

# Des systèmes IA pour la classification des nuages de points LiDAR HD

■ **Matthieu PORTE**

*L'apprentissage profond a permis des avancées significatives en vision par ordinateur. En particulier, les algorithmes obtenus par apprentissage donnent des performances remarquables pour la classification des nuages de points 3D obtenus par LiDAR aérien. On présente ici les principes à la base de cette approche d'apprentissage supervisé, l'apport de l'apprentissage profond pour traiter les données de grande dimension, celui des biais inductifs géométriques pour tirer parti des structures des nuages de points. Ces modèles sont ensuite intégrés dans des systèmes IA plus larges : on présente les principes et performances de systèmes utilisés à l'IGN pour la classification des bâtiments.*

■ **MOTS-CLÉS**

*Apprentissage machine, apprentissage profond, nuages de points 3D, segmentation sémantique, classification, intelligence artificielle*

La couverture fine en 3D permise par LiDAR Haute Densité (HD) est un outil clé pour décrire le territoire et ses transformations à l'heure de l'Anthropocène, à condition d'être sémantisée. Pour en dériver des modèles numériques de terrain, des cartes de couvert végétal, des maquettes 3D urbaines, il est nécessaire en effet de réaliser une classification du nuage de points – c'est-à-dire d'identifier parmi les points ceux correspondant, sur le terrain, au sol, à de la végétation, à des bâtiments, à des surfaces d'eau, etc. Cette opération de classification peut être réalisée de plusieurs manières. La première consiste à mobiliser des professionnels de l'analyse visuelle de ces nuages de points pour les interpréter et affecter aux différents points leur classe. Bien souvent, cette approche demeure utilisée, au moins en partie, car elle permet d'atteindre la meilleure qualité de classification.

Toutefois, pour faciliter le passage à l'échelle<sup>1</sup> – et la réalisation en des temps acceptables de ces classifications à une échelle nationale – différentes formes de traitements automatiques sont également possibles.

Un premier ensemble d'approches tente de construire directement des

<sup>1</sup> Un peu plus de 550 000 km<sup>2</sup> seront couverts par LiDAR HD.

algorithmes pertinents pour tout ou partie des thèmes de la classification, en assemblant diverses règles formalisant des heuristiques les concernant. Par exemple, pour identifier le sol au sein du nuage de points, on pourra s'appuyer sur la formalisation des deux intuitions "le sol se trouve en bas" et "les points à proximité horizontale du sol sont eux-mêmes du sol" pour développer un algorithme explicite. Celui-ci identifiera dans une zone donnée les points les plus bas, leur affectera la classe "sol", puis propagera cette information de façon itérative aux points voisins dans le même plan horizontal ou dans une pente faible par rapport à celui-ci.

La principale limite de cette approche réside dans sa difficulté à généraliser – c'est-à-dire de trouver de telles séries de règles adaptées à des paysages très variés. Dans notre exemple, comment trouver des règles qui parviennent à gérer à la fois des zones de faible et de forte pente, de rupture de pente, des dévers en montagne où deux niveaux de sol peuvent être à l'aplomb l'un de l'autre – le tout sans considérer à tort des bâtiments ou de la végétation comme du sol ? Une réponse peut consister à particulariser ces algorithmes selon le type de paysages et d'objets considérés. Cela permet de

limiter l'ampleur de la généralisation nécessaire, au prix d'une augmentation rapide des besoins de développement d'algorithmes chaque fois particuliers. C'est pour lever cette limite qu'une autre famille d'approches est mobilisée, qui consiste à construire des algorithmes de classification des nuages de points par apprentissage machine (ou *machine learning*) et, plus particulièrement, par apprentissage profond (ou *deep learning*).

## Principes de l'apprentissage machine

L'apprentissage machine supervisé peut être compris, schématiquement, par analogie avec un étudiant apprenant à résoudre un nouveau type d'exercice. Pour ce faire, l'étudiant dispose d'une banque d'exercices corrigés. En pratiquant sur cette banque et en comparant ses réponses aux corrigés, l'étudiant peut ajuster sa pratique jusqu'à parvenir à des résultats satisfaisants. Il sera alors prêt à passer un examen, pourvu :

- que celui-ci soit "semblable" aux exercices d'entraînement ;
- que l'étudiant ne se soit pas contenté d'apprendre par cœur les corrigés – auquel cas tout énoncé même légèrement différent conduira à un échec certain.

