

Classez des images à l'aide de Deep Learning

L.khelliouf¹ Mentor: B.Beaufils²

^{1,2}OpenClassRooms

Projet 7, juin 21

Sommaire

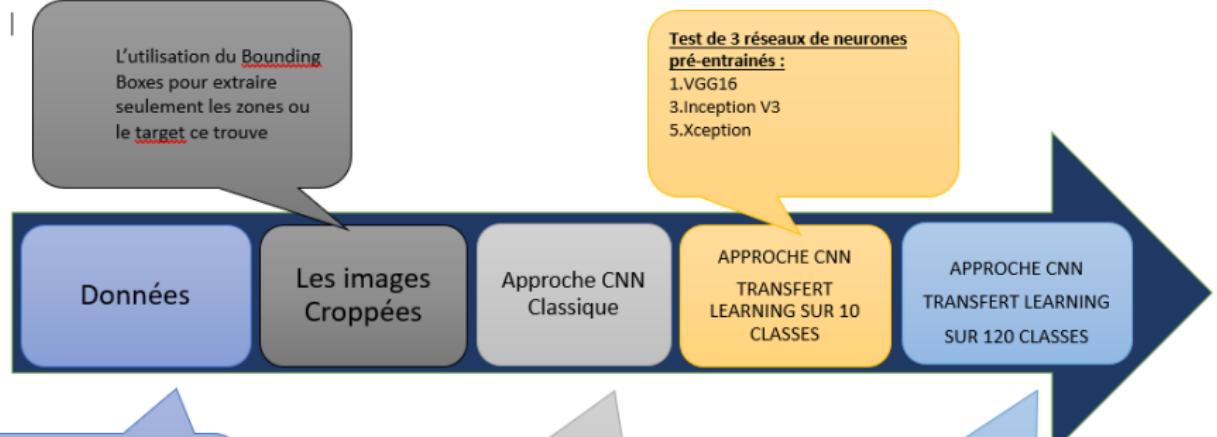
- Problème
- Introduction
- Récupération Des Données
- Approche CNN Classique
- Approche CNN Transfert Learning
- Evaluation
- Conclusion

Problématique

Une association de protection des animaux aimerait avoir un algorithme capable de classer les images en fonction de la race du chien présent sur l'image.



Pré-Processing



Classification d'image de races de chiens :

Standford Dogs Dataset :

20580 pour 120 races

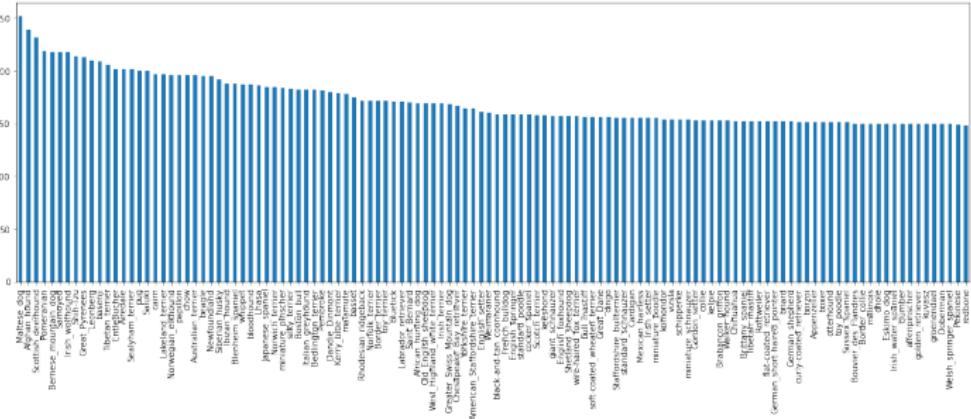
- Réseaux de neurones convolutionnels
- 3 modèles de CNN à partir de zéro (fromscratch)
- Data Augmentation

Test de le réseaux de neurones pré-entraînés :

