DeLoRes: Decorrelating Latent Spaces for Low-Resource Audio Representation Learning

Sreyan Ghosh [1*], Ashish Seth [1*], Deepak Mittal [2], Maneesh Singh [2], S. Umesh [1]

[1]Speech Lab, Department of Electrical Engineering, Indian Institute of Technology Madras
[2]Verisk Analytics
gsreyan@gmail.com, cs20s030@smail.iitm.ac.in,deepak.mittal@verisk.com,
maneesh.singh@verisk.com,umeshs@ee.iitm.ac.in

Abstract

Inspired by the recent progress in self-supervised learning for computer vision, in this paper we introduce **DeLoRes**, a new general-purpose audio representation learning approach. Our main objective is to make our network learn representations in a resource-constrained setting (both data and compute), that can generalize well across a diverse set of downstream tasks. Inspired from the Barlow Twins objective function, we propose to learn embeddings that are invariant to distortions of an input audio sample, while making sure that they contain non-redundant information about the sample. To achieve this, we measure the cross-correlation matrix between the outputs of two identical networks fed with distorted versions of an audio segment sampled from an audio file and make it as close to the identity matrix as possible. We use a combination of a small subset of the large-scale AudioSet dataset and FSD50K for self-supervised learning and are able to learn with less than half the parameters compared to state-of-the-art algorithms. For evaluation, we transfer these learned representations to 9 downstream classification tasks, including speech, music, and animal sounds, and show competitive results under different evaluation setups. In addition to being simple and intuitive, our pre-training algorithm is amenable to compute through its inherent nature of construction and does not require careful implementation details to avoid trivial or degenerate solutions. Furthermore, we conduct ablation studies on our results and make all our code and pre-trained models publicly available [1].

***Keywords**:* Self-supervised learning, audio classification, representation learning

Introduction

In recent times, unsupervised representation learning, including self-supervised and semi-supervised learning has shown great success across different modalities, for example, text, vision, and speech. Although self-supervised learning (SSL) and semi-supervised learning have achieved great performance on Automatic Speech Recognition (ASR), limited attempts have been made for other speech and audio processing tasks such as speech emotion recognition, speaker identification, language identification, and acoustic scene identification.

Self-supervised learning in speech and audio aims towards learning representations that contain high-level information from acoustic signals which can be further used in diverse sets of downstream tasks. Model weights learned through self-supervision are either used as feature extractors under the linear evaluation protocol (Niizumi et al. 2021; Ling and Liu 2020) or used together with transfer learning for end-to-end fine-tuning with an added prediction-head for the downstream task (Chen et al. 2020; Baevski et al. 2020; Hsu et al. 2021). Features learned through self-supervised speech representation learning have already proven to outperform other low-level features such as filter-banks and mel-frequency cepstral coefficients (MFCCs).

In the past, a limited amount of work has been done in learning general-purpose audio representations that can perform well across a diverse set of downstream tasks beyond speech recognition and the one's proposed suffers from shortcomings. For example, triplet-based objectives used in (Jansen et al. 2017; Shor et al. 2020a) rely on the mining of negative samples and the quality of learned features varies significantly with the sample generation scheme (Shor et al. 2020a). Contrastive learning systems used in (Saeed, Grangier, and Zeghidour 2020) typically work online and rely on a large number of explicit pairwise feature comparisons, which are computationally challenging, given the fact that they require large batches for mining negative samples. Moreover, (Baevski et al. 2020) show that the quality of mined negative samples can affect the performance. On the other hand, the method proposed by (Niizumi et al. 2021) uses a *momentum encoder*, where a moving-average network is used to produce prediction targets for optimizing the MSE loss between two batches of augmented samples of the same audio segments. However, the symmetry-breaking network design has been found crucial for this approach. Moreover, (Chen and He 2020) also conclude that the asymmetry of the learning update, "stop-gradient", is critical to preventing trivial solutions in this type of learning. Adding to this, the requirement of a separate teacher network also doubles the number of parameters required for training.

Prior studies in this domain use large data sets and large architectures for SSL pre-training. However, as correctly

[*] These authors contributed equally.

[1] https://github.com/Speech-Lab-IITM/DeLoRes

pointed out by (Hannun 2021), the main challenges with self-supervision are those of scale, and hence accessibility. SSL would therefore be more accessible given lighter-weight models which could be trained efficiently on fewer data. Additionally, another problem is that many of these methods utilize the time-series aspect of audio signals, i.e., audio segments cropped closer in time are expected to have closer representations, whereas those far away in time are expected to have distanced representations. However, as pointed out by (Niizumi et al. 2021), using contradictory examples of music and gunshot, this cannot be deemed as the optimal strategy in all cases.

In this paper we propose, **DeLoRes**, **De**correlating Latent Spaces for **Lo**w **Res**ource Audio Representation Learning, a simple yet powerful self-supervised pre-training framework to learn general-purpose audio representations of sounds beyond and including speech. Inspired by the Barlow Twins framework (Zbontar et al. 2021), we employ an invariance and redundancy-reduction based objective function, where we try to make the cross-correlation matrix, computed from embeddings of a pair of augmented samples of the same audio segments, as close to the identity matrix as possible. This idea fits well to the acoustic domain wherein the cross-correlation measure is generally used to calculate the correlation between two signals shifted in time. Compared to other methods in the self-supervised audio pre-training domain, our methodology is conceptually simple and avoids trivial solutions by the inherent nature of its construction. We alleviate the requirement of large batches as in (Saeed, Grangier, and Zeghidour 2020) and avoid the reliance on symmetry-breaking network designs for distorted samples using an extra teacher network as in (Niizumi et al. 2021), the requirement for carefully curation of a sample generation scheme for mining negatives as in (Jansen et al. 2017), or the possibility of degenerate solutions as in (Ghosh et al. 2021).

