

Rapport : Classification d'espèces d'arbres via nuages de points 3D et approches multi-vues 2D

Arehal Btissam
El Faiz Rekaia
Er-rregghay Hajar

Master Data Science

Academic Year 2024-2025

Table des matières

1	Introduction	2
2	Dataset et splits	2
2.1	Résumé global du dataset	3
2.2	Répartition Train	4
2.3	Répartition Test	4
3	Méthodologie	5
3.1	Multi-vues	5
3.2	Prétraitement	5
3.3	Méthodes indirectes (multi-vues 2D)	7
3.4	Méthodes quasi-directes (descripteurs 3D + ML)	7
3.5	Méthodes directes (deep learning sur nuages 3D)	7
4	Résultats	7
4.1	Stratégies d'évaluation	7
4.2	Comparaison des méthodes indirectes (2D Multi-vues)	8
4.3	Méthodes quasi-directes (3D Descripteurs + SVM)	9
4.4	Méthodes directes (PointNet et DGCNN)	9
4.5	PointNet	9
4.6	DGCNN	10
5	Discussion	11
6	Conclusion	11

Résumé

Ce rapport présente la démarche suivie pour la classification d'espèces d'arbres à partir de données 3D. Après la préparation du dataset et la séparation train/test, trois familles de méthodes sont comparées. Les approches indirectes exploitent des vues 2D avec CNN et descripteurs classiques. Les méthodes quasi-directes extraient descripteurs 3D (PCA, géométrie, densité) combinés à un SVM polynomial. Les méthodes directes (PointNet, DGCNN) apprennent directement sur les nuages de points. L'évaluation met en évidence les avantages et limites de chaque stratégie.

1 Introduction

La classification automatique des essences d'arbres est un enjeu important pour la foresterie, la gestion des ressources naturelles et l'étude de la biodiversité. Grâce aux progrès récents de la vision par ordinateur et de l'apprentissage automatique, il est aujourd'hui possible d'exploiter aussi bien des représentations 2D (images multi-vues générées à partir de modèles 3D) que des nuages de points 3D pour caractériser la forme et la structure des arbres.

L'objectif de ce travail est de comparer différentes approches de classification appliquées à un même jeu de données 3D d'arbres, en analysant leurs performances, leurs limites et leurs avantages respectifs. Trois grandes familles de méthodes sont étudiées :

1. **Indirectes** : utiliser des images 2D générées à partir des nuages de points et les traiter avec des réseaux convolutifs (CNN multi-vues, ou combinaisons de descripteurs classiques et CNN).
2. **Quasi-directes** : extraire des descripteurs géométriques et statistiques 3D (PCA, moments, densités, histogrammes) puis entraîner des modèles de machine learning classiques comme un SVM polynomial.
3. **Directes** : qui apprennent directement sur les nuages de points :
 - PointNet, premier réseau profond invariant à l'ordre des points ;
 - DGCNN (Dynamic Graph CNN), qui exploite les graphes de voisinage pour capturer la géométrie locale.

2 Dataset et splits

Le dataset contient 7 classes d'arbres. Chaque arbre est représenté par un nuage de points, projeté en plusieurs vues 2D.

2.1 Résumé global du dataset

Classe	Nombre d'arbres	Pourcentage (%)
Ash	39	5.74
Beech	164	23.7
Douglas Fir	183	26.40
Oak	22	3.30
Pine	25	3.73
Red Oak	100	14.35
Spruce	158	22.81
Total	691	100

TABLE 1 – Répartition globale des arbres par classe.

