



Variational Information Bottleneck for Effective Low-resource Audio Classification

Shijing Si¹, Jianzong Wang^{1*}, Huiming Sun^{1,2}, Jianhan Wu³, Chuanyao Zhang³, Xiaoyang Qu¹, Ning Cheng¹, Lei Chen² and Jing Xiao¹

¹Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China

²Hong Kong University of Science and Technology, Hong Kong, China

³University of Science and Technology of China, Hefei, China

jzwang@188.com

Abstract

Large-scale deep neural networks (DNNs) such as convolutional neural networks (CNNs) have achieved impressive performance in audio classification for their powerful capacity and strong generalization ability. However, when training a DNN model on low-resource tasks, it is usually prone to overfitting the small data and learning too much redundant information. To address this issue, we propose to use variational information bottleneck (VIB) to mitigate overfitting and suppress irrelevant information. In this work, we conduct experiments on a 4-layer CNN. However, the VIB framework is ready-to-use and could be easily utilized with many other state-of-the-art network architectures. Evaluation on a few audio datasets shows that our approach significantly outperforms baseline methods, yielding $\geq 5.0\%$ improvement in terms of classification accuracy in some low-source settings.

Index Terms: audio classification, variational information bottleneck, overfitting, low resource data

1. Introduction

Deep learning [1, 2, 3] has emerged as the de facto standard technique in all areas of artificial intelligence, including speech processing [4, 5], computer vision (CV) [6, 7] and natural language processing (NLP) [8, 9]. Currently DNNs have been very successful in the audio processing domain [10, 11] for their strong capacity. The DNN models can produce good-quality representation features for audio processing tasks, for example, audio classification [12], automatic speech recognition [13] and speaker verification [14]. However, modern DNN-based audio classifiers typically require large amount of labeled data for training or fine-tuning [15], which might pose a challenge for many real applications. Because audio data labeling is both time-consuming and tedious, in many real situations there are only a limited number of training examples available [16].

Applying DNN classifiers to low-resource datasets often leads to overfitting because DNNs have too much capacity and extract too many features of the low-resourced data that are irrelevant to the target labels [17]. Variational information bottleneck (VIB) addresses the overfitting problem by eliminating irrelevant information and only retaining target related information [18]. Before elaborating VIB, we describe some basics on information bottleneck (IB).

IB was proposed by [19] to explain and enhance the generalization ability of neural networks. The main idea is: for an input data X and its corresponding output (label) Y , we aim to

learn a low-dimensional representation Z that is maximally informative about our target Y with minimal redundant information. IB method maximizes the mutual information $I(Z, Y)$ between Y and Z , and minimizes the mutual information $I(Z, X)$ between X and Z , so as to reserve the most useful data and discard redundant information.

The IB principle is appealing, since it defines what we mean by a good representation, in terms of the fundamental tradeoff between having a concise representation and good predictive power [20]. The main drawback of the IB principle is that mutual information is, in general, computationally challenging. To address the computing issue, [21, 18] proposed a variational inference approach, *i.e.*, the VIB. It has been used in various areas of deep learning research. For example, [22] proposed VIB-ERT that can suppress irrelevant features and yield a concise representation for text classification tasks in NLP. [23] utilized the VIB principle to prune individual neurons for model compression. Though VIB has been shown great promise in low-resource text classification, few research have investigated its use in audio classification. In this paper, we propose to implement the VIB method to address the overfitting problem in low-resourced audio classification. To illustrate how the VIB framework works, we take convolutional neural networks (CNNs) as our feature extractor and conduct extensive experiments to verify its effectiveness. Actually, the VIB framework can also be used with other state-of-the-art network architectures, like Transformer encoder [24], etc.

The main contributions are summarized as follows:

- We explore the VIB approach to address the overfitting in audio classification
- We conduct extensive experiments to verify the effectiveness of VIB in comparison with other baselines

2. Methodology

The objective of IB is to learn a maximally compressed representation Z of the input X that maximally preserves information about the output Y . Its mathematical formula is shown in Eq. (1),

$$\mathcal{L}_{IB} = \underbrace{\beta I(X, Z)}_{\text{Compression}} - \underbrace{I(Z, Y)}_{\text{Predictive}} \quad (1)$$

where $I(X, Z)$ and $I(Z, Y)$ are used for compression and predictive purposes, respectively, and $\beta \geq 0$ controls the balance between compression and prediction.