We demonstrate the effectiveness of DeLoRes over 9 challenging and diverse downstream tasks including speech, music, and animal sounds. We pre-train DeLoRes on a small subset (10%) of the large-scale AudioSet dataset (Gemmeke et al. 2017) with a much smaller architecture compared to prior art, and show that only a linear classifier trained over DeLoRes embeddings is competitive in performance to other state-of-the-art (SOTA) algorithms for learning general-purpose audio representations, using much lesser data and compute than their implementations.

Related Work

Recently, SSL applied to on unlabelled data has proven to be very effective, achieving SOTA performance when used alongside transfer learning on various low-resource downstream tasks, in various modalities such as text, image, and speech. It has also achieved significant performance boosts or close to SOTA results under the linear evaluation protocol, which allows only a linear layer to be trained on top of the representations learned by SSL, for a particular downstream task.

SSL algorithms, that have shown great success, include variations of contrastive learning, masked prediction, clustering, or the use of *momentum encoders* optimized with diverse objective functions. A simple yet powerful idea is to compare pairs of image representations to push away representations from different images while pulling together those from transformations, or views, of the same image. (Chen et al. 2020) was the first to propose the use of this framework in CV and proposed to calculate a contrastive loss by taking different augmentations of the same image as positive pairs and other images in the batch as negatives.

Recent *non-contrastive* SSL methods include (Chen and He 2020) and (Grill et al. 2020), where both the network architecture and the parameter updates are modified to introduce asymmetry. The asymmetry in network architecture is created by using a specific "predictor" network and the parameter updates are asymmetric such that the model parameters are only updated using one distorted version of the input, while the representations from another distorted version are used as a fixed target.

In another line of work, clustering methods such as (Caron et al. 2019) propose computing "pseudo-labels" from one view and predicting the cluster assignment using another view of the same image. On similar lines, (Caron et al. 2021) and (Asano, Rupprecht, and Vedaldi 2020) propose the use of non-differential operators. (Caron et al. 2021) draws its inspiration partly from contrastive learning and proposes a swapped cluster prediction problem, whereby they try to predict the cluster assignment of one view from the other and optimize their network using cross-entropy loss.

SSL solving the masked prediction task has been prevalent in the text domain in the form of Masked Language Modelling (MLM) and in speech as Masked Acoustic Modelling (MAM). Most of these approaches aim to either predict the class of the masked entity using a classification objective as in (Hsu et al. 2021; Devlin et al. 2019) or reconstruct the original frame as in (Liu et al. 2020; Liu, Li, and Lee 2021) or to enforce similarity between the prediction of the network for the masked frame and a quantized representation of the original masked frame as in (Baevski et al. 2020).

Prior work on unsupervised audio representation domain is diverse. For example, L3 (Arandjelović and Zisserman 2017), AuDeep (Freitag et al. 2017), autoregressive predictive coding (Chung and Glass 2020), contrastive predictive coding (van den Oord, Li, and Vinyals 2019), metric learning (Jansen et al. 2017), autoencoding (Latif et al. 2020). However, these methods were evaluated on just one or a limited set of downstream tasks. TRILL (Shor et al. 2020b) or TRIpLet Loss network was one of the first works to test its learned representations on a diverse set of downstream tasks. TRILL optimizes a triplet-based objective function whereby the network represents audio such that segments that are closer in time are also closer in the embedding space. The anchor and positives were sampled from the same audio sample while the negatives were sampled from a different sample.

COLA (Saeed, Grangier, and Zeghidour 2020) solves a contrastive task for learning general-purpose audio representations, which outperforms prior art in this domain. They also employ a mining strategy similar to TRILL wherein they generate similar pairs by simply sampling segments from the same audio sample and negative samples from different audio samples. They demonstrate the effectiveness of two objective functions, namely cosine and bilinear similarities, and also show that larger batch sizes help in learning better representations.

On the other hand, BYOL-A (Niizumi et al. 2021) proposes a different approach in that it does not use negative samples. Instead, it directly minimizes the mean squared error of embeddings originating from the same audio segment with contrasts created by data augmentations. This approach also differs from prior-art in which they learn audio representations from a single audio segment without expecting relationships between different time segments of audio samples.

Very recently, DECAR (Ghosh et al. 2021) proposed a clustering framework to learn general-purpose audio representations, based on the Deepcluster framework in CV. Their pre-training approach is based on two forward passes through the network whereby they first cluster the output of the feature extractor network and then use the subsequent cluster assignments as “pseudo-labels” for an augmented version of the same audio sample. In their second pass through the network, the loss between the network prediction of the augmented sample and the pseudo-label from the first pass is used to optimize the network weights.

Inspired by the transformers, that were originally developed for text processing but have shown great success in image and speech domains too, (Gong et al. 2021) proposed SSAST, a self-supervised pre-training strategy for the audio spectrogram transformer AST (Gong, Chung, and Glass 2021). AST is the first convolution-free, purely attention-based model for audio classification. In SSAST, the authors optimize a joint discriminative and generative masked spectrogram patch modeling (MSPM) objective whereby they sum InfoNCE loss and mean-squared-error, where both of the losses are weighted by a hyper-parameter lambda.

