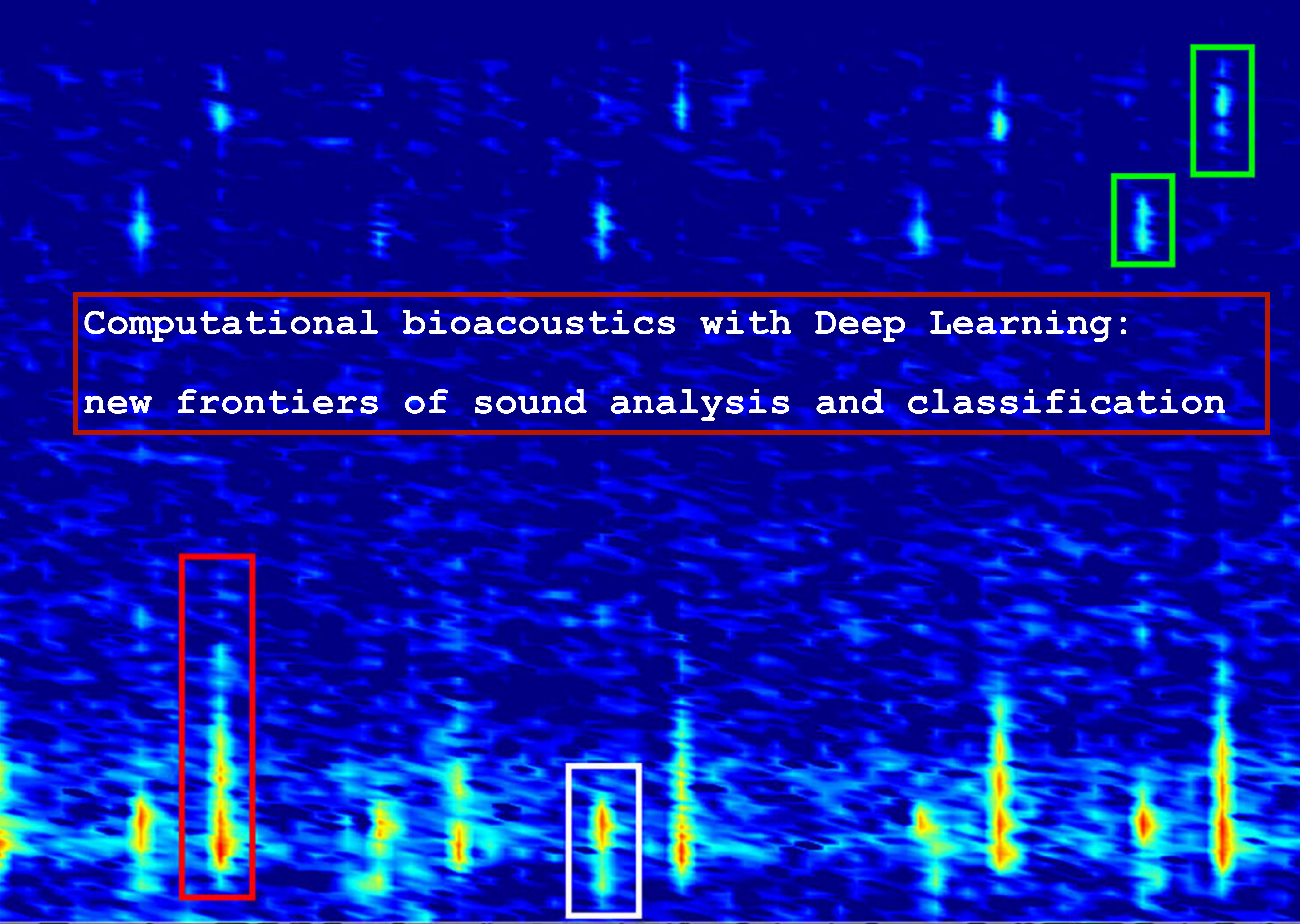
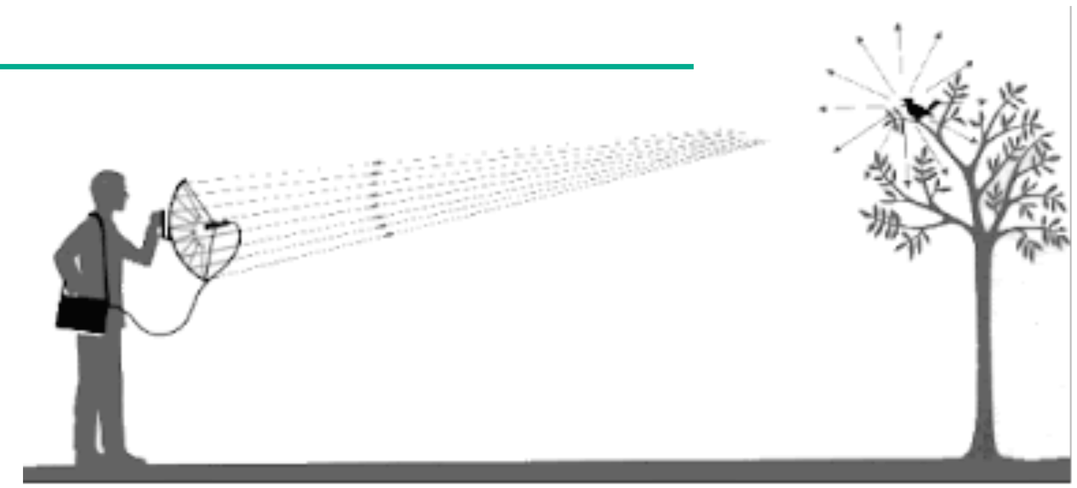
Flavia Grenga
20th February 2025

Computational bioacoustics with Deep Learning: new frontiers of sound analysis and classification



Introduction



Computational bioacoustics with deep learning: a review and roadmap

Dan Stowell[1,2]

[1] Department of Cognitive Science and Artificial Intelligence, Tilburg University, Tilburg, The Netherlands
[2] Naturalis Biodiversity Center, Leiden, The Netherlands

Bioacoustics is the study of animal sounds

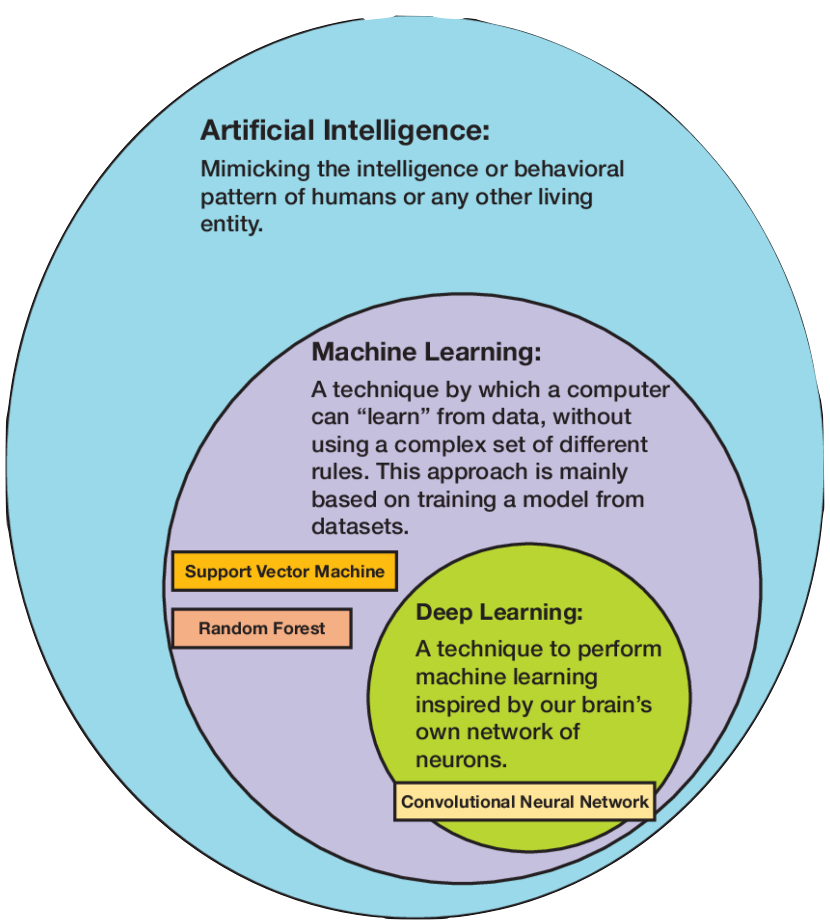
↓

huge datasets to analyze

↓

Methods mainly used so far:

- signal processing
- data mining
- AI → machine learning → **Deep learning**



Deep learning →

10 yrs of application in audio task but recent in bioacustic

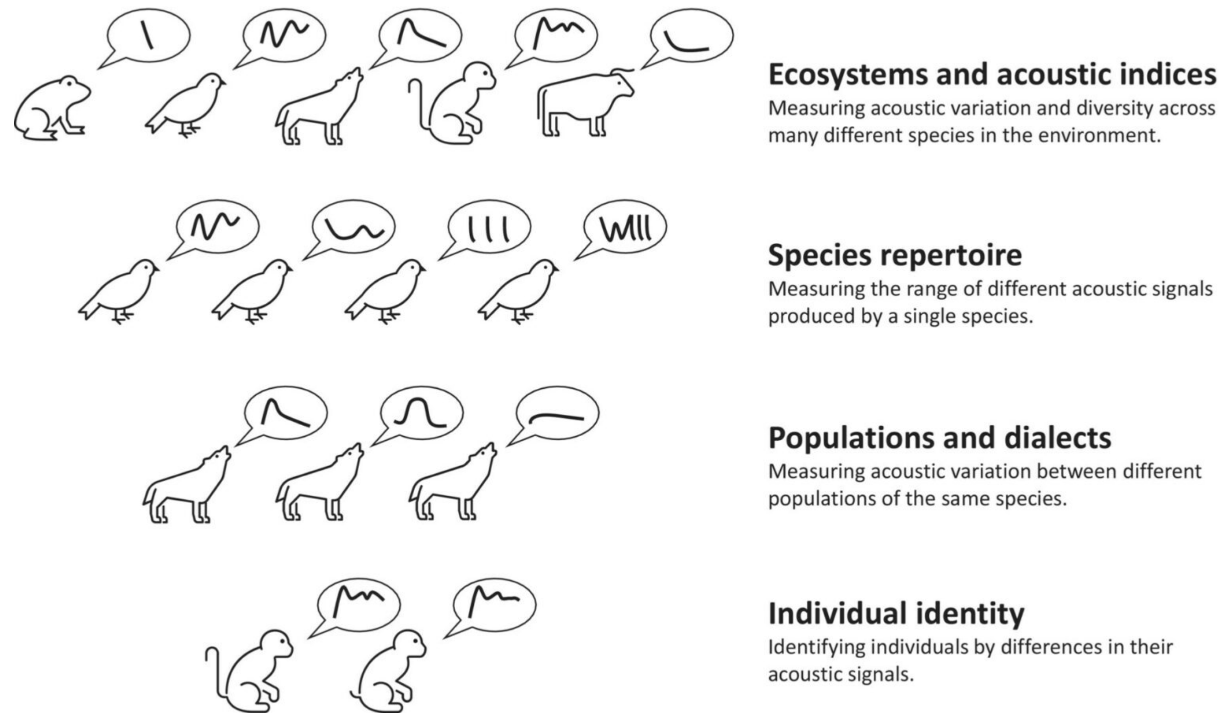
classification main use in computational bioacustic

flexible can be applied to several tasks:
- Classification
- Regression
- Signal enhancement
- Synthesis of new data

Taxonomic coverage

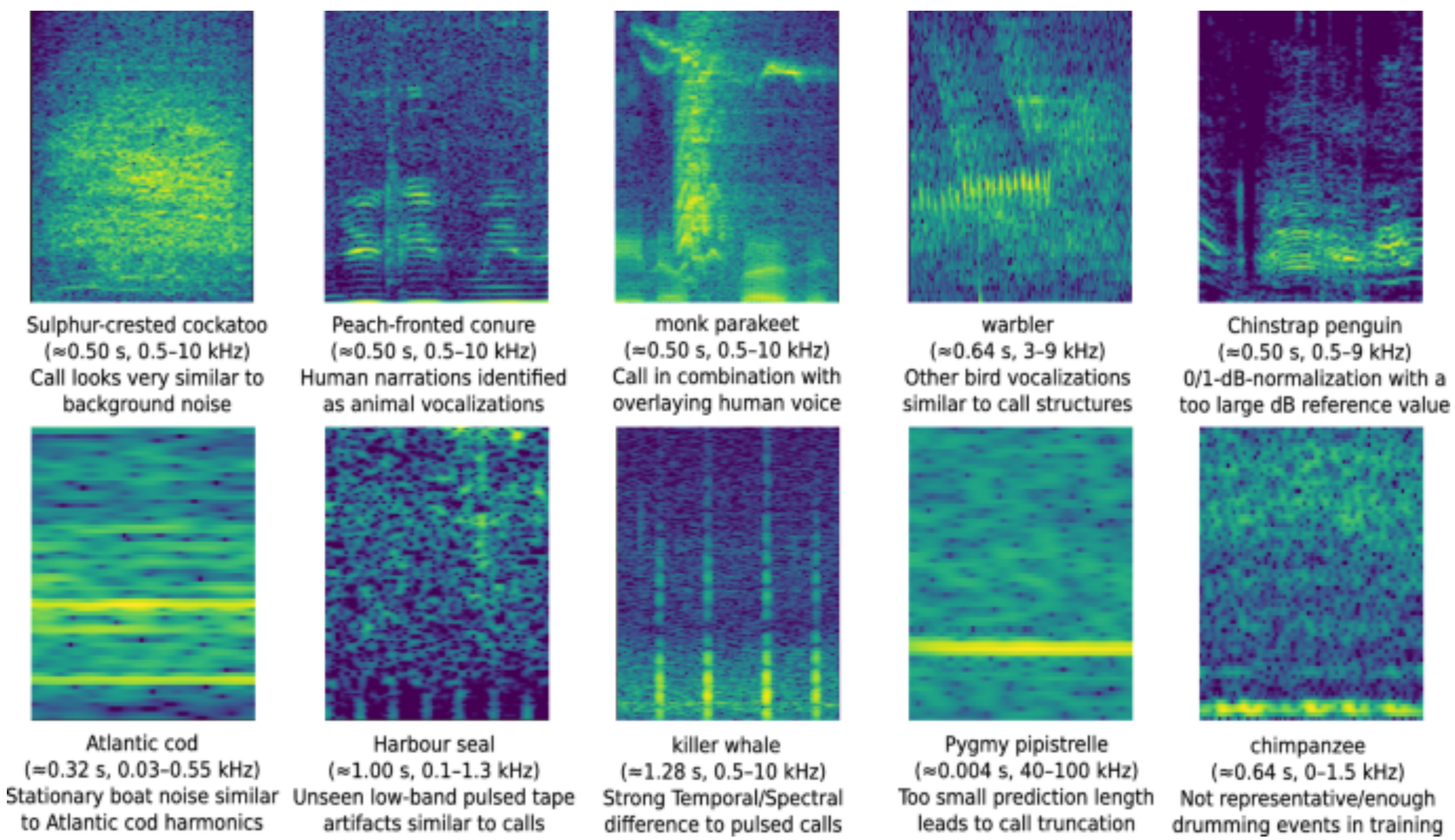
Species/taxa whose vocalisations have been analysed through DL include:

- 🐦‍⬛Birds—the most commonly studied group
- 🐳Cetaceans and other marine mammals
- 🦇Bats
- 🐘Terrestrial mammals (excluding bats): including primates
- 🐁Mouse and rat ultrasonic vocalisations (USVs)
- 🐸Anurans
- 🪲Insects



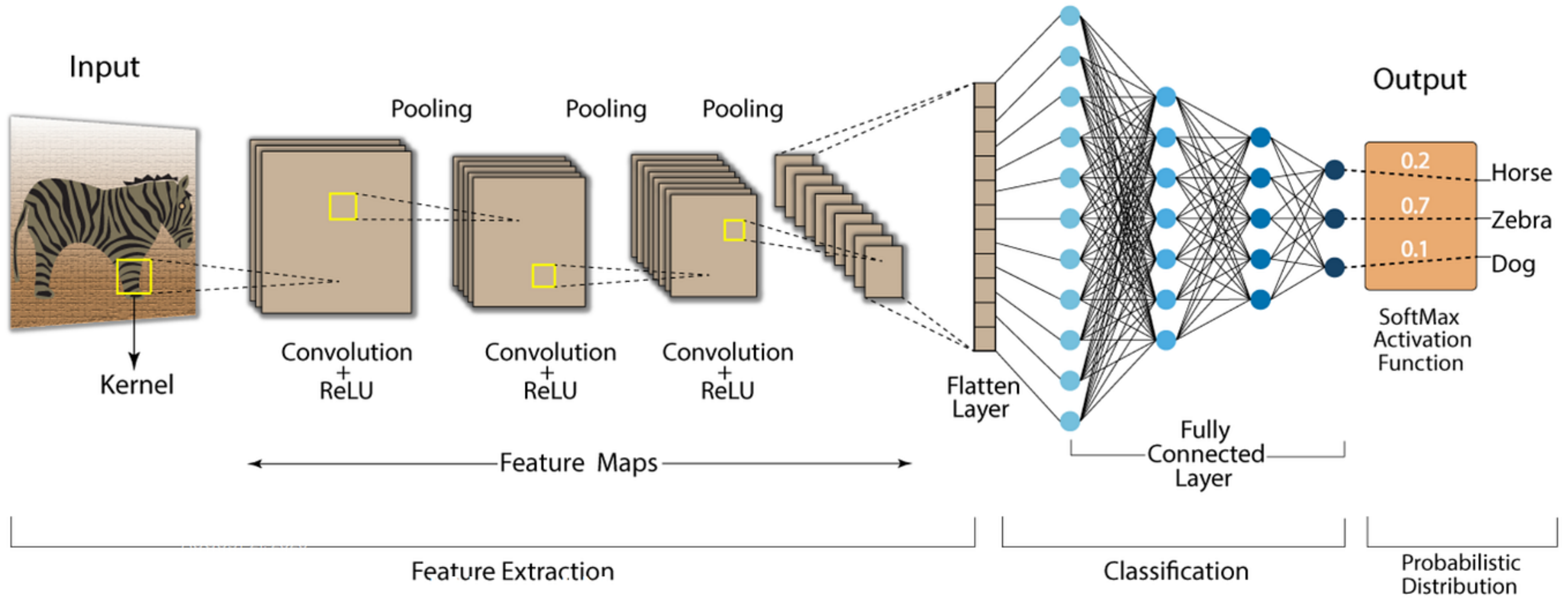
Kershenbaum, Arik, et al. "Automatic detection for bioacoustic research: a practical guide from and for biologists and computer scientists."

Bergler, Christian, et al. "ANIMAL-SPOT enables animal-independent signal detection and classification using deep learning."



Convolutional Neural Network (CNN):

- dominate the field, often pre-trained on generic datasets (e.g., AudioSet)
- specialized in processing data structured in grids (such as images or audio spectrograms)
- uses convolutional filters to automatically extract relevant features



<u>CNNs work well for Bioacoustics:</u>

✓**Detects Frequency Patterns:** finds species-specific features (e.g., bird chirps, bat echolocation).

✓**Handles Background Noise:** learns to ignore irrelevant sound patterns.

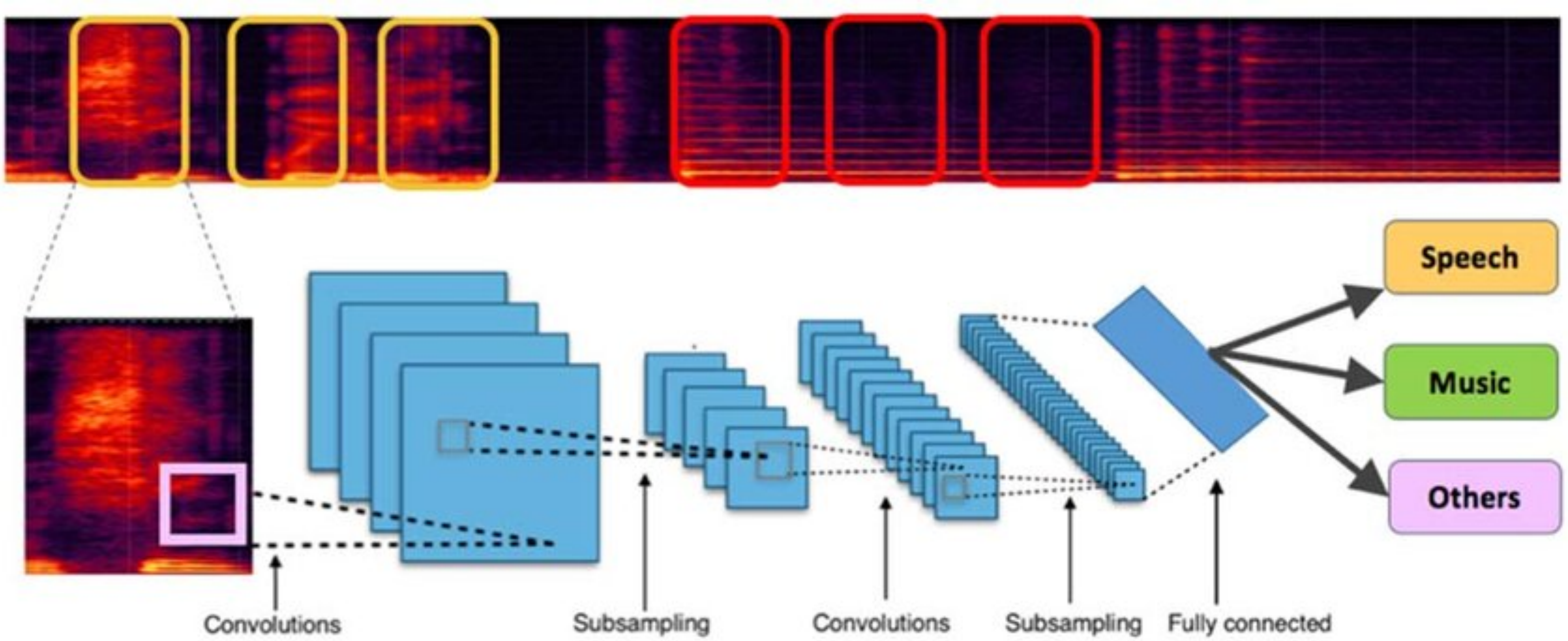
✓**Works with Large Datasets:** learns from thousands of spectrograms efficiently.

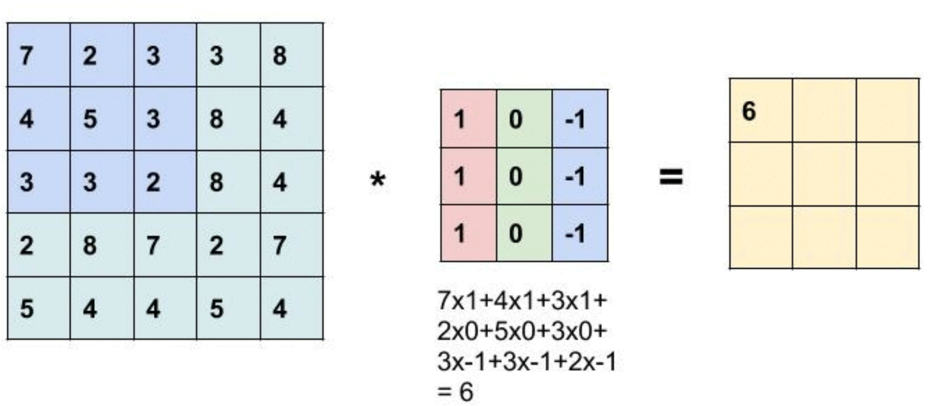
✓**Pretrained Models Help:** CNNs pretrained on large audio datasets can be fine-tuned for bioacoustics.

Why CNNs for audio?

Patterns in sound (like bird calls or whale songs) **appear in frequency-time representations** (spectrograms).
CNNs are great at detecting **localized patterns**, such as harmonics or frequency shifts in spectrograms.
Unlike traditional machine learning, CNNs **automatically learn features** without needing manual engineering.

Step	What Happens?	Purpose
1. Convert to Spectrogram	Audio is converted to a time-frequency representation	Makes sound interpretable for CNNs
2. Convolutional Layers	Small filters "**kernels**" (e.g., 3×3 or 5×5 grids) detect patterns in spectrograms -> each filter produces a **feature map**,	Identify pitch, rhythms, harmonics
3. Pooling Layers	Reduce dimensions while keeping important features **Max pooling**: keeps most important values (e.g., strongest frequency peaks **Average pooling**: Averages nearby values	Improve efficiency, prevent overfitting
4. Fully Connected Layer	Extracted features are mapped to output classes. **Pooled feature maps** are **flattened** into a 1D vector which is fed into a fully connected neural network that makes the final classification decision.	Make predictions (species, call types)
5. Output Layer	**Outputs probabilities** for different classes: Softmax for multi-class problems Sigmoid (for binary classification)	Classify the sound (i.e. Sparrow 85% Robin 10 Crow 5%





Kernels filters

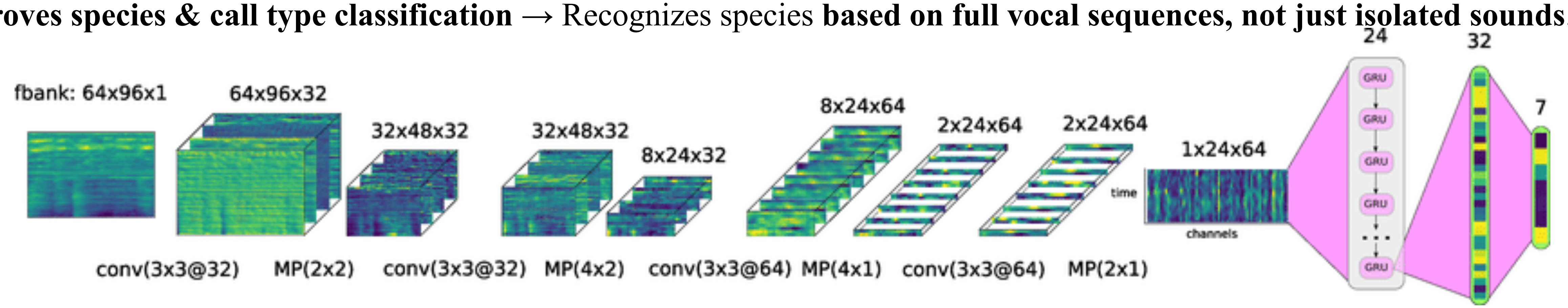
Convolutional-Recurrent Neural Networks (CRNN):

- analyze the full sequence and not just the patterns.
- good for tasks like detecting repeated sounds, tracking changes in pitch over time, and recognizing call sequences (e.g., bird songs, whale calls).
- combine **CNNs with LSTM/GRU** units to model temporal sequences

Step	Process	Purpose	Example in Bioacoustics
1. Convert Audio	Convert **raw waveform → spectrogram**	Makes data CNN-compatible by converting it into a **2D image-like representation**	Transforming a **birdsong waveform** into a spectrogram to visualize its pitch changes
2. CNN Feature Extraction	CNN applies **convolutional filters** to scan local frequency-time patterns in the spectrogram	Detects **harmonics, syllables, textures, and patterns**	Recognizing **harmonic structures in whale calls** or **sharp pulses in bat echolocation**
3. Pooling Layers	**Reduces dimensions** while keeping important spectral features	Improves **efficiency, reduces overfitting**	Keeping only the **most relevant frequency bands** in frog calls
4. Flattening & Time Distribution	Converts **CNN feature maps into sequences** (so they can be processed by RNN layers)	Converts spectrogram features into a format that **tracks time evolution**	Preparing features for analyzing the **temporal order of bird song syllables**
5. RNN Temporal Processing	RNN (LSTM or GRU) processes the **sequence of extracted features**	Captures **long-term dependencies and sequential patterns** in the sound	Identifying **whether an elephant is making an alarm or social call** based on previous sounds
6. Fully Connected Layer	Dense layer maps extracted features into a **high-level representation**	Helps the model learn **species, call types, or behavior categories**	Classifying a **whale species based on its call sequences**
7. Output Layer	Final **Softmax (multi-class) or Sigmoid (binary)** layer gives the **final classification decision**	Assigns a **probability to each possible sound category**	Output: 🐦 "This is a Robin (95%)" or 🦇 "This is a Bat Echolocation Call (99%)"

<u>CRNNs Are Useful in Bioacoustics:</u>

✓ **Better at capturing temporal patterns** → Ideal for sounds with structure (songs, sequences, alarms).
✓ **Handles noise more effectively** → CNN filters out irrelevant background, RNN focuses on important sequences.
✓ **Works on varying sound durations** → Unlike CNNs, which process fixed-length spectrograms, RNNs adapt to variable durations.
✓ **Improves species & call type classification** → Recognizes species **based on full vocal sequences, not just isolated sounds**.



Convolutional recurrent neural network (CRNN) architecture. The input features are matrix of consecutive frames of log-Mel filter banks (64 filter banks by 96 time frames). The convolutions and max-poling operations are sequentially applied to extract beneficial features. Then these are fed into the gated recurrent unit (GRU) to capture the temporal information. The network outputs are sigmoid scores, these indicate several active acoustic events in audio signal.

