Protection des animaux

Classer des images à l'aide d'algorithmes de Deep Learning

Presentation

Sofiane Mouhab
Fevrier 2022

Sommaire

1 - Généralités

1.1 - Problematique

Vous êtes bénévole pour l'association de protection des animaux de votre quartier. C'est d'ailleurs ainsi que vous avez trouvé votre compagnon idéal, Snooky. Vous vous demandez donc ce que vous pouvez faire en retour pour aider l'association.

Vous apprenez, en discutant avec un bénévole, que leur base de données de pensionnaires commence à s'agrandir et qu'ils n'ont pas toujours le temps de référencer les images des animaux qu'ils ont accumulées depuis plusieurs années. **Ils aimeraient donc obtenir un algorithme capable de classer les images en fonction de la race du chien présent sur l'image.**

1 – Généralités

1.2 – Objectifs

L'association vous demande de réaliser un algorithme de détection de la race du chien sur une photo, afin d'accélérer leur travail d'indexation.

- Préprocesser des images avec des techniques spécifiques
- Réaliser de la data augmentation
- Réaliser votre propre réseau CNN
- Utiliser le transfer learning
- Évaluer les performances d'un modèle de Deep Learning

1 - Généralités

1.3 - Conditions de mise en oeuvre

Pour pouvoir sereinement réaliser ses objectifs, il nous faut donc diverses informations qui pourrait se trouver dans notre base de données.

À nous donc, d'examiner celle-ci, de déterminer à quel point les informations sont viables, ou perfectible.

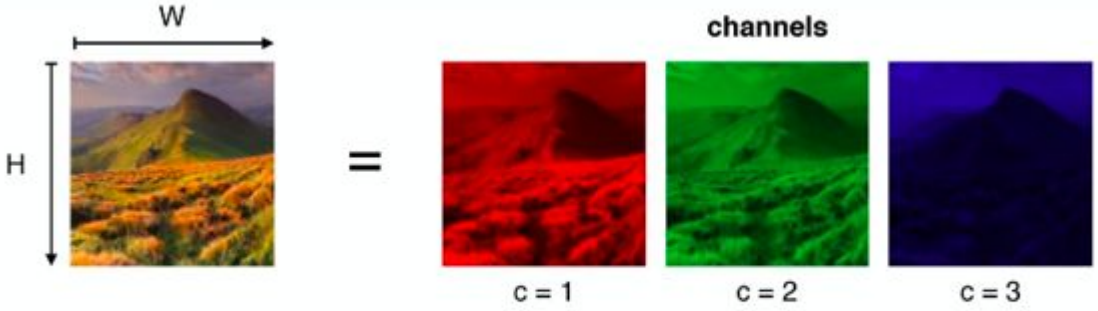
Il y a donc 3 grandes interrogations :

- A-t-on assez de données ?
- Peut-on les traiter et les entraîner ?
- Les résultats obtenus sont-il cohérent ?

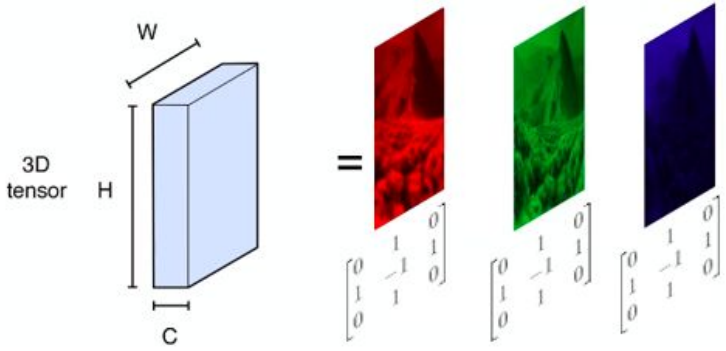
Passons de suite à ce travail, en commençant par rapidement prendre connaissance des données en présence...

