**User guide for the French Biodiversity Office**

*Author: Marie Joigneau (intern supervised by Olivier Gimenez)*

*Internship carried out between February and August 2023*

Below is a user guide to apply Artificial Intelligence for individual identification with an already annotated database.

**The best model found at the end of the course requires images of size 260 x 260.**

**Its threshold is 0.53.**

**Step 1: Updating packages**

**List of packages to update**

In order to use the codes in Python, it is necessary to update packages. To do this, use the following command in the console or terminal:

*pip install package*

Below is the list that you can find in the first lines of the codes:

dateutil.parser

keract

kerasgen.balanced\_image\_dataset

bone

matplotlib.pyplot

numpy

pandas

PIL

sklearn.neighbors

tensorflow

tensorflow\_addons

tensorflow.keras.applications.vgg16

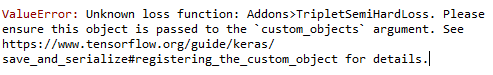
tensorflow.keras.preprocessing.image

tqdm

**Resolve a Tensorflow error**

In order to overcome an error in Tensorflow for saving and loading EfficientNet deep learning models, you must downgrade the package. We need 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](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)



To do this, use the following command in the console or terminal:

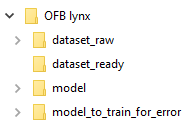
*pip install --upgrade tensorflow==2.9.1*

**Step 2: Prepare the images**

This step involves preparing the images so that it can be consistent with the model and relevant.

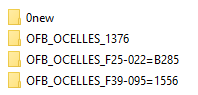
**Option 1: The images are already centered on the individual and are the size requested by the model.**

In your OFB lynx file, create a dataset\_ready subfile as below.



Put your ready images in the subfile named dataset\_ready. Images should be categorized as below:

* The images to be predicted are in the 0new folder.
* The images for which the individual is known are collected in the file of the corresponding individual.

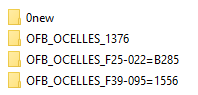


**Option 2: The images are already centered on the individual but have different sizes than requested for the model.**

**Setting up the images**

Put your ready images in the subfile named dataset\_raw. Images should be categorized as below:

* The images to be predicted are in the 0new folder.
* The images for which the individual is known are collected in the file of the corresponding individual.



**Code usage**

Open the 1\_Pre-processing\_OFB\_resize-only.py file using a Python utility like Spyder.

Run the code.

You will be asked two questions in the console:

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

You answer 260 for example if the template requires images of size 260 x 260.

**2nd 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')***

You answer for example D:/my\_code/OFB lynx.

The results are in a new dataset\_ready folder.

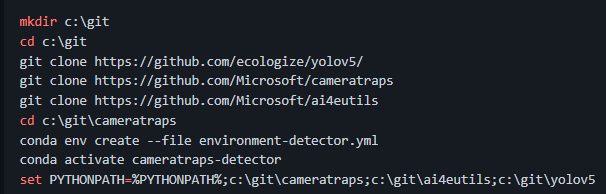
Your images are ready: they all have the same size n\*n.

**Option 3: Images are not focused on the individual and have different sizes than requested for the model.**

**Install Megadetector**

On<https://github.com/microsoft/CameraTraps/blob/main/megadetector.md>, download the Megadetector mV5a model. The file, once downloaded, is named md\_v5a.0.0.pt.

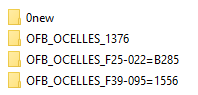
Import the utilities to finish installing Megadetector (see below) into your cmd (command prompt).



**Setting up the images**

Put your ready images in the subfile named dataset\_raw. Images should be categorized as below:

* The images to be predicted are in the 0new folder.
* The images for which the individual is known are collected in the file of the corresponding individual.



**Obtain a Megadetector file**

For each Megadetector file, you must write the following 3 lines on your Anaconda terminal:

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

Then you type in the 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: Megadetector model path (must end with “OFB/md\_v5a.0.0.pt”)

c:\some\_image\_folder: path to the folder containing your images (/!\ you must provide a path ending with “…/OFB/dataset\_raw”, given that your images are in this folder).

c:\megadetector\test\_output.json: path to Megadetector result (must end with “OFB/*megadetector\_results.json*")

If you have not already done so, rename the Megadetector result file to megadetector\_results.json and put it in the /OFB lynx or /OFB jaguar folder.

**Code usage**

Open the 1\_Pre-processing\_OFB\_crop-resize.py file using a Python utility like Spyder.

Run the code.

Several questions will be asked in the console:

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

You answer 260 for example if the template requires images of size 260 x 260.

**2nd 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)*.**

You answer for example D:/my\_code/OFB lynx.

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

By default, put 0.26. If you see that the code offers you too many boxes, increase it. 0.26 is a value that I chose arbitrarily based on my experiences with Megadetector.

