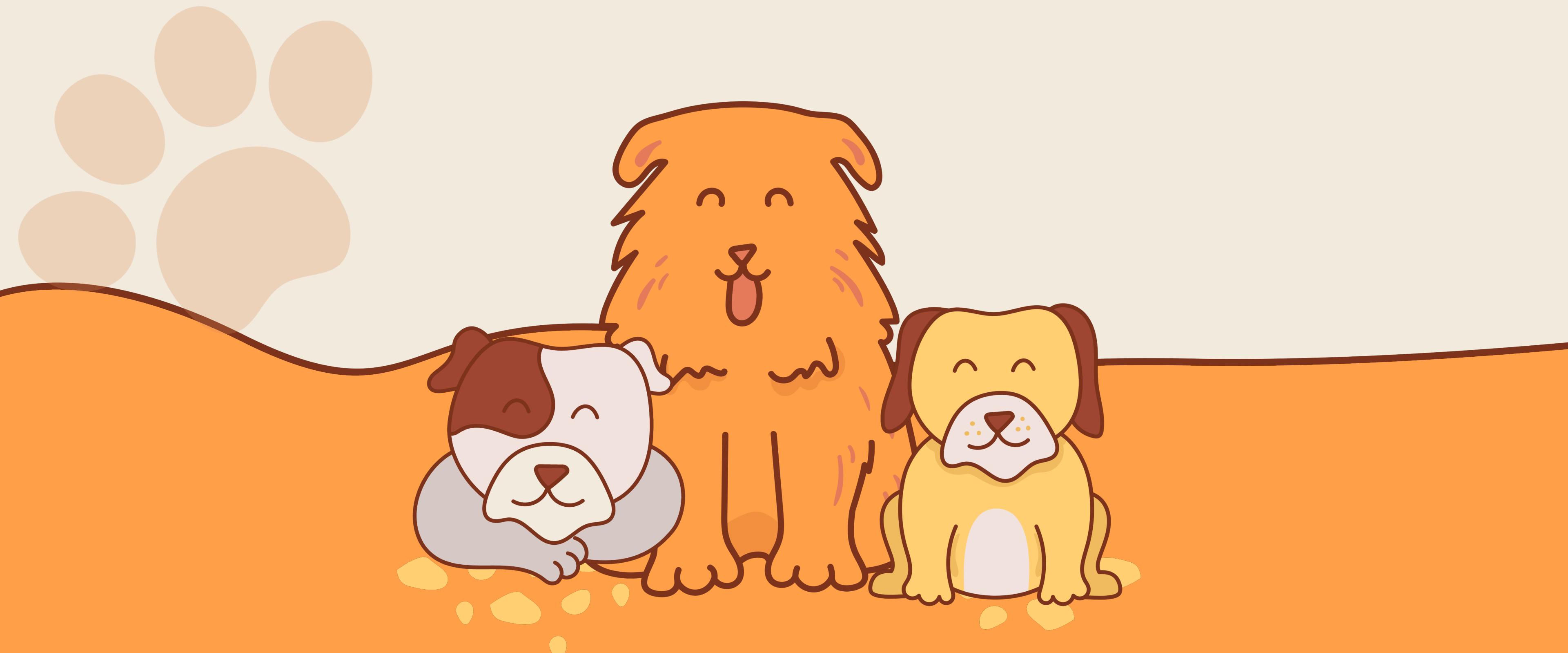
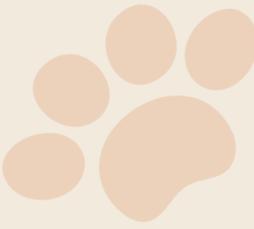


CLASSEZ DES IMAGES À L'AIDE D'ALGORITHMES DE DEEP LEARNING

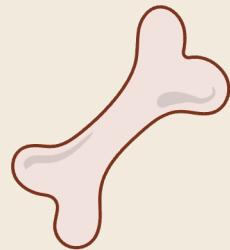


PARCOURS MACHINE LEARNING ENGINEER



Introduction

Dans le cadre de notre bénévolat pour l'association Le Refuge, nous allons développer un algorithme de classification d'images pour détecter la race des chiens à partir de photos, afin de faciliter l'indexation des pensionnaires. Pour ce faire, nous utiliserons le Stanford Dogs Dataset et mettrons en œuvre deux approches : la création d'un réseau CNN personnalisé et l'utilisation du transfer learning. L'objectif est de créer un outil efficace qui accélérera le travail des bénévoles et améliorera la gestion des données de l'association.



Objectifs

- Préparer les images en utilisant des techniques de prétraitement .
- Appliquer des techniques de data augmentation
- Développement d'Algorithmes de Classification d'Images
- Évaluation et Optimisation du Modèle
- Intégration et Démonstration



Le jeu de données

Pour ce projet, nous utiliserons le Stanford Dogs Dataset. Ce dataset provient de l'université de Stanford et contient des images de 120 races de chiens différentes, fournissant une large diversité pour l'entraînement des modèles de classification. Cependant, en raison des limitations de puissance de calcul de notre matériel, nous nous concentrerons uniquement sur trois races de chiens. Cette sélection restreinte nous permettra de développer et tester efficacement notre algorithme tout en assurant des performances optimales avec les ressources disponibles.



Chihuahua



Boxer



Eskimo dog

Preparation des images

Cette étape consiste à appliquer diverses techniques pour améliorer la qualité des données d'entraînement et ainsi optimiser les performances du modèle.

Ces techniques sont essentielles pour plusieurs raisons :

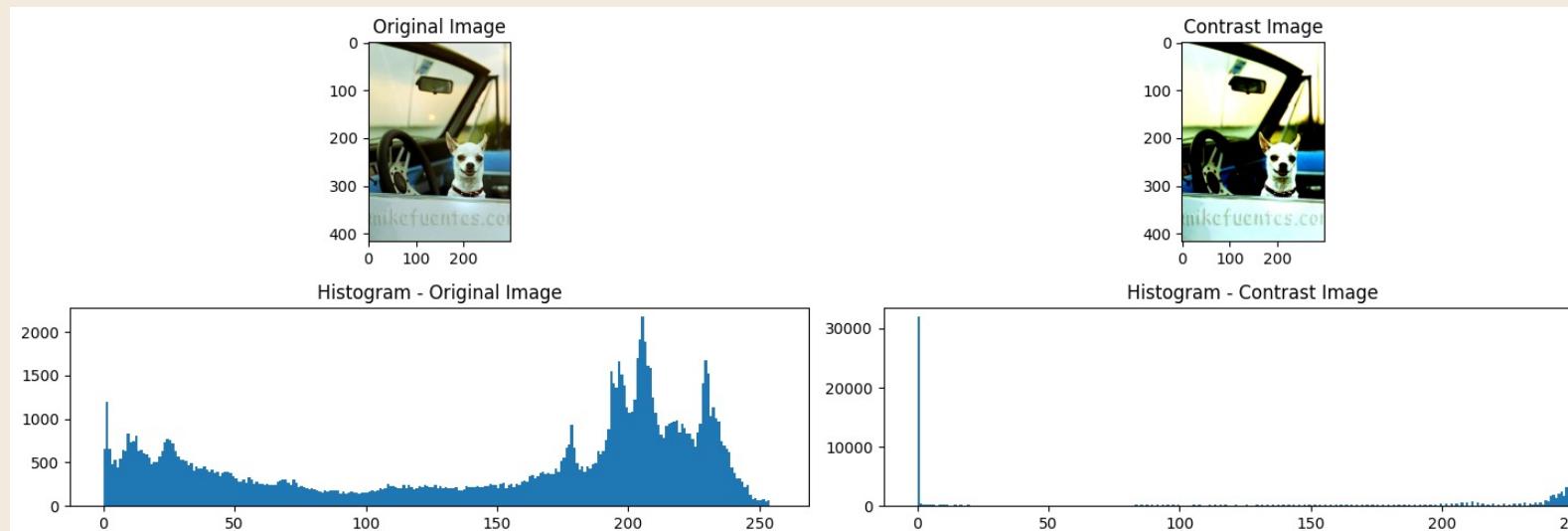
Amélioration de la Robustesse : Elles permettent au modèle de généraliser mieux en lui fournissant des variations d'images qu'il pourrait rencontrer en pratique.

Réduction du Surapprentissage : En augmentant le nombre de données d'entraînement, on réduit le risque que le modèle surapprenne des détails spécifiques aux images d'entraînement.

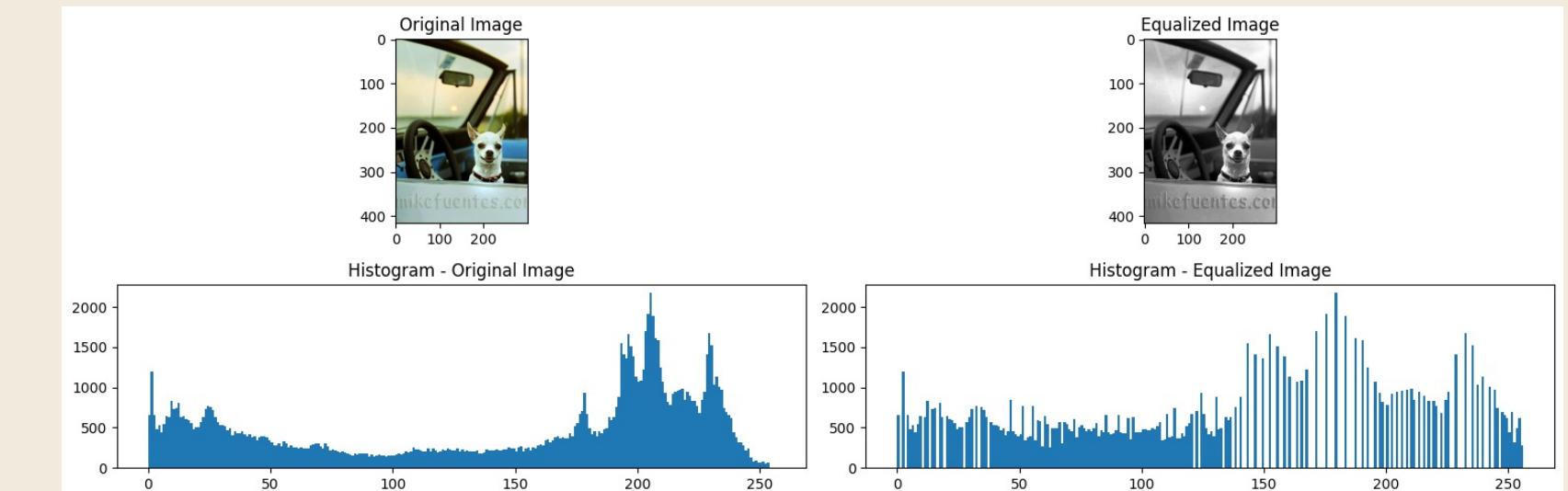
Optimisation des Performances : Un bon prétraitement accélère la convergence de l'algorithme et améliore la précision des prédictions.



Le contraste dans le prétraitement d'images consiste à ajuster la différence entre les zones claires et sombres d'une image pour améliorer la visibilité des détails.