While contrastive learning approaches need large batch sizes and careful mining of negative samples for training, one common problem with *non-contrastive* SSL approaches in both CV and the audio domain, is that there exist trivial solutions to the learning objective that these methods avoid via particular implementation choices or they are the result of non-trivial learning dynamics adopted by them. Thus, the recently proposed Barlow Twins makes itself unique from prior art by its new loss function which is conceptually much simpler and avoids pitfalls through its inherent nature of construction. Similar to other approaches, the Barlow Twins also works on a pair of differently augmented views of the same image, however, now it calculates the cross-correlation matrix between the embeddings of the network which was fed with the batches of these augmented views, and tries to make this matrix as close to identity as possible. Our approach in this paper is inspired by the Barlow Twins framework.

Methodology

DeLoRes Architecture

For pre-training the SSL, we use a convolution-based feature encoder. The learned weights are then transferred to other downstream tasks. We experiment with several ConvNet architectures, including the highly-scalable EfficientNet-B0 which has been commonly used in prior work (Saeed, Grangier, and Zeghidour 2020; Ghosh et al. 2021). However, we achieved better results when we resorted to a simpler architecture which was submitted as a solution of Task 6 of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 Challenge (Koizumi et al. 2020; Takeuchi et al. 2020). Beyond being much simpler, the architecture has proven to perform reasonably well in a challenge related to the task we are trying to solve. This architecture has also been used in (Niizumi et al. 2021) for pre-training the SSL.

The network architecture is composed of three Conv2D layers, each followed by a BatchNormalization layer, a rectified linear unit (ReLU) activation function, and a MaxPool2D layer. Finally, we pass the embeddings obtained through a set of linear layers, each followed by a ReLU activation function again and we also use dropout layer between the two layers.

In the SSL pre-training stage, for the *projection head*, instead of three linear layers as in (Zbontar et al. 2021), we use two, with the first one followed by BatchNorm1D and a ReLU activation function. Additionally, we employ a dropout layer between the projection head and encoder output to add regularisation and then pass the final embeddings through a BatchNormalization layer to make them zero mean and unit variance. The number of units in the projection head is a tunable parameter and affects the size of our cross-correlation matrix $\mathcal{C}$ as defined in equation (4).

As mentioned earlier and shown in Fig. 1, we use the projection head just for our SSL pre-training task and discard it after this step. More layer-specific details can be found in Table 1.

DeLoRes Learning Algorithm

Co-variance and Cross-correlation Given two time series x_t and y_t of N number of samples, where t is the time index, and $x_{t-\tau}$ is the delayed version of x_t by τ samples, The cross-covariance matrix between the pairs of signals can be calculated as:

$$\sigma_{xy}(\tau) = \frac{1}{N-1} \sum_{t=1}^{N} (x_{t-\tau} - \mu_x)(y_t - \mu_y) \qquad (1)$$

where μ_x and μ_y are the means of x_t and y_t, respectively. By normalizing the equation (1), the cross-correlation of the two signals can be calculated as:

$$\rho_{xy}(\tau) = \frac{\sigma_{xy}(\tau)}{\sqrt{\sigma_{xx}(0)\sigma_{yy}(0)}} \qquad (2)$$

Objective Function We employ the Barlow Twins objective function (Zbontar et al. 2021), defined in equation (3), to optimize our network.

$$\mathcal{L}=\underbrace{\sum_i \left(1-\mathcal{C}_{ii}\right)^2}_{\text{Invariance Term}}+\lambda \underbrace{\sum_i \sum_{j\neq i} \mathcal{C}_{ij}^2}_{\text{Redundancy Reduction Term}} \tag{3}$$

where $\mathcal{C}$ is defined in equation (4). The *invariance term* of the objective function tries to make the diagonal elements of the cross-correlation matrix close to 1, which intuitively makes the embedding invariant to the distortions applied. The *redundancy reduction term*, which is weighted by a positive constant, λ, sums up the off-diagonal elements of the cross-correlation and tries to make it close to 0. Intuitively, this decorrelation reduces the redundancy between output embeddings so that they contain non-redundant information about the sample and preserve only useful information.

The Barlow Twins objective function calculates the cross-correlation matrix between the two embeddings obtained from the encoder output for a pair of differently augmented versions of an audio sample and tries to make it as close to the identity matrix as possible.

Layer – #	**Layer Parameters**	**Output**
Conv2D-1	3x3@64	[B, 64, 64, 96]
BatchNorm2D-2		[B, 64, 64, 96]
ReLU-3		[B, 64, 64, 96]
MaxPool2D-4	2x2,stride=2	[B, 64, 32, 48]
Conv2D-5	3x3@64	[B, 64, 32, 48]
BatchNorm2D-6		[B, 64, 32, 48]
ReLU-7		[B, 64, 32, 48]
MaxPool2D-8	2x2,stride=2	[B, 64, 16, 24]
Conv2D-9	3x3@64	[B, 64, 16, 24]
BatchNorm2D-10		[B, 64, 16, 24]
ReLU-11		[B, 64, 16, 24]
MaxPool2D-12	2x2,stride=2	[B, 64, 8, 12]
Reshape-13		[B, 12, 512]
Linear-14	out=2048	[B, 12, 2048]
ReLU-15		[B, 12, 2048]
Dropout-16	0.3	[B, 12, 2048]
Linear-17	out=2048	[B, 12, 2048]
ReLU-18		[B, 12, 2048]
$\max(\cdot) \oplus (\cdot)$-19		[B, 2048]

Table 1: Encoder network architecture

Learning Framework Figure 1 shows the complete end-to-end pre-training and fine-tuning procedure. For a batch of audio samples, we first chunk a random segment x of length 980 ms, from each audio sample, and then extract the log-compressed mel-filterbanks for each sample. Next, these samples are batched to obtain X, and we create exactly two augmented views X^{A} and X^{B}, for each sample in the batch by applying different augmentations K. The ConvNet feature encoder f is then used to obtain batch of embeddings Z^{A} and Z^{B}. These outputs are then used to calculate $\mathcal{C}$, the cross-correlation matrix, where each element in the matrix is calculated by equation (4) along the batch dimension.

$$\mathcal{C}_{ij}=\frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b \left(z_{b,i}^A\right)^2}\sqrt{\sum_b \left(z_{b,j}^B\right)^2}} \tag{4}$$

where b indexes batch samples, i and j indexes the vector dimension of the embeddings, and $\mathcal{C}$ can take values in a range -1 and 1. While (Zbontar et al. 2021) show how the design of the loss function was motivated by the Information Bottleneck theory, (Tsai et al. 2021) have a great explanation why Barlow Twins can also be called *negative-sample-free contrastive learning*. This explanation is beyond the scope of this paper and we would like our readers to refer to these papers directly.

