Final Project Proposal

"Bird Sound Classification Using MEL Spectrograms"

- Wali Siddiqui & Anurag Surve

1. Problem Selection and Motivation:

We have selected the task of bird species identification from audio recordings to address the challenge of biodiversity monitoring and environmental conservation. Traditional fieldwork and manual audio screening are labor-intensive and subject to errors. With the growing availability of large-scale bioacoustics data, automated deep learning approaches offer a promising solution to accurately and efficiently monitor bird populations. The BirdCLEF 2025 dataset from Kaggle provides a diverse collection of audio recordings captured under real-world conditions, making it ideal for training a robust deep network to handle noisy data and varied acoustic environments.

2. Dataset Description:

The dataset is provided by the BirdCLEF 2025 competition on Kaggle (see Kaggle BirdCLEF 2025). It contains thousands of audio clips recorded in different regions with annotated species labels. The size and diversity of the dataset ensures that it is large enough to train a deep network while also posing challenges like background noise and class imbalance commonly observed in ecological data.

3. Proposed Deep Learning Network:

Our approach will be to implement a convolutional neural network (CNN) tailored for processing MEL spectrograms—visual representations of the audio frequency content over time. To better capture the temporal dependencies within the audio signals, we will explore integrating recurrent layers (LSTM/GRU) on top of the CNN architecture. We will begin with a standard CNN designed for image-like data and then incorporate customized modifications as needed to improve performance on noisy recordings. In particular, we will use the Kaggle notebook "lb-0-778-efficientnet-b0-pytorch-inference" as a benchmark, with our objective being to further enhance its accuracy and robustness in bird species classification.

4. Framework and Implementation:

We will use the PyTorch framework for implementing our deep learning model. PyTorch was chosen because of its dynamic computation graph, ease of experimentation, and strong community support, all of which are instrumental during model tuning and troubleshooting.

5. Reference Materials and Background:

To build a strong foundation for our project, we will consult:

- Research papers on bioacoustics and deep learning applications to audio that focus on methods utilizing MEL spectrograms.
- Documentation and tutorials for PyTorch and audio processing libraries such as Librosa.
- Kaggle kernels and discussion forums related to the BirdCLEF competitions.
- The benchmark notebook "lb-0-778-efficientnet-b0-pytorch-inference" for insights on efficient network implementation and inference procedures.

6. Performance Evaluation:

- **Macro-Averaged ROC-AUC:**
  For this contest, the evaluation metric is a modified version of the macro-averaged ROC-AUC. In traditional macro-averaging, ROC-AUC scores are computed individually for each class and then averaged, giving equal importance to every class regardless of its frequency. However, this contest's version skips any class that has no true positive labels. This modification is necessary because calculating ROC-AUC requires both positive and negative instances for a class. If a class has no true positives, the ROC curve cannot be constructed accurately and including it could distort the overall metric. By excluding such classes, the metric focuses solely on those where meaningful performance can be measured, thereby providing a more robust and fair evaluation of the model's ability to differentiate between classes, even in the presence of class imbalances.

7. Rough Schedule (4 Weeks):

- **Week 1:**
  - Finalize the literature review and set up the development environment.

    - Conduct initial data exploration and build the preprocessing pipeline, including audio cleaning and MEL spectrogram extraction.
    - Run preliminary experiments on data augmentation strategies to mitigate class imbalances.
- **Week 2:**
    - Implement a baseline CNN architecture inspired by the benchmark notebook.
    - Begin initial training cycles and perform hyperparameter tuning.
    - Evaluate early model performance using a dedicated validation set.
- **Week 3:**
    - Integrate enhancements such as recurrent layers (LSTM/GRU) to effectively capture temporal features.
    - Refine the model architecture and optimize training parameters to reduce overfitting.
- **Week 4:**
    - Conduct a final evaluation on a test set and perform a detailed error analysis using figures and confusion matrices.
    - Compile the results, prepare the presentation, and draft the final project report.

8. References:

1. **Kaggle BirdCLEF 2025 Competition Data.** Available from: https://www.kaggle.com/competitions/birdclef-2025/data
2. **Benchmark Notebook – lb-0-778-efficientnet-b0-pytorch-inference.** Available from: https://www.kaggle.com/code/xiaoazuzong/lb-0-778-efficientnet-b0-pytorch-inference
3. **Cornell Lab of Ornithology – Birds of the World.** Official website: https://birdnet.cornell.edu/