2 - Introduction

2.1 - Image numerique



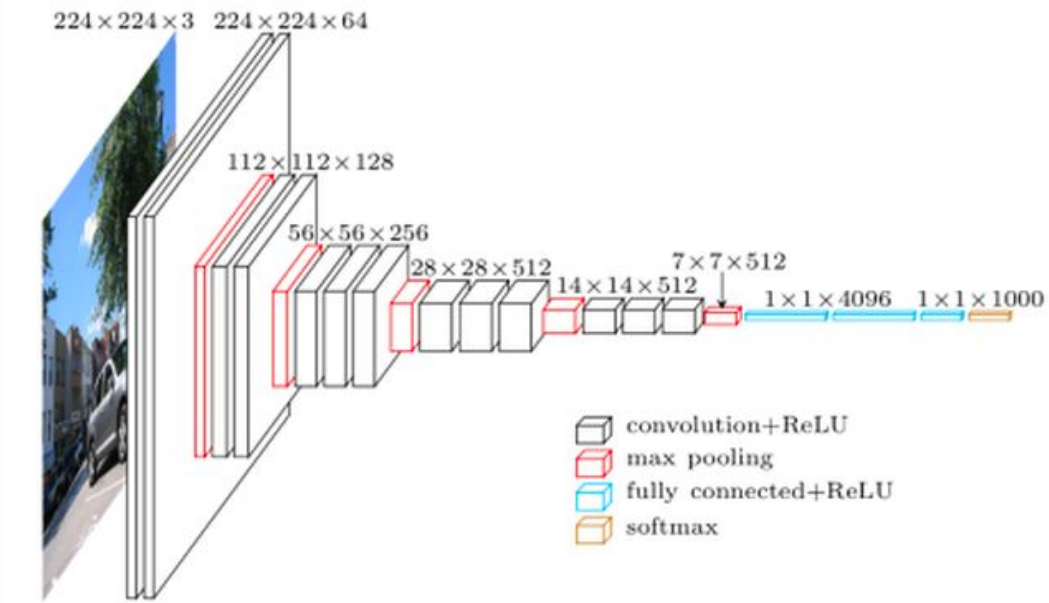
Les 3 canaux RGB constituant une image



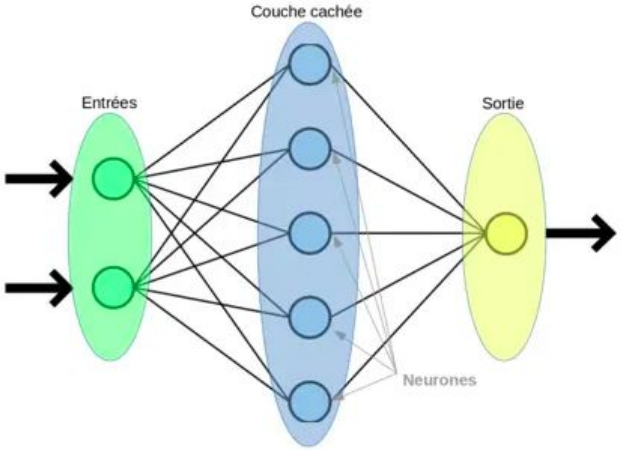
Empilement de chaque matrice de nos canaux qui forment un tenseur d'ordre 3

2 - Introduction

2.2 - CNN



Représentation 3D de l'architecture de VGG-16



Représentation d'un perceptron multicouche

2 - Introduction

2.3 - Etapes

Étape 1 : Entrainement du modèle
Étape 2 : Importation des images à prédire
Étape 3 : Prétraitement des images
Étape 4 : Entraîner du modèle
Étape 5 : Prédiction du modèle
Étape 6 : Evaluation du modèle

3 - Les données

3.1 - Présentation

L'ensemble de données Stanford Dogs contient des images de 120 races de chiens du monde entier. Cet ensemble de données a été construit à l'aide d'images et d'annotations d'ImageNet pour la tâche de catégorisation fine des images. Contenu de cet ensemble de données :

Nombre de catégories : 120

Nombre de photos : 20580

3 - Les données

3.2 - Nos images

- Maltese_dog
- Pekinese
- Shih-Tzu
- Blenheim_spaniel
- papillon
- toy_terrier
- Rhodesian_ridgeback
- Afghan_hound
- beagle
- bloodhound
- bluetick
- black-and-tan_coonhound
- Walker_hound
- English_foxhound
- redbone
- borzoi
- Irish_wolfhound
- Italian_greyhound
- whippet
- Ibizan_hound
- Norwegian_elkhound

...

