

Rapport de projet



Forest's Ears

Étude bioacoustique dans la vallée de Sainte-Anne du Portzic



Introduction :

Qui sommes nous:

Nous sommes des étudiants en troisième année à l'école d'ingénieur ENIB à Brest, une école généraliste qui enseigne l'électronique, l'informatique et la mécanique.

Nous avons réalisé ce projet dans le cadre du projet scolaire IHH (Ingénieur Honnête Homme) qui nous demande d'agir pour la société en contribuant à un ou plusieurs Objectifs du Développement Durable (ODD) proposé par l'ONU.



Les membres du projet sont Antoine Guillemaud, Antonin Suzanne et Corentin Colivet.

Les objectifs du projet:

Notre projet est en lien avec la bioacoustique, c'est la science qui étudie les espèces animales grâce à des enregistrements audio de leurs environnements.

Nous avons pour ambition de développer et de fabriquer des enregistreurs audio pour enregistrer en continu le son de la forêt.

Puis de développer un algorithme pour analyser les enregistrements audio, qui va compter le nombre de cris enregistré pour chaque espèce d'oiseaux. Ce qui va nous permettre de suivre l'évolution des effectifs d'espèces que nous sommes capables d'enregistrer.

Ainsi, cette solution permet de constater une amélioration ou une diminution de la biodiversité.

Si une organisation décidait de faire une action pour la biodiversité, cette solution permettrait de mesurer l'impact de l'action et de dire si elle a été efficace ou non.

Les enregistreurs doivent être étanches pour résister à la météo, et autonome pendant 7 jours pour qu'on ait besoin de recharger les batteries et changer la carte SD une seule fois par semaine.

Nous avons fabriqué 4 enregistreurs et nous les avons placés dans la forêt de Sainte-Anne du Portzic à Brest pour faire une étude pendant 3 mois, de mars 2022 à mai 2022.



Ce que nous avons fait:

Partie enregistreurs:



Nous avons réussi à développer nos propres enregistreurs, en nous inspirant des enregistreurs open source [AudioMoth](#). Ils nous ont permis d'atteindre plus rapidement un prototype fonctionnel.

Choix du microcontrôleur:

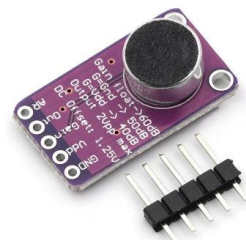
Au début, nous voulions utiliser un microcontrôleur en puce à souder nous même avec de la mémoire vive (RAM) externe à souder à côté, mais nous étions en pleins dans la pénurie de semi-conducteurs donc nous avons dû trouver une autre solution.

Et nous avons aussi beaucoup de mal à souder ce genre de composant sur les cartes électroniques, c'est pourquoi nous avons décidé d'utiliser une teensy 3.5.

Une teensy est une carte de développement qui est facile à utiliser car on peut la programmer directement en usb avec le même logiciel que arduino, mais l'avantage des teensy c'est qu'elles sont plus puissantes et qu'elles ont déjà un lecteur de carte sd intégré.

Choix des microphones:

Pour faire les premiers prototypes d'enregistreurs, on avait acheté des microphones pas chère sur AliExpress (les MAX9814), mais vu qu'on aurait eu du mal à souder des microphones MEMS et que ceux qu'on avait déjà acheté étaient de suffisamment bonne qualité, on a décidé de les garder jusqu'à la fin.



Choix de la boîte en pvc:



Pour résister à la météo, on devait faire rentrer le système dans une boîte étanche en pvc, on a mis des étiquettes pour donner un identifiant à chaque boîte et pour dire aux gens qui trouveraient une boîte de ne pas la toucher. On a aussi attaché une sangle avec un clip d'attache pour pouvoir les installer facilement sur les arbres.

Choix de l'alimentation:

Nous utilisons 2 batteries 18650 Li-ion en parallèle qui fournissent ensemble 3.7V et 5000mAh de capacité. C'est moins cher que les piles AA rechargeables qu'utilise AudioMoth, mais c'est plus dangereux et instable. Pour protéger les batteries il faut utiliser un BMS (battery management system), il empêche que la tension des batteries tombe trop bas et qu'un court-circuit ne tire trop de courant.



Nous nous sommes rendu compte que le système n'avait pas assez d'autonomie, nous avons donc décidé de mettre l'appareil en veille pendant la nuit de 23h à 7h du matin, pour pouvoir tenir pendant 7 jours.

Nous aurions aussi pu ajouter 2 batteries pour augmenter la capacité mais nous nous en sommes rendu compte un peu tard et nous devons rapidement déployer les enregistreurs dans la forêt pour commencer l'étude donc on a préféré la première solution.

