Computing and Informatics, Vol. 42, 2023, 1255–1280, doi: 10.31577/cai_2023_5_1255

MULTI-LABEL BIRD SPECIES CLASSIFICATION USING SEQUENTIAL AGGREGATION STRATEGY FROM AUDIO RECORDINGS

Noumida Abdul Kareem

College of Engineering Trivandrum
APJ Abdul kalam Technological University, Thiruvananthapuram, India
e-mail: noumidaa@gmail.com

Rajeev Rajan

Government Engineering College, Barton Hill,Thiruvananthapuram
APJ Abdul kalam Technological University, Thiruvananthapuram, India
e-mail: rajeev@cet.ac.in

Abstract. Birds are excellent bioindicators, playing a vital role in maintaining the delicate balance of ecosystems. Identifying species from bird vocalization is arduous but has high research gain. The paper focuses on the detection of multiple bird vocalizations from recordings. The proposed work uses a deep convolutional neural network (DCNN) and a recurrent neural network (RNN) architecture to learn the bird's vocalization from mel-spectrogram and mel-frequency cepstral coefficient (MFCC), respectively. We adopted a sequential aggregation strategy to make a decision on an audio file. We normalized the aggregated sigmoid probabilities and considered the nodes with the highest scores to be the target species. We evaluated the proposed methods on the Xeno-canto bird sound database, which comprises ten species. We compared the performance of our approach to that of transfer learning and Vanilla-DNN methods. Notably, the proposed DCNN and VGG-16 models achieved average F1 metrics of 0.75 and 0.65, respectively, outperforming the acoustic cue-based Vanilla-DNN approach.

Keywords: Multi-label, sequential, augmentation, recurrent neural network, convolutional neural network, transfer learning

1 INTRODUCTION

Over the last few decenniums, significant research efforts have been devoted to automatic speech analysis. However, there has recently been an upsurge in the study towards the automated analysis of animal and avian vocalizations. Bird detection is critical for avian biodiversity conservation because it allows ornithologists to count the number of birds in a particular location. A bird may listen to other birds and determine whether they are neighbours or strangers, kin or non-kin.

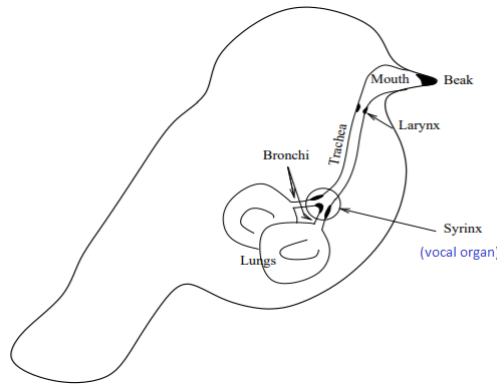
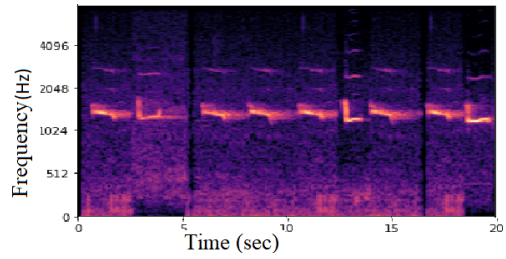


Figure 1. Avian sound production [1]

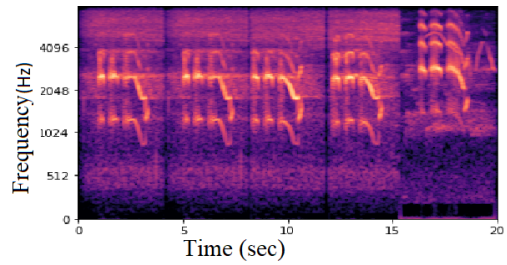
Figure 1 illustrates the avian sound production system [1]. In birds, the sound production mechanism consists of the lungs, bronchi, syrinx, trachea, larynx, mouth, and beak [1]. The vocal tract modulates airflow from the lungs as it traverses through the bronchi to the syrinx. The trachea is made of complete cartilage rings. Complete C-shaped cartilage rings with open ends against each other make up the bronchial elements. Airflow causes the syringeal medial tympaniform membrane (MTM) in each bronchus to vibrate nonlinearly opposite the cartilage wall during a bird's song. In birds, the mouth functions as a cavity resonator in the same way that it does in humans, but it is less flexible. Both bird calls and songs are examples of bird vocalization.

Bird songs are more intricate vocalizations than bird calls, which are thought to be simple[1]. Typically, songs are produced spontaneously by the male. The hierarchical levels of a bird's song consist of phrases, syllables, and elements. When a bird changes the order of the phrases in the songs, it can generate diverse singing types. Bird calls are short and are produced by connecting a series of sounds [1]. Figure 2 depicts the vocalization patterns of Eurasian Owl and Red-wattled Lapwing through mel-spectrograms.

[1] https://en.wikipedia.org/wiki/Bird-vocalization



a)



b)

Figure 2. Mel-spectrograms of vocalizations of Eurasian Owl a), Red. Lapwing b)

Traditional field methods for tracking and identifying various bird species require significant human labour. The Global Biodiversity Information Facility (GBIF)[2], which builds biological multimedia databases, also works on automatic species identification from field recordings. Acoustic bird monitoring is an effective strategy since most birds communicate primarily through vocalizations [2]. Some of the speech and audio processing techniques for the recognition of bird calls can be referred to in [3, 4, 5].

Despite weather noise and a wide variety of bird call types, machine learning approaches, particularly deep learning, can obtain very high recognition rates on remote monitored auditory data [6]. There have been numerous endeavours in the literature to classify birds, from pre-segmented acoustic single-label audio recordings [7, 8, 9, 10, 11]. Multi-label bird classification is difficult because of the time-frequency overlapping in the audio recordings. A bag generator is proposed to convert an audio recording into a bag-of-instances representation, followed by a multi-instance multi-label (MIML) classifier to forecast the set of species present in the recording [12]. It is formulated as a problem in the MIML framework for supervised classification. A multi-label classification model for finding simultaneous auditory patterns in long-duration recordings is proposed in [13]. Some of the previous works in multi-label bird call classification include [14, 15, 16, 17].

[2] `https://www.gbif.org`

The algorithm in [18] employs a deep learning technique based on convolution layers to predict the most dominant foreground species in an acoustic scenario. While forecasting the major species of each sound file, the network design yields a mean average precision (MAP) score of 0.686. The efficacy of various CNN-derived features for detecting bird vocalization is explored in [19]. Due to the difficulties in acquiring annotated training sounds, the utilization of transfer learning in CNN might be advantageous in bird call classification. An efficient CNN-based transfer learning approach for bird-call identification is explored in [20]. A particularly challenging task of bio-acoustic classification pertains to detecting overlapping events in an acoustic scene. In this context, our work aims to discern both multiple simultaneous and isolated bird vocalizations in audio recordings.

The main contributions of the paper are:

1. The sequential aggregation strategy has been implemented effectively on MFCC and mel-spectrogram features for bird call identification.
2. We present a DCNN model for mel-spectrogram inputs and systematically compare its performance with RNN, Vanilla-DNN, and transfer learning schemes.
3. SpecAugment-based data augmentation scheme (time masking, frequency masking and time warping) has been implemented for creating additional training files for the network.

A brief overview of the multi-label classification system is provided in Section 2. The performance evaluation, including the detailed dataset description, is explained in Section 3, followed by the result analysis in Section 4. Finally, the paper is concluded in Section 5.

2 SYSTEM DESCRIPTION

We proposed two sequential aggregation models for multi-label bird species classification, namely, the Acoustic RNN/DNN models and the mel-spectrogram CNN models. Features namely, mel-frequency cepstral coefficients (MFCC) and mel-spectrograms are extracted. MFCC features are queried to Acoustic RNN/DNN models (LSTM, GRU and Vanilla-DNN), whereas mel-spectrogram features to mel-spectrogram CNN models (pre-trained models, proposed DCNN) for classification. Multi-label classification is performed using a novel sequential aggregation strategy. The scheme for detecting the vocalization is illustrated in Figure 3.

