



TP2 RAPPORT INF5081

Pour le 30 Avril 2024



AMADOU SARA BAH
[BAHA09019703]
Tamrat Beede Mikael
[TAMB89080102]

1. Introduction

Contexte général du projet :

Ce projet constitue un travail pratique qui explore l'application de l'apprentissage automatique dans le domaine de l'analyse d'images. Plus précisément, il se concentre sur le développement de modèles de réseaux de neurones convolutifs (CNN) capables de classer des images de différentes espèces marines, dans notre cas les requins, les baleines et les requins-baleines. Le projet utilise un ensemble de données composé de 15 000 images, dont 12 000 sont destinées à l'apprentissage et 3 000 aux tests, représentant ces trois espèces. Ces images, issues de diverses sources, présentent un défi significatif en termes de classification en raison de la variabilité des contextes marins dans lesquels ces espèces sont photographiées. Nous souhaitons développer une méthode efficace pour distinguer ces espèces les unes des autres.

Objectif du travail :

L'objectif de ce projet est de concevoir, entraîner, et évaluer un réseau de neurones à convolution (CNN) qui peut identifier avec précision ces trois espèces marines à partir des images. Le projet nécessite l'adaptation d'un code préexistant, initialement conçu pour la classification de chiffres manuscrits. Pour atteindre cet objectif, plusieurs étapes sont nécessaires :

1. **Adaptation du Code Existant** : Modifier `1_Modele.py` et `2_Evaluation.py` qui initialement étaient conçus pour différencier les chiffres 2 et 7, pour les ajuster à la classification des espèces marines. Cela comprend la reconfiguration des fonctions de traitement des caractéristiques et des couches pleinement connectées pour répondre aux exigences du nouveau problème de classification.
2. **Préparation et augmentation des données** : cela inclut le traitement des images pour augmenter la diversité des données d'entraînement et améliorer ainsi la robustesse du modèle.
3. **Définition et optimisation de l'architecture du CNN** : cette étape implique la sélection des couches, des fonctions d'activation, des couches de pooling et des paramètres de manière à maximiser la capacité du réseau à extraire des caractéristiques pertinentes des images.
4. **Entraînement et validation du modèle** : le CNN est entraîné sur les données préparées, en utilisant une portion des données pour la validation afin de surveiller et prévenir le surajustement.
5. **Évaluation du modèle sur les données de test** : une fois le modèle entraîné, il est testé sur des nouveaux ensemble de données non vue pour évaluer sa performance générale, notamment sa précision et sa capacité à généraliser à de nouvelles images.
6. **Analyse des résultats et ajustements** : basée sur les performances observées, cette phase vise à affiner le modèle en ajustant ses paramètres ou son architecture. Une fois le réajustement fait on répète les étape 2 à 6 jusqu'à on est satisfait.

2. Montage de l'architecture et entraînement du modèle:

2.1 Ensemble de données

Le modèle a été entraîné et validé sur un ensemble de 15 000 images réparties équitablement entre les trois classes marines. Les données d'entraînement

comprenaient 12 000 images (4 000 par classe), avec une division en sous-ensembles de formation (9 600 images) et de validation (2 400 images), soit un taux de 20% utilisé pour la validation. De plus, 3 000 images réservées aux tests étaient réparties équitablement entre les trois classes marines.

2.2 Traitement de données

Pour renforcer la capacité de généralisation du modèle, des techniques de data augmentation ont été utilisées, notamment des rotations, translations, zooms et retournements horizontaux. Pour plus de précision, voici les détails des changements ou ajouts :

- **rotation_range=24 (Ajout):**
 - **Objectif :** Augmenter la robustesse du modèle face aux rotations de l'image. Ici les images peuvent être pivotées aléatoirement de +/- 24 degrés. Cela permet au modèle de mieux généraliser à des images où les objets ne sont pas parfaitement alignés verticalement ou horizontalement.
- **width_shift_range=0.2 et height_shift_range=0.2 (Ajout):**
 - **Objectif :** Augmenter la robustesse du modèle aux variations de position des objets dans les images. Ici les images peuvent être traduites horizontalement et verticalement. Une translation de 20% de la largeur ou de la hauteur de l'image est appliquée aléatoirement. Cela aide le modèle à reconnaître les objets peu importe où ils sont positionnés dans l'image.
- **shear_range=0.2:**
 - **Objectif :** Simuler une inclinaison des objets dans les images, pour augmenter la capacité du modèle à identifier des objets présentés sous différents angles. Dans notre cas on applique un cisaillement aléatoire jusqu'à 20%. Le cisaillement déforme l'image le long de l'axe x ou y, ce qui change l'angle des objets dans l'image.
- **zoom_range=0.2:**
 - **Objectif :** Préparer le modèle à reconnaître des objets de différentes tailles et à différentes distances. Dans notre cas, les images peuvent être agrandies ou réduites aléatoirement jusqu'à 20%. Cela simule le fait de voir le même objet à différentes distances ou tailles.