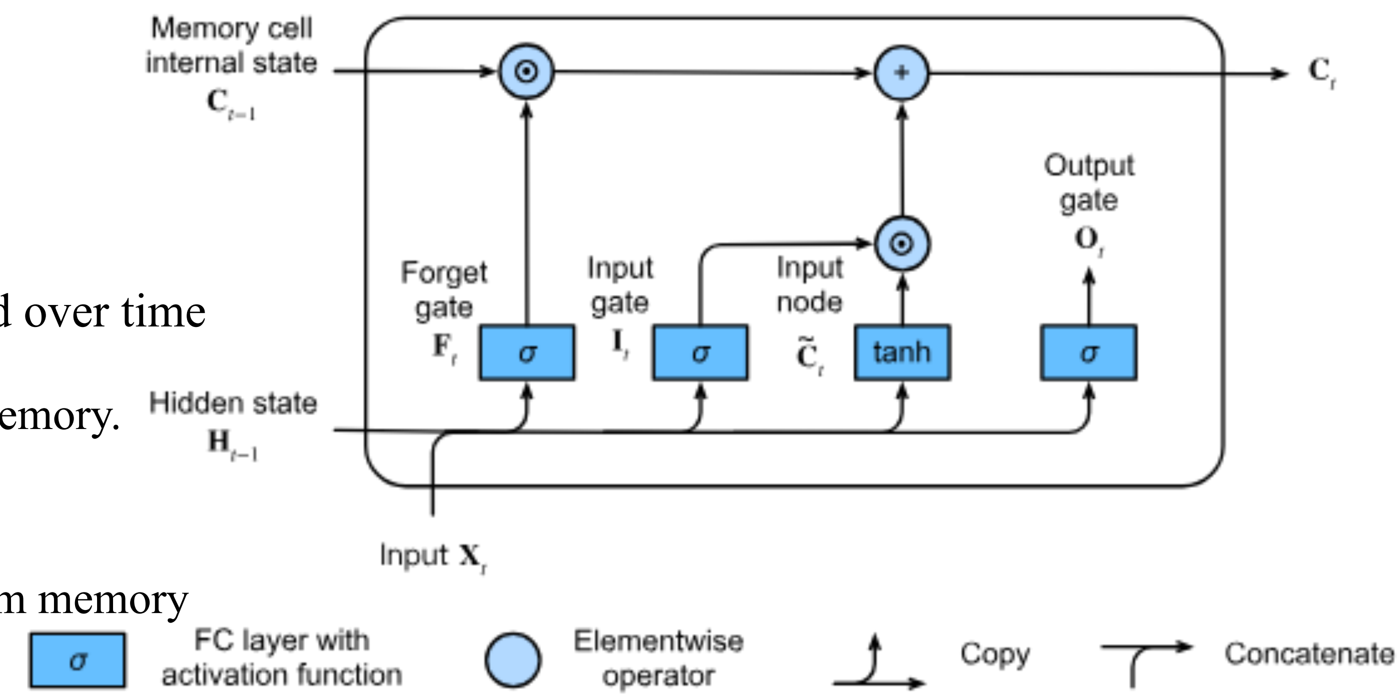
CRNN (Convolutional-Recurrent Neural Networks):
combine **CNNs with LSTM/GRU** units to model temporal sequences

LSTM (Long Short-Term Memory):

ideal for long sequences (complex bioacustic audio)

memory cells that allow information to be maintained and updated over time

gate
- **input**: decides what new information to add to memory.
- **output**: determines what to send to the output
- **forget**: determines what information to delete from memory

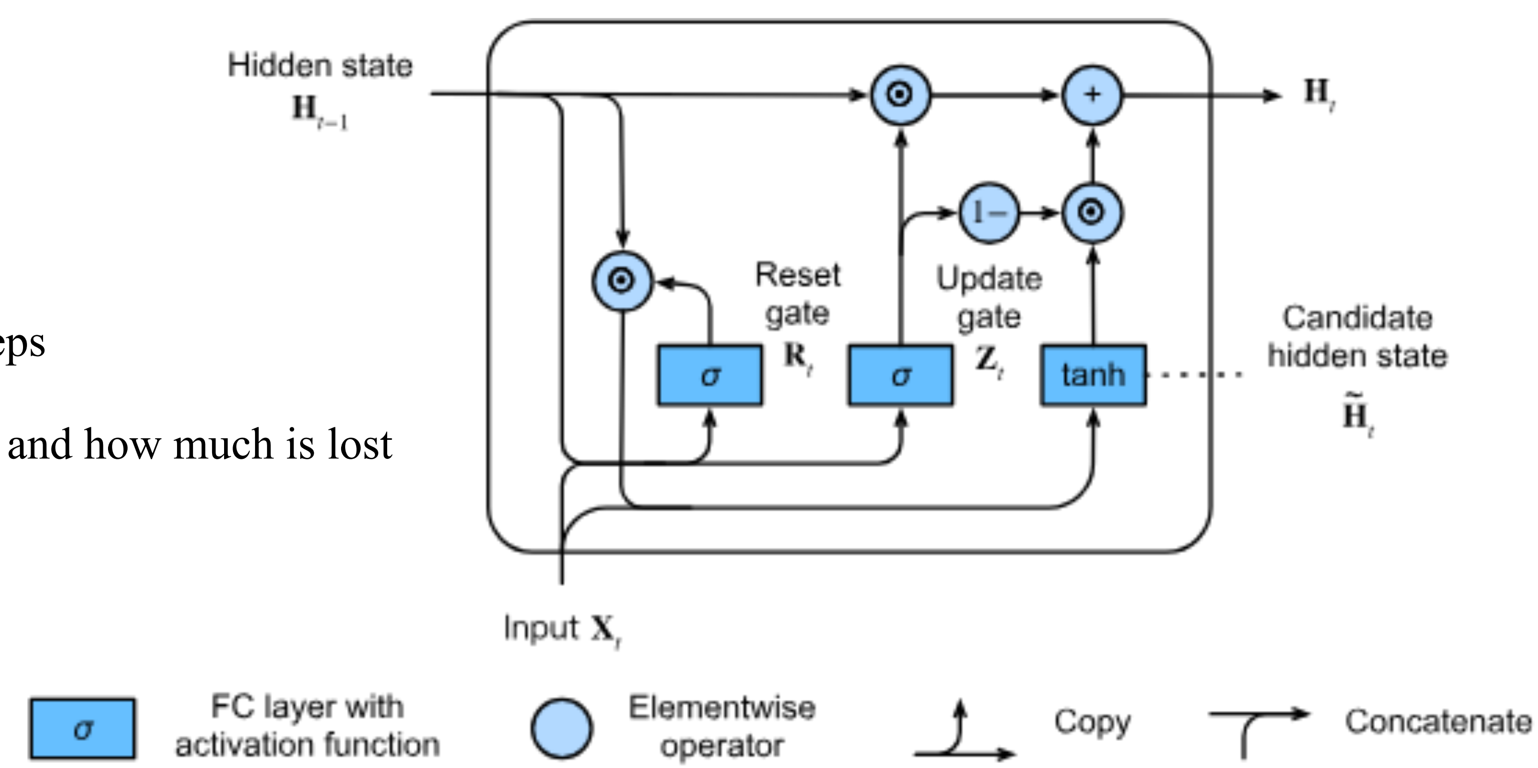


GRU (Gated Recurrent Units):

short-to-medium sequences

- lighter and faster than LSTMs
- speech recognition
- automatic translation

gate
- **update**: regulates the flow of data in time steps
- **reset**: how much previous data is transferred and how much is lost



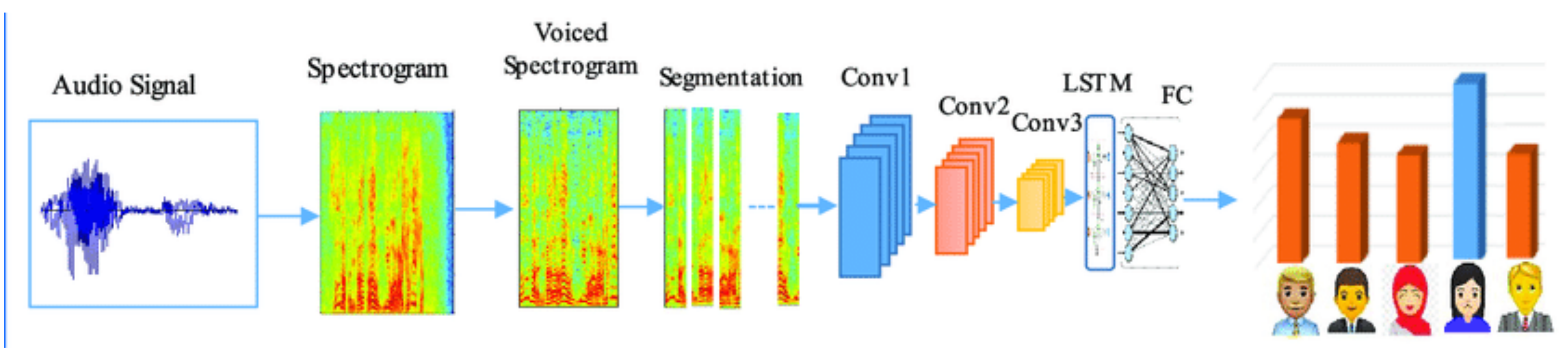
CNN *vs* CRNN

If sound patterns depend on sequences (e.g., a series of chirps or pulses), **CRNNs outperform standard CNNs**

Feature	CNN	CRNN
Strength	Detects local patterns (harmonics, syllables)	Captures long-term sound patterns (syllable sequences, call rhythms)
Best for	Species classification (single call)	Call sequence analysis (songs, communication patterns)
Weakness	Ignores long-term dependencies	Slower and more computationally intensive
Example Task	"What species made this sound?"	"Is this an alarm call or a mating call?"

Transformer networks:

- recently adopted
- processes sequences all at once
- tokenization: divide into small time-frequency chunks for efficient processing
- self-attention mechanism: focus on important parts of the input



Classification, Detection, Clustering

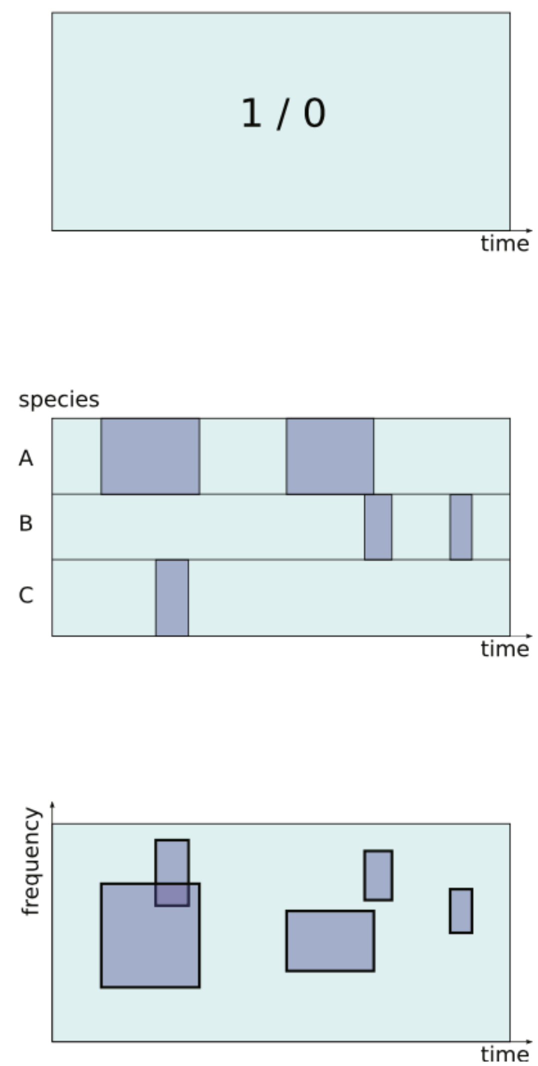
Task	What It Does	Common Models	Example in Bioacoustics
Classification	Assigns a label to a sound	CNN, CRNN, Transformers	"Is this a bird, bat, or whale?"
Detection	Finds **when** a sound occurs in a recording	Binary Classification (Occupancy Detection), SED, Object Detection	"When did the bat call happen?"
Clustering	Groups similar sounds **without labels**	K-Means, Autoencoders, Siamese Networks	"Are these frog calls from the same species?"

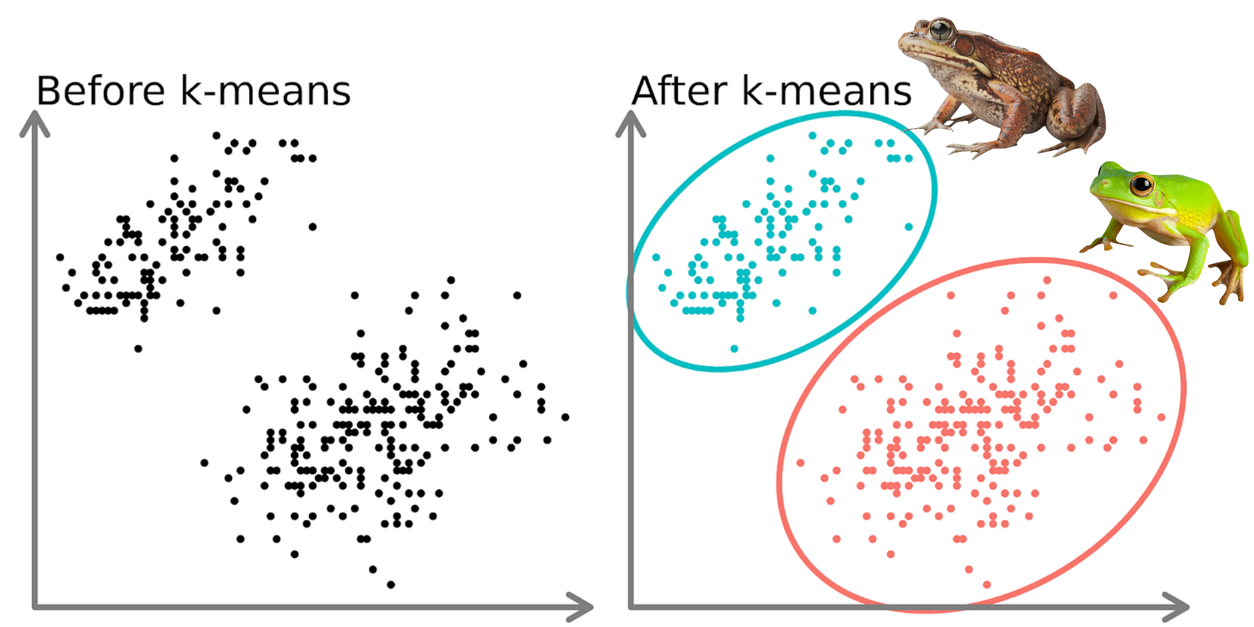
(a) Binary classification

(b) SED (multi-species)

