Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss

Anshul Thakur,[a] Daksh Thapar, Padmanabhan Rajan, and Aditya Nigam
School of Computing and Electrical Engineering, IIT Mandi, Mandi, Himachal Pradesh-175005, India

(Received 27 February 2019; revised 25 June 2019; accepted 28 June 2019; published online 29 July 2019)

Bioacoustic classification often suffers from the lack of labeled data. This hinders the effective utilization of state-of-the-art deep learning models in bioacoustics. To overcome this problem, the authors propose a deep metric learning-based framework that provides effective classification, even when only a small number of per-class training examples are available. The proposed framework utilizes a multiscale convolutional neural network and the proposed dynamic variant of the triplet loss to learn a transformation space where intra-class separation is minimized and inter-class separation is maximized by a dynamically increasing margin. The process of learning this transformation is known as deep metric learning. The triplet loss analyzes three examples (referred to as a triplet) at a time to perform deep metric learning. The number of possible triplets increases cubically with the dataset size, making triplet loss more suitable than the cross-entropy loss in data-scarce conditions. Experiments on three different publicly available datasets show that the proposed framework performs better than existing bioacoustic classification methods. Experimental results also demonstrate the superiority of dynamic triplet loss over cross-entropy loss in data-scarce conditions. Furthermore, unlike existing bioacoustic classification methods, the proposed framework has been extended to provide open-set classification.
https://doi.org/10.1121/1.5118245

[PG] Pages: 534–547

I. INTRODUCTION

Habitat destruction induced by global warming and human activities has pushed many avian and amphibian species to the brink of extinction. With this looming threat of population decline and species extinction, a large number of initiatives for wildlife conservation have been witnessed.[1,2] Surveying and monitoring are the principal steps in any conservation effort. Manual surveying and monitoring are difficult, time-consuming, and require experienced personnel.[3,4] Owing to the rich acoustic communication in birds and frogs, automated acoustic monitoring provides an appropriate way to survey different species of interest in their natural habitat and alleviates the requirement of manual monitoring.[5] The bioacoustic signal classification module is the mainstay of such acoustic monitoring systems,[6] and often includes tasks such as bird and frog species classification. The major impediment in many bioacoustic classification tasks is the scarcity of the labeled training data. Moreover, the target species and, hence, the training data requirements, often vary from one ecosystem to other. This makes it unfeasible to collect and label a large amount of bioacoustic data for all the possible species. Thus, there is a requirement for classification frameworks that could provide effective classification with a small number of labeled training examples.

In recent times, deep convolution neural networks (CNN) have become the cornerstone for achieving state-of-the-art performances in various audio classification tasks.[7–9] In comparison to the shallow learning techniques, CNNs often require a large amount of training data (subject to the task in hand) to generalize and provide effective classification. However, the scarcity of the labeled data for many bioacoustic tasks makes it undesirable to utilize these data-intensive CNNs. The lesser amount of training data often leads to over-fitting in CNNs. This over-fitting can be avoided by using regularizers and early stopping, which can restrict the modeling capabilities. Many studies on CNN-based audio classification have used data augmentation techniques to overcome the training data scarcity.[10,11] These methods augment the training data with synthetic examples that are generated by deforming the original data. Some common deformations used for data augmentation include pitch alterations and time stretching. However, these augmentation techniques are not always useful and can affect the classification performance.[11] As a result, the effectiveness of these techniques is data dependent and often requires a trial–error approach. In case of bioacoustics, coming up with the effective augmentation requires domain knowledge about the nature of vocalizations of each target species. Apart from augmentation, many studies have explored transfer learning for overcoming training data scarcity.[12,13] In the case of CNNs, existing (or pre-trained) networks trained for any audio classification tasks can be fine-tuned for achieving effective performance.[14] Fine-tuning helps in transferring the knowledge from the pre-trained network to the domain and task of interest.[15] In data-scarcity scenarios, fine-tuning an existing network is easier and more effective than training the network from scratch. Thus, CNN-based transfer learning presents an effective way to overcome the labeled training data scarcity in bioacoustic applications.

[a]Electronic mail: anshul_thakur@students.iitmandi.ac.in

In literature, CNNs have mildly been explored for different bioacoustic classification tasks. Due to the recently conducted *bird activity detection (BAD) challenges*,[16] fairly large datasets have been publicly released for the task of bird activity detection. This led to an influx of CNN-based frameworks that provide state-of-the-art performance for the aforementioned task.[17–19] However, only a few studies have addressed the task of vocalization segmentation and species identification using deep learning approaches. Lostanlen *et al.*[20] released a bird flight call detection dataset along with a CNN-based benchmark. Salamon *et al.*[21] experimentally showed that the late fusion of scores obtained from CNN (deep learning) and a random forest classifier (shallow learning) results in better performance for the task of bird species classification from flight calls. The same CNN architecture, consisting of three convolution layers and two dense layers, is used in both the aforementioned studies. Ibrahim *et al.*[22] proposed to use a recurrent neural network (RNN) and CNN to classify grouper species. Tóth and Czeba,[23] Sprengel *et al.*,[24] and Piczak[25] utilized spectrogram enhancement methods before applying CNNs to identify bird species from their songs or calls. This spectrogram enhancement helps in removing the effect of overwhelming background disturbances on the classification procedure.

Apart from deep learning, many classical machine learning techniques have successfully been utilized for bioacoustic classification. Stowell and Plumbley[26] proposed spherical K-means-based unsupervised feature learning for large scale bird species classification. Building on their work, Thakur *et al.*[27,28] proposed to use archetypal analysis[29] and deep archetypal analysis for obtaining supervised convex representations for bioacoustic classification. Kernel-based extreme learning machines are used by Qian *et al.*[30] for bird species classification. This study utilizes active learning to alleviate the problem of unlabeled bioacoustic data. Many studies have used dynamic kernels- (such as the intermediate matching kernel and probabilistic sequence kernels) based support vector machines (SVM) for different bioacoustic classification tasks such as bird activity detection and bird species classification.[31–33]

In this work, the authors propose to use CNN-based deep metric learning (DML)[34] for bioacoustic classification. DML deals with learning a mapping from the input space to a compact Euclidean space where similarity among examples is in direct correspondence with the distance among them. As a result, DML directly provides class-specific clustering. Thus, a classifier trained in this space can provide better classification than the one trained in the input feature space. This study utilizes CNN powered by the triplet loss[35–37] to map the input examples to 128-dimensional embeddings in the desired transformation space. The triplet loss processes three examples, called a triplet, at a time. A triplet consists of an anchor, a positive example and a negative example. The anchor and the positive examples belong to the same class whereas the negative example can be from any other class. A CNN with triplet loss tries to learn a transformation where a triplet constraint is imposed on all the training examples. This constraint states that the distance between negative-anchor pair should be greater than the distance between positive-anchor by a fixed margin in the transformation space. Only triplets that violate this constraint are chosen for training. More details about the triplet loss and its implementation are in Sec. II C. Triplet loss has successfully been utilized for many applications such as face recognition[35] and person re-identification.[34] It is of particular interest for bioacoustic classification due to the following reasons:

- **Effective training with less training data:** The number of triplets in a training set is cubical in terms of the number of training examples. Hence, more triplets are available for training than the number of training examples. More triplets result in more weight updates and lead to better training. Thus, in comparison to the cross-entropy loss, the triplet loss can provide effective training with lesser training examples.
- **Overcomes class imbalance:** The class imbalance has no major impact on performance of the triplet loss. This can be attributed to the nature of training procedure that learns inter-class separation by comparing the training examples of each class with other examples of other classes, one at a time. Hence, irrespective of the number of examples per class, each class is represented in the training process.

Multiscale CNN used in the proposed DML framework is characterized by the utilization of different filter sizes in the convolution layers. Each filter size helps in analyzing the input bioacoustic events at a different scale. The smaller filters help in extracting the minute local details whereas the large filters analyze a larger receptive field and help in obtaining the global details from the input bioacoustic event. This notion of multiscalar analysis is inspired by the Inception[38] model that was proposed for large scale image classification. This multiscale CNN is empowered by a dynamic variant of the classical triplet loss[35] to learn the desired transformation space. During training, the margin of the loss function is slowly increased based on a pre-defined heuristic (see Sec. II). This dynamically varying triplet loss has a twofold advantage:

(1) Starting with a smaller margin and slowly increasing the margin can be seen as warm start. First, the CNN is taught to learn a relatively simpler task of separating the examples of one class from the others in the embedding space by a smaller margin. Then, the complexity of this task is slowly increased by increasing the margin. This warm start may help in achieving better convergence even when the number of classes is very large.
(2) Dynamically varying margin increases the number of triplets used for training. It can be attributed to the fact that triplets which satisfy the triplet constraint at a lower margin can violate the constraint as the margin is increased.

The main contributions of this study are as follows:

- To the best of authors' knowledge, this is the first study that utilizes deep metric learning for bioacoustics.
- A simple multiscale CNN architecture is proposed for bioacoustic classification.

- This study experimentally shows that the utilization of triplet loss helps in overcoming the training data scarcity without utilizing any data augmentation and transfer learning.
- A dynamic variant of triplet loss is proposed.
- To the best of authors' knowledge, unlike any existing bioacoustic classification methods, the proposed framework can perform open-set classification by utilizing a simple extension.

