Bird Song Recognition using Deep Learning

12/06/2024

Introduction

01

●

Objective

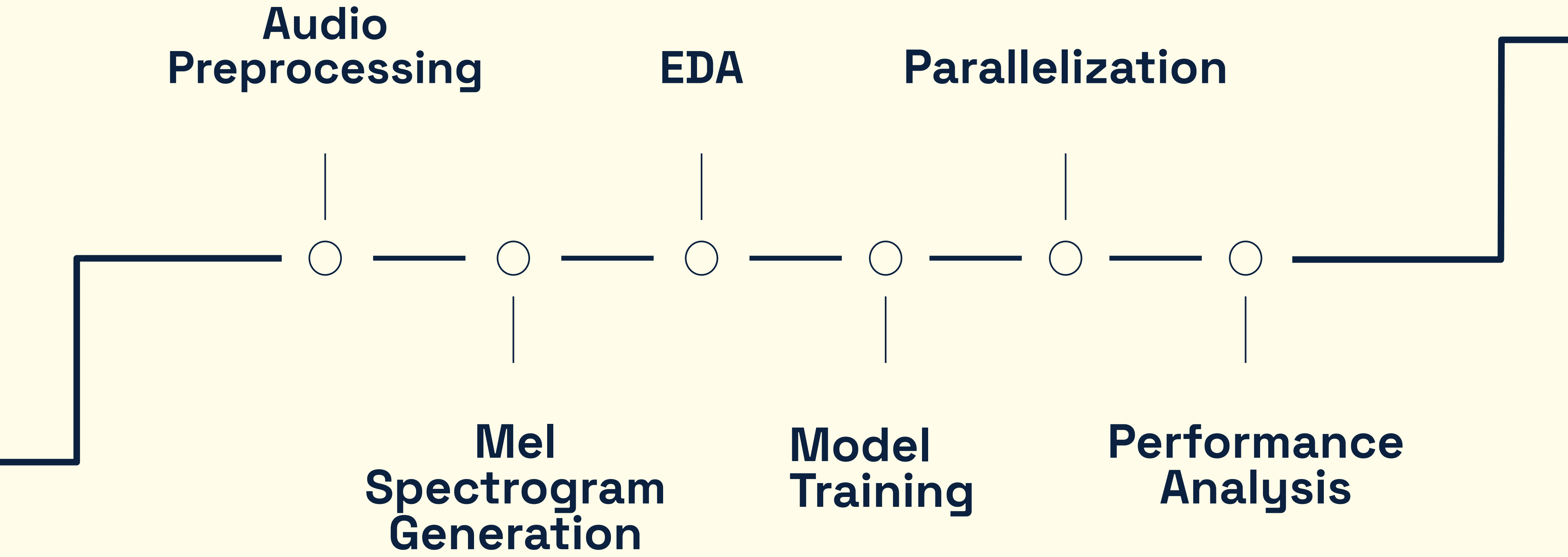
01

- **Goal:** Utilize deep learning models to classify bird species based on their vocalizations.

- **Challenge:** Large-scale datasets (~26GB) result in lengthy and resource-intensive model training.

- **Solution:** Parallel computing optimizes data processing and model training, enabling efficient bird song classification.

