# Identifying bird species through their calls with EfficientNetV2

**Stefanela Stevanovic (63220492)**

## 1 Introduction

This project is inspired by the BirdCLEF24 competition on Kaggle, which challenges participants to develop machine learning solutions for identifying under-studied Indian bird species through their calls.

Monitoring bird populations is crucial for assessing ecosystem health, but traditional observer-based bird biodiversity surveys over large areas are expensive and logistically challenging. This aim of this project is to leverage passive acoustic monitoring (PAM) and machine learning to develop reliable classifiers for supporting conservation efforts in the Western Ghats, India.

## 2 Methodology

### 2.1 Data

The data consists of short recordings of individual bird calls belonging to 182 different bird species. These files have been downsampled to 32 kHz where applicable and made available to us in ogg format. This dataset contains 24 459 labeled recordings of different length. It is also important to mention that the dataset contains imbalanced classes, with some classes having around 500 samples, while others have only about 10-20. The distribution of the number of labels per class is shown in Figure 1

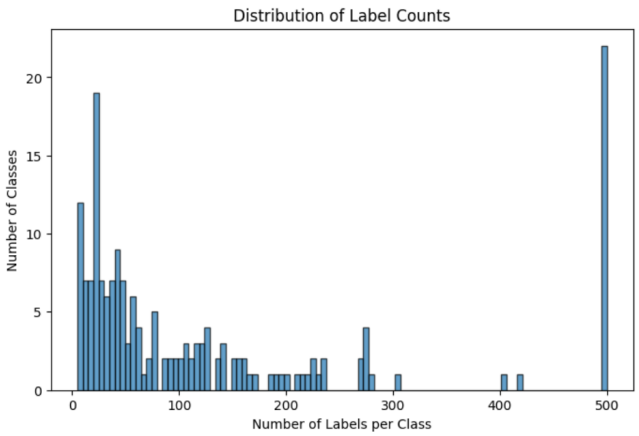


Figure 1: Distribution of label counts per class.

Per competition rules, the evaluation should be done on a secret dataset with approximately 1,100 recordings, which are used for scoring. This dataset is made available to the notebook only after submission, meaning we don't have access to the labeled test set. Since we would like to have a labeled dataset for testing, we opted to create our own test dataset from the original dataset meant for training. We took 15% of the original data as the test set. The remaining data was used for training and split into training and validation sets, with 20% used for validation and the rest for training. The data was split using a stratified method to ensure that each class is proportionally represented across all set.

### 2.2 Data Preparation

The data preparation process involved several critical steps to ensure that the audio data was appropriately formatted and preprocessed for input into a machine learning model. TensorFlow and TensorFlow I/O were used to efficiently decode and preprocess audio data, while 'tf.data' was used to generate datasets and create optimized data pipelines. The following steps were used in data preparation:

- **Audio length adjustment:** Since audio files can vary in duration, it was necessary to standardize the length of each audio input. This was achieved by either padding or cropping the audio files to a fixed duration (5 seconds for baseline model and 15 seconds for other models). When an audio file was shorter than the target length, it was padded with zeros using a random padding strategy. When an audio file was longer than the target length, it was randomly cropped to the desired length.
- **Mel-spectrogram generation:** The standardized audio signals were converted into Mel-spectrograms that represents the power spectrum of frequencies over time. Mel-spectrograms were generated using 128 Mel bins, a Fast Fourier Transform (FFT) length of 2028 and a frequency range of 20 Hz to 16 kHz.
- **Normalization and standardization:** Each spectrogram was standardized by subtracting the mean and dividing by the standard deviation of its values. Then, a Min-Max normalization was applied to rescale the spectrogram values to a range between 0 and 1.
- **Conversion to 3-channel image format:** To adapt the Mel-spectrograms for use with pre-trained ImageNet

models each spectrogram was duplicated across three channels to mimic the RGB channels of an image.

Figure 2 presents a simplified preprocessing pipeline for the audio signal of a Black-hooded Oriole, while Figure 3 shows four sample Mel-spectrograms from a training batch.