The rest of this paper is organized as follows. In Sec. II, the proposed DML-based bioacoustic classification framework is described. Datasets, designed experiments and comparative methods used for the performance evaluation are presented in Sec. III. Experimental results obtained during the designed experiments are discussed in Sec. IV. Section V concludes this paper.

II. METHOD: PROPOSED DML FRAMEWORK

In this section, the proposed DML framework for bioacoustic classification is described. The overall design of the framework is depicted in Fig. 1. The proposed framework is composed of two neural networks: a multiscale CNN and a multilayer perceptron (MLP). The multiscale CNN, equipped with dynamic triplet loss, is used to learn a transformation from input to the embedding space. The embeddings generated by CNN are given as input to the MLP for learning the discrimination between classes.

This section starts with the feature extraction procedure. Then, the architectures of the proposed multiscale CNN and MLP are described. Later, dynamic triplet loss and other details regarding training of neural networks are highlighted. Then, the procedure to classify the bioacoustic signals using the trained DML framework is explained. Finally, the proposed framework is extended for open-set classification.

A. Feature extraction

Most bird species such as passerines are known for producing harmonically rich sounds. However, there are many species such as woodpeckers, snipes, and storks that are characterized by drumming, winnowing, clattering, and other mechanically produced sounds. These sounds are more or less percussive in nature. Thus, the difference in harmonic and percussive components of a bioacoustic sound has some class-specific characteristics. This difference in harmonic and percussive components of sounds produced by white-bellied woodpecker and Indian peafowl is evident in Fig. 2. Inspired by this observation, Mel-spectrogram along with its harmonic and percussive components[39] are given as a three-channel input to the proposed framework. The spectrogram of the input audio recording is decomposed into its harmonic and percussive components using the method proposed in Ref. 39. The original, harmonic and percussive spectrograms are multiplied by Mel filterbank to obtain the respective Mel spectrograms that form the three channels of an input example. All three channels are converted to decibel scale and are normalized with respect to the maximum value.

B. Neural network designs

1. Multiscale CNN architecture

The proposed CNN consists of five convolution (*CONV*) layers, four Inception[38] inspired multiscale analysis modules, three dense layers, and has 1 286 410 trainable parameters. Each multiscale analysis module consists of seven convolution layers having different filter sizes that enable the network to analyze each input at different scales. The overall network design is illustrated in Fig. 3(a). The main components of the network are below.

a. Input. As discussed earlier, audio examples, represented by Mel-spectrograms and their harmonic and percussive components ($40 \times M \times 3$, 40 Mel-filters and M frames) are given as input.

b. Multiscale analysis module. Multiscale analysis modules (*MAM*) utilize kernels of different sizes (1×1, 3×3, 5×5, and 7×7). This multiscale analysis helps in better feature extraction from short duration vocalizations (such as birdsong syllables or flight calls) as well as from the longer vocalizations such as birdsong phrases. The shorter vocalizations occupy smaller spatial space on Mel-spectrograms as

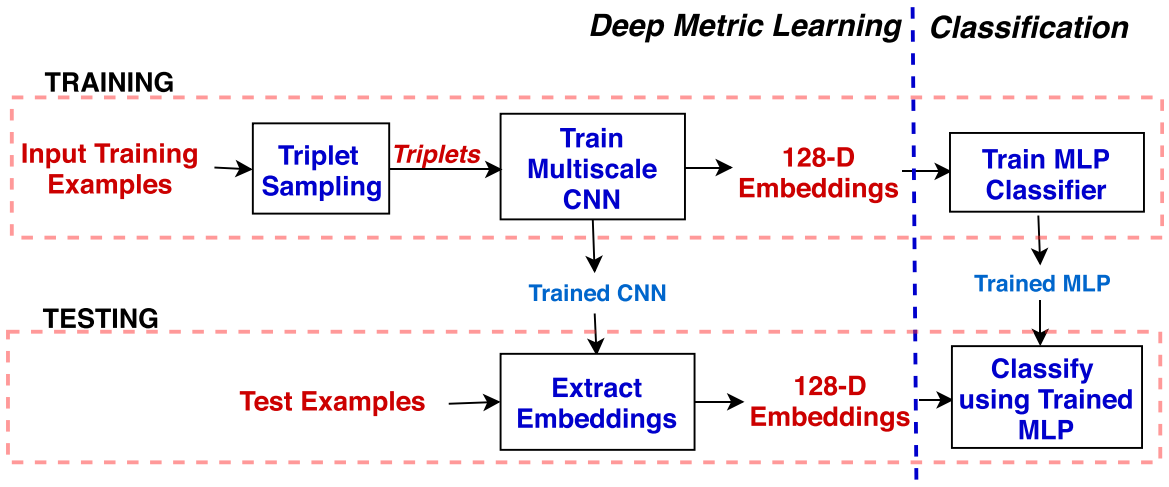

FIG. 1. (Color online) Proposed DML framework for bioacoustic classification.

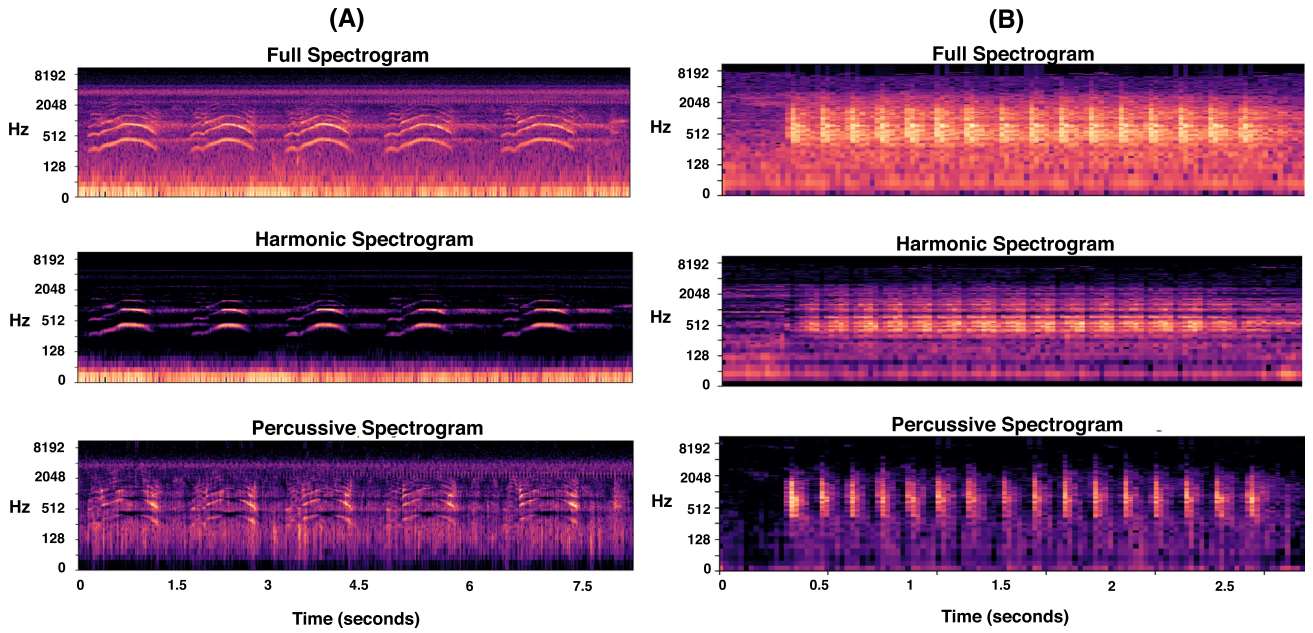

FIG. 2. (Color online) Difference in harmonic and percussive components of sounds produced by (A) Indian peafowl and (B) white-bellied woodpecker.