2.3 Paramètres et Hyperparamètres

Le modèle a été compilé avec l'optimiseur Adam, utilisant un taux d'apprentissage (learning rate) défini à 0.001. Notre modèle a également été entraîné avec des lots (batch size) de 32 images, et ce, sur 40 époques.

2.4 Architecture

Notre réseau de neurones convolutif (CNN) est spécifiquement conçu pour traiter les images en couleur (RGB) de taille 150x150 pixels. Cette section détaille l'architecture adoptée pour la classification des images de trois espèces marines différentes.

On commence par l'**Entrée**, l'architecture débute par une entrée prenant des images de 150x150 pixels en format RGB.

On passe ensuite à l'**Extraction des caractéristiques**, avec la première couche convolutionnelle qui utilise 32 filtres pour scanner l'image, appliquant une fonction d'activation Relu afin d'introduire une non-linéarité nécessaire pour capter les nuances dans les données. Cette couche est suivie d'une opération de pooling de taille 2x2 qui réduit les dimensions des feature maps de 150x150 à 75x75.

La deuxième couche convolutionnelle augmente le nombre de filtres à 64 pour extraire des caractéristiques plus complexes. Elle utilise également la fonction d'activation Relu et est suivie par un pooling similaire qui réduit les dimensions de 75x75 à 37x37.

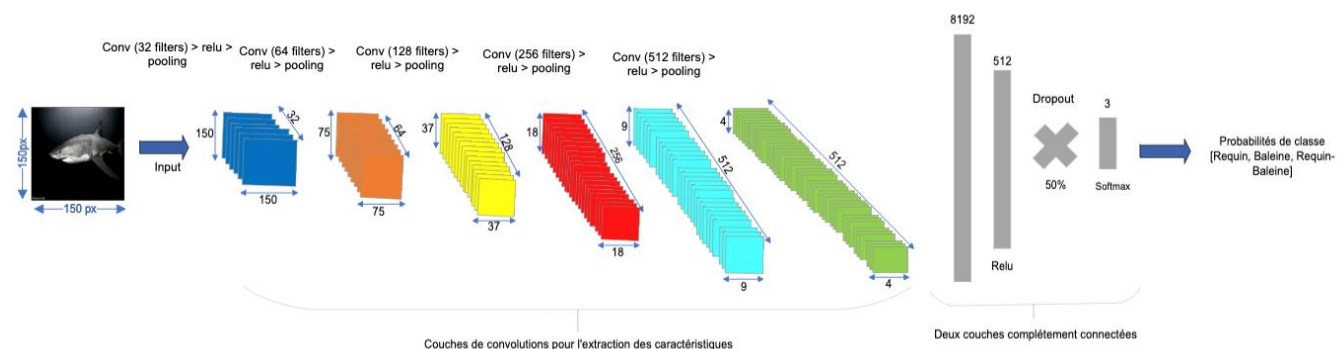
La troisième couche poursuit l'augmentation du nombre de filtres à 128, continuant d'utiliser la fonction d'activation Relu. Le pooling associé réduit encore les dimensions des cartes de caractéristiques de 37x37 à 18x18.

Dans la quatrième couche, le nombre de filtres est porté à 256. Comme les couches précédentes, elle emploie la fonction d'activation Relu et le pooling réduit les dimensions à 9x9.

La cinquième couche convolutionnelle, et la dernière de cette série, utilise 512 filtres. Elle maintient la fonction d'activation Relu et après le pooling, les dimensions des cartes de caractéristiques sont réduites à 4x4.

