

Projet : Classez des images à l'aide d'algorithme de Deep Learning

Antoine Maby - 27/10/2021

Problématique

Association de protection des animaux

Ils n'ont pas le temps de référencer les images des animaux

Faire un algorithme capable de classer les images en fonction de la race

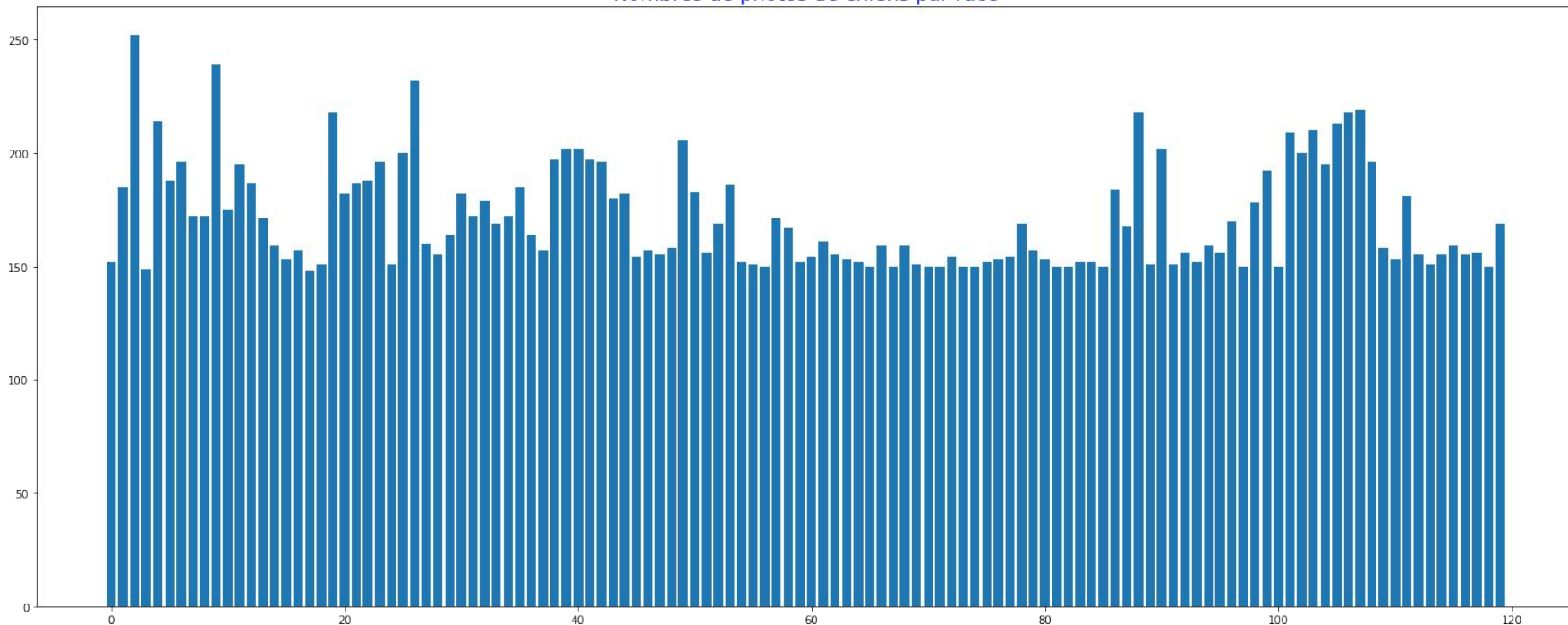
Les données

- Base de données : Stanford Dogs Dataset
- Plus de 20 000 images de chiens
- 120 races différentes