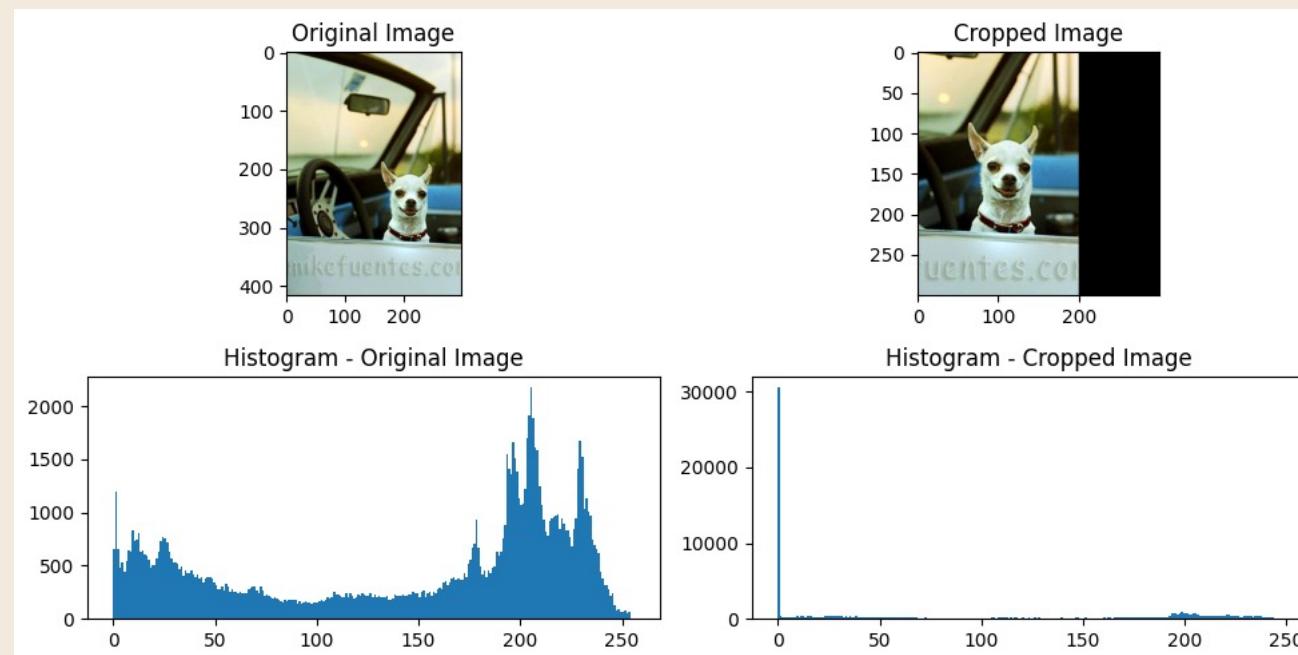
L'equalization dans le prétraitement d'images consiste à redistribuer les niveaux de gris pour uniformiser l'histogramme et améliorer le contraste de l'image.



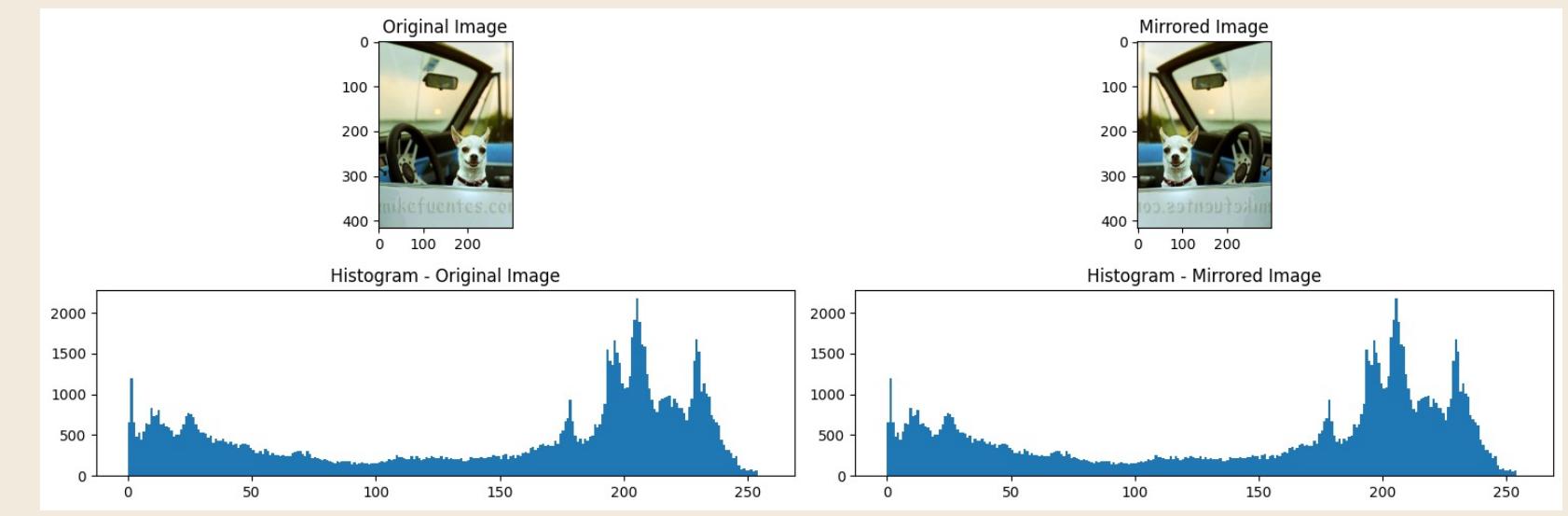
Preparation des images



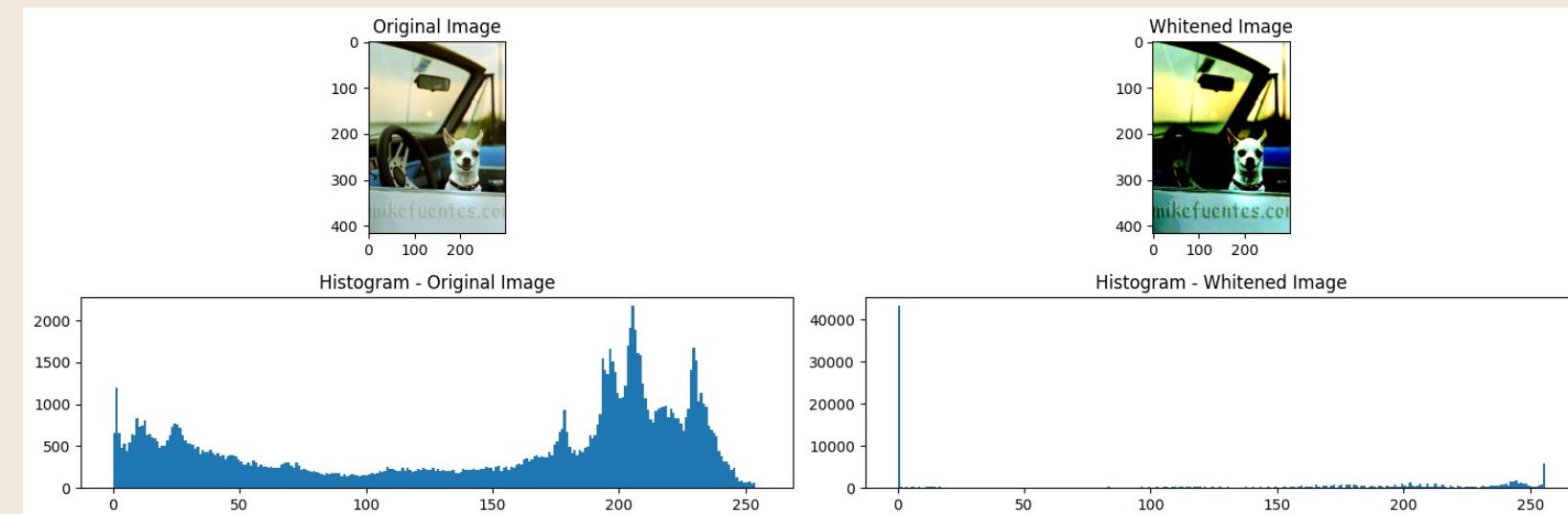
Le Cropping consiste à découper une partie spécifique d'une image pour se concentrer sur une région d'intérêt ou supprimer des éléments indésirables.



Le Mirroring permet de créer une image miroir en inversant l'image original horizontalement ou verticalement pour augmenter les données et améliorer la robustesse des modèles.