On a ensuite les **Couches Complètement Connectées**, où les cartes de caractéristiques issues de la dernière couche de pooling sont transformées en un seul vecteur long dans une opération appelée "Flatten". Ce vecteur de 8192 éléments alimente ensuite la première couche dense qui compte 512 neurones, utilisant la fonction d'activation **Relu**. Un **dropout** de 50% est appliqué à cette couche pour prévenir le surapprentissage.

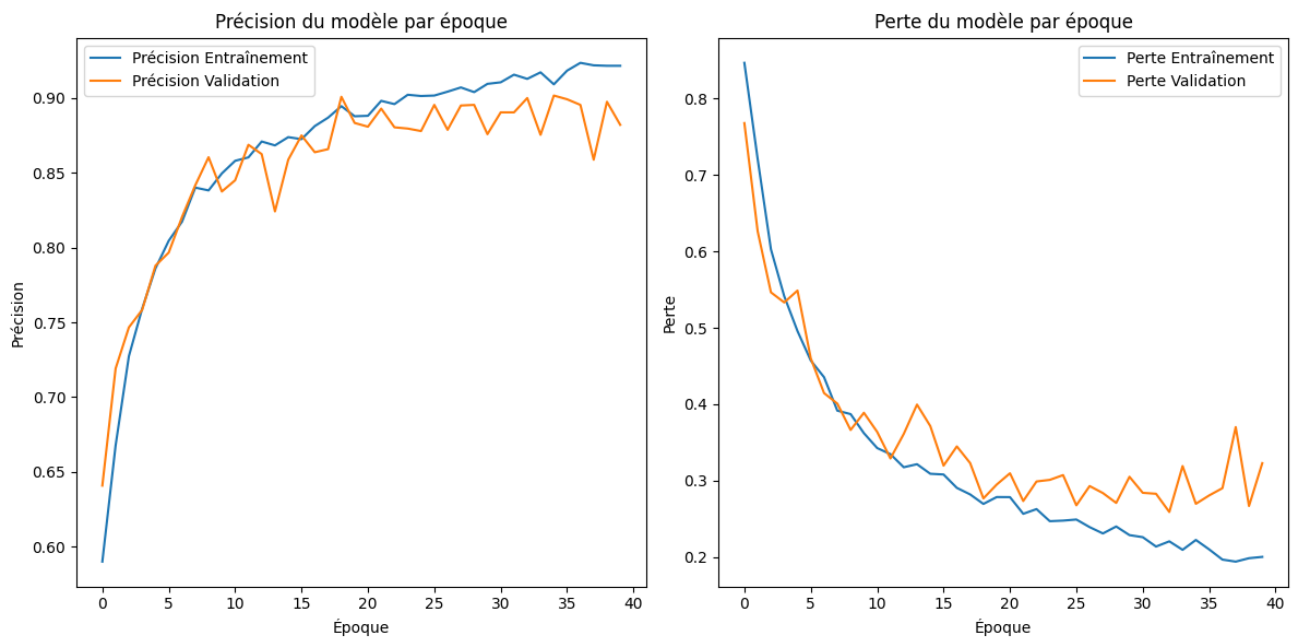
La seconde couche dense se compose de trois unités, correspondant aux trois classes de sortie, et utilise une fonction d'activation **Softmax** pour générer une distribution de probabilité sur les trois classes possibles, permettant une classification multi-classe. Cette architecture élaborée permet au réseau non seulement de distinguer clairement entre les espèces marines mais aussi de traiter efficacement les variations complexes dans les images d'entrée, assurant ainsi une classification précise et robuste.



2.5 Affichage des résultats d'entraînement

Une fois avoir lancer notre code on obtient les résultats suivants :

- **Le temps total d'entraînement en minutes** : L'entraînement du modèle a duré environ 2734.39 secondes, ce qui équivaut à environ 45.57 minutes.
- **L'erreur minimale commise lors de l'entraînement (Minimum Loss)** : L'erreur la plus faible enregistrée lors de l'entraînement était de 0.1940, atteinte à l'époque 38.
- **L'exactitude maximale de l'entraînement (Maximum Accuracy)** : La plus haute précision atteinte lors de l'entraînement était de 90.167% (0.90167), observée à l'époque 35.



