# Projet : Image classification à l’aide du Deep Learning

# **Mise en situation**

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.

[](https://user.oc-static.com/upload/2019/02/23/15509395932064_image1.jpg)

Votre première rencontre avec Snooky

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.**

# **Les données**

Les bénévoles de l'association n'ont pas eu le temps de réunir les différentes images des pensionnaires dispersées sur leurs disques durs. Pas de problème, vous entraînerez votre algorithme en utilisant le [Stanford Dogs Dataset](http://vision.stanford.edu/aditya86/ImageNetDogs/).

# **Vos missions**

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.

Vous avez peu d’expérience sur le sujet, vous décidez donc de contacter un ami expert en classification d’images.

Il vous conseille dans un premier temps de **pré-processer des images avec des techniques spécifiques** (e.g. whitening, equalization, éventuellement modification de la taille des images) et de réaliser de la **data augmentation** (mirroring, cropping...).

Ensuite, il vous incite à **mettre en œuvre deux approches s’appuyant sur l’état de l’art et l’utilisation de CNN** (réseaux de neurones convolutionnels), que vous comparerez en termes de temps de traitement et de résultat :

1. Une première **en réalisant votre propre réseau CNN**, en vous inspirant de réseaux CNN existants. Prenez soin d'optimiser certains hyperparamètres (des layers du modèle, de la compilation du modèle et de l’exécution du modèle)
2. Une deuxième **en utilisant le transfer learning**, c’est-à-dire en utilisant un réseau déjà entraîné, et en le modifiant pour répondre à votre problème.

Concernant le**transfer learning**, votre ami vous précise que :

* Une première chose obligatoire est de réentraîner les dernières couches pour prédire les classes qui vous intéressent seulement.
* Il est également possible d’adapter la structure (supprimer certaines couches, par exemple) ou de réentraîner le modèle avec un très faible learning rate pour ajuster les poids à votre problème (plus long) et optimiser les performances.

# **Livrables attendus**

Un **repo** [**Github**](https://github.com/) contenant :

* Un notebook Python (non cleané, pour comprendre votre démarche).
* Un programme Python qui prend une image (array) en entrée et retourne la race la plus probable du chien présent sur l'image.
* Votre support de présentation à destination du bénévole qui gère la base de données, qui devra déployer en production la solution que vous préconisez.

# **Modalités de présentation du travail**

Votre présentation pourra prendre cette forme (à titre indicatif) :

|  |  |
| --- | --- |
| 5 min. | Présentation de la problématique, du cleaning effectué, du feature engineering et de l'exploration |
| 10 min | Présentation des différentes pistes de modélisation effectuées |
| 10 min | Présentation du modèle final sélectionné (pour chaque approche) ainsi que des performances et améliorations effectuées |
| 10 min | Séance de questions-réponses |

# **Ressources complémentaires**

#### **Prétraitement des images**

* Présentation d’une technique de preprocessing : le [whitening](http://mccormickml.com/2014/07/24/intuition-behind-whitening-image-patches/).
* Ce [tutoriel](https://machinelearningmastery.com/image-augmentation-deep-learning-keras/) présente les concepts de data augmentation. À vous de voir si cela peut améliorer vos performances sur ce projet !
* Présentation d’une autre technique de preprocessing : [l’equalization](https://en.wikipedia.org/wiki/Histogram_equalization).
* Deux cours sur le pre-processing d’images : le [premier](https://courses.cs.washington.edu/courses/cse457/11au/lectures/image-processing.pdf) est succinct et illustre les techniques, c’est une bonne première approche ; le [second](http://www.bioss.ac.uk/people/chris/ch3.pdf) fournit beaucoup de détails qui permettent de mieux appréhender les cas d’utilisation des différents filtres, notamment.

#### **2) CNN et transfer learning**

* Guide d’utilisation de [Google Colaboratory avec GPU](https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d).
* Une [présentation](https://towardsdatascience.com/transfer-learning-using-keras-d804b2e04ef8) du transfer learning, ses avantages, les cas d’application et un exemple de code en Keras.

# **Compétences data à valider**

* Mettre en place un modèle de Deep Learning
* Se familiariser avec les réseaux de neurones profonds
* Adapter les paramètres d'un modèle de Deep Learning an de l’améliorer
* Sélectionner un modèle d'apprentissage Deep Learning adapté à une problèmatique métier
* Transformer les variables pertinentes d'un modèle de Deep Learning
* Évaluer les performances d’un modèle de Deep Learning