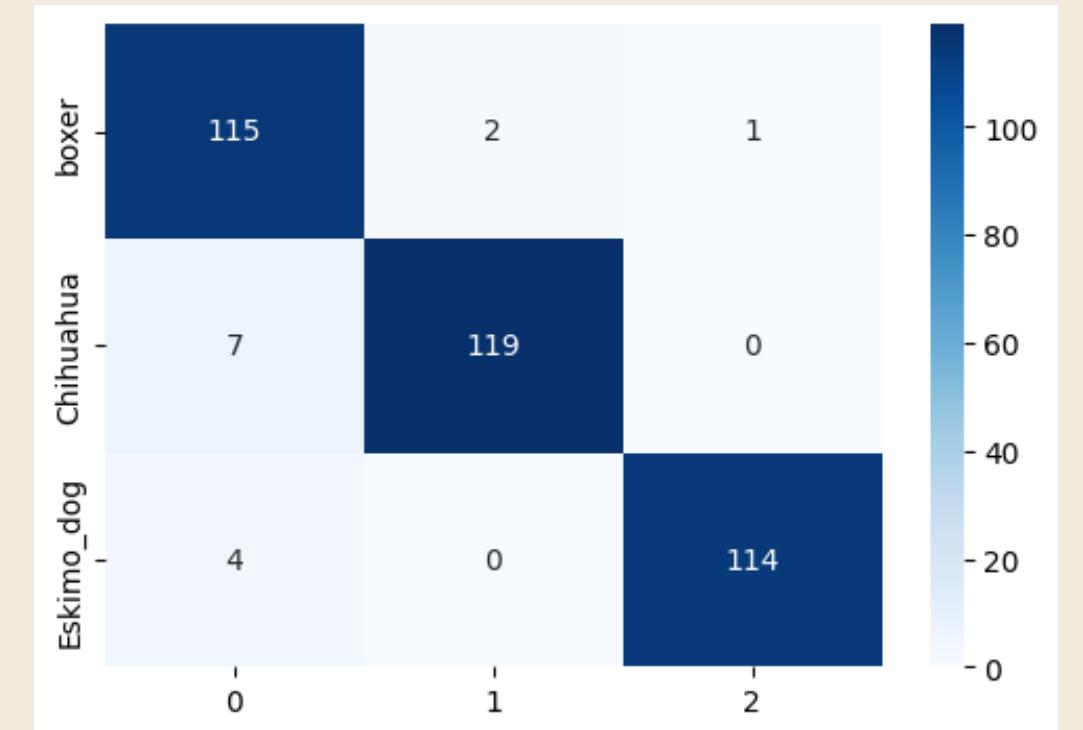
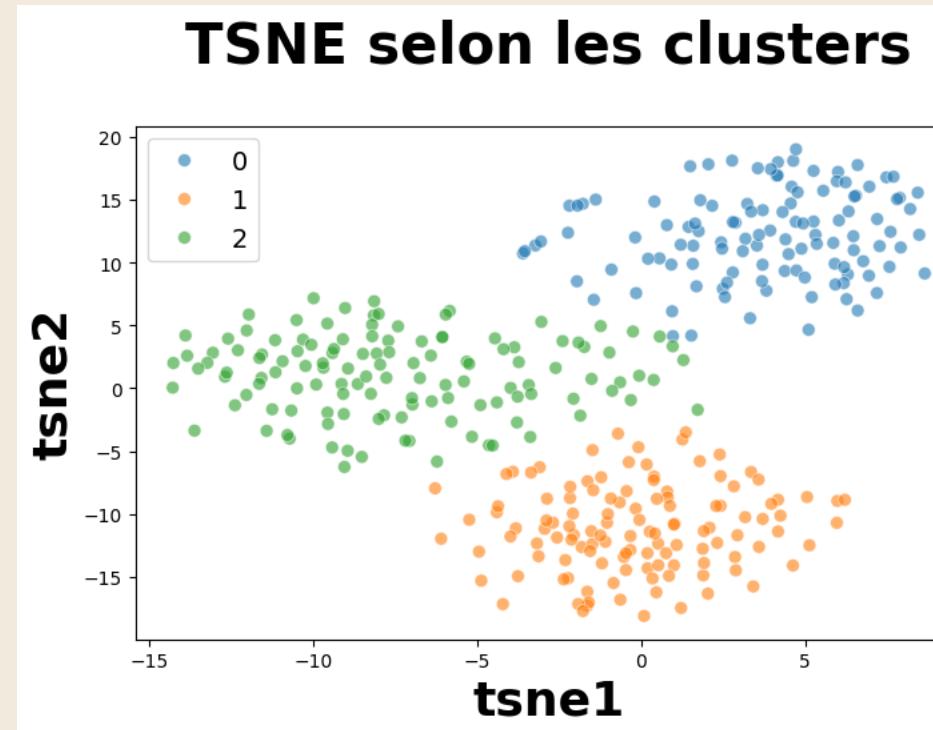
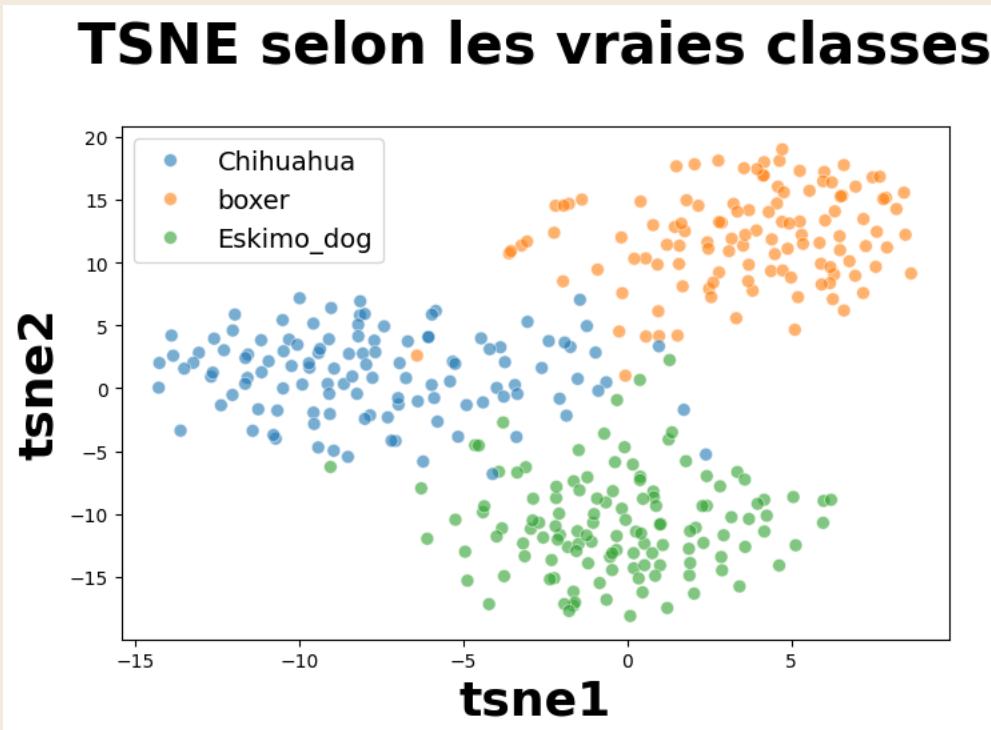
Le whitening consiste à normaliser les valeurs des pixels pour avoir une moyenne de zéro et une variance unitaire, réduisant ainsi les redondances et améliorant les performances des algorithmes d'apprentissage.





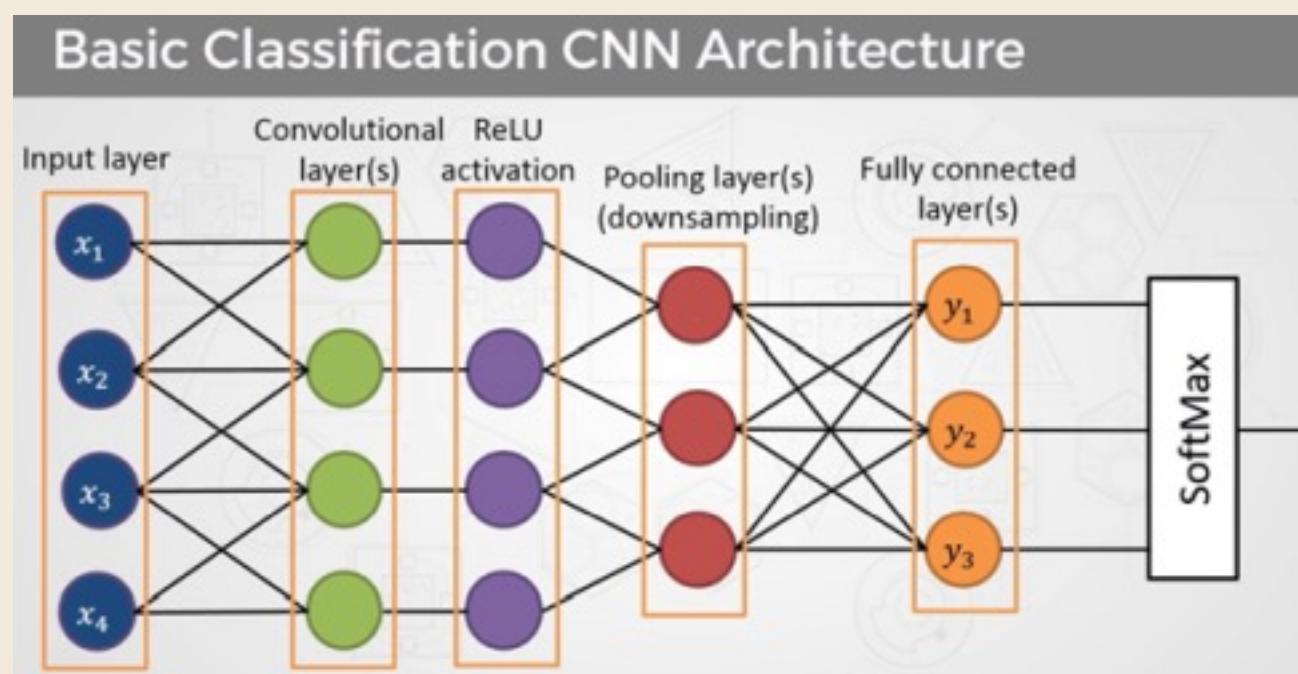
Etude de faisabilité

Avant toute chose, il est nécessaire de s'assurer de la faisabilité de la tache demandée. Un CNN pre-entraîné, ici VGG16, a été utilisé pour prédire nos races de chien, puis une réduction de dimension a été appliquée. Nous pouvons visuellement constater que les différentes races de chiens possèdent des caractéristiques différentes, ce qui devrait permettre d'en prédire un bon nombre de manière correcte, ce qui est confirmé par la matrice de confusion.



Réseaux de neurones convolutifs

Les Convolutional Neural Networks (CNN), ou réseaux de neurones convolutifs en français, sont une classe de réseaux de neurones artificiels principalement utilisés pour le traitement et l'analyse d'images, bien qu'ils puissent également être appliqués à d'autres types de données.



<https://www.jeveuxetredatascientist.fr/convolutional-neural-network/>



Création d'un CNN

Structure du CNN

• Couches Convolutives:

- Ce sont les premières couches du réseau. Elles appliquent des filtres pour extraire les caractéristiques importantes des images, comme les contours et les textures.

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 222, 222, 64)	1,792
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 64)	0
conv2d_10 (Conv2D)	(None, 109, 109, 64)	36,928
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_11 (Conv2D)	(None, 52, 52, 128)	73,856
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 128)	0
dense_6 (Dense)	(None, 32)	4,128
dropout_3 (Dropout)	(None, 32)	0
dense_7 (Dense)	(None, 3)	99

• Couches de Max-Pooling:

- Réduisent la taille de l'image en ne conservant que les informations les plus importantes (comme les points de caractéristiques les plus significatifs).

• Global Average Pooling:

- Cette couche moyenne les valeurs des caractéristiques restantes après les convolutions.
- Elle prépare l'information pour la classification finale en réduisant encore plus la dimensionnalité.

• Couches Denses (Fully Connected):

- Ces couches prennent les caractéristiques extraites et les utilisent pour faire des prédictions.

• Dropout:

- Cette technique est utilisée pour réduire le surapprentissage en désactivant aléatoirement une partie des neurones pendant l'entraînement.



Tunning des hyperparamètres

Les hyperparamètres influencent directement la capacité de généralisation et la précision d'un modèle. Régler les hyperparamètres permet de trouver la meilleure combinaison qui minimise l'erreur de prédiction tout en évitant le surapprentissage.



- Nombre de Filtres dans les Couches de Convolution

`conv_X_filters` : Nombre de filtres dans la X eme couche de convolution.

À quoi ça sert ? Les filtres dans une couche de convolution détectent les caractéristiques (features) des images, comme les contours, les textures, etc. Ajuster leur nombre permet de contrôler la complexité et la capacité du modèle à capturer des informations importantes dans les images.



-Nombre de Neurones dans la Couche Dense

`dense_units` : Nombre de neurones dans la couche entièrement connectée (dense).

À quoi ça sert ? Les neurones dans la couche dense combinent les features extraites par les couches de convolution pour effectuer la classification. Ajuster leur nombre impacte la capacité du modèle à intégrer ces informations pour effectuer des prédictions précises.



Taux de Dropout

`dropout` : Pourcentage de neurones désactivés aléatoirement dans une couche pendant l'entraînement.

À quoi ça sert ? Le dropout aide à prévenir le surapprentissage (overfitting) en forçant le modèle à ne pas trop dépendre d'un ensemble particulier de neurones.



Optimiseur

`optimizer` : Algorithme utilisé pour ajuster les poids du modèle pendant l'entraînement (options : adam, sgd, rmsprop).

À quoi ça sert ? L'optimiseur détermine comment les poids du réseau sont ajustés lors de la rétropropagation. Différents optimisateurs ont des avantages spécifiques, comme la vitesse de convergence (adam) ou la simplicité et robustesse (sgd).



