Multiview Embeddings for Soundscape Classification

Dhanunjaya Varma Devalraju, Padmanabhan Rajan.
School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi
s18023@students.iitmandi.ac.in, padman@iitmandi.ac.in

***Abstract*—Acoustic scenes (or soundscapes) can be composed of various background and foreground sound events. Classification of soundscapes have to deal with the overlap of these sound events. In this work, we propose to reduce this overlap by considering foreground sound events and background sound events as multiple views of the soundscape. Robust principal components analysis is used to decompose a soundscape into the background view and the foreground view. We can control the amount of information retained in this decomposition by using the subspace projection technique of nuisance attribute projection. We represent the audio samples as features from a convolutional neural network, and view-invariant representations are derived using a deep neural network based multi-view learning algorithm. Experimental results demonstrate the effectiveness of our proposed method on standard datasets for acoustic scene classification.**

***Index Terms*—Acoustic scene classification, robust principal component analysis, subspace projection, multi-view learning.**

I. Introduction

In numerous real-world applications, the data involved can be considered as being made up of constant (or slow-changing) background events, and sparse (or occasional) foreground events. A good example of this is a video captured by a surveillance camera. Here, the slow-changing background is inter-spaced by occasional movement of one or more objects or people. For analysing applications such as these, it may be useful to decompose the data into the background and the foreground. In general, the data matrix M can be decomposed as

$$M = L + S,$$

where, the matrix L corresponds to the low-rank component (the constant or slow-changing background events) and the matrix S represents the sparse component (the occasional or sparse foreground events). Such a decomposition of data into the background and the foreground components can be achieved by employing the statistical technique of *robust principal component analysis* (RPCA) [1], [2].

Data representing acoustic scenes (soundscapes) also can be seen in this fashion. For example, if we consider a soundscape from the "beach" scene, sounds like "waves", "wind" and "water splashing" constitute the background events, and are most likely to be present in the entire duration of the recording. The other sound events like "children playing" and "people shouting", which are sparse and are likely to be present for a short duration at some particular time in the recording, corresponds to the foreground events. Another example is the soundscape "bus". The sound of the engine is the low-rank component; the sound of door opening, people talking is the sparse component.

In previous work [3] [4], we had applied RPCA to the analysis of acoustic soundscapes in order to separate the data into foreground and background. This separation allows us to use the background to help discriminate events with similar foreground, and vice versa. Controlled suppression of the background (foreground) can be performed via nuisance attribute projection or NAP [5], by considering the background (foreground) as the nuisance attribute. For a data vector $\mathbf{x}$ representing an audio recording, this is achieved by

$$\tilde{\mathbf{x}} = \mathbf{x} - BB^T\mathbf{x} = (I - BB^T)\mathbf{x} \tag{1}$$

Here $\tilde{\mathbf{x}}$ denotes the nuisance-removed vector, B is a basis matrix whose columns span the nuisance space, and I is the identity matrix. The representation corresponding to the suppression of the foreground is termed as the *background view* (here the basis $B = B_f$ spans the foreground), and the representation corresponding to the suppression of the background is termed as the *foreground view* (here the basis $B = B_b$ spans the background).

Since both representations come from the same audio signal, this motivates considering them as *multiple views* of the same data. Conventionally, representations corresponding to multiple views can be combined at the feature level or at the decision level. These approaches may not be able to effectively leverage the information specific to individual views. Multi-view learning, on the other hand, annotates one function to model a particular view, and jointly optimizes all the functions to exploit the redundant or complementary information present in the different views.

The challenges of handling multiple view information include view discrepancy [6]–[10]. Additionally, discriminant information (i.e., class labels) can be incorporated into multi-view analysis. Typically, this is done by utilizing the Fisher criterion, simultaneously maximizing the between-class variation and minimizing within-class variation.

Deep learning versions of discriminant multi-view analysis have been developed. This helps to learn complex non-linear relationships from multiple views. A recently proposed method is the Multi-view Linear Discriminant Analysis Network (MvLDAN) [10], which simultaneously considers the cross-view relationship of all views and the discriminant information of all labels.

The main contribution of this work can be summarized as

follows:

- We introduce acoustic scene classification as a multi-view learning problem with two views. These views are obtained from two different representations, namely the background-suppressed representation (the foreground view) and the foreground-suppressed representation (the background view).
- We utilize a framework that uses robust PCA and NAP to derive the multiple views.
- We explore a deep neural network (DNN) based subspace learning algorithm, the Multi-view Linear Discriminant Analysis Network (MvLDAN) [10] to solve the proposed two-view problem, resulting in multiview embeddings that can be used with any classifier.

The rest of this paper is organized as follows. Section II reviews the related works. Section III describes the RPCA framework in brief. Section IV describes the proposed framework. Section V presents the experimental results and section VI analyses the proposed framework. We conclude the paper in Section VII.

II. Related Work

The DCASE challenge website[1] has detailed information including technical reports and code for many techniques for acoustic scene classification (ASC.) Recently, several studies have proposed to use convolutional neural networks (CNN) to tackle the ASC problem. Broadly, CNN based techniques either use a 1D-CNN on the raw audio signal, or a 2D-CNN on the time-frequency representation of the signal. 1D-CNNs are used to extract features directly from time-domain waveform like conventional feature extraction methods that operate in the time domain. Vinayak *et al* [11], proposed to use raw waveform as input to the 1D-CNN based DNN to learn ASC systems in an end-to-end fashion. Arshdeep *et al* [12] proposed to extract feature maps from the intermediate layers of SoundNet (a pre-trained 1D-CNN model) to learn ensemble models for classification. 2D-CNNs, on the other hand, use inputs including constant-Q transform spectrogram [13] - [14], scalogram [15] and log-mel spectrogram [16] - [17]. These time-frequency representations of an audio signal are treated as images, and features are extracted from these representations using 2D-CNN's. In SubSpectralNet [18], sub-spectrograms are used as input to a 2D-CNN, where sub-spectrograms are band-wise crops of the mel-spectrogram. More recently, Zhang *et al* [19], represents the input audio as a log-mel spectrogram and treats it as an ordered segment-level sequence, and trained 3D-CNN's in an end-to-end fashion for classification. In this paper, we use the pre-trained 2D-CNN model L^3-Net [20] on the mel-spectrogram representaion of the audio signal to extract a fixed-length vector $\mathbf{x}$.

Initial studies of using RPCA followed by NAP, for foreground and background separation and its role in soundscape classification were published in [3] and [4]. In [3], RPCA was utilized in the construction of foreground and background bases to be used for NAP. The vectors obtained after NAP was used to perform classification. In [4], class-specific NAP bases were created utilizing RPCA, to get class-specific embeddings. An attention mechanism was used to develop a final embedding, combining the information from the class-specific embeddings. The present study has the following differences from the published works [3] [4]: (a) RPCA is used to obtain foreground and background representations, and NAP bases are learnt from class means. No class-specific bases are used, nor is there an attention mechanism. (b) Foreground-suppressed embeddings and background-suppressed embeddings are considered as multiple-views of the same audio recording. The MvLDAN is used to solve this multi-view classification problem. (c) A detailed study of the information available in the foreground and background components is done, the method is evaluated on standard ASC datasets, and is compared with recent methods.