En apprentissage machine, faire fonctionner cette analogie mobilise (*figure 1*) :

- un jeu de données d'entraînement (parfois appelé données d'apprentissage, annotations, labels ou vérités terrain) ; il s'agit d'un ensemble de paires réalisant la tâche visée pour l'algorithme recherché, c'est-à-dire dans notre cas d'un ensemble de nuages de points dans leur version "brute" (l'entrée recherchée de notre algorithme) et dans leur version classée (la sortie recherchée) ;

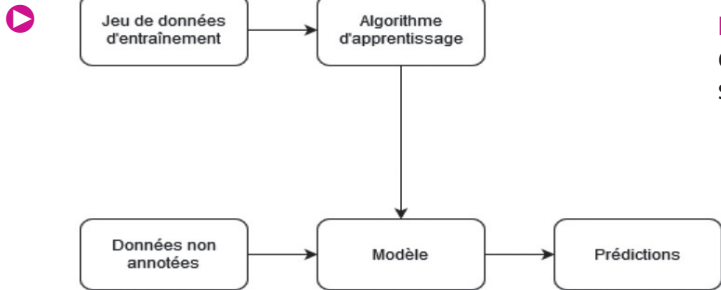


Figure 1. Principe de l'apprentissage supervisé.

## Le fléau de la dimension

Une première approche, naïve, pour s'en assurer consiste à exiger de couvrir avec une grande exhaustivité l'espace des données possibles. Par exemple, si notre variable d'entrée  $X$  est caractérisée par  $n$  variables continues suit une loi uniforme  $X \sim \mathcal{U}([0,1]^d)$ , on peut chercher à disposer d'un volume  $N$  de données  $\{x_1, \dots, x_N\}$  tel que, pour un niveau de proximité  $\epsilon$  demandé, on ait statistiquement, dans une majorité des cas, pour une nouvelle donnée  $x$  de loi  $X$ :  $\min_{1 \leq i \leq N} d(x, x_i) \leq \epsilon$ .

Dans ce cas, si  $d(x, x') = \sum_k (x_k - x'_k)^2$  est la distance euclidienne sur  $\mathbb{R}^n$ , le volume  $V(B_n, \epsilon)$  de la boule de dimension  $n$  et de rayon  $\epsilon$  vérifie :  $V(B_n, \epsilon) \sim C\epsilon^n$ .

Par conséquent, il est nécessaire de se doter d'un volume de données vérifiant :  $N \sim C'\epsilon^{-n}$ .

Si  $\epsilon = 0.1$ , par exemple, alors  $N \sim C'10^n$ . Autrement dit, le volume de données nécessaires croît exponentiellement avec la dimension de l'espace des données d'entrée.

Or, dans le cas de la classification des nuages de points, on a typiquement :  $n \approx 10 \left(\frac{pts}{m^2}\right) \times (50 m)^2 \times 10$  (variables par points)  $\approx 250\ 000$

Les volumes de données nécessaires pour couvrir suffisamment cet espace (de l'ordre de  $10^{250\ 000}$ ) sont tout à fait prohibitifs – par comparaison, le nombre d'atomes dans l'univers observable est estimé à  $10^{80}$ . Cette situation récurrente est connue sous le nom de fléau de la dimension : les problèmes d'apprentissage machine sur des espaces de grande dimension (c'est-à-dire avec un grand nombre de paramètres en entrée) sont ainsi *a priori* très difficiles – il va être nécessaire d'extrapoler l'information des données d'apprentissage vers des données qui en sont très éloignées.

## L'apprentissage profond pour lever le fléau de la dimension

C'est précisément dans ce cadre, celui de l'apprentissage en grande dimension, que des avancées significatives ont été obtenues durant la dernière décennie par une famille de méthodes particulières : les réseaux de neurones profonds, donnant leur nom à l'apprentissage profond. Ces architectures, inspirées

- une architecture disposant de multiples paramètres indéterminés à un choix de valeurs pour l'ensemble des paramètres de l'architecture correspond un algorithme possible pour notre tâche ;

- un algorithme d'apprentissage qui est un algorithme d'optimisation cherchant les valeurs de paramètres de l'architecture permettant d'obtenir les meilleures performances sur le jeu de données d'entraînement.

Une fois ces paramètres déterminés et figés, on obtient notre algorithme de classification, aussi appelé modèle dans ce contexte, prêt à être appliqué sur de nouvelles données pour réaliser des prédictions.

