'arrive pas à faire la différence entre les deux. Néanmoins, la fusion de ces classes ne peut pas être une stratégie à long terme car, comme l'indiquent nos résultats, le consensus général en matière d'apprentissage en profondeur est que plus il y a de classes utilisées lors de l'entraînement, meilleurs sont les résultats globaux. Cela vient du fait que moins de classes signifie moins de contre-exemples pour chacune des autres classes. C'est pourquoi la baisse de précision des classes piéton et voiture n'est pas surprenante.

		Vérité terrain (annotations)			
		arbre (75)	*voiture (39)*	*TSLP(31)*	*piéton (15)*
Classification	*"arbre"*	69	3	3	0
	"voiture"	1	27	0	0
	"TSLP"	2	0	28	5
	"piéton"	0	0	0	9
	"bâtiment"	1	0	0	0
	"bruit"	2	9	0	1
	F mesure	0.920	0.806	0.849	0.750

TABLE 3 – Matrice de confusion avec fusion des classes signalisation et poteaux.F mesure globale : 0.831

4.3.3. Augmentation virtuelle de la base

Dans cette expérience, nous voulons tester une nouvelle manière de faire de l'augmentation de données, qui consiste à simuler des occultations. Pour chaque nuage de notre BaseA, nous en générons plusieurs versions "occultée". Un moyen simple d'y parvenir est de générer des plans aléatoires pour découper les nuages de points. Les plans sont définis par l'axe des z, voir Fig. 3 pour la visualisation des coordonnées, un angle aléatoire, et passent par l'isobarycentre du nuage. Le nuage est alors divisé en deux et nous ne conservons que la plus grande partie. Cette opération est répétée k fois pour chaque nuage.

Les résultats sont reportés dans les tableaux 4, 5 et 6 pour chaque k. Avec k = 5, le score F1 global est de 0.669 (-7.5%), pour k = 10, il est de 0.444 (-30%) et pour k = 50, il est de 0.578 (-16.6%). Le score de la classe poteau augmente de 0.074 dans l'expérience de base à 0,080 avec k = 5 (+0.6%) et 0.083 (+0.9%) avec k = 10 et k = 50. Le score F1 de la classe signalisation augmente de 0.267 à 0.370 (+10.3%) avec k = 10 et à 0.387 (+12%) avec k = 50. Toutes les autres classes voient leurs scores chuter parfois de manière drastique, par exemple le score de la classe de voiture passe de 0.904 à 0.453 (-45.1%) pour k = 10.

L'enrichissement de notre ensemble d'entraînement cause plus de diminution que d'augmentation au niveau des scores. Cela a tendance à être plus remarquable pour les valeurs plus élevées de k. Une explication de ce phénomène serait que plus on passe d'objets «coupés» au réseau, plus il apprend à reconnaître des plans géométrique et devient ainsi moins efficace lorsqu'il est testé sur un objet «entier».

4.3.4. Rotations

Notre ensemble de test provient d'une acquisition LiDAR différente de notre ensemble d'entraînement, ce qui signifie que les nuages de points peuvent être orientés différemment, ce qui pourrait entraîner une baisse des performances si le réseau n'est pas invariant à la rotation. Notre expérience consiste à appliquer des rotations aléatoires à notre ensemble de test selon un axe à la fois. Les angles de rotations sont choisis au hasard entre 0 et 2π.

Pour la rotation selon l'axe X, tableau 7, le score F1 global diminue de 0.744 à 0,669 (-7.5%). De plus, le réseau n'arrive plus à reconnaître les poteaux et les

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	58	2	0	4	0
	"voiture"	3	34	0	0	0
	"signalisation"	1	0	4	18	2
	"poteau"	0	0	1	1	0
	"piéton"	0	0	3	0	10
	"bâtiment"	6	3	0	0	3
	"bruit"	7	0	0	0	0
	F mesure	0.835	0.895	0.242	0.080	0.714

TABLE 4 – Matrice de confusion pour l'augmentation virtuelle avec k = 5. F mesure globale : 0.669

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	45	1	1	2	3
	"voiture"	0	12	0	1	1
	"signalisation"	0	0	5	14	0
	"poteau"	0	0	0	1	0
	"piéton"	0	0	2	0	8
	"bâtiment"	1	0	0	1	1
	"bruit"	29	26	0	4	2
	F mesure	0.709	0.453	0.370	0.083	0.640

TABLE 5 – Matrice de confusion pour l'augmentation virtuelle avec k = 10. F mesure globale : 0.444

