**Guide d’utilisation pour l’Office Français de la Biodiversité**

*Auteur : Marie Joigneau (stagiaire encadrée par Olivier Gimenez)*

*Stage réalisé entre février et août 2023*

Ci-dessous un guide d’utilisation afin d’appliquer l’Intelligence Artificielle pour l’identification d’individu avec une base de données déjà annotée.

**Le meilleur modèle trouvé à la fin du stage nécessite des images de taille 260 x 260.**

**Son seuil est de 0.53.**

**Etape 1 : Mise à jour des packages**

**Liste des packages à mettre à jour**

Afin de pouvoir utiliser les codes sous Python, il est nécessaire de mettre à jour des packages. Pour ce faire on utilise la commande suivante dans la console ou le terminal :

*pip install package*

Ci-dessous la liste que vous pouvez retrouver dans les premières lignes des codes :

dateutil.parser

keract

kerasgen.balanced\_image\_dataset

os

matplotlib.pyplot

numpy

pandas

PIL

sklearn.neighbors

tensorflow

tensorflow\_addons

tensorflow.keras.applications.vgg16

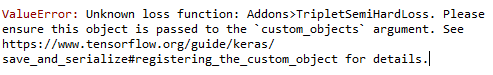
tensorflow.keras.preprocessing.image

tqdm

**Résoudre une erreur de Tensorflow**

Afin de palier à une erreur de Tensorflow pour la sauvegarde et le chargement des modèles de deep learning EfficientNet, il faut rétrograder le package. On a besoin de tensorflow 2.9.

<https://discuss.tensorflow.org/t/using-efficientnetb0-and-save-model-will-result-unable-to-serialize-2-0896919-2-1128857-2-1081853-to-json-unrecognized-type-class-tensorflow-python-framework-ops-eagertensor/12518/10>



Pour ce faire, on utilise la commande suivante dans la console ou le terminal :

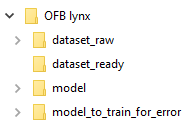
*pip install --upgrade tensorflow==2.9.1*

**Etape 2 : Préparer les images**

Cette étape consiste à préparer les images afin qu’elle puisse être conforme au modèle et pertinente.

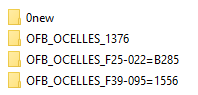
**Option 1 : Les images sont déjà centrées sur l’individu et ont la taille demandée par le modèle.**

Dans votre fichier *OFB lynx*, créez un sous fichier *dataset\_ready* comme ci-dessous.



Mettez vos images prêtes dans le sous fichier nommé *dataset\_ready*. Les images doivent être classées comme ci-dessous :

* Les images à prédire sont dans le dossier *0new*.
* Les images dont on connaît l’individu sont rassemblés dans le dossier de l’individu correspondant.

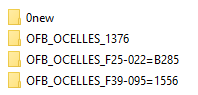


**Option 2 : Les images sont déjà centrées sur l’individu mais ont des tailles différentes de celle demandée pour le modèle.**

**Mise en place des images**

Mettez vos images prêtes dans le sous fichier nommé *dataset\_raw*. Les images doivent être classées comme ci-dessous :

* Les images à prédire sont dans le dossier *0new*.
* Les images dont on connaît l’individu sont rassemblés dans le dossier de l’individu correspondant.



**Utilisation du code**

Ouvrez le fichier *1\_Pre-processing\_OFB\_resize-only.py* grâce à un utilitaire Python comme Spyder.

Lancez le code .

Deux questions vont vous être posées dans la console :

**1ère question :** ***Write here the size you want / Ecrivez ici la taille que vous désirez.***

Vous répondez 260 par exemple si le modèle nécessite des images de taille 260 x 260.

**2ème question : *Write here the directory of the OFB folder (end by '/OFB lynx' or '/OFB jaguar') / Ecrivez ici le chemin d’acces du dossier OFB (fini par '/OFB lynx' ou '/OFB jaguar')***

Vous répondez par exemple *D:/my\_code/OFB lynx*.