2.1 Acoustic RNN Models

MFCCs find extensive application in diverse audio classification tasks driven by human perception [21], serving as predictors of timbre similarity perception [22]. MFCC converts the raw audio data into a compact and informative representation that captures the relevant information in the signal while removing irrelevant or redundant information. Gated Recurrent Unit (GRU) and Long Short-Term Memory

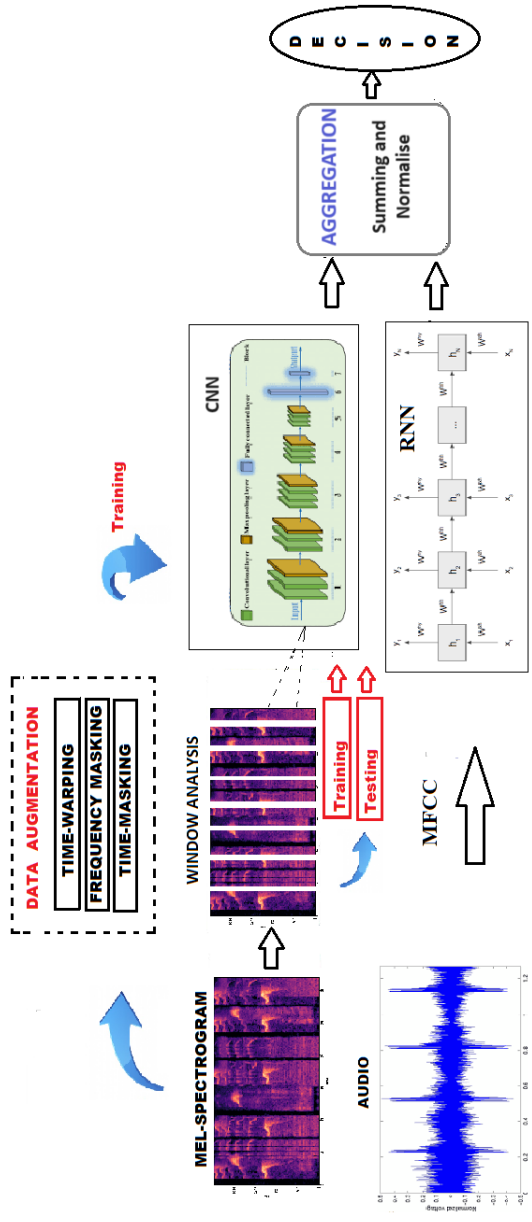


Figure 3. Block diagram: proposed multi-label bird classification using sequential aggregation strategy

(LSTM) stand out as two popular variants of Recurrent Neural Networks (RNN) that possess long-term memory capabilities. By effectively storing past inputs within their internal states and leveraging this historical context to target vectors, these networks excel at processing sequential data and retaining valuable memories. The present study delves into the examination of LSTM and GRU, exploring their capacity to capture long-range dependencies and learn temporal patterns using MFCCs. These models were trained using our multi-label dataset and implemented using sequential aggregation strategy.

2.1.1 Sequential LSTM

The LSTM cell, a recurrent network node equipped with an input, output, and a forget gate to mitigate vanishing gradients [23, 24], is harnessed to efficiently capture temporal patterns inherent in audio, as illustrated in Table 1. By leveraging frame-wise computed MFCC, LSTM effectively taps into the sequential nature of the data. LSTM can memorise previous data and predict the future with the aid of the information stored in the memory [25].

No.	**Output Shape**	**Description**
1	(None, 64, 1 048)	LSTM, 1 048 hidden nodes
2	(None, 64, 728)	LSTM, 728 hidden nodes
3	(None, 432)	LSTM, 432 hidden nodes
4	(None, 432)	Dropout, 0.2
5	(None, 10)	Dense, 10 hidden nodes

Table 1. LSTM architecture

An LSTM can be mathematically represented as follows:

$$u_i = \tanh(\xi_{xu} * x_i + \xi_{hu} * h_{i-1} + d_u), \tag{1}$$

$$r_i = \sigma(\xi_{xr} * x_i + \xi_{hr} h_{i-1} + d_r), \tag{2}$$

$$f_i = \sigma(\xi_{xf} * x_i + \xi_{hf} * h_{i-1} + d_f), \tag{3}$$

$$o_i = \sigma(\xi_{xo} * x_i + \xi_{ho} * h_{i-1} + d_o), \tag{4}$$

$$c_i = r_i u_i + f_i c_{i-1}, \tag{5}$$

$$h_i = \tanh(c_i o_i), \tag{6}$$

$$output_{class} = \sigma(h_i * \xi_{outpara}), \tag{7}$$

where u_i, r_i, f_i, o_i, c_i represents update equations for input gate, forget gate, output gate, cell state and cell output, respectively. ξ_{xu}, ξ_{xr}, ξ_{xf}, ξ_{xo} and ξ_{hu}, ξ_{hr}, ξ_{hf}, ξ_{ho}, $\xi_{outpara}$ are weights, and d_u, d_r, d_f, d_o are biases to be computed during training. The output of a neuron at time i is denoted as h_i, and elementwise multiplication is represented by '.'. The activation functions employed in our model are $\sigma(.)$ for the

sigmoid function and $tanh(.)$ for the hyperbolic tangent function. x_i denotes the input feature vector at time i. $output_{class}$ is the classification output.

2.1.2 Sequential GRU

In the realm of sequence modelling techniques, GRU emerges as the latest addition, succeeding RNN and LSTM, and thereby holding the promise of enhancing various sequential processing applications. RNNs have gained widespread adoption in language recognition [26] due to their ability to process sequential data effectively. For deep sentence processing, different cell types have been devised to improve neural networks' ability to capture long-term dependencies. The fundamental difference between GRU and LSTM neural network architectures resides in their gate structures. GRU, characterized by its simplicity, features two primary gates: the reset gate and the update gate. In contrast, LSTM, a more complex architecture, incorporates three distinct gates into its design. The GRU cells exhibit comparable power to LSTM cells [27], especially for smaller data sets, while requiring fewer computational resources.

No.	**Output Shape**	**Description**
1	(None, 64, 1 048)	GRU, 1 048 hidden nodes
2	(None, 768)	GRU, 728 hidden nodes
3	(None, 10)	Dense, 10 hidden nodes

Table 2. GRU architecture

The governing equations for GRU is presented as follows [28]:

$$z_i = \sigma_g(W_z.x_i + U_z.h_{i-1} + b_z), \tag{8}$$

$$r_i = \sigma_g(W_r.x_i + U_r.h_{i-1} + b_r), \tag{9}$$

$$\hat{h_i} = \phi_h(W_h.x_i + U_h.(r_i \odot h_{i-1}) + b_h), \tag{10}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \hat{h_i}. \tag{11}$$

Here, the variables x_i, h_i, $\hat{h_i}$, g_i, and f_i represent the input, output, candidate activation, update gate, and reset gate vectors, respectively. The matrices W, U, and b represent parameter matrices, while σ_g and ϕ_h denote the activation functions. The symbol $\odot$ denotes the Hadamard product, and '.' signifies elementwise multiplication. The GRU architecture utilized in the proposed work is conveniently summarized in Table 2.

2.2 Mel-Spectrogram CNN Models

A mel-spectrogram is a visual representation that shows how the frequencies of a signal change over time [29, 30]. It uses a special filter called a mel-scale filter bank

to highlight the frequencies most important for human perception. The mel scale was developed to scale frequency data in a manner that more closely resembles how humans perceive sound. Mels are units on the mel scale, and a reference frequency of 1 000 Hz, 40 dB above a listener's threshold is defined as 1 000 mels. The number of mels associated with a tone closely corresponds to its frequency below 500 Hz. Above 500 Hz, the number of mels between pitches perceived as "evenly spaced" increases as frequency increases. The mel-spectrogram is a smoothed spectrogram with highly emphasized low-frequency components. Here mel-spectrogram is computed with 128 bins and a frame size of 40 ms and a hop size of 10 ms. Figure 4 depicts mel-spectrograms of two audio files containing two and three species.

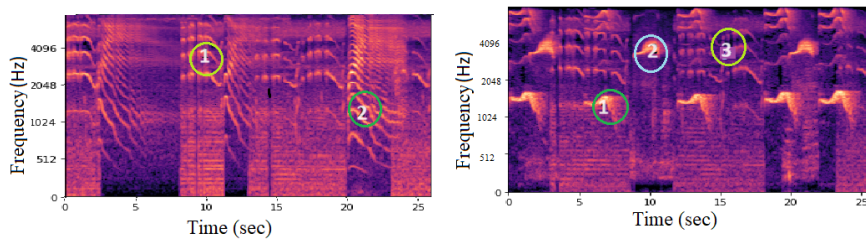


Figure 4. Mel-spectrogram of bird's vocalization with multiple bird sounds in a single audio recording. Repetitive patterns in the vocalization are shown in circles for 2 species a) and 3 species b).