[1]http://dcase.community/

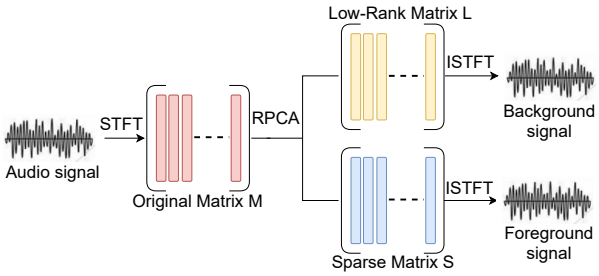

Fig. 1: RPCA based foreground and background separation applied to an audio signal [21].

III. Applying Robust Principal Component Analysis for the ASC task

Principal component analysis (PCA) is widely used in data analysis to find the underlying low-dimensional structure of data. However, PCA is sensitive to corrupted observations and performs poorly when the data is noisy. In the real world, data corruption is quite common and PCA tends to find the directions which are far from the true directions. Robust PCA (RPCA) overcomes some of these limitations with reasonable assumptions about the data [1]. A well-used formulation of RPCA is the problem of decomposing the data matrix M into the sum of a low-rank matrix L and sparse matrix S [2]. By solving the following convex problem we can recover the low-rank matrix:

$$\begin{aligned} &\text{minimize } ||L||_* + \lambda||S||_1, \\ &\text{subject to } L + S = M. \end{aligned} \tag{2}$$

where M, L and $S \in \mathbb{R}^{n1 \times n2}$, $\lambda > 0$ is a tunable parameter, $||\cdot||_*$ denotes the nuclear norm, i.e., sum of singular values and $||\cdot||_1$ denotes the l_1-norm. We use the procedure proposed by Huang et al. [21], including details on how to solve the above convex problem and the choice of the λ value. The constrained optimization problem given in Equation 2 can be represented as the following augmented Lagrangian function (an unconstrained optimization problem). This unconstrained optimization problem can be solved by using the augmented Lagrange multiplier (ALM) method as described in Algorithm

1.

The augmented Lagrangian for ALM method is given below:

$$\mathcal{L}(L,S,\Lambda) = ||L||_* + \lambda||S||_1 + \langle \Lambda, M-L-S\rangle + \frac{\beta}{2}||M-L-S||_F^2 \quad (3)$$

Here, Λ represents the Lagrangian multiplier, β is a positive scalar and $\frac{\beta}{2}||M-L-S||_F^2$ is the penalty term.

Algorithm 1: (RPCA via the Inexact ALM Method [22])

Input: Observation matrix $M \in \mathbb{R}^{n1\times n2}$, λ.

1 $\Lambda_0 = M/J(M)$; $S_0 = 0$; $\beta_0 > 0$; $\rho > 1$; $k = 0$.

2 **while** *not converged* **do**

3 // Lines 4-5 solve $L_{k+1} = \text{argmin}_L\, \mathcal{L}(L, S_k, \Lambda_k, \beta_k)$.

4 $(U, \Sigma, V) = svd(M - S_k + \beta_k^{-1}\Lambda_k)$;

5 $L_{k+1} = U\, \mathbb{S}_{\beta_k^{-1}}[\Sigma]\, V^T$;

6 // Lines 7 solve $S_{k+1} = \text{argmin}_S\, \mathcal{L}(L_k, S, \Lambda_k, \beta_k)$.

7 $S_{k+1} = \mathbb{S}_{\lambda\beta_k^{-1}}[M - L_{k+1} + \beta_k^{-1}\Lambda_k]$;

8 $\Lambda_{k+1} = \Lambda_k + \beta_k(M - L_{k+1} - S_{k+1})$;

9 $\beta_{k+1} = \rho\beta_k$;

10 $k = k + 1$;

11 **end**

Output: (L_k, S_k)

The procedure given in Algorithm 1, first minimizes $\mathcal{L}$ with respect to L by keeping S constant (lines 4-5). Next, it minimizes $\mathcal{L}$ with respect to S by keeping L constant (line 7). Finally, the Lagrangian multiplier Λ is updated in line 8. These steps are repeated till the algorithm converges. Further, Lin *et al.* [22] proposed to use the initialization of $\Lambda_0 = M/J(M)$ for faster convergence, where $J(M) = max(||M||_2, \lambda^{-1}||M||_\infty)$ and $||\cdot||_\infty$ is the maximum absolute value of the matrix entries.

The $\mathbb{S}_\epsilon[X]$ in line 5 and 7 is the shrinkage or soft-thresholding operator applied to each element $x \in X$ and is defined as:

$$\mathbb{S}_\epsilon(x) = \begin{cases} x - \epsilon, & \text{if } x > \epsilon. \\ x + \epsilon, & \text{if } x < -\epsilon. \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Figure 1 illustrates the procedure of applying RPCA to an audio signal to obtain the background and the foreground signal components. We use the procedure proposed by Huang *et al.* [21], where, RPCA is applied on the spectrogram representation of the audio signal, by treating spectrogram as the data matrix M. The low-rank matrix L and sparse matrix S obtained from data matrix M, post applying RPCA, corresponds to the spectrogram representations of the background and the foreground signal respectively. Later, by using the inverse short-time Fourier transform and the phase of the original signal, the background and the foreground audio signals are reconstructed from the respective spectrograms. The process of RPCA applied to an audio example from the soundscape "beach" is illustrated in Figure 2 using spectrograms.

IV. The Proposed Framework

Let $D = \{\mathbf{x}_{ij} \mid 1 \leq i \leq C; 1 \leq j \leq n_i\}$ be a set of pooled training samples from C classes. Let $D_i^f = \{\mathbf{x}_{i1}^f, \mathbf{x}_{i2}^f, \mathbf{x}_{i3}^f, ..., \mathbf{x}_{in_i}^f\}$ be n_i foreground training samples for class i, $i \in \{1, 2, ..., C\}$. Let $D_i^b = \{\mathbf{x}_{i1}^b, \mathbf{x}_{i2}^b, \mathbf{x}_{i3}^b, ..., \mathbf{x}_{in_i}^b\}$ be n_i background training samples for class i. All these samples $\mathbf{x}_{ij}, \mathbf{x}_{ij}^f, \mathbf{x}_{ij}^b \in \mathbb{R}^d$ are the feature vectors extracted from the last layer of the CNN L^3-Net [20], [23]. $\mathbf{x}_{ij}$ is derived from the input audio sample, $\mathbf{x}_{ij}^f$ and $\mathbf{x}_{ij}^b$ are derived after performing RPCA on the input audio sample, without performing NAP.

Let $\mathbf{m}_i^f$ and $\mathbf{m}_i^b$ be the class means for the foreground and background samples from the i^{th} class i.e., D_i^f and D_i^b respectively. Let $\mathbf{m}^f$ be the set of the all foreground means, $\{\mathbf{m}_i^f \mid \forall i \in \{1, 2, .., C\}\}$ and $\mathbf{m}^b$ be the set of the all background means, $\{\mathbf{m}_i^b \mid \forall i \in \{1, 2, .., C\}\}$. By performing PCA on $\mathbf{m}^f$ and $\mathbf{m}^b$, we obtain the basis $B_f \in \mathbb{R}^{d\times d}$ and $B_b \in \mathbb{R}^{d\times d}$ respectively. This is shown in Figure 3.

