

A
Project Report
On
**AvianNET: An Intelligent Bird Species
Identification System through Audio
Recordings using CRNN- GRU**

By
Sunny Chowdhary, Muppala - 200483456

Abstract

This project uses advanced deep learning algorithms to improve bird species identification from audio recordings, which is a critical component of ecological monitoring. AvianNET, our proposed framework, uses a comprehensive audio pre-processing methodology to obtain Mel Frequency Cepstral Coefficients and Mel Spectrograms for feature analysis. By training Convolutional Recurrent Neural Networks with Gated Recurrent Units, our model outperforms prior approaches, reaching 44.26% accuracy on the validation set—a significant gain over previous benchmarks. This interesting technique demonstrates the potential for future improvements in automated audio-based bird identification systems.

Table of Contents

1. Introduction

1.1 Background

1.2 Problem Statement

1.3 Objectives

2. Literature Review

2.1 Improvements in Contests for Bird Sound Identification

2.2 Scientific Improvements in the Analysis of Bird Vocalization

3. Dataset Information

3.1 Dataset Specifications

4. Methodology

4.1 Audio Data Pre-processing

4.2 Feature Extraction

4.2.1 MFCC Extraction

4.2.2 Mel Spectrogram Extraction

4.2.3 Visualization and Saving Features

4.3 Deep learning model architectures

4.3.1 Convolutional Neural Network (CNN)

4.3.2 Convolutional Recurrent Neural Network-Gated Recurrent Unit (CRNN-GRU)

4.4 Evaluation Metrics and Experiments

5. Results

6. Conclusion

7. References

1. Introduction

Bird watching, also known as birding, is a popular and enduring activity enjoyed by many around the globe. This hobby involves observing and identifying birds in their natural environments, offering a peaceful and engaging way to connect with nature. Enthusiasts often equip themselves with binoculars or telescopes to better view these creatures, and they utilize guidebooks or mobile applications to help identify various species. Whether practiced alone or as part of a group, bird watching offers a delightful escape into the world of avian life.

1.1 Background

Traditionally, there are two main ways to identify birds while bird watching: visual and auditory. Visual identification is based on the physical characteristics of the bird, such as its plumage patterns and behaviors, as seen through telescopes or binoculars. On the other hand, auditory identification relies on the ability to distinguish bird calls and songs, which are frequently more accurate indicators than visual cues. This is especially true in situations where it is difficult to observe things visually, like heavily forested areas or dimly lit areas. Because they make it possible to identify birds even when they are hidden or far away, auditory methods can greatly improve the experience of bird watching.

Thanks to technological tools, amateur ornithologists can now identify birds by their sounds as well as those of more seasoned observers. The use of audio identification makes use of each bird species' distinct vocal signatures, which are dependable and less likely to be mistaken for those of other birds. The development of digital recorders and smartphone apps that can instantly record and analyze sounds has given this technique more traction. Because they offer accurate information on bird presence and behavior in diverse habitats, these tools have the potential to revolutionize bird watching and make it more interesting and beneficial to science.

1.2 Problem Statement

While the use of audio recordings for bird species identification has grown, several challenges hinder its effectiveness and accessibility. Current systems often struggle with the

accurate recognition of species due to the presence of background noises, varying qualities of audio recordings, and the intrinsic variability in bird calls across different environments and contexts. These limitations can lead to inaccuracies that frustrate users and diminish the reliability of data for scientific research. Consequently, there is a pressing need for an improved solution that can handle these complexities more effectively.

1.3 Objective of the Project

Using the potential of deep learning technologies, the main goal of AvianNET is to develop and improve a methodology for correctly classifying bird species based on their chirps. Our goal is to examine and modify current deep learning models, especially those that have shown promise in audio processing domains such as speech recognition, to satisfy the particular requirements of bird call identification. To optimize the model for the dynamic nature of bird calls, we plan to use Convolutional Recurrent Neural Networks (CRNNs), which combine the spatial feature recognition of CNNs with the temporal pattern learning capabilities of RNNs. Improving the model's accuracy and dependability in various difficult acoustic environments—like those with a lot of background noise or overlapping bird sounds—will be a major priority.