CNN has been extensively used as one of the representation learning methods that enable a machine to automatically detect the representations or patterns required for classification tasks [31]. We utilized two CNN architectures for the sequential aggregation model. One is based on transfer learning models and the other one is based on a proposed DCNN shown in Table 3. Data augmentation techniques such as time-warping, frequency masking, and time-masking are used to create additional mel-spectrograms during the training phase of the network

2.2.1 Sequential DCNN

The architecture shown in Table 3 is used in the proposed analysis. The resulting $432 \times 1\,008$ arrays of the mel-spectrograms after data preparation were fed into the CNN model. The model comprises multiple convolutional layers with 3×3 kernels, 'same' padding, and 'glorot_uniform' kernel initialization, followed by Leaky ReLU activation (LeakyReLU (alpha = 0.33)). Maxpooling layers with 3×3 pool size and stride 3 are inserted after each pair of convolutional layers. Dropout layers with a 0.25 dropout rate follow maxpooling layers. After every two layers, the number of channels for the convolution layer is increased by a multiple of two, from 32 to 256. The model concludes with a GlobalMaxPooling2D layer to reduce spatial

dimensions, two dense layers (Dense(1 024) and Dense(10, activation = 'sigmoid')) for classification, and Leaky ReLU activation in the first dense layer.

No.	**Input Shape**	**Description**
1	$3 \times 432 \times 1\,008$	Mel-spectrogram
2	$32 \times 434 \times 1\,010$	$32 \times 3 \times 3$ Conv1
3	$32 \times 436 \times 1\,012$	$32 \times 3 \times 3$ Conv2
4	$32 \times 145 \times 337$	3×3 Maxpooling, Dropout (0.25)
5	$64 \times 147 \times 339$	$64 \times 3 \times 3$ Conv3
6	$64 \times 149 \times 341$	$64 \times 3 \times 3$ Conv4
7	$64 \times 49 \times 113$	3×3 Maxpooling, Dropout (0.25)
8	$128 \times 51 \times 115$	$128 \times 3 \times 3$ Conv5
9	$128 \times 53 \times 117$	$128 \times 3 \times 3$ Conv6
10	$128 \times 17 \times 39$	3×3 Maxpooling, Dropout (0.25)
11	$256 \times 19 \times 41$	$256 \times 3 \times 3$ Conv7
12	$256 \times 21 \times 43$	$256 \times 3 \times 3$ Conv8
13	$256 \times 1 \times 1$	GlobalMaxPooling2D
14	1 024	Fully connected (Flatten)
15	1 024	Dropout (0.5)
16	10	Sigmoid

Table 3. Proposed DCNN architecture

The equations for the Leaky ReLU activation (LReLU) and sigmoid activation are:

Leaky ReLU:

$$\mathrm{LReLU}(a) = \begin{cases} a, & \text{if } a > 0, \\ 0.33a, & \text{if } a \leq 0, \end{cases}$$

$$\mathrm{Sigmoid}(a) = \frac{1}{1 + \exp(-a)}.$$

These activations introduce non-linearity in the model. The Leaky ReLU helps mitigate the vanishing gradient problem, while the sigmoid activation in the final layer maps the model's output to a range between 0 and 1 for multi-class classification.

To learn a non-linear function from input to output that generalises well and yields good classification accuracy on unknown data, CNN heavily relies on the availability of massive amounts of training data [32]. Data augmentation, which involves deforming a set of annotated training files to produce additional training data, is an elegant solution to this problem. We adopted SpecAugment [33] as a data augmentation technique for the proposed scheme.

2.2.2 Sequential Transfer Learning

In transfer learning, a model created for one application is customized for another task. It is a popular deep learning approach that commences with pre-trained models for pattern recognition and computer vision tasks. We experimented with five pre-trained networks for the proposed task, namely VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and Efficient-NetB3. These models were re-trained using our multi-label dataset and implemented using sequential aggregation strategy. The details of the baseline pre-trained models are presented in Table 4.

No.	**Model**	**Parameters**	**Layers**	**Activation**
1	**InceptionResNetV2**	56 M	164	ReLU
2	**InceptionV3**	23.9 M	48	ReLU
3	**VGG-16**	138 M	16	ReLU
4	**Efficient-NetB3**	12 M	300	ReLU
5	**ResNet50**	25.6 M	50	ReLU

Table 4. Model description (M-Million)

VGG-16. The deep convolutional VGG-16 model is retrained in this experiment to detect multiple species. Table 5 shows the VGG-16 architecture, including 13 convolutions and three fully connected layers. The convolution layers are all 3×3 layers, with the same padding and stride size of 1, and the pooling layers are all 2×2 layers, with a stride size of 2. After data preparation, the resulting $432 \times 1\,008$ arrays of the mel-spectrograms are resized to 256×256 pixels. Before the fully connected layers, the last feature map has 512 channels and is flattened into a vector with 32 768 values. Finally, the dense layer with 4 096 neurons is used to add the fully connected layers, followed by a dropout layer with a value of 0.5. The proposed VGG-16 architecture for our experiment requires ten classes. The sigmoid function is chosen in the output layer.

ResNet50. Residual Networks (ResNet) are a type of deep neural network commonly used as the framework for many computer vision applications. ResNet50 is a 50-layer DCNN architecture with 48 convolutional layers, one maxpooling layer, and one average pooling layer. It is a variant of the ResNet model that uses residual learning [20]. To solve the vanishing gradient problem, the concept called residual network architecture is introduced. ResNet50 uses skip connections to propagate the activations to reduce the impact.

InceptionV3. The InceptionV3 is a 48-layer deep CNN architecture. Convolutions, poolings, dropouts, and fully connected layers make up the model. Sigmoid loss is computed and batch normalization is used throughout the model [34].

InceptionResNetV2. InceptionResNetV2, a 164-layer deep CNN architecture based on the Inception family's design but with residual linkages, is a variation of InceptionV3. The number of auxiliary classifiers is reduced from three to two.

No.	**Input Shape**	**Description**
1	$3 \times 256 \times 256$	Conv 3×3 ($\times 2$), Stride $= 1$
2	$64 \times 256 \times 256$	Maxpooling 2×2, Stride $= 2$
3	$64 \times 128 \times 128$	Conv 3×3 ($\times 2$), Stride $= 1$
4	$128 \times 128 \times 128$	Maxpooling 2×2, Stride $= 2$
5	$128 \times 64 \times 64$	Conv 3×3 ($\times 3$), Stride $= 1$
6	$256 \times 64 \times 64$	Maxpooling 2×2, Stride $= 2$
7	$256 \times 32 \times 32$	Conv 3×3 ($\times 3$), Stride $= 1$
8	$512 \times 32 \times 32$	Maxpooling 2×2, Stride $= 2$
9	$512 \times 16 \times 16$	Conv 3×3 ($\times 3$), Stride $= 1$
10	$512 \times 16 \times 16$	Maxpooling 2×2, Stride $= 2$
11	32 768	Fully connected (Flatten)
12	4 096	Dropout (0.5)
13	4 096	Dropout (0.5)
14	10	Sigmoid

Table 5. VGG-16 architecture

EfficientNet. EfficientNet is a CNN model that uses a compound coefficient to scale all width/depth/resolution dimensions uniformly. There are eight models in the EfficientNet architecture group, ranging from B0 to B7. Each model number denotes a variant with greater precision and a greater number of parameters [35]. To significantly reduce calculation costs while maintaining accuracy, EfficientNet divides the original convolution into two stages: depthwise and pointwise convolution. Because of its linear bottleneck architecture, which uses linear activation in the final layer of each block to prevent data loss from ReLU, the network is efficient.

2.3 Vanilla-DNN Model

A Vanilla-DNN framework is also used for performance comparison. The Librosa Python library is used to compute 40-dimensional MFCCs in the front-end.

Our model has two hidden layers, each with 256 perceptrons, followed by the ReLU activation function and a 50 % dropout. The sigmoid activation function is chosen in the output layer. The model is trained using our multi-label dataset and implemented using sequential aggregation strategy.

3 PERFORMANCE EVALUATION

3.1 Data Set

Recordings of the bird species are collected from the Xeno-canto website [36][3]. We standardized all the files to a minimum sampling rate of 16 kHz because the original

[3] `https://www.xeno-canto.org` (Xeno-canto)

files' sample rates ranged from 16 kHz to 44.1 kHz.