sound event detection

(c) Object detection

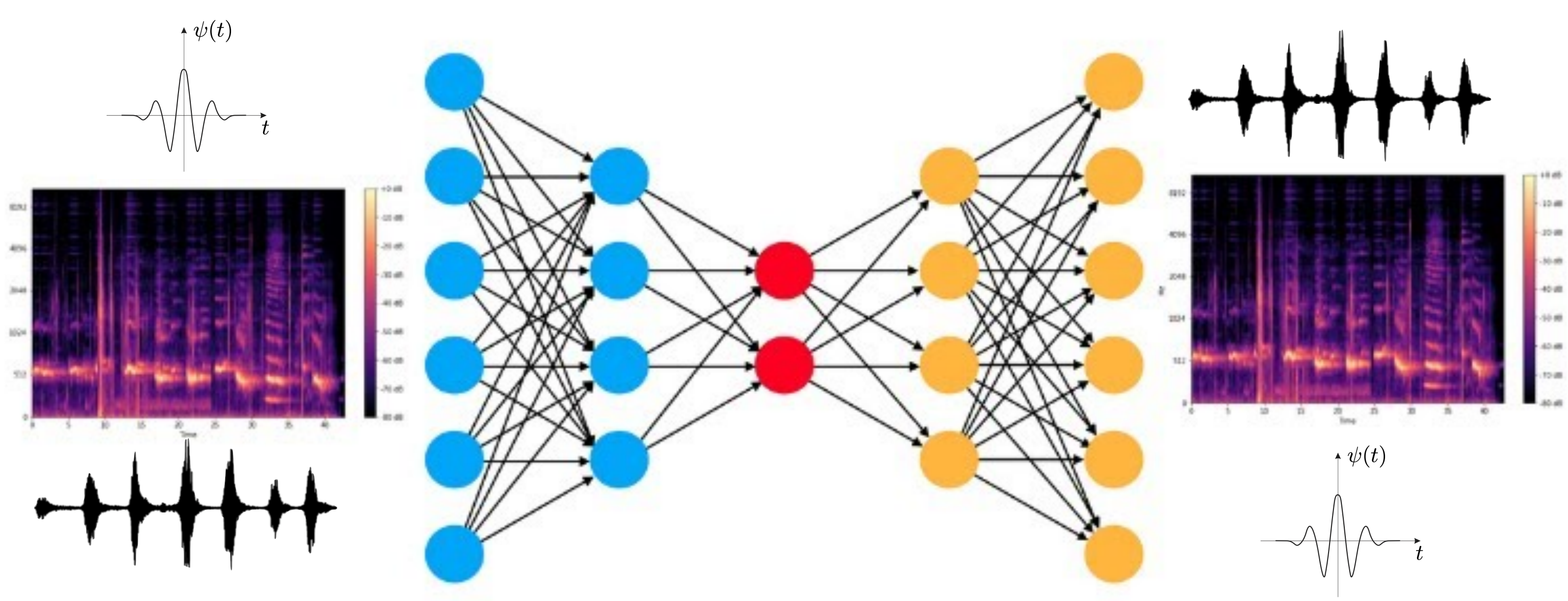


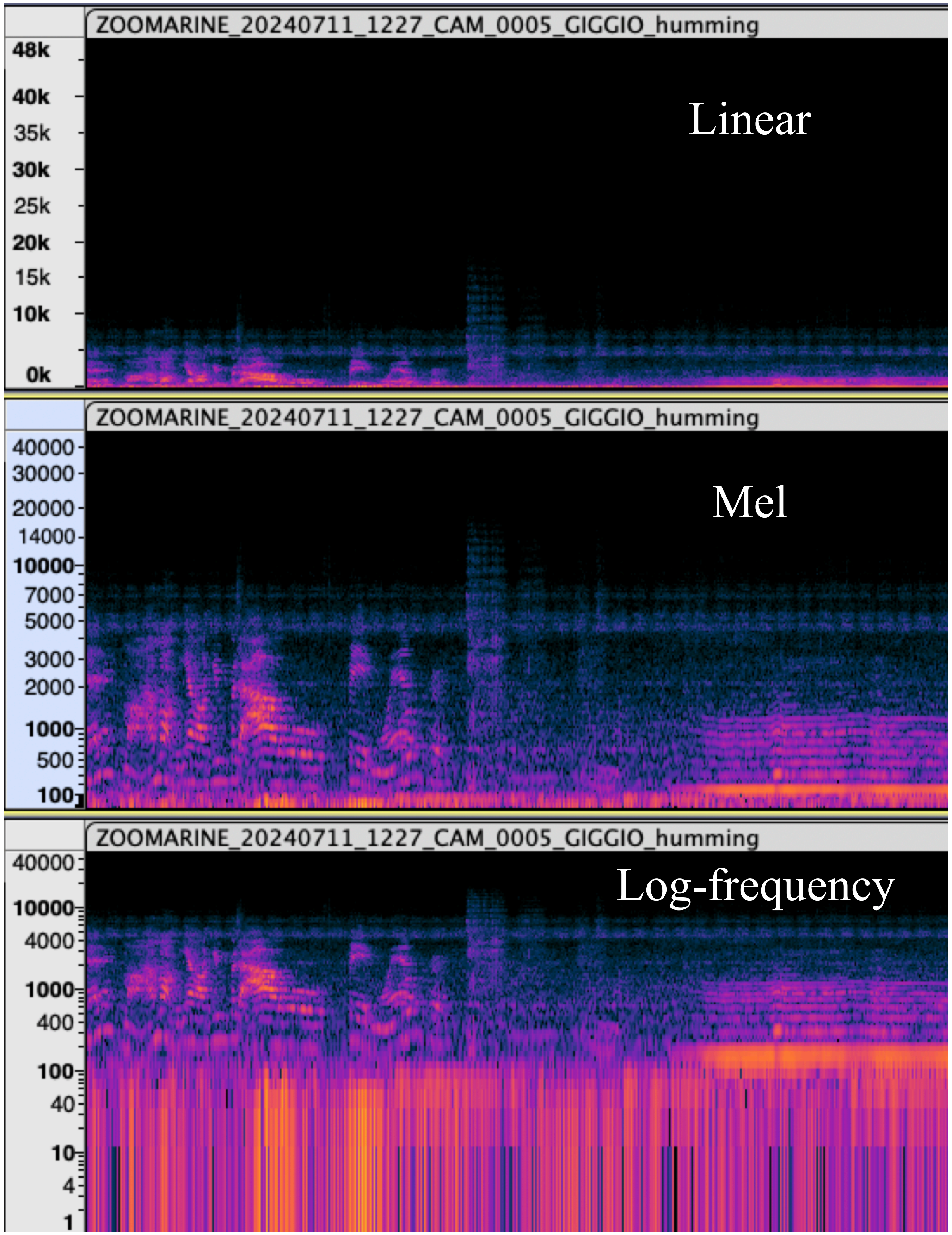


Acoustic features: spectrograms, waveforms, and more

Audio data can be represented for deep learning models but the choice of **acoustic features** greatly affects the performance of models in bioacoustics.
The most commonly used representations include:
1. **Spectrograms** → The most widely used feature, converting sound into an image-like format.
2. **Waveforms** → Raw audio input, directly used in some deep learning models.
3. **Other Representations** → Alternatives like wavelets and learnable filterbanks.

Feature Type	Description	✅Pros	❌ Cons	Best Used For
Spectrogram	2D time-frequency representation	•Works well with CNNs •Easy to interpret	•Can lose phase information •Needs manual transformation	*General bioacoustics (birds, whales, insects)*
Mel Spectrogram	Spectrogram with Mel-scale frequencies	•Mimics human hearing •Used in speech/music	May not be optimal for animals	*Speech-like animal sounds*
Log-Frequency Spectrogram	More resolution in low frequencies	Good for low-frequency sounds	Less common in DL models	*Whale calls, elephant sounds*
Raw Waveform	Direct 1D sound signal	•No transformation needed •End-to-end learning	•Requires large datasets •Computationally expensive	*Advanced DL models like WaveNet, wav2vec 2.0*
Wavelets	Captures multi-resolution features	Good for dynamic sounds	Less studied in DL	*Frog calls, complex bird songs*
Learnable Filterbanks	DL learns the best frequency filters	Optimized automatically	Needs large datasets	*Cutting-edge DL approaches (LEAF, SincNet)*





Linear spectrogram is a **time-frequency representation** of sound shows **true frequency values**

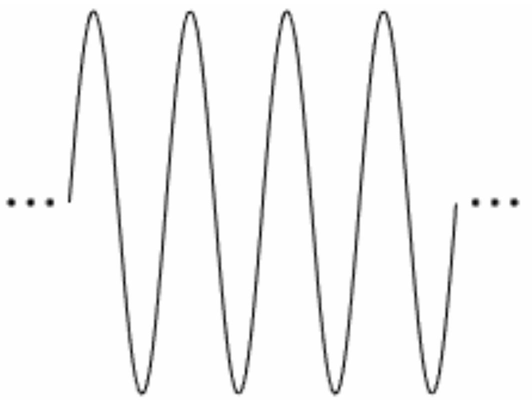
Mel Spectrogram is a spectrogram that uses the **Mel scale**, which **compresses high frequencies** while giving **more resolution to low frequencies**.

- matches human hearing perception
- common in deep learning models

A **Log-Frequency Spectrogram** is similar to a **standard spectrogram**, but the **frequency axis is logarithmic** instead of linear.

- higher resolution for low frequencies
- compresses high frequencies (like a piano keyboard, where lower notes are spaced further apart).
- it preserves fine frequency details

Acoustic features: waveforms and more



Sine Wave

Raw waveform is simply the **1D time-domain representation of sound**—how air pressure changes over time

$$F(\omega)=\int f(t)e^{-j\omega t}dt$$

Fuorier analyzes the signal in terms of frequency, loses time because of infinite sinusoids as a basis.

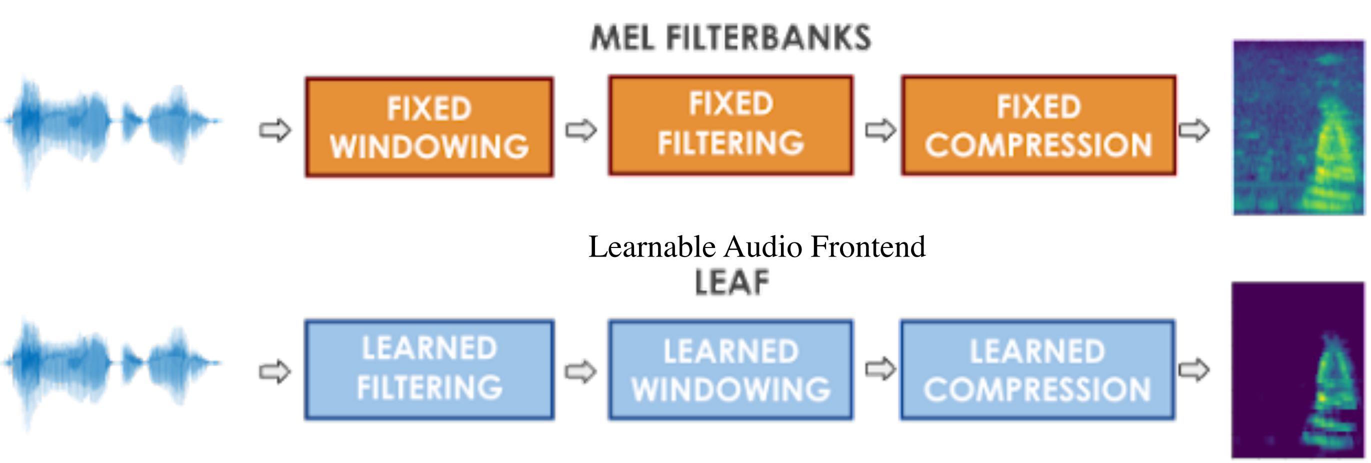
Wavelet (db10)

Wavelet transform is a technique that **decomposes audio into different time-frequency scales:**

- **better for non-stationary signals** (e.g., bird songs, dolphin clicks, bat echolocation)
- **multi-resolution analysis** (long and short sounds)

$$W(a,b)=\int f(t)\psi_{a,b}^{*}(t)dt$$

Wavelet analyzes the signal in both time and frequency with a variable scale because of localized waves with variable width.

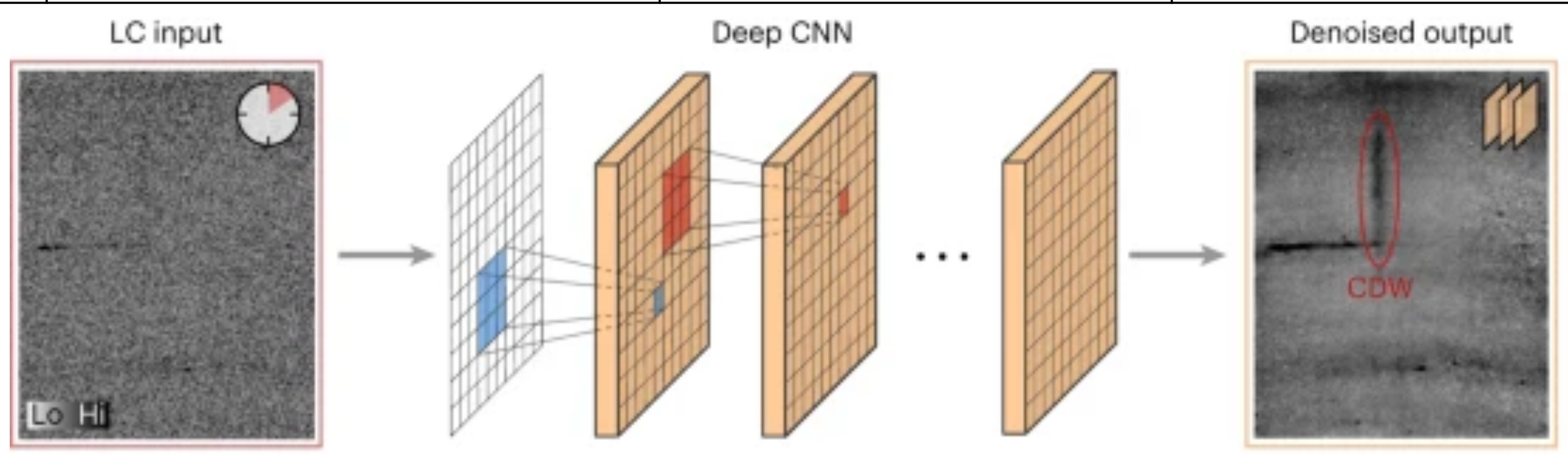


Learnable Filterbanks in deep learning models **learn the best frequency filters automatically:**

- **Self-supervised learning models** (like wav2vec 2.0, SincNet, and LEAF)
- **More efficient for deep learning** (removes the need for predefined spectrograms)

- Traditional **signal processing** in bioacoustics relied on **hand-crafted features** (e.g., spectrogram analysis, filtering etc.)
- Deep learning improves signal processing by automating noise reduction, separation, and feature extraction
- Large-scale analysis of wildlife sounds for species monitoring, conservation, and ecoacoustic studies.