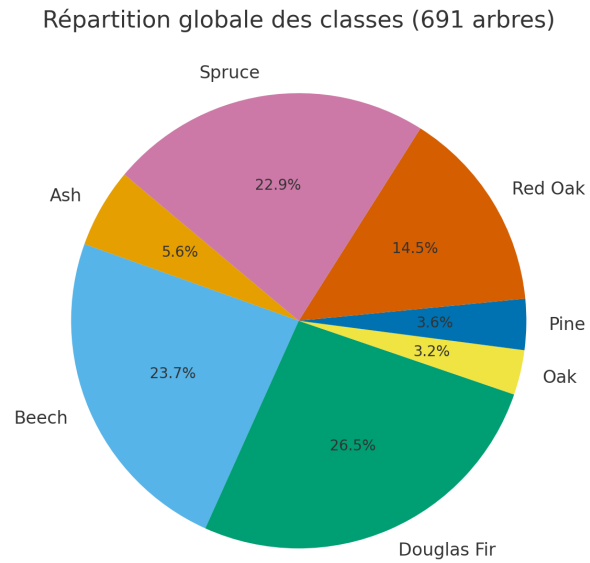


FIGURE 1 – Répartition global des classes.

2.2 Répartition Train

Classe	Nombre (train)	Proportion
Ash	33	5.99%
Beech	133	24.14%
Douglas Fir	138	25.05%
Oak	17	3.09%
Pine	21	3.81%
Red Oak	81	14.70%
Spruce	128	23.23%
Total	551	100%

TABLE 2 – Répartition des arbres dans l’ensemble d’entraînement.

2.3 Répartition Test

Classe	Nombre (test)	Proportion
Ash	7	5.34%
Beech	32	24.43%
Douglas Fir	34	25.95%
Oak	3	2.29%
Pine	5	3.82%
Red Oak	19	14.50%
Spruce	31	23.66%
Total	131	100%

TABLE 3 – Répartition des arbres dans l’ensemble de test.

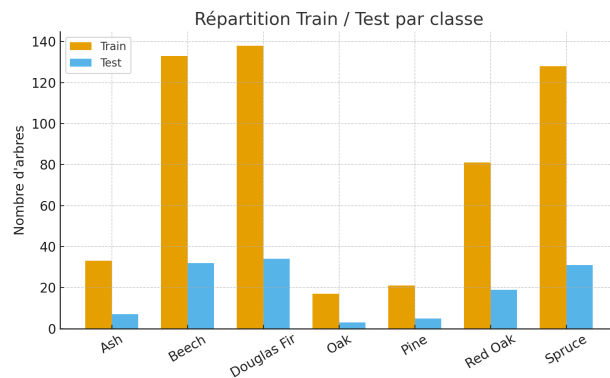


FIGURE 2 – Répartition Train/Test.

3 Méthodologie

Cette section décrit les étapes de prétraitement appliquées, ainsi que les différentes familles de méthodes de classification testées. L'implémentation a été réalisée avec `PyTorch`, `Scikit-learn` et `Open3D`.

3.1 Multi-vues

Afin de transformer les nuages de points en données exploitables par des réseaux CNN, chaque arbre a été projeté sous plusieurs angles de vue.

Après expérimentation, le nombre de vues N a été fixé à **8**, car :

- un nombre plus faible (4 vues) ne permet pas de capturer suffisamment la géométrie de l'arbre,
- un nombre plus élevé (12 ou 16 vues) augmente le temps de calcul sans amélioration notable des performances,
- 8 vues offrent donc un bon compromis entre **qualité de représentation** et **efficacité computationnelle**.