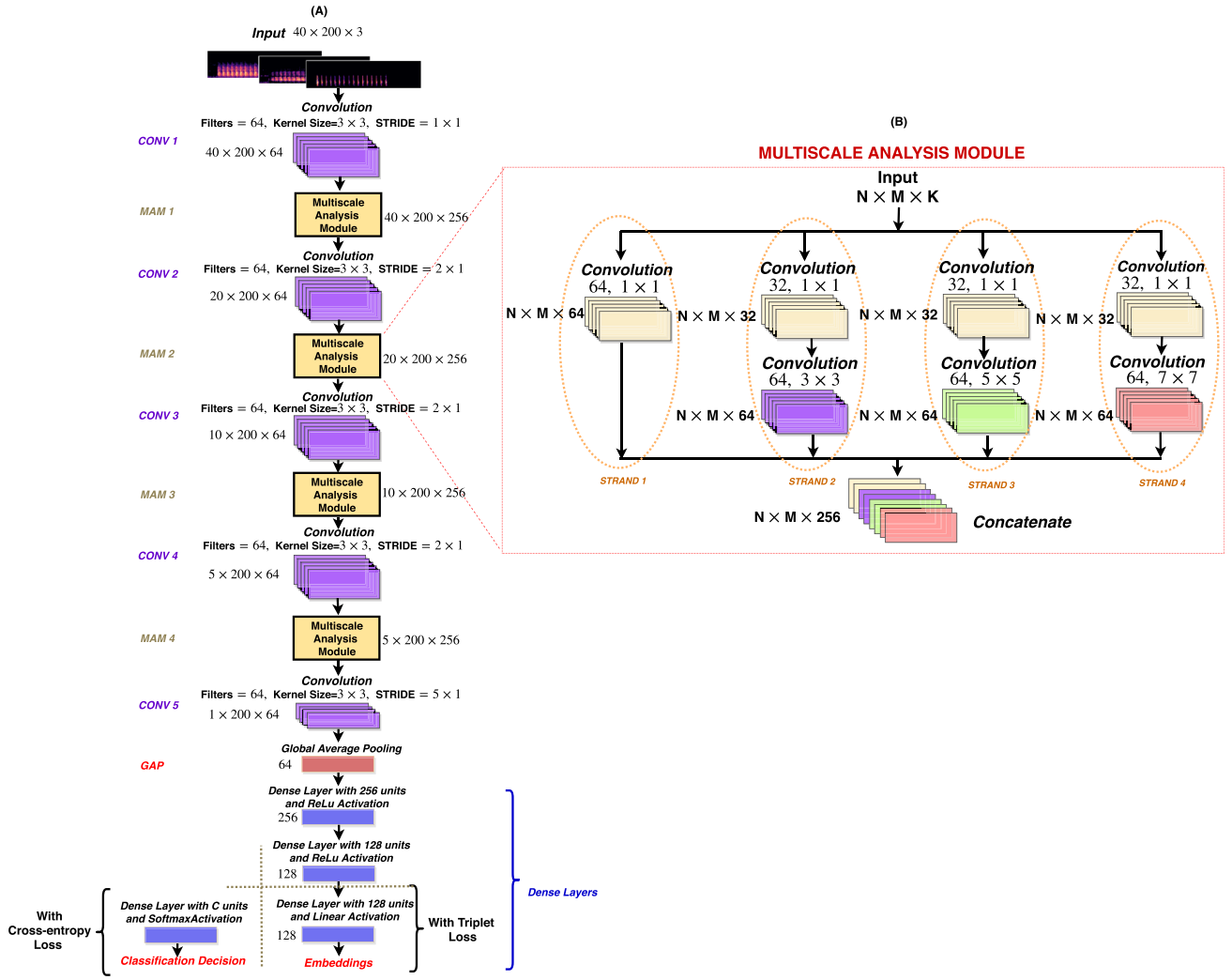

FIG. 3. (Color online) Illustration of (A) the proposed multiscale CNN architecture and (B) a multiscale scale analysis module. Triplet loss is used to perform deep metric learning. Whereas, to use the proposed multiscale CNN architecture for classification (with cross-entropy loss), the last layer of the architecture is replaced by a fully connected layer having C (number of classes) units and softmax activation.

compared to the longer vocalizations. Hence, a smaller kernel size is more appropriate for the shorter bioacoustic events and vice versa. The smaller 3×3 filter helps in learning minute details from an input Mel-spectrogram whereas the larger filters (5×5 and 7×7) help in capturing more global traits due to the larger receptive fields. In bioacoustics, these minute details can be low energy harmonics or vocalizations recorded in a far-field setting. The global traits can include the information about the frequency contents or bandwidth and the coarser time-frequency modulations of bioacoustic events.

Each MAM receives an input of 64 feature maps ($N \times M \times 64$) that are processed by seven convolution layers arranged in four parallel strands as shown in Fig. 3(b). The first strand contains one convolution layer that has 64 filters of 1×1 kernel size. These filters are mainly concerned with selecting the inter feature map patterns rather than the spatial analysis of feature maps. Most feature maps have some complementary information.[40] Hence, learning these inter feature map patterns can be helpful in obtaining discriminative features. The second, third, and fourth strands consist of two convolution layers. The first convolution layer in all these strand consists of 32 filters of 1×1, whereas, the second convolution layers have 64 filters of sizes 3×3, 5×5, and 7×7, respectively. Here the 1×1 convolution layers serve two purposes: (1) It decreases the number of input channels from 64 to 32 and reduces the computational requirements for the following convolutional layer in each strand. (2) As discussed earlier, 1×1 filters are used for selecting the discriminative feature patterns from the input feature maps. Since the appropriate feature patterns may be scale dependent, a separate 1×1 convolution layer in each strand provides independence in the feature selection at different scales. The output of these layers are processed by convolution layers having a filter size of 3×3, 5×5, and 7×7 in the second, third, and fourth strand, respectively. The difference in the responses of filters of different strands is illustrated in Fig. 4. The feature maps obtained from all the four strands are concatenated in a channel-wise manner to output 256 feature maps from each module. Zero-padding is used to make sure that each feature map is of same dimension before concatenation.

c. Bottleneck, global pooling, and dense layers. The network consists of five convolution layers having 64 filters of 3×3. Apart from the first convolution layer, all other convolution layers act as the bottleneck layers. They use strided convolutions to down-sample the feature maps by a factor of 2 (or 5 in case of the last convolution layer) along the Mel-energy axis. Apart from down-sampling, they also help in selecting the relevant scale-dependent features from the feature maps, obtained from MAMs, by analyzing the inter feature map correlations.[40] Due to this feature selection, the number of channels are decreased from 256 (generated by multiscale analysis module) to 64 which also helps in decreasing the computation requirements for the corresponding layers. After all the convolution layers and multiscale analysis modules, global average pooling (GAP) is applied to obtain a 64-D vector. This averaging operation helps in making the framework invariant towards the time differences in onsets–offsets of the bioacoustic events in audio recordings or their Mel-spectrograms. Then, this 64-D vector is passed to the dense layers. The network has three dense layers having 256, 128, and 128 hidden units.

d. Activation, regularization, and optimizer. Each convolution layer (whether stand-alone or in multiscale analysis module) and first two dense layers are followed by rectified linear unit (relu) activation. The output of the last dense layer is normalized to have the unit norm such that embeddings produced by the proposed CNN lie on the unit hypersphere. Dropouts are used before each dense layer to avoid over-fitting. Along with dropout, exponential weight decay is also used to avoid over-fitting and improve generalization.[41] Adagrad with a fixed learning rate is used as an optimizer.

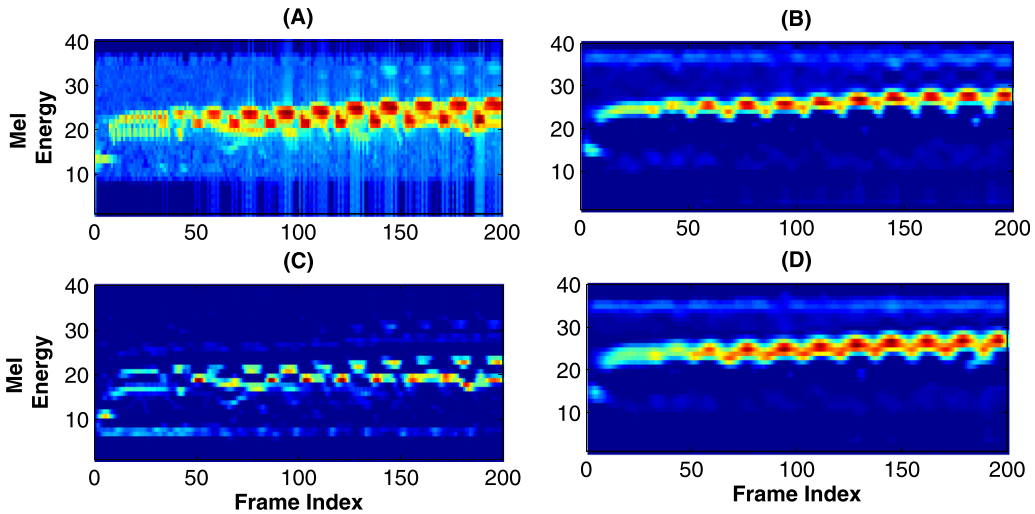

FIG. 4. (Color online) (A) Mel-spectrogram given as input to the proposed CNN. (B), (C), and (D) depicts the filter responses obtained for 32nd 5×5, 6th 3×3, and 62nd 7×7 filters of the first multiscale analysis module (*MAM 1*) of the trained multiscale CNN. These particular filters are chosen for their expressivity.

2. MLP architecture

The MLP used in the proposed framework consists of three layers: an input layer with 128 units, a hidden layer with 256 units and an output layer with C units (C is the number of classes). The relu and softmax activations are used after the hidden layer and the output layer, respectively. The categorical cross-entropy entropy is used as the loss function. The weight optimization is performed by Adam solver. Exponential weight decay is employed to avoid overfitting.