Une formalisation de ces idées consiste à voir les nuages de points bruts comme une variable aléatoire  $X \in \mathcal{X}$ .  $\mathcal{X}$  est, par exemple, l'ensemble de tous les nuages de points LiDAR HD de 50 m de côté. Les nuages de points ayant dix attributs (trois coordonnées spatiales, intensité, numéro d'écho, nombre total d'échos, quatre bandes de couleur RGB-IR), chaque tel nuage est un élément d'un certain  $\mathbb{R}^{n \times 10}$  et  $\mathcal{X} \subset \cup_{n \geq 0} \mathbb{R}^{n \times 10}$ . De la même manière, les nuages classés sont vus comme une variable aléatoire  $Y \in \mathcal{Y} \subset \cup_{n \geq 0} \mathbb{R}^{n \times 7}$ , chaque point pouvant prendre sept valeurs de classe (sol, bâtiment, végétation, ponts, eau, autre sursol pérenne, autre) et l'information étant codée par 0 ou 1 en chaque point sur chaque dimension d'un tel "vecteur de classe".

Le jeu de données d'apprentissage est alors considéré comme un ensemble  $\{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$  de réalisations indépendantes et identiquement distribuées de la loi jointe des variables  $(X, Y)$ . L'architecture est alors un ensemble de fonctions  $\{f_\theta : \mathcal{X} \rightarrow \mathcal{Y}\}_{\theta \in \Theta}$  paramétrées par  $\theta \in \Theta$  dont on cherche à estimer la meilleure valeur. Pour cela, on se dote d'une

fonction de coût (ou pénalité)  $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  mesurant l'écart entre des prédictions et la vérité terrain. Le problème devient alors celui de l'estimation de  $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \mathbb{E}_{X, Y} [\mathcal{L}(f_\theta(X), Y)]$ . En lui-même, ce problème ne peut être résolu sans connaître la loi jointe  $\mathbb{P}(X, Y)$  – ou au moins la loi conditionnelle  $\mathbb{P}(Y|X)$ . On utilise alors le jeu de données d'entraînement, qui la réalise, pour réaliser l'approximation par le coût moyen empirique

$$\theta^* \approx \underset{\theta \in \Theta}{\operatorname{argmin}} \mathbb{E}_{\{(x_1, y_1), \dots, (x_N, y_N)\}} [\mathcal{L}(f_\theta(X), Y)]$$

$$= \underset{\theta \in \Theta}{\operatorname{argmin}} \left( \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_\theta(x_i), y_i) \right)$$

L'algorithme d'apprentissage consistera alors précisément en la procédure d'optimisation permettant d'estimer  $\theta^*$ , et notre modèle sera l'algorithme particulier  $f_{\theta^*}$ . Celui-ci ne sera pertinent que s'il parvient à généraliser. Les conditions posées dans l'analogie avec l'étudiant correspondent ici à :

- les données sur lesquelles le modèle sera appliqué doivent être "semblables" (c'est-à-dire issues de la même loi  $\mathbb{P}(X, Y)$  ;
- le modèle ne doit pas avoir "appris par cœur", ce qui se traduit dans ce cas par le fait que l'approximation de l'erreur moyenne théorique par l'erreur moyenne empirique sur les données d'entraînement doit être valable ; une des manières de s'en assurer consiste à réserver une part des données pour tester les modèles, sans les rendre jamais accessibles durant l'apprentissage.

L'approximation empirique par le jeu de données n'est, de plus, valable qu'à condition que l'on dispose de suffisamment d'exemples. Le volume des jeux de données d'apprentissage est ainsi une condition critique du succès de ces méthodes. Avec des jeux trop restreints, il sera en effet nécessaire d'extrapoler trop fortement des exemples connus vers de nouvelles données.



de la transmission de signaux entre neurones dans le cerveau humain, ont en effet obtenu en pratique des résultats remarquables dans de nombreux domaines nécessitant de tels traitements en grande dimension, comme le traitement du langage naturel ou la vision par ordinateur – dont la classification des nuages de points 3D. Cette efficacité pratique est en bonne partie une surprise, dont les explications théoriques demeurent partielles.