3 - Les données

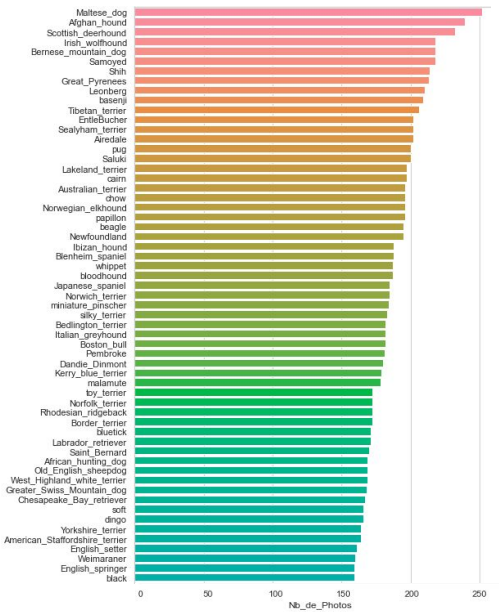
3.3 - En détails

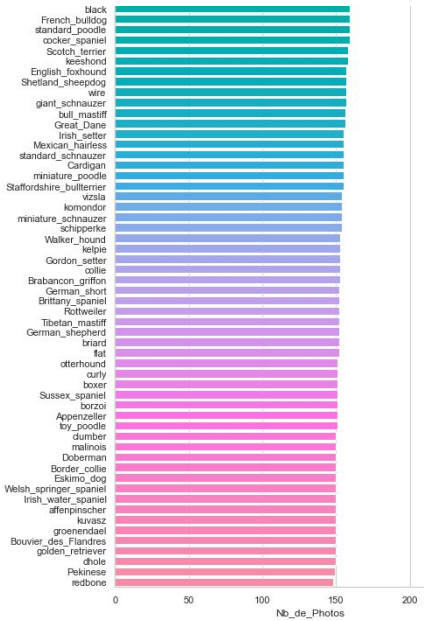
Weights

count	19507.000000
mean	385.506229
std	125.700438
min	100.000000
25%	333.000000
50%	375.000000
75%	453.000000
max	2562.000000

Heights

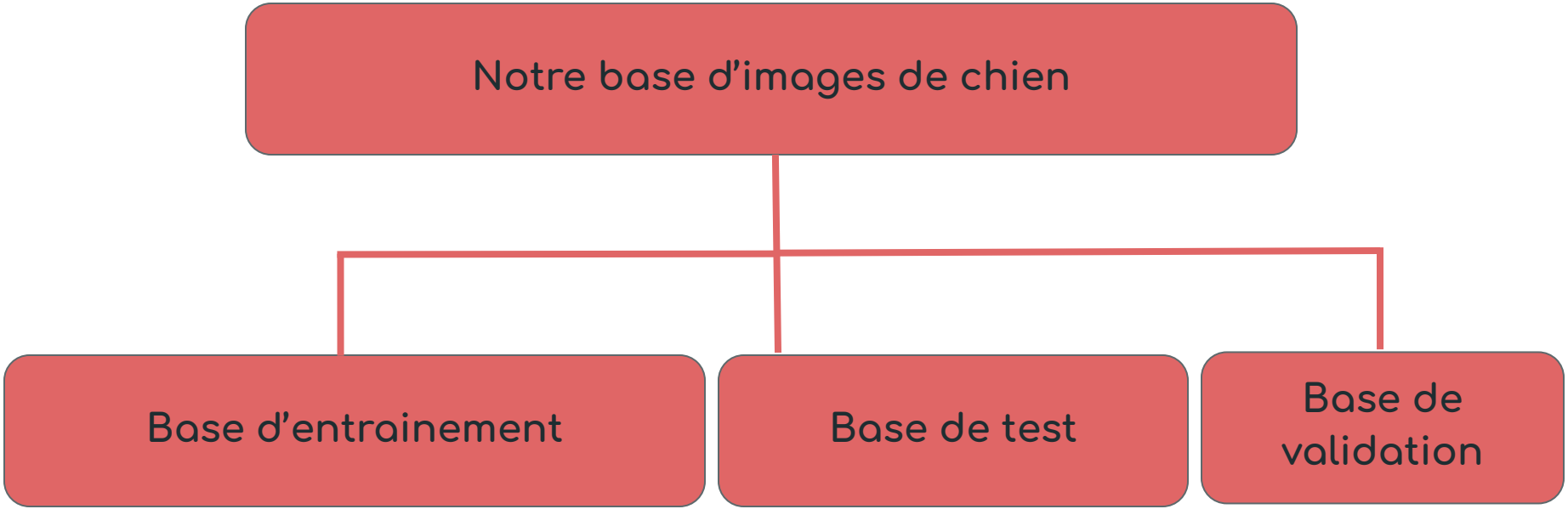
count	19507.000000
mean	441.509304
std	143.319610
min	97.000000
25%	360.000000
50%	500.000000
75%	500.000000
max	3264.000000