The proposed framework is illustrated in Figure 4. Let $\mathbf{x}_{ij}$ represent the embedding of an audio sample. For the moment, let us consider the foreground sound events as nuisance

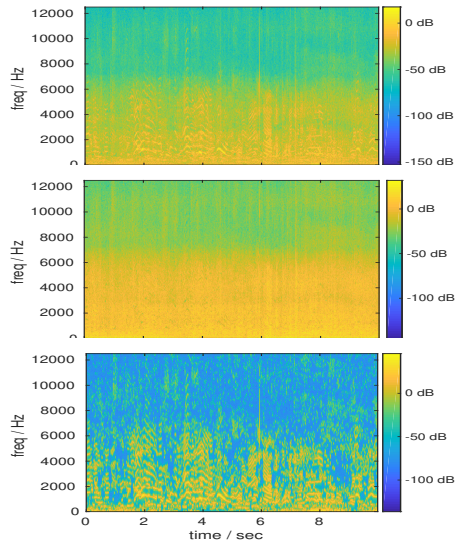

Fig. 2: Mel spectrograms, for an audio example from soundscape "beach", before and after RPCA. Top: The original audio M. Middle: The background L. Bottom: The foreground S. The spectrograms are shown in log scale for better visualization.

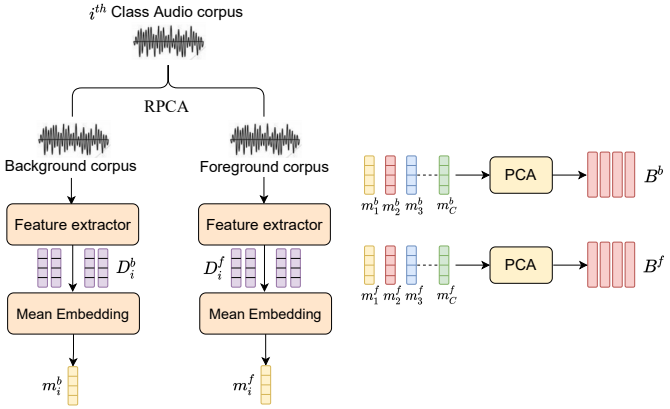

Fig. 3: The background and foreground class-mean basis construction for NAP. The RPCA is performed on the spectrograms as described in section III.

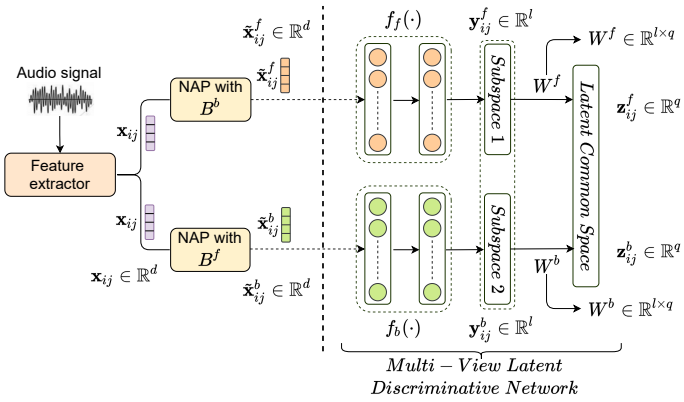

Fig. 4: Proposed framework to suppress the foreground and the background using class-mean NAP bases. A multi-view network, trained end-end is used to derive a final embedding from the latent common space for classification.

attributes. In this case, NAP is performed on $\mathbf{x}_{ij}$ as shown below.

$$\tilde{\mathbf{x}}^b_{ij} = \mathbf{x}_{ij} - B_f B_f^T \mathbf{x}_{ij}, \quad \forall \mathbf{x}_{ij} \in D \tag{5}$$

$$\widetilde{X}^b = \{\tilde{\mathbf{x}}^b_{ij} \,|\, 1 \leq i \leq C; 1 \leq j \leq n_i\} \tag{6}$$

Here $\tilde{\mathbf{x}}^b_{ij} \in \mathbb{R}^d$ represents the foreground-suppressed representation for the embedding $\mathbf{x}_{ij}$. Let $\widetilde{X}^b \in \mathbb{R}^{d\times M}$ be the set of vectors that represents foreground-suppressed representation for all training samples (M). Similarly, if we consider background sound events as nuisance attributes, then NAP is performed on $\mathbf{x}_{ij}$ as shown below.

$$\tilde{\mathbf{x}}^f_{ij} = \mathbf{x}_{ij} - B_b B_b^T \mathbf{x}_{ij}, \quad \forall \mathbf{x}_{ij} \in D \tag{7}$$

$$\widetilde{X}^f = \{\tilde{\mathbf{x}}^f_{ij} \,|\, 1 \leq i \leq C; 1 \leq j \leq n_i\} \tag{8}$$

Here $\tilde{\mathbf{x}}^f_{ij} \in \mathbb{R}^d$ represents the background-suppressed representation for the embedding $\mathbf{x}_{ij}$ corresponding to a single audio sample. The $\widetilde{X}^f \in \mathbb{R}^{d\times M}$ is the set of vectors that represents background-suppressed representation for all training samples. These embeddings projected using the foreground and the background NAP bases can be seen as embeddings representing the same audio sample observed from different views. Let $\tilde{\mathbf{x}}^k_{ij}$ represent the j^{th} sample from the i^{th} class and from the k^{th} view, where $k \in \{b, f\}$ such that b stands for background and f for foreground.

Once we have data for both the views, our objective is to leverage the foreground view and the background view information to learn a discriminative embedding. This learned embedding should retain the common discriminative information: common between the views and discriminative between the classes. The Multi-View Linear Discriminant Analysis Network (MvLDAN) in [10] is used for solving this objective. We use MvLDAN to learn the latent common subspace, projected into which, the embeddings of samples belonging to the same class will be together and the embeddings of the samples from different classes will be apart, irrespective of the views.

The MvLDAN uses multiple neural networks (sub-networks) to model views, with one network for each view. Furthermore, it uses an eigenvalue based objective function to jointly optimize these sub-networks to learn the multi-view space (for nonlinear discriminative representations) and the latent common space (for view invariant representations). The MvLDAN projects the representations obtained from the learned multi-view space into the latent common space to get the view invariant representations (henceforth termed multiview embeddings.)

As shown in Figure 4, the network has two sub-networks, one for each view, and one layer for learning the LDA-like common latent space, which is common for both the sub-networks. We denote the input to the multi-view model as the ordered pair $\widetilde{X} \in \{(\tilde{\mathbf{x}}^b_{ij}, \tilde{\mathbf{x}}^f_{ij}) \mid \tilde{\mathbf{x}}^b_{ij} \in \widetilde{X}^b \; and \; \tilde{\mathbf{x}}^f_{ij} \in \widetilde{X}^f\}$. We represent the sub-network for the k-th view with input $\tilde{\mathbf{x}}^k_{ij}$ and output $\mathbf{y}^k_{ij} \in \mathbb{R}^l$ as:

$$\mathbf{y}^k_{ij} = f_k(\tilde{\mathbf{x}}^k_{ij}), \;\; k \in \{b, f\} \tag{9}$$

The weights, that transforms the sub-network output's to the latent common space are the set of linear transformations and are represented by $\{W^k \mid k \in \{b, f\}; W^k \in \mathbb{R}^{l\times q}\}$. The resulting multiview embedding in the latent common space is represented by $Z = \{\mathbf{z}^k_{ij} = (W^k)^T \mathbf{y}^k_{ij} \mid \mathbf{z}^k_{ij} \in \mathbb{R}^q\}$. The development of the objective function as in [10] is summarized below.