Details of scientific names and the number of Xeno-canto files used for each bird species are illustrated in Table 6. Table 7 gives the dataset specification after pre-processing. The train set contains 1 078 isolated audio files of 10 species. The files are refined such that one vocalization of 1.5 s duration is in each audio file. HC (House Crow-111), MD (Mallard Duck-106), AK (Asian Koel-121), EO (Eurasian Owl-107), HS (House Sparrow-100), BJ (Blue Jay-109), RL (Red-wattled Lapwing-104), GG (Grey Go-away-109), IP (Indian Peafowl-103), and WW (Western Wood-Pewee-108) are among the birds featured. The names and number of audio files after pre-processing are indicated in brackets. The test set contains 434 audio files that contain overlapping and multiple calls, often consisting of 2 or 3 distinct vocalizations.

No.	**Code – Scientific Name**	**# XC**	**Specialities**
1	HC – Corvus splendens	27	loud, cawing call, "caw"
2	MD – Anas platyrhynchos	25	low pitch "quacks", "whistles", "grunts", "hank"
3	AK – Eudynamys chinensis	26	loud, repetitive cuckoo-like call, "cooing"
4	EO – Bubo bubo	25	deep haunting, hooting call, "hoo-hoo-hoo"
5	HS – Passer domesticus	24	cheerful, trilling, chirping call, "chip"
6	BJ – Cyanocitta cristata	27	"jay jay" or "scold-call", "chak", "wheeoo"
7	RL – Vanellus vanellus	24	loud, wailing call
8	GG – Corythaixoides concolor	19	loud, honking, clear-territorial call, "kwaa"
9	IP – Pavo cristatus	29	piercing screams, "gobbling", "peacock"
10	WW – Contopus sordidulus	24	loud, clear whistle, peenting "pee-a-wee"

Table 6. Details of Xeno-Canto (XC) files

No.	**Class**	**Count (Bird Files)**	**# Calls**
1	Audio Files (Train)	1 078	1 078
2	Audio Files (Test)		
	Calls with two species	334	668
	Calls with three species	100	300
	Total	1 512	**2 046**

Table 7. Dataset specification

The selection is based on some below-mentioned rules. Firstly, the ten selected species represent a broad range of bird call patterns well-defined in previous

works [37, 38]. The bird call structures mainly consist of chirp, whistle, block, warbles, and click. This renders the proposed system to satisfy the generic requirement. Secondly, the selected species should have adequate samples to train and test the proposed method.

3.2 Data Augmentation

As cited in [39, 40], CNN's efficacy is highly reliant on abundant data for achieving superior results, which can be limited when dealing with small data sizes. To address this limitation and enhance the training process, data augmentation techniques are employed. In the proposed scheme, SpecAugment [33] serves as the augmentation strategy, involving the masking of frequency channels and time frames within the mel-spectrogram image representation. This augmentation includes time warping, frequency masking, and time masking of mel-spectrograms, as illustrated in Figure 5.

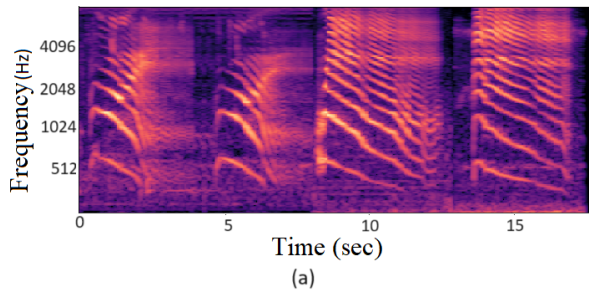
A log mel-spectrogram, comprising τ time steps, can be visualized as an image, with the time axis running horizontally and the frequency axis running vertically. Within the time steps ranging from W to $(\tau - W)$, a point randomly selected along the central horizontal line of the mel-spectrogram can undergo a warp to the left or right, covering a distance of W. Additionally, this visualization incorporates frequency masking and time masking techniques, where certain frequency bands and time segments are selectively masked. To enrich the train set, we generated 8 923 mel-spectrograms for the DCNN model and an additional 3 344 mel-spectrograms as augmented data for the transfer learning models [33].

3.3 Sequential Aggregation Strategy

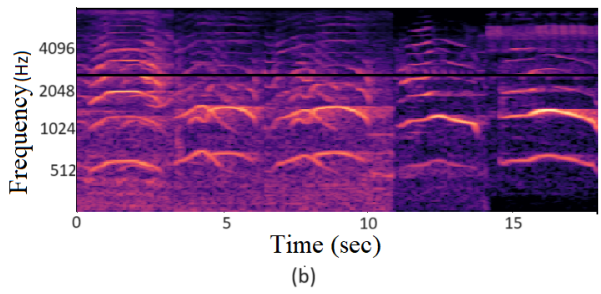
Audio recordings are sliced into fixed-length segments. For acoustic models, MFCCs of sliced audios are extracted and fed to RNN/DNN models. Sliced mel-spectrograms are utilized for training and testing with DCNN and pre-trained models. The model trained on ten classes is used to predict the probability of ten bird species. The trained neural network then predicts the probability of each species present in a segment. We used an aggregation strategy to decide on the test data. Since multiple species exist per audio clip, multiple sigmoid outputs from slices are aggregated and normalized. The nodes corresponding to the highest probability values are considered the target species. Figure 6 illustrates the schematic of the sequential aggregation process used in all the proposed methods.

3.4 Experimental Framework

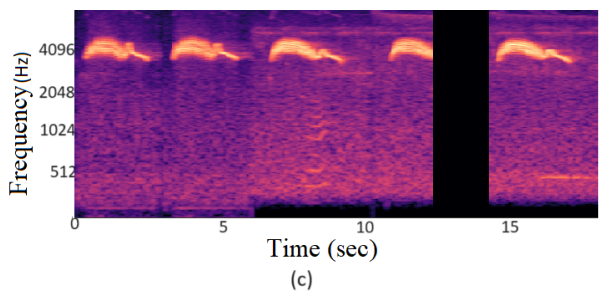
MFCC and mel-spectrogram features are extracted using Librosa Python package. Proposed DCNN model, pre-trained models (VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and EfficientNetB3), DNN and RNN-based models (GRU and



a) Time warped



b) Frequency masked



c) Time masked

Figure 5. SpecAugment-generated mel-spectrograms of bird calls

LSTM) are implemented using the novel sequential aggregation strategy. Additionally, some existing models in the literature are also implemented using sequential aggregation strategy. All these models were trained on a Google Colab notebook.

Audio files are converted to a time-frequency representation using short-time Fourier transform (STFT) with 480 samples for the window. The mel-spectrogram is segmented into small duration chunks and fed to DCNN. Adaptive moment estimation (Adam) was used in the training process of the network for optimizing the

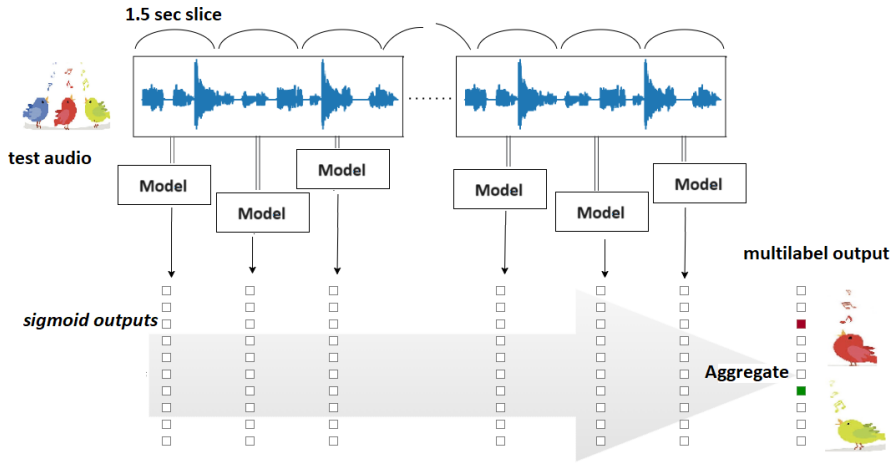


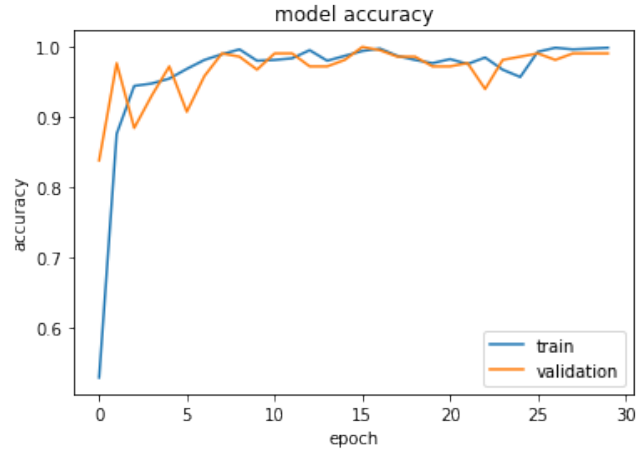
Figure 6. Schematic of window sliding technique used in the experiments