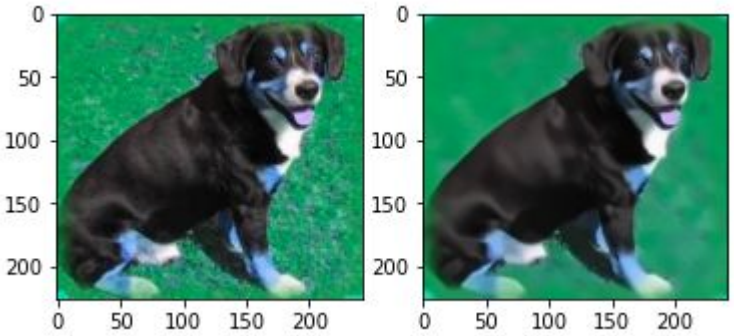
3 - Les données

3.4 - Préparation

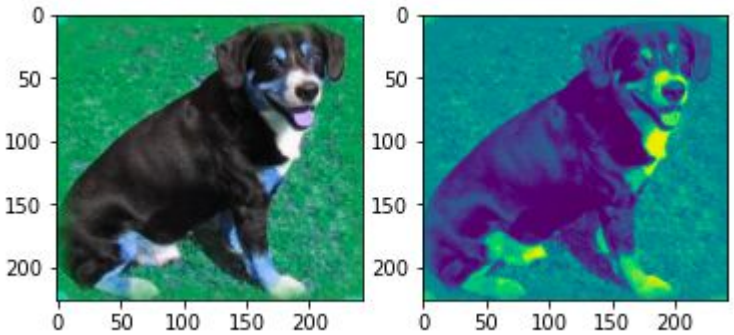


3 - Les données

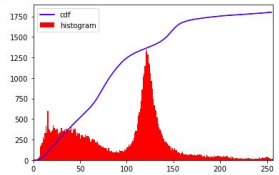
3.5 - Exemples de prétraitements



Sh1. Denoising



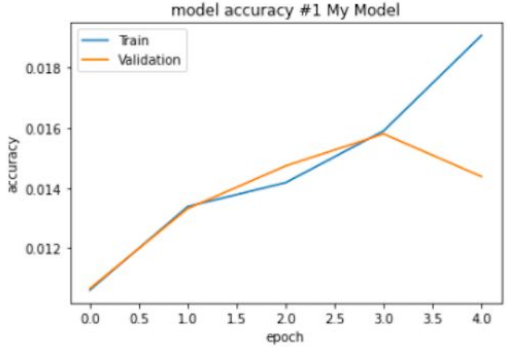
Sh2. Histogram Equalization

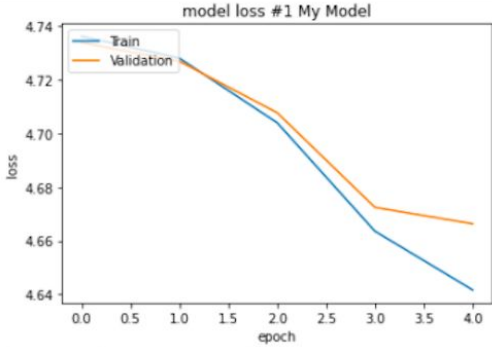


4 - Les resultats

4.1 - Mon Modele

4.1.1 - Mon Modele de base



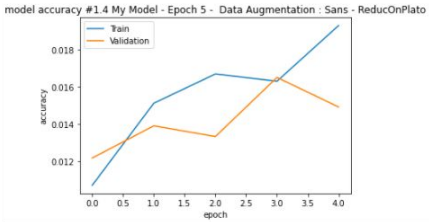


Mon Model - Accuracy : 2%

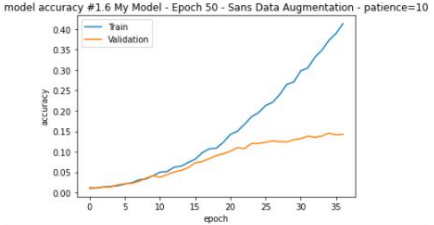
4 - Les resultats / 4.1.2 - Mon Modele avec Amelioration

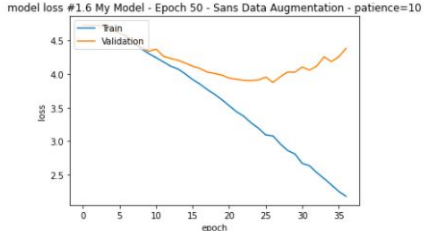
On effectue à ce stade différentes techniques pour améliorer notre modèle :

- Modification du **Batch Size** (64, 128, 256)
- Modification du **CallBack** (Early Stopping avec patience : 5 ,10 , LearningRateSheduler)
- Modification des **Optimizers** (Adam, ReduceLROnPlateau, Sdg)
- Modification du **Nombre d'époque** (5, 20, 50)