The view discrepancy can be eliminated by considering the pairwise view information in the objective function. This can be formulated as regularized discriminant analysis, where the Tikhonov regularizer is imposed on the LDA objective function, which helps in reducing overfitting. The objective function is formulated as:

$$\underset{f_f, f_b, W^f, W^b}{\operatorname{argmax}} \frac{Tr(S_b) + \lambda Tr(S_c)}{Tr(S_w) + \beta \sum\limits_{k\in\{f,b\}} ||W^k||_F^2} \tag{10}$$

where, $Tr(\cdot)$ is the trace operator, $||\cdot||_F$ is the Frobenius norm and $||W^k||_F^2$ is the Tikhonov regularization term. S_b, S_w represents the between-class and within-class scatter matrices, and S_c represents the pairwise view covariance matrix. The matrices S_b, S_w and S_c, in the latent common space, can be written as $S_b = W^T B W$, $S_w = W^T C W$ and $S_c = W^T D W$ respectively, where $W = [W^{f^T} \; W^{b^T}]^T$, and the partitioned matrices B, C and D can be constructed as follows:

$$B_{kl} = \sum_{i=1}^{C} \frac{1}{N_i} \mathbf{s}^k_i \mathbf{s}^{l^T}_i - \frac{1}{N} \mathbf{s}^k \mathbf{s}^{l^T} \tag{11}$$

$$C_{kl} = \begin{cases} \sum\limits_{i=1}^{C} \left(\sum\limits_{j=1}^{N^k_i} \mathbf{y}^k_{ij} \mathbf{y}^{l}_{ij}{}^T - \frac{1}{N_i} \mathbf{s}^k_i \mathbf{s}^{l^T}_i \right), & \text{if } k = l \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{12}$$

$$D_{kl} = \begin{cases} \mathbf{0}, & \text{if } k = l \\ Y^k Y^{l^T}, & \text{otherwise} \end{cases} \tag{13}$$

where $Y^k = \{y^k_{ij} \,|\, 1 \leq i \leq C; 1 \leq j \leq n_i\}$, $\mathbf{s}^k_i = \Sigma_{j=1}^{N^k_i} \mathbf{y}^k_{ij}$ denotes the sum of i-th class samples from the k-th view, $\mathbf{s}^k = \Sigma_{i=1}^{C} \mathbf{s}^k_i$ denotes the sum of all samples from k-th view and the matrix $\mathbf{0}$ denotes the zero matrix. The variable N_i denotes the total i-th class samples from both the views, and

N is the total number of samples from both the views.

The function in Eq.10 can be transformed and relaxed into equivalent tractable determinant ratio form as given below:

$$\begin{aligned} &\underset{f_f, f_b, W^f, W^b}{\operatorname{argmax}} \; Tr\left((S_w + \beta W^T W)^{-1}(S_b + \lambda S_c)\right) \\ &\qquad = \underset{f_f, f_b, W^f, W^b}{\operatorname{argmax}} \frac{|W^T(B + \lambda D)W|}{|W^T(C + \beta I)W|} \end{aligned} \tag{14}$$

where $|\cdot|$ denotes the matrix determinant. This equation is equivalent to the generalized eigenvalue decomposition problem formulated as $(B + \lambda D)\mathbf{w}_i = \gamma_i(C + \beta I)\mathbf{w}_i$, where $\gamma_i,\ i \in \{1, .., C-1\}$ is the i-th largest eigenvalue and $\mathbf{w}_i$ is the corresponding eigenvector, and $\mathbf{w}_i$ makes up the i-th column vector in W (projection matrix). The γ_i and $\mathbf{w_i}$ are the eigenvalues and eigenvectors of $(C + \beta I)^{-1}(B + \lambda D)$ respectively.

Once the eigen decomposition above is solved, and using Eq. 14, the objective function becomes

$$\begin{aligned} &\underset{f_f, f_b}{\operatorname{argmax}} \; Tr\left((S_w + \beta W^T W)^{-1}(S_b + \lambda S_c)\right) \\ &\qquad = \underset{f_f, f_b}{\operatorname{argmax}} \; Tr\left(W^{-1}(C + \beta I)^{-1}(B + \lambda D)W\right) \\ &\qquad = \underset{f_f, f_b}{\operatorname{argmax}} \; Tr\left((C + \beta I)^{-1}(B + \lambda D)\right) \end{aligned} \tag{15}$$

This is equivalent to:

$$\underset{f_f, f_b}{\operatorname{argmax}} \sum_{i=1}^{C-1} \gamma_i \tag{16}$$

This form of objective function aims to maximize the discriminative power of the two sub-networks by maximizing each one of the eigenvalues. Every eigenvalue quantifies the magnitude of the separation (discriminative variance), from the linear multi-view space to the latent common space, in the direction of the corresponding eigenvector. In [10] it is also discussed that, directly optimizing the objective function in Eq.16 would result in the undesired solution of maximizing only the largest eigenvalue. This means, the function tries to increase the distance between already separated classes, and thus cause a large overlapping between neighboring classes. To overcome this problem, a threshold is applied to filter out the larger eigenvalues and consider only the smaller magnitude eigenvalues when optimizing the parameters of the sub-networks. If there are m eigenvalues below the threshold, then the objective function can be reformulated as:

$$\begin{aligned} &\underset{f_f, f_b}{\operatorname{argmin}} \; -\frac{1}{m}\sum_{i=1}^{m} \gamma_i \;\; \text{with} \\ &\{\gamma_1, ..., \gamma_m\} = \{\gamma_i \mid \gamma_i < min\{\gamma_1, ..., \gamma_{C-1}\} + \epsilon\} \end{aligned} \tag{17}$$

More details regarding the propogation of the gradients to the subnetworks are available in [10]. Also, we set the free parameters λ, β and *threshold* to values as in [10]. Once the model is trained, multiview embeddings are extracted from the latent common space and any classifier can be utilized for classification.

The effect of applying the multi-view discriminative learning is illustrated in Figure 5 as t-SNE plots. Plots (a) and (b) show the effect of reduction of the view discrepancy due to the multiview embeddings. Plots (c), (d) and (e) show the embeddings corresponding to feature extraction, foreground supression after NAP, and multiview projection in the latent common space. The colours represent classes, and it can be seen that the overlap reduces in the multiview embedding space.

V. Experimental details

In this section, we describe the experiments used to evaluate the proposed framework for acoustic scene classification, with the multiview embeddings $\mathbf{z}^k, k \in \{f, b\}$ used as inputs to a classifier. Primarily, we use a simple one-against-one support vector machine (SVM) with a linear kernel; though we evaluate with other classifiers as well. For comparison, we use a baseline model that uses features from the input audio sample (considered together with foreground and background), without NAP or MvLDAN. We also compare the proposed approach with only NAP and no MvLDAN, and with recent works that use several other approaches.