Task	Problem	Deep Learning Solution	Example
Noise Reduction	Background noise affects analysis	Denoising Autoencoders, Wave-U-Net	Removing wind noise from bird calls
Sound Separation	Overlapping sounds make classification hard	Deep Clustering, U-Net	Separating multiple bird species in a forest
Feature Extraction	Hand-crafted features are limited	CNNs, Wav2Vec 2.0	Learning unique bat echolocation features
Compression	Large datasets are hard to store	Autoencoders, VAEs	Reducing storage size of whale song datasets



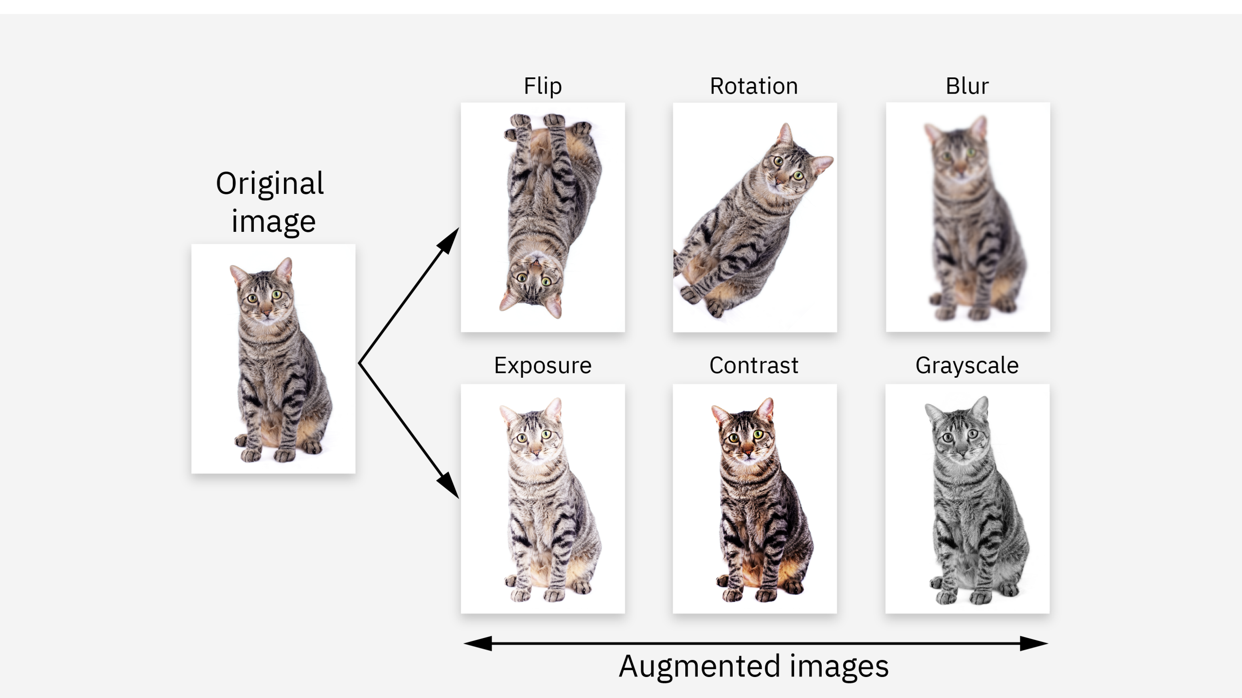
Oppliger, Jens, et al. "Weak signal extraction enabled by deep neural network denoising of diffraction data." *Nature Machine Intelligence* 6.2 (2024): 180-186.

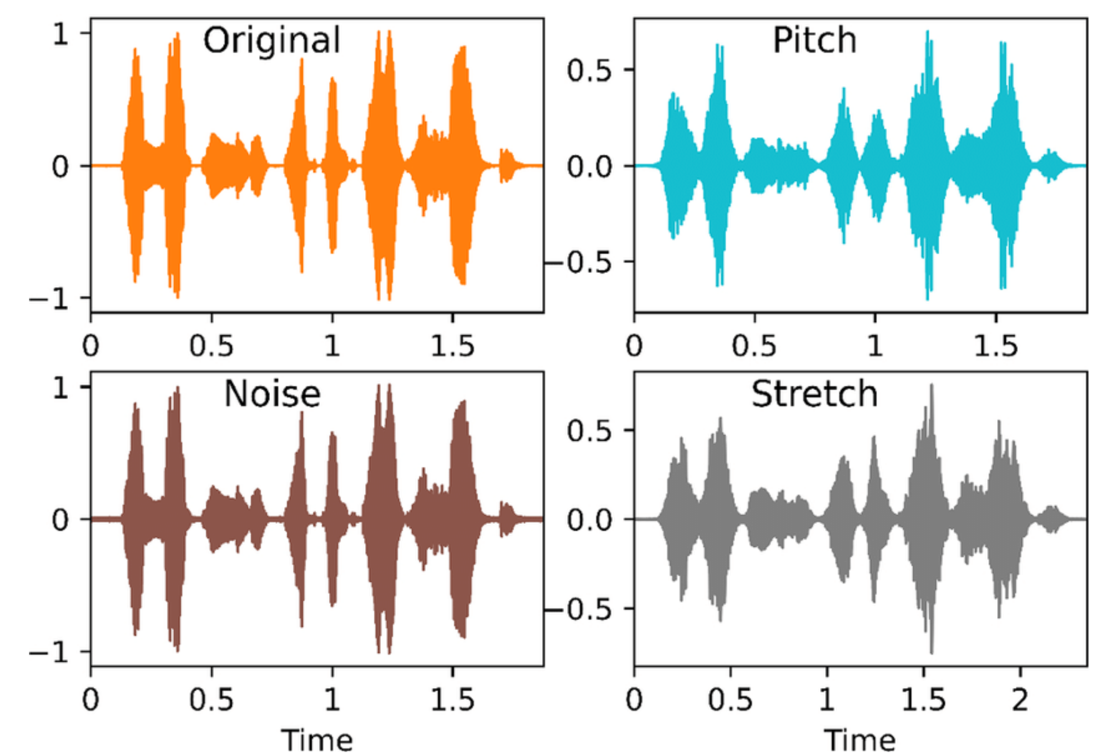
Small Data: Data Augmentation, Pre-Training, Embeddings

Bioacoustic studies usually have **small datasets** because labelling animal sounds is **time-consuming and expensive** but…Deep learning models **need a lot of data**, so they can struggle with small datasets.

To improve performance with limited data, researchers use **three key techniques**:

Method	What It Does?	How It Helps with Small Data?	Example in Bioacoustics
Data Augmentation	Creates new training samples from existing data	Expands dataset without collecting new recordings	Adding background noise to whale songs to make models more robust
Pre-Training	Uses a model trained on a large dataset	Reduces the need for lots of labeled data	Fine-tuning a speech-trained model for dolphin calls
Embeddings	Uses pre-trained feature representations	Allows small datasets to be classified with better accuracy	Using VGGish embeddings to classify bird species





Generalisation and domain shift

A model trained on one dataset **might not work well on new data from different locations, species, or recording conditions**.

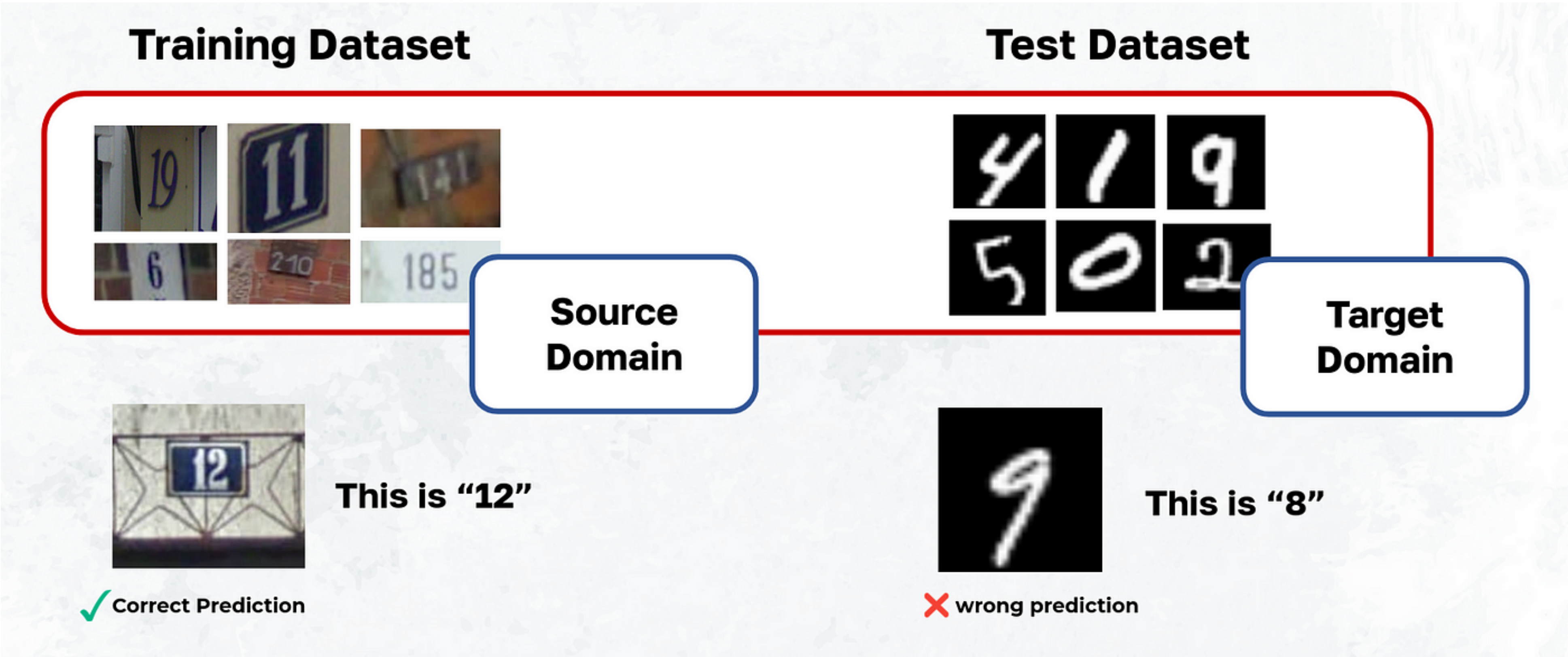
Domain shift is when a model performs well in one setting but poorly in another:
it happens when the conditions of the training data **don’t match** the conditions of the test data.

Improving **Generalization:**
the model should work well across **different conditions, species, and environments**

Generalisation refers to a model’s ability to correctly classify new, unseen data that wasn’t in the training set:
- A well-generalised model can detect bird calls in different forests, at different times of the year, or with different microphones.

How to Reduce Domain Shift and Improve Generalisation?

1. **Data Augmentation**: artificially simulate different environments in training data
2. **Transfer Learning** (Pre-Training on Large Datasets): use a pre-trained model that has already learned general sound patterns
3. **Domain Adaptation** (Fine-Tuning with New Data): adapt a model trained in one domain to work in another
4. **Active Learning** (Ask Humans to Correct Mistakes): have experts correct wrong classifications and retrain the model



Open-set and novelty

What happens when a deep learning model encounters a sound it has never heard before?

↓

Develop **open-set recognition** techniques so models can **detect novel (unknown) sounds**

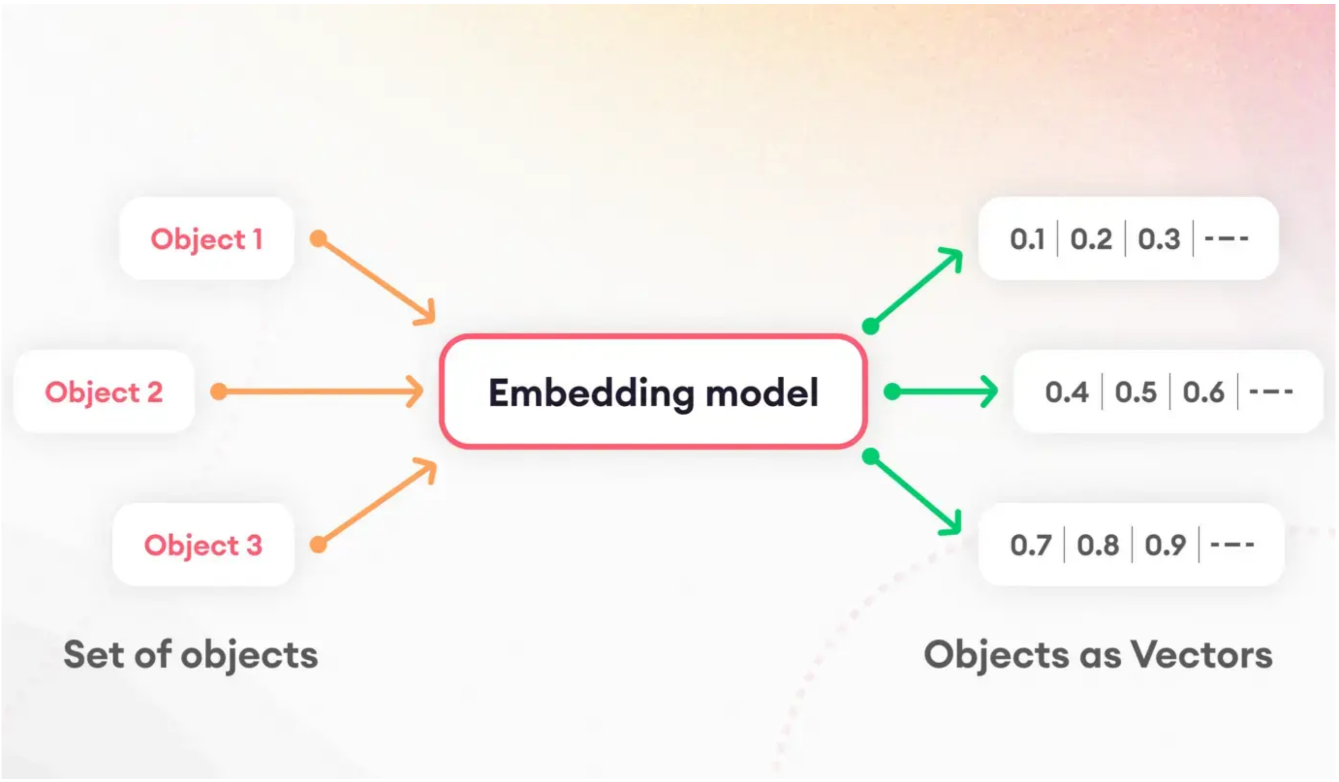
Novelty Detection is the ability of a model to **detect new, unseen sounds** without prior training on them.

instead of misclassifying an unknown sound, the model **flags it as novel** so researchers can analyze it

How do we handle Open-Set and Novelty Detection?

1. **Outlier Detection with Confidence Scores:** assign a confidence score (fin whale 90% accepted; fin whale 40% unknown)
2. **Anomaly Detection Models**:
- Autoencoders → train on known calls; if an unknown sound doesn't fit, it's flagged as **novel**.
- One-Class SVMs → learn a normal pattern, then detect anything that doesn't match
3. **Using Embeddings to Compare Similarity**: comparison to known sounds
- Siamese Networks & Triplet Loss → If a sound is too different from known species, mark it as novel
- Pre-trained Embeddings → Use embeddings to measure sound similarity

Scenario	Closed-Set Model	Open-Set Model
A model trained on 10 bird species hears a **new species**	Misclassifies it as one of the 10 known species	Flags it as **"unknown"**
Whale sound classifier detects a **new call type**	Assigns it to the closest known category	Recognizes it as a **new type of vocalization**



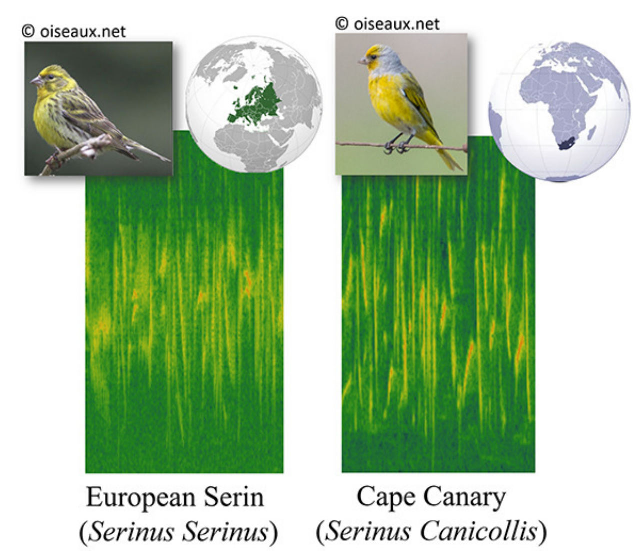
Additional information can improve deep learning models for bioacoustics!

