Inductive Representation Learning with LSTM Aggregator for Multi-label Detection of Avian Calls in Field Recordings

Noumida A
College of Engineering Trivandrum
APJ Abdul Kalam Technological University, Kerala, India.
noumidaa@gmail.com

Rajeev Rajan
Government Engineering College, Idukki
APJ Abdul Kalam Technological University, Kerala, India.
rajeev@cet.ac.in

***Abstract*—This paper presents a novel methodology that utilizes inductive representation learning with a long short-term memory aggregator to detect multiple avian vocalizations from field recordings. Initially, a graph is constructed from the Mel-spectrogram of the audio file using a trained deep convolutional neural network (Deep CNN). This graph is then fed into a GraphSAGE-LSTM module in the subsequent phase for classification. To enhance the training of the Deep CNN, the SpecAugment technique is employed to generate additional Mel-spectrograms. The proposed algorithm is evaluated on the Xeno-canto bird sound database, and its performance is compared to state-of-the-art models. The proposed approach outperforms existing methods and spectral graph-based models, achieving a macro F1 score of 0.90.**

***Index Terms*—GraphSAGE, long short term memory, data augmentation, multi-label bird classification, inductive representation learning**

I. Introduction

Bird recognition through vocalizations relies on speech recognition, audio classification, and pattern recognition techniques. Acoustic features play a crucial role in accurately representing bird calls, which directly impacts recognition success [1], [2]. Methods from speech and audio processing [3], [4], along with artificial neural networks [5], have been widely used for bird vocalization identification, with many studies focusing on classifying pre-segmented single-label acoustic recordings [6]–[8]. Some of the previous works based on deep learning frameworks for multi-label bird classification are [9]–[14].

Several graph-based models have been widely studied, including ChebNet [15], GraphSAGE [16], GCN [17], and GAT [18]. GCN models are notable for their semi-supervised classification using layer-wise propagation based on first-order spectral convolutions [17]. The graph neural tangent kernel (GNTK) explores node correspondences using graph topology and node features [19], while adaptive graph models [20] enhance intra-class relationships for smooth predictions. Recently, researchers [21] has also tackled multi-label classification using GCN, which introduces a relation matrix based on correlation and sparsity among samples. An end-to-end audio tagging GNN (ATGNN) [22] combines CNN-extracted local features with graph convolutions to tag audio from spectrogram-based k-nearest neighbor graphs.

Several papers [21], [23]–[28] explore the use of graph networks in audio classification. For instance, [29] introduces a subgraph-based framework incorporating self-supervision tasks. Here, subgraphs are derived by sampling from the training data, extracting pertinent features, and establishing connections between data samples based on similarity or relationship. This iterative process generates multiple subgraphs from different subsets of the training data, with the addition of random edges to streamline graph construction during inference. In [30], graphs are directly constructed from spectrograms, combining GCN with CNN features to form an ensemble approach. In [31], the initial node representations are generated from the word embeddings of the labels. Subsequently, the GCN learns final node representations, which are employed for classifying acoustic representations. Unlike [30], [31], the novelty of our method lies in the graph generation approach from Mel-spectrograms. We utilize a Deep CNN model trained on isolated bird calls to generate graphs from Mel-spectrograms of raw audio containing multiple calls, using sliding window analysis. These graphs are then processed by GraphSAGE-LSTM for classification. The key novelty of this work is the Mel_GraphSAGE-LSTM framework, which combines Deep CNN with GraphSAGE-LSTM for analysis and classification.

II. Inductive Representation Learning

Inductive representation learning refers to the process of learning representations (embeddings) for nodes in a graph, allowing for generalization to unseen data or nodes that were not present during the training phase.