Datasets: We evaluate the proposed framework using four acoustic scene classification (ASC) datasets.

a) *DCASE datasets*: We use datasets from DCASE 2017 [24], 2018 [25] and 2019 [26] challenges. DCASE 2017 task 1 dataset has 2 subsets namely development dataset (4680 10-second audio samples) and evaluation dataset (1620 10-second audio samples). The audio samples of this dataset correspond to 15 different acoustic scenes. Whereas, DCASE 2018 and 2019 task 1A datasets have 10 classes and their respective development datasets have samples 8640 and 13370. For DCASE 2017, we use the development dataset to train the model and report the results obtained over the evaluation dataset. For DCASE 2018 and 2019, since the evaluation dataset labels are not publicly available, we use only the development dataset to validate the proposed framework by following the challenge guidelines.

b) *LITIS Rouen Audio scene dataset* [27]: This dataset comes with 19 different scenes comprising 19 acoustic classes. This dataset set contains 1500 minutes of audio recordings, recorded at a sampling rate of 22050 Hz. Each recording is further split into audio segments of length 30 seconds. This dataset comes with a total of 3026 audio samples. The test and train sets are prepared by following the protocol provided in [27], such that the training set contains 80% of the samples and the testing set contains the remaining 20% of the samples.

Feature extraction: We use Openl3 python library to extract deep audio embeddings from an audio sample [20] [23]. Openl3 is a open source implementation of L^3-Net for deep audio and image embeddings [20] [23]. We use the default model trained with the music subset of AudioSet [28] to extract deep audio embeddings. The extracted embedding corresponding to an audio sample is averaged to get a column vector of length 6144×1 vector ($d = 6144$).

Learning bases for NAP: We utilize the Matlab implementation of RPCA in [21] to separate foreground and

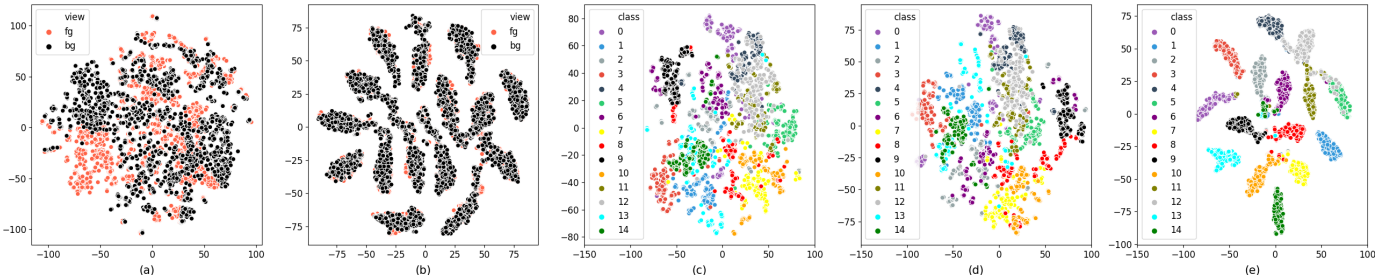

Fig. 5: Various t-SNE plots of embeddings, for DCASE2017 dataset. Plot (a) shows the embeddings $\tilde{\mathbf{x}}^f$ and $\tilde{\mathbf{x}}^b$, which show high overlap. (b) shows the embeddings $\mathbf{z}^f$ and $\mathbf{z}^b$; the view discrepancy is more-or-less removed. (c) shows the embeddings $\mathbf{x}$ from the feature extractor, with different colours representing classes. (d) shows the background suppressed embeddings $\tilde{\mathbf{x}}^f$. (e) shows the multiview embeddings $\mathbf{z}^f$; it can be seen that the multiview embeddings have reduced class overlap.

TABLE I: View sub-network configuration.

Layer	*Output Dimension*	*Activation function*
fc1	512	ReLU
fc2	128	ReLU
fc3	15	Linear

background audio signals from an audio sample (see Figure 1). Post separation, features are extracted from the foreground as well as background audio signals using Openl3 library as discussed earlier. Then, class-means are calculated for each class from the foreground as well as background samples. PCA is then performed on the class means of foreground and the background samples to get nuisance basis B_f (for the foreground) and B_b (for the background) as shown in Figure 3. We can control the amount of foreground and background to be removed by varying the number of principal components used as columns in B_f and B_b respectively.

Model Architecture: The neural network architecture of view sub-network is given in the Table I. Both the sub-networks uses the same architecture. The model configuration has three fully connected layers with number of neurons 512, 128 and 15; all the layers other than last layer uses *ReLU* activation and last layer uses linear activation. We use same model configuration across all the datasets for training and evaluation. Also, the free parameters λ, β and threshold values are set as 0.0001, 1 and 1. The model is trained by using the Adam optimizer and the objective function given in Eq.16. The network in Figure 4 transforms each input ordered pair $(\tilde{\mathbf{x}}^b_{ij}, \tilde{\mathbf{x}}^f_{ij})$ to $(\mathbf{z}^b_{ij}, \mathbf{z}^f_{ij})$ in the latent common space.

Results: We get the multiview embeddings $\mathbf{z}^f$ and $\mathbf{z}^b$ from the latent common space and use them in four ways: (a) using the foreground view embeddings $\mathbf{z}^f$, (b) the background view embeddings $\mathbf{z}^b$, (c) their early fusion and (d) their late fusion. The results of the baseline model and the proposed method using the SVM classifier are given in Table II (last 5 rows).

The reported results in Table II are obtained by using the following number of principal components while applying NAP using bases B_f and B_b. For DCASE 2017, the foreground view uses 1 column of B_b and the background view uses 250 columns of B_f. For all other datasets, the foreground view uses 1 column of B_b and the background view uses 150 columns of B_f. These values are determined experimentally.

For DCASE 2017, we obtained 5% improvement while using the foreground view embeddings , 1% while using the background view embeddings, 3% after their early fusion and 3% after their late fusion. For DCASE 2018, the results were around 1%, 2%, 3% and 2%, and for DCASE 2019, 3%, 4%, 4% and 4%. For LITIS Rouen, we get 3%, 3%, 3% and 3% improvements. These results are with respect to the baseline.

We also compare our results with the systems reported in the DCASE challenge, that do not use data argumentation, and to systems with number of ensemble subsystems less than 10 (first 8 rows in Table II). Our results are consistent across the datasets, and are comparable with the results of the above systems.

VI. Discussion

A. Design choices

In this section, we justify some of the design choices made in the proposed framework.

The use of RPCA: RPCA is used to estimate the NAP bases, which in turn is used to estimate the foreground and background views. As shown in Table III, the RPCA algorithm reliably estimates the low-rank and sparse components for short (2 sec, 5 sec), as well as long (60 sec) duration recordings. The classification accuracy does not change significantly with the duration of the recording.

The performance of separation using RPCA was also evaluated on synthetic soundscapes of varying complexity. The following procedure was followed:

1) Synthetic audio samples of varying complexity levels are generated using Scaper [51]. Complexity levels range from low (meaning foreground dominates) to high (background dominates). The source background and foreground events used to compile each audio sample are saved for future use (required for step 3).
2) RPCA is performed on the audio samples to obtain the corresponding background and foreground signals.
3) SI-SNR improvement (SI-SNRi) [52] metric is computed between the background obtained from RPCA and the source background. The same is also computed between the foreground obtained from RPCA and the source foreground.