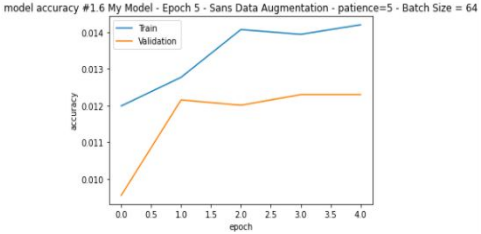


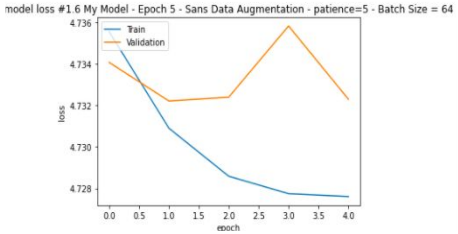
Test #1 - Accuracy : 1%



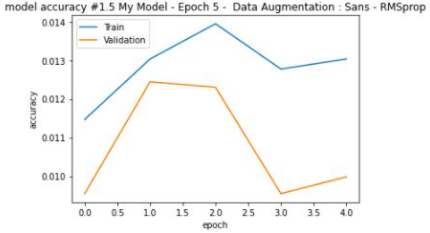


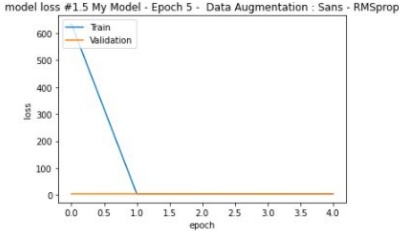
Test #2 - Accuracy : 12%





Test #3 - Accuracy : 1%

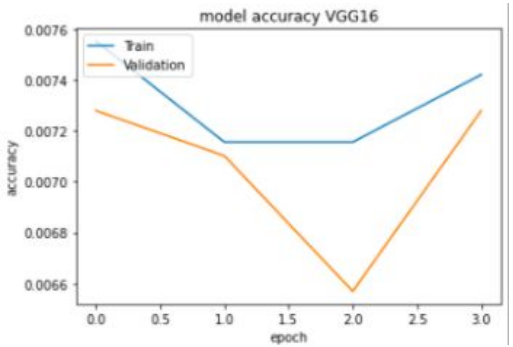


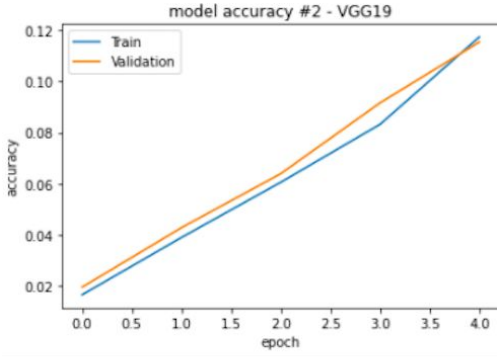


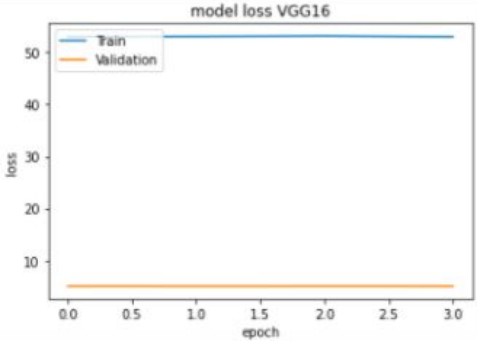
Test #4 - Accuracy : 1%

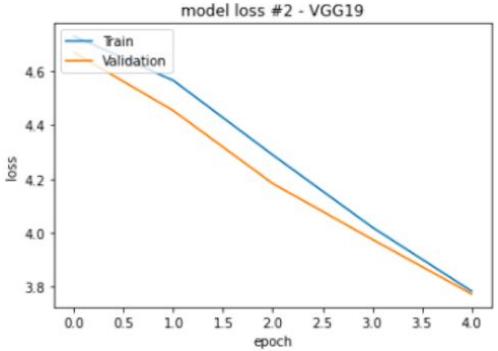
Sh13. Differents exemples de tests d'amélioration

