

Rapport de TP3

Vision Artificielle et Traitement des Images

Alan Fresco, Pierre Teodoresco

Novembre 2024



Table des matières

1	Introduction	3
2	AlexNet	4
2.1	Présentation d'AlexNet	4
2.2	Pipeline de prétraitement	4
2.3	Résultats et analyse	4
3	Custom CNN	6
3.1	Architecture du modèle	6
3.2	Résultats et analyse	6
4	Conclusion	8

1 Introduction

L'objectif de ce travail est de mettre en pratique des concepts avancés en vision par ordinateur et en apprentissage profond pour résoudre un problème de classification d'images. Ce rapport explore deux approches distinctes pour classifier un dataset constitué d'images de chiens et de chats :

- L'utilisation d'AlexNet, un modèle de réseau de neurones convolutifs pré-entraîné sur ImageNet.
- La construction et l'entraînement d'un CNN personnalisé (Custom CNN).

L'enjeu principal est de comparer les performances des deux architectures sur un dataset équilibré en termes de classes. Les métriques d'évaluation incluent la loss, l'accuracy globale et les observations sur les courbes d'apprentissage.

Nous avons fait le choix d'un dataset de 1068 images divisé en deux classes distinctes. Il s'agit d'images de chien et de chat. Les modèles d'apprentissage devront réussir à reconnaître et discriminer les deux classes sur l'ensemble d'images.

Le dataset que nous avons téléchargé contenait bien plus d'images (25 000), mais pour des raisons principalement matérielles nous avons restreint notre travail à un nombre plus restreint d'images.

Nous avons utilisé 70% du dataset pour l'entraînement des modèles, 20% pour la validation et 10% pour les tests.

2 AlexNet

2.1 Présentation d'AlexNet

AlexNet est un réseau convolutif largement utilisé pour des tâches de classification. Il se compose de plusieurs couches convolutives, suivies de couches fully connected, et utilise des fonctions d'activation ReLU. Pour ce projet, nous avons utilisé une version pré-entraînée d'AlexNet.

2.2 Pipeline de prétraitement

Le prétraitement des images a consisté à :

- Appliquer des augmentations de données (*RandomHorizontalFlip*, *RandomRotation*, *ColorJitter*).
- Redimensionner les images à 224×224 .
- Normaliser les images en fonction des moyennes et écarts-types.

Ces étapes permettent d'adapter les données à l'architecture d'AlexNet et d'améliorer sa généralisation.

2.3 Résultats et analyse

Après 10 époques d'entraînement, les performances obtenues sont les suivantes :

- **Loss sur le dataset de test** : 0,1077
- **Accuracy sur le dataset de test** : 94,34%
- **Score Kappa sur le dataset de test** : 0,8868

Le score Kappa proche de 0,89 reflète une excellente concordance entre les prédictions d'AlexNet et les labels réels, même après prise en compte des prédictions aléatoires. Ce score souligne la capacité d'AlexNet à généraliser sur le dataset de test.

Les courbes ci-dessous montrent une convergence rapide de la loss et une stabilisation de l'accuracy après environ 6 époques. Ces résultats reflètent la puissance du pré-entraînement d'AlexNet, qui réduit le besoin d'entraîner toutes les couches à partir de zéro.

AlexNet a obtenu des scores élevés pour les deux classes, avec une précision, un recall et un F1-score dépassant 94% pour les deux catégories. Cela démontre que le modèle est équilibré dans sa capacité à prédire correctement les deux classes.

Cependant, AlexNet montre une légère difficulté à améliorer davantage les performances, probablement limitée par la taille restreinte du dataset.

Courbes d'apprentissage pour AlexNet



3 Custom CNN

3.1 Architecture du modèle

Le modèle Custom CNN a été conçu pour s'adapter au dataset. Il est constitué de :

- Trois couches convolutionnelles avec des filtres de taille 3×3 et des activations ReLU.
- Une couche de *max-pooling* pour réduire la taille des cartes de caractéristiques.
- Deux couches fully connected pour classifier les images.
- Une couche *Dropout* (0.5) pour réduire l'overfitting.

Le code du modèle est détaillé dans le fichier `homemade_cnn.py`.

3.2 Résultats et analyse

Après 10 époques d'entraînement, les performances obtenues pour le Custom CNN sont les suivantes :

- **Loss sur le dataset de test** : 0,6293
- **Accuracy sur le dataset de test** : 66,98%
- **Score Kappa sur le dataset de test** : 0,3396

Le score Kappa de 0,3396 indique une concordance modérée entre les prédictions du modèle et les labels réels, bien que les performances soient inférieures à celles d'AlexNet. Cette performance limitée est attribuable à l'absence de pré-entraînement, ainsi qu'à la simplicité relative de l'architecture.

Les courbes d'apprentissage du Custom CNN montrent une convergence plus lente que celles d'AlexNet. De plus, la différence notable entre les pertes d'entraînement et de validation suggère un problème de généralisation. Cela pourrait être dû à la capacité limitée du modèle à capturer les caractéristiques nécessaires pour discriminer correctement les deux classes.

Contrairement à AlexNet, le Custom CNN montre une précision, un recall et un F1-score bien inférieur à ceux obtenus avec AlexNet. Malgré ces différences, les trois paramètres restent cohérent aux résultats obtenus par le Custom CNN (qui eux mêmes sont inférieurs à AlexNet).

Pour améliorer ces résultats, des pistes comme l'ajustement des paramètres, l'ajout de régularisation, ou l'utilisation d'un dataset plus large pourraient être explorées.

Courbes d'apprentissage pour Custom CNN



4 Conclusion

Ce projet a permis d'analyser deux approches pour la classification d'images : un modèle pré-entraîné (AlexNet) et un modèle personnalisé (Custom CNN). Les résultats obtenus montrent que :

- AlexNet a surperformé le modèle personnalisé, avec une **accuracy** de 94,34% et un **score Kappa** de 0,8868, démontrant une grande capacité à généraliser sur les données de test.
- Le Custom CNN, malgré sa simplicité, a atteint une **accuracy** de 66,98% et un **score Kappa** de 0,3396. Ces résultats indiquent un potentiel d'amélioration, notamment avec des optimisations d'architecture ou un pré-entraînement.

Les courbes d'apprentissage montrent une convergence rapide et stable pour AlexNet, tandis que le Custom CNN présente des signes de sous-apprentissage ou de mauvaise généralisation. Ces observations soulignent l'importance des modèles pré-entraînés dans des contextes où les ressources ou les données sont limitées.

Pour améliorer les performances des deux modèles, plusieurs pistes peuvent être envisagées :

- Augmenter la taille du dataset en exploitant des images supplémentaires.
- Optimiser les hyperparamètres pour le Custom CNN, comme le taux d'apprentissage ou l'utilisation de régularisations supplémentaires.
- Tester des architectures alternatives, comme ResNet ou VGG, pour évaluer leurs performances par rapport à AlexNet.
- Utiliser des techniques avancées comme le fine-tuning d'AlexNet pour adapter encore mieux le modèle au dataset spécifique.

En conclusion, AlexNet s'est avéré être un choix robuste pour cette tâche, tandis que le Custom CNN constitue une base intéressante pour un apprentissage sur des datasets plus complexes.