A prominent example of this approach is the GraphSAGE (Graph Sample and Aggregate) model which operates in spatial domain, where the embedding for a node v at the kth layer is computed as:

$$\mathbf{h}_v^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot \mathrm{AGG}^{(k)}\left(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}\right) + \mathbf{b}^{(k)}\right) \quad (1)$$

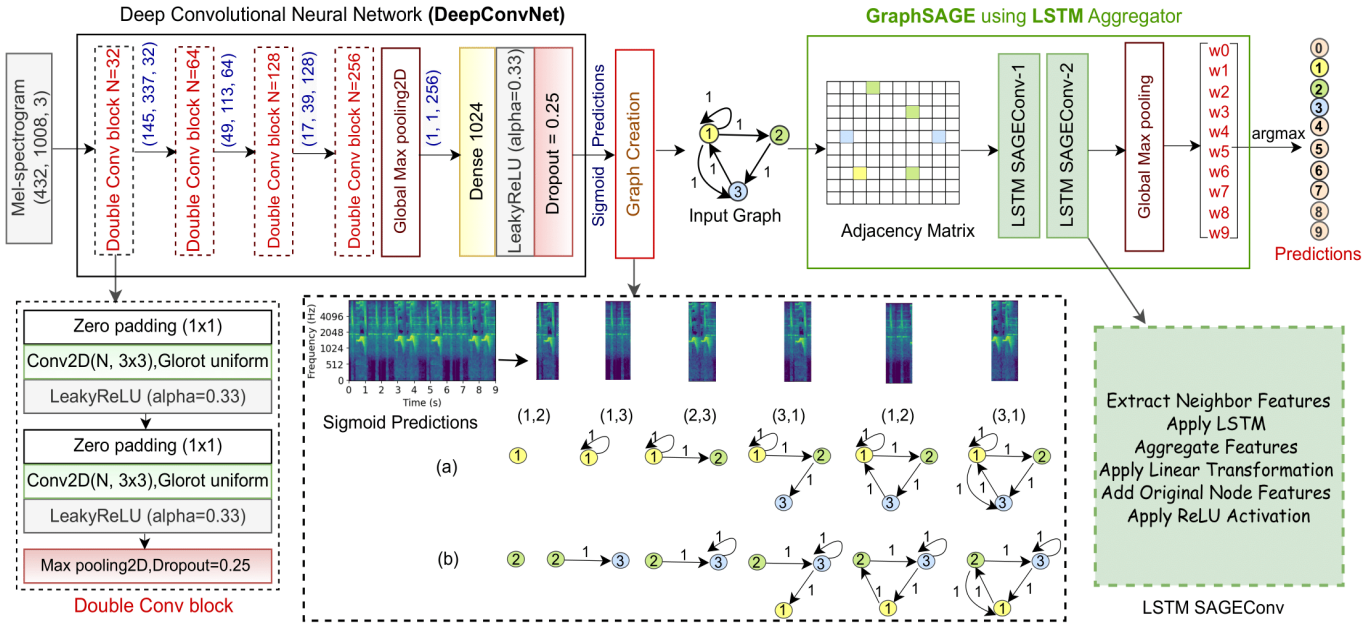

Fig. 1: The block diagram of the proposed method for multi-label bird classification

In this equation, $\mathbf{h}_v^{(k)}$ represents the embedding of node v at the kth layer, $\mathbf{W}^{(k)}$ is a learnable weight matrix, σ is an activation function (such as ReLU), and $\mathcal{N}(v)$ denotes the set of neighboring nodes of v. The AGG function can be realized using different methods like mean, LSTM and pooling aggregator, each capturing different aspects of the neighborhood information.

LSTM Aggregator: uses a Long Short-Term Memory network to process the sequence of neighboring node features, capturing more complex dependencies. The aggregation process is described as:

$$\mathbf{h}_{\mathcal{N}(v)}^{(k)} = \text{LSTM}\left(\left[\mathbf{h}_u^{(k-1)}\right]_{u\in\mathcal{N}(v)}\right) \tag{2}$$

$$\mathbf{h}_v^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot \left[\mathbf{h}_v^{(k-1)} \| \mathbf{h}_{\mathcal{N}(v)}^{(k)}\right] + \mathbf{b}^{(k)}\right) \tag{3}$$

GraphSAGE with LSTM Aggregator aggregates features from neighboring nodes using an LSTM. The LSTM processes the features of neighboring nodes and outputs aggregated features for each node. For each node, the features of its neighbors are collected, processed by the LSTM, and then aggregated into a single feature vector. The LSTM's last hidden state is used as the aggregated feature for each node.

III. Proposed Framework

We present a novel graph-based method using for multi-label bird species classification from raw audio recordings as given in Fig. 1. The detailed steps are as follows.