Les résultats sont dans un nouveau dosser *dataset\_ready*.

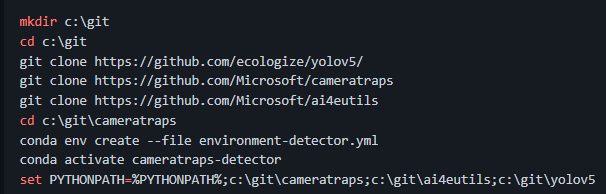
Vos images sont prêtes : elles ont toutes la même taille n\*n.

**Option 3 : Les images ne sont pas centrées sur l'individu et ont des tailles différentes de celle demandée pour le modèle.**

**Installer Megadetector**

Sur <https://github.com/microsoft/CameraTraps/blob/main/megadetector.md>, téléchargez le modèle Megadetector mV5a. Le fichier, une fois téléchargé, se nomme *md\_v5a.0.0.pt*.

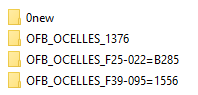
Importer les utilitaires pour finir d’installer Megadetector (voir ci-dessous) dans votre cmd (invite de commande).



**Mise en place des images**

Mettez vos images prêtes dans le sous fichier nommé *dataset\_raw*. Les images doivent être classées comme ci-dessous :

* Les images à prédire sont dans le dossier *0new*.
* Les images dont on connaît l’individu sont rassemblés dans le dossier de l’individu correspondant.



**Obtenir un fichier Megadetector**

Pour chaque obtention d’un fichier Megadetector, vous devez écrire sur votre terminal d’Anaconda, les 3 lignes suivantes :

To use run\_detector.py on Windows, when you open a new Anaconda prompt, don't forget to do this:

cd c:\git\CameraTraps

conda activate cameratraps-detector

set PYTHONPATH=%PYTHONPATH%;c:\git\cameratraps;c:\git\ai4eutils;c:\git\yolov5

Puis vous tapez dans le terminal :

python detection\run\_detector\_batch.py "c:\megadetector\md\_v5a.0.0.pt" "c:\some\_image\_folder" "c:\megadetector\test\_output.json" --output\_relative\_filenames --recursive --checkpoint\_frequency 10000

c:\megadetector\md\_v5a.0.0.pt : chemin d’accès du modèle Megadetector (doit se terminer par « OFB/md\_v5a.0.0.pt »)

c:\some\_image\_folder : chemin d’accès du dossier regroupant vos images (/!\ vous devez bien mettre un chemin d’accès se terminant par « …/OFB/dataset\_raw », étant donné que vos images sont dans ce dossier).

c:\megadetector\test\_output.json : chemin d’accès du résultat de Megadetector (doit se terminer par « OFB/*megadetector\_results.json*»)

Renommez si ce n’est déjà fait le fichier du résultat de Megadetector en *megadetector\_results.json* et mettez-le dans le dossier */OFB lynx* ou */OFB jaguar*.

**Utilisation du code**

Ouvrez le fichier *1\_Pre-processing\_OFB\_crop-resize.py* grâce à un utilitaire Python comme Spyder.

Lancez le code .

Plusieurs questions vont vous être posées dans la console :

**1ère question :** **W*rite here the size you want / Ecrivez ici la taille que vous désirez.***

Vous répondez 260 par exemple si le modèle nécessite des images de taille 260 x 260.

**2ème question :** ***Write here the directory of the OFB folder (end by '/OFB lynx' or ‘OFB jaguar’) / Ecrivez ici le chemin d’acces du dossier OFB (fini par /OFB)*.**

Vous répondez par exemple *D:/my\_code/OFB lynx*.

**3ème question : Which treshold you want for the bounding boxes? 0.26 recommanded / Quel seuil voulez-vous pour les boîtes qui cadrent ? 0.26 recommandé**

Par défaut, mettez 0.26. Si vous voyez que le code vous propose trop de boites, augmentez-le. 0.26 est une valeur que j’ai choisi arbitrairement au vu de mes expériences avec Megadetector.

