Preliminary Results

Project Title: Cross-Species Translation
Team Members: Ian Gifford, Deborah Sinishaw, Sara Yang
GitHub Repository: MAIS202_Final_Project
Assigned TPM: Emma

1. Project Statement

Animal communication is a complex system that humans have yet to fully understand. Some species may exhibit structured vocalization patterns, but interpreting meaning through structure remains a challenge. This project aims to advance one step closer by developing a **Cross-Species Translation Model**, which will use machine learning techniques to analyze classified animal vocalizations and generate human-readable interpretations of new animal vocalizations.

2. Data Preprocessing

Confirmed dataset: https://github.com/earthspecies/library/tree/main

This dataset is preprocessed and has labels for each audio recording. Here is the list of the ones we plan to use:

- Dogs
  - 693 recordings of dog barks from a sample of 10 adult dogs
  - Each recording is labeled with the context (3 possible scenarios) and dog name (10 adult dogs)
  - Each recording is also labeled with age, weight, sex, and breed
  - The dataset from Earth Species is already preprocessed and given in the following format

	filename	name	context	age	weight	sex	breed
0	Mac-3-A-3.aif	Mac	aggression	5	34	male	German shorthair pointer
1	Mac-3-P-3.aif	Mac	play	5	34	male	German shorthair pointer
2	Mac-2-P-2d.aif	Mac	play	5	34	male	German shorthair pointer
3	Mac-2-P-2b.aif	Mac	play	5	34	male	German shorthair pointer
4	Mac-2-A-2a..aif	Mac	aggression	5	34	male	German shorthair pointer

- Bird Songs
- Egyptian Fruit Bat
- Macaques

- Orcas
- Giant otters
- Zebra Finch

We will create a model for only the dog barks that will predict the numerous labels associated with each recording. We will also start with a single label. For now, we will attempt to determine the breed. Then, once we have a working model, we will train additional models using the same technique for each label and species, adjusting the data preprocessing and hyperparameters as we proceed.

For the dog dataset we will further preprocess the data for our model by:

1. Convert the audio files into spectrograms and removing the frequencies that do not contain any significant information. This will increase the efficiency of our model.
2. Shorten the length of audio to some standard length, as not all audio files are the same length (ranging from 1s to 180s). There are multiple ways we could go about this.
    a. Shorten each recording to a standard length
    b. Segment specific sounds (barks) either manually or automatically
    c. Slice all the audio files into chunks of a certain length

    We will start by choosing option (a), as it is the easiest for now, but in the future we will see if slicing the audio or segmenting the barks will result in a better model.
3. Split the data into training, validation, and test sets.

3. Machine learning model

We will use a CNN model. CNN models are used for spatially structured data. In addition, CNNs are often used to identify patterns in sounds by finding patterns within the spectrograms. An important property of CNNs is that they are translation invariant, meaning that they can recognize patterns, regardless of their positions. This is especially useful when looking at audio of animals, as dog barks, for example, will not all be positioned the same in each audiofile without manual adjusting. This can save us time, since manually segmenting specific sounds becomes not as important.

Other benefits of using a CNN is that they are robust against background noise and variation, and we can change the number of layers in our model to determine how complex the patterns are based on species. Fewer layers can detect simple patterns, while more layers can capture more complex features. This can give us insight into the complexity of each animal’s vocalizations.

We plan to implement our model using the PyTorch deep learning library.

4. **Preliminary results**

https://colab.research.google.com/drive/1xDdsUJEwpXX-bnXZEkap4YXRnPKJrEs0?usp=sharing
So far we have not been able to complete our model.

5. **Next steps**

Up to this point, we have only explored the dataset without diving into the actual model development. The next steps involve building the model and fine-tuning it to align with the goals of our final project.

Here is what we would like to do for the next deliverable:

1. **Train the Model** – Implement training using the dataset, adjusting parameters and optimizing performance.
2. **Evaluate Performance** – Assess the model using metrics such as accuracy, precision, recall, or loss functions.
3. **Fine-Tune Hyperparameters** – Adjust learning rates, epochs, and other parameters to improve accuracy and efficiency.
4. **Repeat 4 - 5** – We will keep on fine-tuning to get a better result.