categorical cross-entropy between predictions and targets. After each maxpooling layer, the training was regularized with dropouts at 0.25. The sigmoid activation function was used and the model was trained for a maximum of 25 epochs with a batch size of 64. In this study, we retrained five deep CNN models: VGG-16, ResNet50, InceptionV3, InceptionResNetV2, and EfficientNetB3. After hyperparameter tuning, all transfer learning models are trained with a batch size of 32 for 30 epochs using the Adam optimizer. The softmax activation function is chosen in the output layer. This work uses an acoustic DNN-based model and RNN-based methods like GRU and LSTM. The RNN models LSTM and GRU are trained using Adam optimization in 30 and 20 epochs, respectively, with a batch size of 32. During the experiment, 10 % of the corpus was used for validation.

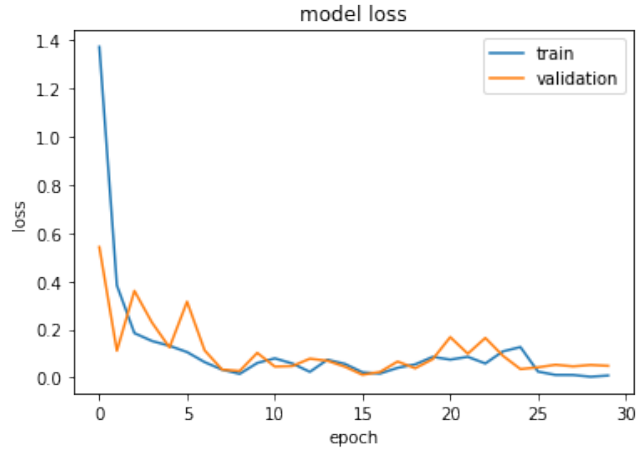
The accuracy and loss for training and validation data in LSTM model is shown in Figure 7. It can be observed from Figure 7 that the model exhibits almost stable but slightly fluctuating curves, and it achieves the highest training and validation accuracy. The accuracy curves for the training data show rapid improvement, reaching around 94 % from epoch 0 to 5, and then stabilizing at a value slightly over 100 % after epoch 25. Similarly, the accuracy curve for the validation data reaches approximately 99 %. As for the loss curves of the training and validation data, the loss quickly decreases to approximately 2 % within the first five epochs, then continues to decrease with slight fluctuations until 25 epochs before stabilizing beyond 25 epochs.

4 RESULTS AND DISCUSSION

The performance metrics with variable slicing length are shown in Figure 8. The 3 s window indeed performed worse than the shorter windows, and 0.5 s is too short



a)



b)

Figure 7. Accuracy and loss for training and validation data in LSTM model

for identifying the bird call. So, we have chosen a slicing length of 1.5 s for the performance evaluation.

The confusion matrix for the Vanilla-DNN, LSTM, VGG-16, and proposed DCNN models for the target dataset comprising two and three species are given in Tables 9, 10, 12 and 13. It is found that the proposed DCNN models outperform the VGG-16 model and the acoustic LSTM model by 10 % and 13 %, respectively. In our experiments, the best-performing DCNN model outperforms all other transfer learning schemes and acoustic models with an accuracy of 75 %.

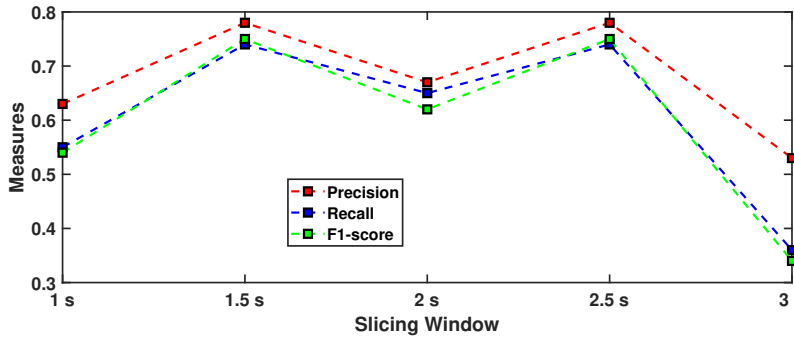


Figure 8. Performance metrics with various slicing window lengths

No.	**Species Name**	**Vanilla-DNN**			**GRU**			**LSTM**		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.76	0.77	0.77	0.90	0.86	0.88	0.86	0.62	0.72
2	Mallard Duck	0.71	0.34	0.46	0.77	0.76	0.76	0.74	0.71	0.73
3	Asian Koel	0.48	0.94	0.64	0.62	0.77	0.69	0.70	0.78	0.73
4	Eurasian Owl	0.36	0.30	0.31	0.52	0.22	0.31	0.71	0.09	0.16
5	House Sparrow	0.38	0.12	0.18	0.80	0.23	0.36	0.56	0.38	0.45
6	Blue Jay	0.62	0.72	0.67	0.37	0.71	0.48	0.44	0.74	0.55
7	Red-wattled Lapwing	0.97	0.58	0.73	0.70	0.52	0.60	0.89	0.57	0.70
8	Grey Go-away	0.51	0.83	0.63	0.48	0.76	0.60	0.50	0.87	0.64
9	Indian Peafowl	0.34	0.52	0.41	0.43	0.46	0.44	0.65	0.63	0.64
10	Western Wood-Pewee	0.96	0.84	0.90	1.00	0.50	0.67	0.93	0.77	0.84
	Macro Average	0.61	0.60	**0.57**	0.66	0.58	**0.58**	0.70	0.62	**0.62**

Table 8. Precision (P), recall (R), and F1 metric of the acoustic DNN/RNN model

The class-wise accuracy of Mallard Duck, Eurasian Owl, House Sparrow, and Indian Peafowl using the Vanilla-DNN approach is less than 50 %. In the proposed DCNN approach, however, all classes report an accuracy greater than 50 %. The proposed DCNN significantly reduced Indian Peafowl, Mallard Duck, and House Sparrow misclassification errors. The class House Sparrow has made significant progress. Compared to the Vanilla-DNN approach, the classification accuracy of three target classes, Grey Go-away, Asian Koel, and Western Wood-Pewee, decreased slightly.

Visualization of feature maps is given in Figure 9. The purpose of visualizing a feature map for a specific input image is to understand which features from the input are being detected or highlighted in these maps, as discussed in [41]. It is generally assumed that feature maps closer to the input layer capture finer details, while those closer to the model's output focus on more generalized characteristics.

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**65**	0	0	1	3	0	0	0	0	0
BJ	6	**73**	2	6	11	0	1	2	0	0
HC	5	1	**77**	1	13	0	0	2	1	0
MD	5	8	11	**30**	25	0	0	8	1	0
GG	6	13	2	0	**120**	0	0	4	0	0
RL	7	2	0	0	26	**79**	2	7	13	0
EO	9	3	2	2	5	0	**15**	14	5	0
IP	22	0	0	0	2	1	4	**33**	1	0
HS	8	18	7	2	26	1	13	27	**14**	3
WW	2	0	0	0	3	0	7	1	2	**77**

Table 9. Confusion matrix: Vanilla-DNN

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**54**	6	0	0	3	1	2	2	1	0
BJ	4	**75**	0	3	12	0	0	4	2	1
HC	0	5	**62**	7	22	0	0	2	2	0
MD	0	5	0	**63**	12	2	0	3	3	0
GG	0	16	0	0	**127**	0	0	0	2	0
RL	7	18	0	3	18	**77**	0	3	9	1
EO	1	11	5	4	13	1	**5**	7	8	0
IP	5	1	0	2	15	0	0	**40**	0	0
HS	5	32	5	3	20	6	0	0	**45**	3
WW	2	3	0	0	9	0	0	0	8	**70**

Table 10. Confusion matrix: LSTM

When filters are applied in the initial convolutional layer, it results in multiple variations of the bird call, each emphasizing different attributes. Notably, the highly activated neurons in the first layer across all species strongly indicate their primary role in identifying horizontal edges within the input mel-spectrogram, aiming to detect harmonic components. This observation is in line with our expectations and aligns with our understanding of early-stage feature extraction. To further explore these patterns, we have updated the model to visualize feature maps based on the outputs of other convolutional layers, specifically layers 1, 4, and 8, as depicted in Figure 9. The proposed deep CNN demonstrates its ability to extract more discriminative information from feature maps and effectively preserve critical edges related to multiple overlapping species within the mel-spectrogram. The distinctive spectral patterns of species like Asian Koel, Crow, and Sparrow are clearly discernible. It is worth noting that as we move deeper into the model, the feature maps progressively lose fine-grained detail, as evident from the visualizations. Although it may not be entirely clear from the final image how the model perceived the intricate patterns within the bird call mel-spectrogram, our ability to interpret these deeper feature maps diminishes.