Les données

Nombres de photos de chiens par race

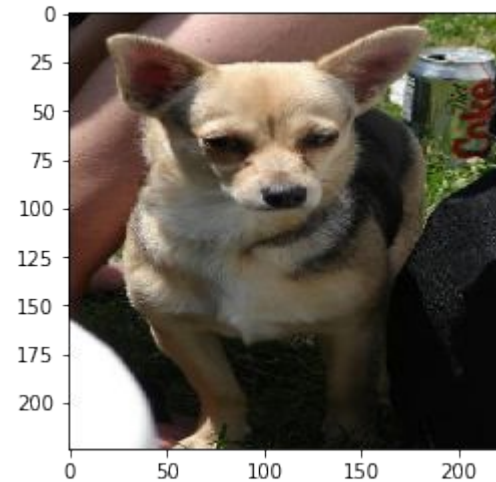
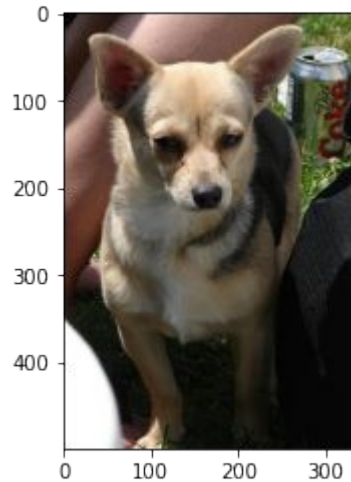


Prétraitement des images

- Avant de pouvoir appliquer nos modèles de Deep Learning
- Il est nécessaire de prétraiter les images
- Détaillons les différents prétraitements

Prétraitement des images

Dimension des images

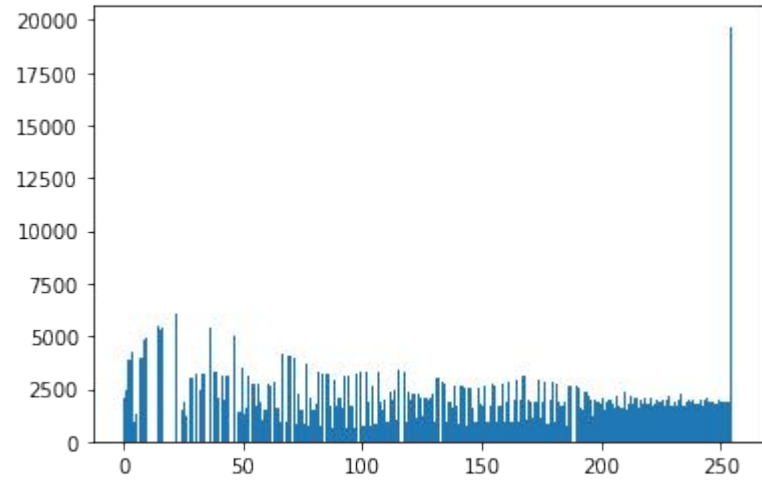
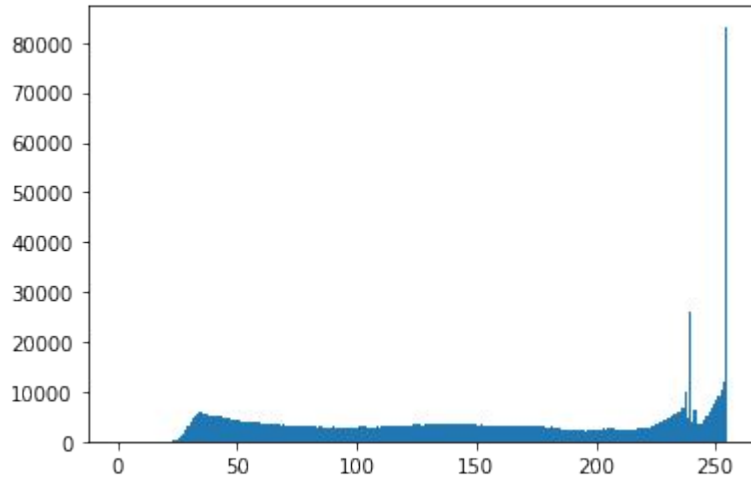


Prétraitement des images

- Dataset de grandes diversités de contrastes, d'expositions ...
- Nécessaire de prétraiter les images à l'aide des histogrammes

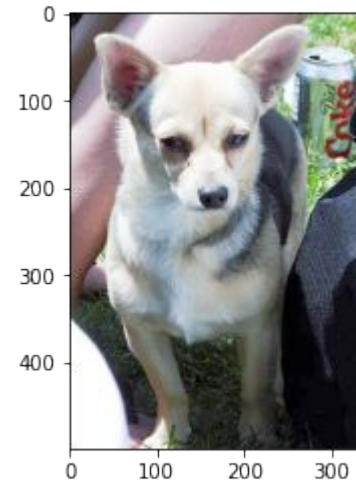
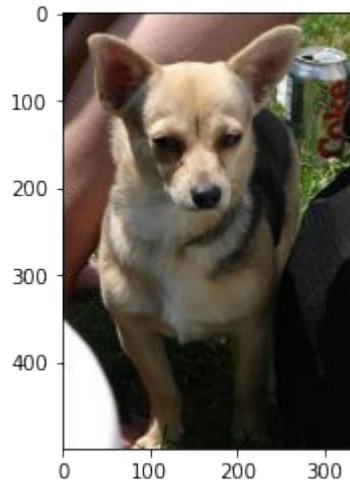
Prétraitement des images