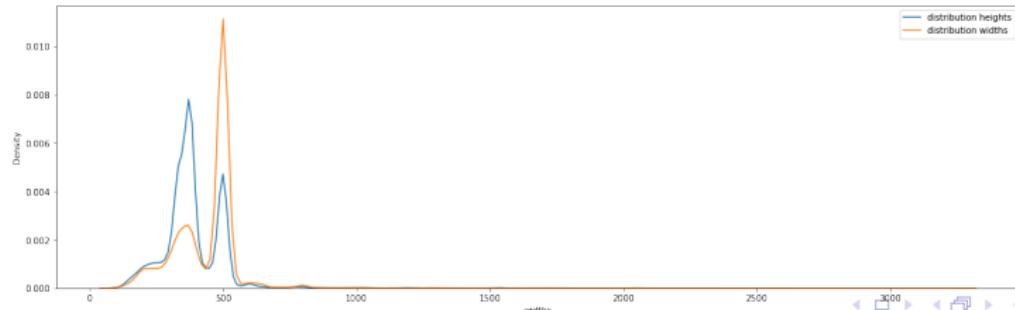
- Inception V3

Données

Nombre d'images par race



La distribution de height et width des images



Les images croppées

Images avant le cropping

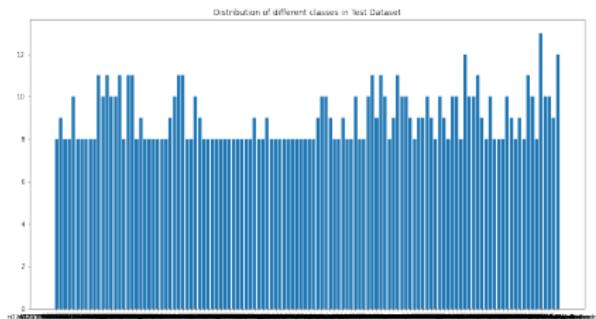
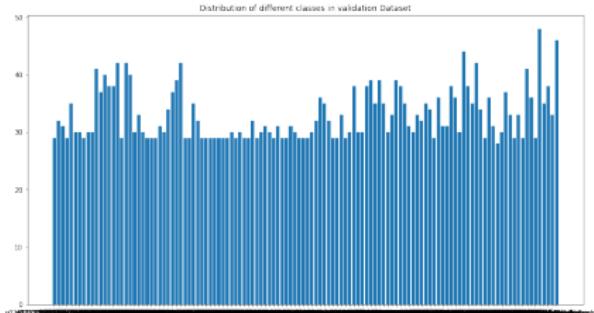
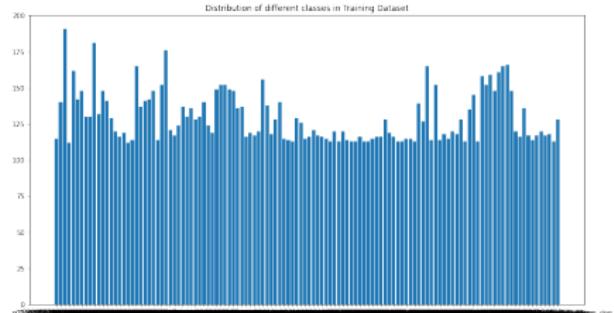


Images après le cropping



Distribution des datasets

Séparation Train / Validation / Test : 80 % = 15551 / 15% = 3952 / 5% = 1077



ImageDataGenerator

ImageDataGenerator est une classe de Keras connue qui possède 2 fonctionnalités principales :

Data Augmentation : il s'agit d'une technique qui permet de réduire l'overfitting en augmentant le nombre d'observations dans les données d'entraînement. Loading image data from disk : cela permet à l'utilisateur de s'entraîner sur plus que celui qui est capable de tenir en mémoire.

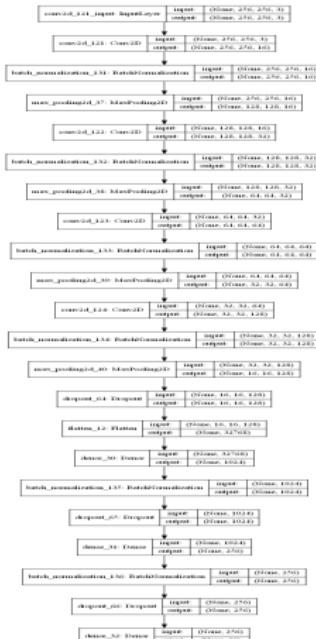
- Rescale 1/255.0
- Rotation de 30
- Zoom de 10
- brightness range [0.8, 1.5]
- horizontal_flip=True,
- width_shift_range=0.1,height_shift_range=0.1,