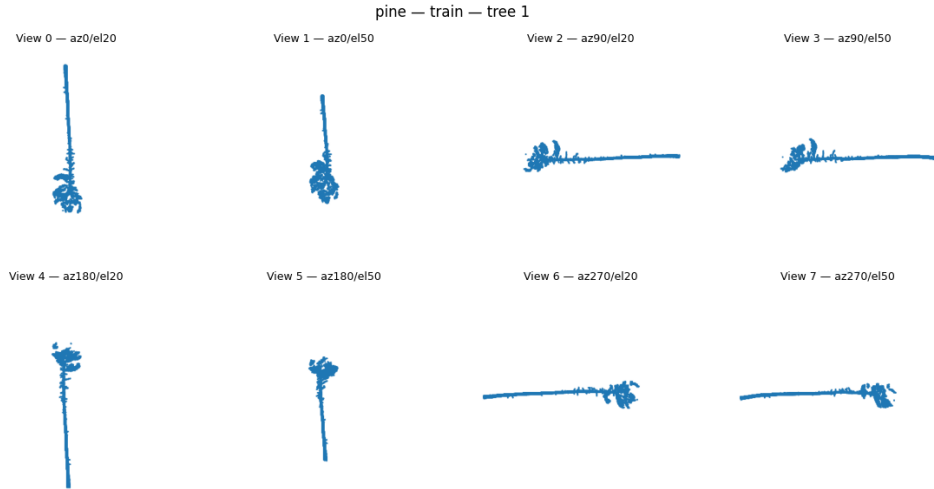


FIGURE 3 – Exemple de représentation multi-vues ($N = 8$) d'un arbre.

3.2 Prétraitement

Le prétraitement a été adapté selon la nature des méthodes de classification utilisées (multi-vues 2D, quasi-directes 3D ou directes 3D). L'objectif est d'assurer une qualité homogène des données et de préparer des représentations exploitables par les modèles.

Prétraitement multi-vues (2D).

- Nettoyage des nuages de points pour supprimer les points aberrants.
- Réduction de densité par downsampling voxel afin de diminuer le nombre de points tout en conservant la structure.
- Alignement des arbres par analyse en composantes principales (PCA), garantissant une orientation cohérente avec l'axe vertical.
- Normalisation par boîte englobante (AABB) pour homogénéiser la taille.

- Génération de $N = 8$ vues 2D par arbre, projetées sous différents angles.
- Redimensionnement des images (224×224 px) et normalisation des canaux selon les statistiques d'ImageNet.
- Augmentation de données : recadrage aléatoire, retournements horizontaux, légères perturbations de couleur.

Prétraitement quasi-direct (3D + descripteurs).

- Nettoyage des nuages (suppression d'outliers, réduction voxel).
- Alignement PCA et normalisation dans une sphère unité.
- Extraction de descripteurs géométriques et statistiques : valeurs propres PCA, ratios d'extension, densités locales multi-échelles, moments statistiques, histogrammes (radiaux, verticaux, cylindriques).
- Génération de vecteurs de caractéristiques normalisés pour apprentissage avec un SVM.

Prétraitement direct (3D deep learning).

- Opérations de nettoyage, réduction de densité, alignement PCA et normalisation AABB identiques aux quasi-directes.
- Échantillonnage par *Farthest Point Sampling (FPS)* pour obtenir un nombre fixe de 1024 points par arbre.
- Les nuages ainsi standardisés servent directement d'entrée aux réseaux PointNet et DGCNN.