C. Multiscale CNN training: Dynamic triplet loss

Dynamic triplet loss is utilized for training the proposed multiscale CNN. As discussed in Sec. I, dynamic triplet loss processes a set of three examples, referred to as a triplet, at a time to learn the desired embedding space where all possible triplets satisfy the triplet constraint. Given a triplet of embeddings, $\mathcal{X}_i = \{\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{x}_i^n\}$, the triplet constraint can be defined as

$$\|\mathbf{x}_i^a - \mathbf{x}_i^n\|_2^2 - \|\mathbf{x}_i^a - \mathbf{x}_i^p\|_2^2 \geq \alpha. \tag{1}$$

Here $\mathcal{X}_i$ represents the ith triplet of embeddings sampled from the training data. $\mathbf{x}_i^a$, $\mathbf{x}_i^p$, and $\mathbf{x}_i^n$ are an anchor, a positive, and a negative example of the ith triplet. Also, α is the enforced margin or distance between the positive examples and the negative examples. Thus, based on the triplet constraint, the loss function to be minimized is[35]

$$\mathcal{L} = \sum_{i=1}^{N} \max(\|\mathbf{x}_i^a - \mathbf{x}_i^p\|_2^2 - \|\mathbf{x}_i^a - \mathbf{x}_i^n\|_2^2 + \alpha, 0), \tag{2}$$

where N is the possible number of triplets to be used for training.

For training, triplets are sampled from a mini-batch in an online fashion. A forward pass is performed on the CNN to obtain embeddings for a mini-batch of input examples. Using the corresponding class labels, these embeddings are analyzed to form triplets that are later used for optimizing the current state of CNN. The performance of triplet loss is directly dependent on the choice of triplets to be used for training. Choosing triplets that satisfy triplet constraint [Eq. (1)] will lead to no change in the state of the CNN. Hence, only triplets that violate the triplet constraint are considered for training. These triplets are of two types: hard triplets and semi-hard triplets. A triplet $\mathcal{X}_i$ is classified as hard or semi-hard according to the following criteria:[35]

$$\mathcal{X}_i \text{ is } \begin{cases} hard & d(\mathbf{x}_i^a, \mathbf{x}_i^n) < d(\mathbf{x}_i^a, \mathbf{x}_i^p) \\ semi-hard & d(\mathbf{x}_i^a, \mathbf{x}_i^p) < d(\mathbf{x}_i^a, \mathbf{x}_i^n) \text{ and} \\ & d(\mathbf{x}_i^a, \mathbf{x}_i^n) < d(\mathbf{x}_i^a, \mathbf{x}_i^p) + \alpha. \end{cases}$$

Here $d()$ refers to the Euclidean distance.

Although hard triplets appear to be more informative for training, they result in higher loss values, leading to the larger weight updates. These larger weight updates result in significant change to the current state of network, hence, undoing the optimization work done by the previous weight updates. Thus, utilizing these hard triplets may lead to instability during training. It has been shown in Ref. 35 that semi-hard triplets often leads to faster convergence and effective training than the hard triplets. Building on this information, the semi-hard triplets are used for training the proposed CNN. In semi-hard triplet, the distance between anchor-negative pairs is greater than the anchor-positive pairs as desired. However, this distance is not greater than the desired separation margin α. Thus, the weight updates obtained in case of semi-hard triplets are not as large as the hard triplets, leading to a stable training. More details about the online triplet sampling and the utilization of semi-hard triplet for training can be found in Ref. 35.

ALGORITHM 1: Training CNN using dynamic triplet loss.

input: $f()$: CNN (randomly initialized) X: Training dataset L: Labels
α_i: Initial value of margin
α_f: Final value of margin
thresh: Threshold for margin updates
K: Number of epochs
output: $f()$: Trained CNN for metric learning

$\alpha = \alpha_i$ // Initialize margin
count_list = [] // List to store number of triplets sampled in each iteration
for $J \leftarrow 1$ **to** K **do**
  $\mathcal{I}, \mathcal{L} =$ createBatches(X, L, n) // Returns n batches stored in List $\mathcal{I}$ and corresponding batch labels in $\mathcal{L}$
  for $i \leftarrow 1$ **to** n **do**
    $\mathcal{E} = f(\mathcal{I}[i])$ // Forward pass to get embeddings for ith batch
    $\mathcal{T}, t =$ getTriplets$(\mathcal{E}, \mathcal{L}[i], \alpha)$ // Returns $\mathcal{T}$, a set containing t semi-hard triplets, sampled from ith batch
    count_list.append(t) // Store the number of semi-hard triplets sampled from ith batch
    $num = len($*count_list*$)$ // Number of elements in *count_list* currently
    if $num \geq 3$ *AND* $\alpha \leq \alpha_f$ **then**
      if ($count[num] <$ *thresh AND* $count[num-1] <$ *thresh AND* $count[num-2] <$ *thresh*) **then**
        $\alpha = \alpha + 0.05$ // Update margin when number of triplets are less than threshold for three consecutive iterations
      end
    end
    $L =$ calculateTripletLoss$(f(), \mathcal{T}, \alpha)$ // Calculate Triplet Loss using Eq. 2
    $f() =$ UpdateWeights$(f(), L)$ // Back-propagate L through $f()$ to get the updated $f()$
  end
end

As discussed in Sec. I, a dynamic variant of the triplet loss is used in this work. In the proposed implementation of triplet loss, α or the margin is considered as a dynamic variable whose value is changed over the course of training. The overall procedure to calculate dynamic triplet loss is depicted in Algorithm 1. We start with a small margin, $\alpha = 0.2$, and force the network to learn the embedding space where examples of each class are separated from other by a

distance of 0.2. As the network is trained, the number of semi-hard triplets mined from training dataset decreases. If this number of mined triplets is less than a pre-defined threshold for three consecutive iterations, the value of α is incremented by 0.05. Again, the network is trained to satisfy the new triplet loss induced by new value of α. This process is continued until α reaches a pre-defined maximum value of 0.6. As discussed in Sec. I, this dynamic triplet loss provides more triplets for training the model. The minimum and maximum values of α are determined empirically.

D. Classification

As illustrated in Fig. 1, first the multiscale CNN is trained to learn the transformation or embedding space using dynamic triplet loss. This trained CNN is used to extract 128-D embeddings from all the training examples. These embeddings show high class-specific signatures as evident in Fig. 5. This figure exhibits two-dimensional t-sne (Ref. 42) representations of embeddings generated from vocalizations of 12 different bird species. Note that t-sne is a data visualization method that nonlinearly maps the high-dimensional data to a desired low-dimensional space. Once embeddings are obtained, a MLP with Adam optimizer (described in Sec. II B 2) is trained for classification. During inference, the trained CNN and MLP are used to obtain embeddings from the test examples and to classify those embeddings, respectively.

E. Open-set classification using the proposed DML framework

Open-set classification is a challenging issue in designing bioacoustic classification frameworks for field conditions. The existing bioacoustic classification methods assign a test example to a class whose training examples exhibit maximum similarity to this test example, even if the test example does not belong to any of the classes involved in training. Thus, there must be a way to reject such test examples without affecting the classification performance of the involved classes. To tackle open-set classification, the metric learning module of the proposed framework can be used. The distance between test embedding and training embeddings of a class can be exploited to perform open-set classification.

To model the distance from training embeddings, an unimodel Gaussian distribution is utilized. The embeddings of the training and validation examples of each class are obtained from the trained multiscale CNN. All the training embeddings are averaged to obtain a mean vector. For each class, a Gaussian distribution is fitted over the distance between validation embeddings and the mean vector. The maximum likelihood estimation is used to estimate parameters of these Gaussian distributions.

During testing, an input test example is classified by the proposed framework. To perform outlier rejection, the distance between test embedding and the mean vector of the output class is calculated. The likelihood of this distance is computed with respect to the Gaussian distribution of the output class. If this likelihood is less than a particular threshold, the test example is rejected.

III. EXPERIMENTAL SETUP

In this section, datasets, comparative methods, and parameter setting used for the performance evaluation are described.

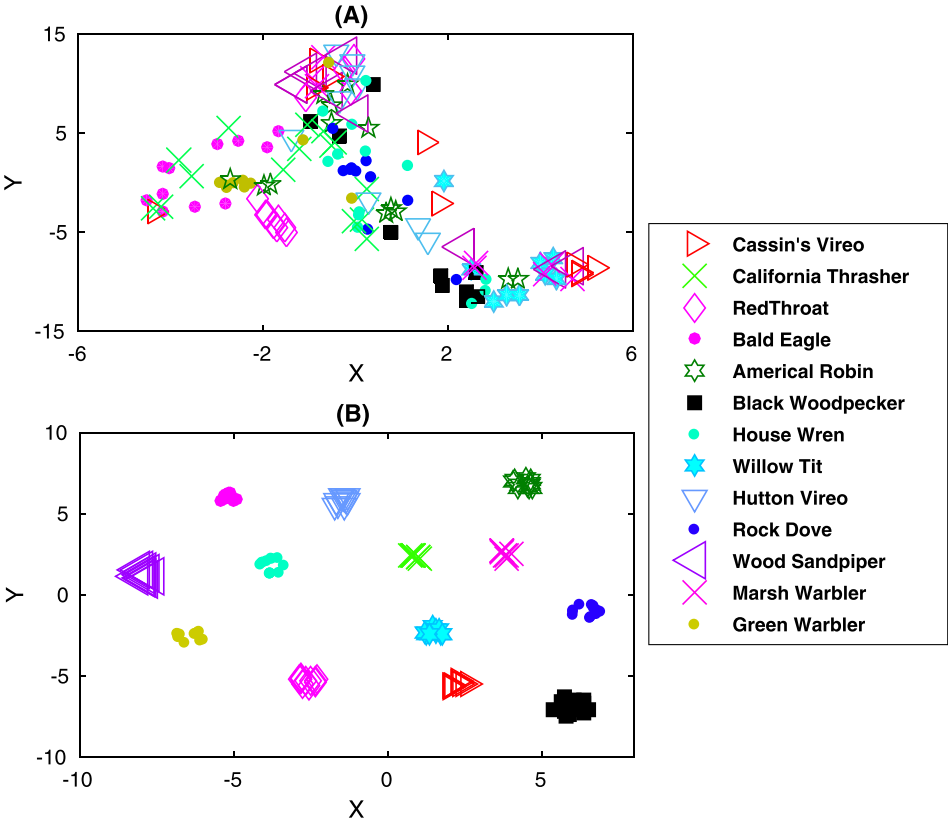

FIG. 5. (Color online) Two dimensional t-SNE visualization of 128-days embeddings extracted from audio examples of 12 different bird species using (A) untrained multiscale CNN and (B) fully trained multiscale CNN.