2.6 Justification du choix de l'architecture

L'architecture de notre réseau de neurones convolutifs (CNN) a été élaborée pour répondre aux défis spécifiques posés par la classification d'images de différentes espèces marines. Lors de notre premier essai, nous avons observé que certaines des images mal classifiées n'étaient pas très claires, pour être plus précis les caractéristiques des espèces n'étaient pas clairement distinguables. Cette observation a souligné l'importance et le besoin de l'augmentation des images pour améliorer la capacité de généralisation du modèle. On a donc fait ceci :

Utilisation de la data augmentation : Nous avons implémenté plusieurs techniques de data augmentation. Ces techniques incluent des rotations, des translations horizontales et verticales, des zooms et des retournements horizontaux. Par exemple, l'ajout de rotations (`rotation_range=24`) aide le modèle à reconnaître des espèces marines peu importe leur orientation dans l'image, qui est utile dans notre projet car les espèces marines peuvent être photographiées sous différents angles en fonction de leur mouvement et de la position du photographe.

Les translations (`width_shift_range=0.2` et `height_shift_range=0.2`) et le zoom (`zoom_range=0.2`) permettent au modèle de mieux identifier les caractéristiques pertinentes des espèces même lorsqu'elles sont partiellement hors du centre ou de tailles variées. Avec tous ces choix nous avons pu obtenir de meilleur résultat lors des classifications.

Une fois que nous avons pensé avoir trouvé les meilleurs choix pour le Data augmentation, nous nous sommes ensuite conduit à optimiser l'architecture du CNN :

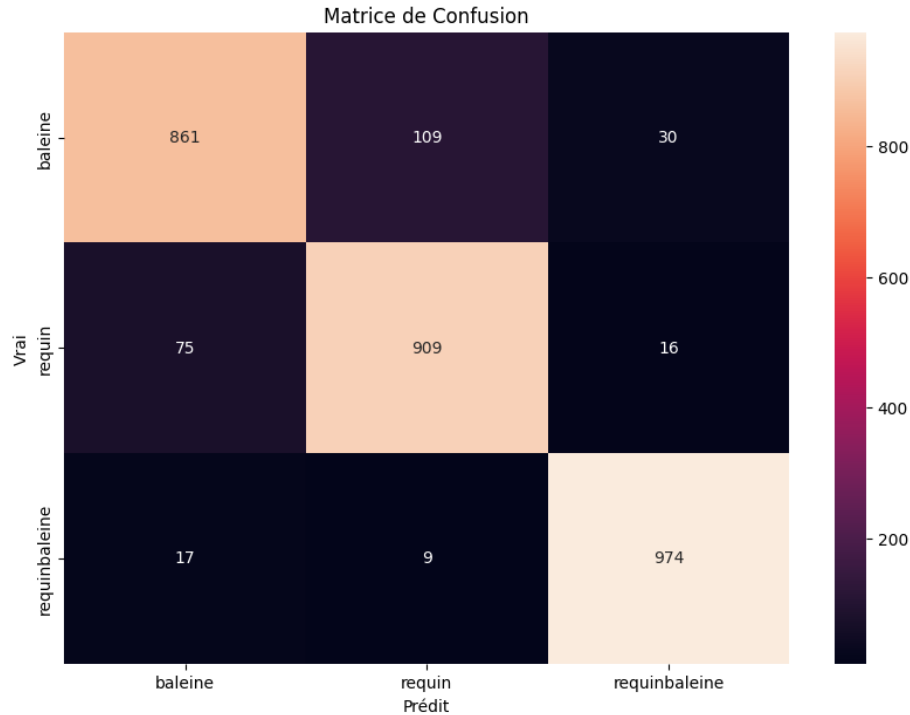
Optimisation de l'architecture du CNN : En ce qui concerne la structure même du CNN, nous avons expérimenté avec un nombre varié de couches convolutives et de neurones. Initialement, on commence avec un nombre de filtres modéré et en augmentant progressivement, nous permettons de capturer d'abord des

caractéristiques générales comme les bords et les textures, puis des détails plus complexes et spécifiques tels que les formes particulières aux espèces marines. Les couches d'activation ReLU (implémentées dans chaque couche convolutive) ont été choisies pour leur efficacité à introduire la non-linéarité dans le processus d'apprentissage, elle aide aussi à prévenir les problèmes de disparition des gradients, permettant ainsi au réseau d'apprendre de façons plus rapides et plus efficaces. Les couches de pooling (MaxPooling2D) réduisent la dimensionnalité des cartes de caractéristiques tout en conservant les informations essentielles. Cela réduit le nombre de paramètres et de calculs nécessaires dans les couches subséquentes, ce qui améliore l'efficacité du modèle.

