



PROJET HAI923I - 2022-2023

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/1i6RW9CAz4kPZF47WjDyEuTbsU0dB760mLmr2XBf0ICE/edit?usp=sharing>

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://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/save_and_load.ipynb#scrollTo=2IZxbwiRRSD2

Travail à faire :

- 1) Pour chacun des jeux de données vous devez créer un modèle de classifieur de base (baseline) qui servira de comparaison par la suite : *modelebaselineTiger*, *modelebaselineFox*, *modelebaselineElephant*. Attention les données étant très différentes, le modèle de base ne sera pas forcément le même.
- 2) Proposer des améliorations aux modèles précédents : *modeleTiger*, *modeleFox*, *modeleElephant*. L'objectif est d'obtenir le meilleur classifieur pour chacun des jeux de données. Vous testerez si un modèle est le meilleur sur l'ensemble des données.
- 3) A l'aide d'Image Data Generator, générer de nouvelles données et comparer les résultats de vos classifieurs précédents avec les nouvelles données : *modeleIDGTiger*, *modeleIDGFox*, *modeleIDGElephant*. Est-ce que vous obtenez de meilleurs résultats ?



- 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 deux 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. Faire 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, 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 : 4/12**) 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 des modèles : *modelebaselineTiger*, *modelebaselineFox*, *modelebaselineElephant* (c'est juste un summary() du modèle), les valeurs des hyperparamètres ainsi que les courbes Loss et Accuracy.
- La définition des modèles : *modeleTiger*, *modeleFox*, *modeleElephant*, les valeurs des hyperparamètres ainsi que les courbes Loss et Accuracy.
- La partie de code correspondant à l'apprentissage pour le meilleur des trois modèles : *modeleIDGTiger*, *modeleIDGFox*, *modeleIDGElephant*, les valeurs des hyperparamètres ainsi que les courbes Loss et Accuracy.
- La définition des modèles : *modeleTLTiger*, *modeleTLFox* et *modeleTLElephant*, les valeurs des hyperparamètres ainsi que les courbes Loss et Accuracy.
- 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) avec quelques explications éventuelles sur les raisons pour lesquelles un modèle est meilleur qu'un autre l'autre et d'une conclusion personnelle.