Modification de l'histogramme des images



Prétraitement des images

Modification de l'histogramme des images

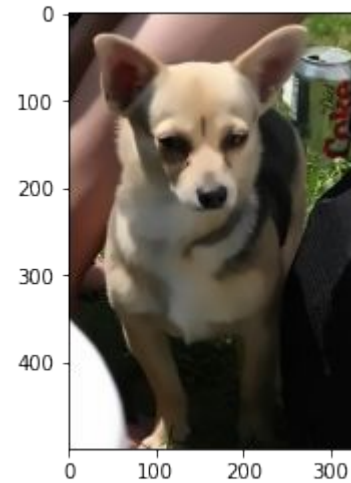
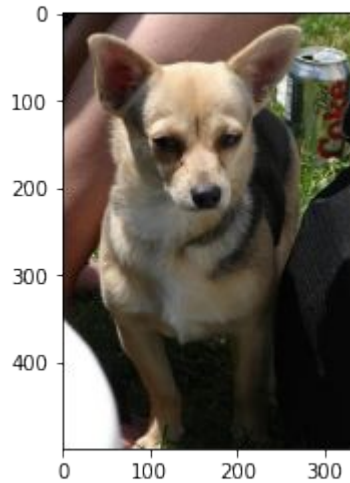


Prétraitement des images

- Les images peuvent être bruitées
- Il est possible d'appliquer un filtre pour réduire ce problème
- Nous utiliserons le filtre non local means

Prétraitement des images

Filtre non local means

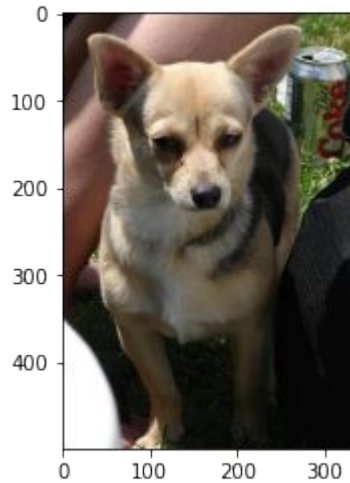


Prétraitement des images

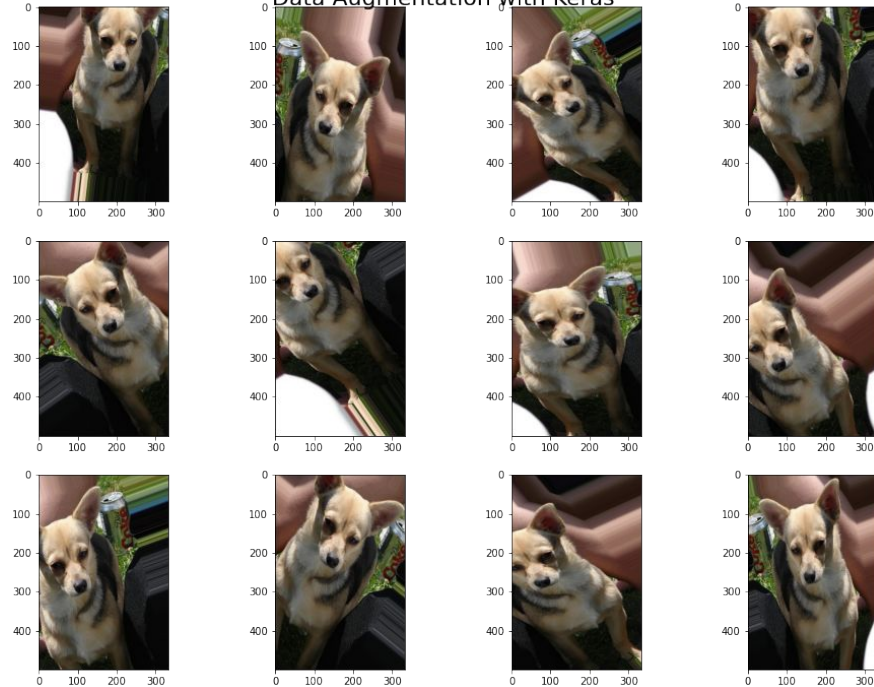
- Nous allons augmenter nos données en appliquant une transformation à nos images
- Du mirroring, des zooms , des rotations

Prétraitement des images

Data Augmentation



Data Augmentation with Keras



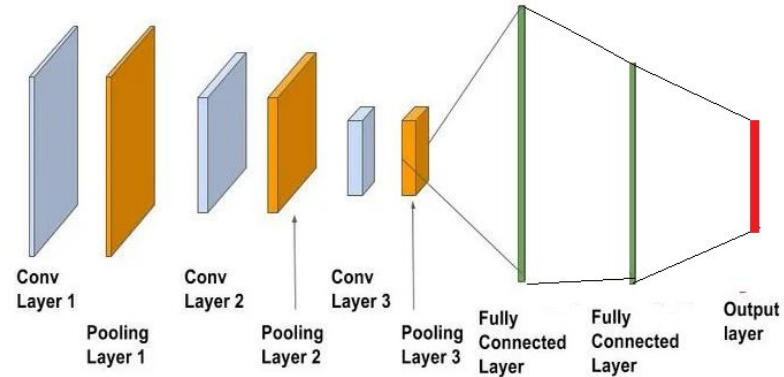
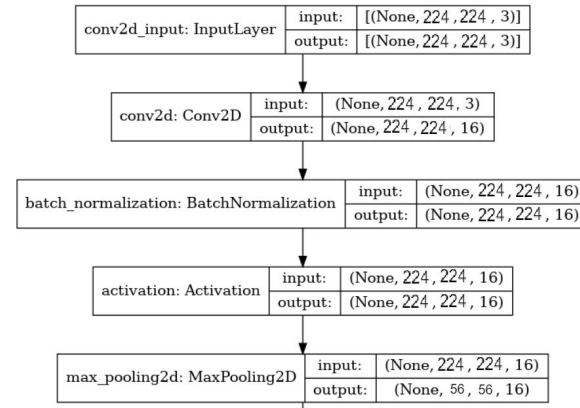
Modèles de Deep Learning

- Plusieurs méthodes différentes pour mettre en place notre algorithme
- Réaliser notre réseau de neurones à convolutions
- Utiliser des CNN existant : VGG16 et InceptionV3

Modèles de CNN

Différentes couches que nous appliquerons :

- Couches de convolution
- Couches de Pooling
- Couches de correction Relu
- Couches Fully connected
- DropOut



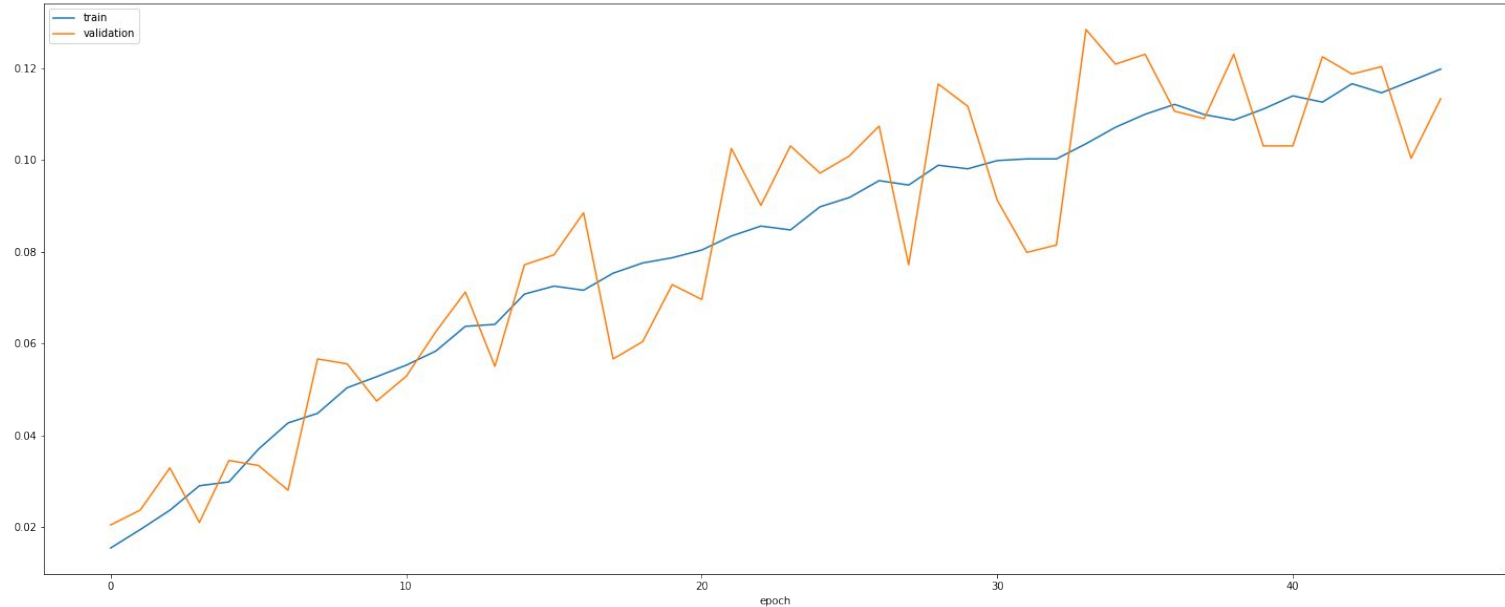
Modèles de CNN

Après avoir tester notre modèle une première fois nous avons fine tune les paramètres suivant :

- Couche Dense 1 et 2
- Drop Out 2
- Learning Rate
- Nombres d'epoch

Modèles de CNN

Entraînement de notre modèle CNN final



Transfert Learning de VGG16

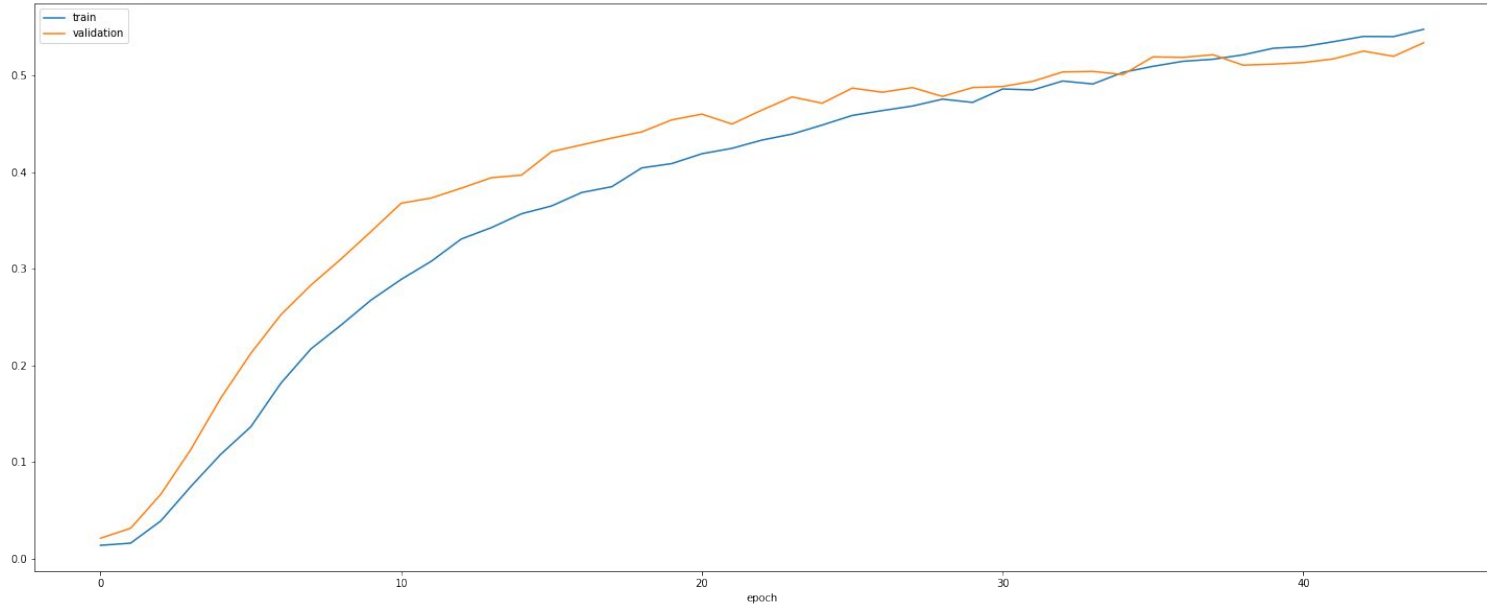
Dans un premier temps, nous allons entraîner seulement le classifieur, celui-ci sera composé

- Deux couches Denses d'activation Relu
- Un DropOut
- Une couche Dense d'activation Softmax

Nous allons fine tune ces différentes couches, le learning rate et le nombre d'epoch

Transfert Learning de VGG16

Entraînement du modèle final



Transfert Learning de InceptionV3

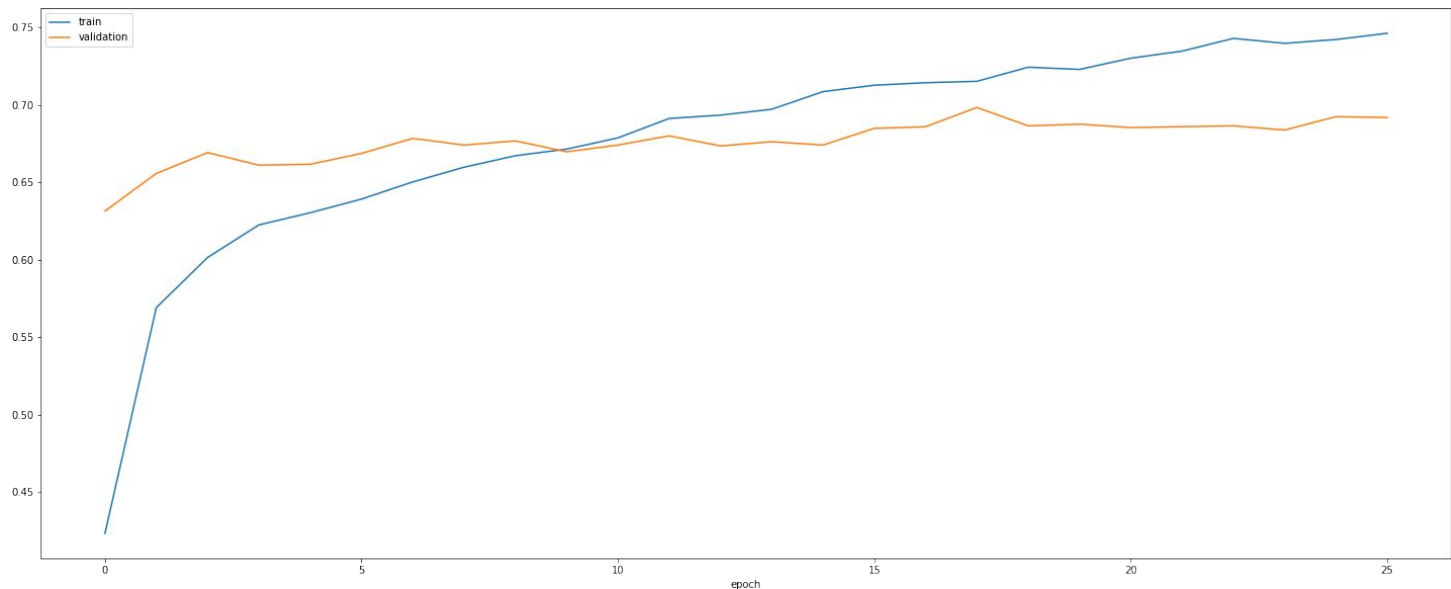
Nous allons entraîner seulement le classifieur, celui-ci sera composé

- Deux couches Denses d'activation Relu
- Un DropOut
- Une couche Dense d'activation Softmax

Nous allons fine tune ces différentes couches, le learning rate et le nombre d'epoch

Transfert Learning de InceptionV3

Entraînement du modèle finale



Transfert Learning de InceptionV3

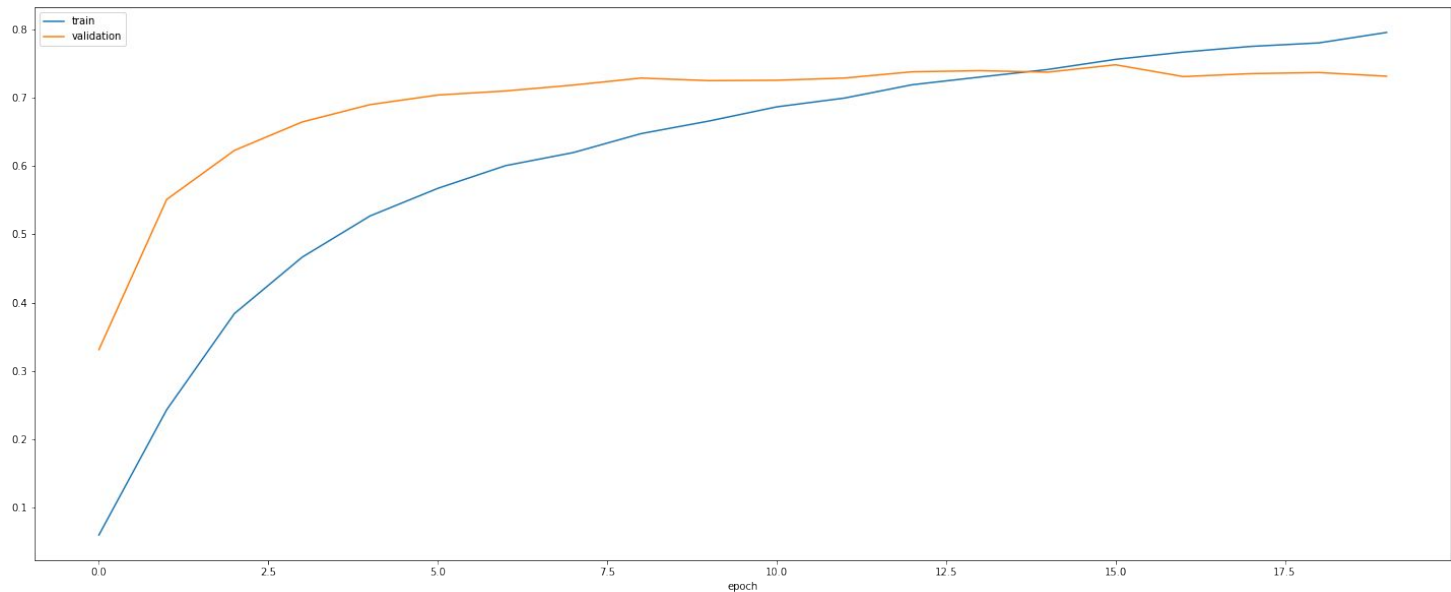
Nous allons entraîner seulement le dernier bloc et le classifieur, celui-ci sera composé

- Deux couches Denses d'activation Relu
- Un DropOut
- Une couche Dense d'activation Softmax

Nous allons fine tune ces différentes couches, le learning rate et le nombre d'epoch

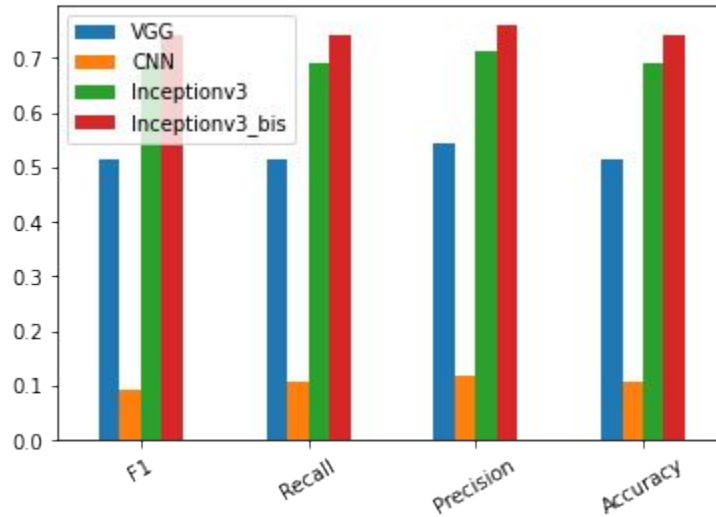
Transfert Learning de InceptionV3

Entraînement du modèle finale



Evaluation des modèles sur données Tests

Maintenant que nous avons tous les modèles optimisés, regardons les performances



Complément

- Le meilleur modèle est donc InceptionV3
- Piste d'amélioration : Tester de nouveaux modèles
- Création d'un programme qui prend une photo et qui renvoie la race du chien



Merci pour votre
attention