Choix du stockage de données:

Le microcontrôleur enregistre le son en 16bit à 40khz, donc une semaine d'enregistrement représente 48Go de données, nous avons donc utilisé des cartes micro SD de 64Go pour stocker l'audio dans la boîte.

Une fois par semaine, on vidait les cartes sd dans un disque dur capable de stocker toutes les données de l'étude.

Sur les 3 mois d'étude imaginé au départ, nous n'avons gardé que les 2 derniers mois (avril et mai) car en mars nous n'avions pas encore la version finale des enregistreurs.

Finalement, nous avons utilisé 250Go pour l'analyse des données.

Disponible en open source:

Nous allons mettre à disposition le plan de la carte électronique et le code dans le microcontrôleur ainsi que le code de l'algorithme pour analyser les données.

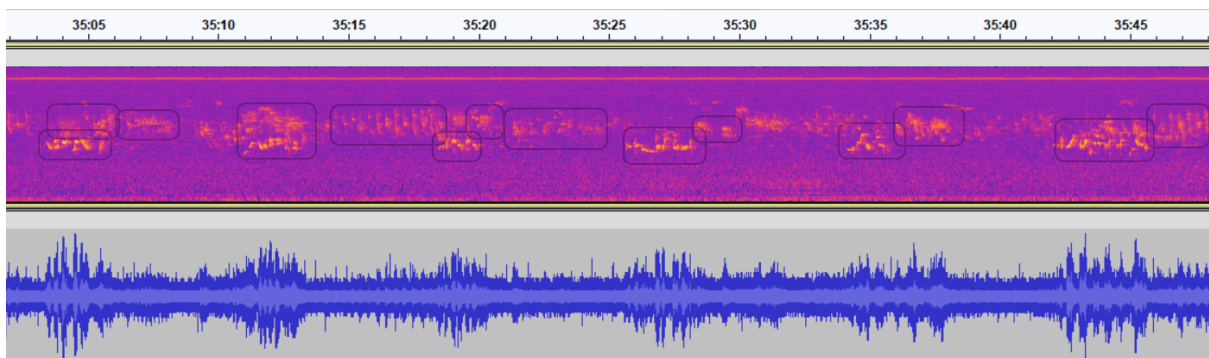
Vous pouvez consulter ou réutiliser ces informations pour vous inspirer si vous voulez reproduire un projet similaire.

le lien du github est : <https://github.com/AntoineGuillemaud/Forest-s-Ears---Bioacoustic-project>

Partie analyse des données:

Nous n'avons malheureusement pas eu le temps de finir tous les algorithmes pour l'analyse des données mais nous allons quand même vous expliquer ce qu'on avait prévu.

On voulait pouvoir compter le nombre de cris pour chaque espèce animale (surtout des oiseaux) en l'espace d'une semaine dans le but de suivre l'évolution de l'activité des espèces au fil du temps.



Résumé du fonctionnement de l'algorithme d'analyse des données :

1. On découpe les fichiers audio en petit bout de 2 secondes
2. On les transforme en spectrogramme (la partie supérieure de l'image au dessus)
3. On les passe dans un auto encodeur convolutionnel pour obtenir leurs coordonnées dans un espace à 64 dimension qui représente l'ensemble des cris d'animaux possible.
4. On applique un algorithme de clustering pour identifier les groupes de points proches qui représentent des espèces animales
5. Pour compter le nombre de cris d'une espèce, on compte le nombre de points dans le groupe de points qui correspond à l'espèce.

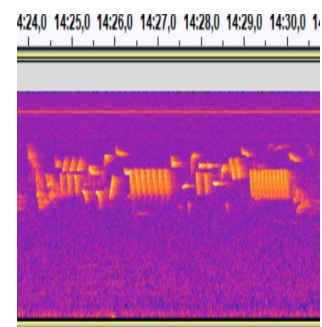
Nous allons maintenant expliquer avec plus de détails les différentes étapes de l'algorithme:

Étape 1 : produire les spectrogrammes :

Les spectrogrammes sont des représentations fréquentielles du signal sonore.

L'axe horizontal représente le temps, il s'écoule de la gauche vers la droite.