TABLE II: Comparison of classification results. The first three columns give the results for DCASE 2017, 2018 and 2019 datasets, the last column gives the results for LITIS Rouen dataset. The first row specifies the dataset name and the last row gives the baseline and proposed framework results. For DCASE datasets the second and third row provides the results from DCASE challenge entries and from the published papers respectively. For LITIS Rouen datasets the second row provides the results from published papers. View fg uses $\mathbf{z}^f$ as the embedding, View bg uses $\mathbf{z}^b$.

DCASE 2017 (eval)	*Acc. (%)*	*DCASE 2018 (dev)*	*Acc. (%)*	*DCASE 2019 (dev)*	*Acc. (%)*	*LITIS Rouen*	*Acc. (%)*
Ivan [29]	71.7	Hao [30]	73.6	Wang [31]	73.5	Yin [32]	96.4
Hyder [33]	74.1	Christian [34]	74.7	Naranjo-Alcazar [35]	77.1	Huy [36]	96.6
Zhengh [37]	77.7	Zhang [38]	75.3	Wang [39]	78.8	Ye [40]	97.1
Han [41]	80.4	Li [42]	76.6	Lei [43]	79.6	Huy [44]	97.8
Waldekar [45]	69.88	Zhao [46]	72.70			Huy [47]	98.7
Devalraju [4]	75.06	SubSpectralNets [18]	74.08				
Wu [48]	75.40	SeNoT-Net [19]	77.19	SeNoT-Net [19]	80.34		
Chen [49]	77.16	ATReSN-Net [50]	77.87	ATReSN-Net [50]	80.68		
Baseline model	71.54	Baseline model	71.88	Baseline model	72.04	Baseline model	94.05
View fg + SVM	77.09	View fg + SVM	73.11	View fg + SVM	75.32	View fg + SVM	97.52
View bg + SVM	72.78	View bg + SVM	74.10	View bg + SVM	76.13	View bg + SVM	97.52
Early Fusion + SVM	74.63	Early Fusion + SVM	74.86	Early Fusion + SVM	76.77	Early Fusion + SVM	97.85
Late Fusion + SVM	75.18	Late Fusion + SVM	74.42	Late Fusion + SVM	76.68	Late Fusion + SVM	97.19

TABLE III: Results for various recording durations from DCASE 2017 data. The standard DCASE 2017 data has 10 second recordings.

System Name	*2 sec*	*5 sec*	*10 sec*	*60 sec*
Baseline model	69.1	71.26	71.54	73.39
NAP-background + SVM	67.51	69.01	69.26	66.95
NAP-foreground + SVM	52.89	52.87	53.83	59.65

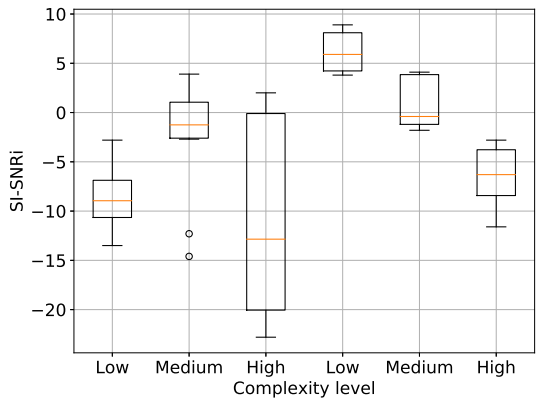

Fig. 6: Illustration of SI-SNR improvement corresponding to audio samples of varying complexities (High, Medium and Low). The first three boxplots correspond to TDCN++ and the last three correspond to RPCA.

4) For comparison, we compare the separation performance of RPCA with that of the TDCN++ method described in [52] using the same metric. The results are shown in Figure 6.

The plots below indicate that the separation of RPCA degrades gracefully as the complexity of the soundscape increases. More details about generating the soundscapes, computing the SI-SNR, and links to the synthetic data are provided in the supplementary material.

The assumption of the RPCA algorithm is that the data has

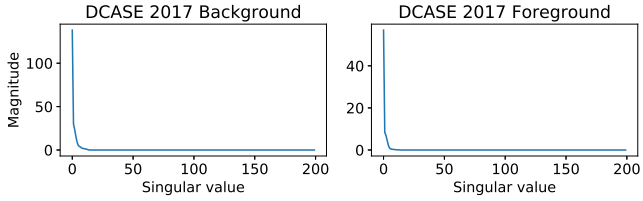

Fig. 7: Plot of singular values obtained with the means of the classwise embeddings from the background as well as the foreground (see Figure 3) corresponding to DCASE 2017. Other datasets too have a similar structure.

a low-rank background component and a sparse foreground component. If the underlying acoustic scene does not have this structure (for example, if it is a soundscape with only a dominant foreground component and little or no background), then the foreground and background views produced may no longer be reliable.

PCA bases for NAP: For deriving the bases B_b and B_f for NAP, we use simple PCA. Figure 7 shows the plot of the singular values of the means of the class-wise embeddings, used to determine the basis for NAP. It can seen that the singular values have a sharp bend, indicating an underlying linear subspace structure, where most of the information lies. This subspace is captured by PCA. More advanced techniques such as kernel PCA and dictionary learning did not result in estimating NAP bases which are more useful. More details are given in the supplementary material.

Other classifiers: The multiview embeddings generalize well with other classifiers such as K-nearest neighbours (KNN), naive Bayes, and deep neural networks (DNN.) These classifiers give performance similar to that of the SVM. The performance of these classifiers are given in Table IV.

Need for multi-view analysis: The utility of the MvLDAN framework was evaluated by omitting the blocks after the dotted line in Figure 4. The background-suppressed ($\tilde{\mathbf{x}}^f$) and the foreground-suppressed ($\tilde{\mathbf{x}}^b$) embeddings were used in a supervised multi-input deep neural network, and the

TABLE IV: Results of using the proposed embeddings with various classifiers on datasets DCASE 2017, 2018 and 2019. *View fg* uses $\mathbf{z}^f$ as the embedding, *View bg* uses $\mathbf{z}^b$. The last two columns indicate early fusion and late fusion of the embeddings.