Première Approche CNN Classique

CNN From Scratch , construction d'un réseau neuronal depuis zéro :

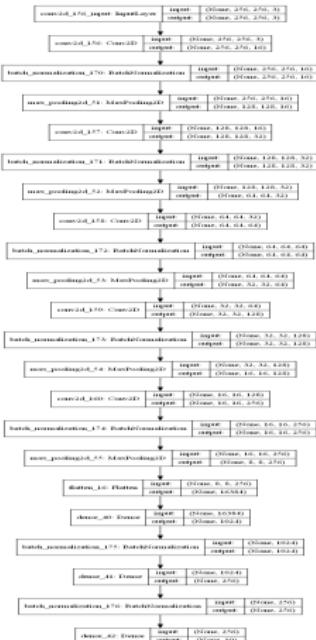
- Redimensionnement en 256*256
- Utilisation des images en couleurs
- Séparation Train / Validation / Test :80% / 15 % / 5 %
- Utilisation du Dropout de 25%
- Utilisation de l'optimizer RMSProp



Deuxième Approche CNN Classique

CNN From Scratch , construction d'un réseau neuronal depuis zéro :

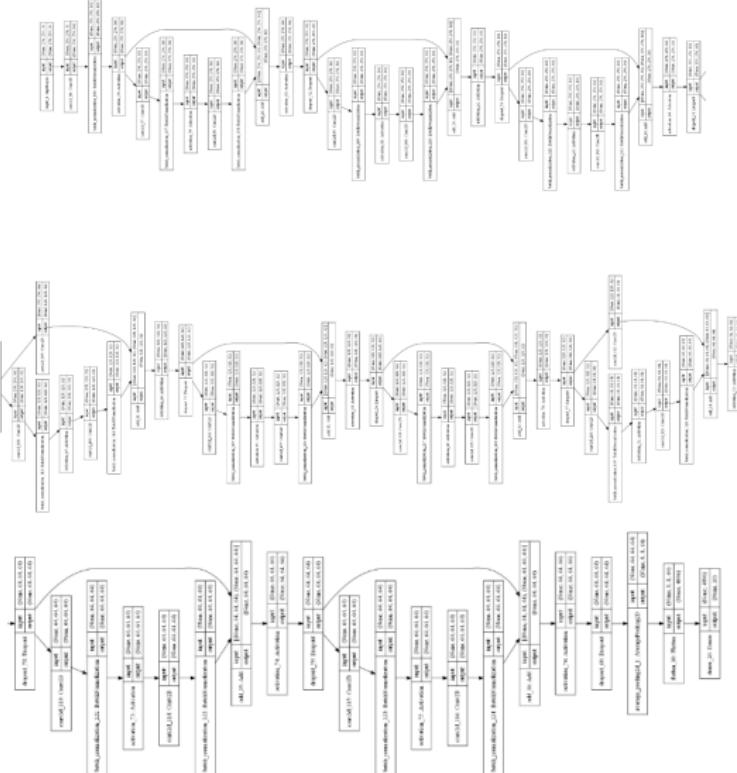
- Redimensionnement en 256*256
- Utilisation des images en couleurs
- Séparation Train / Validation / Test :80% / 15 % / 5 %
- Utilisation de l'optimizer ADAM