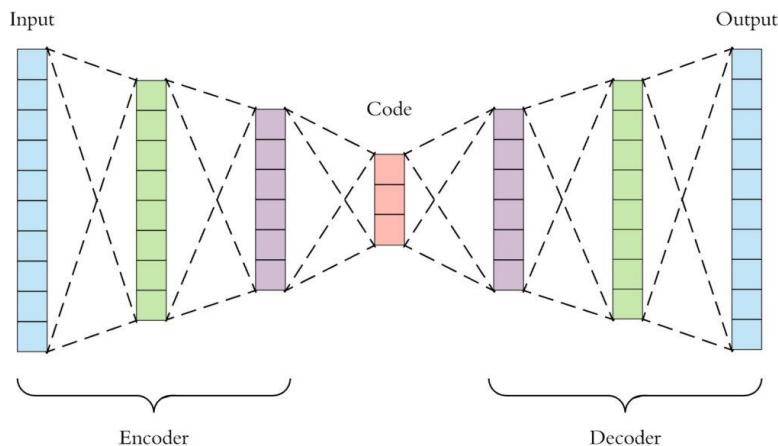
Et l'axe vertical représente les fréquences, en bas c'est les basses fréquences et en haut c'est les hautes.



On prend donc les fichiers WAVE de 1 heure et on les découpe en petit bout de 2 secondes pour en faire des spectrogrammes de résolution 32x32. (2s = 1 spectrogramme)

Etape 2 : passer les spectrogrammes dans l'auto encodeur

Un auto encodeur est un réseau de neurones qui permet d'extraire les caractéristiques fondamentales de la donnée qui est mise en entrée du réseau, qui la différencie de toutes les autres données possible du dataset d'entraînement qui est utilisé pour entraîner le réseau de neurones.



Pour entraîner l'ia, on lui demande de rendre en sortie le même spectrogramme qu'en entrée, il faut lui montrer un grand nombre de spectrogramme en entrée pour qu'elle soit capable d'extraire des caractéristiques abstraites (tel que l'espèce de l'oiseau). Nous l'avons entraîné sur 200Go d'audio soit 1 300 000 spectrogrammes.

On peut voir sur le graphique que au centre du réseau de neurones, il y a une couche qui a moins de neurone que les autres, le réseau doit donc trouver un moyen d'extraire les informations essentielles du spectrogramme en entrée (c'est l'encodeur) pour pouvoir reconstruire le plus fidèlement possible le spectrogramme en sortie (c'est le décodeur). Les informations essentielles d'un spectrogramme sont donc disponibles sur la couche du milieu. Dans notre cas la couche possède 64 neurones qui peuvent prendre la valeur d'un nombre réel de -1 à 1 en fonction de l'entrée, donc quand on met en entrée un spectrogramme, on obtient sur la couche du milieu 64 valeurs qu'on représente par un point qui a 64 coordonnées pouvant aller de -1 à 1. Donc quand on fait passer un spectrogramme dans l'auto encodeur, on obtient un point dans un repère en 64 dimensions.

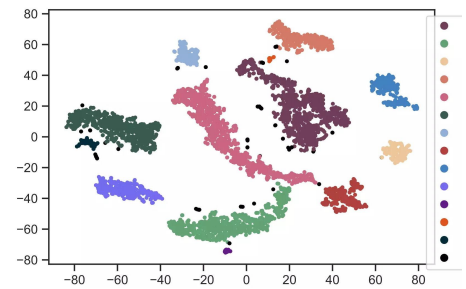
C'est donc un apprentissage sans supervision, on n'a pas besoin de dire à l'IA quelle espèce correspond à ce spectrogramme pour l'entraînement, mais on devra le faire plus tard pour compter le nombre de cris.

C'est ici que nous nous sommes arrêtés, entraîner l'ia prend du temps, et nous n'avons pas pu implémenter les étapes suivantes.

Etape 4 : former les groupes grâce à un algorithme de clustering

Une fois qu'on a fait passer tous les spectrogrammes de la semaine dans l'auto encodeur, on obtient un grand nombre de points dans un repère en 64 dimensions.

On applique ensuite un algorithme de clustering pour former des groupes de points, un groupe correspond à une espèce, et le nombre de points dans ce groupe correspond au nombre de cris enregistrés pendant la semaine.



(exemple clustering en 2 dimension)

On aurait aussi pu implémenter un algorithme de Tsne qui permet de visualiser les points sur une image en 2 dimensions sans altérer les groupes de points, pour qu'un humain puisse visualiser toute l'activité de la semaine en une image.

Conclusion:

Nous sommes fiers de ce que nous avons fait en un an, nous avons réussi à enregistrer pendant de nombreuses semaines le son de la forêt grâce à des enregistreurs que nous avons fait nous même.

Nous avons commencé à développer un algorithme pour analyser les données qui semble pertinent, même si nous n'avons pas eu le temps de le finir.

Les enregistreurs ne sont peut être pas aussi professionnels que les AudioMoth mais nous avons beaucoup appris en les concevant.

Nous espérons que ce rapport et les documents que nous vous mettons à disposition puissent vous inspirer si vous prévoyez de reproduire un projet similaire.