A. Mel_Graph Extraction

Mel-spectrograms are computed using a 30 ms frame size and a 10 ms hop size, and additional Mel-spectrograms can be generated using data augmentation [32]. The proposed deep CNN is trained with Mel-spectrograms of single-labeled audio files (each containing one isolated bird). A short-segment analysis with varying slicing length (1s, 1.5s, 2s, 2.5s, 3s) is performed on multi-label audio recordings. Among them, a slicing length of 1.5 s is empirically chosen, and Mel-spectrogram corresponding to each segment is fed to the deep CNN to identify the most probable species(see Fig. 1).

A graph G={V, E} with adjacency matrix $A(i, j)$ is constructed from the labels obtained sequentially from each segment. These labels corresponds to the most predominant bird (highest probability) at the nodes of the deep CNN. Here, i and j represent nodes of the graph. Similarly, separate graphs are generated for the second most predominant species and so on. The process is illustrated in Fig. 1 and Algorithm 1. Each node in the graph represents a bird species (label), and edges denote connections between different species. These connections are determined based on the relationships observed in the audio recording. For instance, if two segments of the audio recording contain bird calls of the same species, there will be a self-loop connecting the corresponding node in the graph, and if they are of different species, there will be a directed edge connecting the former and the latter. The weights on the edges of the graph represent the frequency of occurrence of the label. The graph and the corresponding ground truth for each audio file is used to train the GraphSAGE in the second phase

B. Classification using GraphSAGE

The model is constructed using graph convolutional layers as shown in Fig. 1. The input to the network is the graph adjacency matrix whose nodes have d-dimensional features. The d-dimensional features that are input to the GraphSAGE are obtained from the deep CNN model. When GraphSAGE uses these features to classify the graph, it takes into account

TABLE I: Precision (P), recall (R), and F1 score of the various deep learning models

No.	Species name (ID)	Puget et al. [33] Transformer			Yang et al. [34] SENet			Mel_Graph-ChebNet			Mel_Graph-GCN			Mel_Graph-SAGE-LSTM		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	House Crow (HC)	0.95	0.91	0.93	0.98	0.62	0.76	0.70	0.76	0.72	0.90	0.95	0.92	0.81	0.91	0.86
2	Mallard Duck (MD)	0.73	0.66	0.70	0.62	0.62	0.62	0.65	0.68	0.67	0.81	0.83	0.82	0.89	0.87	0.88
3	Asian Koel (AK)	0.82	0.67	0.74	0.64	0.81	0.72	0.74	0.76	0.75	0.98	0.88	0.93	0.93	0.83	0.88
4	Eurasian Owl (EO)	0.30	0.45	0.35	0.41	0.25	0.31	0.91	0.80	0.85	0.94	0.62	0.75	1.00	0.89	0.94
5	House Sparrow (HS)	0.76	0.48	0.56	0.55	0.41	0.47	0.75	0.80	0.77	0.81	0.94	0.87	0.95	0.96	0.96
6	Blue Jay (BJ)	0.68	0.44	0.54	0.46	0.78	0.58	0.73	0.68	0.70	0.64	0.86	0.74	0.88	0.72	0.79
7	Red. Lapwing (RL)	0.66	0.70	0.68	0.54	0.34	0.42	0.93	0.84	0.88	0.92	0.76	0.83	0.96	0.93	0.94
8	Grey go-away (GG)	0.69	0.80	0.74	0.47	0.85	0.60	0.76	0.81	0.78	0.87	0.83	0.85	0.81	0.96	0.88
9	Indian Peafowl (IP)	0.54	0.93	0.68	0.91	0.81	0.86	0.75	0.80	0.77	0.82	0.97	0.90	0.92	0.95	0.93
10	W.Pewee (WW)	0.85	0.80	0.83	1.00	0.33	0.49	0.90	0.81	0.86	0.96	0.78	0.86	0.93	0.88	0.90
	Macro Average	0.69	0.68	**0.67**	0.65	0.58	**0.58**	0.78	0.77	**0.77**	0.87	0.84	**0.85**	0.83	0.89	**0.90**

