



## PROJET HAI923I - 2023-2024

Le projet est à faire en groupe. Un groupe est composé de **3 (max 4)** personnes.

Vous devez inscrire la composition du groupe sur le lien suivant :

<https://docs.google.com/spreadsheets/d/1L9YbPsfJQmwJEYrp0VtabN3EFquowAd29TdoDzOHWI/edit?usp=sharing>

(Vous devez bien identifier la personne qui remettra le projet sur Moodle)

Le projet consiste à travailler sur des images. Vous avez à votre disposition 3 jeux de données différents :

- Tiger vs autres animaux
- Fox vs. autres animaux
- Elephant vs autres animaux.

Les jeux de données ainsi que les premiers pré-traitements sont disponibles dans le notebook de la section projet 2022-2023.

**Remarque** : les jeux de données ne sont pas très volumineux pour que les apprentissages ne prennent pas trop de temps. Cependant en fonction de vos modèles ou lorsque vous devez générer des images par exemple l'apprentissage peut être long. N'oubliez pas que sur Colab vous pouvez utiliser des GPU (Menu Modifier/Paramètres du Notebook) qui pourront accélérer l'apprentissage de manière significative. N'oubliez pas non plus que vous pouvez mettre des checkpoints dans vos code afin qu'il sauvegarde vos modèles en cours d'apprentissage. Cela est très pratique si un apprentissage est trop long et que votre session est coupée : il suffit de poursuivre l'apprentissage.

Voir par exemple :

<https://saturncloud.io/blog/how-to-continue-training-a-saved-and-loaded-keras-model-a-comprehensive-guide/>

<https://androidkt.com/how-to-save-load-model-and-continue-training-using-the-hdf5-file-in-keras/>

### Travail à faire :

- 1) Tout d'abord vous devez créer un modèle de classifieur de base (baseline), *modeleBaseline*, (1 seule couche de CNN) et l'évaluer sur les différents jeux de données. Est-ce que le modèle baseline est performant pour toutes les classes d'animaux. Vous pouvez faire varier les hyperparamètres (sans modifier le modèle) et vérifier si vous pouvez mieux classer une classe. Il est important de sauvegarder les résultats qui seront mis dans le rapport final.



- 2) Proposer des améliorations aux modèles précédents. Pour chacune des classes vous pouvez modifier l'architecture du modèle : *modeleTiger*, *modeleFox*, *modeleElephant*. L'objectif est d'obtenir le meilleur classifieur avec les meilleurs hyperparamètres pour chacun des jeux de données. Pour chaque modèle défini (par exemple *modeleTiger*) vous le testerez sur les autres classes afin de voir s'il est performant ou pas. Comme précédemment n'oubliez pas de sauvegarder les résultats qu'il faudra mettre dans le rapport.
- 3) A l'aide d'Image Data Generator, générer de nouvelles données et comparer les résultats de vos classifieurs précédents (*modeleBaseline*, *modeleTiger*, *modeleFox*, *modeleElephant*) avec les nouvelles données. Est-ce que vous obtenez de meilleurs résultats ? Vous pouvez ici modifier et proposer de nouveaux modèles pour prendre en compte ces nouvelles données : *modeleIDGTiger*, *modeleIDGFox*, *modeleIDGElephant*. Sauvegardez bien les résultats pour le rapport.
- 4) Appliquer une approche de transfer learning sur les trois jeux de données : *modeleTLTiger*, *modeleTLFox* et *modeleTLElephant*. Est-ce que les résultats obtenus sont meilleurs que les modèles précédents ?
- 5) A l'aide d'un GAN, sélectionner les données de Fox et générer des images jusqu'à ce qu'elles ressemblent le plus possible, en les regardant, à des renards. Sauvegarder 10 images générées. Faites de la prédiction de ces images avec le meilleur modèle précédent (*modeleFox* ou *modeleIDGFox* ou *modeleTLFox*). L'objectif est de vérifier comment ces images sont perçues dans le modèle.
- 6) Pour votre modèle le plus complexe, afficher le contenu des sorties des CNN.
- 7) (Optionnel). Pour les personnes intéressées par les autoencodeurs ou les variational autoencoders, vous pouvez faire de la coloration d'images. Le principe est le suivant. Prendre un jeu de données, par exemple Tiger, et sauvegarder les images en noir et blanc (voir par exemple : <https://www.delftstack.com/fr/howto/python/convert-image-to-grayscale-python/>). Par la suite il suffit de créer un modèle d'autoencodeur qui prend en entrée les images noir et blanc et les images couleurs.

## Travail à rendre :

Vous devez déposer sur Moodle (**Date de remise : 11/12**) une archive tar ou zip identifiée par votre numéro de groupe (par exemple groupe1.tar) contenant un rapport ainsi que les codes réalisés.

Attention la page de garde du rapport ainsi que tous les codes doivent contenir le numéro d'étudiant, le nom et le prénom de tous les membres du groupe.

Le rapport doit contenir :

- La définition de tous les modèles : (c'est juste un `summary()` du modèle), les valeurs des hyperparamètres ainsi que les courbes Loss et Accuracy. Une conclusion pour chacune des parties sur ce que vous tirez des expérimentations. Il s'agit de prendre du recul sur vos expérimentations. Par exemple vous pouvez répondre à des questions du type : « pourquoi le modèle X est plus performant ou moins performant que le modèle Y dans la classification des images de Tigre ? », « quel est l'impact du nombre d'images ? » etc. **Nous serons très regardant sur le recul que vous prendrez sur vos expérimentations.**



- La définition du modèle, les hyperparamètres, les 10 images de Fox générées et le résultat de la prédiction pour chacune des images.
- Des plots correspondant aux sorties des CNN
- Pour ceux qui ont fait la partie optionnelle, il faut montrer le modèle, les valeurs des hyperparamètres, les courbes Loss et Accuracy ainsi que des exemples d'images colorisées.
- Une partie conclusion personnelle sur ce que vous avez appris ou sur les difficultés rencontrées.

**Remarques** : pour le rapport il ne faut pas perdre de temps à décrire les données ou les méthodes utilisées.

En gros il s'agit de copies d'écrans de vos résultats (modèles, courbes, hyperparamètres utilisés) et de tableau résumant et surtout de prendre du recul sur les résultats obtenus.