L'architecture la plus simple de réseau de neurones profond est celle du perceptron multicouche (ou MLP pour *MultiLayer Perceptron*). Elle consiste en une composition de multiples "couches" élémentaires, dont le grand nombre constituera la "profondeur" du réseau et lui permettra d'extraire des informations de plus en plus complexes des signaux d'entrée. Un vecteur de dimension  $n$  en entrée est successivement transformé en une suite de représentations "cachées" de dimensions  $h_1, \dots, h_D$  par une alternance d'opérations linéaires et de non-linéarités "ponctuelles", c'est-à-dire appliquées indépendamment dans chaque dimension.

L'architecture

$$f_W = \sigma^{D-1} \circ W^{D-1} \circ \dots \circ \sigma^1 \circ W^1 \text{ où}$$

$$W^i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_{i+1}} \in \mathcal{M}_{d_i, d_{i+1}}(\mathbb{R})$$

est une transformation linéaire et où

$$\sigma^i : \begin{matrix} \mathbb{R}^{d_{i+1}} \rightarrow \mathbb{R}^{d_{i+1}} \\ (x_1, \dots, x_{d_{i+1}}) \mapsto (\sigma(x_1), \dots, \sigma(x_{d_{i+1}})) \end{matrix}$$

est une non-linéarité "ponctuelle", avec  $\sigma$  une même fonction non linéaire dite "d'activation", typiquement :

$$\sigma : \begin{matrix} \mathbb{R} \rightarrow \mathbb{R} \\ x \mapsto \max(0, x) \end{matrix}$$

Les paramètres de l'architecture sont ici les valeurs de chaque matrice  $W_i$ . La surprise mentionnée plus haut est alors que ces architectures, lorsqu'elles sont entraînées avec des volumes de données importants – mais pas prohibitifs – et des puissances de calcul suffisantes – en particulier en utilisant les capacités de parallélisation permises par les processeurs graphiques – parviennent à approximer des fonctions non linéaires complexes en grande dimension avec un haut degré de généralisation. Le fléau de la dimension est en partie, mais seulement en partie, conjuré ici.

Les besoins demeurent toutefois très élevés dans cette première approche. Si l'on travaille avec des couches inter-

médiaires toutes de même taille, par exemple, il faudra optimiser un nombre de paramètres  $\alpha \sim Dn^2$ , ce qui revient, dans notre exemple, à des centaines de milliards de paramètres. Pour réduire cette complexité, on mobilise alors la géométrie du problème en incorporant des *a priori* (ou biais inductifs) sur les symétries du problème. C'est l'approche commune à ce qui est connu comme deep learning géométrique [1]. En traitement d'images, c'est cette idée qui fonde par exemple les réseaux de neurones convolutifs.

Pour présenter un nuage de points 3D à un perceptron multicouche, il est en effet nécessaire de le réduire à un vecteur "plat", par exemple en fixant un ordre d'énumération des points du nuage puis en listant, dans l'ordre, chacun des descripteurs de chacun des points. Or, cette numérotation des points est arbitraire : on souhaite qu'intervertir deux points dans la représentation informatique du nuage se traduise, en sortie de notre modèle, par une interversion correspondante des classifications. C'est exactement cette intuition que formalise PointNet [2] l'une des architectures de base pour le traitement des nuages de points 3D (figure 2).

Les réseaux de neurones profonds utilisés pour classer les nuages de points LiDAR HD s'appuient sur une architecture semblable, RandLA-Net [4]. Celle-ci amène un *a priori* supplémentaire, de séparation des échelles, et permet d'extraire de façon économique des informations à de multiples échelles

spatiales au sein du nuage de points. Les réseaux ainsi obtenus sont alors, pour des performances bien supérieures, des temps d'entraînement réduits et des volumes de données d'apprentissage nécessaires plus limités, ramenés à des ordres de grandeur importants, mais praticables, de millions de paramètres à entraîner.

Le problème de l'apprentissage de classification de nuages de points étant ramené à des proportions accessibles, il est alors nécessaire de se doter de jeux d'entraînement pertinents. Celui-ci doit être constitué au regard d'une série de critères :

- le jeu de données doit être semblable aux données sur lesquelles le modèle sera appliqué ; cet aspect est difficile à estimer *a priori*, mais c'est celui qui rend, par exemple, plus difficile de mobiliser d'autres données LiDAR (LiDAR terrestre, LiDAR aérien dans d'autres conditions d'acquisition ou avec une autre nomenclature de classification) ;
- les données doivent être suffisamment diverses pour permettre aux modèles de généraliser – en particulier à travers la diversité des paysages et des objets du territoire national ;
- la qualité des données, en particulier des labels, est décisive ; les erreurs ou incohérences dans la labellisation se répercutent directement sur la qualité des modèles : l'apprentissage machine ne fournit que des "machines à systématiser" le contenu des bases d'apprentissage, pour cette raison, on privilégie tant que possible d'utiliser