Algorithm 1 Graph Creation for an Audio File

Input:
  $arr[i][j]$ (size: 6×10)
  - Each column represents one of 10 species
  - Each row corresponds to a 1.5s frame within 9s
  - Entries indicate the probability of each species occurring during a time interval
Output:
  A graph with vertices as labels, edges as connections, and edge weights as their frequency of occurrence
Initialize an empty list mp
for $i = 0$ to 5 **do**
  Set $mp[i]$ to the most probable species in row $arr[i]$
  if $mp[i]$ is not in the set of graph vertices **then**
    Add $mp[i]$ as a new vertex to the graph
  end if
  if $i > 0$ **then**
    if an edge between $mp[i-1]$ and $mp[i]$ exists **then**
      Increment the edge weight between $mp[i-1]$ and $mp[i]$ by 1
    else
      Create an edge between $mp[i-1]$ and $mp[i]$ with weight 1
    end if
  end if
end for

the temporal dependencies between different segments of the audio file, which is important for capturing the complex interactions between different bird calls. This allows GraphSAGE to effectively identify patterns and relationships in the data that might not be apparent from individual segments alone.

IV. Experimental Framework

We evaluate performance using the Xeno-canto database [35]. The dataset includes 32-bit mono WAV files sampled at 16 kHz. The training set consists of 1,078 files, each 1.5 seconds long, featuring isolated vocalizations from 10 species: House Crow (111), Mallard Duck (106), Asian Koel (121), Eurasian Owl (107), House Sparrow (100), Blue Jay (109), Red Lapwing (104), Grey Go-away (109), Indian Peafowl (103), and W. Wood Pewee (108). The test set comprises 434 audio files, each 9 seconds long, containing overlapping vocalizations and multiple bird calls (334 files with 2 species; 100 files with 3 species). Additionally, we generated 3,344 Mel-spectrograms for CNN training through data augmentation techniques described in [32].

Mel-spectrograms are extracted using the `librosa` Python package. The deep CNN employs SpecAugment [32], which operates on the log Mel-spectrogram of the input audio. We use the LibriSpeech basic (LB) approach with parameters $W = 80$, $F = 27$, $T = 100$, $mF = 1$, and $mT = 1$, where W, F, and T denote warping, frequency, and time masking parameters, respectively, and mF and mT represent the number of frequency and time masks applied. The Mel-spectrograms, with dimensions of $432 \times 1008 \times 3$, are input into the deep CNN. The CNN architecture features successive double convolutional layers with filter sizes of 32, 32, 64, 64, 128, 128, 256, and 256, as depicted in Fig. 1.

A. Spectral Graph-based Models

ChebNet: Utilizes three Chebyshev Convolution (ChebConv) layers. The first layer expands node features to 32 dimensions using K=3 polynomials, the second layer maintains these dimensions, and the final layer reduces them to 1. ReLU activation is applied after each layer.
GCN Model: Consists of three GCNConv layers, each with 32 hidden dimensions and ReLU activation. It focuses on propagating node features across the graph.

B. Proposed Spatial Graph-based Model

The GraphSAGE-LSTM model integrates GraphSAGE with LSTM to handle graph-based data, using dynamic node features for prediction tasks. It includes two LSTMGraphSAGEConv layers: the first maps node features from a dimensionality of 1 to 16 (hidden channels), and the second maintains this 16-dimensional feature space. Each LSTMGraphSAGEConv layer incorporates an LSTM with an input size of 1 and a hidden size of 16, followed by a linear transformation to project the LSTM output to the 16-dimensional space. The final prediction is produced by a linear output layer that reduces the 16-dimensional feature space to a single output. Finally, Node features are aggregated using global max pool to create a graph-level representation before applying the output layer.

All the graph models are trained for up to 100 epochs with a batch size of 32, using the Adam optimizer with a learning rate of 0.01, the MSE loss function, and a sigmoid activation function at the output. 10% of the dataset is used for validation.