DCASE 2017				
Classifier	*View fg*	*View bg*	*Early Fusion*	*Late Fusion*
KNN	76.36	74.32	75.93	75.74
Naive Bayes	71.05	68.27	69.94	69.81
DNN	76.17	73.46	74.75	75.06
DCASE 2018				
Classifier	*View fg*	*View bg*	*Early Fusion*	*Late Fusion*
KNN	73.95	74.94	75.58	75.06
Naive Bayes	70.53	72.88	72.24	72.16
DNN	73.19	73.99	74.74	74.10
DCASE 2019				
Classifier	*View fg*	*View bg*	*Early Fusion*	*Late Fusion*
KNN	75.58	76.63	76.80	77.04
Naive Bayes	74.34	75.03	76.13	76.03
DNN	74.67	75.05	76.00	75.69

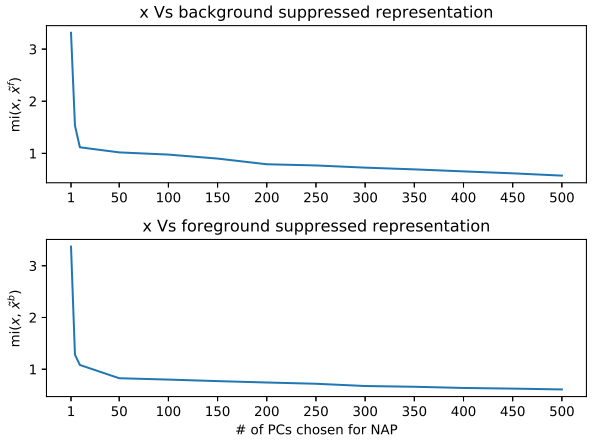

Fig. 8: Plot of average pair-wise mutual information score of all training samples, from all classes. Top row: between audio representation ($\mathbf{x}$) and background suppressed representation ($\tilde{\mathbf{x}}^f$). Bottom row: between audio representation ($\mathbf{x}$) and foreground suppressed representation ($\tilde{\mathbf{x}}^b$).

classification accuracy obtained is 68.52 %, which is much lower than the proposed systems that utilize the multi-view information (last four rows of Table II). This indicates that the embeddings $\tilde{\mathbf{x}}^f$, $\tilde{\mathbf{x}}^b$ contain significant classwise overlap. This is not surprising, since the NAP basis was learnt without any class information. More details about the setup and results using various supervised fusion methods to combine the background-suppressed ($\tilde{\mathbf{x}}^f$) and the foreground-suppressed ($\tilde{\mathbf{x}}^b$) embeddings are provided in the supplementary material.

B. Analysis of information and distance

In this section, we estimate the information available between various representations in the proposed pipeline. We also estimate the distance between classes to measure separability.

Figure 8 illustrates how the mutual information (mi) between the audio representation ($\mathbf{x}$) and foreground or background suppressed representation ($\tilde{\mathbf{x}}^b$ or $\tilde{\mathbf{x}}^f$) varies with respect to the number principal components (PC) chosen for NAP. From the plot it is evident that mutual information decreases as we choose more and more PCs (indicating that more of the nuisance component is removed.) Also, the fall is drastic after the first few PCs.

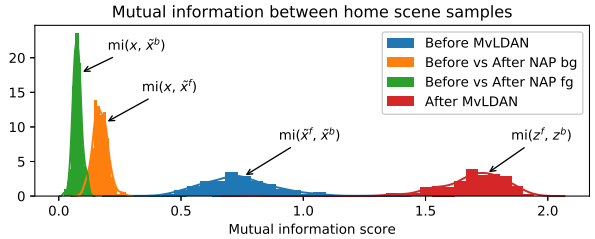

Fig. 9: Histogram of the pair-wise mutual information between a sample of "home" class from different representations. The green plot shows the distribution of the pair-wise mutual information between audio representation ($\mathbf{x}$) and foreground suppressed representation ($\tilde{\mathbf{x}}^b$) of "home" class, and the orange plot between audio representation ($\mathbf{x}$) and background suppressed representation ($\tilde{\mathbf{x}}^f$). The blue plot shows the distribution between background view and foreground view representations of before MvLDAN and the red plot after MvLDAN.

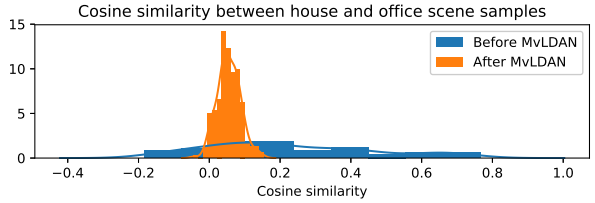

Fig. 10: Histogram of the cosine similarity between a sample of "home" class and the mean of a "office" class. The blue plot shows the histogram before applying MvLDAN and the orange plot after applying MvLDAN.

Figure 9 shows the histogram of the pair-wise mutual information between a sample of "home" class from various representations $\mathbf{x}$, $\tilde{\mathbf{x}}^b$, $\tilde{\mathbf{x}}^f$, $\mathbf{z}^\mathbf{b}$ and $\mathbf{z}^\mathbf{f}$. This plot shows that the mutual information increases as we progress with each step in the proposed framework. From this plot, we can see that there is slight increase in the mutual information score between the audio representation ($\mathbf{x}$) and background suppressed representation ($\tilde{\mathbf{x}}^f$) compared to audio representation ($\mathbf{x}$) and foreground suppressed representation ($\tilde{\mathbf{x}}^b$). Furthermore, the mutual information (blue histogram) between $\tilde{\mathbf{x}}^b$ and $\tilde{\mathbf{x}}^f$, which are input to the MvLDAN is high compared to their respective mutual information scores with $\mathbf{x}$. The MvLDAN further increases the mutual information (red histogram) between the background view and the foreground view samples in the latent common space. This shows that the proposed framework gradually increases the mutual information between the views while removing the view discrepancy.

Figure 10 shows the histogram of the cosine similarity

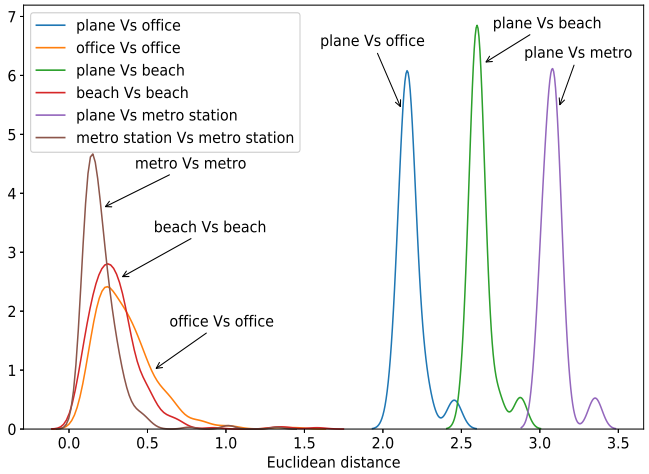

Fig. 11: Histogram of the Euclidean distances between multiview embeddings of various classes. The blue color plot is the distribution of the Euclidean distances between samples of "plane" (unseen class) and the mean of "office" class (seen class). The orange color plot is the distribution of the Euclidean distances between samples of "office" and the mean of "office" class. The other colours indicate other seen and unseen pairs.

between multiview embeddings of one class and the mean of a confusable class before and after applying MvLDAN. The x-axis represents the cosine similarity and it takes values between -1 and 1, 1 being most similar and -1 being not similar. This plot shows that, most of the samples are similar to the mean of the confusable class before MvLDAN. Post MvLDAN there is a shift in distribution, meaning that the samples of one class has cosine similarity close to 0 with respect to the mean of the other confusable class. This is shown in the Figure 10 for scenes "home" and "office".

The multiview embeddings can also be used to handle unseen classes in out-of-set classification. The histograms in Figure 11 show the Euclidean distance between seen classes ("metro", "beach", "office", from DCASE 2017 dataset) and an unseen class "plane" (from LITIS dataset).