4 - Les resultats / 4.2 - Transfer Learning /4.2.1 - Partie 1







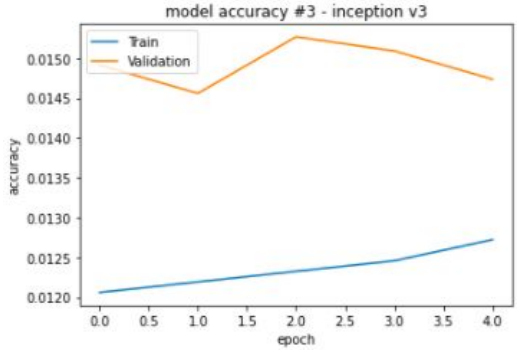


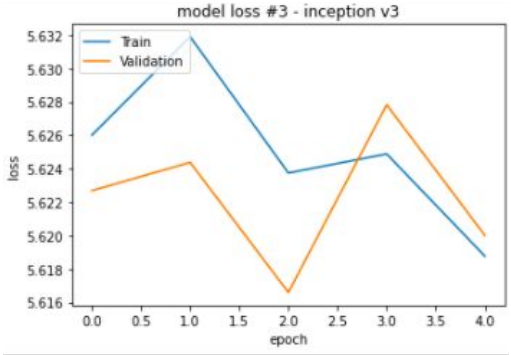
VGG16 - Accuracy : 1%

VGG19 - Accuracy : 12%

4 - Les resultats / 4.2.2 - Partie 2

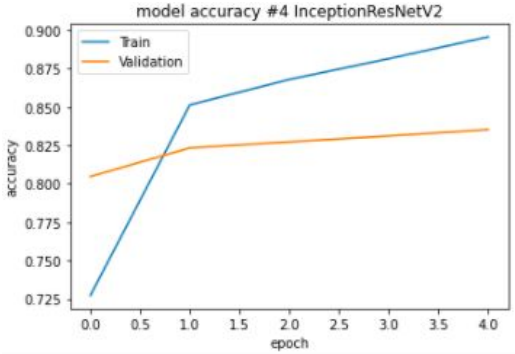
Inception V3

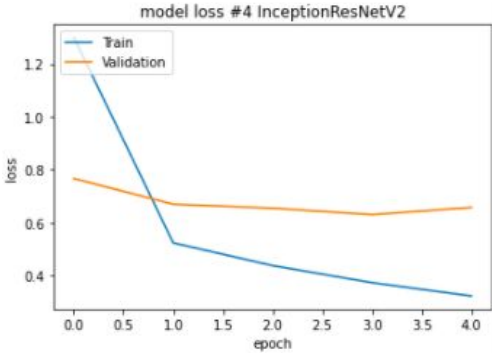




Inception V3 - Accuracy : 3%

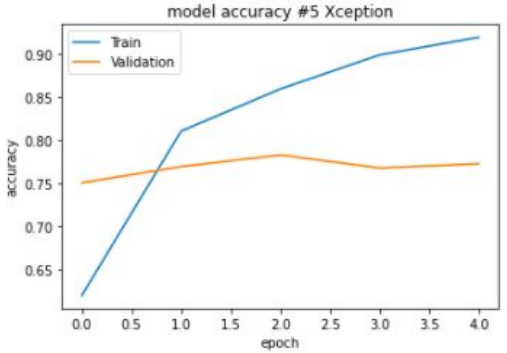
InceptionResNetV2

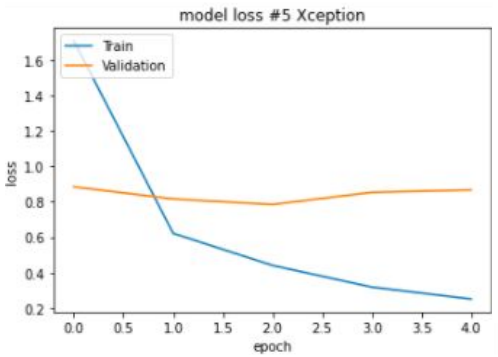




InceptionResNetV2 - Accuracy : 82%

Xception

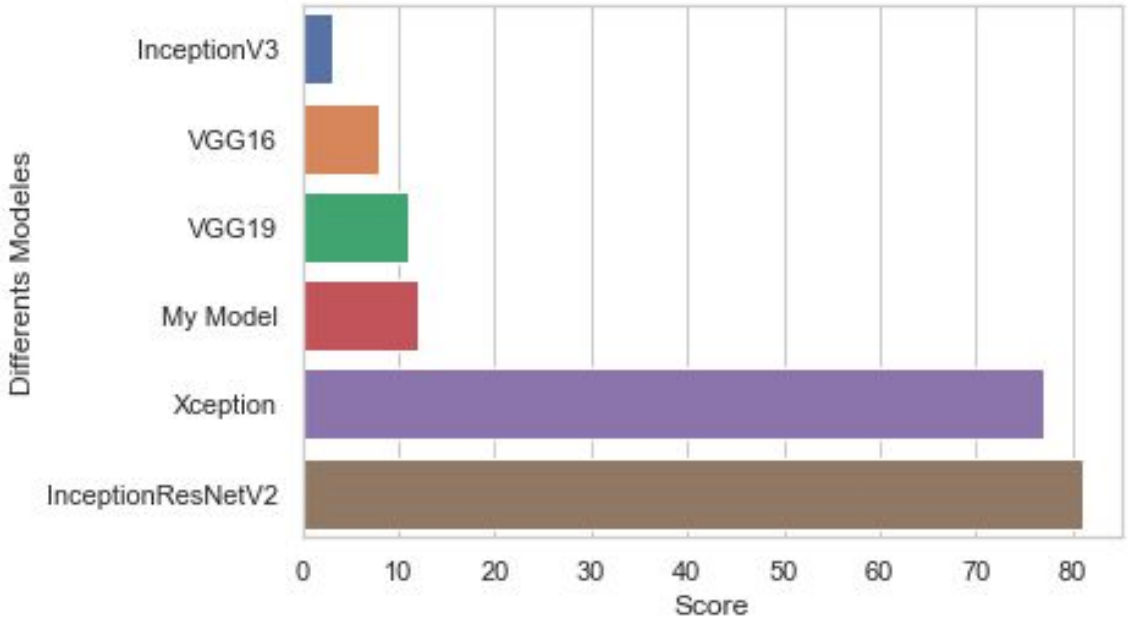




Xception - Accuracy : 77%

4 - Les resultats