Incorporate auxiliary (extra) data—such as time, location, and environmental conditions—alongside the acoustic features. This helps models **make better predictions** by understanding the **context of a sound event**.

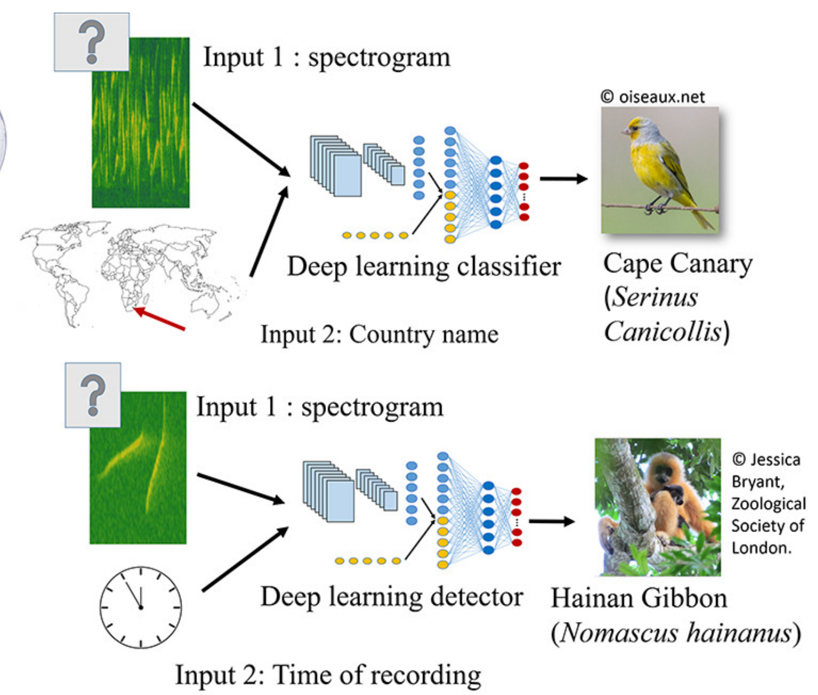
Multi-Modal Learning: combining Audio + Metadata (e.g., time, location, weather) reduces errors and misclassifications

Monitoring wildlife with acoustic recorders produces large volumes of data that can require a lot of human effort to analyse

Deep learning has transformed acoustic classification and detection, but achieving high accuracy and low false positive rates remains a significant challenge



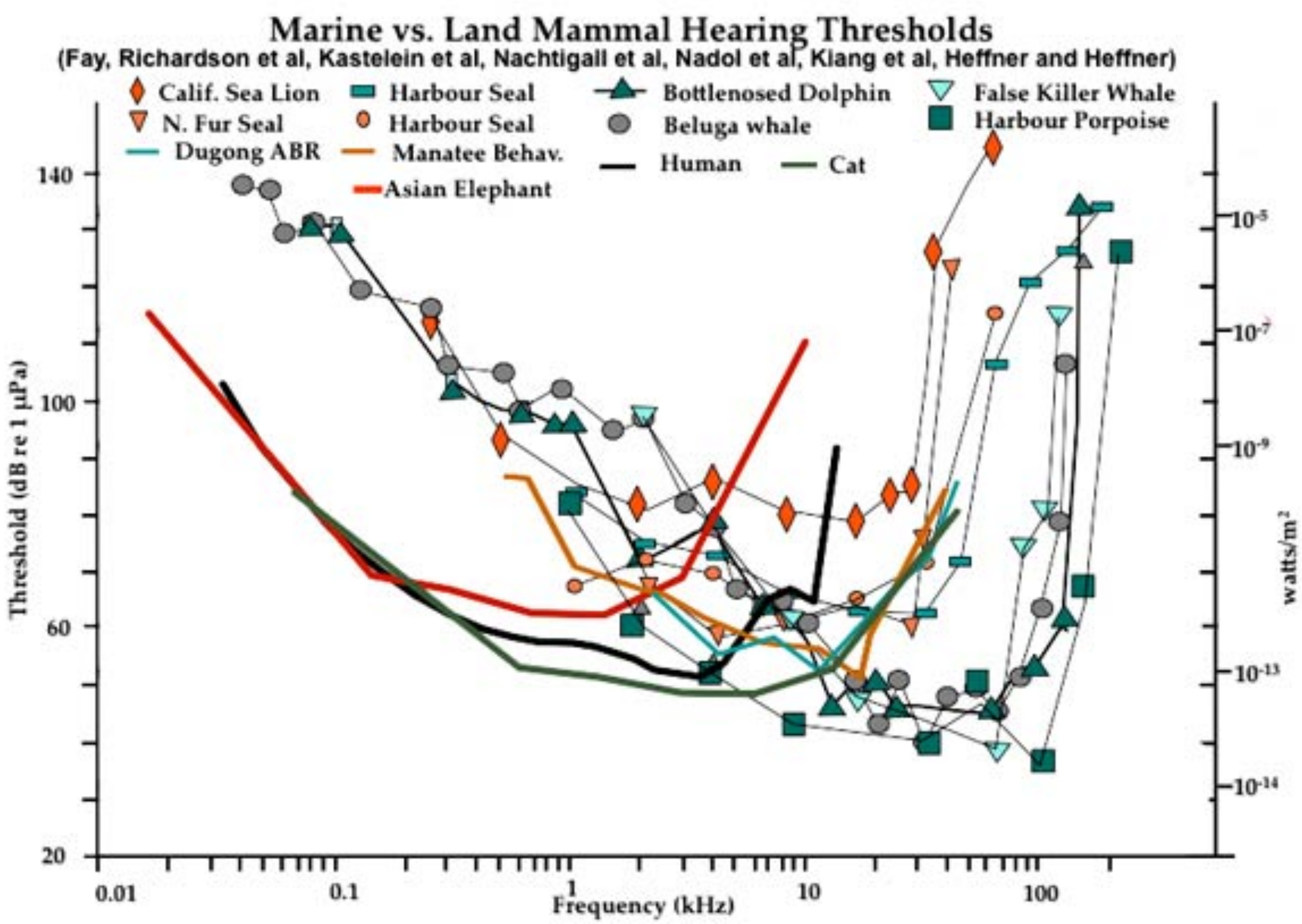
Spectrograms (visual representation of acoustic recording) can appear similar, however, we can use **metadata** such as geographical location to tell them apart



Our findings show that by incorporating **metadata** we can improve the performance of deep learning classifiers, with a range of potential applications in ecology

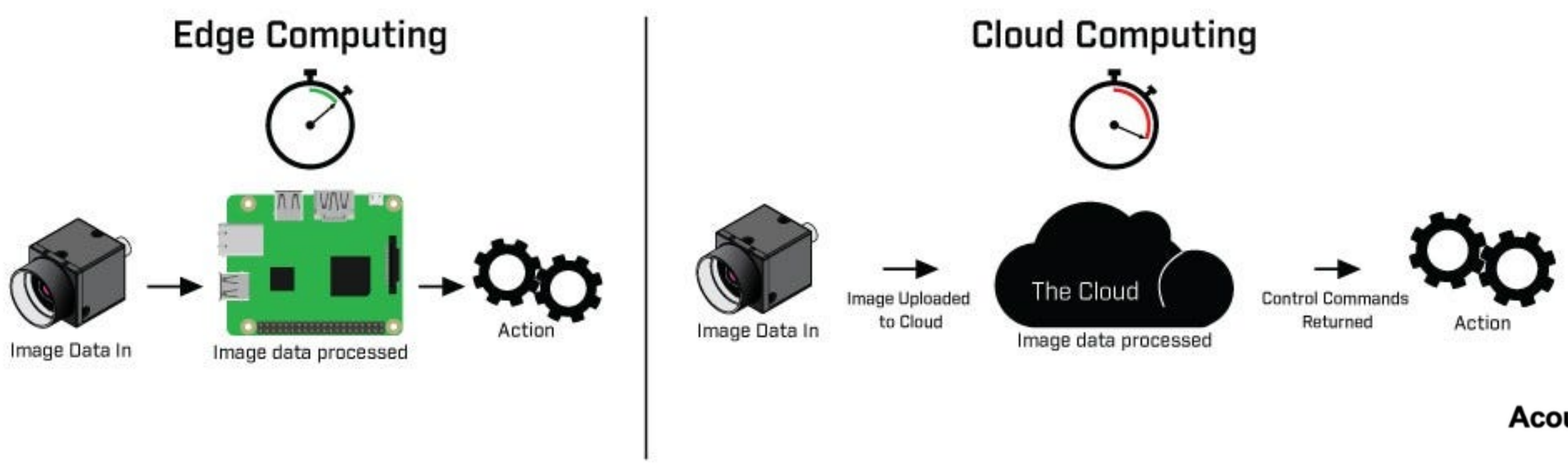
Deep learning models perceive and process sounds differently from animals and humans

Challenge	Why It's a Problem?	How to Solve It?
Deep learning models "**hear**" **differently** from animals	Models detect frequencies outside an animal's hearing range	Use **species-specific frequency filters**
Noise affects machine detection differently than animal perception	Animals can focus on meaningful signals	Use **bio-inspired auditory features (PCEN, wavelets)**
No **ecological context** in deep learning models	Animals process sound based on behavior and habitat	Combine **contextual + perceptual data** https://www.earthspecies.org/what-we-do/publications

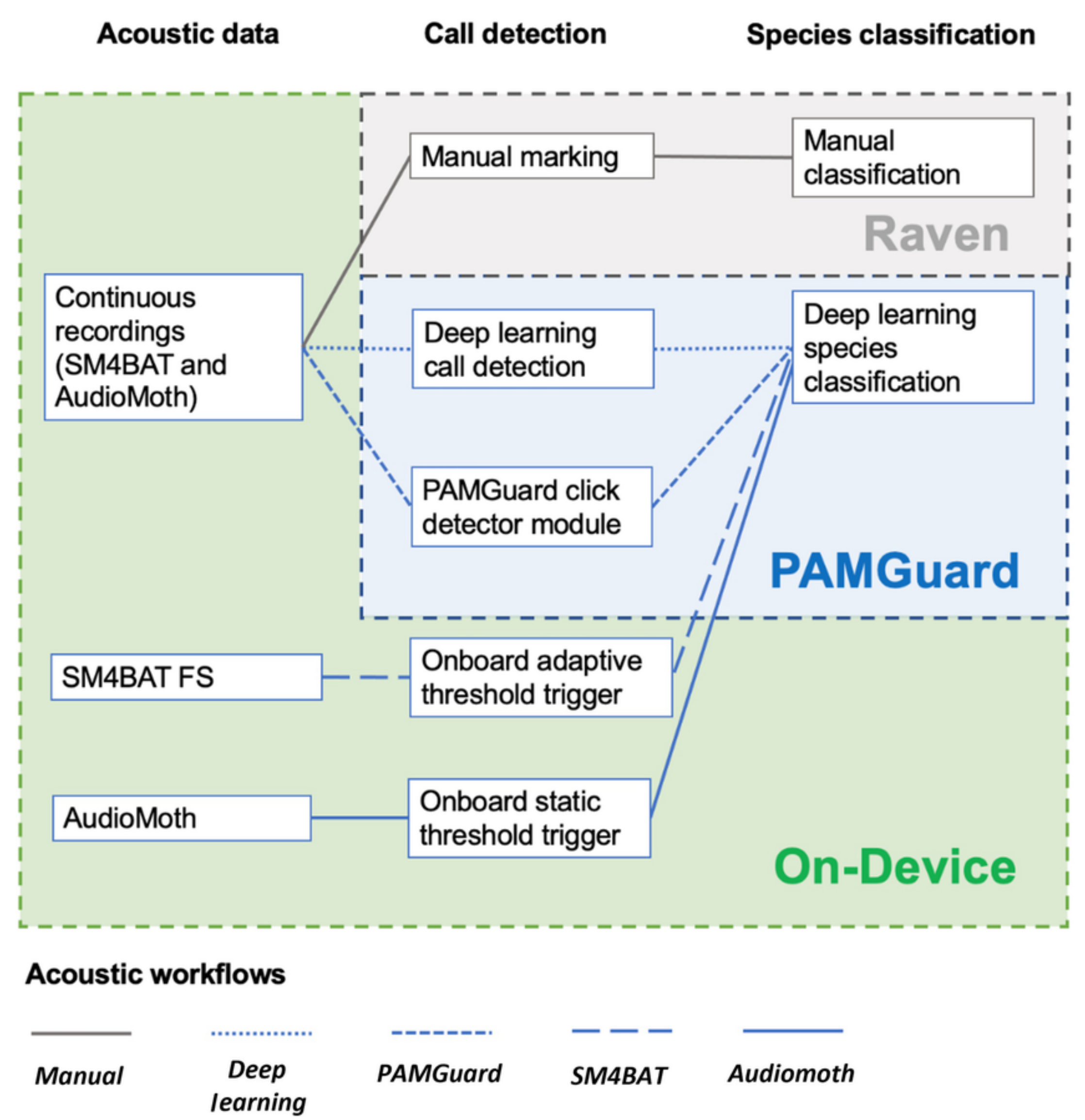


Perception and on-device DL

Instead of sending audio recordings to a cloud server for processing, the **entire deep learning model runs locally on a small, embedded device** (Raspberry Pi, AudioMoth, and NVIDIA Jetson Nano)



Feature	Benefits ✅	Challenges ❌
Real-time processing	Immediate detection of species or threats	Requires fast, optimized models
No need for internet	Works in remote environments	Cannot update models easily
Energy-efficient	Reduces power consumption	Must balance accuracy with efficiency
Cost-effective	Avoids expensive cloud processing	Limited storage and memory

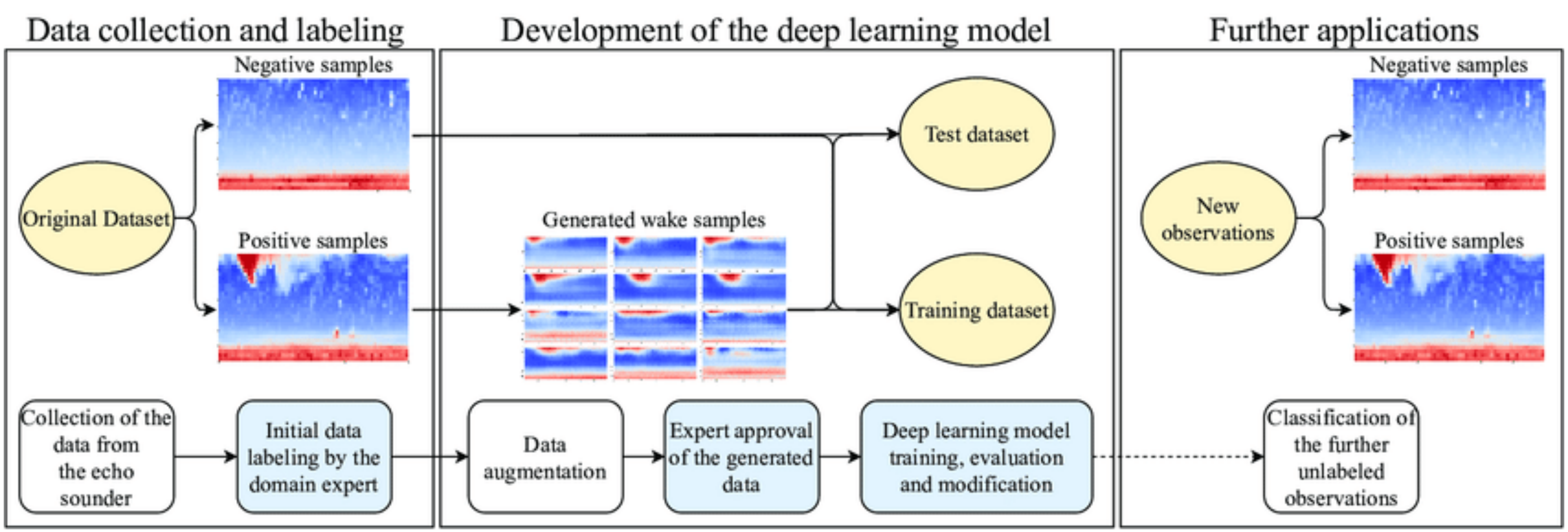


Brinkløv, Signe MM, et al. "Open-source workflow approaches to passive acoustic monitoring of bats." *Methods in Ecology and Evolution* 14.7 (2023): 1747-1763.