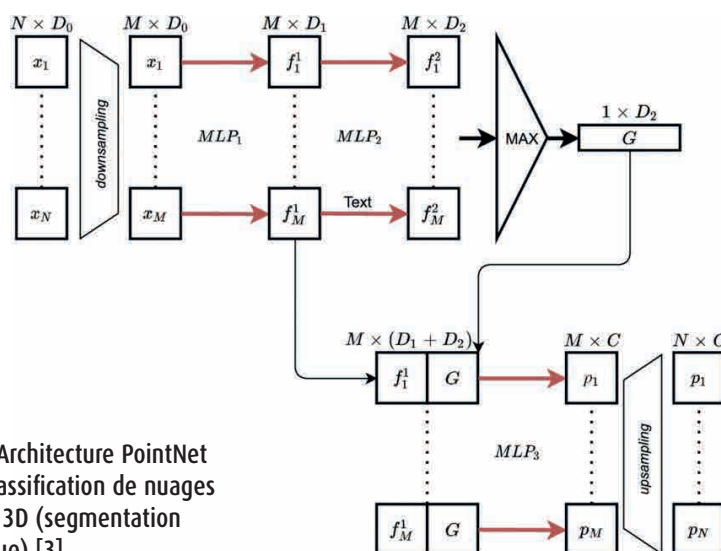
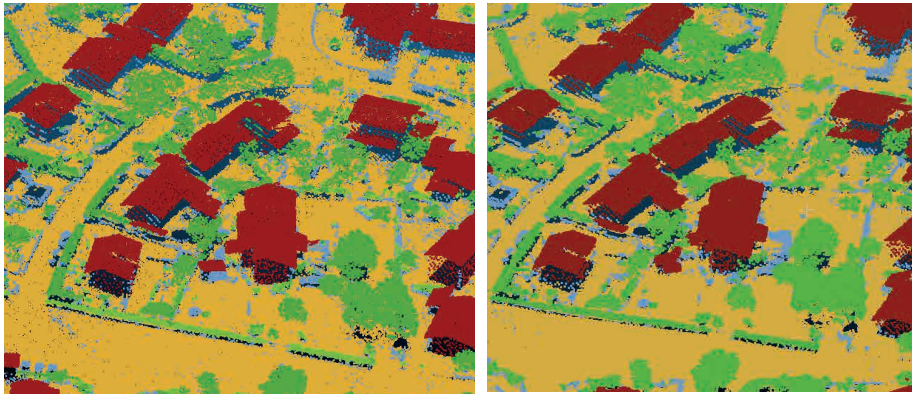


Figure 2. Architecture PointNet pour la classification de nuages de points 3D (segmentation sémantique) [3].



	Bâti
	Sol
	Végétation
	Autre

Figure 3. Labels (gauche) et prédictions des modèles (droite) sur une zone test.



des données ayant été obtenues (ou au moins reprises) par un travail expert de saisie sur les nuages de points ;  
- enfin, ces trois autres conditions étant remplies, la quantité de données permet d'améliorer les performances et la généralisation des modèles.

Les premiers modèles développés pour la classification du LiDAR HD France entière reposent ainsi sur un jeu de données de 150 dalles kilométriques (130 pour l'entraînement, 10 pour valider les modèles et 10 dans un jeu de "test" servant à évaluer le modèle) réparties sur quatre départements (04, 34, 39, 71). Sans être représentatifs de l'ensemble du territoire, ils ne sont cependant pas limités à un type de paysage particulier.

## Évaluation

Les performances des modèles sont évaluées (tableau 1) en calculant, pour chaque classe  $c$ , l'indice de Jaccard ( $I_0U_c$  pour "Intersection over Union") à partir des points vrais positifs (TP), faux positifs (FP) et faux négatifs (FN) pour cette classe :

$$I_0U_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

La valeur moyenne de cet indice à travers les classes donne une métrique

globale d'évaluation du modèle – celle-ci donne un poids important, rapporté à leur abondance, aux classes rares comme les ponts.