4.3 - Comparatif & meilleur modèle

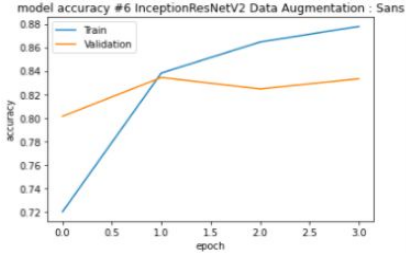
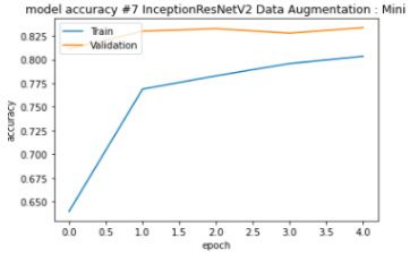
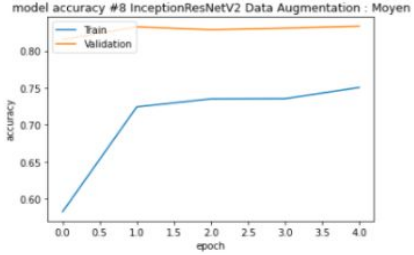
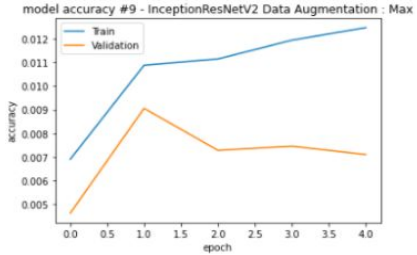
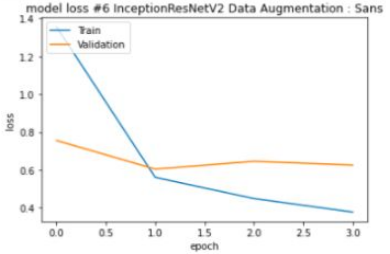
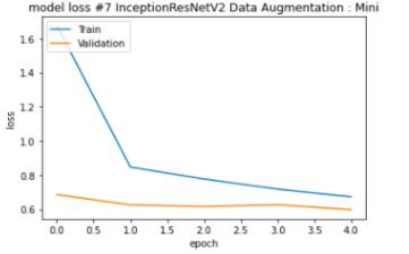
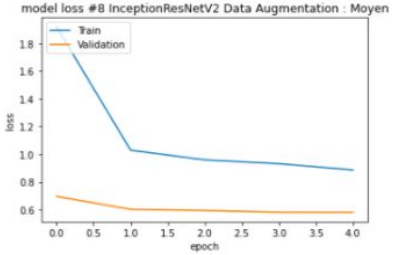
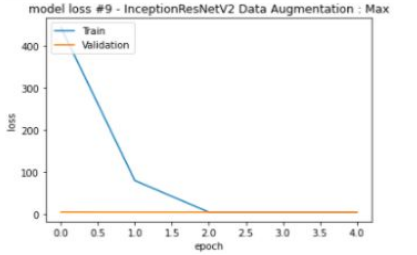


Meilleur Modèle intermédiaire:

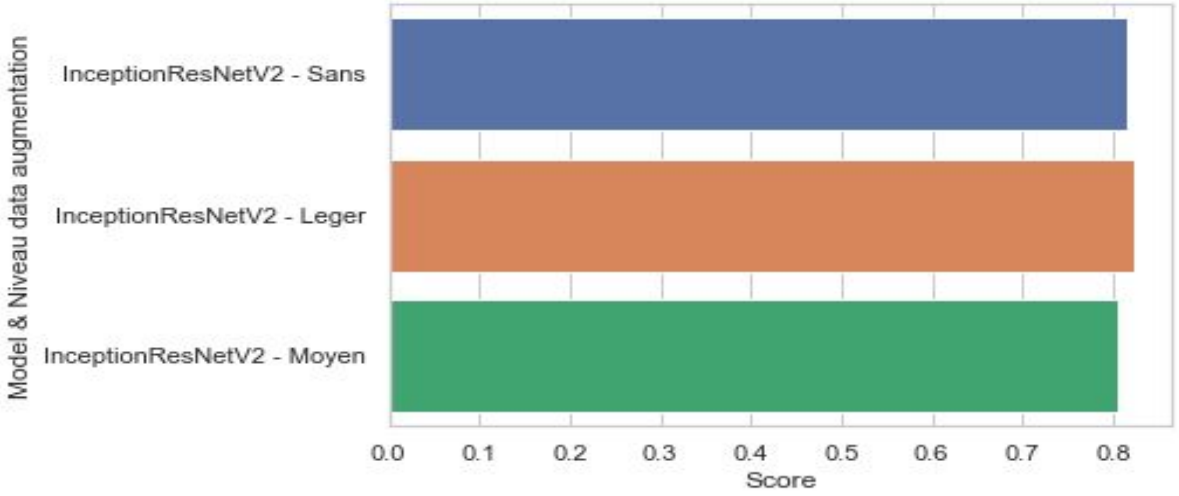
InceptionResNetV2

4 - Les resultats / 4.4 - Amelioration & Data Augmentation

On effectue à ce stade différentes techniques pour améliorer notre modèle (cf. page 15)