Data Sets

We pre-train DeLoRes embeddings on a combination of a class-balanced subset of the large-scale Google AudioSet (Gemmeke et al. 2017) and the FSD50K dataset (Fonseca et al. 2020). The AudioSet subset consists of 0.2 million utterance or approx. 10% of the total number of audio files in the original large-scale data set (2 million). We do this for two primary reasons:

- All the AudioSet files are not readily available for download and need to be sliced manually from YouTube videos. Moreover, with time, a lot of these videos have been taken down by YouTube. Thus, results reported on pre-training on complete 2 million utterances of the AudioSet might not be reproducible by independent researchers.
- As pointed out by (Hannun 2021), self-supervised learning in speech and audio would be more accessible to smaller labs given lighter-weight models which could be trained efficiently on a lesser amount of data. Thus, we wanted to compare the performance of our system on a smaller pre-training dataset setting, contrary to the prior art in this domain.

For the downstream tasks, we test the efficiency of our learned embeddings on a diverse set of audio tasks. We take a mixture of both speech and non-speech tasks to increase variability and test generalizability. For speaker identification we resort to two commonly used datasets, namely, LibriSpeech (LBS) (Panayotov et al. 2015) and Voxceleb (Nagrani, Chung, and Zisserman 2017). For the task of keyword spotting, we use the Google Speech Commands V1 (SC) and V2 datasets (Warden 2018). This dataset is used in various label settings in prior art. We use both the 12 and 35 label settings for our experiments. For speech classification, we also apply speech emotion classification to the IEMOCAP data set (Busso et al. 2008) and language identification on the Voxforge (Voxforge.org) data set. For classifying acoustic signals beyond speech, we experiment with three common data sets. First, we do bird song detection (Stowell et al. 2019) to solve a binary classification task with an objective to detect if the sound segment has a bird song in it. Finally, for music classification, we use the NSynth data set (Engel et al. 2017) of musical notes from different instruments.

More details about the data sets can be found in Table 2.

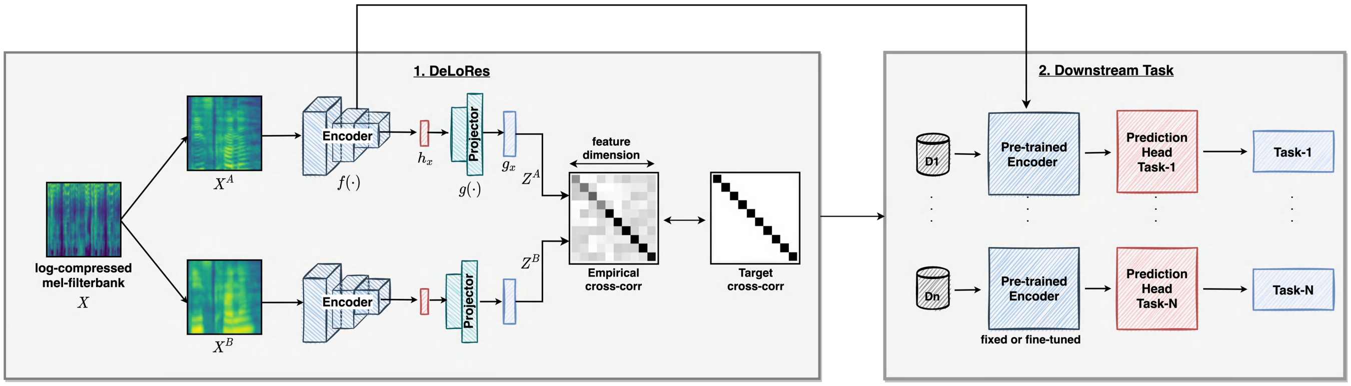

Figure 1: The block diagram of DeLoRes in pre-training and fine-tuning phases

Implementation Details

Augmentations

As pointed out by (Zbontar et al. 2021), the Barlow Twins objective function is sensitive to the data augmentations used between the pair of samples for which the cross-correlation matrix is to be calculated and hence optimized. In this section, we will describe the different augmentations K that is applied to both the twin batches before it is fed to our encoder network. We borrow ideas from (Niizumi et al. 2021) and implement a normalization step, followed by a mixup and random resized crop block each, which we will explain in detail in the following section.

Normalization We normalize the mel-spectrogram, x, using:

$$\tilde{x} = \frac{x - \mu}{\nu}, \tag{5}$$

where μ and ν are the mean and standard deviation of the training samples, respectively.

Mixup Mixup as an augmentation scheme has shown great success in supervised settings in CV (Zhang et al. 2018), were the first ones to have used it for an audio pretraining task in a SSL setting. They show significant performance boosts when they use it compared to when they do not (an average degradation of 8.4% for 5 downstream tasks under the linear evaluation protocol).