No.	Species Name	VGG-16			ResNet50			InceptionV3		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.76	0.72	0.74	0.70	0.55	0.62	0.77	0.46	0.57
2	Mallard Duck	0.62	0.57	0.60	0.64	0.39	0.48	0.51	0.41	0.45
3	Asian Koel	0.61	0.80	0.70	0.92	0.68	0.78	0.42	0.71	0.53
4	Eurasian Owl	0.67	0.53	0.60	0.80	0.65	0.72	0.32	0.25	0.28
5	House Sparrow	0.71	0.50	0.58	0.80	0.55	0.66	0.70	0.23	0.35
6	Blue Jay	0.51	0.64	0.57	0.49	0.84	0.62	0.45	0.70	0.54
7	Red. Lapwing	0.44	0.50	0.46	0.65	0.43	0.51	0.48	0.43	0.45
8	Grey Go-Away	0.66	0.65	0.66	0.70	0.61	0.65	0.71	0.71	0.71
9	I. Peafowl	0.74	0.89	0.80	0.43	0.94	0.59	0.38	0.90	0.54
10	W. Wood-Pewee	0.82	0.72	0.77	0.48	0.71	0.57	0.86	0.60	0.70
	Macro Average	0.65	0.65	**0.65**	0.66	0.62	**0.62**	0.56	0.54	**0.51**

No.	Species Name	InceptionResNetV2			EfficientNetB3			Proposed DCNN		
		P	R	F1	P	R	F1	P	R	F1
1	House Crow	0.73	0.54	0.62	0.73	0.60	0.65	0.80	0.81	0.80
2	Mallard Duck	0.59	0.60	0.60	0.54	0.67	0.60	0.57	0.57	0.57
3	Asian Koel	0.62	0.75	0.68	0.72	0.74	0.73	1.00	0.81	0.90
4	Eurasian Owl	0.80	0.42	0.55	0.37	0.18	0.24	0.74	0.47	0.58
5	House Sparrow	0.67	0.44	0.53	0.71	0.48	0.57	0.81	0.90	0.85
6	Blue Jay	0.50	0.63	0.56	0.53	0.33	0.40	0.50	0.84	0.63
7	Red. Lapwing	0.58	0.46	0.51	0.51	0.51	0.51	0.93	0.63	0.75
8	Grey Go-Away	0.60	0.67	0.63	0.59	0.63	0.61	0.74	0.76	0.75
9	I. Peafowl	0.40	0.97	0.57	0.40	0.90	0.55	0.73	0.92	0.82
10	W. Wood-Pewee	0.89	0.60	0.71	0.65	0.68	0.67	0.95	0.68	0.80
	Macro Average	0.64	0.61	**0.60**	0.57	0.57	**0.55**	0.78	0.74	**0.75**

Table 11. Performance metrics of the proposed model and transfer learning models

	AK	BJ	HC	MD	GG	RL	EO	IP	HS	WW
AK	**55**	5	2	2	3	2	0	0	0	0
BJ	5	**65**	1	4	8	10	1	3	3	1
HC	2	1	**72**	3	8	7	2	1	3	1
MD	7	2	7	**50**	5	16	0	0	0	1
GG	2	14	6	5	**94**	8	1	1	8	6
RL	12	18	1	10	6	**67**	7	6	4	5
EO	4	5	1	1	3	5	**29**	6	1	0
IP	0	2	2	0	0	0	2	**56**	1	0
HS	3	8	2	3	12	28	1	3	**59**	0
WW	0	6	1	2	3	10	0	0	4	**66**

Table 12. Confusion matrix: VGG-16

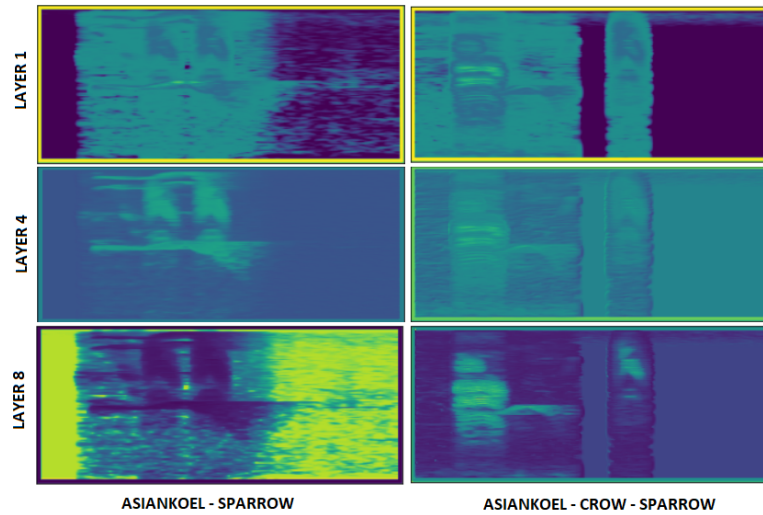


Figure 9. Visualization of feature maps with multiple species: Layer 1 (row 1), Layer 4 (row 2), Layer 8 (row 3) of Asian Koel-Sparrow (column 1), Asian Koel-Crow-Sparrow (column 2), respectively

	AK	**BJ**	**HC**	**MD**	**GG**	**RL**	**EO**	**IP**	**HS**	**WW**
AK	**56**	7	1	2	2	1	0	0	0	0
BJ	0	**85**	4	2	4	2	0	4	0	0
HC	0	8	**81**	2	1	2	0	3	3	0
MD	0	11	5	**50**	16	0	4	0	2	0
GG	0	20	1	4	**111**	0	2	1	6	0
RL	0	11	3	16	7	**86**	2	4	6	1
EO	0	7	4	4	6	0	**26**	5	3	0
IP	0	1	0	2	0	1	0	**58**	0	1
HS	0	6	1	4	0	0	0	0	**107**	1
WW	0	14	1	2	2	0	1	4	5	**63**

Table 13. Confusion matrix: Proposed DCNN

Tables 8 and 11 show the precision, recall, and F1 measure of the experiments. Overall classification scores for VGG-16, ResNet50, InceptionV3, Inception-ResNetV2, EfficientNetB3, Vanilla-DNN, GRU, LSTM, and the proposed DCNN model are 0.65, 0.62, 0.51, 0.60, 0.55, 0.57, 0.58, 0.62, 0.75, respectively. The LSTM framework's macro average precision, recall, and F1 measures are 0.70, 0.62, and 0.62, respectively, while the transfer learning-based VGG-16 model's macro average precision, recall, and F1 measures are 0.65, 0.65, and 0.65, respectively. The metrics reported by the proposed DCNN approach are 0.78, 0.74, and 0.75, respectively. The VGG-16 gives the best performance among the five pre-trained models used. The

average F1 measure for proposed DCNN, VGG-16, and Vanilla-DNN based frameworks are 0.75, 0.65, and 0.57. Compared to the Vanilla-DNN framework based on acoustic cues, there is a significant improvement in visual processing performance. Even in the case of overlapping vocalization, effective pattern learning from visual representation could be a possible cause.