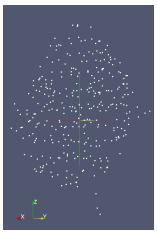

FIGURE 3 – Repère géométrique utilisé pour les nuages de points

piétons, les deux classes ayant un score F1 nul. On peut également noter une légère augmentation pour les classes voiture et signalisation, respectivement de 0.904 à 0.947 (+4.3%) et de 0.267 à 0.314 (+4.7%). La rotation selon l'axe Y, tableau 8 entraîne une légère diminution du score F1 global, de 0.744 à 0.731 (-1.3%), identique pour les classes voiture et arbre, respectivement de 0.896 à 0.890 (-0.6%) et 0.904 à 0.873 (-3.1%). La diminution est plus nette pour la classe piéton qui passe de 0.828 à 0.667 (-16.1%). Il y a également une augmentation pour les classes poteaux et signalisation, respectivement de 0.074 à 0.149 (+7.5%) et de 0.267 à 0.343 (+7.6%). Pour la rotation selon Z, tableau 9, on observe que le score F1 global passe de 0.744 à 0.706 (-3.8%). Les classes impactées par la diminution sont les classes voiture, arbre et piéton qui descendent respectivement à: 0.842 (de 0.904 : -6.2%), 0.730 (de 0.896 : -16.6%) et 0.800 (de 0.828 : -2.8%). Il y a également une augmentation pour les classes

		Vérité terrain (annotations)				
		abre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	66	7	0	1	2
	"voiture"	0	12	0	0	1
	"signalisation"	1	0	6	16	0
	"poteau"	0	0	0	1	0
	"piéton"	0	0	2	0	10
	"bâtiment"	1	0	0	0	2
	"bruit"	7	20	0	5	0
	F mesure	0.874	0.462	0.387	0.083	0.740

TABLE 6 – Matrice de confusion pour l'augmentation virtuelle avec k = 50. F mesure globale : 0.578

poteau et signalisation, respectivement de 0.074 à 0.286 (+21.2%) et de 0.267 à 0.412 (+14.5%).

Nous pouvons déduire des résultats que le réseau n'est pas invariant par rapport à l'orientation des objets. Nous supposons que l'architecture PointNet prend en compte l'orientation des formes lors de leur apprentissage. Néanmoins, nous ne savons pas à quel point PointNet est robuste aux petites rotations.

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	63	1	0	2	0
	"voiture"	1	36	0	0	0
	"signalisation"	2	0	8	19	13
	"poteau"	0	0	0	0	0
	"piéton"	1	0	0	0	0
	"bâtiment"	4	0	0	2	0
	"bruit"	4	2	0	0	2
	F mesure	0.894	0.947	0.314	0.000	0.000

TABLE 7 – Matrice de confusion pour une rotation selon X. F mesure globale : 0.669

		Vérité terrain (annotations				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	69	5	0	6	0
	"voiture"	1	31	0	0	0
	"signalisation"	2	0	6	14	5
	"poteau"	0	0	1	2	0
	"piéton"	2	0	1	0	9
	"bâtiment"	1	0	0	1	0
	"bruit"	0	3	0	0	1
	F mesure	0.890	0.873	0.343	0.149	0.667

TABLE 8 – Matrice de confusion pour une rotation selon Y. F mesure globale : 0.731

4.3.5. Taille des nuages de points

Dans l'expérience témoin, nous avons échantillonné toutes nos données à 512 points, en les réduisant parfois de 300 000 à 512 points (ce qui était le cas pour certains arbres) et parfois de 200 à 512 points (cela a été fait pour certains piétons). Afin d'évaluer l'influence du nombre de points, nous avons fait le même échantillonnage mais en remplaçant la taille final par 2048 points au lieu de 512.

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	69	13	0	7	0
	"voiture"	1	23	0	0	0
	"signalisation"	4	0	7	11	4
	"poteau"	0	0	1	4	0
	"piéton"	0	0	0	0	10
	"bâtiment"	1	1	0	1	0
	"bruit"	0	2	0	0	1
	F mesure	0.842	0.730	0.412	0.286	0.800

TABLE 9 – Matrice de confusion pour une rotation selon Z. F mesure globale : 0.706

Nous avons donc fixé la taille des nuages à 2048 points (le maximum recommandé par les auteurs de PointNet) pour nos deux jeux de données et réalisé une fois de plus l'expérience de classification.

Les résultats, dans le tableau 10, montre que le score F1 global augmentant légèrement de 0.744 à 0.763 (+1.9%). Certains scores montent comme pour les classes arbre : 0.935 (à partir de 0.896: +3.9%), signalisation : 0.385 (à partir de 0.267: +11.8%) et poteau : 0.080 (à partir de 0.074 : +0.6%). D'autres descendent comme pour les classes voiture : 0.886 (à partir de 0.904 : -1.8%) et piéton : 0.813 (à partir de 0.828: -1.8%). Cependant, ces variations restent faibles.

L'interprétation globale de ces résultats est que le nombre de points a peu d'impact sur les prédictions du réseau. C'est la même conclusion que les auteurs de Charles *et al.* (2017) ont tiré dans leurs expériences. Une suite intéressante à cette expérience serait de déterminer à quel point il est possible de sous échantillonner les nuages de points avant de constater une diminution significative des scores F1.