The basic functionality of mixup is to *mix* randomly selected input audio (x_k) in a small ratio from prior batches, which acts as background sound in the final *mixed audio* ($\tilde{x}_i$). Formally, this would help the model learn embeddings invariant to noise.

We use mixup in a similar setting to (Niizumi et al. 2021) where we use audio features only (instead of audio and labels both which was originally proposd by (Zhang et al. 2018)), because of the lack of labels in a SSL setting. Since audio is log-scaled before passing it through this augmentation block, the input is converted to linear scale before mixup and converted back to log-scale again. A FIFO queue of size 2048 is maintained for storing past inputs, from where spectrograms are randomly sampled for mixup.

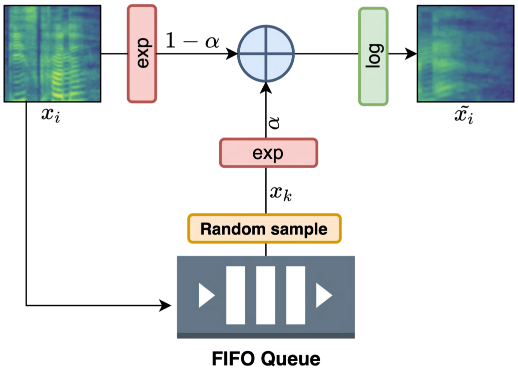

Figure 2: Mixup

Random Resized Crop Random resized crop (RRC) is one of the most prevalent image augmentation functions used in both supervised and semi-supervised settings in CV. Unlike mixup, RRC does not need labels. The primary motive of RRC is to crop a random portion of image and resize it to a given size. Most SSL algorithms use it (Chen et al. 2020; Caron et al. 2021; Zbontar et al. 2021) and especially (Zbontar et al. 2021) show a significant drop in performance when they do not use it. In this paper, we use RRC in a fashion which tries to do a task exactly similar to the original RRC used in CV. For adapting it to the audio domain, we draw inspiration from (Niizumi et al. 2021).

The input spectrogram consists of frequency bins F, and T time frames. First, we sample the random crop area from a virtual crop boundary with time frames longer than the input, i.e., 1.5 × T. The size of the crop area is randomly sampled as:

$$F_c = \lfloor \min\left(U\left(h_1, h_2\right), 1.0\right) \times F \rfloor \tag{6}$$

$$T_c = \lfloor U\left(w_1, w_2\right) \times T \rfloor \tag{7}$$

where F_c and T_c are the number of frequency bins and number of time frames of random crop size, respectively, h_1 and h_2 form a frequency bin range [h_1, h_2], w_1 and w_2

Data Set	Target	No. of Classes	No. of Samples	Avg. Duration (sec)
LibriSpeech (LBS)	Speaker Identification	585	28,538	12.69
VoxCeleb 1 (VC)	Speaker Identification	1,211	153,397	8.20
IEMOCAP (IC)	Emotion Recognition	4	4,490	4.49
Speech Commands V1 (SC-V1)	Keyword Recognition	12	64,721	0.98
Speech Commands V2 (SC-V2)	Keyword Recognition	12/35	105,829	0.98
Bird Song Detection (BSD)	Song detection	2	15,690	10.08
VoxForge (VF)	Language Identification	6	176,438	6.68
NSynth (NS)	Musical Instruments Identification	11	301,883	4.00

Table 2: Dataset statistics for downstream benchmark tasks. The settings have been inspired from and is in-lines with prior-art (Saeed, Grangier, and Zeghidour 2020; Niizumi et al. 2021)

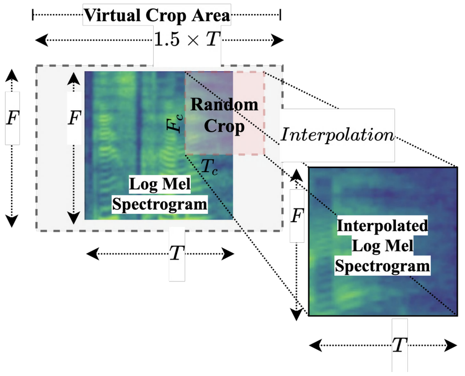

Figure 3: Random Resized Crop

form a time frame range $[w_1, w_2]$ and U stands for the range function. We allow the dimension on the time axis to exceed the original dimension of the spectrogram through padding, but not on the frequency domain. Contents in the crop are then resized to the size of the original input with bicubic interpolation.

Experimental Setup

Given an audio input sequence, we extract log-compressed mel-filterbanks with a window size of 64 ms, a hop size of 10 ms, and N = 64 mel-spaced frequency bins in the range 60 – 7800 Hz. As mentioned earlier, for pre-training we randomly crop a small segment of the original audio from the AudioSet (originally 10 seconds each). This is in line with previous work in this domain (Saeed, Grangier, and Zeghidour 2020; Niizumi et al. 2021). For our pre-training setup, we use T = 96 frames, which corresponds to 1024 seconds. For downstream tasks, the number of frames T depends on the average duration of each downstream data set (for example 12.69 s for LibriSpeech and 4.49 s for IEMOCAP).

For our ConvNet encoder we choose an embedding size $h_x \in R^{2048}$. During pre-training, we pass h_x through a set of linear layers g, also called the projection head, which contains two fully connected layers with 8192 units each and produces $g_x \in R^{8192}$, followed by a BatchNormalization layer, to make the embeddings zero mean and unit variance. For training and inference on downstream tasks, we discard g after our pre-training step and replace it with a single linear layer with units equal to the number of classes in the task.

