
Capteurs/actionneurs & IA embarquée

Projet Biodiversité CERN

Aurélia CHABANNIER, Sami EL KATEB



UNIVERSITÉ CÔTE D'AZUR

MAI 2023

Table des matières

1	Introduction	2
2	Matériel et Méthodes	3
2.1	Flux de travail	3
2.2	Jeu de données	6
2.3	Description du CNN	6
2.4	Architecture du traitement de données par le capteur	6
3	Résultats	7
4	Discussion	8

1 — Introduction

Le domaine du CERN est un terrain riche en matière de biodiversité et présente plusieurs espèces rares [1]. Dans le cadre d'une politique de la sensibilisation à l'environnement l'Organisation a mis en places plusieurs mesures visant à favoriser la biodiversité sur ses terrains. Une des mesures mise en œuvre implique des relevés des diverses espèces qui composent la faune et la flore du domaine du CERN [2]

Dans ce projet, notre objectif sera de proposer un moyen de faciliter le relevé des différentes espèces d'oiseaux présent sur le domaine du CERN. Pour cela, nous utiliserons un microcontrôleur STM32 NUCLEO-L476RG ainsi qu'un microphone analogique fourni par l'Université Côte d'Azur. Notre objectif sera de pouvoir différencier 10 espèces d'oiseau d'intérêt potentiellement présentes sur le domaine du CERN. Pour tester nos résultats, nous vérifierons que les espèces soient correctement reconnues par la carte. Nous effectuerons ces tests en lançant sur hauts parleurs différents sons d'oiseaux. Les sons utilisés pour réaliser ces tests en temps réel seront différents des sons présents dans le jeu de données d'entraînement et de test.

2 — Matériel et Méthodes

2.1 Flux de travail

Pour réaliser le jeu de données nécessaire à l'entraînement de notre réseau de neurone, nous avons utilisé le site Xeno-Canto [3]. Celui-ci recense un grand nombre de chants d'oiseau de différentes espèces enregistrés par des utilisateurs. Cependant, les chants d'oiseaux enregistrés possèdent des caractéristiques différentes (durée d'enregistrement, format d'enregistrement, sample rate, bruit de fond). Notre premier défi a donc été de créer un jeu de données permettant d'obtenir des résultats corrects. Pour cela, nous nous sommes fixé pour objectif d'obtenir une précision supérieure à 80% sur les données de test avec un réseau de neurone de type M5 [4]. Cela nous permet entre autre de valider notre jeu de données. Nous avons donc suivi une approche itérative 2.1

À chaque itération, nous avons essayé d'obtenir une meilleure précision qu'à l'itération précédente. Notre première approche a consisté à récupérer le son de 10 espèces d'oiseaux depuis Xeno-Canto puis à les convertir en .wav avant de tester le réseau de neurone M5 dessus.

Pour ce faire nous avons développé en script de récupération de données (scrapping) en Python nous permettant de sélectionner les espèces que nous voulions télécharger depuis le site. Lors de la récupération des sons, nous avons choisi de laisser un délai de 1 seconde entre chaque téléchargement pour ne pas surcharger le site. Nous avons également utilisé un script de conversion des fichiers en wav et de resampling en 16 000 Hz. Les scripts utilisés sont disponibles sur notre repository GitHub [5].

Cette approche ne nous a malheureusement pas permis d'obtenir des résultats satisfaisant. En effet, le réseau de neurone M5 ne nous permettait d'obtenir qu'une précision de 20%.

Nous avons donc choisis de découper les sons de notre jeu de données en échantillons de 1 seconde. Ce traitement nous a permis d'obtenir une amélioration de 20% de la précision du réseau de neurone M5 qui est passée de 20% à 40%.

Cependant, nous pouvons trouver plusieurs défauts à cette approche. La principale est que sur les enregistrements de plusieurs minutes, les chants d'oiseau ne sont pas constant et il existe des périodes de silence. Nous en déduisons qu'une des raisons de la mauvaise performance de notre réseau de neurone et que certains échantillons ne contiennent pas de chants d'oiseaux.