Milestones



Audio Preprocessing

Mel Spectrogram Generation

EDA

Model Training

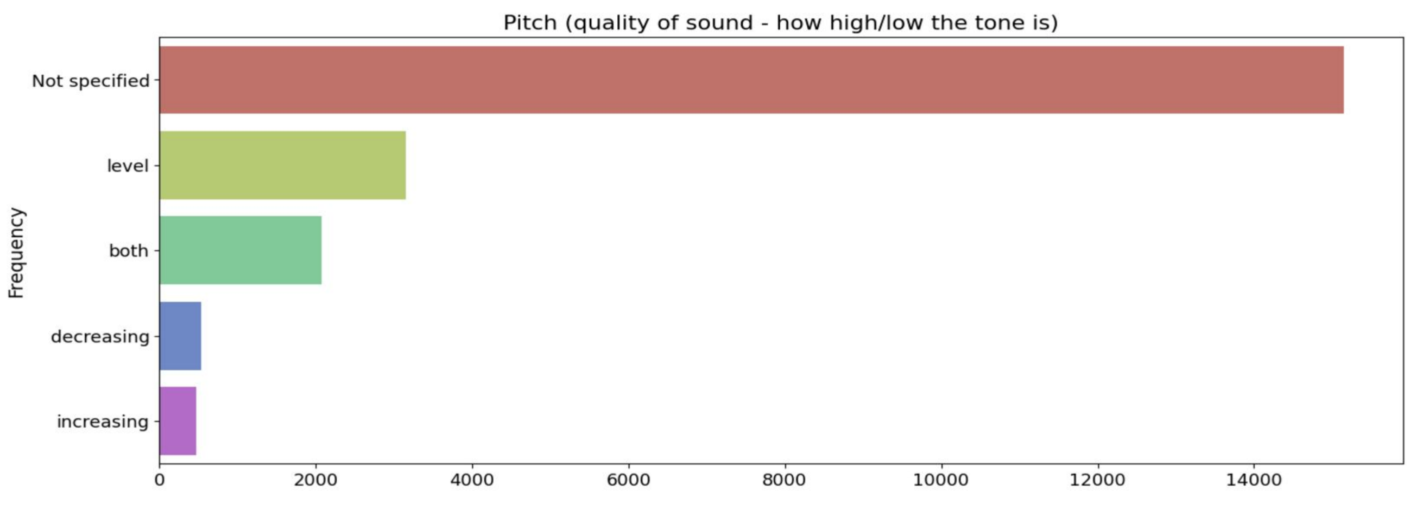
Parallelization

Performance Analysis

EDA

- **Dataset Overview**: Loaded and inspected for structure, missing values, and distribution of bird species.
- **Feature Extraction**: Audio clips converted to Mel-spectrograms for time-frequency representation.
- **Data Splitting**: Divided into training, validation, and test sets.
- **Label Encoding**: Bird species labels encoded into numeric format for model training.

EDA Analysis



The data is heavily skewed towards **"Not specified"**, suggesting that in most cases, pitch quality was not labeled or measured.

Audio Preprocessing

02

Load Audio:

- Use **Librosa** to load audio file with a sample rate of 32,000 Hz.
- Extract a 5-second segment from the audio signal.

Generate Mel Spectrogram:

- Convert the audio signal into a Mel spectrogram using **Librosa's `melspectrogram` function**.
- Parameters: 128 Mel bands, frequency range from 20 Hz to 16,000 Hz.

Convert to Decibels:

- Apply **logarithmic scaling** to convert the spectrogram to decibels for better visualization.

Convert to RGB:

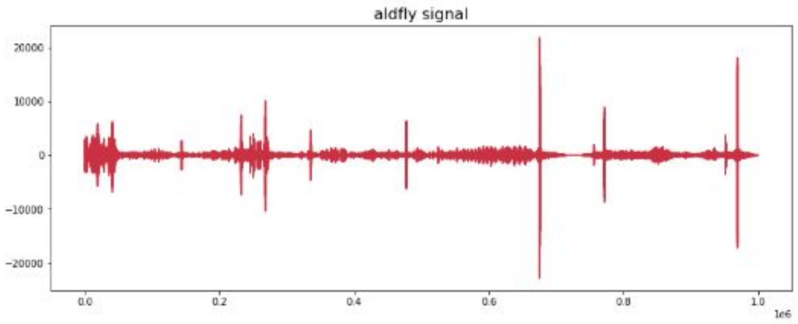
- Normalize and convert the Mel spectrogram into a 3-channel **RGB image** for model input.

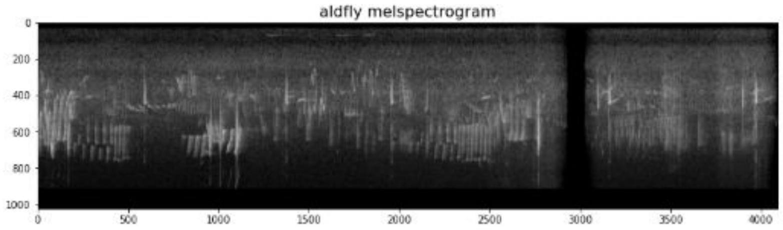
Resize:

- Resize the image to a fixed size (224x224) to match model input dimensions.

Why Use Mel-Spectrogram?

Improved Classification: When using deep learning models (especially Convolutional Neural Networks, CNNs), Mel-spectrograms work as input images for the model, capturing the key features of the audio signal that are most important for classification tasks.





Workflow

02

- **Audio preprocessing using Mel spectrograms.**
- **Model training with ResNet50 architecture for feature extraction.**
- **Parallelization for faster computation using Dask.**

Data Augmentation

02

- **Mel Spectrogram Extraction**:
    -
    - A spectrogram converted to the mel scale is called a melspectrogram and is a great way to convert audio signal data to a visual feature map to train an image model (like ResNet-34)Audio files are transformed into **Mel spectrograms** using librosa.
    - Spectrograms are resized to a specific shape (48x128) to match model input.

- **Label Encoding**:
    - Bird species are encoded into numeric labels using `LabelEncoder`.
    - There are 264 unique bird species in the dataset.

Model Training

02

1. **Efficient Deep Learning**: ResNet50 uses residual connections to train deep networks, avoiding the vanishing gradient problem and enabling effective learning in very deep architectures (e.g., ResNet50).