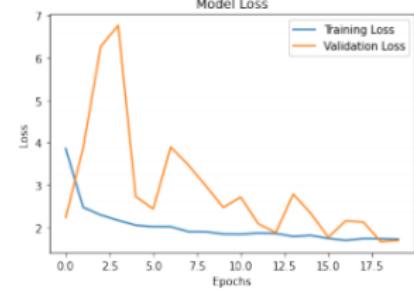
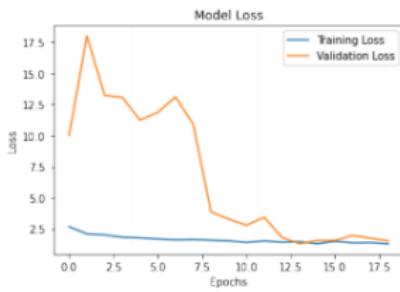
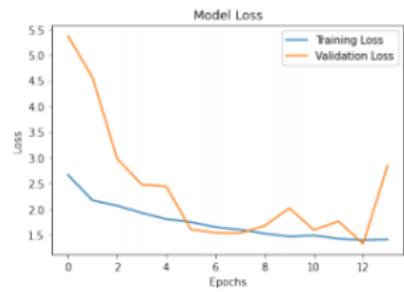
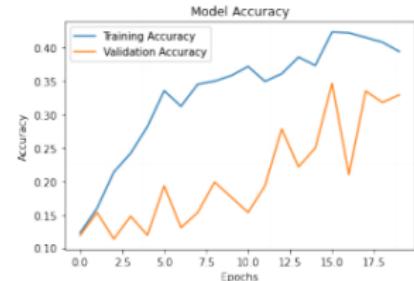
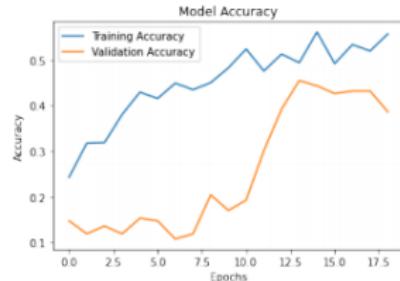
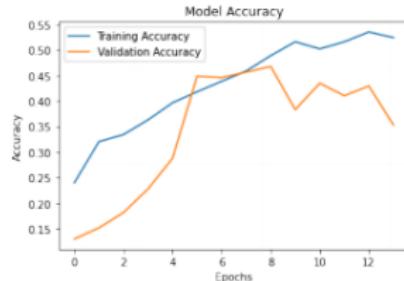
Troisième Approche Resnet From Scratch

Resnet From Scratch , construction d'un réseau neuronal depuis zéro :

- Redimensionnement en 256*256
- Utilisation des images en couleurs
- Séparation Train / Validation / Test :80% / 15 % / 5 %
- Utilisation de l'optimizer ADAM
- Nombre d'epoch =20



Résultats



Model1: Test Accuracy=47%

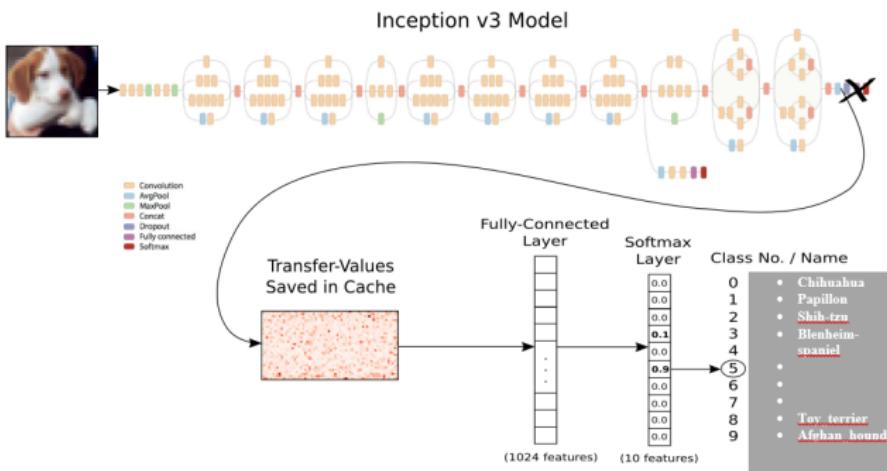
Model2: Test Accuracy=54%

Model3: Test Acc= 37%

Approche CNN Transfer Learning

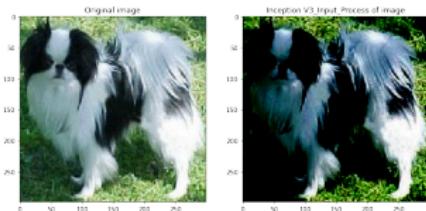
Le transfer learning

Comme son nom l'indique, est une approche permettant de transférer le savoir d'un modèle, à un autre. En l'occurrence, transférer les poids d'un modèle pré entraîné à une tâche précise, pour les associer à un autre modèle.