Model	Params	Data	Epochs (SSL/Down.)
TRILL	27M	1M+	–
COLA	11M	2M	500/-
BYOL-A	5.3M	0.25M	100/200
DECAR	11M	0.2M	100/50
SSAST	87M	2M	8.5/50
DeLoRes	**5.3M**	**0.25M**	100/100

Table 3: Comparison of DeLoRes with prior-art on general-purpose audio representation learning. DeLoRes is trained on a low-resource setting in terms of both data and compute compared to prior-art.

We pre-train our model using SSL for a total of 100 epochs. We learn the parameters of our model with the optimization scheme proposed in (Zbontar et al. 2021; Grill et al. 2020). We use the LARS optimizer (You, Gitman, and Ginsburg 2017) with a batch size of 1024. (Zbontar et al. 2021) however show that Barlow Twins loss function performs well with smaller batches too.

As used in (Zbontar et al. 2021), we set the learning rate to 0.2 for the weights and 0.0048 for the biases. The learning rate is multiplied by the batch size and divided by 256. We use a learning rate warm-up period of 10 epochs, after post which we reduce the learning rate by a factor of 1000 and use a cosine decay schedule (Loshchilov and Hutter 2017) for the same. Finally, the trade-off parameter λ of the loss function was set to 0.0051.

We train all our downstream tasks with Adam optimizer, a learning rate of 10^{-3}, in batched-mode with a batch size of 64, and train it for a maximum of 100 epochs.

In this work, we have borrowed most of the hyper-parameter settings from the original implementation (Zbontar et al. 2021).

Results

Linear Evaluation Protocol

We compare the effectiveness of DeLoRes embeddings over 9 challenging and diverse downstream tasks. Table 4 reports a comparison with other SOTA methodologies on the *linear evaluation protocol* where only the linear layer is trained

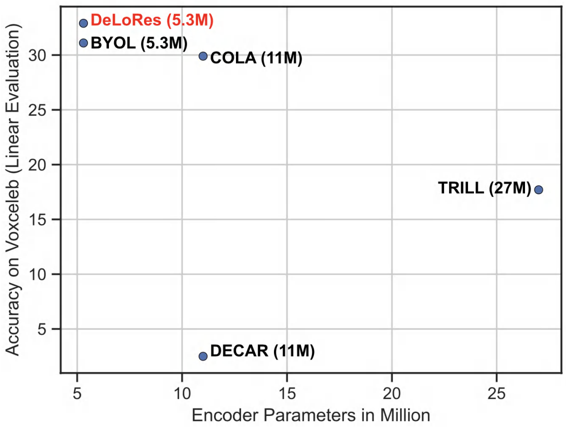

Figure 4: Number of parameters vs. performance for Voxceleb on the linear evaluation protocol setup

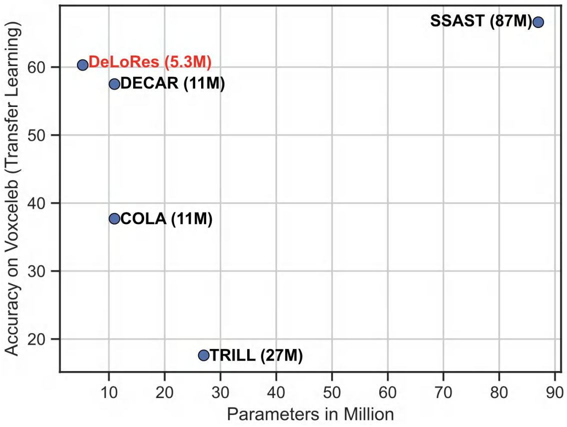

Figure 5: Number of parameters vs. performance for Voxceleb on the transfer learning setup

for the particular downstream over DeLoRes embeddings [2]. Even with our smaller architecture and lesser pre-training data, we are competitive to other prior-work in this domain, except BYOL-A which serves as the current SOTA for the *linear evaluation protocol* for generalizing across 5 tasks as can be seen in Table 4. BYOL-A was pre-trained using SSL on 10 times more data than our setting, with double the number of parameters to be learned (because of the presence of a teacher network) and was pre-trained for 400 epochs which is 4 times more than our implementation.

Transfer Learning Setup

Table 5 reports a comparison with other SOTA methodologies on the *transfer learning* setup where all the model weights are fine-tuned end-to-end on the downstream task. Though this would not a be fair comparison because a bigger model architecture might learn task-specific features better than ours, similar to the linear evaluation protocol, we show through our experiments that DeLoRes overpowers other approaches in the general-purpose audio representation learning space in 5 out of 9 tasks while losing marginally in the other 4 tasks.

Low-Resource Setting

Table 3 shows the comparison of other SSL pre-training tasks with our proposed method, with respect to model size, training time (number of epochs) and total amount of data used for SSL (number of AudioSet samples). We claim our training to be on *low-resource* for three main reasons:

- We use only a fraction of the total pre-training data compared to prior work. SSL has been known to benefit from large amounts of data.
- Our architecture uses the least number of encoder parameters for SSL with unlabeled audio. BYOL-A, which uses the same network as ours, uses a separate teacher network which double the number of parameters needed for training. Fig. 4 and 5 give a pictorial representation for the same for one particular downstream task. DeLoRes is competitve to prior art with a fraction of the total trainable parameters present in the model.
- We pre-train for much lesser number of epochs when compared to prior-art. Our model pre-training converges by 100 epochs.

Analysis of Results

As seen in Table 6, we show significant gains compared to using DeLoRes embeddings as compared to randomly initialized embeddings. This makes it very evident that we learn powerful representations from our pre-training step. The choice of tasks for Table 6 is made such that all tasks for comparison have fewer training samples than our total pre-training data.

Additionally, we see in Fig. 6, that our model starts at a much better accuracy on the first epoch, and jumps higher afterward in later epochs. With SSL based pre-training the model trains faster and better.