2. Literature review

The identification of bird species from their calls can be accomplished with the help of strong audio processing tools that have been made possible by recent advances in machine learning, especially in deep learning. Our project uses Mel Frequency Cepstral Coefficients (MFCCs) and Convolutional Neural Networks (CNNs) for bird sound identification, drawing on techniques used in human emotion recognition research. To accurately classify a variety of bird species, these techniques emphasize important components like feature extraction and model optimization. Research such as that done by Aishwarya et al. (2020) shows how these sophisticated methods can be used to identify patterns and features in intricate audio data [1].

2.1 Improvements in Contests for Bird Sound Identification

In tackling the particular issues of the area, recent bird sound identification competitions have shown significant progress. The study by Lasseck (2019) produced noteworthy outcomes in the BirdCLEF 2019 competition [2]. Finding and classifying birds that could be heard in different acoustic environments was the goal of this contest. The Classification Mean Average Precision (cmAP) was the primary assessment parameter, and each soundscape was divided into 5-second intervals using a thorough methodology [3]. With a focus on improving the accuracy of bird sound identification technology, this technique highlights the meticulous and deliberate efforts needed to identify species within realistic auditory situations reliably.

2.2 Scientific Improvements in the Analysis of Bird Vocalization

Individual research investigations have also contributed to the growth of bird sound recognition systems in parallel with competitive improvements. Hiatt's work on bird vocalizations provides information about how identification accuracy may be increased by integrating Mel Spectrograms and MFCCs into a single model.

Hiatt developed a Convolutional Neural Network (CNN) architecture using a dataset of bird recordings from Nevada and California. With several layers of 2D convolutions, MaxPooling, ReLU activation, dropout layers to prevent overfitting, and a final dense layer using softmax activation, this architecture was meticulously constructed [4]. The model's validation

accuracy of 19.27% shows how important architectural decisions are to the performance of models intended to identify bird sounds.

Together, this research highlights the opportunities and difficulties associated with using deep learning methods for bird sound recognition. The investigation of CNNs and the incorporation of other spectral properties, such as MFCCs and Mel Spectrograms, are essential to the field's advancement and direct the methodology of our research toward the development of more efficient and trustworthy instruments for bird species identification.

3. Dataset Information

To improve the audio analysis of bird species identification, we make use of the large amount of data that Xeno-canto provides. The extensive collection of bird audio recordings on this community-driven platform comes from both professional and amateur ornithologists worldwide.

3.1 Dataset Specifications

With a concentration on bird sounds captured in California and Nevada [4], we have carefully chosen a selection of recordings from Xeno-canto to meet the unique criteria of our research. This carefully selected dataset is essential for our deep learning models' accurate training:

- **Species Diversity:** The dataset includes recordings from 91 different bird species. Each species is represented by approximately 30 sound samples, ensuring a comprehensive variety of vocal patterns.
- **Sample Size and Format:** It comprises a total of 2,730 sound samples, all provided in MP3 file format. This common audio format allows for easy handling and processing in our analytical framework.
- **Range of Durations:** The lengths of these recordings vary widely, from less than 1 second to up to 195 seconds, capturing the full spectrum of possible bird call durations.
- **Total Duration:** The combined duration of all the audio samples amounts to 20 hours, 25 minutes, and 8 seconds (73,508 seconds). This extensive collection of audio data is invaluable for the depth and breadth it adds to the training process.

This dataset's substantial volume and diverse range of audio recordings provide a robust foundation for our models. By training on such varied data, our system is better equipped to accurately identify and generalize bird species based on their calls, enhancing both the accuracy and reliability of the identification system.

4. Methodology

AvianNET's technique is a methodical way of processing, analyzing, and modeling bird call data to identify species. Pre-processing, feature extraction, deep learning modeling, and Evaluation Metrics and Experiments are its three essential phases. Figure 1 depicts the process of these steps, which is further explained as follows:

4.1 Audio Data Pre-processing

The initial step of pre-processing is loading our system with the raw bird call data obtained from Xeno-canto. To prepare the audio files for feature extraction, several data cleaning and conditioning procedures are involved in this stage. To guarantee consistency throughout the dataset, pre-processing operations include segmenting calls, normalizing audio levels, and reducing noise.