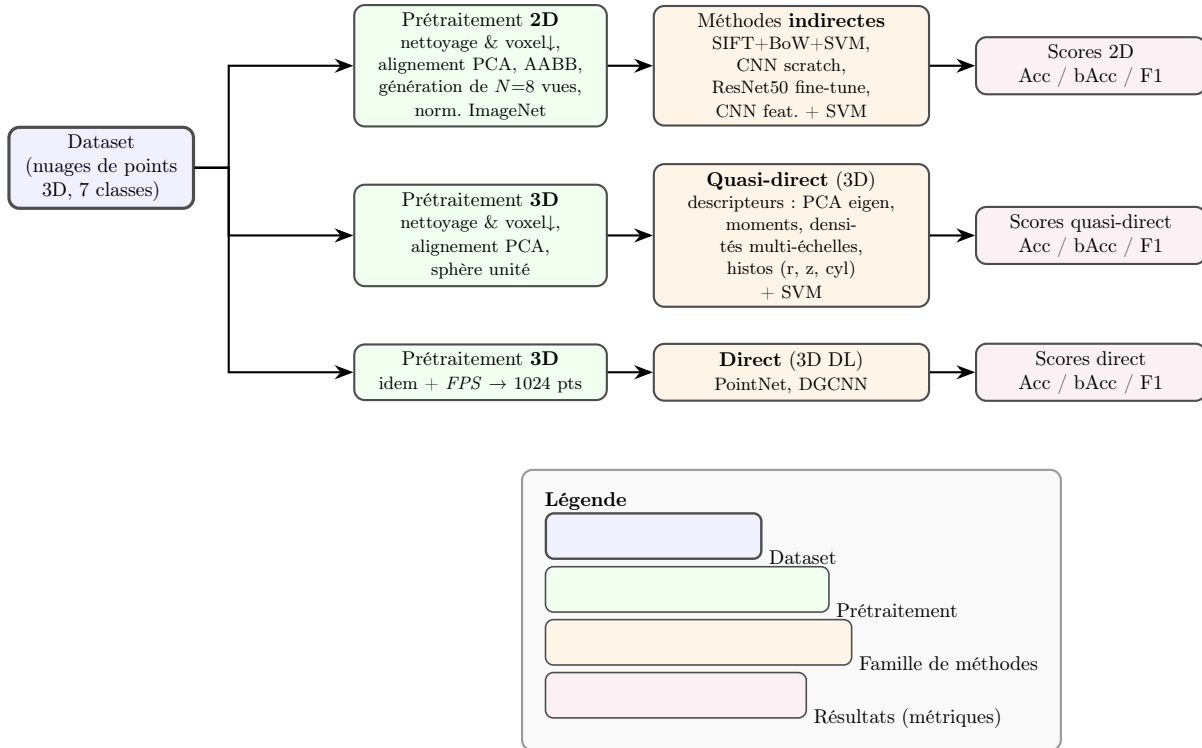


FIGURE 4 – Schéma récapitulatif du pipeline : génération 2D (indirecte), descripteurs 3D (quasi-directe) et deep learning 3D (directe).

3.3 Méthodes indirectes (multi-vues 2D)

Les nuages 3D sont convertis en images multi-vues, qui servent d'entrée à différents modèles :

1. **Descripteurs classiques + SVM polynomial** : extraction SIFT, quantification Bag-of-Words, classification par SVM polynomial.
2. **CNN from Scratch** : apprentissage d'un CNN directement sur les vues générées (poids initialisés aléatoirement).
3. **CNN pré-entraîné (fine-tuning)** : adaptation d'un ResNet50 pré-entraîné sur ImageNet ; seules les dernières couches sont réentraînées.
4. **CNN pré-entraîné + SVM** : le CNN sert d'extracteur de caractéristiques, remplacé par un SVM polynomial comme classifieur.
5. **Transfer Learning partiel** : gel des couches profondes d'un ResNet50, entraînement uniquement de la dernière couche fully connected.
6. **Fusion SIFT + CNN** : concaténation des descripteurs classiques (SIFT-BoW) et des représentations CNN, suivie d'un SVM polynomial.

3.4 Méthodes quasi-directes (descripteurs 3D + ML)

Ici, on reste dans l'espace 3D sans projection en vues. Après prétraitement, des descripteurs géométriques et statistiques sont extraits :

- Caractéristiques PCA (linéarité, planéité, anisotropie).
- Mesures globales (volume, ratios géométriques, compacité).
- Moments statistiques (moyenne, écart-type, skewness, kurtosis).
- Histogrammes de densité (z-histogramme, histogrammes radiaux, densités multi-échelles).

Ces vecteurs de caractéristiques sont utilisés pour l'apprentissage via un **SVM polynomial**, parfois comparé à un SVM RBF ou à une combinaison (ensemble).

3.5 Méthodes directes (deep learning sur nuages 3D)

Ces méthodes exploitent directement les points bruts :

- **PointNet** : applique des multilayer perceptrons (MLP) sur chaque point puis agrège par max pooling. Invariant à l'ordre des points, simple et efficace, mais limite la capture des structures locales.
- **DGCNN** : extension de PointNet introduisant des graphes k NN mis à jour dynamiquement et des convolutions sur arêtes (EdgeConv). Cela permet de mieux modéliser la géométrie locale, au prix d'un coût de calcul supérieur.

4 Résultats

Dans cette section, nous présentons les performances obtenues par les différentes méthodes testées. Les évaluations ont été réalisées selon un protocole commun détaillé ci-dessous.

4.1 Stratégies d'évaluation

- **Splits** : séparation train/test fixe (80%/20%) avec stratification par classe.
- **Validation croisée** : 5-fold stratifiée sur l'ensemble d'entraînement afin d'optimiser les hyperparamètres (via `GridSearchCV`).

- **Métriques utilisées :**
 - Accuracy : proportion totale de prédictions correctes.
 - Balanced Accuracy : moyenne des rappels par classe, plus robuste en cas de déséquilibre.
 - F1-score (macro et weighted) : mesure harmonique précision/rappel, reflétant la qualité globale des prédictions.
 - Matrices de confusion : analyse des confusions entre classes.
- **Implémentation :**
 - PyTorch pour les modèles PointNet et DGCNN.
 - Scikit-learn pour les SVM et pipelines classiques.
 - Open3D pour le prétraitement des nuages de points.

4.2 Comparaison des méthodes indirectes (2D Multi-vues)

Les résultats obtenus avec les méthodes basées sur des représentations 2D multi-vues sont résumés dans le tableau 4.

Méthode	Accuracy	Balanced Acc.	F1-score (weighted)
SIFT Bag-of-Words + SVM polynomial	0.70	0.70	0.689
CNN from scratch	0.75	0.65	0.654
ResNet50 fine-tuning	0.77	0.75	0.77
ResNet50 features + SVM	0.83	0.739	0.829
CNN pré-entraîné (Transfer Learning)	0.79	0.77	0.76
Fusion SIFT + CNN	0.85	0.77	0.79

TABLE 4 – Résultats comparatifs des méthodes indirectes (2D multi-vues) avec temps d’exécution.

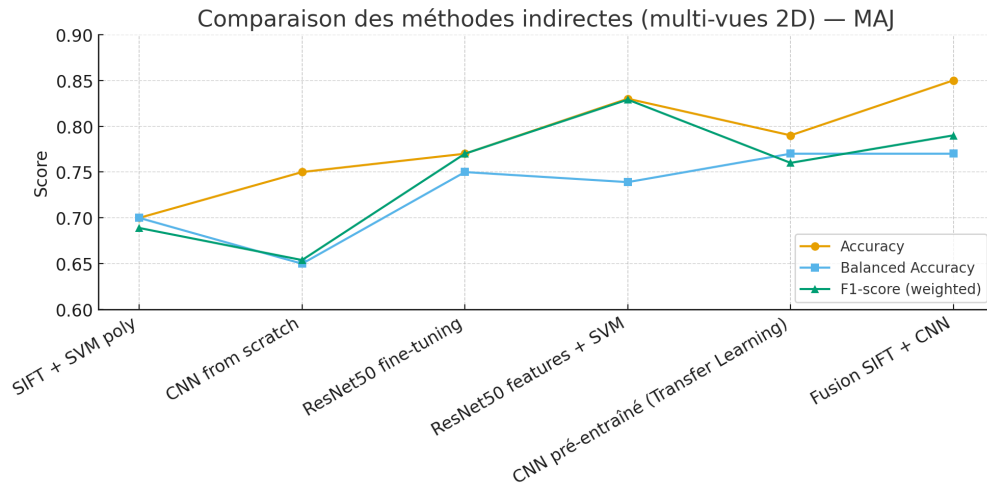


FIGURE 5 – Résultats des méthodes indirectes.

4.3 Méthodes quasi-directes (3D Descripteurs + SVM)

Le tableau suivant présente les résultats des méthodes quasi-directes fondées sur des descripteurs 3D couplés à un SVM ; la Balanced Accuracy est mise en avant pour tenir compte du léger déséquilibre de classes.

Méthode	Accuracy	Balanced Acc.	F1-score (weighted)
PCA Features + SVM poly	0.641	0.548	0.638

TABLE 5 – Résultats des méthodes quasi-directes.

4.4 Méthodes directes (PointNet et DGCNN)

Ce tableau compare les modèles opérant directement sur nuages de points (PointNet, DGCNN) sur le même split ; les scores reportés sont calculés par arbre avec les mêmes métriques que précédemment.

Méthode	Accuracy	Balanced Acc.	F1-score (weighted)
PointNet	0.93	0.93	0.929
DGCNN	0.994	0.994	0.994

TABLE 6 – Résultats des méthodes directes

4.5 PointNet

Les figures ci-dessous montrent l'évolution de la perte et des métriques de validation lors de l'entraînement du modèle PointNet.

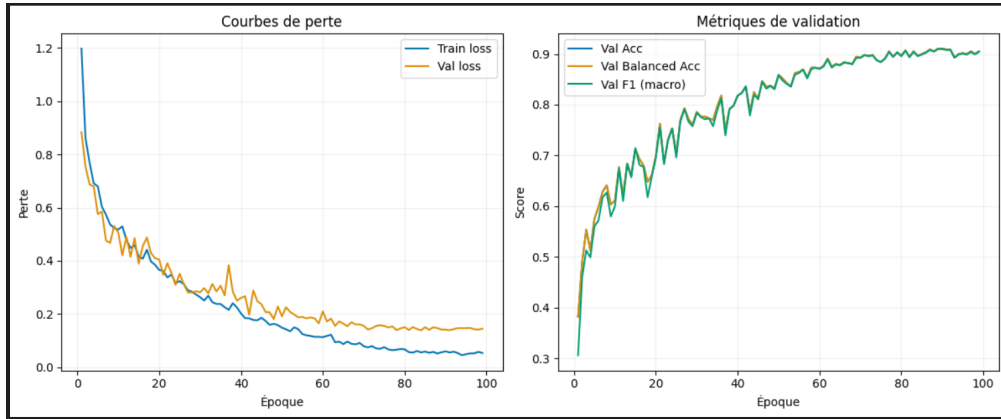


FIGURE 6 – Résultats PointNet.

Les principaux constats issus des courbes sont :

- **Courbes de perte (train vs val) :**
 - Train loss diminue régulièrement de $1,2 \rightarrow \sim 0,05$.
 - Val loss suit la même tendance, légèrement plus haute ($\sim 0,15-0,20$).
 - Pas de sur-apprentissage : les deux baissent en parallèle avec plateau final.
- **Courbes de validation (accuracy / balanced accuracy / F1 macro) :**
 - Progression rapide : $0,3 \rightarrow 0,75$ en ~ 20 époques.

- Plateau vers 0,90–0,91 entre 70–100 époques.
- Superposition des 3 métriques → homogénéité entre classes.
- **Lecture croisée :**
 - Modèle stable et sans sur-apprentissage marqué.
 - Bonne généralisation : $\sim 90\%$ F1 macro en validation.
 - Cohérence val ($\sim 0,90$) vs test ($\sim 0,93$) → modèle robuste et fiable.

4.6 DGCNN

Les figures ci-dessous présentent les résultats de l'entraînement du modèle DGCNN.

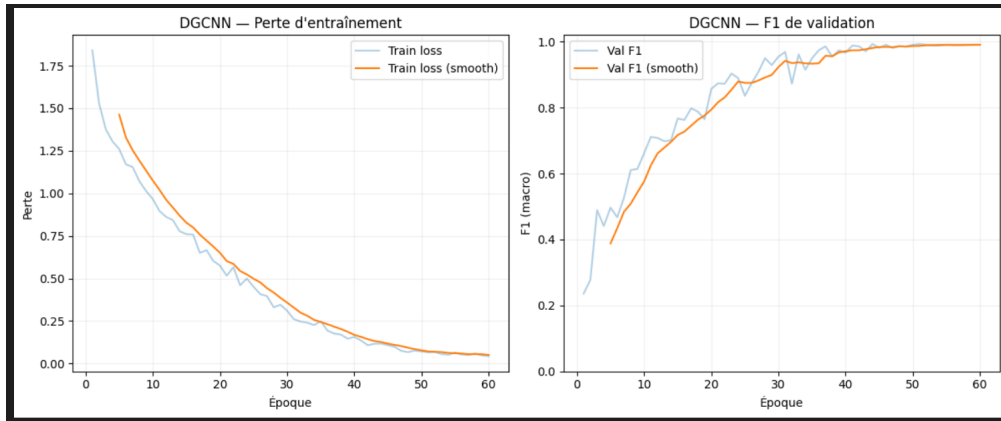


FIGURE 7 – Résultats DGCNN.

Les principaux constats issus des courbes d'apprentissage du DGCNN sont :

- **Perte d'entraînement :**
 - Diminue régulièrement de > 1.8 à < 0.05 .
 - Descente progressive et stable, sans stagnation ni oscillations.
 - Optimisation bien contrôlée, pas de sous-apprentissage.
- **F1-score de validation :**
 - Augmentation rapide : $\sim 0.23 \rightarrow 0.70$ (0–10 époques).
 - Progression régulière jusqu'à ~ 0.90 (25–30 époques).
 - Plateau haut et stable autour de 0.99 (30–60 époques).
- **Analyse croisée :**
 - Pas d'overfitting : perte \downarrow et F1 \uparrow en parallèle.
 - Plateau stable en fin d'entraînement → capacité optimale atteinte.
 - Robustesse confirmée : F1 proche de 1.0 sur validation.

5 Discussion

Les résultats mettent en évidence une hiérarchie claire entre les approches testées :

- **Méthodes directes (PointNet, DGCNN)** : Elles offrent les meilleures performances. PointNet atteint environ 93% d’accuracy tandis que DGCNN frôle les 99%. Leur force réside dans l’exploitation complète de la structure des nuages de points, au prix toutefois d’un coût de calcul plus élevé.
- **Méthodes indirectes (multi-vues 2D)** : Les CNN pré-entraînés et leurs variantes (CNN + SVM) obtiennent de bons résultats (80–88%). Elles bénéficient du transfert d’apprentissage mais perdent une partie de l’information 3D lors de la projection en vues 2D.
- **Méthodes quasi-directes (descripteurs 3D + SVM)** : Avec environ 64% d’accuracy, elles restent les plus légères et interprétables. Elles offrent une alternative rapide et peu coûteuse, mais peinent à capturer la micro-géométrie fine des arbres.

En résumé, les **méthodes directes** sont les plus performantes, les **méthodes indirectes** présentent un bon compromis entre coût et précision, et les **méthodes quasi-directes** conviennent mieux à des environnements contraints où la simplicité prime.

6 Conclusion

Nous avons comparé trois familles d’approches pour la classification d’essences d’arbres à partir de données 3D. Les méthodes **directes** se révèlent les plus performantes, en particulier **DGCNN** qui exploite la géométrie locale via des graphes dynamiques et atteint des scores quasi parfaits sur notre jeu de données. **PointNet** offre un excellent compromis entre simplicité et précision. Les approches **indirectes** multi-vues tirent profit du transfert d’apprentissage et restent compétitives avec un coût maîtrisé. Les méthodes **quasi-directes** (descripteurs 3D + SVM) fournissent une base robuste, rapide et interprétable, mais plafonnent en présence d’essences morphologiquement proches.

Travaux futurs. (i) Élargir et équilibrer le jeu de données (collecte ciblée, augmentation 3D dédiée) ; (ii) tester des variantes plus récentes (PointNet++, *3D Graph/Transformer*) ; (iii) intégrer des signaux supplémentaires (normales, intensité, features radiométriques) ; (iv) explorer des schémas de fusion *multi-vues + 3D direct* ; (v) évaluer en conditions réelles (capteurs, bruit, densités variables) pour valider la robustesse opérationnelle.

Références

- [1] Jonathon Shlens. (2014) *A Tutorial on Principal Component Analysis*. arXiv :1404.1100, <https://arxiv.org/abs/1404.1100>
- [2] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017) *PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation*. arXiv :1612.00593 <https://arxiv.org/abs/1612.00593>
- [3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012) *ImageNet Classification with Deep Convolutional Neural Networks*. https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html