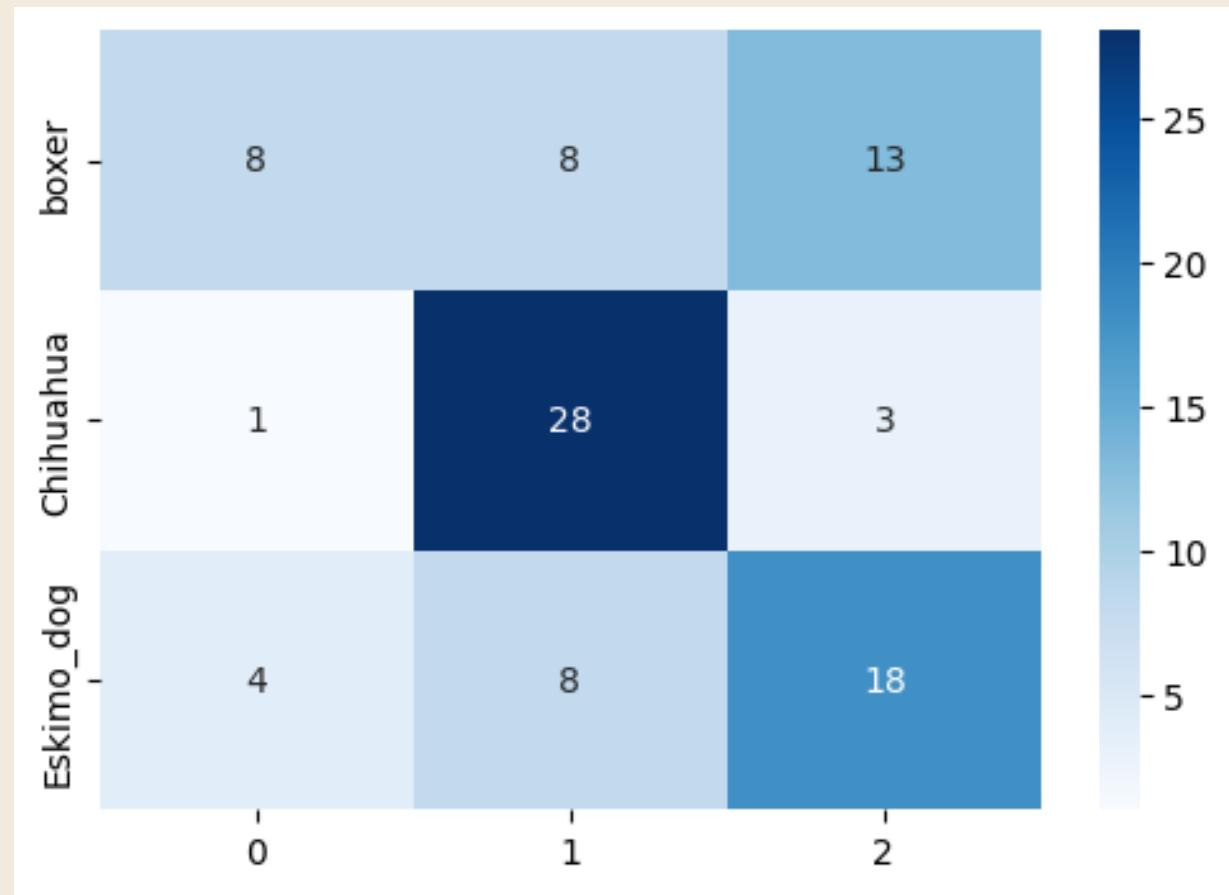
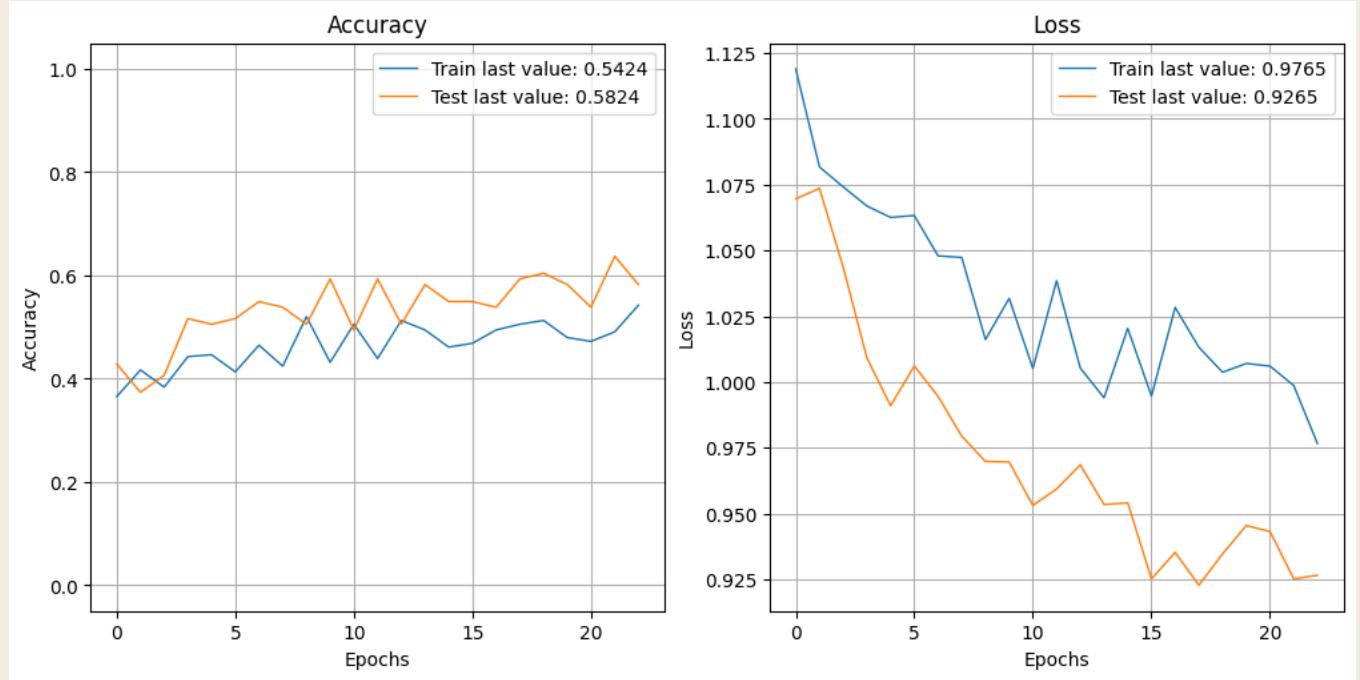
Taux d'Apprentissage (Learning Rate)

`learning_rate` : Taux auquel les poids du modèle sont mis à jour pendant l'entraînement.

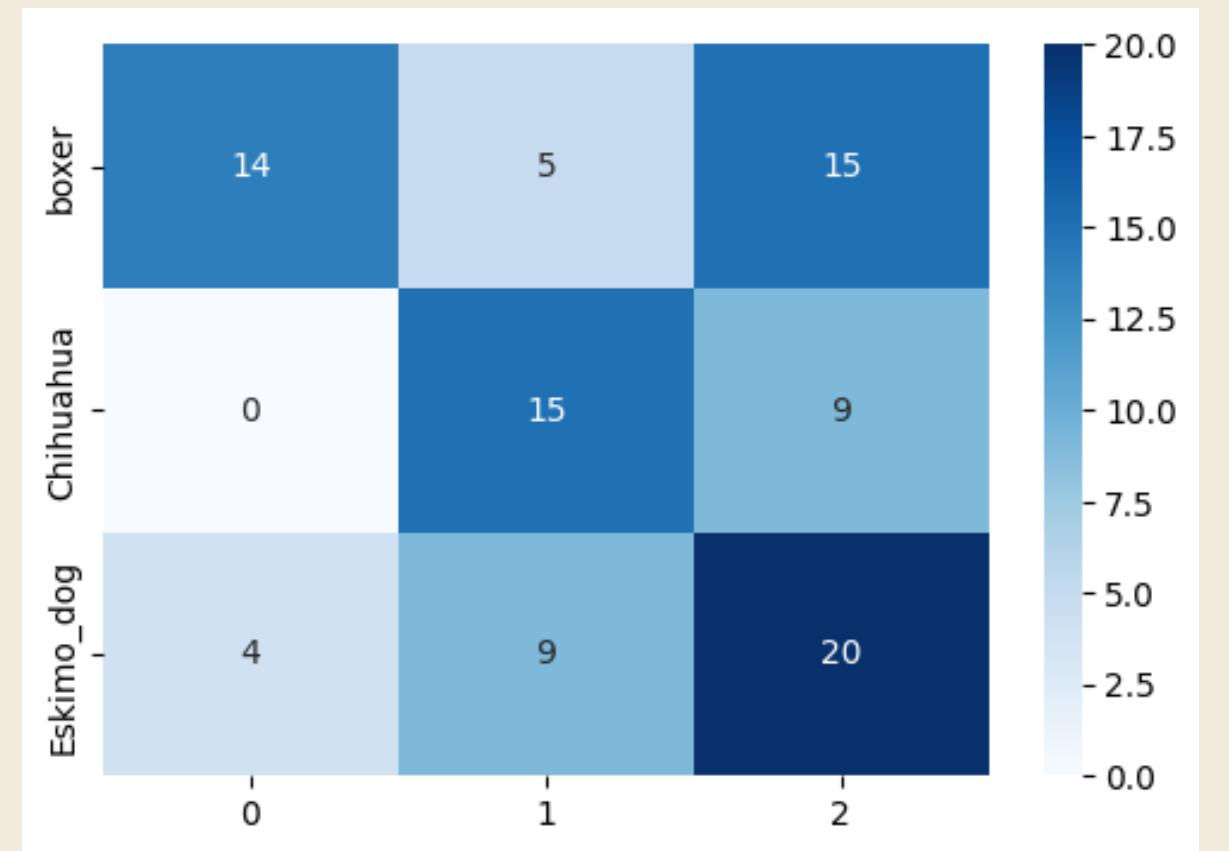
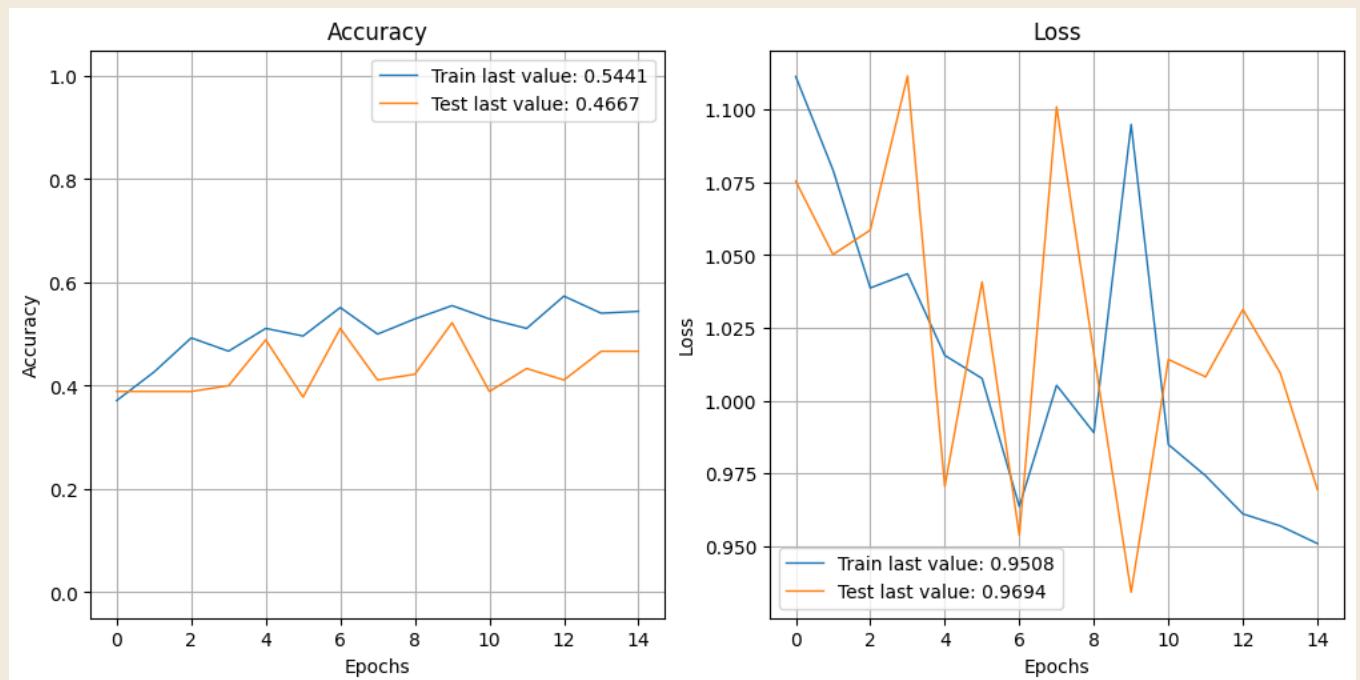
À quoi ça sert ? Le taux d'apprentissage contrôle l'ampleur des mises à jour des poids après chaque itération. Un taux trop élevé peut entraîner une convergence trop rapide et instable, tandis qu'un taux trop bas peut rendre l'entraînement lent ou empêcher la convergence.