C. Analysis of Results

Our model reports macro average precision(P), recall(R), and F1 score as 0.83, 0.89, and 0.90, respectively in Table I. For the proposed approach, all the classes report an F1 score greater than 75% as opposed to the performance of other models [15], [33], [34]. This model shows superior results for House Sparrow (96%), Eurasian Owl (94%), Red. Lapwing (94%), Indian Peafowl (93%), and W. Peewe (90%) species in terms of F1 score. In Fig. 2, we present the Hamming loss and exact match results for the proposed classification algorithms. The Deep CNN exhibits a Hamming loss of 21.18% and an exact match of 61.86%. The GraphSAGE-LSTM model stands out with a significantly lower Hamming loss of 14.03% and a higher exact match of 84.89%. Other models, such as GCN, also show competitive performance.

We experimented with varying number of LSTM-SAGEConv blocks Table. II. It is noticed that the model provides optimum performance for two blocks and for 16 hidden dimensions, as shown in Fig. 3.

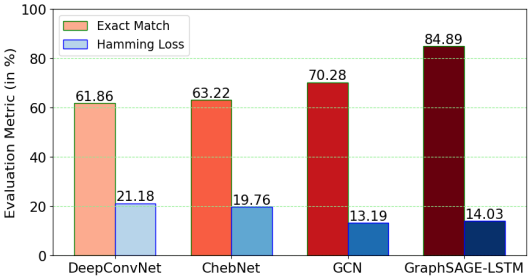

Fig. 2: Hamming loss and exact match metrics for evaluation

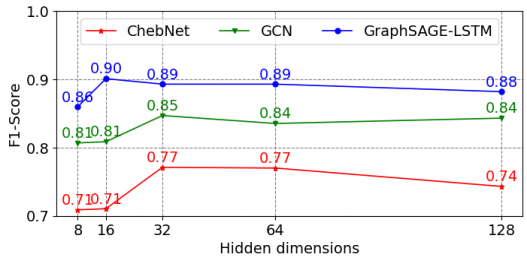

Fig. 3: Metrics with varying no. of hidden dimensions

TABLE II: Varying number of layers in graph models. Best values are highlighted

Method \ Block	2	3	4
ChebNet (K=3)	0.74	**0.77**	0.74
GCN	0.80	**0.85**	0.84
GraphSAGE-LSTM	**0.90**	0.90	0.88

The training analysis in Table III shows that GCN has 1153 parameters, while ChebNet has 3345 and GraphSAGE-LSTM has 3953 parameters. The reduced parameters in graph models compared to transformer, SENet, and Transfer learning models enhance efficiency and scalability, enabling faster training and lower computational demands, crucial for real-time processing

TABLE III: Training of the systems. M stands for Million

Method	Data	Parameters	Time
Transformer	10,000	0.26 M	9.09 hrs
SENet	1078	1.45 M	1.83 hrs
Transfer Learning	3345	24.8 M	13.09 mins
Mel_Graph-ChebNet (K=3)	587	1.4M+3345	8 mins
Mel_Graph-GCN	**587**	**1.4 M+1153**	**7.38 mins**
Mel_GraphSAGE-LSTM	**587**	**1.4 M+3953**	**7.38 mins**

TABLE IV: Performance comparison with existing methods

Method	Approach	P	R	F1
Grill et al. [Model1] [36]	CNN-Global	0.50	0.50	0.45
Grill et al. [Model2] [36]	CNN-Local	0.51	0.48	0.48
Efremova et al. [37]	Transfer Learning	0.61	0.55	0.53
Puget [33]	Transformer	0.69	0.68	0.67
Yang et al. [34]	SENet	0.65	0.58	0.58
Gao et al. [38]	Res2Net	0.61	0.61	0.60
Junyan Liu et al. [39]	SE-Protonet-DCASE	0.58	0.57	0.58
Proposed CNN only	Deep CNN	0.60	0.60	0.59
Proposed CNN only	Deep CNN (+aug)	0.78	0.74	0.75
Defferrard et al. [15]	Mel_Graph-ChebNet	0.78	0.77	0.77
CNN-GCN	Mel_Graph-GCN	0.87	0.84	0.85
CNN-GraphSAGE	**Mel_GraphSAGE**	**0.83**	**0.89**	**0.90**

on resource-constrained devices. This efficiency may also improve generalization by reducing overfitting, highlighting the importance of graph representation learning in deep learning applications.