"Sans"	"Leger"	"Moyen"	"Haut"
			
			
Accuracy : 82 %	**Accuracy : 82.5%**	**Accuracy : 81%**	**Accuracy : 1%**

4 - Les resultats / 4.5 - Le meilleur modèle



Meilleur Modèle:

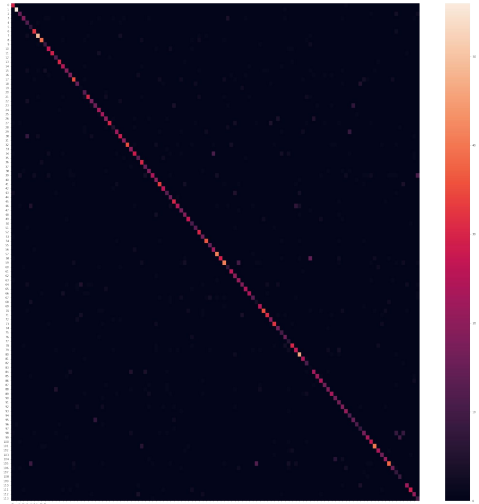
InceptionResNetV2 -
Avec une légère Data Augmentation

Settings definitif:

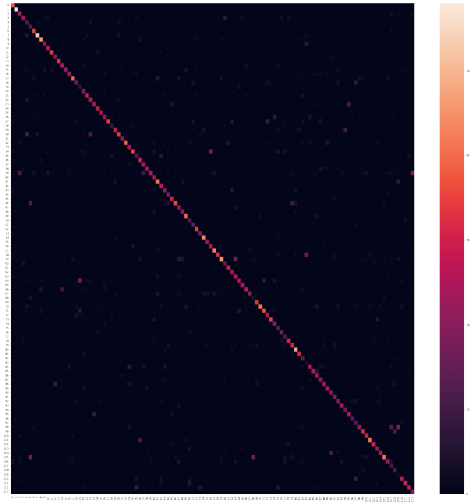
Transfer Learning = InceptionResNetV2
Callbacks = tf.keras.callbacks.EarlyStopping(patience=5)
Reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5, min_lr=0.001)
Nombre d'Epoque = 20

5 - Scoring

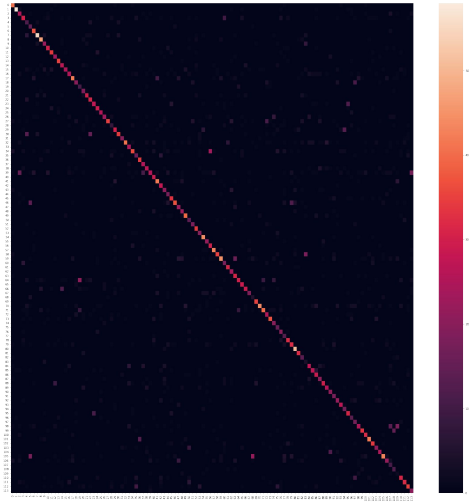
```
--- THRESHOLD :  0.99 ---

df Before :  (5853, 4)
df After :  (3227, 4)

Acc Général :  0.7551905794855903
```



```
--- THRESHOLD :  0.75 ---

df Before :  (5853, 4)
df After :  (4699, 4)

Acc Général :  0.6277931474781868
```



```
--- THRESHOLD :  0.5 ---

df Before :  (5853, 4)
df After :  (5392, 4)

Acc Général :  0.5816023738872403
```



6 - Conclusion

En conclusion, on peut constater que les données ne sont pas parfaites mais néanmoins opérationnelle à la réalisation de ce travail. Nous sommes donc en mesure de les traiter et de les entraîner. Les résultats obtenus sont cohérents.

On répond donc au cahier des charges sur ces points

- Processing de d'image
- Data augmentation via le Data image generator
- Création d'un réseau CNN personnel, mais peu performant
- Utilisation d'un transfer learning

Ce dernier nous permet d'obtenir de très bons résultats.

Sur ses bases et avec quelques réserves nous sommes donc en mesure d'obtenir un algorithme capable de classer les images en fonction de la race de chien.

7 - Perspective

- Tester d'autres modèles pour le transfert learning
- Il faudrait donc souscrire ou investir dans une puissance de calcul
- Améliorer la qualité des photos
- Améliorer la sélection des photos via un programme par exemple
- Décider ensemble d'un seuil de reconnaissance pour le modèle actuel

Annexe

Exemple de Data Augmentation: