

TP

Conception et Optimisation d'Architectures CNN pour la Classification d'Écorces d'Arbres

Dataset : Bark-101

Département d'Informatique

2024-2025

Table des matières

1	Introduction	2
2	Présentation du Dataset	2
2.1	Caractéristiques des Images	2
3	Objectifs du TP	2
4	Partie 1 : Exploration du Dataset	3
4.1	Questions d'Exploration	3
5	Partie 2 : Conception d'Architectures CNN	3
5.1	Architecture de Base	3
5.2	Questions sur la Conception	3
6	Partie 3 : Optimisation des Hyperparamètres	3
6.1	Hyperparamètres à Optimiser	3
6.2	Questions sur l'Optimisation	4
7	Partie 4 : Évaluation et Comparaison	4
7.1	Métriques d'Évaluation	4
7.2	Questions d'Évaluation	4
8	Barème d'Évaluation	4
9	Conclusion	5
10	Références	5

1 Introduction

Ce projet a pour objectif de vous familiariser avec la conception et l'optimisation d'architectures de réseaux de neurones convolutifs (CNN) pour la classification (multi-classes) d'écorces d'arbres. Vous apprendrez à :

- Comprendre et explorer le jeu de données Bark-101
- Concevoir différentes architectures CNN et comprendre leur impact sur les performances
- Optimiser les hyperparamètres pour améliorer les résultats
- Évaluer et comparer les performances des modèles

2 Présentation du Dataset

Le jeu de données Bark-101 est un ensemble d'images d'écorces d'arbres conçu pour évaluer les algorithmes de reconnaissance de textures et d'écorces dans des conditions naturelles. Il contient 2,587 images appartenant à 101 espèces d'arbres différentes.



FIGURE 1 – Exemples d'images d'écorces du dataset Bark-101

2.1 Caractéristiques des Images

- Format : JPEG/JPG
- Canaux : Images en couleur (3 canaux)
- Défis : Variations d'illumination, présence d'ombres, mousses, et autres éléments naturels
- Variabilité intra-classe : Élevée en raison des différences d'âge et de taille des arbres

3 Objectifs du TP

1. Explorer et comprendre le jeu de données Bark-101
2. Concevoir au moins trois architectures CNN différentes
3. Implémenter une approche basée sur des patches pour améliorer la classification
4. Optimiser les hyperparamètres pour chaque architecture
5. Évaluer et comparer les performances des modèles

6. Analyser l'impact des choix architecturaux sur les performances
7. Implémenter des techniques de visualisation pour interpréter les décisions du modèle

4 Partie 1 : Exploration du Dataset

4.1 Questions d'Exploration

1. Quelle est la distribution des classes dans les ensembles d'entraînement et de test ?
2. Quelles sont les dimensions minimales, maximales et moyennes des images ?
3. Comment pouvez-vous gérer les variations de taille des images ?
4. Quelles techniques de prétraitement seraient appropriées pour ce type d'images de texture ?
5. Comment pouvez-vous augmenter le jeu de données pour améliorer la généralisation ?

5 Partie 2 : Conception d'Architectures CNN

5.1 Architecture de Base

Concevez une architecture CNN de base avec les composants suivants :

- Couches de convolution (nombre, taille des filtres)
- Couches de pooling
- Couches entièrement connectées
- Fonction d'activation
- Régularisation (dropout, batch normalization)

5.2 Questions sur la Conception

1. Comment le nombre de couches de convolution affecte-t-il la capacité du modèle à capturer les textures d'écorces ?
2. Quel est l'impact de la taille des filtres sur la détection des motifs texturaux ?
3. Comment choisir le nombre optimal de patches par image ?
4. Pourquoi utiliser ou ne pas utiliser le transfert d'apprentissage pour ce problème ?
5. Comment les architectures comme VGG, ResNet ou Vision Transformer (ViT) pourraient-elles être adaptées à ce problème ?

6 Partie 3 : Optimisation des Hyperparamètres

6.1 Hyperparamètres à Optimiser

- Taux d'apprentissage
- Taille du batch
- Nombre d'époques
- Optimiseur (Adam, SGD, RMSprop)
- Régularisation (L1, L2, dropout)
- Taille et nombre de patches
- Stratégies d'augmentation de données

6.2 Questions sur l'Optimisation

1. Comment déterminer le taux d'apprentissage optimal ?
2. Quel est l'impact de la taille du batch sur la convergence et la généralisation ?
3. Comment éviter le surapprentissage avec un jeu de données limité ?
4. Quelles techniques d'augmentation de données sont les plus appropriées pour les images d'écorces ?
5. Comment optimiser la stratégie de vote pour la classification basée sur des patches ?

7 Partie 4 : Évaluation et Comparaison

7.1 Métriques d'Évaluation

- Précision (Accuracy)
- Précision par classe (Precision)
- Rappel par classe (Recall)
- F1-Score
- Matrice de confusion

TABLE 1 – Tableau de Comparaison des Modèles

Modèle	Accuracy	Precision	Recall	F1-Score	Temps d'entraînement	Taille du modèle
Modèle 1						
Modèle 2						
Modèle 3						

7.2 Questions d'Évaluation

1. Comment interpréter la matrice de confusion pour un problème à 101 classes ?
2. Quelles classes sont les plus difficiles à distinguer et pourquoi ?
3. Comment évaluer l'impact de l'approche par patches par rapport à l'utilisation d'images entières ?

8 Barème d'Évaluation

TABLE 2 – Barème d'Évaluation du TP

Critère	Description	Points
Architecture du modèle	Conception appropriée des architectures CNN, justification des choix architecturaux, implémentation de l'approche par patches	25
Performance du modèle	Précision globale, métriques par classe, comparaison avec l'état de l'art	25
Qualité d'implémentation	Organisation du code, documentation, reproductibilité	15
Analyse et interprétabilité	Profondeur de l'analyse, qualité des visualisations, interprétation des résultats	25
Rapport technique	Clarté, structure, qualité des explications, références	10

9 Conclusion

Ce projet vous a permis d'explorer la conception et l'optimisation d'architectures CNN pour la classification d'écorces d'arbres. Vous avez appris à :

- Comprendre les spécificités d'un jeu de données de textures naturelles
- Concevoir différentes architectures CNN adaptées au problème
- Implémenter une approche basée sur des patches pour améliorer la classification
- Optimiser les hyperparamètres pour améliorer les performances
- Évaluer et comparer les modèles selon des métriques pertinentes

Ces compétences sont essentielles pour développer des applications d'IA en botanique, foresterie et écologie qui soient à la fois performantes et interprétables.

10 Références

1. Ratajczak, R., Bertrand, S., Crispim-Junior, C., Tougne, L. (2019). Efficient Bark Recognition in the Wild. International Conference on Computer Vision Theory and Applications (VISAPP2019).
2. Misra, D., Crispim-Junior, C., Tougne, L. (2020). Patch-Based CNN Evaluation for Bark Classification. Workshop 2020 European Conference on Computer Vision (ECCV 2020).
3. Yamabe, T., Saitoh, T. (2023). Vision Transformer-Based Bark Image Recognition for Tree Identification. International Conference on Image and Vision Computing New Zealand (IVCNZ 2022).
4. Wu, F., Gazo, R., Benes, B., Haviarova, E. (2021). Deep BarkID : a portable tree bark identification system by knowledge distillation. European Journal of Forest Research, 140, 1391-1399.
5. Selvaraju, R. R., et al. (2017). Grad-CAM : Visual Explanations from Deep Networks via Gradient-based Localization. Proceedings of the IEEE International Conference on Computer Vision (ICCV).