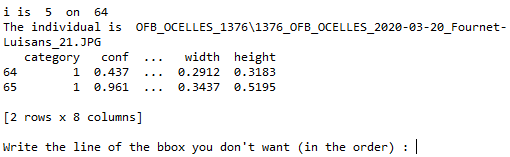
**4ème question :** **Do you want to filter automatically? (yes/no) / voulez-vous filtrer automatiquement? (yes/no)**

Si vous voulez que le code prenne automatiquement l’emplacement le plus probable de l’individu sur la photo, vous écrivez *yes*. Sinon vous écrivez *no*. 98% du temps il n’y a pas de problème avec Megadetector et la manière automatique de cadrer l’individu marche pour toutes les photos.

Si vous avez choisis le côté manuel, on va vous demandez d’écrire une lettre quand vous êtes prêt (Note the pictures you want to keep, and when you are ready, enter a letter / Notez les images à garder et écrivez une lettre quand vous êtes prêts).

Vous allez avoir toutes les propositions d’images croppées dans le dossier */OFB lynx/ bbox\_pb*. Vous allez avoir par exemple les images idx10\_1376\_OFB\_OCELLES\_2020-03-20\_Fournet-Luisans\_21 et idx11\_1376\_OFB\_OCELLES\_2020-03-20\_Fournet-Luisans\_21 dans le dossier */OFB lynx/bbox\_pb*.

Et vous allez devoir écrire l’index des propositions que vous ne gardez pas (ci-dessous si vous voulez garder la 1ère image vous notez 65).



Une fois fini, supprimez le fichier */OFB lynx/bbox\_pb*.

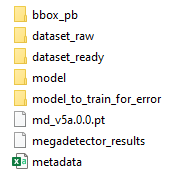
Les résultats sont dans un nouveau dosser *dataset\_ready*.

Vos images sont prêtes.

**Etape 2 : Lancer un modèle quelconque**

Dû à un problème de non-reconnaissance de la fonction de perte du modèle comme expliqué dans l’étape 1, partie *Résoudre une erreur de Tensorflow*, vous devez lancer un modèle avec cette fonction de perte pour pouvoir réaliser l’étape suivante.

L’algorithme va utiliser le fichier *model\_to\_train\_for\_error* pour générer un modèle, modèle que vous ne sauvegarderez pas.



Ouvrez le fichier *2\_Model-construction\_OFB.py* grâce à un utilitaire Python comme Spyder.

Lancez le code 

Une question va être posées dans la console.

**Question : *Write here the directory of the OFB folder (end by '/OFB lynx' or ‘/OFB jaguar’) / Ecrivez ici le chemin d’acces du dossier OFB (fini par ‘/OFB lynx’ ou ‘/OFB jaguar’)*.**

Vous répondez par exemple *D:/my\_code/OFB lynx*.

**Etape 3 : Lancer le modèle**

**Option 1 : Vous ne voulez pas filtrer votre jeu de données**

Ouvrez le fichier *5\_Model-prediction-already-dataset\_manuel\_OFB.py* grâce à un utilitaire Python comme Spyder.

Lancez le code .

Deux questions vont vous être posées dans la console

**1ère question :** **Write here the treshold / Ecrivez le seuil**

Vous répondez le seuil donné au début du manuel.

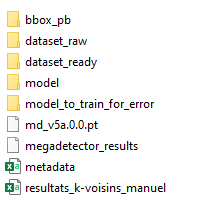
**2ème question : *Write here the directory of the OFB folder (end by '/OFB lynx' or ‘/OFB jaguar’) / Ecrivez ici le chemin d’acces du dossier OFB (fini par ‘/OFB lynx’ ou ‘/OFB/jaguar’)*.**

Vous répondez par exemple *D:/my\_code/OFB lynx*.