It is worth noting that the proposed DCNN based architecture outperforms the Vanilla-DNN, LSTM, and VGG-16 frameworks. The Vanilla-DNN framework could not perform well for the audio files when overlapping vocalizations are present. By adopting the sequential aggregation approach, DCNN, originally designed for image classification, is adapted and fine-tuned to detect the presence of birds in audio recordings in the proposed work. The majority of the existing frameworks were refined using neural networks pre-trained on ImageNet's "trimmed" Large Scale Visual Recognition Challenge (LSVRC) [42] version, a dataset with almost 1.5 pictures of 1 000 object categories scraped from the web [43]. However, re-training the whole network, not just the final layers, is vital when fine-tuning a network originally intended for image classification.

No.	**Method**	**Precision**	**Recall**	**F1 Metric**
1	Grill and Schlüter [Model 1] [44]	0.50	0.50	0.45
2	Grill and Schlüter [Model 2] [44]	0.51	0.48	0.48
3	Efremova et al. [20]	0.61	0.54	0.53
4	Puget [45] [Transformer]	0.69	0.68	0.67
5	Yang et al. [46] [SENet]	0.65	0.58	0.58
6	Gupta et al. [47] [CNN+GRU]	0.68	0.65	0.67
7	Sequential EfficientNetB3	0.57	0.57	0.55
8	Sequential InceptionV3	0.56	0.54	0.51
9	Sequential InceptionResNetV2	0.64	0.61	0.60
10	Sequential ResNet50	0.66	0.62	0.62
11	Sequential Vanilla-DNN	0.61	0.60	0.57
12	Sequential GRU	0.66	0.58	0.58
13	Sequential LSTM	0.70	0.62	0.62
14	Sequential VGG-16	0.65	0.65	0.65
15	**Proposed Sequential DCNN**	0.78	0.74	**0.75**

Table 14. Performance comparison (Implemented using sequential aggregation and the Xeno-canto dataset)

The performance comparison of various algorithms using our multi-label dataset, assessed in terms of precision, recall, and the F1 metric, is detailed in Table 14. Grill and Schlüter [44] conducted a study comparing two approaches for detecting the presence of bird calls in audio recordings. For model 1 (Global architecture), they reported precision, recall, and an F1 metric of 0.50, 0.50, and 0.45, respectively. Model 2 (Local architecture) achieved slightly improved metrics with precision, recall, and an F1 metric of 0.51, 0.48, and 0.48, respectively. Efremova et al. [20] employed a transfer learning-based ResNet-50 model to evaluate bird call classifica-

tion. With the use of our multi-label dataset, this model achieved an F1 metric of 0.53. Puget [45] proposed an STFT Transformer, where time slices of spectrograms are used as the input patches to the ViT. Another neural network architecture, the SENet, is employed in [46] to enable the network to perform dynamic channel-wise feature re-calibration, which is also mentioned as future work in the paper [48]. The CNN+GRU part of [47] is implemented, and the system gives an F1 metric of 0.67. The F1 metric for our best-performing CNN model using sequential aggregation is 0.75, which is 30 %, 27 %, and 22 % superior to the existing models [44, 20]. It is worth noting that the proposed sequential aggregation strategy shows promise in recognition of bird vocalization in multi-label audio recordings in comparison with the existing models.

5 CONCLUSION

The issue addressed here is identifying multiple bird species from noisy or overlapping raw audio recordings. A DCNN architecture with a sequential aggregation strategy was proposed for the multi-label bird call classification task. Five different transfer learning models and an acoustic DNN/RNN based network were also implemented, and the best outcome in the test data was obtained using our proposed DCNN model. Data augmentation methods like time masking, frequency masking, and time-warping have been proposed to generate additional training data for DCNN learning. The DCNN-based scheme achieves an average F1-metric of 0.75, and it performs better than the transfer learning and acoustic approaches.

REFERENCES

[1] Fagerlund, S.: Automatic Recognition of Bird Species by Their Sound. Master Thesis. Helsinki University of Technology, Finland, 2004.

[2] Jancovic, P.—Köküer, M.: Bird Species Recognition Using Unsupervised Modeling of Individual Vocalization Elements. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 27, 2019, No. 5, pp. 932–947, doi: 10.1109/TASLP.2019.2904790.

[3] Stowell, D.—Wood, M.—Stylianou, Y.—Glotin, H.: Bird Detection in Audio: A Survey and a Challenge. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), 2016, pp. 1–6, doi: 10.1109/MLSP.2016.7738875.

[4] Gelling, D.: Bird Song Recognition Using GMMs and HMMs. Master Thesis. University of Sheffield, 2001.

[5] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Local Compressed Convex Spectral Embedding for Bird Species Identification. The Journal of the Acoustical Society of America, Vol. 143, 2018, No. 6, pp. 3819–3828, doi: 10.1121/1.5042241.

[6] Stowell, D.—Wood, M. D.—Pamuła, H.—Stylianou, Y.—Glotin, H.: Automatic Acoustic Detection of Birds Through Deep Learning: The First Bird Au-

dio Detection Challenge. Methods in Ecology and Evolution, Vol. 10, 2019, No. 3, pp. 368–380, doi: 10.1111/2041-210X.13103.

[7] Potamitis, I.—Ntalampiras, S.—Jahn, O.—Riede, K.: Automatic Bird Sound Detection in Long Real-Field Recordings: Applications and Tools. Applied Acoustics, Vol. 80, 2014, pp. 1–9, doi: 10.1016/j.apacoust.2014.01.001.

[8] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Compressed Convex Spectral Embedding for Bird Species Classification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 261–265, doi: 10.1109/ICASSP.2018.8461814.

[9] Noumida, A.—Rajan, R.: Deep Learning-Based Automatic Bird Species Identification from Isolated Recordings. 8th International Conference on Smart Computing and Communications (ICSCC), 2021, pp. 252–256, doi: 10.1109/ICSCC51209.2021.9528234.

[10] Thakur, A.—Abrol, V.—Sharma, P.—Rajan, P.: Deep Convex Representations: Feature Representations for Bioacoustics Classification. Proceedings of Interspeech 2018, 2018, pp. 2127–2131, doi: 10.21437/Interspeech.2018-1705.

[11] Rajan, R.—Johnson, J.—Abdul Kareem, N.: Bird Call Classification Using DNN-Based Acoustic Modelling. Circuits, Systems, and Signal Processing, Vol. 41, 2022, No. 5, pp. 2669–2680, doi: 10.1007/s00034-021-01896-2.

[12] Briggs, F.—Lakshminarayanan, B.—Neal, L.—Fern, X. Z.—Raich, R.—Hadley, S. J. K.—Hadley, A. S.—Betts, M. G.: Acoustic Classification of Multiple Simultaneous Bird Species: A Multi Instance Multi-Label Approach. The Journal of the Acoustical Society of America, Vol. 131, 2012, No. 6, pp. 4640–4650, doi: 10.1121/1.4707424.

[13] Zhang, L.—Towsey, M.—Xie, J.—Zhang, J.—Roe, P.: Using Multi-Label Classification for Acoustic Pattern Detection and Assisting Bird Species Surveys. Applied Acoustics, Vol. 110, 2016, pp. 91–98, doi: 10.1016/j.apacoust.2016.03.027.

[14] Noumida, A.—Rajan, R.: Multi-Label Bird Species Classification from Audio Recordings Using Attention Framework. Applied Acoustics, Vol. 197, 2022, Art. No. 108901, doi: 10.1016/j.apacoust.2022.108901.

[15] Noumida, A.—Mukund, R.—Nair, N. M.—Rajan, R.: Multi-Label Bird Species Classification Using Ensemble of Pre-Trained Networks. International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), IEEE, 2023, pp. 644–649, doi: 10.1109/ICISCoIS56541.2023.10100519.

[16] Noumida, A.—Rajan, R.: Multi-Label Bird Species Classification Using Hierarchical Attention Framework. 2022 IEEE 19th India Council International Conference (INDICON), 2022, pp. 1–6, doi: 10.1109/INDICON56171.2022.10039791.

[17] Rajan, R.—Noumida, A.: Multi-Label Bird Species Classification Using Transfer Learning. International Conference on Communication, Control and Information Sciences (ICCISc), IEEE, Vol. 1, 2021, pp. 1–5, doi: 10.1109/ICCISc52257.2021.9484858.

[18] Sprengel, E.—Jaggi, M.—Kilcher, Y.—Hofmann, T.: Audio Based Bird Species Identification Using Deep Learning Techniques. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (Eds.): Working Notes of CLEF 2016 (Conference and Labs of the Evaluation Forum). CEUR Workshop Proceedings, Vol. 1609, 2016,

pp. 547–559, https://ceur-ws.org/Vol-1609/16090547.pdf.