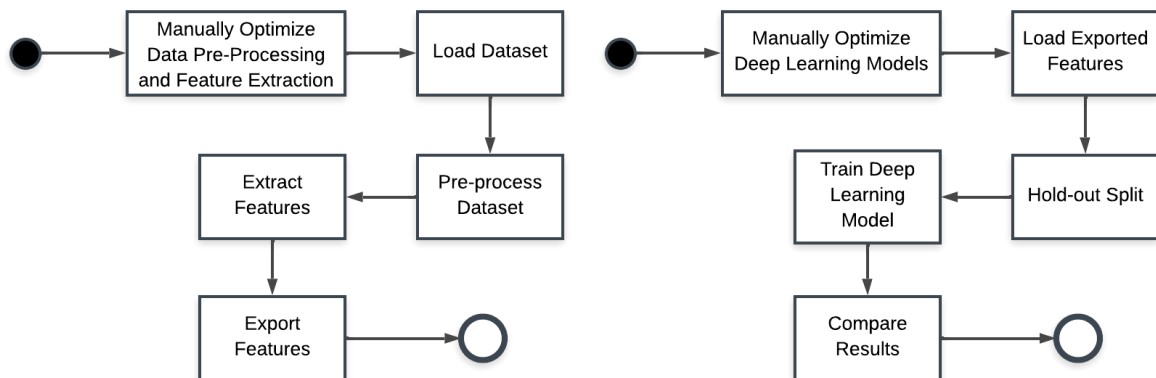


Fig. 1. Deep learning model.

To guarantee the accuracy and consistency of the characteristics that will be retrieved to identify bird species, pre-processing audio data is an essential step. Below is a summary of the steps we have used:

- **File Conversion and Normalization:**

- Audio files were initially in MP3 format and varied in sample rates between 22.5kHz and 44.1kHz.

- All files were converted to WAV format for uniformity and normalized to a sample rate of 22.5kHz to align with the lowest original sample rate. This standardization is crucial despite the increase in data size—from 1.4 GB (MP3) to 9 GB (WAV).
- **Channel Conversion and Noise Reduction:**
 - With multithreading support to expedite processing, the audio channels were converted from Stereo to Mono.
 - An envelope filter was then applied to remove low-amplitude noise, isolating the significant portions of the audio likely to contain bird calls.
- **Signal Enhancement and Bandpass Filtering:**
 - Padding was added to signals, as illustrated with the Black-tailed Gnatcatcher species example, where 1.5 seconds is appended to both the start and end of the signal for a 3-second window length.
 - The Butterworth bandpass filter [4], with frequency cuts set at 1500Hz (low) and 8000Hz (high), was utilized to eliminate frequencies out of the typical bird vocalization span. This step was particularly effective in attenuating background noise, such as wind or rain, which significantly improved the clarity of bird sounds both visually and audibly.

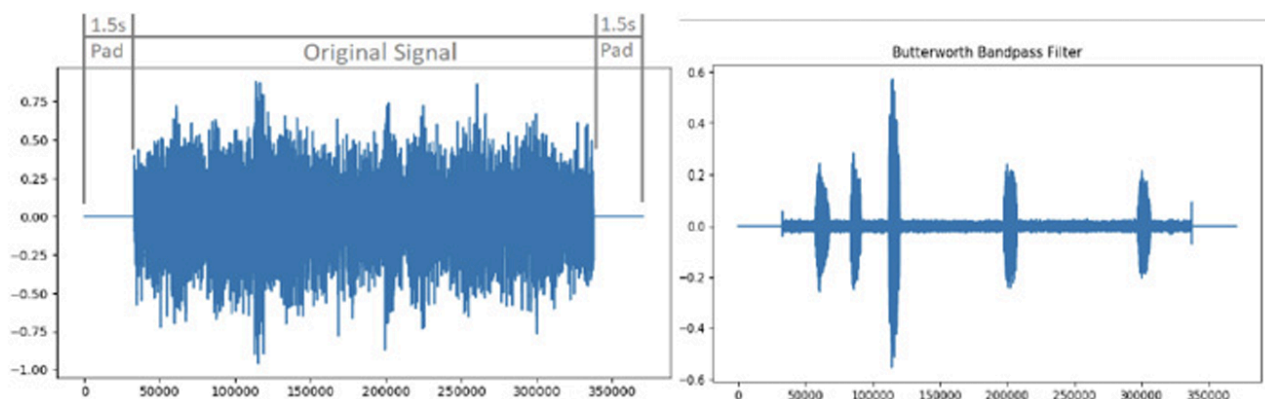


Fig. 2. Butterworth bandpass filter Pass through.

- **Splitting and Peak Detection:**

- The splitting process starts by identifying crests in the audio using SciPy's peak finding function, which enables adjustment of parameters such as the minimum gap between peaks and peak threshold.
- Once peaks are identified, the audio data are segmented into multiple 3-second samples centered around these peaks, ensuring a uniform length and focus on the most prominent vocalizations.