We may be curious to see the results with deep CNN alone, without using the CNN-GraphSAGE-LSTM combo for multi-vocalization detection. From the Table IV, it is evident that the dual system outperforms the deep CNN model. We implemented all the methods given in Table IV on our multi-label dataset and reported the results. The proposed spatial-based GNN, GraphSAGE-LSTM performs better than the ChebNet-based model [15], [40]. The inferior performance of ChebNet could be attributed to learning illegal coefficients while approximating analytic filter functions, leading to over-fitting [40]. The proposed Mel_GraphSAGE-LSTM achieves an F1 score of 0.90 shows a relative improvement of 37%, 32% and 23% over the existing state-of-the art models discussed in [33], [34], [37].

V. Conclusion

This paper presents a novel Mel_GraphSAGE-LSTM framework for multi-label bird species classification, combining GraphSAGE-LSTM with a Deep CNN trained on Mel-spectrograms. The framework processes graphs created by the front-end trained Deep CNN. A SpecAugment-based augmentation scheme is used to create additional train data. Compared to recent methods and other spectral-based GNNs, the proposed spatial-based Mel_GraphSAGE-LSTM achieves the highest F1-score of 0.90 on the Xeno-canto dataset, with fewer parameters and faster training. Although spectral GNNs slightly outperform spatial GNNs in terms of parameter efficiency, the results underscore the effectiveness of the Deep CNN-GraphSAGE-LSTM approach for accurately classifying multiple bird species from audio recordings.

REFERENCES

[1] S. Fagerlund, "Automatic recognition of bird species by their sound." *Masters Thesis, Helsinki University of Technology, Finland*, 2004.

[2] I. Guyon and A. Elisseeff, "An introduction to feature extraction," *Feature Extraction: Foundations and Applications*, pp. 1–25, 2006. [Online]. Available: https://doi.org/10.1007/978-3-540-35488-8_1

[3] D. Stowell, M. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, pp. 1–14, 10 2018.

[4] D. Gelling, "Bird song recognition using GMMs and HMMs." *Masters Project Dissertation, Department of Computer Science, University of Sheffield*, pp. 1–46, 2001.

[5] T. Schrama, M. Poot, M. Robb, and H. Slabbekoorn, "Automated recording, detection and identification of nocturnal flight calls: Results of a pilot study during autumn migration in the netherlands," *Journal of Ornithology*, vol. 147, no. 5, p. 248, 2006.

[6] A. Thakur, V. Abrol, P. Sharma, and P. Rajan, "Compressed convex spectral embedding for bird species classification," *in proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 261–265, 2018.

[7] I. Potamitis, S. Ntalampiras, and K. R. Olaf Jahn, "Automatic bird sound detection in long real-field recordings : Applications and tools," *Applied Acoustics*, pp. 1–9, 2014.

[8] J. Elias, Sprengeland Martin, K. Yannic, and H. Thomas, "Audio based bird species identification using deep learning techniques." *in proc. of CLEF*, pp. 1–13, 2016.

[9] R. R. Abdul Kareem, N., "Identifying overlapping bird species from raw field audio recordings by assembling grouped channel feature attention with multi-scale residual cbam," *Neuralcomputing and applications*, vol. 42, no. 5, 2025.

[10] R. Rajan, J. Johnson, and N. Abdul Kareem, "Bird call classification using dnn-based acoustic modelling." *Journal of Circuits, Systems, and Signal Processing*, vol. 41, no. 5, pp. 2669–2680, 2022.

[11] A. Noumida, R. Rajan, and J. Thomas, "Detecting multiple overlapping birds in audio recordings using self attention-based wavelet convolutional neural network," *in proc. of Int. Conf. on Recent Advances in Intelligent Computational Systems*, 2024.

[12] A. Noumida and R. Rajan, "Stacked res2net-cbam with grouped channel attention for multi-label bird species classification." *in proc. of European Signal Processing Conference*, 2023.

[13] N. Abdul Kareem and R. Rajan, "Multi-label bird species classification using sequential aggregation strategy from audio recordings," *Computing and Informatics*, vol. 42, no. 5, p. 1255–1280.

[14] N. A and R. Rajan, "Multi-label bird species classification from audio recordings using attention framework," *Applied Acoustics*, vol. 197, p. 108901, 2022.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *in proc. of Advances in Neural Information Processing Systems*, vol. 29, pp. 1–9, 2016.