* corresponding author

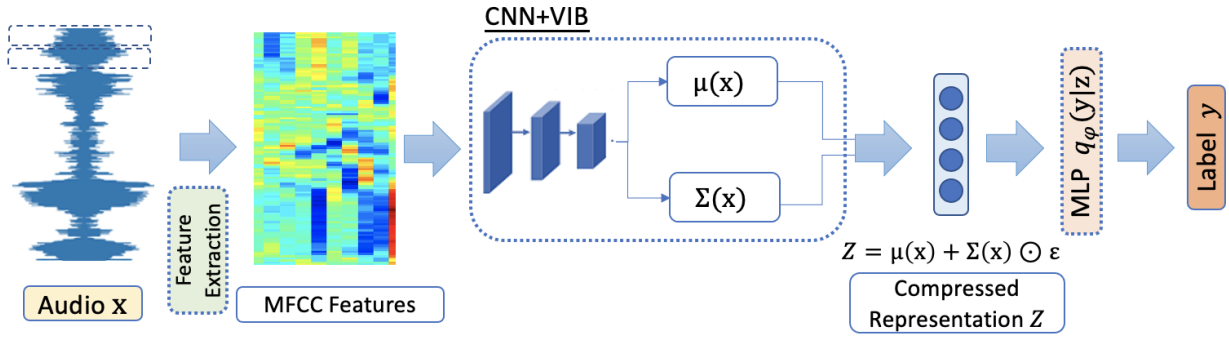


Figure 1: Schematic diagram of CNN+VIB framework.

2.1. Variational Information Bottleneck

Directly optimizing \mathcal{L}_{IB} is hard, because it is usually computationally demanding. Then [21, 18] developed a variational approximate estimate of IB (VIB) in light of

$$\begin{aligned} & \beta I(X, Z) - I(Z, Y) \\ & \leq \beta \int dx dz p(x) p_\theta(z|x) \log \frac{p_\theta(z|x)}{r(z)} \\ & \quad - \int dx dy dz p(x) p(y|x) p_\theta(z|x) \log q_\phi(y|z), \end{aligned} \quad (2)$$

where $q_\phi(y|z)$ is a variational distribution to approximate $p(y|z)$, and $r(z)$ is a specified prior distribution for latent code Z (usually taking standard normal distribution), and $p_\theta(z|x)$ is an estimate of the posterior probability of z . Alternatively, $p_\theta(z|x)$ and $q_\phi(y|z)$ can also be interpreted as encoder and decoder, respectively as in the variational auto-encoder [25]. For each training example (x, y) , the IB loss is upper bounded by

$$\mathcal{L}_{VIB} = \beta \mathbf{E}_x [KL(p_\theta(z|x), r(z))] + \mathbf{E}_{z \sim p_\theta(z|x)} [-\log q_\phi(y|z)]. \quad (3)$$

Therefore, in this work Eq. (3) is applied to minimize the IB objective.

2.2. Deep VIB Audio Classifier

Here we construct an audio classifier by incorporating a CNN encoder inside the VIB. The schematic diagram of our framework is shown in Fig. 1. Specifically, during training, for each audio data x , we extract its Mel-Frequency Cepstral Coefficients (MFCC), and pass them through a CNN encoder, yielding the mean and standard deviation vectors, *i.e.*, $\mu(x)$ and $\Sigma(x)$, of the posterior distribution $p_\theta(z|x)$. Then we simulate Gaussian samples of z and feed them a multi-layered perceptron (MLP) $q_\phi(y|z)$ to predict the logits of label y . During inference, $z = \mu(x)$ is used instead of sampling from $p_\theta(z|x)$.