A. Datasets used

The performance of the proposed DML framework is evaluated on three different datasets.

- **Birdcalls71**: This dataset contains audio recordings of 71 different bird species that are obtained from three different sources. The recordings of 38 bird species were provided by the Macaulay Library[43] on an academic license. The recordings of seven bird species were downloaded from bird database maintained by Art & Science Centre, UCLA.[44] The recordings of 26 bird species were obtained from the Great Himalayan national park (GHNP) dataset[31] and were provided on request. Out of 71 species involved in this dataset, 43 species are found in North America while the remaining 26 bird species are found in Northern India. A list of bird species present in this dataset is available at http://tiny.cc/qi8q7y. All the audio recordings are sampled at 44.1 kHz and vary in duration from 0.5 to 320 s. Due to licensing issues, the authors cannot make this dataset public. However, the processed Mel-spectrograms extracted from these recordings are hosted on a public platform for analysis.[45]
- **Anuran dataset**: The anuran dataset contains audio recordings of ten different species of order *anura* that contains frogs and toads. The dataset is composed 60 bioacoustic recordings containing anurans' crocks and ribbits. All the recordings were obtained from the Amazon rainforest, and contain various background disturbances. These recordings are sampled at 44.1 kHz and vary in duration (from 3 to 360 s). The list of Anuran species involved in this dataset can in found in Ref. 6 as well as at http://tiny.cc/qi8q7y.
- **Flight calls dataset** (CLO-43SD): This public dataset is provided by Salamon *et al.*[46] and is composed of audio clips containing flight calls of 43 different North American wood-warblers. These audio clips are recorded in different conditions using different recording devices. Some audio clips are clean and were recorded using highly directional microphones, whereas some clips are noisy and were recorded using omnidirectional microphones.[46] Each audio clip is processed and clipped to contain a single flight call only. Along with processed clips, Mel-spectrograms extracted from these audio clips are also provided in the dataset. The audio clips are sampled at 22.05 kHz, and Mel-spectrograms are obtained using 11.6 ms frame size with an overlap of 1.25 ms and 40 Mel bands. It must be noted that 11.6 ms frame size is optimum to analyze flight calls.[21,46] The wood-warbler species involved in this dataset are listed at http://tiny.cc/qi8q7y.
- **Combined**: To analyze the scalability of the proposed framework, all three datasets are combined together to create a larger dataset having 124 classes.

These particular datasets are chosen due to their availability to the authors.

B. Data pre-processing and train-test distribution

1. Pre-processing

Audio recordings in all the datasets are of variable duration. For a uniform input to CNN, these recordings are processed to have the same duration. All the audio recordings are divided into fixed length segments of 2 s. These segments are used for training and performance comparison. If the duration of any recording is less than 2 s, then the signal is repeated (from the beginning) to force the fixed duration of 2 s. Short-term spectral analysis is performed, using a frame size of 20 ms with 50% overlap, to obtain the respective feature representations (from all databases except CLO-43SD). Thus, each input example consists of 200 frames. In case of CLO-43SD dataset, frames of a pre-computed Mel-spectrogram are repeated to obtain a fixed number of frames, i.e., 200 per example (for maintaining uniformity between datasets). Note that there is a difference in frame sizes used for the short-term analysis of CLO-43SD and other datasets. However, this difference is ignored to create a combined dataset for the sake of analyzing scalability of the proposed DML framework.

2. Train-test data distribution

All datasets are divided into train, test, and validation sets. Fifteen percent of examples from each class are used for validation. A random tenfold cross-validation is applied on remaining examples to create ten different train-test sets. In each fold, 60% of the remaining examples per class are randomly sampled for training and the remaining examples are used for testing. Due to random selection, in each fold, a different subset of available examples are used for training and testing. The total number of examples used for training, testing, and validation (in each fold) are tabulated in Table I. In Birdcalls71 dataset, the number of per-class training examples varies from 5 to 98. Out of 71 classes, 54 classes exhibit less than 20 training examples. Hence, a majority of classes have only a handful of training examples. Similarly, in CLO-43SD, the number of per-class training examples lie in range of 5–641. Thus, along with the training data scarcity for a majority of classes, both these datasets also exhibit a large class imbalance. The details about the number of per-class examples used for training, testing, and validation are available at http://tiny.cc/qi8q7y.

C. Comparative studies and performance metric

In this study, ten comparative methods are chosen for the performance evaluation. All these methods are tabulated in Table II, and can be divided into three categories: shallow learning methods, CNN models with cross-entropy loss, and existing CNNs with dynamic triplet loss.

TABLE I. Number of training, testing and validation examples used for the performance comparison.

Dataset	Number of training examples (in each fold)	Number of test examples (in each fold)	Number of validation examples	Total examples
Birdcalls71	1218	822	390	2430
Anuran	1199	801	357	2350
Flight calls (CLO-43SD)	2773	1858	831	5428
Combined	5190	3481	1578	10 208

TABLE II. Comparative methods used for the performance evaluation.

Method	Input feature representation	Nature
Spherical K-means and random forest (SKM) (Ref. 26)	Mel-spectrograms	Unsupervised feature learning
Deep convex representations and random forest (DCR) (Ref. 28)	Compressed spectral frames	Supervised dictionary learning
Kernel-based extreme learning machines (KELM) (Ref. 30)	ComParE (Ref. 47) feature set, consists of 65 low-level signal descriptors (Time aggregation using statistical and modulation functionals to obtain 6373-dimensional vector)	Shallow learning
VGG (Ref. 9)	Mel-spectrogram	CNN
Fine-tuned VGG (VGG-FT) (Ref. 9)	Mel-spectrogram	CNN/transfer learning
CNN proposed by Salamon *et al.* (SAL) (Ref. 21)	Mel-spectrogram	CNN
Multiscale CNN with cross-entropy loss (MS-CNN)	Mel-spectrogram	CNN
VGG with dynamic triplet loss (VGG-TL)	Mel-spectrogram	CNN-based deep metric learning
Pre-trained VGG with dynamic triplet loss (VGG-FT-TL)	Mel-spectrogram	Transfer learning/CNN-based deep metric learning
SAL with dynamic triplet loss	Mel-spectrogram	CNN-based deep metric learning
Multiscale CNN with dynamic triplet loss (MS-CNN-TL)	Mel-spectrogram	CNN-based deep metric learning

- ***Shallow learning methods:*** These shallow learning baselines utilize polynomial kernel-based extreme learning machines (KELM)[30] and random forest as classifiers. In first baseline, the KELM is trained on low-level audio descriptors [ComParE (Ref. 47) feature set]. Whereas, in second and third baselines, random forest classifier is trained on unsupervised and supervised feature representations, respectively. The unsupervised feature representations are obtained using spherical K-means[26] (SKM) while the supervised representations[28] are acquired by deep convex matrix factorization (DCR). The input feature representations (Mel-spectrogram and compressed spectral frames) used in the respective studies are also used here. In SKM, frame-wise feature representations for each input example are aggregated using mean and standard deviation to obtain a fixed dimensional representation. In DCR, a random forest classifier is trained on a frame-wise deep convex representation. During testing, a voting rule is used on frame-wise decisions to classify the input example.
- ***CNN models with cross-entropy loss:*** CNN proposed by Salamon *et al.*[21] (SAL) and VGG (used in Ref. 9 for audio classification) are used as deep learning baselines. The total number of trainable parameters in SAL and VGG are 1 631 179 and 8 480 891, respectively (if last dense layer has 124 units). To evaluate the proposed DML framework against transfer learning, the pre-trained VGG network is fine-tuned for bioacoustic classification. This VGG network is pre-trained on AudioSet database having approx. 2×10^6 audio examples (see Ref. 9 for more details). The dense layers of VGG are replaced by three dense layers having 256, 128, and C (number of classes) hidden units. A dropout of 0.5 is used before each dense layer. The first two dense layers have relu activation where as the last dense layer is followed by softmax activation. The final CNN baseline is the proposed multiscale CNN with the cross-entropy loss (MS-CNN). The last dense layer of the proposed CNN is replaced with a dense layer having C units and softmax activation (C is the number of target classes).
- ***Existing CNN models with dynamic triplet loss:*** For a thorough comparison between cross-entropy and dynamic triplet loss, all CNN baselines [i.e., VGG, VGG-FT (fine-tuned) and SAL] are also used in the proposed DML framework (Fig. 1) for performing classification. These baselines are trained using dynamic triplet loss to output the desired 128-dimensional embeddings, which are later utilized by a MLP to classify the input examples.