The audio data pre-processing processes have been carefully planned to improve the quality of the dataset for the feature extraction and deep learning classification phases that follow. By putting these strategies into practice, we are building the foundation for a solid system for identifying bird species that work well in loud, complicated auditory situations. All of the processes—from file conversion to noise reduction—have been carried out to separate distinct and unambiguous bird vocalizations, which are crucial to the proper operation of our model. This painstaking preparation guarantees the best quality of data input into our deep learning models, laying the groundwork for a dependable and advanced identification tool.

4.2 Feature Extraction

We used two main feature extraction techniques, Mel Frequency Cepstral Coefficients (MFCCs) and Mel spectrograms, to capture the unique qualities of bird vocalizations.

4.2.1 MFCC Extraction:

We utilized the `python_speech_features` [5] library to extract MFCCs, which are widely used in speech and audio processing to capture the spectral properties of sound. The function's parameters were configured as follows: `sample_rate`, which is the rate at which the audio is sampled; `numcep`, representing the number of cepstral coefficients to calculate (set to 13 by default); `nfilt`, the no. of filters in the filter bank (set to 26 by default); and the size of the FFT window, `nfft`, derived as `round(sample_rate / 40)`. Additionally, we set the lowest and highest frequencies of the mel filters to 1500Hz and 8000Hz, respectively, to focus on the range most relevant to bird calls. The code for MFCC extraction is as follows:

Python code:

```
from python_speech_features import mfcc

def get_mfccs(time_series, sample_rate):
    nfft = (round(sample_rate / 40))
    return mfcc(time_series, sample_rate, numcep=13, nfilt=26, nfft=nfft, lowfreq=1500, highfreq=8000).
```

4.2.2 Mel Spectrogram Extraction:

For extracting Mel spectrograms, the LibROSA [6] library was employed as another widely used tool for audio analysis in Python. The Mel spectrogram parameters include `sample_rate`, `n_fft` (the FFT window's length, set to 1024 by default); `hop_length`, specifying the total number of samples between successive frames (set to 1024 for our purposes); and `n_mels`, which determines how many Mel bands to produce (defaults to 128). We also configured the lowest (`fmin`) and highest (`fmax`) frequencies to 1500Hz and 8000Hz, respectively, to match the bird vocalization frequency range. The corresponding Mel spectrogram extraction code is:

Python code:

```
from librosa.feature import melspectrogram

def get_melspectrogram(time_series, sample_rate):
    return melspectrogram(y=time_series, sr=sample_rate, n_fft=1024, hop_length=1024, n_mels=128, htk=True, fmin=1500, fmax=8000)
```

4.2.3 Visualization and Saving Features:

To visualize and further analyze the extracted features, the Matplotlib [7] library was used. This visualization process helps in understanding the distinct patterns of the bird calls and ensuring the effectiveness of the feature extraction process. The features obtained from the audio segments are then saved as JPEG files for further use in training the deep learning models.

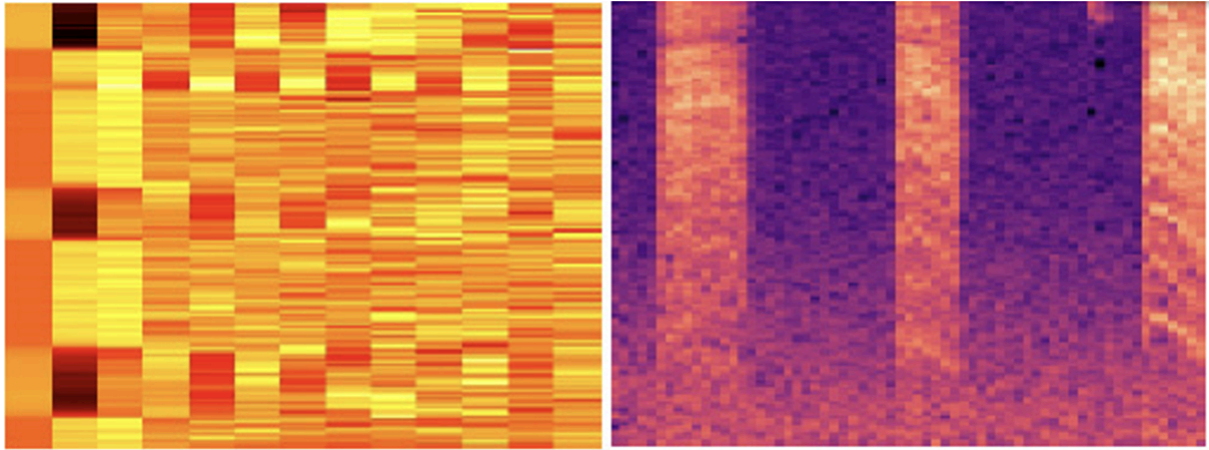


Fig. 3. Extracted MFCC (left) and mel spectrogram (right).