2. **Transfer Learning**: Pretrained on ImageNet, ResNet50 leverages learned features and fine-tunes on birdcall data, improving performance even with smaller datasets.

3. **Powerful Feature Extraction**: ResNet excels at extracting high-level features from mel spectrograms, which are image representations of bird calls, making it ideal for audio-based classification tasks.

4. **State-of-the-Art Accuracy**: ResNet has proven to achieve high accuracy in complex classification tasks, making it well-suited for distinguishing between bird species based on subtle differences in their calls.

Road Block

02

1. **The data is heavily skewed towards the "Not specified" category, suggesting that pitch quality was either not recorded or measured in most cases. The data is highly imbalanced, with over 70% of the entries falling under "Not Specified."**

2. **Due to this, Librosa / Pydub was unable to convert the data into Mel spectrograms, despite having around 21,700 audio files.**

3. **The slow audio feature extraction process further prevented the conversion of the entire dataset into Mel spectrograms.**

4. **As a result, we proceeded with training the ResNet50 model on a smaller test dataset instead of the full dataset.Converting audio files into spectrograms using Librosa is computationally expensive, especially for large datasets. The extraction process itself takes significant time, especially if the spectrogram generation involves complex parameters (like high n_mels or larger window sizes).**

1. **Computational Complexity: Spectrogram generation is inherently resource-intensive due to Fourier transformations and other signal processing operations.**
2. **Processing Overhead: Generating spectrograms for thousands of audio files demands substantial computation, particularly if the audio files are lengthy or if no GPU acceleration is used.**
3. **Resource Constraints: Insufficient memory and CPU/GPU inavailabilty drastically slowed the process.**

Parallelization

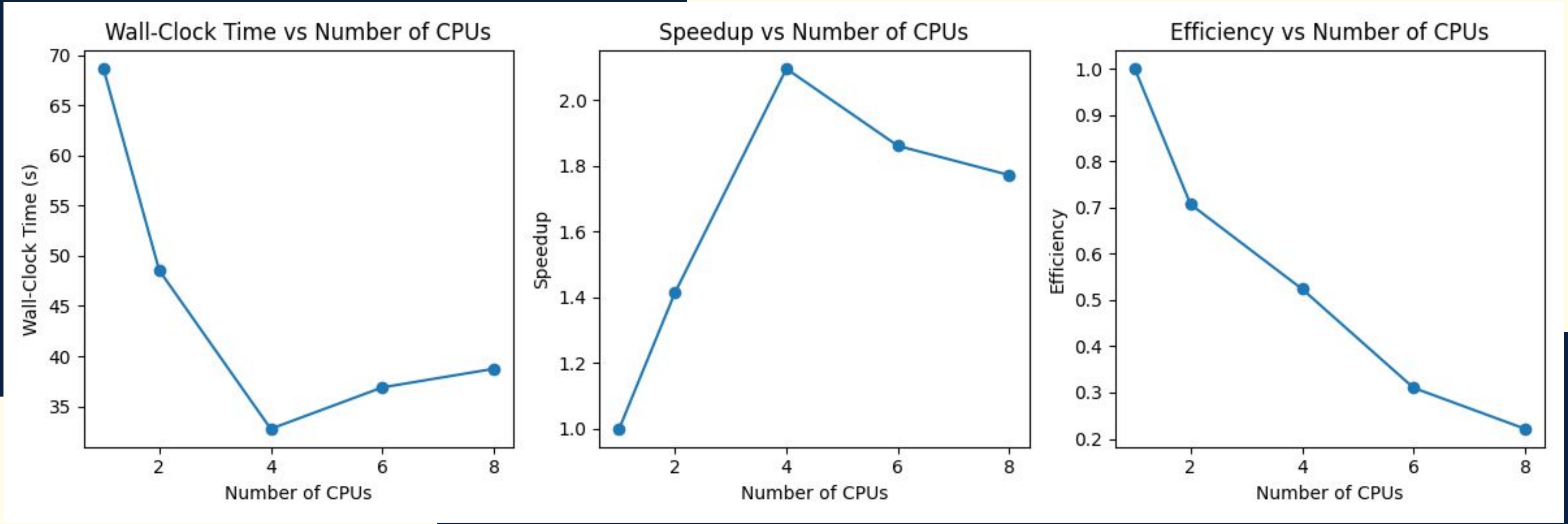
02

Dask Cluster Setup:

- A Dask cluster is created with configurable workers.
- Tasks are distributed and executed in parallel for predictions.

Cluster Performance:

- Performance is evaluated using Dask’s dashboard.
- Speedup and efficiency plots are generated to assess the impact of parallelization.