In all CNN-based methods, three-channeled Mel-spectrograms (see Sec. II A) are given as the input representation. The Keras implementations of these CNN baselines are publicly available along with datasets.[45]

1. Performance metric

The imbalance between classes is large, however, the equal weightage must be given to the classification performance obtained for each class. Hence, macro F1-score[48] is used as a metric for the performance comparison. The macro F1-score is the average of class-specific F1-scores where F1-score is the harmonic mean of precision and recall.

D. Parameter setting

The parameter setting used in CNN baselines and the proposed multiscale CNN are tabulated in Table III. These hyperparameters are determined empirically over the validation examples, and appear to be optimal for datasets considered in this study. An experimental study to determine the prominent hyperparameters of the proposed DML framework such as learning rate, margin (α), and dropouts is provided as the supplementary material.[49] CNN baselines with cross-entropy loss are trained for 200 epochs with a batch-size of 32. Checkpoints are used after each epoch to

TABLE III. Parameter setting used in different CNN-based comparative methods. For cross-entropy loss-based models, the total trainable parameters are calculated for 124 units (number of classes in *Combined* dataset) in the last dense layer.

Model	Layers		Trainable parameters (approx.)	Learning rate	Dropout		Optimizer	Weight decay
	Conv	*Dense*			*Conv layers*	*Dense layers*		
VGG	6	3	8.48 M	0.001	—	0.5	Adam	—
VGG with dynamic triplet loss (VGG-TL)	6	3	8.48 M	0.0001	—	0.5	Adagrad	—
Pre-trained VGG (VGG-FT)	6	3	8.48 M	0.001	—	0.5	Adam	—
VGG-FT with dynamic triplet loss (VGGish-FT-TL)	6	3	8.48 M	0.0001	—	0.5	Adagrad	—
SAL	3	2	1.63 M	0.1	—	0.5	Stochastic GD (SGD)	0.001 on dense layers only
SAL with dynamic triple-loss (SAL-TL)	3	2	1.63 M	0.001	—	0.5	Adagrad	
MS-CNN (multiscale CNN)	33 (5 standalone and 28 in MAMs)	3	1.37 M	0.001	—	0.5	Adam	0.0001 on all layers
Proposed DML framework: MS-CNN with dynamic triplet loss (MS-CNN-TL)			1.37 M	0.001			Adagrad	

determine the setting that provides the least validation loss. For training MLP (in the proposed framework), a learning rate of 0.001 and weight decay of 0.0001 are used.

For implementing dynamic triplet loss, each mini-batch is forced to have at least five examples per each class. Hence, all classes are represented in a mini-batch. The semi-hard triplets are sampled from this mini-batch and are used for training the CNN. The number of triplets that can be processed simultaneously (let us say triplet batch size) is often limited by the available GPU memory. In our implementation, we set this triplet batch size to be 150 input examples or 50 semi-hard triplets. Thus, the semi-hard triplets sampled from a mini-batch are presented to the CNN in an iterative manner where during each iteration, the triplet batch (having 50 or less triplets) is used to calculate triplet loss and update the weights. The number of mini-matches in an epoch are limited to 1000. The margin (α) is varied from 0.2 to 0.6 in all dynamic triplet loss-based CNN models. A threshold of 15 triplets is used for the margin update in Algorithm 1. All the triplet loss-based models are trained till they achieve full convergence. In the current context, convergence simply means that all semi-hard triplets in the training dataset satisfy the triplet constraint across all dynamically chosen margins.

For implementing SKM (for all datasets), spherical K-means with 128 clusters and random forest with 100 trees is used. For implementing DCR, a three-level archetypal analysis-based matrix factorization (with an order of 128, 64, and 32) is used to learn the class-specific dictionaries. A random forest with 100 trees is used for classifying convex representations obtained from class-specific dictionaries. In KELM, a polynomial kernel of 10 deg and a hidden layer of 2048 units is used in the extreme learning machine. In all the comparative methods, a frame-size of 20 ms with 50% overlap and 2048 FFT points are used to obtain the time-frequency representations. All the aforementioned parameters are empirically determined on the validation examples.

IV. RESULTS AND DISCUSSION

In this section, first, the classification performances of the proposed DML framework and different baselines are presented. Then, the outlier rejection performance of the open-set classification module is analyzed. Later, the generalization ability of the proposed framework is discussed. Then, two ablation studies are presented to analyze the effects of the proposed dynamic triplet loss and the multiscale analysis on classification performances. For generalization and ablation studies, only the experimental results on Birdcalls71 dataset are presented here.

A. Classification performance

Figure 6 illustrates the box plots depicting the classification performances of different methods on all four datasets. The two-tailed, paired t-test is performed between scores obtained by different comparative methods to analyze the statistical significance of their classification performances. A significance level of $p < 0.000\,01$ is used for the significance testing. The following inferences can be drawn from the analysis of Fig. 6 and the corresponding significance analysis:

- Shallow learning techniques (SKM, KELM, and DCR) are significantly outperformed by CNN-based frameworks including the proposed MS-CNN and MS-CNN-TL across all datasets.
- MS-CNN performs better than VGG and SAL on all but the Anuran dataset, highlighting the superiority of the proposed multiscale CNN. On Anuran dataset, no significant difference is observed over performances of these baselines.
- VGG-FT outperforms VGG and SAL while showing similar performance to that of MS-CNN. This shows that the utilization of transfer learning or fine-tuning a pre-trained model improves the classification performance.

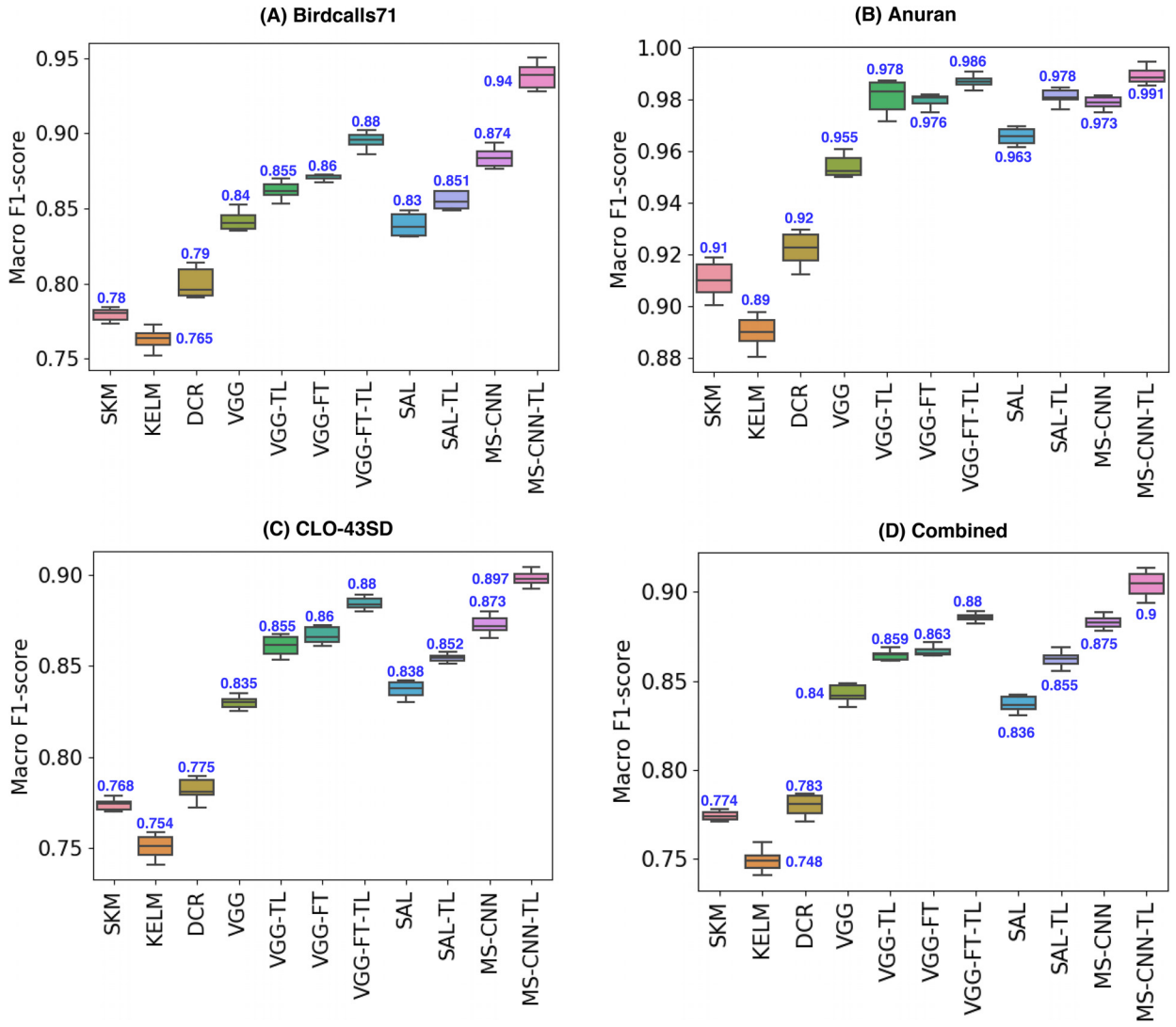

FIG. 6. (Color online) Box plots depicting the classification performances of the proposed framework along with various baselines on (A) birdcalls71, (B) anuran, (C) CLO-43SD, and (D) combined datasets. The number next to each box plot represents the average F1-score obtained across all ten folds.