Les modèles entraînés présentent ainsi des performances hétérogènes :

- les performances d'ensemble sont intéressantes, en particulier sur les classes les plus abondantes (sol, bâti, végétation) dont dépendent de nombreux usages ; à l'échelle des données disponibles, les modèles sont parvenus à généraliser sur ces thèmes ;
- vue la relativement faible ampleur du jeu de données, les classes rares (particulièrement les ponts) sont assez mal identifiées ;
- les limites de ces métriques apparaissent lorsqu'une analyse visuelle est conduite (figure 3) : les calculs sont en effet réalisés à l'échelle du point et ne prennent pas particulièrement en compte :
  - les propriétés géométriques des erreurs : un point sol attribué à tort, 50 cm ou 3 cm au-dessus du sol sont pénalisés de la même manière ;
  - les statistiques "à l'objet" : dans certains cas (notamment pour les "autres sursols pérennes" comme les éoliennes ou les pylônes), la performance de la classification peut être

insuffisante pour délimiter précisément les objets, mais suffisante pour identifier la présence d'un objet ; dans d'autres cas, ce peuvent au contraire être plutôt des objets entiers qui sont manqués.

## Construire des systèmes IA

Pour cette raison, les modèles de classification obtenus par apprentissage profond ne fonctionnent pas seuls : ils sont intégrés à d'autres formes de traitement et à de la saisie humaine.

La classification des bâtiments fournit, à ce titre, un exemple de ces intégrations et de la construction de systèmes d'IA. Cette classification est en effet obtenue en croisant quatre sources d'informations complémentaires :

- les résultats d'algorithmes classiques : généralement, ceux-ci donnent des objets avec de bonnes propriétés géométriques, mais ont des difficultés à généraliser et font d'importantes sur/sous-détections à l'échelle d'objets entiers ;
  - les modèles obtenus par apprentissage profond : ceux-ci présentent plutôt les traits inverses aux précédents algorithmes ;
  - l'information déjà contenue en base, en l'occurrence dans la BDTopo : une information est en effet déjà disponible pour décrire les bâtiments sur territoire et peut permettre de confirmer la validité des résultats des algorithmes ;
  - la saisie par des opérateurs, fournissant la meilleure qualité, mais plus difficile à mobiliser à grande échelle.
- L'ensemble de ces sources d'information sont alors combinées pour permettre d'identifier, parmi la première classification, des bâtiments "confirmés" ou "réfutés" automatiquement ou des situations "incertaines" nécessitant l'apport d'informations par un opérateur (figure 4). Ces croisements sont optimisés pour

		Non-classé	Sol	Végétation	Bâti	Eau	Ponts	Sursol Pérenne
TRAIN	Proportion (% de pts)	7.67 %	46.27 %	43.64 %	2.11 %	0.22 %	0.03 %	0.07 %
	VAL	8.44 %	40.08 %	48.85 %	2.58 %	0.00 %	0.01 %	0.04 %
TEST	Proportion (% de pts)	9.91 %	50.57 %	38.89 %	0.58 %	0.03 %	0.01 %	0.01 %
	Indice Jaccard par classe	35.3 %	83.9 %	96.7 %	91.7 %	51.0 %	19.7 %	37.4 %
	Indice Jaccard moyen	59.4 %						

Tableau 1. Performances des modèles de classification de nuages de points.

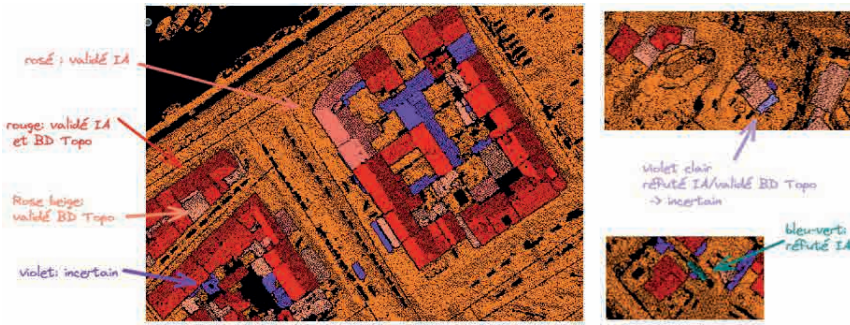


Figure 4. Fonctionnement de la classification hybride des bâtiments.

maximiser le taux d'automatisation tout en garantissant un taux d'erreur contrôlé (<2 % de faux positifs et faux négatifs sur un jeu d'évaluation). C'est ainsi de l'ordre de 98 % des objets qui font l'objet d'une décision automatique (tableau 2), permettant un passage à l'échelle significatif dans les traitements.