[16] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *in proc. of Advances in Neural Information Processing Systems*, vol. 30, pp. 1–11, 2017.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *in proc. of Int. Conf. on Learning Representations*, pp. 1–14, 2017.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *in proc. of Int. Conf. on Learning Representations*, pp. 1–12, 2018.

[19] A. Bayer, A. Chowdhury, and S. Segarra, "Label propagation across graphs: Node classification using graph neural tangent kernels," *in proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 5483–5487, 2022.

[20] R. Zheng, W. Chen, and G. Feng, "Semi-supervised node classification via adaptive graph," *Pattern Recognition*, vol. 124, pp. 1084–92, 2022.

[21] W.-C. Ye and J.-C. Wang, "Multilabel classification based on graph neural networks," in *Data Mining*, C. Thomas, Ed. Rijeka: IntechOpen, 2021, ch. 4.

[22] S. Singh, C. J. Steinmetz, E. Benetos, H. Phan, and D. Stowell, "Atgnn: Audio tagging graph neural network," *IEEE Signal Processing Letters*, vol. 31, pp. 825–829, 2024.

[23] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks." *in proceedings of DCASE*, pp. 143–147, 2018.

[24] H. Wang *et al.*, "Modeling label dependencies for audio tagging with graph convolutional network." *IEEE Signal Processing Letters 27*, pp. 1560–1564, 2020.

[25] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-shot audio classification with attentional graph neural networks." *in proc. of Interspeech*, pp. 3649–3653, 2019.

[26] S. Dokania and V. Singh, "Graph representation learning for audio & music genre classification." *arXiv preprint arXiv:1910.11117*, 2019.

[27] X. Li and J. Gao, "Audioset classification with graph convolutional attention model," *in proc. of Int. Joint Conference on Neural Networks*, pp. 1–6, 2023.

[28] A. Noumida and R. Rajan, "Multi-label bird species classification from field recordings using mel graph-gcn framework," *in proc. of Interspeech 2024*, pp. 4793–4797, 2024.

[29] A. Shirian, K. Somandepalli, and T. Guha, "Self-supervised graphs for audio representation learning with limited labeled data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1391–1401, 2022.

[30] C. Aironi, S. Cornell, E. Principi, and S. Squartini, "Graph-based representation of audio signals for sound event classification," *in proc. of European Signal Processing Conference*, pp. 566–570, 2021.

[31] Y. Hou *et al.*, "Audio event-relational graph representation learning for acoustic scene classification," *IEEE Signal Processing Letters*, vol. 30, pp. 1382–1386, 2023.

[32] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," *in proc. of Interspeech*, pp. 2613–2617, 2019.

[33] J. F. Puget, "STFT transformers for bird song recognition," *in proc. of CLEF (Working Notes)*, vol. 2936, pp. 1609–1616, 2021.

[34] F. Yang, Y. Jiang, and Y. Xu, "Design of bird sound recognition model based on lightweight," *IEEE Access*, vol. 10, pp. 85 189–85 198, 2022.

[35] W. Vellinga and R.Planque, "The xeno-canto collection and its relation to sound recognition and classification." *in proc. of CLEF (Working Notes)*, pp. 1–11, 2015.

[36] T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," *in proc. of European Signal Processing Conference*, pp. 1764–1768, 2017.

[37] D. B. Efremova *et al.*, "Data-efficient classification of birdcall through convolutional neural networks transfer learning." *in proc. of Digital Image Computing: Techniques and Applications*, pp. 1–8, 2019.

[38] G. Shang-Hua *et al.*, "Res2net: A new multi-scale backbone architecture." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43.2, pp. 652–662, 2019.

[39] J. Liu *et al.*, "Se-protonet: Prototypical network with squeeze-and-excitation blocks for bioacoustic event detection," *Report, Detection and Classification of Acoustic Scenes and Events Challenge*, 2023.

[40] M. He, Z. Wei, and J.-R. Wen, "Convolutional neural networks on graphs with chebyshev approximation, revisited," *in proc. of Advances in Neural Information Processing Systems*, 2022.