- All dynamic triplet loss-based methods (VGG-TL, VGG-FT-TL, SAL-TL, and MS-CNN-TL) exhibit significant improvements (at a significance level of $p < 0.00001$) over their cross-entropy counterparts (VGG, VGG-FT, SAL, and MS-CNN). This demonstrates that in the current setup, the utilization of dynamic triplet loss-based framework results in better classification performances over the cross-entropy loss-based deep learning frameworks.
- MS-CNN-TL (proposed) outperforms all the considered baselines across Birdcalls71, Anuran, CLO-43SD and Combined datasets. However, on Anuran dataset, VGG-FT-TL and MS-CNN-TL exhibit comparable performances.

A table containing p-values, obtained during significance analysis, between different comparative methods is included in the supplementary document for further analysis.[49]

B. Performance of open-set classification module

An experiment is designed to analyze the outlier rejection accuracy of the open-set classification module. First, MS-CNN-TL is trained on Birdcalls71 dataset and Anuran dataset is used for outlier rejection. Then, the framework is trained on Anuran dataset and Birdcalls71 is used for outlier rejection. For training, train-test setup described in Sec. III B is also used here. Instead of ten folds, only a single set of training and testing examples are used in this experiment. For outlier rejection, all the available examples in the dataset are used. The results of this experiment are documented in Table IV. The analysis of this table makes it clear that in both setups, MS-CNN-TL with the aforementioned rejection mechanism is able to reject the outliers with good accuracy of 97% and 95%. However, in comparison to the average F1-scores obtained in Fig. 6, a small relative drop in macro F1-scores is observed. This shows that, as expected, incorporating outlier rejection mechanism in MS-CNN-TL leads to a small drop in classification performance. However, this observed classification performance is still competitive in comparison to the other methods considered in this study (see Fig. 6).

C. Generalization of the proposed DML framework

In bioacoustic classification tasks, the training examples often do not contain the whole repertoire of vocalizations

TABLE IV. Classification and outlier rejection performances of the proposed MS-CNN-TL framework in different training-testing setup. In each setup, a dataset is used for training and classification evaluation whereas a different dataset is used for evaluating the outlier rejection mechanism.

	Classification setup		Outlier rejection setup	
Training dataset	Testing dataset	Classification performance (Macro F1-score)	Outlier dataset	Rejection accuracy (%)
Birdcalls71	Birdcalls71	0.91	Anuran	97
Anuran	Anuran	0.95	Birdcalls71	93

that a species can produce. In field conditions, the test examples often contain vocalizations that are not used for training. Thus, an effective classification framework must be able to generalize on these unseen examples. To study the generalization ability of the proposed framework, the unseen vocalizations (not included in training train the model) of Cassin's vireo (one of the species in Birdcalls71 dataset) are used. The ten unseen song phrases are extracted from the audio recordings available at https://goo.gl/x17fYf and are provided for analysis along with Birdcalls71 dataset. The t-SNE (Ref. 42) representation of the embeddings extracted from these unseen song phrases are shown in Fig. 7. The analysis of this figure makes it clear that embeddings generated from the unseen song phrases of Cassin's vireo exhibit more similarity to the training Cassin's vireo examples than embeddings of other species. This is attributed to the fact that the dynamic triplet loss used in the proposed framework deals with grouping the similar vocalizations together and separating them from the dissimilar examples. Generally, vocalizations of a species are more similar to each other than the sounds produced by other species (though exceptions are always present in natural systems). As a result, the embeddings extracted from seen or unseen vocalizations of a species are bound to be grouped together. Thus, the utilization of DML helps in overcoming small variations in the nature of vocalizations as well as differences in the recording environment during training and testing.

D. Dynamic vs fixed margin triplet loss

In this section, the effect of dynamic and classical triplet loss on the classification performance is explored. MS-CNN-TL is trained with different fixed margins (i.e., from 0.2 to 0.6) on Birdcalls71 dataset. The F1-scores obtained for different values of fixed margin are compared against F1-score obtained using dynamic triplet loss where margin is varied from 0.2 to 0.6. Figure 8 depicts the average F1-scores obtained during this experiment. From the analysis of this figure, it is clear that utilizing dynamic triplet loss results in better classification than the static or fixed margin triplet loss. This can be attributed to the fact that the number of triplets involved for training in dynamic margin triplet loss is significantly more than the fixed margin triplet loss, leading to the better training of the multiscale CNN.

E. Effect of multiscale analysis on classification performance

To analyze the impact of multiscale analysis on performance of the proposed framework, MAMs are replaced by 256 standalone convolution layers having 3×3, 5×5, and 7×7 filters in the multiscale CNN. The 256 filters are chosen to match the number of feature-maps generated by MAMs. Three CNN configurations are resulted from this alteration are

- **Configuration 1:** INPUT $\rightarrow$ Conv(64, 3×3) $\rightarrow$ Conv(256, 3×3) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 3×3) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 3×3) $\rightarrow$Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 3×3) $\rightarrow$ Conv(64, 3×3, 5×1) $\rightarrow$ GAP $\rightarrow$ Dense(128) $\rightarrow$ Dense(256) $\rightarrow$ Dense(128).
- **Configuration 2:** INPUT $\rightarrow$ Conv(64, 3×3) $\rightarrow$ Conv(256, 5×5) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 5×5) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$Conv(256, 5×5) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 5×5) $\rightarrow$ Conv(64, 3×3, 5×1) $\rightarrow$ GAP $\rightarrow$ Dense(128) $\rightarrow$ Dense(256) $\rightarrow$ Dense(128).

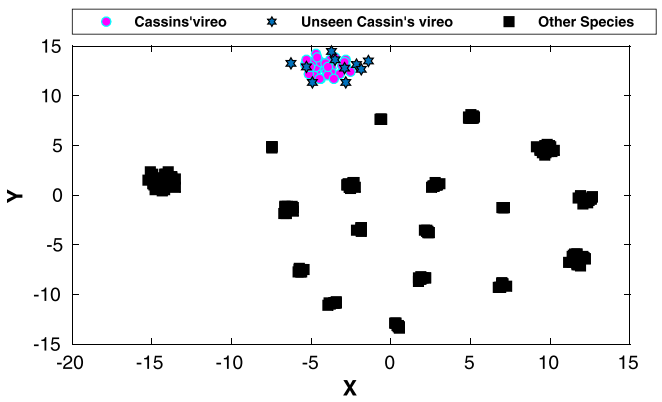

FIG. 7. (Color online) t-SNE (Ref. 42) visualization of embeddings generated from seen and unseen Cassin's vireo song phrases using MS-CNN-TL. For illustration purposes, only 20 species are shown in this figure. The remaining species also exhibit similar behaviour.

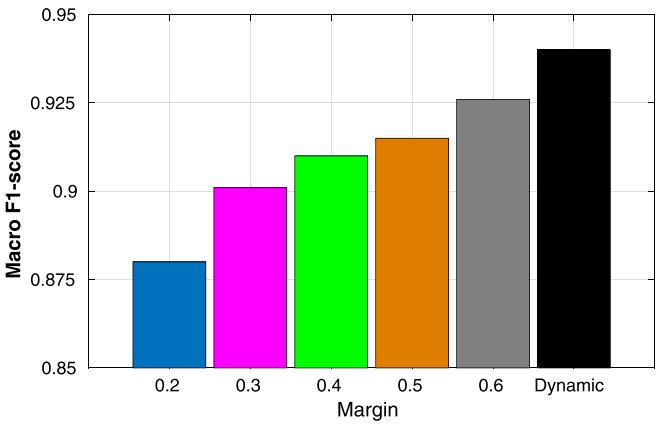

FIG. 8. (Color online) Classification performances of MS-CNN-TL on Birdcalls71 as a function of margin. The macro F1-scores presented here are average of scores obtained across ten folds.

TABLE V. Classification performances of different CNN configurations, obtained by removing multiscale analysis from the proposed multiscale CNN, on *Birdcalls71* dataset.

Model	Average F1-score
Configuration 1	0.87
Configuration 2	0.88
Configuration 3	0.85
Proposed multiscale CNN	0.94

- **Configuration 3:** INPUT $\rightarrow$ Conv(64, 3×3) $\rightarrow$ Conv(256, 7×7) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 7×7) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 7×7) $\rightarrow$ Conv(64, 3×3, 2×1) $\rightarrow$ Conv(256, 7×7) $\rightarrow$ Conv(64, 3×3, 5×1) $\rightarrow$ GAP $\rightarrow$ Dense(128) $\rightarrow$ Dense(256) $\rightarrow$ Dense(128).