The careful selection and customization of these feature extraction parameters are integral to the success of the AvianNET system, optimizing the recognition and classification of bird species from audio recordings.

4.3 Deep learning model architectures

The deep learning architectures used in this research are created using the high-level Keras API [9] for Python, which is part of the TensorFlow framework [8]. This option offers a strong and adaptable framework for designing and refining various model architectures. The final models, which are derived from different existing architectures, including those in Adams' collection [10], are tailored and optimized for the particular job of classifying bird sounds. The network design underwent modifications that involved layer additions, dimension adjustments, and hyperparameter fine-tuning. For instance, our CNN model uses batch normalization layers to increase stability and, in contrast to the more popular relu activation, uses a tanh activation function in the first Conv2D layer to better fit the properties of our data.

4.3.1 Convolutional Neural Network (CNN)

MaxPooling2D follows the six Conv2D layers in the CNN architecture in each case. Table 1 lists the levels and configuration of this design. Batch normalization layers, placed between the convolutional and pooling layers, allow for normalization of the inputs to each layer, accelerating learning and preventing overfitting [11]. Following the last pooling layer, a

flattened layer transitions to a 50% dropout layer to counteract overfitting. The design ends with a dense layer that uses softmax activation to handle multi-class classification. This architecture has a total of 9,33,211 training parameters.

	Type	Kernel	Notes	Input shape
1	Reshape		TimeDistributed , Target = -1	224*2224*3
2	Dense	64	TimeDistributed ; Activation = tanh	224*672
3	GRU	128	BiDirectional	224*64
4	Dense	64	Activation = relu	224*256
5	MaxPoolingID			224*64
6	Dense	32	Activation = relu	112*64
7	Flatten			112*32
8	Dropout		Rate = 0.5	3584
9	Dense	32	Activation = relu	3584
10	Dense	91	Activation = softmax	32

Table 1. Convolutional neural network Model architecture.

4.3.2 Convolutional Recurrent Neural Network-Gated Recurrent Unit (CRNN-GRU)

The CRNN-GRU model combines convolutional and GRU layers to maximize spatial and temporal feature extraction, which is critical for evaluating audio sources. In comparison to the CRNN-LSTM model, our CRNN-GRU model doubles the kernel units in the GRU layers and the penultimate dense layer, as shown in Table 2. The use of GRUs seeks to improve model performance while also increasing computing efficiency. This model has a whopping 150,65,915 trainable parameters.

	Type	Kernel	Notes	Input shape
1	Conv2D	32	Activation = relu	224*224*3
2	MaxPooling2D			224*224*32
3	Conv2D	64	Activation = relu	112*112*32
4	MaxPooling2D			112*112*64
5	Conv2D	128	Activation = relu	56*56*64
6	MaxPooling2D			56*56*128
7	Conv2D	256	Activation = relu	28*28*128
8	MaxPooling2D			28*28*256
9	Dropout		Rate = 0.5	14*14*256
10	Reshape		TimeDistributed , Target = -1	14*14*256
11	GRU	512	BiDirectional	14*3584
12	MaxPooling1D			14*1024
13	Dropout		Rate = 0.5	7*1024
14	GRU	256	BiDirectional	7*1024
15	MaxPooling1D			7*512
16	Flatten			3*512
17	Dropout		Rate = 0.5	1536
18	Dense	64	Activation = relu	1536
19	Dense	91	Activation = softmax	64

Table 2. CRNN–LSTM model architecture.