Approche CNN Transfer Learning

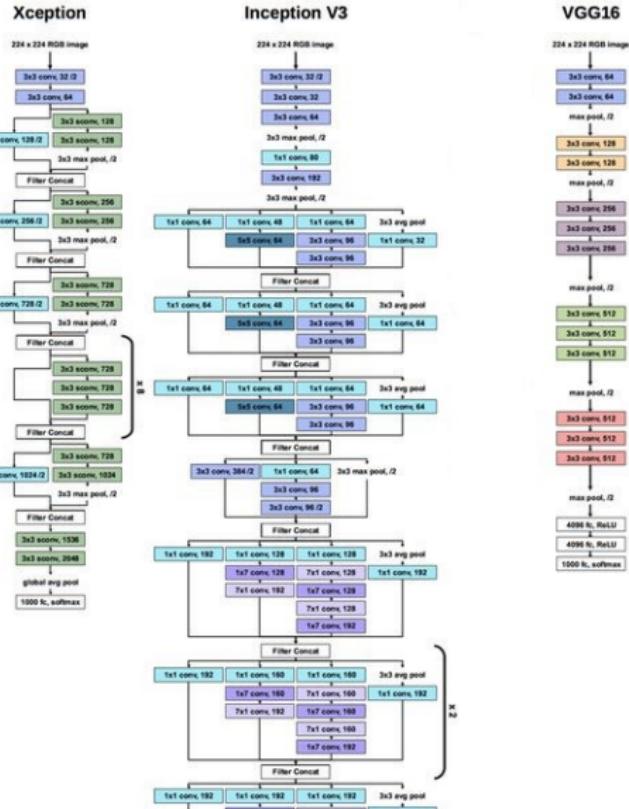
L'effet du process_input du modèle Inception V3:



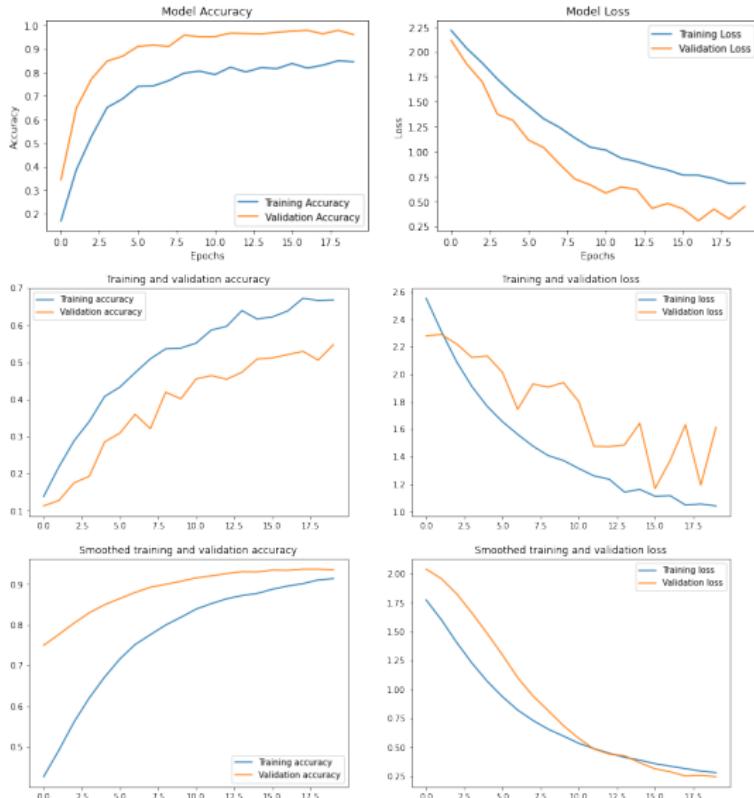
L'effet de notre fonction de prétraitement du modèle Inception V3:



Approche CNN Transfer Learning



Approche CNN Transfer Learning



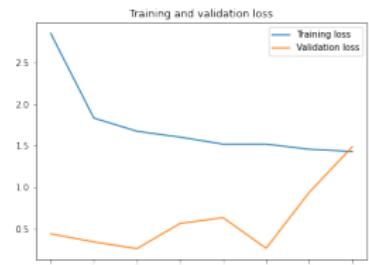
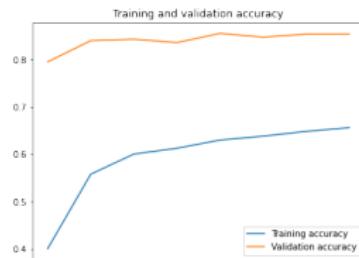
Inception: test accuracy= 91%

VGG16: test accuracy = 67%

Xception: test accuracy= 91%

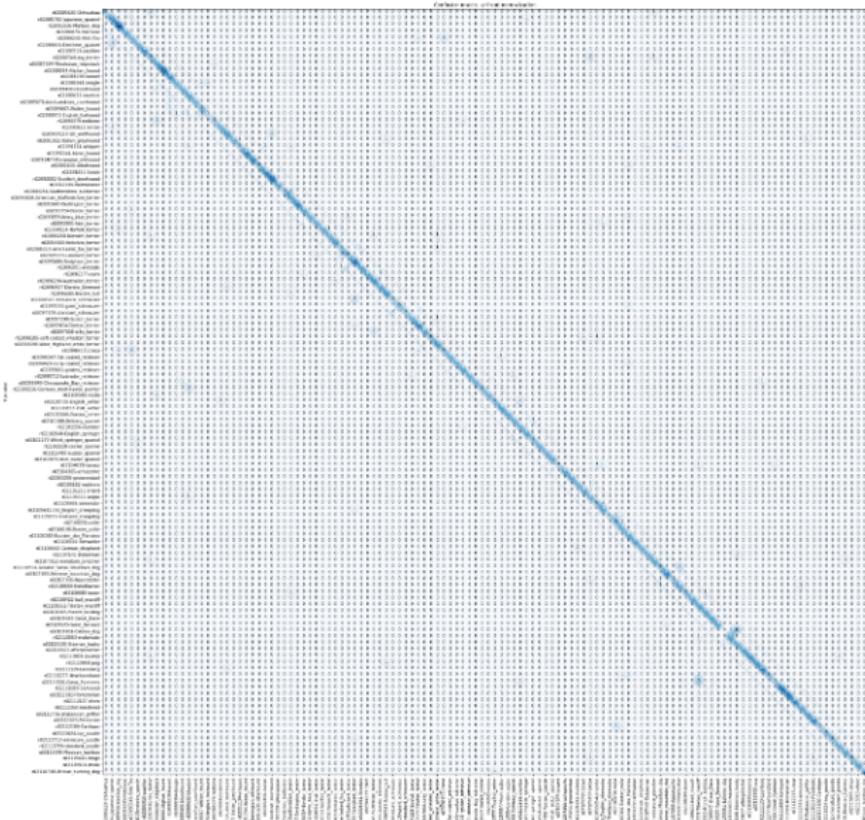
Transfer Learning sur 120 Classes

- Modèle choisis: InceptionV3
- Redimensionnement en 299*299
- Batch_size=20
- Utilisation des images en couleurs
- Rotation de 30%
- Zoom sz 10%
- Séparation Train / Validation / Test :80% / 15 % / 5 %
- Utilisation de l'optimizer RMSprop avec un learning rate = $1e-4$



Test accuracy= 83%

Transfer Learning sur 120 Classes



Erreur!!

True label "Class Eskimo_dog"



Predicted label "Class
Siberian_husky"

Quelques example de la "Class
Siberian_husky"



Conclusion

- LES RESEAUX CONVOLUTIONNELS SONT PLUS EFFICACES QUE LES METHODES CLASSIQUES.
- DATA AUGMENTATION PERMET D'EVITER L'OVER FITTING SI LE NOMBRE DE DONNEES EST FAIBLE.
- FAIRE VARIER LE NOMBRE ET LA STRUCTURE DES COUCHES (DROPOUT, DENSE,etc ...) SUPPLEMENTAIRES EN SORTIE DES MODELES PRÉ ENTRAÎNÉS.
- ENTRAINER LA DATASET SUR D AUTRES MODELES EXESTANT DU TRANSFER LEARNING. AVOIR PLUS DE DONNÉES EN ENTRÉE.

Merci pour votre attention