Nous avons réfléchi à plusieurs façon de résoudre ce problème :

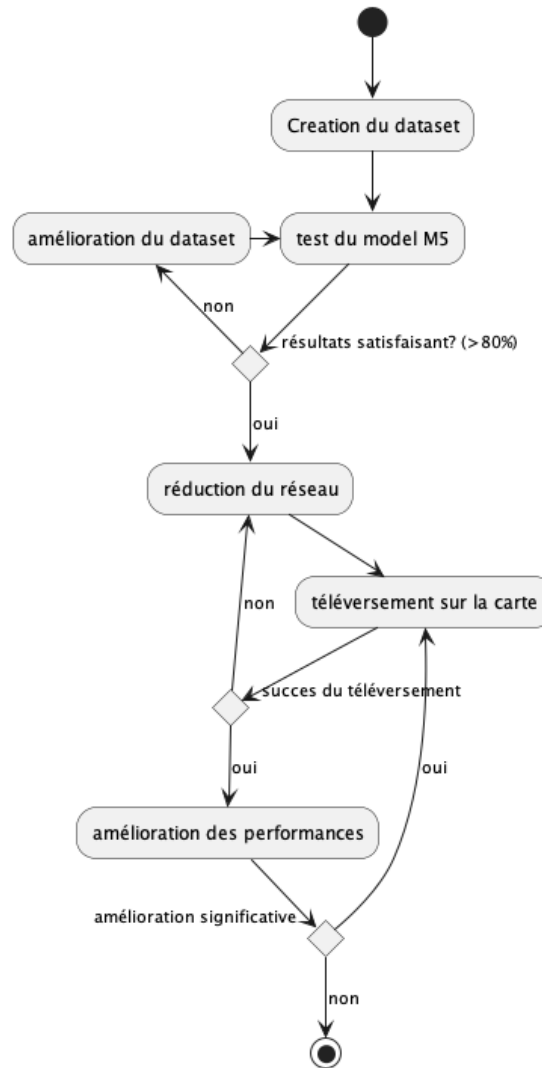


FIGURE 2.1 : Workflow Iteratif

- Contrôler manuellement la présence ou non de chant d'oiseau
- Filtrer les sons automatiquement et supprimer les silences
- Entraîner un algorithme non supervisé pour permettre de détecter les échantillons potentiellement mal labélisés

Pour des raisons de temps et d'efficacité, nous avons choisi la seconde approche. Pour cela, nous avons utilisé le script "Silence Remove" proposé dans l'article Audio Processing and Remove Silence using Python [6]

Cette approche nous a finalement permis d'obtenir un taux de précision adéquat avec une précision de 82% sur les données de tests. Nous avons donc finalement utilisé le flux

de travail Figure 2.2.