Conclusion

In this paper, we propose a new approach to learning general-purpose audio representations via self-supervised learning using an invariance and redundancy-reduction framework. DeLoRes is simple by nature and avoids trivial solution by its nature of reconstruction. We also show that it is possible to learn good representations under a low-resource setting (both data and compute). DeLoRes is competitive with linear evaluation and does extremely well when fine-tuned end-to-end. Future work involves pre-training DeLoRes for more epochs to see it improves performance and analyze the effect of changing different hyperparameters and also the sensitivity of our approach to the different augmentations applied. In a future version of the paper we propose the LAPE Benchmark for a holistic evaluation of learned audio representations with 11 diverse downstream tasks, and DeLoRes-M, which gives state-of-the-art results on 9 out of the 11 tasks in LAPE.

[2] We only compare with prior-art which focuses on leaning *general-purpose audio representations* and not all speech and audio SSL algorithms

Downstream Task	CBoW	SG	TemporalGap	Triplet Loss	TRILL	COLA	BYOL-A	DECAR	DeLoRes
Speech Commands V1	–	–	–	–	74.0	71.7	–	63.9	**86.1**
Speech Commands V2 (12)	–	–	–	–	74.0	–	84.5	65.7	**85.4**
Speech commands V2 (35)	30.0	28.0	23.0	18.0	–	62.4	**87.2**	–	80.0
LibriSpeech	99.0	**100.0**	97.0	**100.0**	–	**100.0**	–	62.5	90.1
VoxCeleb	–	–	–	–	17.7	29.9	31.0	2.5	**31.2**
NSynth	33.5	34.4	35.1	25.7	–	63.4	**71.2**	59.9	66.3
VoxForge	–	–	–	–	**88.1**	71.3	83.1	46.0	76.5
IEMOCAP	–	–	–	–	–	–	–	60.5	**60.7**
Birdsong Detection	71.0	69.0	71.0	73.0	–	77.0	–	76.4	**86.7**

Table 4: Result comparison for *linear evaluation protocol* setup. Results for approaches other than DeLoRes has been taken from literature. "–" signifies that results were not reported for these tasks by these methods.

Downstream Task	TRILL	COLA	DECAR	Wav2Vec	SSAST	DeLoRes
Speech Commands V1	–	**98.1**	97.6	96.2	96.2	97.7
Speech Commands V2 (12)	91.2	–	97.6	–	–	**97.8**
Speech commands V2 (35)	–	95.5	–	–	**98.2**	95.9
LibriSpeech	–	**100.0**	97.0	–	–	95.3
VoxCeleb	17.6	37.7	57.5	56.6	**66.6**	60.3
NSynth	–	73.0	78.4	–	–	**78.6**
VoxForge	94.1	82.9	76.5	–	–	**95.6**
IEMOCAP	–	–	**66.9**	57.1	59.8	63.9
Birdsong Detection	–	80.2	**90.3**	–	–	**90.3**

Table 5: Result comparison for *transfer learning* setup. Results for approaches other than DeLoRes has been taken from literature. "–" signifies that results were not reported for these tasks by these methods.

Downstream Task	Random Init		DeLoRes Init	
	freeze	finetune	freeze	finetune
SC-V1	52.9	97.3	86.1	97.7
SC-V2(12)	53.3	97.6	85.4	97.8
SC-V2(35)	54.7	95.8	80.0	95.8
LBS	57.7	91.1	90.1	95.3
VC	6.0	55.2	31.2	60.3
NS	43.2	77.7	66.3	78.6
VF	56.7	94.4	76.5	95.6
IC	53.3	60.3	60.7	63.9
BSD	60.3	85.1	86.7	90.3
Avg	49.5	80.7	72.1	83.0

Table 6: Randomly initialized vs DeLoRes initialized performance comparison

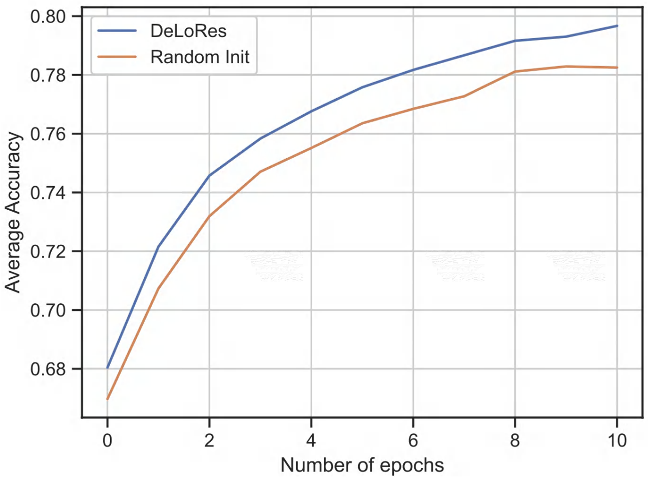

Figure 6: Average score over 9 downstream tasks Vs Number of Epochs (11 epochs) in *transfer learning setup*

References

Arandjelović, R.; and Zisserman, A. 2017. Look, Listen and Learn. arXiv:1705.08168.

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2020. Self-labelling via simultaneous clustering and representation learning. arXiv:1911.05371.

Baevski, A.; Zhou, H.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. arXiv:2006.11477.

Busso, C.; Bulut, M.; Lee, C.-C.; Kazemzadeh, A.; Mower, E.; Kim, S.; Chang, J. N.; Lee, S.; and Narayanan, S. S. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *LREC*, 42(4): 335–359.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2019. Deep Clustering for Unsupervised Learning of Visual Features. arXiv:1807.05520.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2021. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments.

arXiv:2006.09882.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709.