Workflows and other practicalities

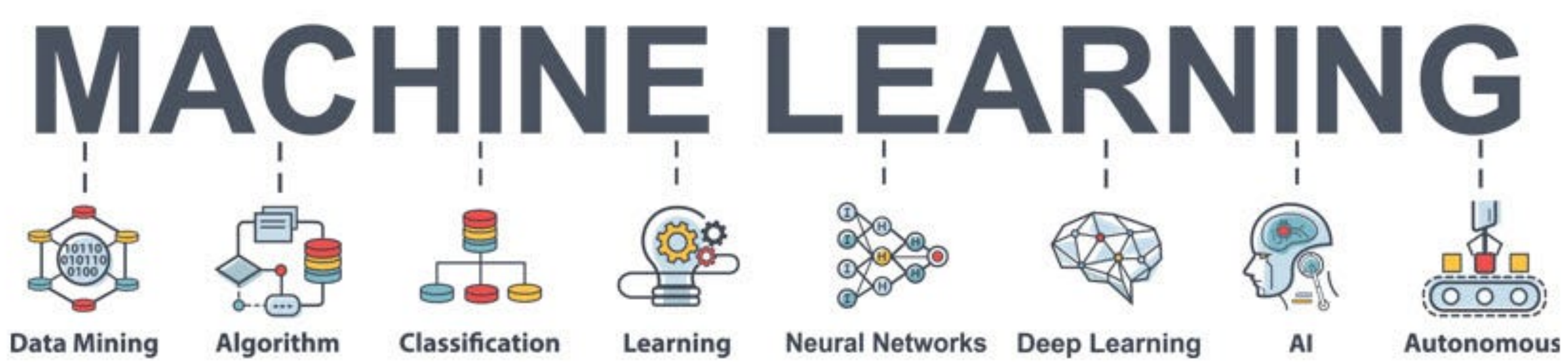
A typical **bioacoustic deep learning pipeline** consists of the following **stages**:

Stage	What Happens?	Example in Bioacoustics
1. Data Collection	Record audio in the field using microphones, hydrophones, or remote sensors	*Collecting whale songs using underwater hydrophones*
2. Data Preprocessing	Clean, filter, and prepare the audio for training	*Removing background noise from rainforest recordings*
3. Feature Extraction	Convert raw waveforms into spectrograms or embeddings	*Creating Mel spectrograms of bird calls*
4. Model Training	Train a deep learning model (CNN, CRNN, Transformer) on labeled data	*Training a CNN to classify frog species based on calls*
5. Model Evaluation	Test performance using accuracy, F1-score, confusion matrix	*Checking if the model correctly classifies known bird songs*
6. Deployment	Run the trained model on real-world audio (on-device or cloud)	*Deploying a real-time bat detector on a Raspberry Pi*
7. Model Updates & Maintenance	Fine-tune the model as new data becomes available	*Updating a whale call classifier with new recordings*



Future perspective and what to improve

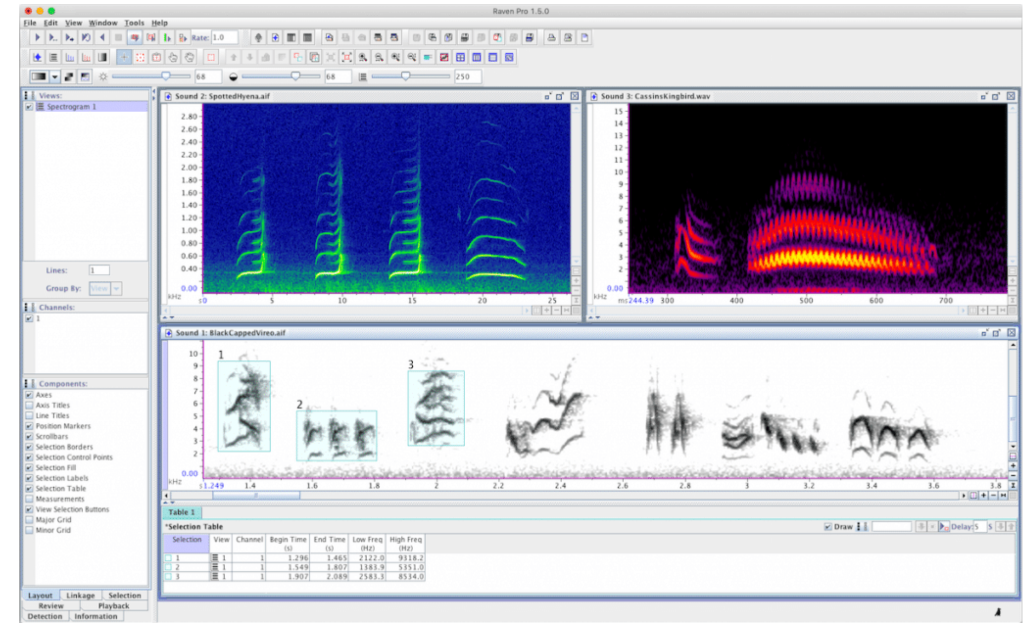
Roadmap Focus	Key Challenges	Solutions & Actions	Expected Benefits
1. **Expand and Share Large-Scale Bioacoustic Datasets**	➡ Many datasets are small, private, or poorly labeled ➡ Limited diversity in training data leads to poor generalization	✓ Create open-access bioacoustic databases (e.g., Xeno-Canto, Macaulay Library) ✓ Use crowdsourcing & citizen science to collect more labeled data ✓ Train with self-supervised learning (SSL) to learn from unlabeled recordings	✴ More data improves AI accuracy ✴ Enables better species identification Reduces need for large labeled datasets
2. **Improve Generalization & Domain Adaptation**	➡ Models trained in one location fail in new environments ➡ Microphone types, background noise, and geography cause domain shift	✓ Train on diverse datasets from multiple locations ✓ Use domain adaptation (fine-tuning, active learning) ✓ Apply data augmentation (adding noise, time shifting, pitch scaling)	✴ Models work across different environments Reduces errors in new habitats ✴ Improves real-world applicability
3. **Develop Open-Set Recognition & Novelty Detection**	➡ Most models assume all species are known ➡ New, unseen vocalizations get misclassified	✓ Implement confidence-based thresholds to flag new sounds Train models using few-shot learning ✓ Use clustering & anomaly detection to group unknown sounds	✴ Helps discover new species & call types Reduces misclassification errors ✴ Improves wildlife monitoring
4. **Integrate Contextual and Perceptual Information**	➡ Models don't consider environmental context (e.g., season, time of day) ➡ AI does not "hear" like animals	✓ Combine acoustic data with contextual metadata (e.g., time, weather, location) ✓ Train models to mimic animal auditory perception ✓ Use bio-inspired auditory features (e.g., PCEN, wavelets)	✴ AI makes more biologically accurate predictions ✴ Fewer false detections due to seasonal or behavioral factors ✴ Models align with real-world animal hearing
5. **Enable On-Device and Real-Time Processing**	➡ Most models require cloud computing & high power ➡ Remote locations lack internet & electricity	✓ Deploy models on low-power edge devices (e.g., Raspberry Pi, AudioMoth, Jetson Nano) ✓ Use model compression (quantization, pruning) ✓ Implement event-based processing to save battery life	✴ AI works without internet ✴ Enables real-time wildlife monitoring Energy-efficient for long-term conservation



Existing Software for Automated Species Detection"

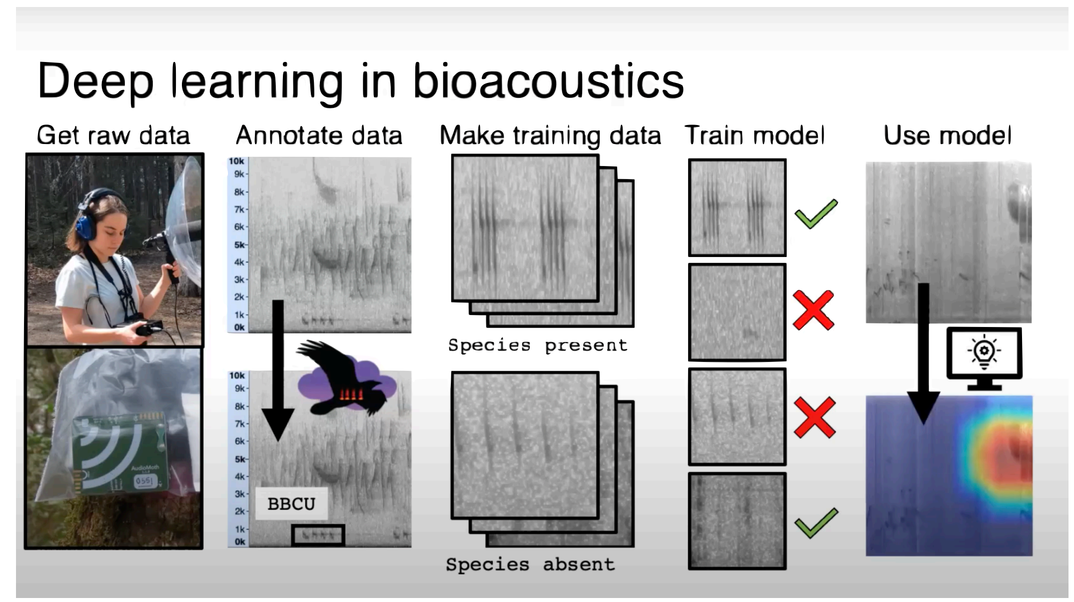
1. Software for Bioacoustic Data Management & Annotation

- **PAMGuard:** used for **data management** and **marine mammal passive acoustic monitoring**.
- **Audacity:** free, open-source tool for **audio exploration** and **basic editing**.
- **Raven Pro:** software for **manual annotation** of audio recordings.



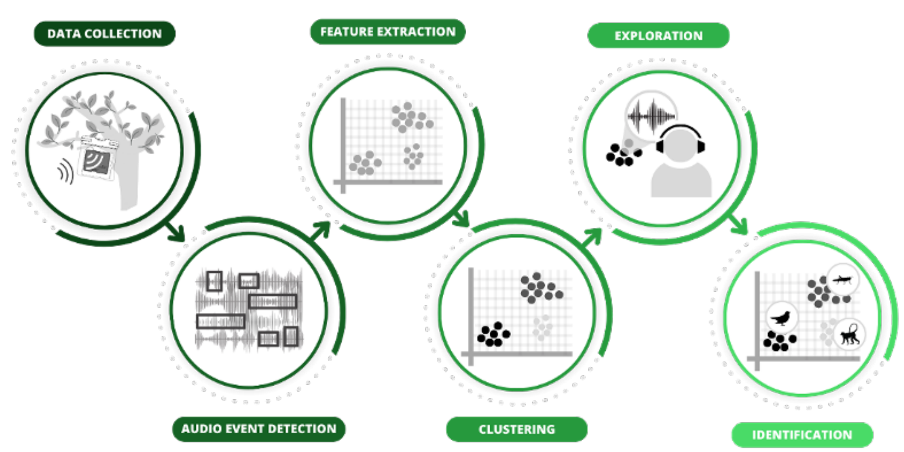
2. Software for Automated Species Detection

- **Kaleidoscope Pro** (Wildlife Acoustics): commercial tool for **species classification and call detection**.
- **BirdNET:** deep learning-based system for **bird sound classification**.
- **Ketos:** package for **underwater acoustic classification**.
- **Koogu:** Python-based package for **general acoustic analysis**.
- **aviaNZ:** designed for the classification of **New Zealand birds**.
- **ANIMAL-SPOT:** deep learning tool for **species-independent sound detection**.
- **gibbonfindR:** specialized tool for **gibbon vocalization detection**.
- **soundClass:** machine learning tool for **general biodiversity monitoring**.
- **OpenSoundscape** (OPSO): free and open source Python utility library analyzing bioacoustic data



3. Software Using Alternative (Non-Deep Learning) Methods

- **ARBIMON: template matching** for species detection.
- **monitoR:** R package for **energy-based sound detection**.



Deep-Learning-Based Earthquake Detection for Fiber-Optic Distributed Acoustic Sensing

Pablo D. Hernández, Jaime A. Ramírez, and Marcelo A. Soto, *Member, IEEE, Senior Member, OSA*

1. Fully Connected Artificial Neural Networks (FC-ANNs)
2. Convolutional Neural Networks (CNNs)
3. Recurrent Neural Networks (RNNs) (CNN+LSTM combination)

Training dataset: labeled seismic data from traditional broadband seismometers (STEAD)
Test data: real DAS earthquake measurements.

Model	Recognized Class	Real Class: Seismic	Real Class: Noise
FC-ANN (th: 0.422)	Seismic	**49.00%** (9799)	**0.28%** (57)
	Noise	**1.00%** (201)	**49.72%** (9943)
CNN (th: 0.964)	Seismic	**49.85%** (9969)	**0.03%** (6)
	Noise	**0.15%** (31)	**49.97%** (9994)
CNN + LSTM (th: 0.542)	Seismic	**49.76%** (9952)	**0.07%** (15)
	Noise	**0.24%** (48)	**49.93%** (9985)

The values indicated between brackets represent the number of waveforms recognized in the respective class.