Here GAP stands for global average pooling, and Conv(N, $n1 \times n2$, $m1 \times m2$) represents convolution layer having N filters of $n1 \times n2$ kernel size with a stride of $m1 \times m2$.

All these configuration are used in the proposed DML framework and their performances are evaluated on Birdcalls71 dataset. The train-test setting described in Sec. III B is also used here. Table V shows the average F1-scores obtained across ten folds by the aforementioned configurations. From the analysis of this table, it is evident that utilizing multiscale analysis helps in improving the classification performances by a noticeable margin.

V. CONCLUSION

In this paper, the authors present a deep metric learning-based framework for bioacoustic classification. The authors proposed a multiscale CNN and dynamic triplet loss to achieve effective deep metric learning in data-scarce conditions. The proposed multiscale CNN utilizes different kernel sizes to extract features at different granularities. The nature of dynamic triplet loss significantly increases the amount of triplets during training. The embeddings extracted from multiscale CNN-based DML are used as a feature representation for classifying an input example. The experimental results on four different datasets show that the proposed DML-based classification framework performs better than existing bioacoustic classification frameworks and various CNN architectures trained using cross-entropy loss. The authors also presented a simple augmentation that enables the proposed framework to perform open-set classification.

A major drawback of the proposed framework (and most of the existing metric learning frameworks) is that it cannot handle multilabel classification effectively. Future work may involve developing metric learning frameworks to overcome this drawback.

[1]F. van Bommel, "Birds in Europe: Population estimates, trends and conservation status," Br. Birds **98**, 269–271 (2005).
[2]S. A. Cushman, "Effects of habitat loss and fragmentation on amphibians: A review and prospectus," Biol. Conserv. **128**(2), 231–240 (2006).
[3]A. L. Borker, M. W. McKown, J. T. Ackerman, C. A. Eagles-Smith, B. R. Tershy, and D. A. Croll, "Vocal activity as a low cost and scalable index of seabird colony size," Conserv. Biol. **28**(4), 1100–1108 (2014).
[4]B. J. Furnas and R. L. Callas, "Using automated recorders and occupancy models to monitor common forest birds across a large geographic region," J. Wildl. Manage. **79**(2), 325–337 (2015).
[5]T. S. Brandes, "Automated sound recording and analysis techniques for bird surveys and conservation," Bird Conserv. Int. **18**(S1), S163–S173 (2008).
[6]B. Gatto, J. Colonna, E. M. dos Santos, and E. F. Nakamura, "Mutual singular spectrum analysis for bioacoustics classification," in *Proceedings of Mach. Learn. Sig. Process. (MLSP)* (September 2017).
[7]E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," IEEE Trans. Audio, Speech, Lang. Process. **25**(6), 1291–1303 (2017).
[8]Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proceedings of Int. Conf. Acoust. Speech, Signal Process. (ICASSP)* (2018), pp. 121–125.
[9]S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proceedings of Int. Conf. Acoust. Speech, Signal Process. (ICASSP)* (2017), pp. 131–135.
[10]J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," IEEE Signal Process. Lett. **24**(3), 279–283 (2017).
[11]R. Lu, Z. Duan, and C. Zhang, "Metric learning based data augmentation for environmental sound classification," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2017), pp. 1–5.
[12]P. Hamel, M. E. P. Davies, K. Yoshii, and M. Goto, "Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity," in *Proceedings of Int. Conf. Music Info. Retrieval* (2013).
[13]S. Ntalampiras, "Bird species identification via transfer learning from music genres," Ecol. Inf. **44**, 76–81 (2018).
[14]N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," Trans. Med. Imag. **35**(5), 1299–1312 (2016).
[15]S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *Proceedings of Int. Conf. Data Mining* (2016), pp. 439–448.
[16]D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," Meth. Ecol. Evol. **10**(3), 368–380 (2018).
[17]E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Proceedings of European Sig. Process. Conf. (EUSIPCO)* (2017), pp. 1744–1748.
[18]T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *Proceedings of European Sig. Process. Conf. (EUSIPCO)* (2017), pp. 1764–1768.
[19]T. Pellegrini, "Densely connected CNNs for bird audio detection," in *Proceedings of Eusipco* (2017), pp. 1734–1738.
[20]V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, "Birdvox-full-night: A dataset and benchmark for avian flight call detection," in *Proceedings of Int. Conf. Acoust. Speech, Signal Process. (ICASSP)* (2018), pp. 266–270.
[21]J. Salamon, J. P. Bello, A. Farnsworth, and S. Kelling, "Fusing shallow and deep learning for bioacoustic bird species classification," in *Proceedings of Int. Conf. Acoust. Speech, Signal Process. (ICASSP)* (2017), pp. 141–145.
[22]A. K. Ibrahim, H. Zhuang, L. M. Chérubin, M. T. Schärer-Umpierre, and N. Erdol, "Automatic classification of grouper species by their sounds using deep neural networks," J. Acoust. Soc. Am. **144**(3), EL196–EL202 (2018).
[23]B. P. Tóth and B. Czeba, "Convolutional neural networks for large-scale bird song classification in noisy environment," in *CLEF (Working Notes)* (2016), pp. 560–568.
[24]E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, "Audio based bird species identification using deep learning techniques," in *CLEF (Working Notes)* (2016), pp. 547–559.
[25]K. J. Piczak, "Recognizing bird species in audio recordings using deep convolutional neural networks," in *CLEF (Working Notes)* (2016), pp. 534–543.

[26]D. Stowell and M. D. Plumbley, "Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning," PeerJ **2**, e488 (2014).
[27]A. Thakur, V. Abrol, P. Sharma, and P. Rajan, "Compressed convex spectral embedding for bird species classification," in *Proceedings of Int. Conf. Acoust. Speech, Signal Process. (ICASSP)* (April, 2018).
[28]A. Thakur, V. Abrol, P. Sharma, and P. Rajan, "Deep convex representations: Feature representations for bioacoustics classification," in *Proceedings of Interspeech* (2018).
[29]Y. Chen, J. Mairal, and Z. Harchaoui, "Fast and robust archtypal analysis for representation learning," in *Proceedings of Comp. Vision Pattern Recog. (CVPR)* (2014), pp. 1478–1485.
[30]K. Qian, Z. Zhang, A. Baird, and B. Schuller, "Active learning for bird sound classification via a kernel-based extreme learning machine," J. Acoust. Soc. Am. **142**(4), 1796–1804 (2017).
[31]D. Chakraborty, P. Mukker, P. Rajan, and A. Dileep, "Bird call identification using dynamic kernel based support vector machines and deep neural networks," in *Proceedings of Int. Conf. Mach. Learn. App.* (2016).
[32]A. Thakur, R. Jyothi, P. Rajan, and A. Dileep, "Rapid bird activity detection using probabilistic sequence kernels," in *Proceedings of European Sig. Process. Conf. (EUSIPCO)* (2017), pp. 1754–1758.
[33]V. Abrol, P. Sharma, A. Thakur, P. Rajan, A. Dileep, and A. K. Sao, "Archetypal analysis based sparse convex sequence kernel for bird activity detection," in *Signal Processing Conference (EUSIPCO), 2017 25th European* (IEEE, 2017), pp. 1774–1778.
[34]D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proceedings of Int. Conf. Pattern Recognition (ICPR)* (2014), pp. 34–39.
[35]F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of Comp. Vis. Pattern Recog.* (2015), pp. 815–823.
[36]O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric learning with adaptive density discrimination," in *Proceedings of Int. Conf. Learn. Represent.* (2016).
[37]G. Pahariya, B. Ravindran, and S. Das, "Dynamic class learning approach for smart CBIR," in *National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics* (2018), pp. 327–337.
[38]C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of AAAI* (2017), Vol. 4, p. 12.
[39]J. Driedger, M. Müller, and S. Disch, "Extending harmonic-percussive separation of audio signals," in *Proceedings of ISMIR* (2014), pp. 611–616.
[40]A. Pankajakshan, A. Thakur, D. Thapar, P. Rajan, and A. Nigam, "All-conv net for bird activity detection: Significance of learned pooling," in *Proc. Interspeech* (2018).
[41]A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proceedings of Advances in Neural Information Processing Systems* (1992), pp. 950–957.
[42]L. Maaten and G. Hinton, "Visualizing data using t-sne," J. Mach. Learn. Res. **9**, 2579–2605 (2008).
[43]http://www.macaulaylibrary.org.
[44]http://artsci.ucla.edu/birds/database.html.
[45]https://figshare.com/s/4af71d71d94e04afcd5f.
[46]J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck, and S. Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," PLoS One **11**(11), e0166866 (2016).
[47]B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeva, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proceedings of Interspeech* (2013).
[48]V. Van Asch, "Macro and micro-averaged evaluation measures," Belgium: CLiPS (2013).
[49]See supplementary material at https://doi.org/10.1121/1.5118245 for more experiments and other details.