4.4 Evaluation Metrics and Experiments

We used holdout validation to analyze the performance of our deep learning models. The dataset was randomly divided into two sets: a training set (80% of the data) and a validation set (the remaining 20%). The models' efficacy was assessed using two main metrics:

- **Accuracy:** Accuracy is a general indicator of the model's performance across all classes and is defined as the ratio of correct predictions to total predictions made.

$$\text{Accuracy} = (\text{Correct Predictions}) / (\text{Total Predictions}) \text{ Eq.(1)}$$

- **Recall:** also referred to as sensitivity, quantifies the model's capacity to identify all pertinent instances of a given class, thereby gauging its ability to identify genuine positives among the real positive cases.

$$\text{Recall} = (\text{True Positives}) / (\text{True Positives} + \text{False Negatives}) \text{ Eq.(2)}$$

These metrics are computed for each bird species, allowing for a thorough examination of the model's performance across the dataset's many classifications.

The evaluation of our models involved a series of trials utilizing the following configurations:

- **CNN with MFCC and Mel Spectrogram:** This configuration evaluates a convolutional neural network's capacity to classify bird species based on data from MFCC and Mel Spectrograms.
- **CRNN-GRU with MFCC and Mel Spectrogram:** This experiment builds on the CNN architecture by using Gated Recurrent Units to better capture temporal relationships in audio data.

The tests were carried out in a controlled computer environment, using the following specifications:

- The **operating system** is Focal Fossa's Ubuntu 20.04 LTS.
- The **tensorFlow** version is 2.2 and the **Python version** is 3.6.
- AMD Ryzen 9 3900X CPU
- 32 GB of **RAM**.

- **GPU:** AMD RX 580X 8 GB, utilizing RadeonOpenCompute 3.5.0 to enhance GPU performance.

This computer configuration offered a stable environment for training and assessing deep learning models, ensuring that the experimental findings were reliable and reproducible.

5. Results

Table 3 presents the results of our experiments, which demonstrate the efficacy of different deep-learning architectures and feature extraction techniques. The Mel Spectrogram with 3-second sample durations was the better feature extraction method when compared to MFCCs, yielding higher recall and accuracy. The most successful model for identifying bird species on our dataset was the Convolutional Recurrent Neural Network (CRNN) with Gated Recurrent Unit (GRU) layers, outperforming the other tested architectures.

We compared our best-performing methodology to a previous approach by Hiatt [3], using the same dataset with a similar holdout split (33% for testing). It is critical to emphasize the lack of information on the exact training and testing cases utilized in Hiatt's study. Despite this, the comparison in Table 4 shows that our strategy surpassed the previous one, with an accuracy of 44.26% on the validation set vs Hiatt's 19.27% [3]. This significant improvement demonstrates the efficacy of our methods.

Table 3 shows a comprehensive comparison of extracted features and model topologies, with the highest recall and least loss on the validation dataset. The Mel Spectrogram features, with a 3-second window, were utilized in combination with the CRNN-GRU model to obtain the maximum recall of 50.17% and accuracy of 53.13% on the validation set, with equivalent training set recall and accuracy of 90.64% and 92.70%.

Model	Feature Type	Sample Length	Validation Set Recall	Validation Set Accuracy	Validation Set Loss	Training Set Recall	Training Set Accuracy	Training Set Loss
CNN	MFCC	3s	40.50%	45.41%	3.040	98.82%	99.13%	0.0787
CRNN-GRU	MFCC	3s	44.40%	47.81%	3.470	90.80%	92.79%	0.3498
CNN	MFCC	1.5s	38.82%	44.44%	3.134	98.19%	98.76%	0.1116

CRNN-G RU	MFCC	1.5s	43.35%	46.47%	3.539	90.82%	92.64%	0.3476
CNN	Mel Spectrogram	3s	47.51%	43.05%	2.574	99.05%	99.37%	0.0761
CRNN-G RU	Mel Spectrogram	3s	50.17%	53.13%	3.003	90.64%	92.70%	0.3659
CRNN-G RU	Mel Spectrogram	1.5s	47.92%	50.71%	3.286	89.74%	91.95%	0.3935

Table 3. Comparing the architecture combinations of extracted features and deep learning models.

Table 4 also compares the outcomes with the least loss on the validation set between our work and Hiatt's 2019 study:

- **This Work:** Achieved 44.26% accuracy with a loss of 2.657, and 70.74% accuracy across 85 epochs of training data.
- **Hiatt 2019:** Reported 19.27% accuracy with a loss of 3.605, and 20.44% accuracy on the training set.

Implementation	At minimum loss (Validation set)			
	Validation set			Training set
	Accuracy	Loss	Epoch	Accuracy
This Work	44.26%	2.657	6	70.74%
Hiatt 2019	19.27%	3.605	85	20.44%

Table 7. Comparison with Previous Approach by Hiatt (2019).