Chen, X.; and He, K. 2020. Exploring Simple Siamese Representation Learning. arXiv:2011.10566.

Chung, Y.-A.; and Glass, J. 2020. Generative Pre-Training for Speech with Autoregressive Predictive Coding. arXiv:1910.12607.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Engel, J.; Resnick, C.; Roberts, A.; Dieleman, S.; Norouzi, M.; Eck, D.; and Simonyan, K. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, 1068–1077. PMLR.

Fonseca, E.; Favory, X.; Pons, J.; Font, F.; and Serra, X. 2020. FSD50K: an Open Dataset of Human-Labeled Sound Events. arXiv:2010.00475.

Freitag, M.; Amiriparian, S.; Pugachevskiy, S.; Cummins, N.; and Schuller, B. 2017. auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks. arXiv:1712.04382.

Gemmeke, J. F.; Ellis, D. P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 776–780. IEEE.

Ghosh, S.; Katta, S. V.; Seth, A.; and Umesh, S. 2021. Deep Clustering For General-Purpose Audio Representations. arXiv:2110.08895.

Gong, Y.; Chung, Y.-A.; and Glass, J. 2021. AST: Audio Spectrogram Transformer. arXiv:2104.01778.

Gong, Y.; Lai, C.-I. J.; Chung, Y.-A.; and Glass, J. 2021. SSAST: Self-Supervised Audio Spectrogram Transformer. arXiv:2110.09784.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap your own latent: A new approach to self-supervised Learning. arXiv:2006.07733.

Hannun, A. 2021. The History of Speech Recognition to the Year 2030. arXiv:2108.00084.

Hsu, W.-N.; Bolte, B.; Tsai, Y.-H. H.; Lakhotia, K.; Salakhutdinov, R.; and Mohamed, A. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. arXiv:2106.07447.

Jansen, A.; Plakal, M.; Pandya, R.; Ellis, D. P. W.; Hershey, S.; Liu, J.; Moore, R. C.; and Saurous, R. A. 2017. Unsupervised Learning of Semantic Audio Representations. arXiv:1711.02209.

Koizumi, Y.; Takeuchi, D.; Ohishi, Y.; Harada, N.; and Kashino, K. 2020. The NTT DCASE2020 Challenge Task 6 system: Automated Audio Captioning with Keywords and Sentence Length Estimation. arXiv:2007.00225.

Latif, S.; Rana, R.; Qadir, J.; and Epps, J. 2020. Variational Autoencoders for Learning Latent Representations of Speech Emotion: A Preliminary Study. arXiv:1712.08708.

Ling, S.; and Liu, Y. 2020. DeCoAR 2.0: Deep Contextualized Acoustic Representations with Vector Quantization. arXiv:2012.06659.

Liu, A. T.; Li, S.-W.; and Lee, H.-y. 2021. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 2351–2366.

Liu, A. T.; Yang, S.-w.; Chi, P.-H.; Hsu, P.-c.; and Lee, H.-y. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6419–6423. IEEE.

Loshchilov, I.; and Hutter, F. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983.

Nagrani, A.; Chung, J. S.; and Zisserman, A. 2017. Vox-Celeb: A Large-Scale Speaker Identification Dataset. *Interspeech 2017.*

Niizumi, D.; Takeuchi, D.; Ohishi, Y.; Harada, N.; and Kashino, K. 2021. BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation. arXiv:2103.06695.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210.

Saeed, A.; Grangier, D.; and Zeghidour, N. 2020. Contrastive Learning of General-Purpose Audio Representations. arXiv:2010.10915.

Shor, J.; Jansen, A.; Maor, R.; Lang, O.; Tuval, O.; Quitry, F. d. C.; Tagliasacchi, M.; Shavitt, I.; Emanuel, D.; and Haviv, Y. 2020a. Towards Learning a Universal Non-Semantic Representation of Speech. *Interspeech 2020.*

Shor, J.; Jansen, A.; Maor, R.; Lang, O.; Tuval, O.; Quitry, F. d. C.; Tagliasacchi, M.; Shavitt, I.; Emanuel, D.; and Haviv, Y. 2020b. Towards Learning a Universal Non-Semantic Representation of Speech. *Interspeech 2020.*

Stowell, D.; Wood, M. D.; Pamuła, H.; Stylianou, Y.; and Glotin, H. 2019. Automatic acoustic detection of birds through deep learning: the first Bird Audio Detection challenge. *Methods in Ecology and Evolution*, 10(3): 368–380.

Takeuchi, D.; Koizumi, Y.; Ohishi, Y.; Harada, N.; and Kashino, K. 2020. Effects of Word-frequency based Pre- and Post- Processings for Audio Captioning. arXiv:2009.11436.

Tsai, Y.-H. H.; Bai, S.; Morency, L.-P.; and Salakhutdinov, R. 2021. A Note on Connecting Barlow Twins with Negative-Sample-Free Contrastive Learning. arXiv:2104.13712.

van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748.

Voxforge.org. ???? Free Speech... Recognition (Linux, Windows and Mac) - voxforge.org. http://www.voxforge.org/. Accessed 06/25/2014.

Warden, P. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. arXiv:1804.03209.

You, Y.; Gitman, I.; and Ginsburg, B. 2017. Large Batch Training of Convolutional Networks. arXiv:1708.03888.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. arXiv:2103.03230.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412.

Acknowledgments

We thank Centre For Development Of Advanced Computing (CDAC), as part of the National Language Translation Mission for providing us with the compute resources required for the experiments. Additionally we thank all other members of Speech Lab IITM and Verisk Analytics for their valuable feedback which helped us correct errors in our paper.