Résultats sans Data augmentation



Résultats avec Data augmentation





Principe de transfer learning

Le transfert learning (ou apprentissage par transfert) est une méthode de machine learning où un modèle développé pour une tâche est réutilisé comme point de départ pour une tâche différente mais liée. Au lieu de partir de zéro, on utilise les connaissances acquises par un modèle pré-entraîné pour résoudre un nouveau problème.

- Les dernières couches du modèle sont remplacées par de nouvelles couches adaptées pour classer les images en trois races de chiens spécifiques.
- Le modèle est ré-entraîné sur un ensemble de données contenant des images des trois races de chiens.

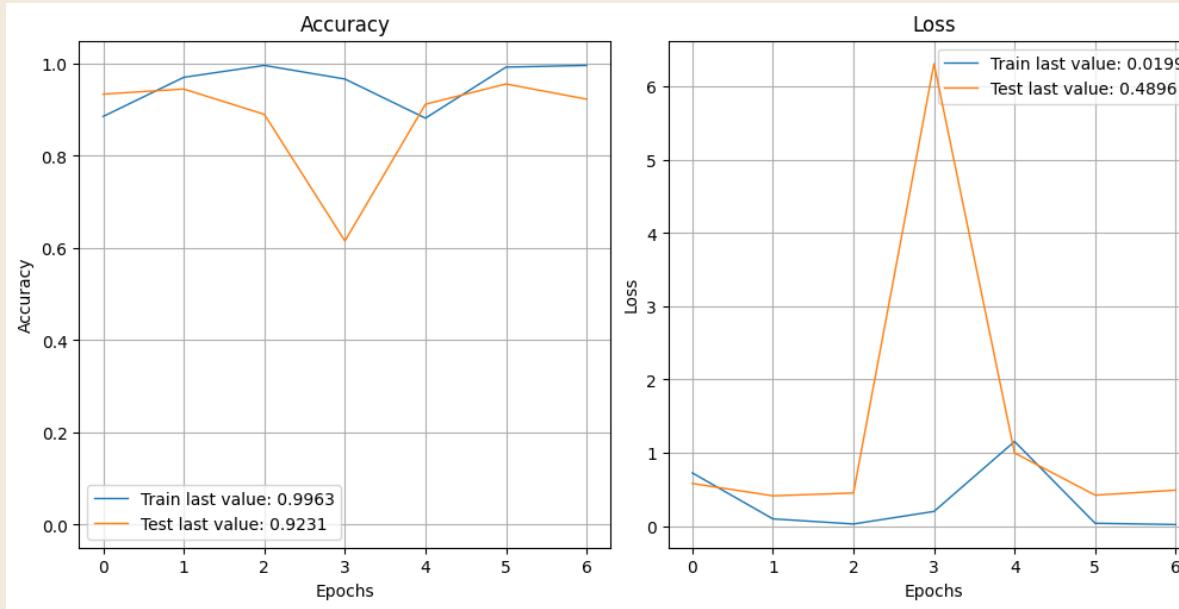


Transfer learning VGG16

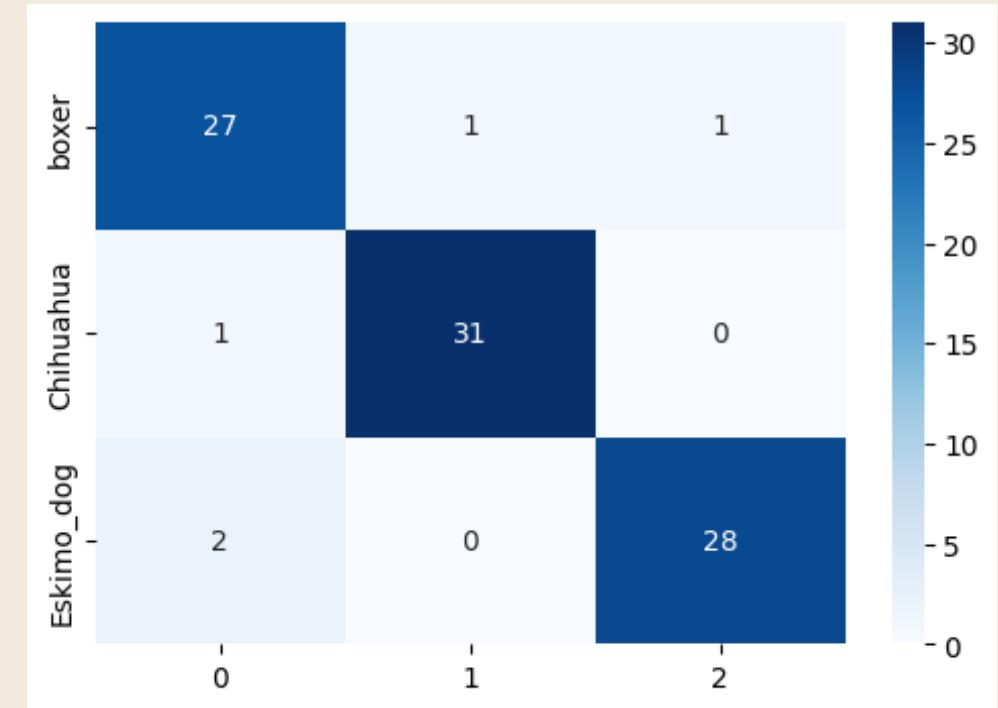
Fonctionnement: VGG16 est conçu pour être simple et efficace, en utilisant de petites convolutions (3x3) pour capturer les caractéristiques de l'image. Les couches convolutives et de pooling réduisent progressivement la taille de l'image tout en augmentant la profondeur des caractéristiques, permettant au modèle d'apprendre des représentations complexes.



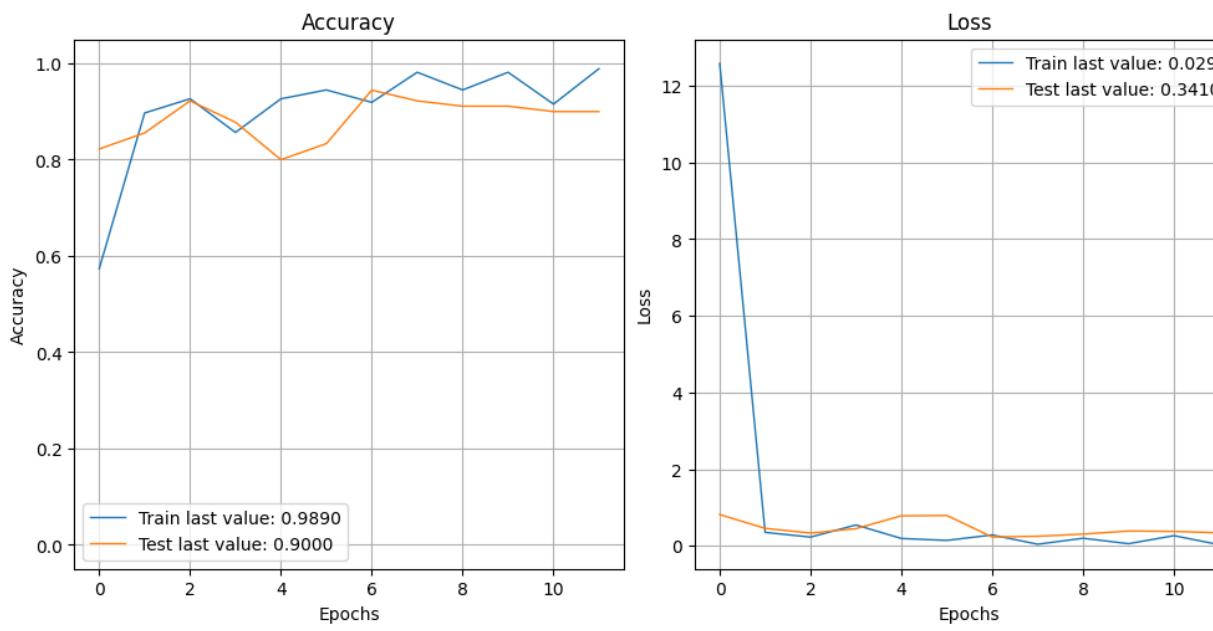
Sans data augmentation



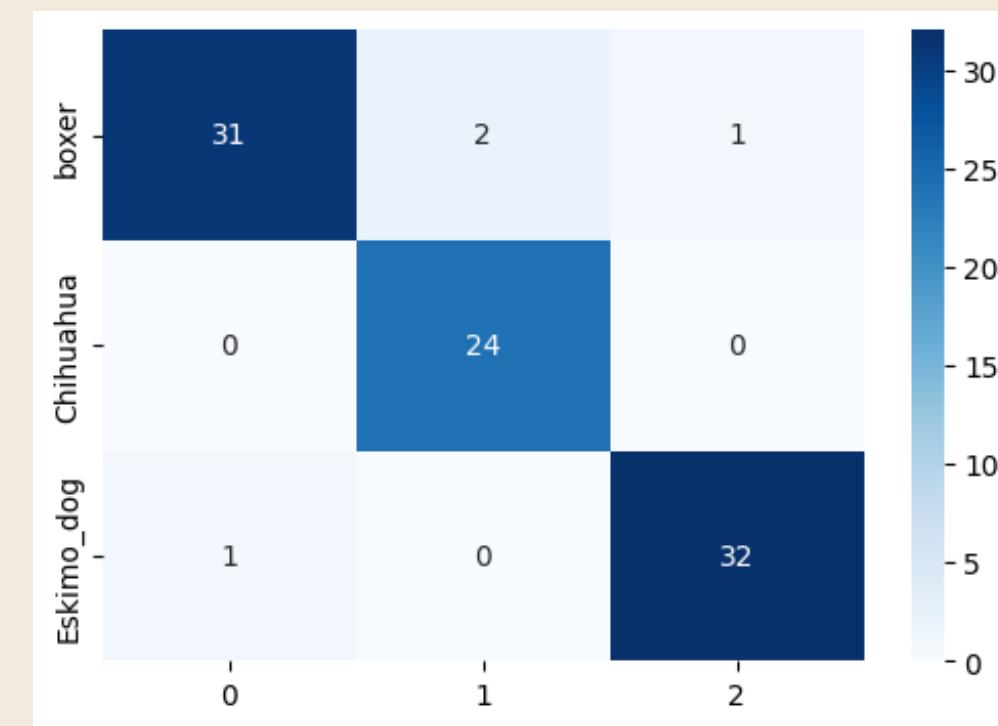
Validation Accuracy :
0.9451
Validation Loss : **0.4126**
Test Accuracy : **0.9560**
Test Loss : **0.2121**