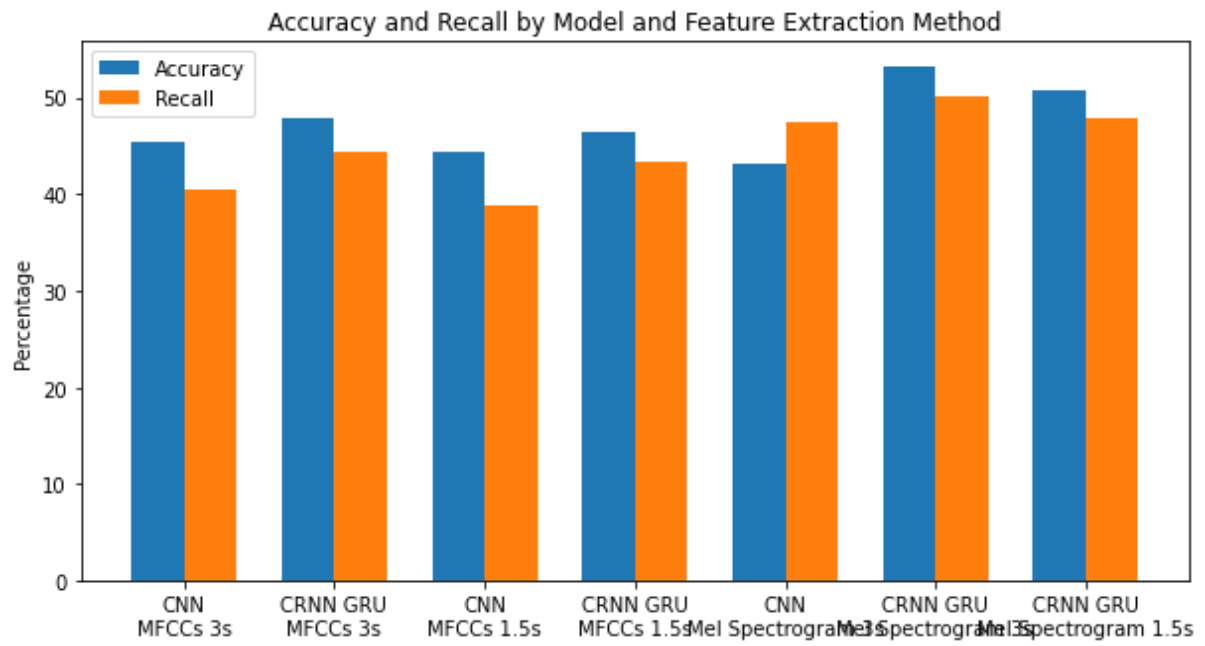


Fig 4. Evaluating Model Performance: Metrics and Experimental Results

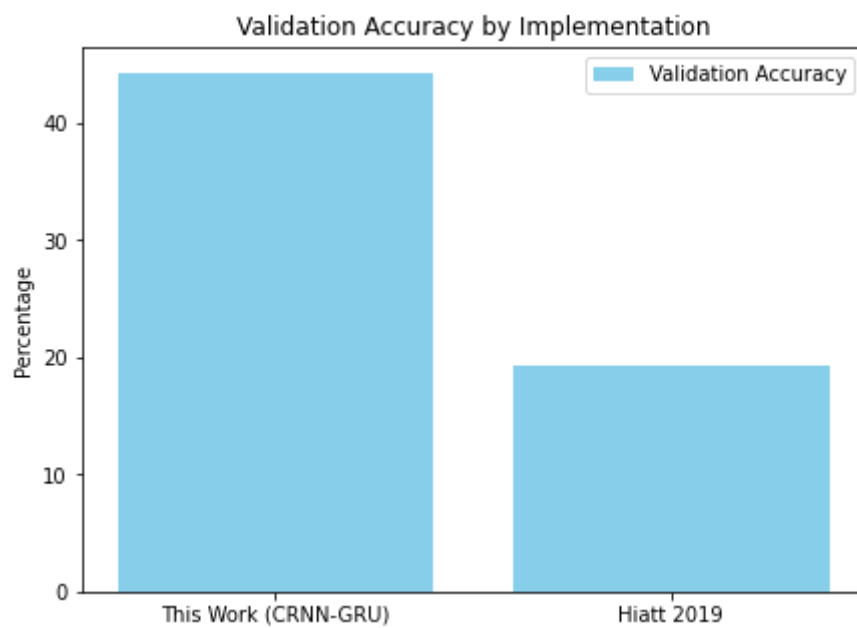


Fig 5. Validation Graph of CRNN- GRU and Hiatt 2019

```
In [9]: audio, sample_rate = librosa.load("/kaggle/input/birdclef-2023/train_audio/afghor1/XC156639.ogg")
sample_rate, wav_data = ensure_sample_rate(audio, sample_rate)
Audio(wav_data, rate=sample_rate)
```