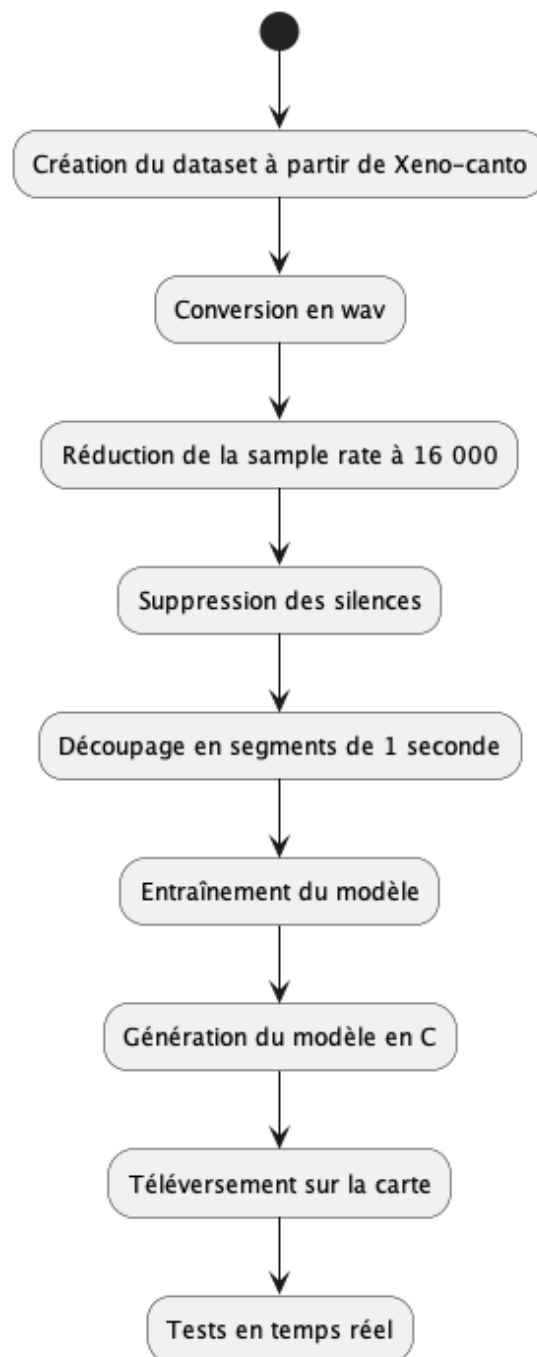


FIGURE 2.2 : Workflow Final

2.2 Jeu de données

- Description of the workflow to solve the problem described before (from training to embedded AI and communication). Add a figure illustrating that flow.
- Description of the CNN model obtained after training by taking inspiration on the first labs
- Presentation of the processing architecture of the sensor's data
- o Provide a definition of DMA and ADC

Jeu de données : Espèces d'oiseaux			
Nom Français	Nom Anglais	Espèce	Nombre d'échantillons
Alouette des champs	Eurasian skylark	Alauda arvensis	3 381
Bruant jaune	Yellowhammer	Emberiza citrinella	3 278
Bruant zizi	Cirl bunting	Emberiza cirlus	3 305
Coucou gris	Common cuckoo	Cuculus canorus	3 323
Effraie des clochers	Barn owl	Tyto alba	3 254
Faucon crécerelle	Common kestrel	Falco tinnunculus	3 311
Fauvette des jardins	Garden warbler	Sylvia borin	3 398
Gobemouche gris	Spotted flycatcher	Muscicapa striata	3 319
Grèbe castagneux	Little grebe	Tachybaptus ruficollis	3 167
Hirondelle de fenêtre	Common house martin	Delichon urbicum	3 369

2.3 Description du CNN

Input : 16000x1
Maxpool : 10x1 (output : $1600 \times n$)
conv1d [80/4, 32]
Maxpool : 4x1 (output : $100 \times n$)
conv1d [3, 32]
Maxpool : 4x1 (output : $25 \times n$)
conv1d [3, 32]
Maxpool : 4x1 (output : $6 \times n$)
conv1d [3, 32]
Maxpool : 3x1 (output : $1 \times n$)

2.4 Architecture du traitement de données par le capteur

3 — Résultats

- Presentation of the results obtained, and the experimental setup to test the generalization of the training to real data (not the ones of the dataset).
 - Analysis of the result of your project on the following criteria
 - o Performance (accuracy on train, test and on real data after quantization)
 - o Memory footprint
 - o Latency
 - o Energy consumption and battery lifetime analysis (provide assumptions of your estimation)

4 — Discussion

- Conclusion and possible evolutions of the project

Bibliographie

- [1] Cian O’LUANAIGH. *The birds and the beams : Biodiversity at CERN*. 2015. URL : <https://home.cern/news/news/cern/birds-and-beams-biodiversity-cern> (visité le 22/04/2023).
- [2] CERN. *Environmental awareness : exploring CERN’s biodiversity*. 2022. URL : <https://home.cern/news/news/cern/environmental-awareness-exploring-cerns-biodiversity> (visité le 22/04/2023).
- [3] Xeno CANTO. *Bird Sound Database*. URL : <https://xeno-canto.org> (visité le 22/04/2023).
- [4] Wei DAI et al. “Very deep convolutional neural networks for raw waveforms”. In : mars 2017, p. 421-425. DOI : 10.1109/ICASSP.2017.7952190.
- [5] Sami EL KATEB AURELIA CHABANIER. *Projet Biodiversité CERN*. URL : https://github.com/SamiElkateb/embedded_ai_project.
- [6] Bala Murugan N G. *Audio Processing and Remove Silence using Python*. 2020. URL : <https://ngbala6.medium.com/audio-processing-and-remove-silence-using-python-a7fe1552007a> (visité le 22/04/2023).