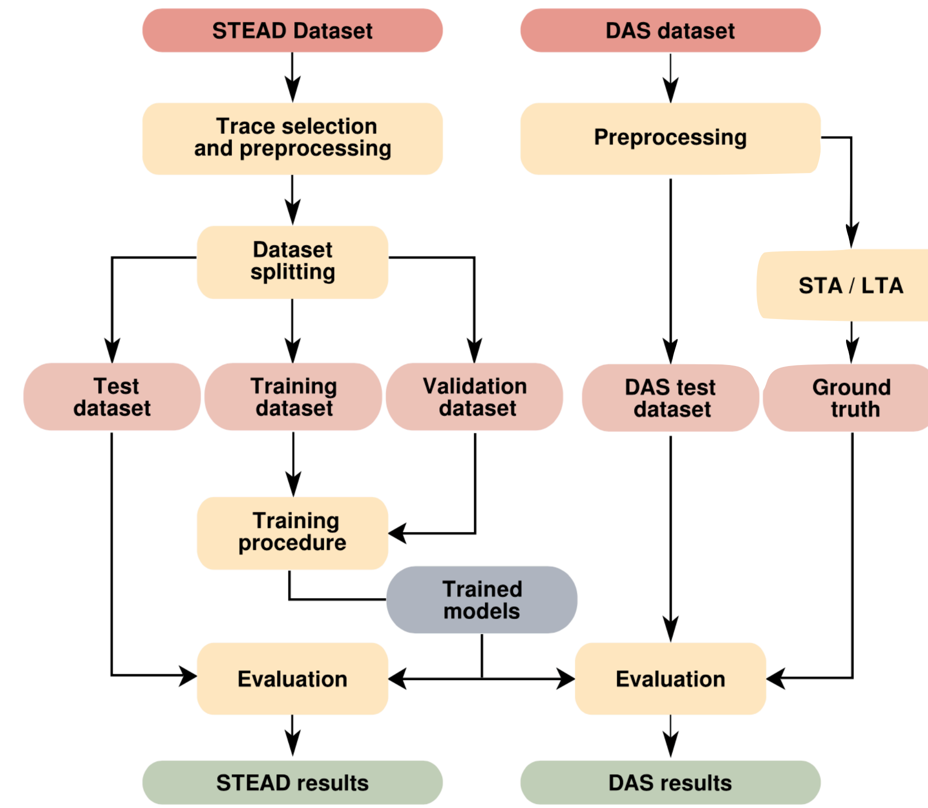
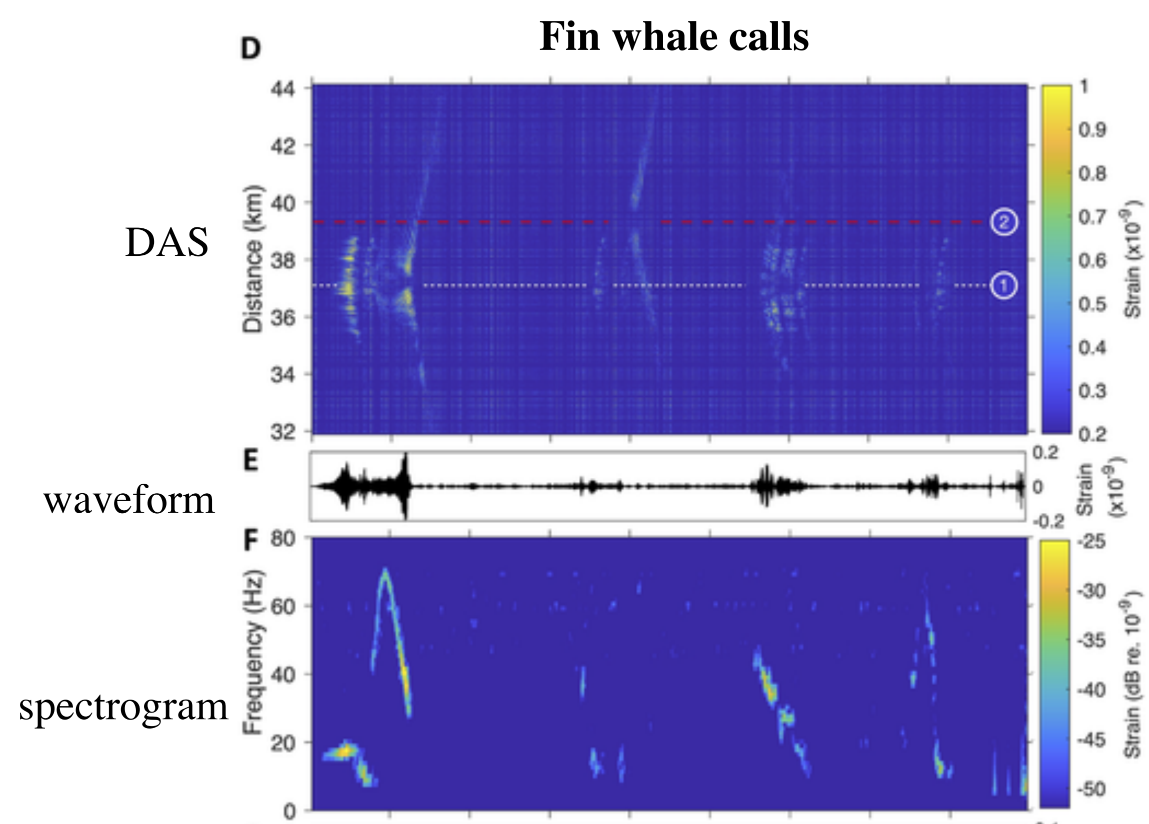
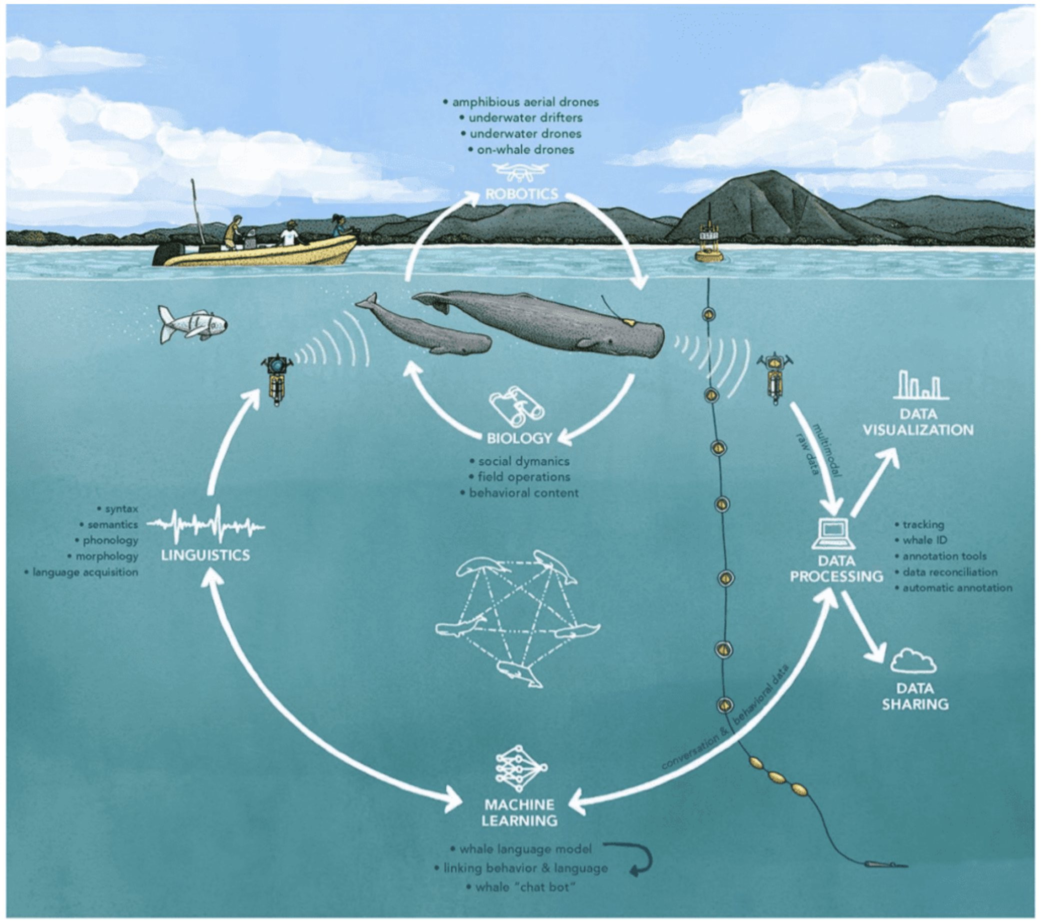


Fig. 6. Flow diagram describing training, validation, and test procedures.

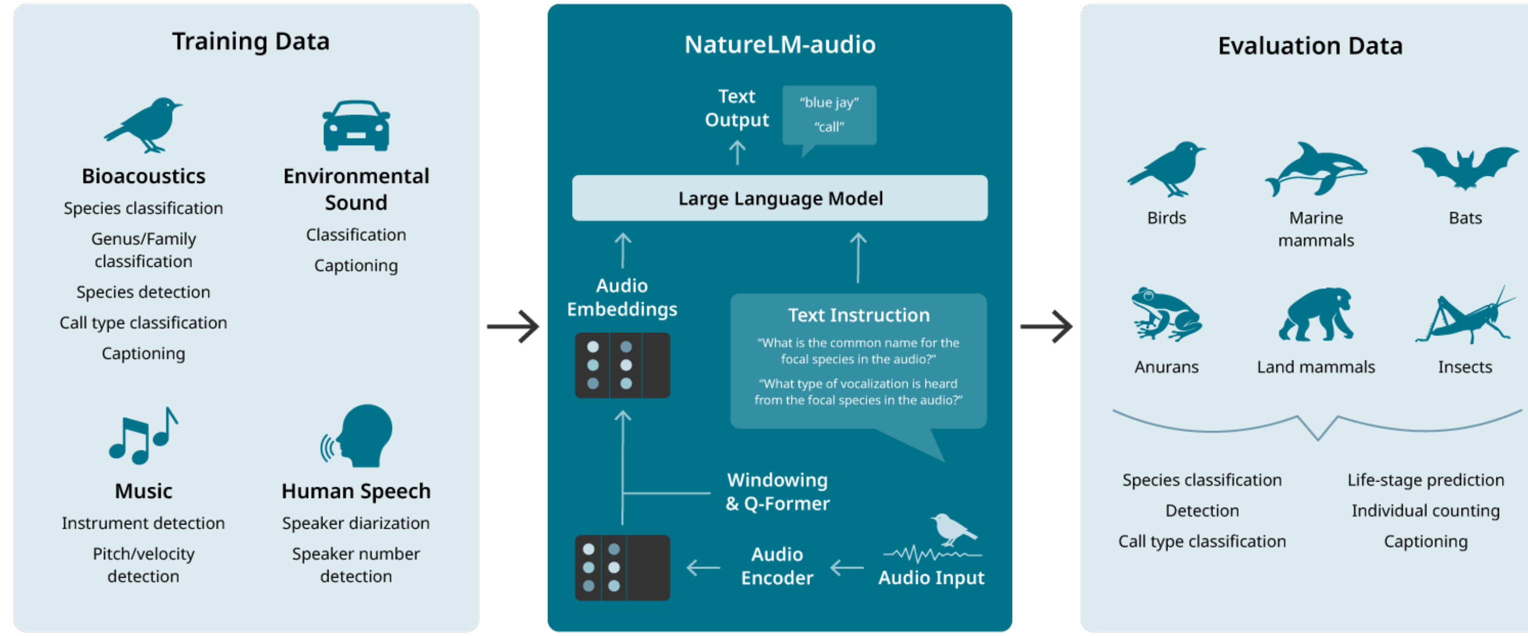


Cool project in bioacoustics DL

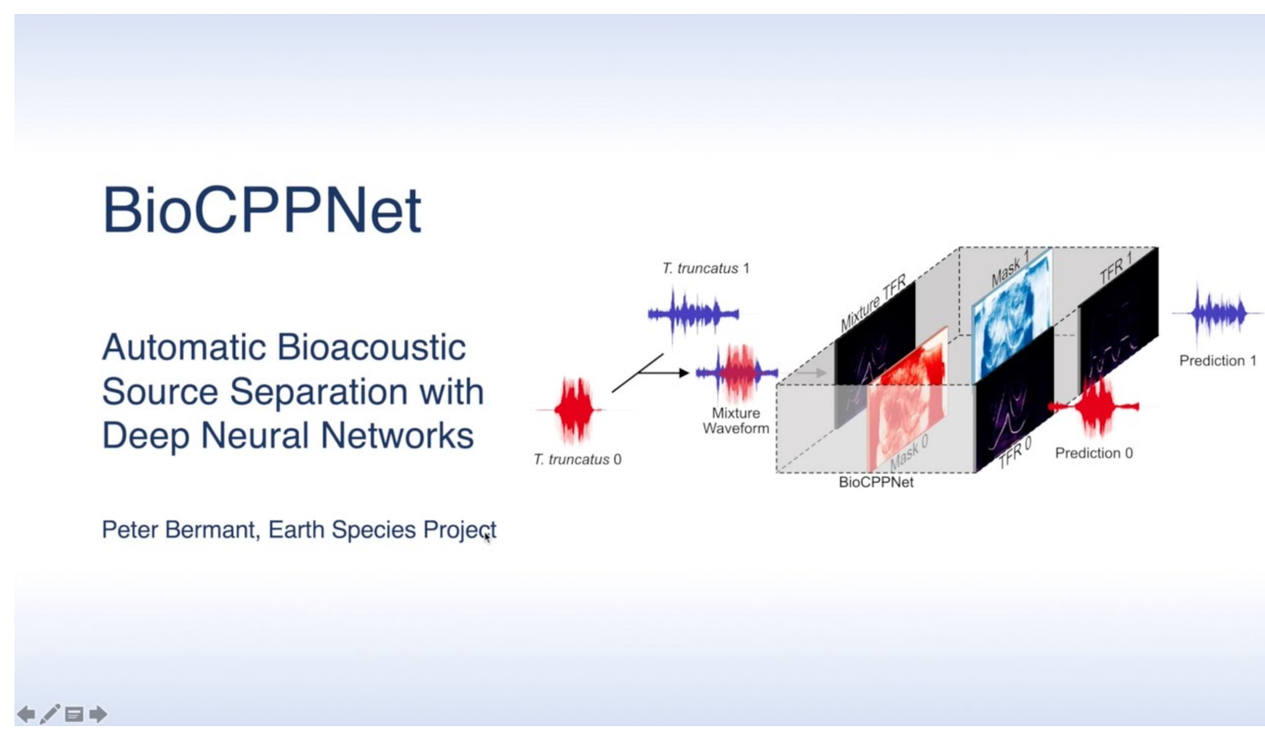
Project CETI



Earth Species Project (ESP)



NatureLM-audio is compiling and training on a large dataset of millions of audio-text pairs. Majority of this data comes from bioacoustic archives such as Xeno-canto, iNaturalist, the Watkins Marine Mammal Sound Database, and the Animal Sound Archive. Also general audio, human speech, and music data are included aiming to transfer learned abilities from human audio processing to animal sounds. NatureLM-audio is trained on this comprehensive dataset by connecting a self-supervised pretrained audio encoder to a leading language model (LLaMA 3.1-8B).




THANKS
A TON