Out[9]:

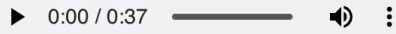


Fig 6. Demonstration how to load a single training sample from the dataset.

```
[10]: fixed_tm = frame_audio(wav_data)
logits, embeddings = model.infer_tf(fixed_tm[:1])
probabilities = tf.nn.softmax(logits)
argmax = np.argmax(probabilities)
print(f"The audio is from the class {classes[argmax]} (element:{argmax} in the label.csv file), with probability of {probabilities[0][argmax]}")
```

The audio is from the class afghor1 (element:46 in the label.csv file), with probability of 0.5590327382087708

Fig 7. Predictions

6. Conclusion

The purpose of this study was to use cutting-edge deep-learning techniques to create a dependable and accurate methodology for identifying different kinds of birds based just on their chirps. We developed an extensive audio pre-processing protocol that includes audio normalization, noise reduction above the frequency range of bird vocalization, and seamless sample segmentation based on syllable or peak detection. Mel Spectrograms and Mel Frequency Cepstral Coefficients (MFCCs) were the two techniques we used to extract features. The training of deep-learning models to identify the species of birds from an audio recording is based on these extracted properties.

Our methodology included an optimization procedure that involves iterative testing and improvement of both pre-processing techniques and deep learning models. Our investigations focused on a variety of deep learning architectures, including, Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Convolutional Recurrent Neural Networks (CRNNs). The CRNN model, which included LSTM layers, Gated Recurrent Unit (GRU) layers, and the CNN, produced the most promising results.

In comparison to Hiatt's earlier technique [3], our methodology outperformed it, achieving an accuracy of 44.26% on the validation set versus Hiatt's 19.27%. Looking ahead, we intend to improve our audio pre-processing strategy by investigating other ways for data preparation and augmentation, as well as experimenting with different feature extraction methods. Further optimization of deep learning models, including multiple topologies and hyperparameter modifications, will be a priority. To close the gap between validation and training outcomes and prevent overfitting, we will investigate methodologies like transfer learning [12].

Future efforts will focus not just on refining the approaches we've established, but also on exploring new possibilities in both pre-processing and model creation to further advance the area of audio-based bird species identification.

References

1. 1. R. Aishwarya et al. "Cognizance the Action of Human by Applying Enhanced Techniques". In: 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). 2020, pp. 1– 8. doi: 10. 1109 / ICSCAN49426 . 2020.9262419.
2. S. Kahl et al., Overview of BirdCLEF 2019: Large-scale bird recognition in soundscapes, CLEF, 9– 12 September 2019, Lugano, Switzerland.
3. S. Hiatt, Avian Vocalizations- Report, Kaggle, <https://www.kaggle.com/samhiatt/avian-vocalizations-report>.
4. S. Butterworth et al., On the theory of filter amplifiers, Wireless Eng. 7(6) (1930) 536– 541.
5. J. Lyons, jameslyons /pythonspeechfeatures: release v0.6. 1, Zenodo, doi :10.5281/zenodo.3607820, <https://odr.chalmers.se/handle/20.500.12380/249467> (2020).
6. B. McFee et al., librosa/librosa: 0.7.2, doi: 10.5281/zenodo.3606573 (2020).
7. J. D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. 9(3) (2007) 9095, doi: 10.1109/MCSE.2007.55.
- 8.
9. M. Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org, <https://www.tensorflow.org/> (2015).
10. F. Chollet et al., Keras, <https://keras.io> (2015).
11. S. Adams, Audio-Classification, <https://github.com/seth814/AudioClassification/tree/2f0032d81dcfa3d662cab1c1c4e7e30520f7edd6>, last accessed 7 June 2020.

12. J. Xie, C. Ding, W. Li, and C. Cai, Audio-only bird species automated identification method with limited training data based on multi-channel deep convolutional neural networks, arXiv:abs/ 1803.01107 (2018).