## Perspectives

Les méthodes d'apprentissage profond fournissent un moyen à ce jour inégalé de traitement automatique des nuages de points. Pour autant, ces algorithmes ne fonctionnent pas isolément : ils nécessitent, pour des productions de qualité contrôlée, de les articuler avec d'autres formes de traitement et d'autres sources d'information dans des systèmes IA. Ils nécessitent de plus des données d'entraînement qui doivent être à la fois de grande qualité et disponibles en suffisamment grande quantité. Ces données de grande qualité ne peuvent être obtenues que par un travail humain expert. Ces algorithmes sont en effet avant tout des "machines à systématiser" celui-ci à grande échelle : ils ne sont jamais meilleurs que la supervision qui leur est donnée.

Partant de ces constats, deux orientations devraient être conservées pour

2 <https://geoservices.ign.fr/lidarhd#telechargementclassifiees>

3 <https://github.com/IGNF/myria3d>

mobiliser ces techniques :

- celles-ci nécessitent un certain niveau de technicité et de savoir-faire particuliers, ainsi que des investissements pour construire des données d'apprentissage ; la mise en commun est décisive pour relever le défi de la cartographie de l'Anthropocène : ainsi, les données classées<sup>2</sup> à la base des jeux d'entraînement comme les librairies<sup>3</sup> développées pour l'apprentissage profond en 3D sont partagées par l'IGN en *open source* ;
- la manière de mobiliser ces techniques (pour automatiser quoi ? comment ?) reste largement sous-déterminée : puisqu'elle affecte le contenu du travail de nombreuses personnes, ces décisions doivent veiller à associer l'ensemble des personnes concernées au plus tôt dans la conception – pouvoir en débattre est en effet la condition nécessaire pour démocratiser l'intelligence artificielle.

## Contact

Matthieu PORTE,  
matthieu.porte@ign.fr  
Institut national de l'information géographique et forestière.

## Références

- [1] M. M. Bronstein, J. Bruna, T. Cohen, et P. Veličković, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and

Gauges". arXiv, 2 mai 2021. doi: 10.48550/arXiv.2104.13478.

[2] C. R. Qi, H. Su, K. Mo, et L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", ArXiv161200593 Cs, avr. 2017, Consulté le : 27 août 2021. [En ligne]. Disponible sur : <http://arxiv.org/abs/1612.00593>

[3] E. Kalinicheva, L. Landrieu, C. Mallet, et N. Chehata, "Predicting Vegetation Stratum Occupancy from Airborne LiDAR Data with Deep Learning", Int. J. Appl. Earth Obs. Geoinformation, vol. 112, p. 102863, août 2022, doi: 10.1016/j.jag.2022.102863.

[4] Q. Hu et al., "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds", présenté à Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, p. 11108-11117. Disponible sur: [https://openaccess.thecvf.com/content\\_CVPR\\_2020/html/Hu\\_RandLA-Net\\_Efficient\\_Semantic\\_Segmentation\\_of\\_Large-Scale\\_Point\\_Clouds\\_CVPR\\_2020\\_paper.html](https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_RandLA-Net_Efficient_Semantic_Segmentation_of_Large-Scale_Point_Clouds_CVPR_2020_paper.html)

## ABSTRACT

Computer vision has tremendously benefited from the advent of deep learning techniques. For instance, learnt deep networks constitute the state-of-art for various tasks concerning 3D point clouds, such as airborne LiDAR semantic segmentation. We expose the basic principles of supervised machine learning, how deep learning has provided a satisfying remedy to the curse of dimensionality and how to leverage the point cloud structure through geometric priors in our deep networks architecture. Models are then integrated in broader AI systems : we expose the principles and performance of an AI system developed by IGN (French mapping agency) for building classification.

Performances du module de classification des bâtiments				Confusion des décisions						
TEST	Précision	98.8 %	Justesse Réfutation	88.5 %			Indécis	Réfuté	Confirmé	N
	Rappel	96.8 %	Justesse Confirmation	97.9 %	Nature de l'emprise	Cas ambigu	6 %	42 %	52 %	159
	Automatisation	97.9 %				Autre	2 %	95 %	4 %	3048
	Performances modèle					Bâti	2 %	3 %	95 %	9568
	Indice de Jaccard bâti	91.7 %			N		267	3265	9243	12775

Tableau 2. Performances de la classification hybride des bâtiments.