		Vérité terrain (annotations)				
		arbre (75)	*voiture (39)*	*signalisation (8)*	*poteau (23)*	*piéton (15)*
Classification	*"arbre"*	72	3	0	4	0
	"voiture"	0	31	0	0	0
	"signalisation"	1	0	5	10	2
	"poteau"	0	0	1	1	0
	"piéton"	2	0	2	0	13
	"bâtiment"	0	4	0	2	0
	"bruit"	0	1	0	6	0
	F mesure	0.935	0.886	0.385	0.080	0.813

TABLE 10 – Matrice de confusion avec une taille de 2048 points. F mesure globale : 0.763

4.3.6. Discussion

Nous pouvons conclure de nos expériences que la classification d'objets urbains à partir de nuages de points LiDAR, donnent des résultats satisfaisants qui ne peuvent que s'améliorer à mesure que nous obtenons plus de données pour entraîner les réseaux. Même si d'autres pistes peuvent être explorées afin

améliorer davantage les performances, telle une architecture plus complexe ou une augmentation virtuelle fonctionnelle, le meilleur moyen reste encore d'ajouter encore plus de données à l'ensemble d'entraînement.

Une autre remarque que l'on peut faire est que dans certains exemples de nuages de points où le réseau fait une erreur, différencier les versions sous-échantillonnées des nuages de points semble très difficile. Nous supposons donc que certaines des erreurs pourraient provenir de la dégradation des nuages de points d'origine par l'algorithme "voxel-grid" plutôt que de la capacité du réseau à reconnaître les formes 3D.

4.4. Conclusion

Dans cet article, nous avons présenté une série d'expériences sur la classification des objets urbains en 3D à l'aide de PointNet. Ces expériences ont montré que le meilleur moyen d'améliorer le score F1 est d'avoir un ensemble d'entraînement plus large. Nous sommes également arrivés à la conclusion que le réseau n'est pas invariant par rapport à l'orientation des objets et que le réseau confond les panneaux de signalisation et les poteaux entre eux, tout en arrivant à les différencier des autres classes.

Nous avons également souligné que l'un des facteurs limitant ces expériences pourrait être la dégradation des nuages de points par le filtre "voxel-grid". Un moyen d'améliorer le réseau de classification serait alors de lui permettre de prendre en entrée les nuages de points LiDAR d'origine. Cette amélioration est également liée à notre premier objectif qui est la détection des objets dans des grandes scènes 3D.

À l'avenir, nous prévoyons d'expérimenter avec des nuages de points synthétisés représentant des objets urbains simples tels que des poteaux ou des arbres et voir si cela a le même effet que l'ajout de données numérisées réelles. Nous souhaitons également aborder la question de la détection d'objets urbains dans les grands nuages de points 3D.

Bibliographie

(s. d.).

Anguelov D., Dulong C., Filip D., Früh C., Lafon S., Lyon R. *et al.* (2010). Google street view: Capturing the world at street level. *Computer*, vol. 43, p. 32-38.

Charles R. Q., Su H., Kaichun M., Guibas L. J. (2017, July). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 ieee conference on computer vision and pattern recognition (cvpr)*, p. 77-85.

Engelmann F., Kontogianni T., Hermans A., Leibe B. (2017). Exploring spatial context for 3d semantic segmentation of point clouds. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, p. 716-724.

Johns E., Leutenegger S., Davison A. J. (2016). Pairwise decomposition of image sequences for active multi-view recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 3813-3822.

Kanezaki A., Matsushita Y., Nishida Y. (2018). Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of ieee international conference on computer vision and pattern recognition (cvpr).*

Klokov R., Lempitsky V. S. (2017). Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 863-872.

Lai K., Fox D. (2010, July). Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *International Journal of Robotics Research*, vol. 29, n° 8, p. 1019–1037.

LeCun Y., Bengio Y., Hinton G. (2015, mai). Deep learning. *Nature*, vol. 521, n° 7553, p. 436–444.

Maturana D., Scherer S. (2015, Sept). VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 ieee/rsj international conference on intelligent robots and systems (iros)*, p. 922-928.

Qi C. R., Yi L., Su H., Guibas L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Nips.*

Quadros A., Underwood J., Douillard B. (2012, May 14-18). An Occlusion-aware Feature for Range Images. In *Robotics and automation, 2012. icra'12. ieee international conference on.*

Ren S., He K., Girshick R. B., Sun J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 1137-1149.

Riegler G., Ulusoy A. O., Geiger A. (2017). Octnet: Learning deep 3d representations at high resolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 6620-6629.

Roynard X., Deschaud J.-E., Goulette F. (2018). Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, vol. 37, n° 6, p. 545-557.

Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S. *et al.* (2014). Imagenet large scale visual recognition challenge. *CoRR*, vol. abs/1409.0575.

Serna A., Marcotegui B., Goulette F., Deschaud J.-E. (2014). Paris-rue-Madame Database - A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods. In *Icpram.*

Zegaoui Y., Chaumont M., Subsol G., Borianne P., Derras M. (2018). First experiments of deep-learning on lidar point clouds for classification of urban object. *CFPT.*