VII. Conclusion

We have presented an approach that exploits the complementary information present in the background-suppressed and the foreground-suppressed representations of an audio sample for acoustic scene classification. The starting point of the audio representation uses L3Net, which is a modern pre-trained CNN. This is followed by the use of robust PCA and nuisance attribute projection to obtain background-suppressed and foreground-suppressed representations. By considering these representations as multiple views, the MvLDAN multiview learning algorithm is utilized to reduce the separation between the views from the same class, and increase the separation for views from different classes. The projection into the MvLDAN latent common space results in the multiview embeddings which help in improved discrimination of acoustic scenes. The performance provided by the proposed approach is comparable to many existing approaches, and illustrates the utility of multiple views of the same data. The multiview embeddings also generalize to other classifiers, and performance may improve with fine-tuning. It is also satisfying to note the consistent ability of RPCA to separate the foreground and background in common acoustic scenes, having varying durations and complexities.

The benefits of the proposed method include an indirect ability to control (though the number of NAP basis used) the amount of information shared between the foreground view and the background view. A disadvantage, however, is the underlying assumption of RPCA: that the data needs to have slow-varying background events and sparse foreground events. This may not be valid in all acoustic scenes and in such situations, alternatives may need to be used.

Extensions of the current approach include the use of other multiview learning frameworks in place of MvLDAN, and alternatives to RPCA for separating the foreground and background. For the latter, methods for source separation may be applicable in certain scenarios.

References

[1] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[3] D. V. Devalraju, P. Rajan, and A. Dileep, "Learning to separate: Soundscape classification using foreground and background," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 21–25.

[4] D. V. Devalraju, H. Muralikrishna, P. Rajan, and A. D. Dileep, "Attention-driven projections for soundscape classification," *Proc. Interspeech 2020*, pp. 1206–1210, 2020.

[5] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition." in *Odyssey*, vol. 4. Citeseer, 2004, pp. 219–226.

[6] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[7] B. Thompson, "Canonical correlation analysis," *Encyclopedia of Statistics in Behavioral Science*, 2005.

[8] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Conference on Data Mining and Data Warehouses (SiKDD 2010)*, 2010, pp. 1–4.

[9] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 188–194, 2015.

[10] P. Hu, D. Peng, Y. Sang, and Y. Xiang, "Multi-view linear discriminant analysis network," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5352–5365, 2019.

[11] V. Abrol and P. Sharma, "Learning hierarchy aware embedding from raw audio for acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1964–1973, 2020.

[12] A. Singh, A. Thakur, P. Rajan, and A. Bhavsar, "A layer-wise score level ensemble framework for acoustic scene classification," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 837–841.

[13] Z. Weiping, Y. Jiantao, X. Xiaotao, L. Xiangtao, and P. Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[14] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, vol. 90, 2016, pp. 1032–1048.

[15] Z. Ren, K. Qian, Y. Wang, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

[16] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1547–1554.

[17] R. Hyder, S. Ghaffarzadegan, Z. Feng, and T. Hasan, "BUET bosch consortium (B2C) acoustic scene classification systems for DCASE 2017," *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[18] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 825–829.

[19] L. Zhang, J. Han, and Z. Shi, "Learning temporal relations from semantic neighbors for acoustic scene classification," *IEEE Signal Processing Letters*, vol. 27, pp. 950–954, 2020.

[20] J. Cramer, H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[21] P. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 57–60.

[22] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[23] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 609–617.

[24] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[25] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13.

[26] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, pp. 164–168, Oct 2019.

[27] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time–frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142–153, 2014.

[28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[29] I. Kukanov, V. Hautamäki, and K. A. Lee, "Recurrent neural network and maximal figure of merit for acoustic event detection," DCASE2017 Challenge, Tech. Rep., September 2017.

[30] W. Hao, L. Zhao, Q. Zhang, H. Zhao, and J. Wang, "DCASE 2018 task 1a: Acoustic scene classification by bi-LSTM-CNN-net multichannel fusion," DCASE2018 Challenge, Tech. Rep., September 2018.

[31] Z. Wang, J. Ma, and C. Li, "Acoustic scene classification based on CNN system," DCASE2019 Challenge, Tech. Rep., June 2019.

[32] Y. Yin, R. R. Shah, and R. Zimmermann, "Learning and fusing multimodal deep features for acoustic scene categorization," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1892–1900.

[33] R. Hyder, S. Ghaffarzadegan, Z. Feng, and T. Hasan, "BUET bosch consortium (B2C) acoustic scene classification systems for DCASE 2017," DCASE2017 Challenge, Tech. Rep., September 2017.

[34] C. Roletscheck and T. Watzka, "Using an evolutionary approach to explore convolutional neural networks for acoustic scene classification," DCASE2018 Challenge, Tech. Rep., September 2018.

[35] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, "DCASE 2019: CNN depth analysis with different channel inputs for acoustic scene classification," DCASE2019 Challenge, Tech. Rep., June 2019.

[36] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[37] Z. Weiping, Y. Jiantao, X. Xiaotao, L. Xiangtao, and P. Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," DCASE2017 Challenge, Tech. Rep., September 2017.

[38] L. Zhang and J. Han, "Acoustic scene classification using multi-layered temporal pooling based on deep convolutional neural network," DCASE2018 Challenge, Tech. Rep., September 2018.

[39] M. Wang and R. Wang, "Ciaic-ASC system for DCASE 2019 challenge task1," DCASE2019 Challenge, Tech. Rep., June 2019.

[40] J. Ye, T. Kobayashi, N. Toyama, H. Tsuda, and M. Murakawa, "Acoustic scene classification using efficient summary statistics and multiple spectro-temporal descriptor fusion," *Applied Sciences*, vol. 8, no. 8, p. 1363, 2018.

[41] Y. Han, J. Park, and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," DCASE2017 Challenge, Tech. Rep., September 2017.

[42] Z. Li, L. Zhang, S. Du, and W. Liu, "Acoustic scene classification based on binaural deep scattering spectra with CNN and LSTM," DCASE2018 Challenge, Tech. Rep., September 2018.

[43] C. Lei and Z. Wang, "Multi-scale recalibrated features fusion for acoustic scene classification," DCASE2019 Challenge, Tech. Rep., June 2019.

[44] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, and A. Mertins, "Audio scene classification with deep recurrent neural networks," *arXiv preprint arXiv:1703.04770*, 2017.

[45] S. Waldekar and G. Saha, "Wavelet transform based mel-scaled features for acoustic scene classification." in *Proc. Interspeech*, vol. 2083, 2018, pp. 3323–3327.

[46] Z. Ren, Q. Kong, J. Han, M. D. Plumbley, and B. W. Schuller, "Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 56–60.

[47] H. Phan, O. Y. Chén, L. Pham, P. Koch, M. De Vos, I. McLoughlin, and A. Mertins, "Spatio-temporal attention pooling for audio scene classification," *arXiv preprint arXiv:1904.03543*, 2019.

[48] Y. Wu and T. Lee, "Enhancing sound texture in cnn-based acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 815–819.

[49] H. Chen, P. Zhang, and Y. Yan, "An audio scene classification framework with embedded filters and a dct-based temporal module," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 835–839.

[50] L. Zhang, J. Han, and Z. Shi, "Atresn-net: Capturing attentive temporal relations in semantic neighborhood for acoustic scene classification," *Proc. Interspeech 2020*, pp. 1181–1185, 2020.

[51] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[52] S. Wisdom, H. Erdogan, D. P. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. R. Hershey, "What's all the fuss about free universal sound separation data?" in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 186–190.