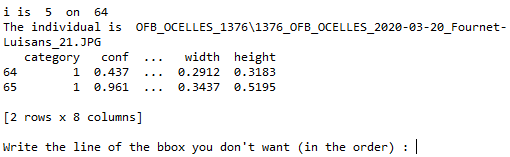
**4th question:** **Do you want to filter automatically? (yes/no) / voulez-vous filtrer automatiquement? (yes/no)**

If you want the code to automatically take the most likely location of the individual in the photo, you write yes. Otherwise you write no. 98% of the time there is no problem with Megadetector and the automatic way of framing the individual works for all photos.

If you chose the manual side, you will be asked to write a letter when you are ready (Note the pictures you want to keep, and when you are ready, enter a letter / Note the images to keep and write a letter when are you ready).

You will have all the cropped image proposals in the /OFB lynx/ bbox\_pb folder. For example, you will have the images idx10\_1376\_OFB\_OCELLES\_2020-03-20\_Fournet-Luisans\_21 and idx11\_1376\_OFB\_OCELLES\_2020-03-20\_Fournet-Luisans\_21 in the /OFB lynx/bbox\_pb folder.

And you will have to write the index of the propositions that you do not keep (below if you want to keep the 1st image you note 65).



Once finished, delete the /OFB lynx/bbox\_pb file.

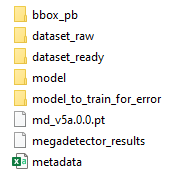
The results are in a new dataset\_ready folder.

Your images are ready.

**Step 2: Launch any model**

Due to a problem of not recognizing the loss function of the model as explained in step 1, part*Resolve a Tensorflow error*, you must run a model with this loss function to be able to perform the next step.

The algorithm will use the model\_to\_train\_for\_error file to generate a model, which you will not save.



Open the 2\_Model-construction\_OFB.py file using a Python utility like Spyder.

Run the code

A question will be asked in the 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’)*.**

You answer for example D:/my\_code/OFB lynx.

**Step 3: Launch the model**

**Option 1: You don't want to filter your dataset**

Open the 5\_Model-prediction-already-dataset\_manuel\_OFB.py file using a Python utility like Spyder.

Run the code.

You will be asked two questions in the console

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

You meet the threshold given at the beginning of the manual.

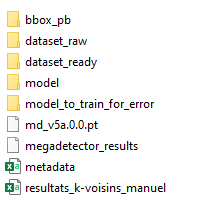
**2nd question: *Write here the directory of the OFB folder (end by '/OFB lynx' or ‘/OFB jaguar’) / Ecrivez ici le chemin d’acces du dossier OFB (fini par ‘/OFB lynx’ ou ‘/OFB/jaguar’)*.**

You answer for example D:/my\_code/OFB lynx.

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

You answer with no.

The results are in the form of a csv named results\_k-voisins\_manuel in '/OFB lynx' or '/OFB jaguar' (see below).



**Option 2: You want to filter your dataset by dates/locations for prediction**

**Setting up metadata**

Put your metadata in the form of a csv named metadata.csv. It should look like the example below with the lynx identifier (lynx\_ID), the places where it was identified (place), the dates it was identified (date) and the name of the different images (picture) .

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

Description générée automatiquement

Put the metadata in the /OFB lynx or /OFB jaguar folder with the other files (see below).

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Code usage**

Open the 5\_Model-prediction-already-dataset\_manuel\_OFB.py file using a Python utility like Spyder.

Run the code.

Several questions will be asked in the console

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

You meet the threshold given at the beginning of the manual.

**2nd 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’)*.**

You answer for example D:/my\_code/OFB lynx.

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

You answer with yes as you want to reduce your dataset in relation to your metadata.

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

You answer for example 2006.

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

For example, you answer none.

The results are in the form of a csv named results\_k-voisins\_manuel in '/OFB lynx' or '/OFB jaguar'.

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

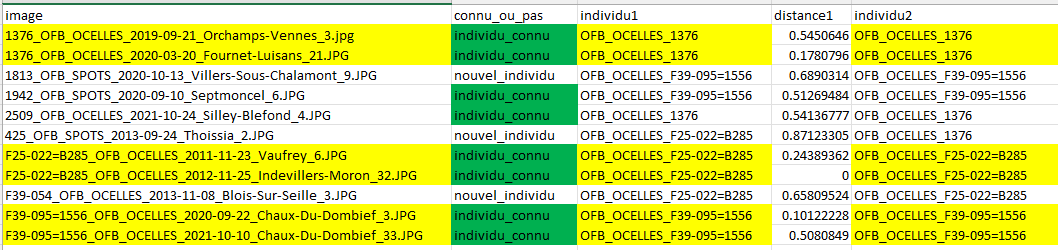
Description générée automatiquement

**Step 4: Read the results**

This csv is a decision support tool for the person responsible for determining which individual is in the image.

In our example (here we did not apply a date/location filter), the individuals highlighted in yellow are those we knew. In green I highlighted when the algorithm said it knew the individual. We notice :

* The closest images are of the same individual (individual1). The algorithm clearly sees that these are known individuals (highlighted in green) in the known\_or\_not column
* For new people, out of 5 images, the algorithm recognizes that 3 images belong to a new individual.



Good use,

Do not hesitate if you have any questions,

Married :-)