En conclusion, notre approche était de construire un modèle capable non seulement de distinguer clairement entre les espèces marines mais aussi d'être robuste face aux variations communes dans les images naturelles. Chaque choix a été justifié par son impact direct sur la performance et la précision du modèle, guidé par une analyse continue des résultats d'entraînement et de validation.

3. Évaluation du Modèle

L'exactitude (accuracy) du modèle développé sur les données de test : Le modèle a atteint une précision de 91.47% sur l'ensemble de test, ce qui dépasse l'objectif minimal de 89% fixé pour ce projet. Cette haute précision indique une excellente capacité du modèle à généraliser à de nouvelles données jamais vues durant l'entraînement, mais elle montre aussi qu'il y a une grande marge pour l'amélioration.





4. Conclusion

Le développement de notre modèle CNN a permis de classer efficacement les images de différentes espèces marines avec une haute précision. Mais, malgré les succès obtenus, plusieurs défis restent à relever et des améliorations sont envisageables pour optimiser davantage la performance du système.

Problèmes rencontrés : Durant ce projet, nous avons été confrontés à des défis multiples qui ont impacté le processus de développement et d'évaluation de notre modèle.

1. **Temps de traitement élevé :** La recherche de la meilleure configuration pour notre modèle a été ralentie par les temps de traitement nécessaires pour chaque essai. En moyenne, l'entraînement d'une seule itération prenait entre 25 et 45 minutes. Cette contrainte de temps a limité notre capacité à expérimenter rapidement avec différentes architectures ou paramètres de data augmentation.
2. **Inconsistance des résultats :** Nous avons également observé une certaine inconsistance dans les résultats d'entraînement à travers différentes sessions. Cette variabilité a compliqué la tâche de déterminer les caractéristiques précises et les paramètres optimaux du modèle.
3. **Limitations de Google Colab :** L'utilisation de Google Colab, bien que ceci aide pour accéder à des ressources de calcul gratuites, il a introduit des restrictions supplémentaires. La limite de temps sur les sessions actives et la variabilité des

allocations de ressources GPU ont parfois résulté en des interruptions inattendues des sessions d'entraînement, ce qui a nécessité des redémarrages fréquents et a rendu le processus d'itération plus difficile.

Améliorations possibles:

1. **Augmentation des données** : Bien que nous ayons déjà implémenté des techniques de data augmentation comme la rotation et le zoom, d'autres méthodes pourraient être explorées pour améliorer le modèle. Par exemple, l'ajustement de la luminosité et du contraste pourrait aider le modèle à mieux performer sous différentes conditions d'éclairage. De même, l'application de filtres de bruit pourrait préparer le modèle à reconnaître les espèces dans des conditions sous-marines moins idéales.
2. **Modèles plus profonds** : Explorer des architectures de réseau plus profondes pourrait également contribuer à capter des caractéristiques plus complexes et améliorer la classification. Cependant, cela nécessite une attention particulière pour éviter le surajustement, notamment à travers des techniques de régularisation plus poussées comme le dropout ou la normalisation par lots.
3. **Enrichissement du jeu de données** : L'ajout de nouvelles images provenant de sources diversifiées ou l'augmentation du nombre d'images par espèce pourrait également aider à mieux équilibrer le jeu de données et à améliorer les capacités de généralisation du modèle.
4. **Feedback continu et apprentissage par renforcement** : Intégrer un système de feedback où les utilisateurs pourraient corriger les classifications erronées pourrait servir à affiner continuellement le modèle en exploitant l'apprentissage par renforcement.

En conclusion, tout en reconnaissant les succès du modèle actuel, Il faut reconnaître qu'il faudrait continuer à explorer ces avenues pour améliorer la précision et la robustesse du modèle face à des données encore plus large de conditions d'imagerie marine. Cela inclut une attention particulière à l'adaptation du modèle pour qu'il puisse traiter efficacement des données encore non vues, assurant ainsi sa pertinence et son applicabilité dans des situations réelles variées.