Avec data augmentation



Validation Accuracy :
0.9333
Validation Loss :
0.1936
Test Accuracy : **0.9560**
Test Loss : **0.1684**

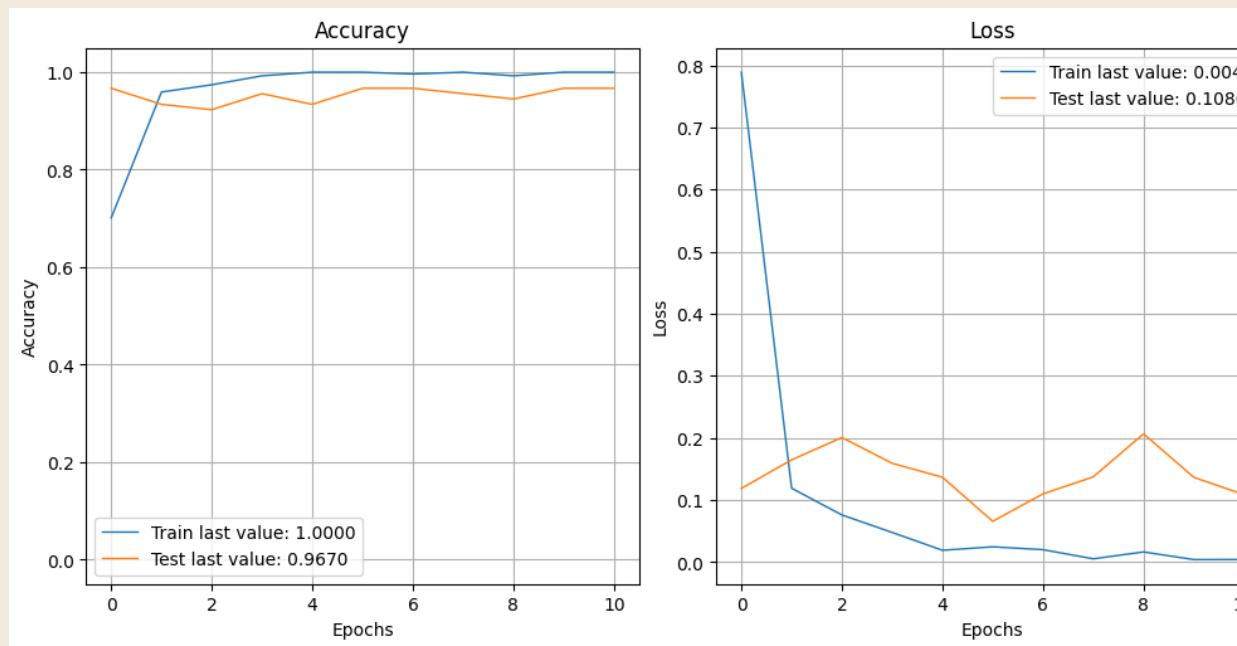




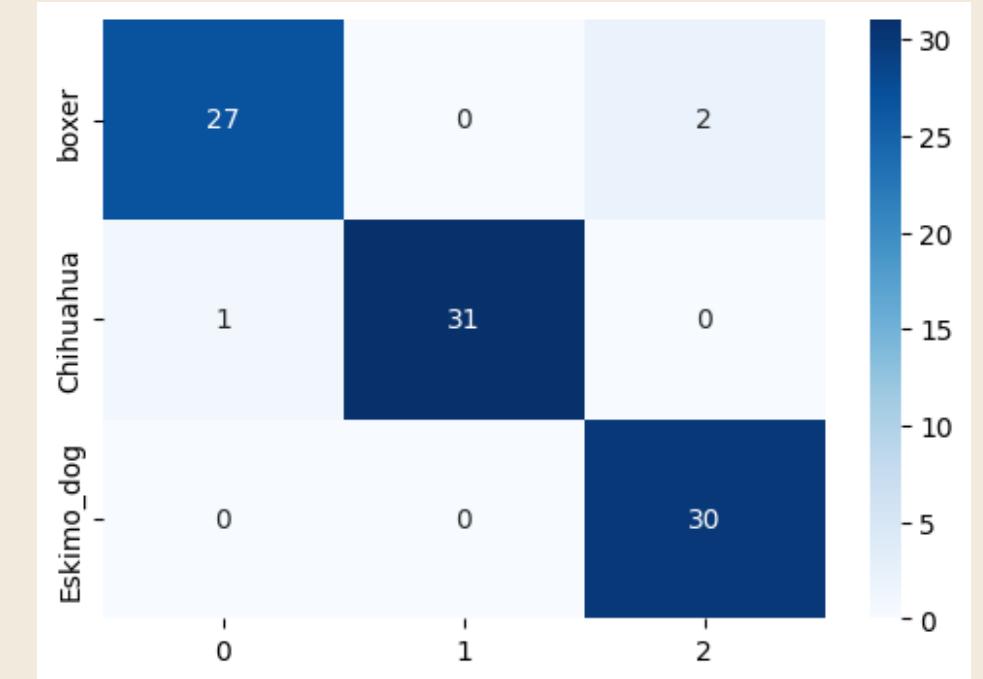
Transfer learning ResNet50

ResNet50 utilise des connexions résiduelles pour lutter contre le problème de la dégradation dans les réseaux profonds. En permettant le passage direct des gradients pendant l'entraînement, ResNet facilite l'apprentissage de réseaux beaucoup plus profonds sans perte de performance.

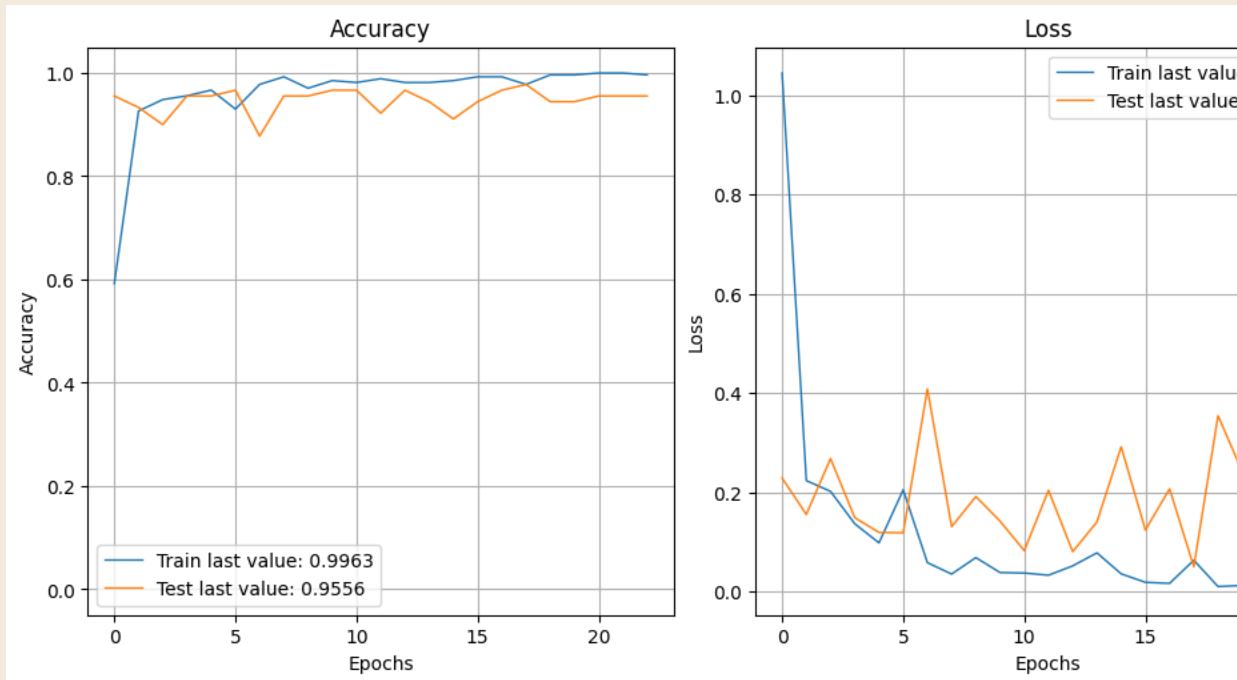
Sans data augmentation



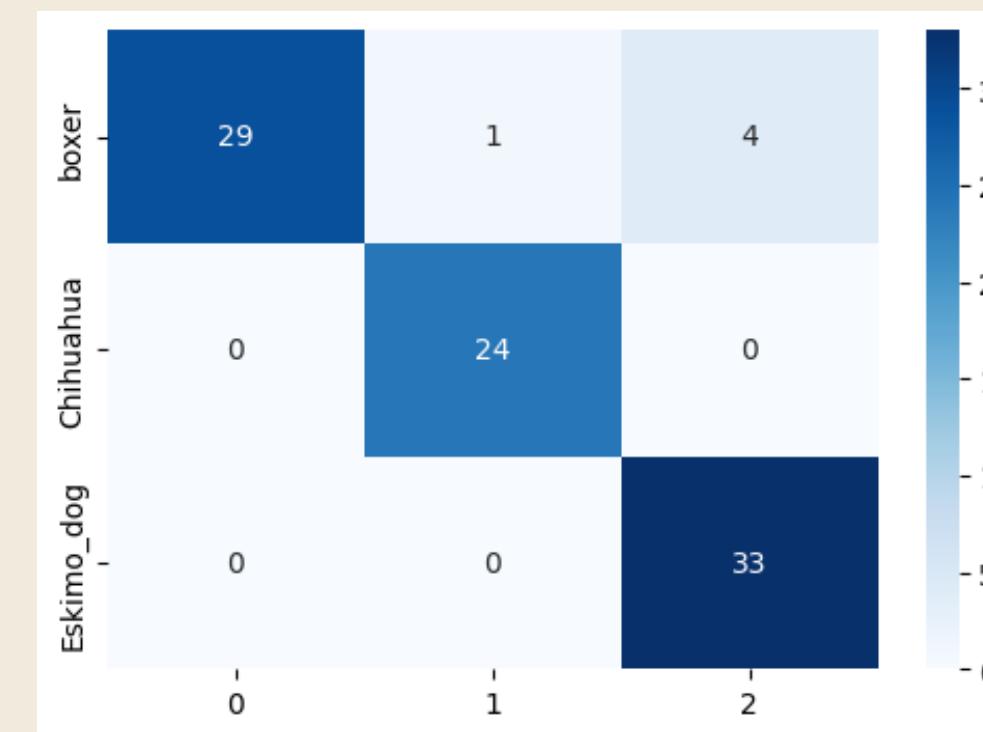
Validation Accuracy : 0.9444
Validation Loss : 0.1240
Test Accuracy : 0.9670
Test Loss : 0.0667



Avec data augmentation



Validation Accuracy : 0.9556
Validation Loss : 0.1641
Test Accuracy : 0.9670
Test Loss : 0.0859