**3ème question :** **Do you want to filter individuals? (yes/no) / Voulez-vous filtrer les individus (écrire yes/no).**

Vous répondez par *no.*

Les résultats sont sous la forme d’un csv nommé *resultats\_k-voisins\_manuel* dans ‘*/OFB lynx’ ou ‘/OFB jaguar’* (voir ci-dessous).



**Option 2 : Vous voulez filtrer votre jeu de données par les dates / lieux pour la prédiction**

**Mise en place des métadonnées**

Mettez vos métadonnées sous la forme d’un csv nommé *metadata.csv*. Il doit ressembler à l’exemple ci-dessous avec l’identifiant du lynx (*lynx\_ID*), les lieux où il a été identifié (*place*), les dates où il a été identifié (*date*) et le nom des différentes images (*picture*).

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

Description générée automatiquement

Mettez les métadonnées dans le dossier */OFB lynx ou /OFB jaguar* avec les autres fichiers (voir ci-dessous).

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Utilisation du code**

Ouvrez le fichier *5\_Model-prediction-already-dataset\_manuel\_OFB.py* grâce à un utilitaire Python comme Spyder.

Lancez le code .

Plusieurs questions vont vous être posées dans la console

**1ère question :** **Write here the treshold / Ecrivez le seuil**

Vous répondez le seuil donné au début du manuel.

**2ème question :** ***Write here the directory of the OFB folder (end by '/OFB lynx' or ‘/OFB jaguar’) / Ecrivez ici le chemin d’acces du dossier OFB (fini par ‘/OFB lynx’ ou ‘/OFB/jaguar’)*.**

Vous répondez par exemple *D:/my\_code/OFB lynx*.

**3ème question :** **Do you want to filter individuals? (yes/no) / Voulez-vous filtrer les individus (écrire yes/no).**

Vous répondez par *yes* comme vous voulez réduire votre jeu de données par rapport à vos métadonnées.

**4ème question :** **Choose the year you want (none if not) / Choisissez l’année à filtrer (écrivez none sinon)**

Vous répondez par exemple 2006.

**5ème question :** **Choose the place you want (none if not) / Choisissez le lieu pour filtrer (écrivez none sinon)**

Vous répondez par exemple *none.*

Les résultats sont sous la forme d’un csv nommé *resultats\_k-voisins\_manuel* dans ‘*/OFB lynx’ ou ‘/OFB jaguar’*.

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

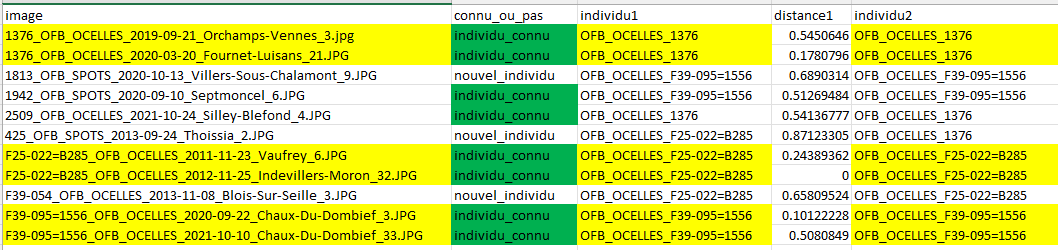
Description générée automatiquement

**Etape 4 : Lire les résultats**

Ce csv est un outil d’aide à la décision pour la personne en charge de déterminer quel individu est sur l’image.

Dans notre exemple (ici nous n’avons pas appliqué de filtre date/lieu), les individus surlignés en jaune sont ceux qu’on connaissait. En vert j’ai surligné quand l’algorithme disait qu’il connaissait l’individu. On remarque :

* Les images les plus proches sont du même individu (individu1). L’algorithme voit bien que ce sont des individus connus (surlignés en vert) dans la colonne *connu\_ou\_pas*
* Pour les nouveaux, sur 5 images, l’algorithme reconnaît bien que 3 images appartiennent à un nouvel individu.



Bonne utilisation,

N’hésitez pas si vous avez des questions,

Marie :-)