Wall-Clock Time vs Number of CPUs
Wall-Clock Time (s)
70
65
60
55
50
45
40
35
2
4
6
8
Number of CPUs
Speedup vs Number of CPUs
Speedup
2.0
1.8
1.6
1.4
1.2
1.0
2
4
6
8
Number of CPUs
Efficiency vs Number of CPUs
Efficiency
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2
4
6
8
Number of CPUs

Wall-Clock Time vs Number of CPUs

- **Key Observations:**
  - **Significant Decrease:** Wall-clock time decreases drastically as CPU count increases from 1 to 4.
  - **After 4 CPUs:** The reduction in time slows down, with little change observed after 4 CPUs.
  - **Conclusion:** Diminishing returns in performance improvements after 4 CPUs.

Speedup vs Number of CPUs

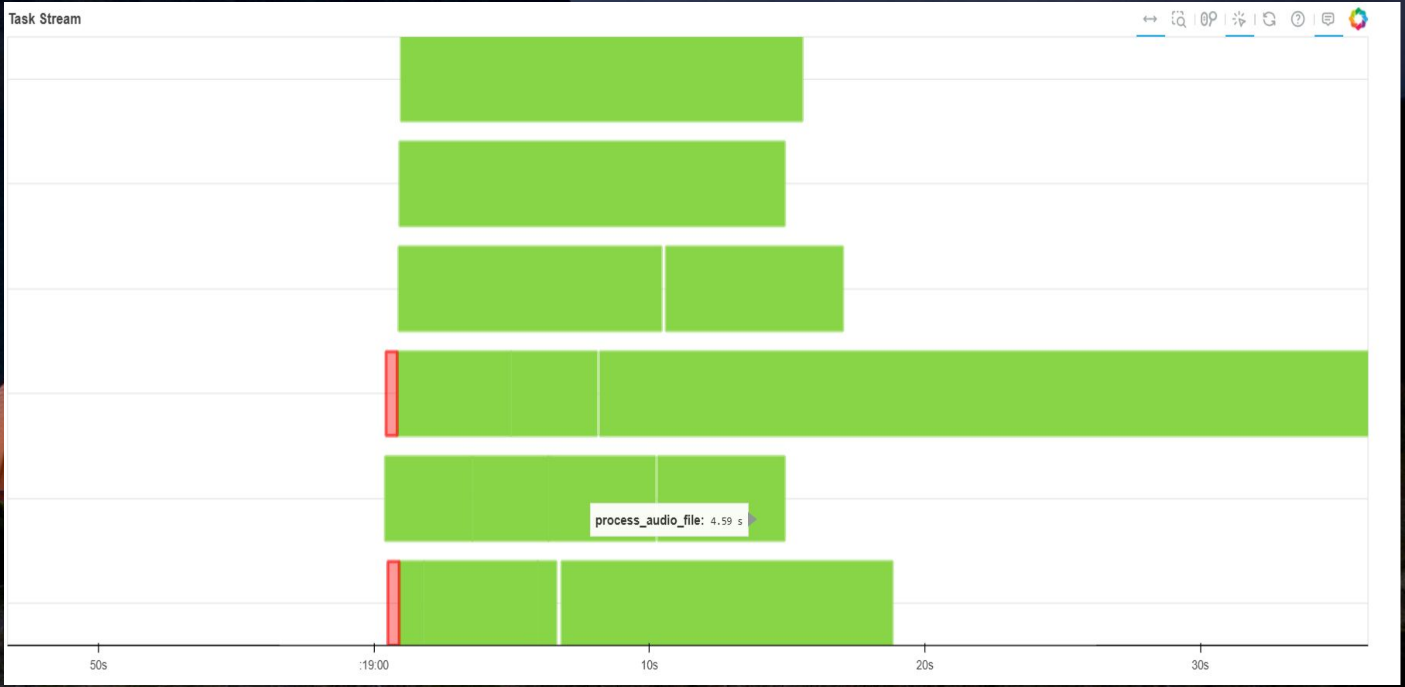
Key Observations:

- **Peak Speedup:** The highest speedup (around 2x) occurs with 4 CPUs, meaning the task completes twice as fast as with a single CPU.
- **Plateau Effect:** Adding more CPUs beyond 4 results in little to no significant speedup.
- **Conclusion:** Speedup increases up to 4 CPUs, after which further CPU additions offer reduced gains.

Observation

- **Performance Gains:** Scaling from 1 to 4 CPUs provides substantial performance improvements.
- **Diminishing Returns:** Performance improvements diminish as more CPUs are added beyond 4.
- **Efficiency Decline:** Efficiency decreases as more CPUs are added, likely due to overheads.
- **Conclusion:** Optimal performance is achieved with 4 CPUs, with minimal gains observed from scaling beyond that point.

Performance Analysis



Performance Analysis:

- **Efficiency**: The system seems to be processing tasks sequentially, with the `process_audio_file` task taking a lot of time compared to others. This could be because audio processing is inherently time-consuming, especially if it involves operations like feature extraction (e.g., Mel spectrogram generation).

Thank

You