[19] Kahl, S.—Wilhelm-Stein, T.—Hussein, H.—Klinck, H.—Kowerko, D.—Ritter, M.—Eibl, M.: Large-Scale Bird Sound Classification Using Convolutional Neural Networks. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (Eds.): Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 1866, 2017, https://ceur-ws.org/Vol-1866/paper_143.pdf.

[20] Efremova, D. B.—Sankupellay, M.—Konovalov, D. A.: Data-Efficient Classification of Birdcall Through Convolutional Neural Networks Transfer Learning. 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1–8, doi: 10.1109/DICTA47822.2019.8946016.

[21] Seppänan, J.: Computational Models for Musical Meter Recognition. Master Thesis. Tampere University of Technology, Tampere, Finland, 2001.

[22] Richard, G.—Sundaram, S.—Narayanan, S.: An Overview on Perceptually Motivated Audio Indexing and Classification. Proceedings of the IEEE, Vol. 101, 2013, No. 9, pp. 1939–1954, doi: 10.1109/JPROC.2013.2251591.

[23] Gruber, N.—Jockisch, A.: Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text? Frontiers in Artificial Intelligence, Vol. 3, 2020, Art. No. 40, doi: 10.3389/frai.2020.00040.

[24] Tang, C. P.—Chui, K. L.—Yu, Y. K.—Zeng, Z.—Wong, K. H.: Music Genre Classification Using a Hierarchical Long Short Term Memory (LSTM) Model. In: Jiang, X., Chen, Z., Chen, G. (Eds.): Third International Workshop on Pattern Recognition. SPIE, Proceedings of SPIE, Vol. 10828, 2018, pp. 334–340, doi: 10.1117/12.2501763.

[25] Olah, C.: Understanding LSTM Networks. 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [accessed 25-Feb-2020].

[26] Géron, A.: Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'reilly Media, Inc., 2017.

[27] Chung, J.—Gulcehre, C.—Cho, K.—Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR, 2014, doi: 10.48550/arXiv.1412.3555.

[28] Cho, K.—van Merrienboer, B.—Gulcehre, C.—Bahdanau, D.—Bougares, F.—Schwenk, H.—Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. CoRR, 2014, doi: 10.48550/arXiv.1406.1078.

[29] Sukhavasi, M.—Adapa, S.: Music Theme Recognition Using CNN and Self-Attention. CoRR, 2019, doi: 10.48550/arXiv.1911.07041.

[30] Ghosal, D.—Kolekar, M. H.: Music Genre Recognition Using Deep Neural Networks and Transfer Learning. Proceedings of Interspeech 2018, 2018, pp. 2087–2091, doi: 10.21437/Interspeech.2018-2045.

[31] LeCun, Y.—Bengio, Y.—Hinton, G.: Deep Learning. Nature, Vol. 521, 2015, No. 7553, pp. 436–444, doi: 10.1038/nature14539.

[32] Salamon, J.—Bello, J. P.: Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters,

Vol. 24, 2017, No. 3, pp. 279–283, doi: 10.1109/LSP.2017.2657381.

[33] Park, D. S.—Chan, W.—Zhang, Y.—Chiu, C. C.—Zoph, B.—Cubuk, E. D.—Le, Q. V.: SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[34] Fritzler, A.—Koitka, S.—Friedrich, C. M.: Recognizing Bird Species in Audio Files Using Transfer Learning FHDO Biomedical Computer Science Group (BCSG). Vol. 1866, 2017, Art. No. 169, https://ceur-ws.org/Vol-1866/paper_169.pdf.

[35] Tan, M.—Le, Q. V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.): Proceedings of the 36th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research (PMLR), Vol. 97, 2020, pp. 6105–6114, http://proceedings.mlr.press/v97/tan19a/tan19a.pdf.

[36] Vellinga, W. P.—Planqué, R.: The Xeno-Canto Collection and Its Relation to Sound Recognition and Classification. In: Cappellato, L., Ferro, N., Jones, G. J. F., San Juan, E. (Eds.): Working Notes of CLEF 2015 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 1391, 2015, https://ceur-ws.org/Vol-1391/166-CR.pdf.

[37] Brandes, T. S.: Automated Sound Recording and Analysis Techniques for Bird Surveys and Conservation. Bird Conservation International, Vol. 18, 2008, No. S1, pp. 163–173, doi: 10.1017/S0959270908000415.

[38] Duan, S.—Towsey, M.—Zhang, J.—Truskinger, A.—Wimmer, J.—Roe, P.: Acoustic Component Detection for Automatic Species Recognition in Environmental Monitoring. 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2011, pp. 514–519, doi: 10.1109/ISSNIP.2011.6146597.

[39] Kaya, M.—Bilge, H. Ş.: Deep Metric Learning: A Survey. Symmetry, Vol. 11, 2019, No. 9, Art. No. 1066, doi: 10.3390/sym11091066.

[40] Liu, C.—Feng, L.—Liu, G.—Wang, H.—Liu, S.: Bottom-Up Broadcast Neural Network for Music Genre Classification. Multimedia Tools and Applications, Vol. 80, 2021, No. 5, pp. 7313–7331, doi: 10.1007/s11042-020-09643-6.

[41] Zeiler, M. D.—Fergus, R.: Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.): Computer Vision – ECCV 2014. Springer, Cham, Lecture Notes in Computer Science, Vol. 8689, 2013, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

[42] Russakovsky, O.—Deng, J.—Su, H.—Krause, J.—Satheesh, S.—Ma, S.—Huang, Z.—Karpathy, A.—Khosla, A.—Bernstein, M.—Berg, A. C.—Li, F. F.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, Vol. 115, 2015, pp. 211–252, doi: 10.1007/s11263-015-0816-y.

[43] Lasseck, M.: Acoustic Bird Detection with Deep Convolutional Neural Networks. Detection and Classification of Acoustic Scenes and Events 2018, 2015, pp. 143–147, https://dcase.community/documents/workshop2018/proceedings/DCASE2018Workshop_Lasseck_134.pdf.

[44] Grill, T.—Schlüter, J.: Two Convolutional Neural Networks for Bird Detection

in Audio Signals. 2017 25th European Signal Processing Conference (EUSIPCO), IEEE, 2017, pp. 1764–1768, doi: 10.23919/EUSIPCO.2017.8081512.

[45] Puget, J. F.: STFT Transformers for Bird Song Recognition. In: Faggioli, G., Ferro, N., Joly, A., Maistro, M., Piroi, F. (Eds.): Proceedings of the Working Notes of CLEF 2021 – Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 2936, 2021, pp. 1609–1616, https://ceur-ws.org/Vol-2936/paper-137.pdf.

[46] Yang, F.—Jiang, Y.—Xu, Y.: Design of Bird Sound Recognition Model Based on Lightweight. IEEE Access, Vol. 10, 2022, pp. 85189–85198, doi: 10.1109/ACCESS.2022.3198104.

[47] Gupta, G.—Kshirsagar, M.—Zhong, M.—Gholami, S.—Ferres, J. L.: Comparing Recurrent Convolutional Neural Networks for Large Scale Bird Species Classification. Scientific Reports, Vol. 11, 2021, No. 1, Art. No. 17085, doi: 10.1038/s41598-021-96446-w.

[48] Koh, C. Y.—Chang, J. Y.—Tai, C. L.—Huang, D. Y.—Hsieh, H. H.—Liu, Y. W.: Bird Sound Classification Using Convolutional Neural Networks. In: Cappellato, L., Ferro, N., Losada, D. E., Müller, H. (Eds.): Working Notes of CLEF 2019 – Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, Vol. 2380, 2019, https://ceur-ws.org/Vol-2380/paper_68.pdf.

Noumida Abdul Kareem received her B.Tech. in electronics and communication from the College of Engineering, Chengannur (Cochin University of Science and Technology, India) in 2018, and M.Tech. in signal processing from the College of Engineering, Trivandrum in 2021. She is currently pursuing her Ph.D. degree with the College of Engineering, Trivandrum. Her research areas are audio signal processing and bioacoustics.

Rajeev Rajan received his B.Tech. in electronics and communication from the College of Engineering, Adoor (Cochin University of Science and Technology, India) in 2000, M.Tech. in applied electronics and instrumentation from the College of Engineering, Trivandrum in 2004, and Ph.D. from the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, Chennai in 2017. His research areas are speech and music signal processing, image processing.