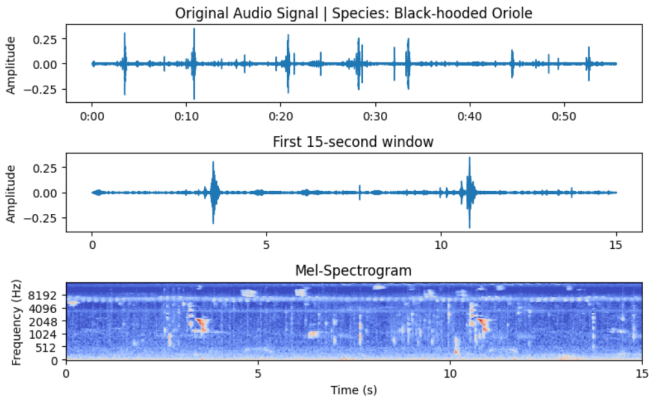


Figure 2: Preprocessing pipeline - Full Waveform, 15-second window, and Mel-spectrogram visualization

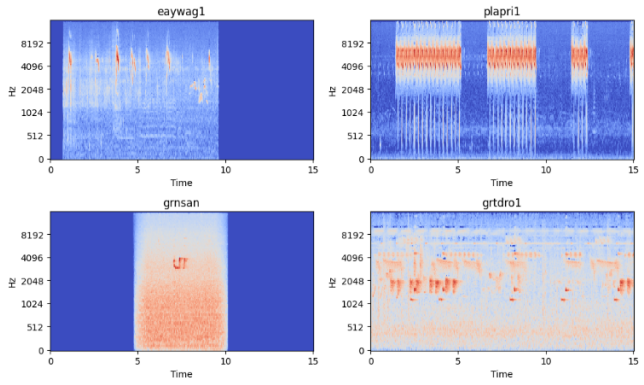


Figure 3: Mel-spectrograms from a training batch.

## 2.3 Model Training

In this project, we utilized the EfficientNetV2-B2 model as the backbone for our bird species classification task. EfficientNetV2 represents a new generation of models that are not only smaller and faster but also outperform previous architectures in terms of both training speed and parameter efficiency. Published weights of EfficientNetV2-B2 are capable of scoring 80.1% top 1 accuracy and 94.9% top 5 accuracy on imagenet [1].

The input layer was defined with a flexible shape of (None, None, 3), allowing the model to process Mel-spectrograms of varying dimensions. This is important since we will not always be training and predicting on images of the same size. The EfficientNetV2 backbone was connected to an ImageClassifier layer specifically designed for bird species classification.

We used Categorical Crossentropy as the loss function and label smoothing of 0.02 to reduce the risk of overfitting. The model was optimized using Adam optimizer and trained with a batch size of 64 samples for a maximum of 20 epochs. Training progress was monitored using the validation loss, with an early stopping mechanism that halted training if no improvement was observed after 3 consecutive epochs. Additionally, a learning rate scheduler was employed, reducing the learning rate by a factor of 0.5 if the validation loss did not improve for more than 2 epochs.

We trained three different models, each utilizing the EfficientNetV2 backbone but with varying training strategies:

- **Baseline model:** Trained on Mel-spectrograms generated from random 5-second segments of the entire audio recordings.
- **Model 1:**: Trained on 15-second segments to increase the likelihood of capturing bird calls within the Mel-spectrogram.
- **Model 2:** Also trained on 15-second segments, but with additional data augmentation for underrepresented classes (defined as classes with fewer than 40 samples). For this model we included multiple 15-second segments from the same audio belonging to the underrepresented classes and also applied random time and frequency masking to prevent overfitting.

## 2.4 Evaluation strategy and Metrics

Given that the recordings in the test set vary in length, we decided to implement a strategy that allows for predictions on variable-length audio clips by segmenting them into 5-second frames, on which predictions are made. For recordings shorter than 5 seconds, padding is applied to reach the required length, creating a single frame for that clip. For longer recordings, the audio is divided into consecutive 5-second frames, with any remaining portion at the end also padded if it is less than 5 seconds. These frame-level probabilities per class are then aggregated by summing and averaging them across all frames, producing a final prediction for the entire recording.