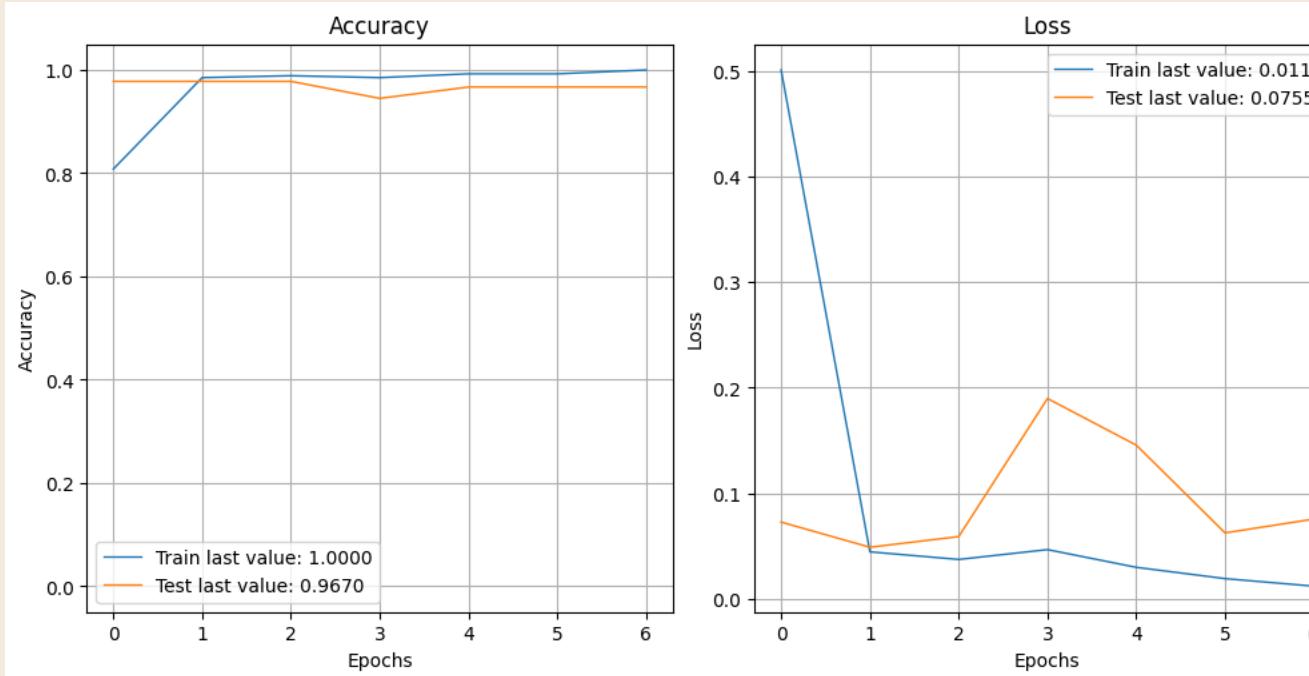


Transfer learning InceptionV3

Inception V3 optimise l'utilisation des ressources en réseau en utilisant des convolutions factorisées (décomposition en convolutions plus petites) et en introduisant des connexions d'augmentation de la dimension. Cela permet de capturer des caractéristiques plus riches tout en réduisant la complexité computationnelle.

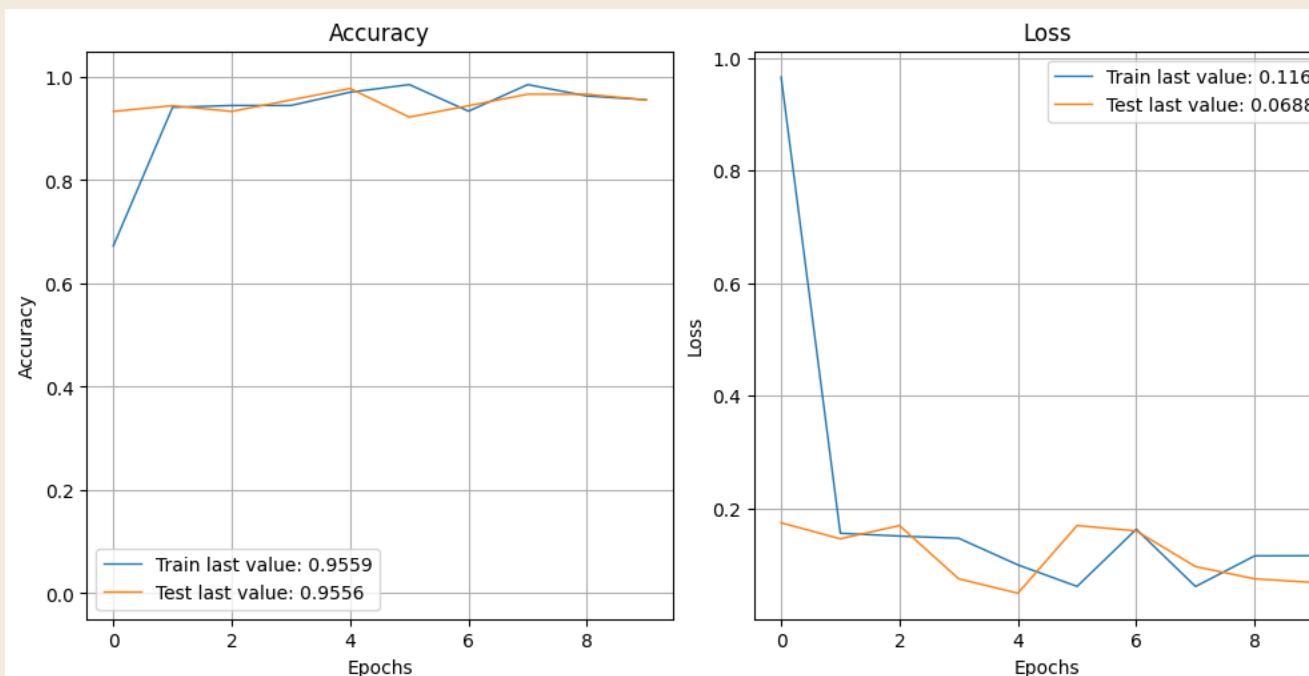


Sans augmentation

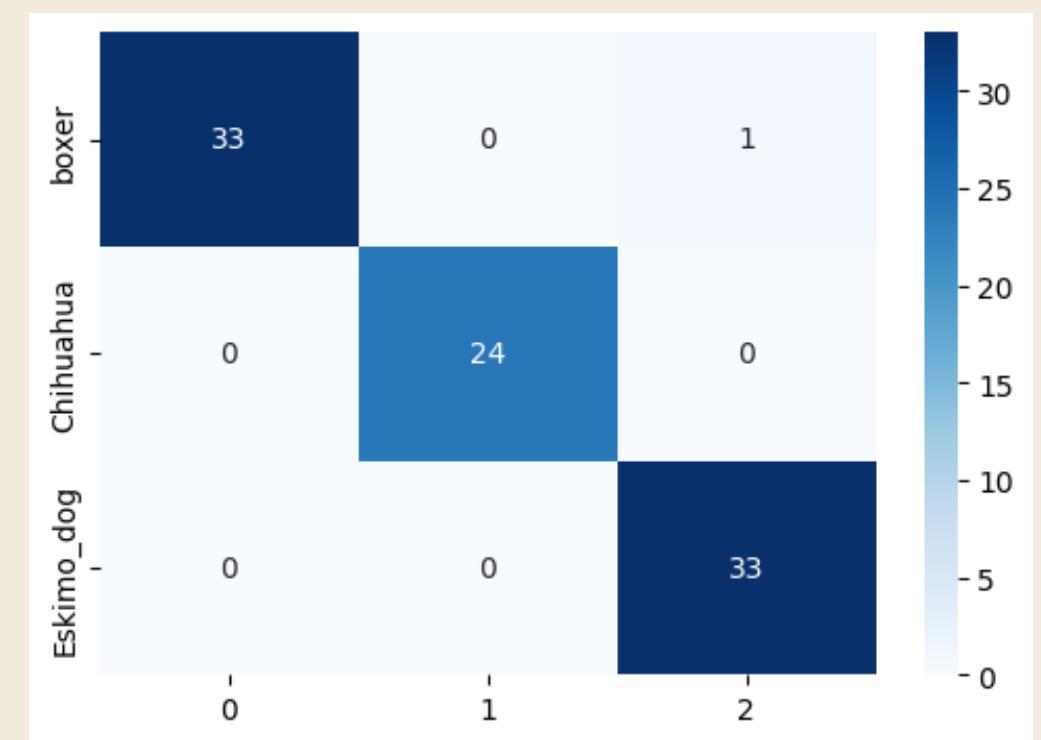
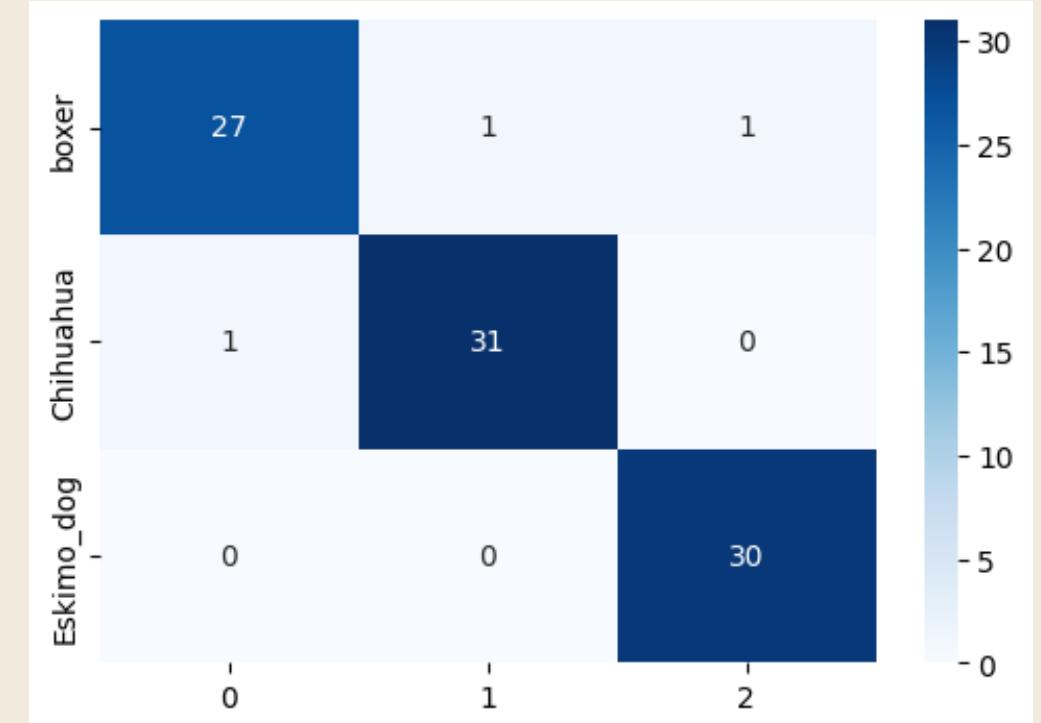


Validation Accuracy :
0.9670
Validation Loss :
0.0310
Test Accuracy : 0.9890
Test Loss : 0.0167

Avec data augmentation



Validation Accuracy :
0.9222
Validation Loss :
0.2185
Test Accuracy : 0.9890
Test Loss : 0.0343





