# Projet DS51

## Introduction

Ce rapport a pour but de détailler notre projet réalisé à l’occasion de l’UV DS51. Avant de parcourir les différentes étapes du projet, il nous semble adéquat de reformuler le sujet selon notre compréhension de celui-ci :

Notre but dans ce projet, est de, depuis une base d’images à récupérer (ImagesNet), organiser les différents groupes d’animaux au sein d’une onthologie RDF, puis d’appliquer un modèle de Machine Learning à chaque « nœud » de cette onthologie pour prédire la classification dans un des différents groupes d’une nouvelle image.

## Récupération des data

La banque d’images que nous devons récupérer est présente sur le site suivant : https://www.kaggle.com/competitions/imagenet-object-localization-challenge/data

Dans cette banque d’images (représentant 170 GB), sont classées plusieurs catégories d’entités, représentés par des ID. Toutes les photos représentant un type d’entité en particulier sont regroupées dans un dossier qui porte comme titre l’ID du type d’entité en question.

Une image contenant texte, capture d’écran, logiciel, Site web

Description générée automatiquement

Une image contenant texte, capture d’écran, nombre, diagramme

Description générée automatiquement

### Etape 1 : déchiffrer les ID

Bien que l’organisation des types d’entités soit fonctionnelle dans le projet ImagesNet, ces ID ne veulent pas dire grand-chose et nous allons devoir les déchiffrer. Pour cela, nous avons la chance de disposer du fichier « LOC\_synset\_mapping.txt », qui assimile à chaque ID plusieurs labels qui sont des synonymes. (Example : « great grey owl », « great gray owl », « Strix nebulosa »). Ainsi à l’aide d’une fonction simple, nous pouvons construire un dictionnaire pour récupérer chacun des libellés associés à l’ID de ImagesNet (nous nous serviront de ce dictionnaire plus tard).

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

***Figure 1 : contenu du fichier LOC\_synset\_mapping.txt***

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

***Figure 2 : fonction pour lire les données du fichier LOC\_synset\_mapping.txt***

### Etape 2 : Repérer uniquement les dossiers d’animaux

Le but de ce projet est de construire une ontologie d’animaux, ainsi nous aimerions disposer uniquement de différents types d’animaux. Or, la base ImagesNet contient d’autres entités (objets, services…). Nous n’allons pas manuellement vérifier si chaque dossier contient ou non des animaux, ainsi nous allons utiliser le site <https://www.wikidata.org> pour nous aider dans cette tâche.

Pour savoir si une entité présente sur WikiData est un animal, nous avons deux possibilités :

* L’entité est une sous-classe de la classe « animal » (d’ID « Q729 »)
* L’entité est présent dans le « taxon » de WikiData, et est un sous-taxon de « animal » ( Q729 )

Ainsi nous allons développer deux fonctions, assez similaires, utilisant l’API de WikiData, pour détecter si une entité est une sous-classe ou un sous-taxon de « animal» :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

***Figure 3 : fonction pour confirmer si une entité est une sous-classe de « animal »***

Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

***Figure 4 : fonction pour confirmer si une entité est un sous-taxon de « animal »***

### Etape 3 : Télécharger les dossiers concernant des animaux

Maintenant que nous pouvons lire le fichier qui associe chaque id à un libellé et que nous pouvons, grâce à l’API de WikiData, confirmer si oui ou non un libellé correspond à un animal, nous pouvons maintenant télécharger uniquement les dossiers qui contiennent des animaux pour fabriquer notre ontologie.

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, Police

Description générée automatiquement

***Figure 5 : fonction pour récupérer uniquement les noms des dossiers à télécharger***

## Construction de l’ontologie

A la suite de l’étape de récupération de data précédente, nous avons télécharge 20 ????????? dossiers représentant 20 types d’animaux différents. Nous allons maintenant constituer une ontologie pour classifier ces 20 types d’animaux, en utilisant la représentation RDF. Pour cela, nous allons utiliser une bibliothèque Python pour la réalisation d’ontologies RDF :