To evaluate the performance of our bird species classification model, we used top-k accuracy, which shows the model's ability to correctly identify the bird species among its top predictions. Since we have imbalanced classes, we also computed macro-averaged precision and recall, which treat each class independently of its frequency. This can help us to understand how well the model can identify less frequent species. Additionally, we calculated micro-averaged (weighed) precision and recall, which aggregate the performance across all classes, giving more weight to the more frequent species.

# 3 Results

Figure 4 shows the top-1 to top-10 accuracy for all three models (baseline, Model 1, and Model 2). We can see that Model 1, trained on longer segments of 15 seconds, demonstrates slightly better performance compared to the baseline model, which was trained on 5-second segments. Model 2 achieves

the highest accuracy across all Top-k metrics, with around 74.30% on Top-1 accuracy.

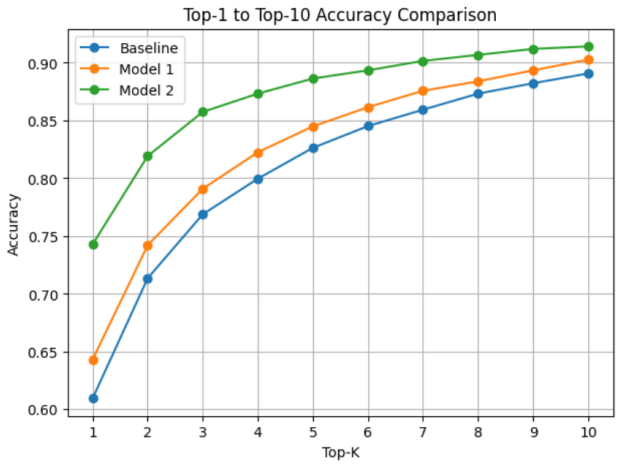


Figure 4: Top-1 to top-10 accuracy for all three models (baseline, Model 1, and Model 2).

Additionally, we notice that the improvement in performance between Model 1 and Model 2 is significantly larger when looking at Top-1 accuracy compared to Top-10. It is likely that using data augmentation for the imbalanced classes has led to the most significant improvement in Top-1 accuracy. By having more samples to train on, the model was better able to assign the highest probability to the correct bird species.

In Table 1 we present the results of macro- and micro-averaged precision and recall for all models.

| | Baseline | Model 1 | Model 2 |
|---|---|---|---|
| Macro-averaged Precision | 39.60% | 47.80% | 63.22% |
| Micro-averaged Precision | 60.97% | 64.32% | 74.30% |
| Macro-averaged Recall | 33.61% | 39.16% | 57.27% |
| Micro-averaged Recall | 60.97% | 64.32% | 74.30% |

Table 1: Comparison of models based on macro- and micro-averaged (weighed) precision and recall.

We observe that the micro-averaged values for precision and recall are significantly higher than the macro-averaged ones, indicating that classes with fewer data are challenging for our model. Again, we see that both Model 1 and Model 2 outperform the baseline, with Model 2 achieving the best performance overall.

## 4 Conclusion

EfficientNetV2 demonstrated strong performance in classifying bird species based on their audio recordings, with promising potential for further enhancement. Our results showed that training on mel-spectrograms derived from longer audio segments (15 seconds) led to superior test set performance compared to the baseline model, which was trained on mel-spectrograms from shorter 5-second audio segments. Additionally, a significant improvement in performance was achieved by addressing class imbalance through the generation of more samples from underrepresented classes, combined with the application of time and frequency masking techniques.

To further improve the model's accuracy and robustness, future work could explore varying the lengths of audio segments used for mel-spectrogram generation and increasing the number of segments extracted from each audio recording.

## References

[1] Keras CV Models.
https://keras.io/api/keras_cv/models/