Modeles sélectionnés

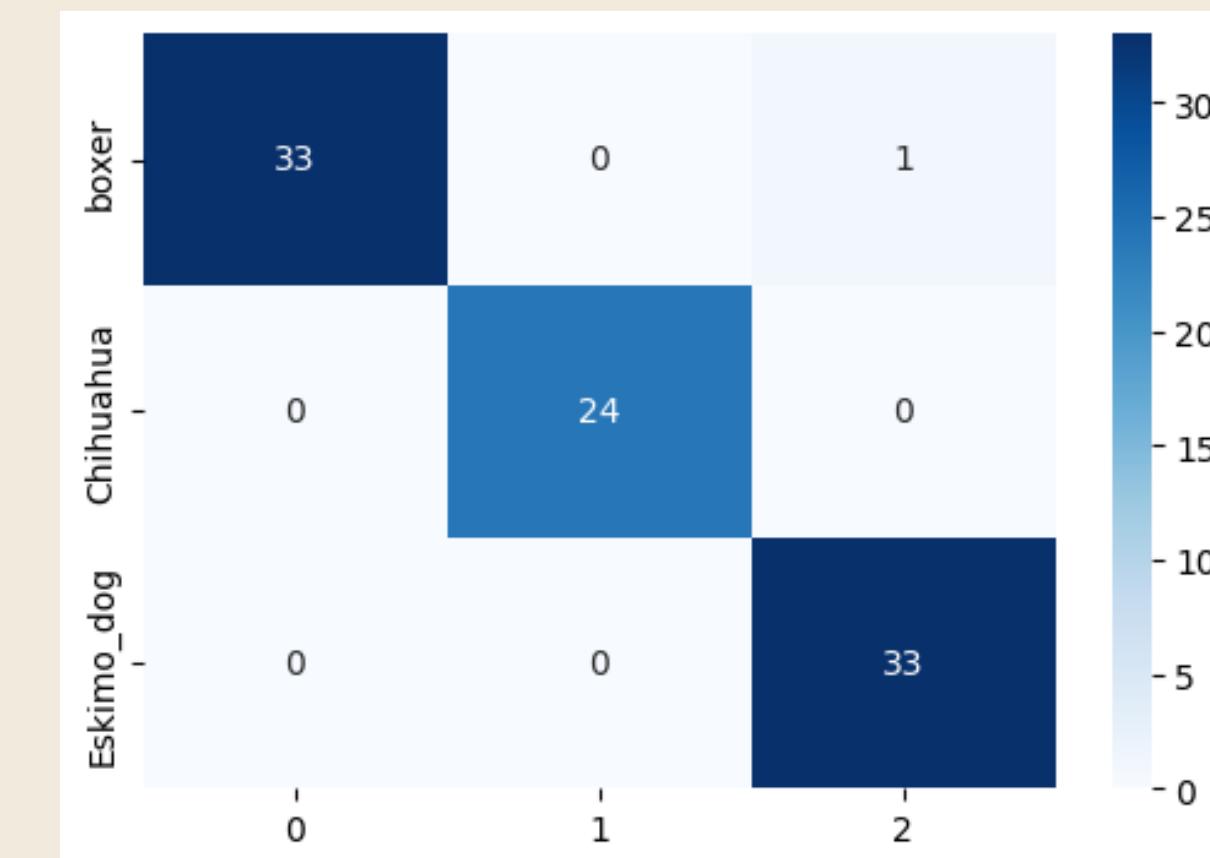
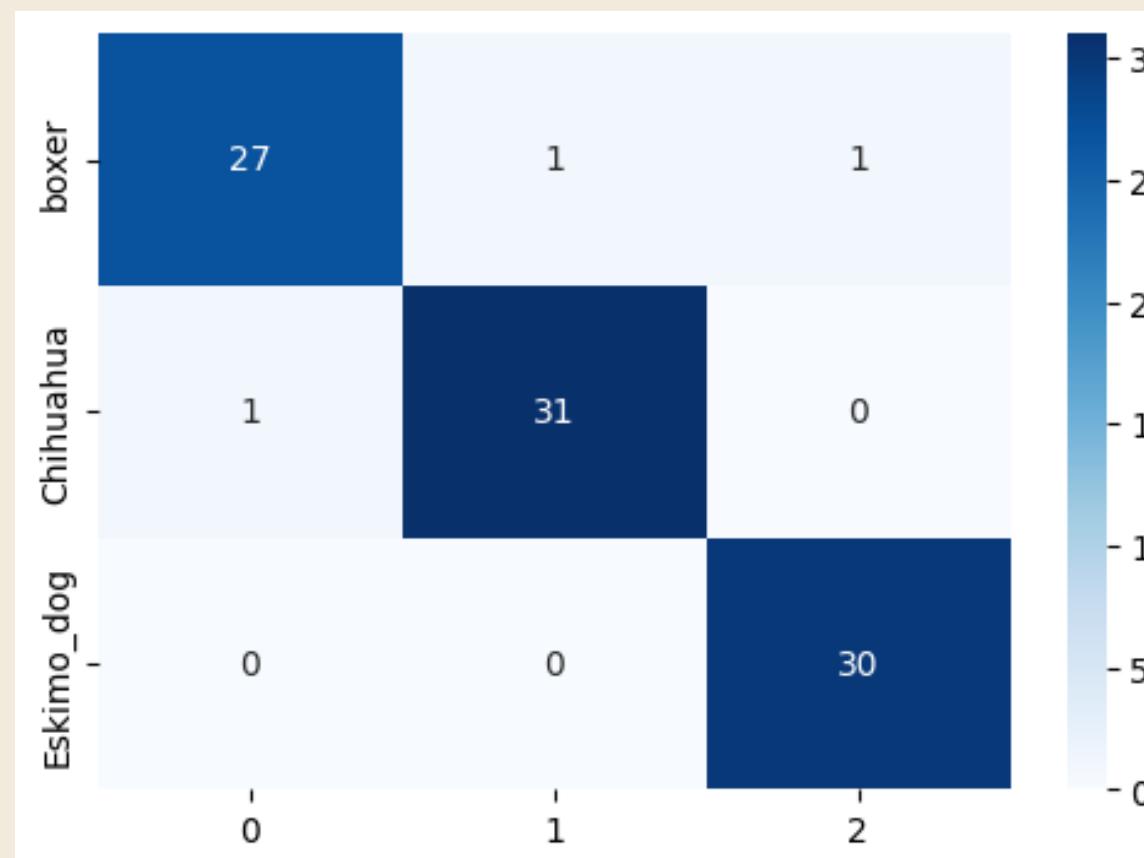
Inceptionv3 donne les meilleurs résultats, que ce soit avec ou sans data augmentation. Les résultats sont néanmoins légèrement meilleurs avec Data augmentation, ce sera donc le modèle choisi pour réaliser l'application.

Inceptionv3 sans data augmentation

	precision	recall	f1-score	support
0	0.96	0.93	0.95	29
1	0.97	0.97	0.97	32
2	0.97	1.00	0.98	30
accuracy			0.97	91
macro avg	0.97	0.97	0.97	91
weighted avg	0.97	0.97	0.97	91

Inceptionv3 avec data augmentation

	precision	recall	f1-score	support
0	1.00	0.97	0.99	34
1	1.00	1.00	1.00	24
2	0.97	1.00	0.99	33
accuracy				91
macro avg	0.99	0.99	0.99	91
weighted avg	0.99	0.99	0.99	91





Création d'une application en local

Pour faciliter l'observation des résultats du modèle, une application locale a été développée en combinant FastAPI pour le backend et Streamlit pour créer une interface utilisateur interactive.

The screenshot shows a browser window displaying the FastAPI documentation for a POST endpoint named 'make_prediction'. The URL is https://e5f5-146-148-48-188.ngrok-free.app/docs#/default/make_prediction_post. The page includes a 'Responses' section with a 'Curl' command and a 'Request URL' field. Below this is a 'Server response' table for status code 200, showing a JSON response body and response headers. At the bottom, there's a 'Responses' table for status code 200 with a 'Successful Response' entry.



 **FastAPI**

- **FastAPI pour le Backend :**

- Framework léger et performant pour créer des APIs en Python.
- Permet de gérer facilement les requêtes HTTP pour interagir avec le modèle de deep learning.
- Facilite l'intégration du modèle CNN pour effectuer des prédictions à partir de nouvelles images soumises via l'interface..



Rendu de l'application



•Streamlit pour l'Interface Utilisateur :

- Framework rapide pour créer des interfaces web interactives avec Python.
- Utilisé pour développer une interface utilisateur simple et réactive, permettant aux utilisateurs de télécharger des images et de visualiser les prédictions en temps réel.

A screenshot of a Streamlit application running in a browser window. The URL in the address bar is "localhost:8501". The main content area displays a brown and tan dog lying on a concrete surface. Below the image, the text "Image téléchargée." is visible. At the bottom, there is a red button labeled "Prédire", followed by the text "Prédition en cours..." and "Résultat de la prédition: chihuahua".

A screenshot of the same Streamlit application running in a browser window. The URL in the address bar is "localhost:8501". The main content area displays a white and grey dog with a black mask-like pattern. Below the image, the text "Image téléchargée." is visible. At the bottom, there is a red button labeled "Prédire", followed by the text "Prédition en cours..." and "Résultat de la prédition: eskimo dog".

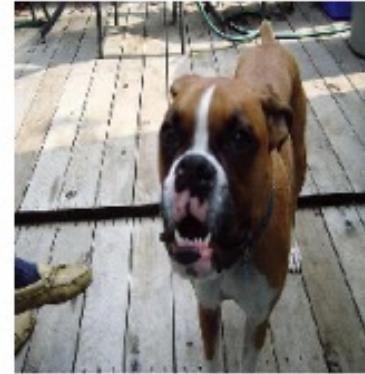


Predictions d'images provenant du set de test

True: Eskimo_dog, Pred: Eskimo_dog



True: Boxer, Pred: Boxer



True: Eskimo_dog, Pred: Eskimo_dog



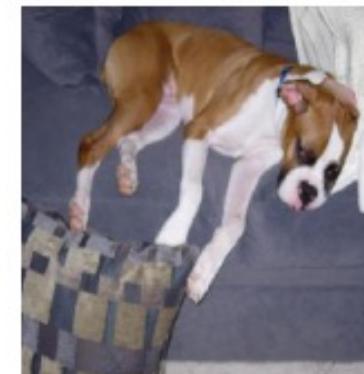
True: Boxer, Pred: Boxer



True: Boxer, Pred: Boxer



True: Boxer, Pred: Boxer



True: Chihuahua, Pred: Chihuahua



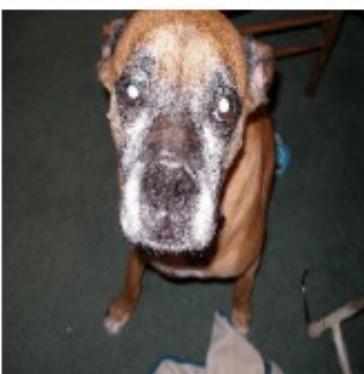
True: Chihuahua, Pred: Chihuahua



True: Boxer, Pred: Boxer



True: Boxer, Pred: Boxer



True: Eskimo_dog, Pred: Eskimo_dog



True: Eskimo_dog, Pred: Eskimo_dog





conclusion



Modèle CNN Personnalisé : Conception et optimisation d'un modèle CNN en utilisant une recherche avancée d'hyperparamètres pour maximiser la précision de classification.



Transfer Learning : Intégration du transfer learning pour améliorer les performances, en exploitant des modèles pré-entraînés sur de larges ensembles de données.

Essais Comparatifs : Tests approfondis avec et sans augmentation de données (Data Augmentation)

Principaux Résultats :

•**Performances du Modèle :** Amélioration significative grâce au transfer learning. La data augmentation n'améliore pas systématiquement les résultats pour tous les algorithmes testés.

Axes d'amélioration

-**Approfondissement du Transfer Learning :** Exploration de modèles pré-entraînés plus avancés ou spécialisés pour améliorer encore la précision.