For each training example, z is the latent bottleneck and its dimension K controls the information relevant to the label y . If the dimension K is large, the compression effect is decreased; otherwise the bottleneck compressed the information used for prediction. The hyper-parameter β in Eq. (3) also controls the compression effect of the model. If β is small, the compression effect is weak and the model tends to overfit the data. If β is large, the model is compressed heavily and little information is used for prediction. In Fig. 1, CNN architecture could be

replaced by other networks, like Transformer [24], etc. In this work, we conduct experiments on a CNN architecture, because it is commonly used in audio classification.

3. Experiment Setup

Here we conduct audio classification experiments over multiple datasets to empirically verify the effectiveness of our method, in comparison with existing approaches. We use Tensorflow and Keras to implement our VIB+CNN classifier and Librosa [26] for audio processing and MFCC feature extraction. We utilize the loss, accuracy and F1 score to evaluate the performance of different approaches. We then present some ablation studies and analyses to investigate the source of the improvements using our VIB method. The Python code for experiments can be found at https://github.com/shijing001/VIB_audio_classification.

3.1. Datasets

We conduct extensive experiments on four datasets: Audio-MNIST, ESC-50, Toronto Emotional Speech Set (TESS), and TUT Acoustic Scenes (TUT). Details on these 4 datasets are presented as follows. Basic statistics, *i.e.*, data size, number of classes (C), train/validation/test size and the number of samples per class, are shown in Table 1.

Audio MNIST consists of 3000 audio recordings of spoken digits (0-9) in English with 50 repetitions per digit for each of the 60 different speakers. Recordings were at a sampling frequency of 48kHz and were saved in 16 bit integer format.

ESC-50 is a collection of short environmental recordings available in a unified format (5-second-long clips, 44.1 kHz, single channel, Ogg Vorbis compressed @ 192 kbit/s). It consists of a labeled set of 2000 environmental recordings (50 classes, 40 clips per class).

Toronto Emotional Speech Set (TESS)[27] A set of 200 target words were spoken in the carrier phrase ‘‘Say the word xxxx’’ by two actresses and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 recordings in total (7 classes, 400 clips per class).

TUT Acoustic Scenes 2017 consists of 4680 recordings of 15 classes. Each audio is about 10 seconds long. More details on this data can be found in [28].

Table 1: Basic statistics of four datasets with C and Num/C representing the number of labels and average number of examples for each class.

Dataset	Size	C	Train/Valid./Test	Num/C
Audio-MNIST	3000	10	1800/600/600	300
ESC-50	2000	50	1600/200/200	40
TESS	2800	7	1680/560/560	400
TUT	4680	15	3808/936/936	312

Table 2: The architecture of the baseline CNN classifier.

Layer	Outputs	Kernel	Stride
Conv2D+ReLU	32	4×4	1
Conv2D+ReLU	96	4×10	1
Conv2D+ReLU	96	4×10	1
Conv2D+ReLU	160	4×10	1
MaxPooling	160	2×2	-
Dense	C	-	-

3.2. Deep Learning models

We employ a 4-layered CNN architecture as the backbone in our experiments. It consists of 4 convolutional layers with 32, 96, 96 and 160 output channels, respectively, followed by a maximum pooling layer and a fully connected (dense) layer with C logits as output. Its architecture is displayed in Table 2. The baseline methods to address overfitting are listed as follows.

Weight Decay is a common regularization technique to improve generalization [29]. It regularizes the large weights w by adding a penalization term $\frac{\lambda}{2} \|w\|^2$ to the loss, where λ is a hyperparameter specifying the strength of regularization. λ is set to $1.0e - 4$ in the experiments.

Dropout [30], is a widely used stochastic regularization techniques used in deep learning models [31, 32] to mitigate overfitting. We implement the spatial dropout for all 4 CNN layers with dropout probability 0.2.

CNN+VIB (Ours) classifier is built on top of the baseline CNN model, and its architecture is the same as Table 2 but with an additional dense layer after maxpooling, which yields the $\mu(x)$ and $\Sigma(x)$ for $p_{\theta}(z|x)$. For the hyper-parameters K and β , taking values in the ranges [20, 50, 100, 200] and $[2.0e - 3, 5.0e - 3, 2.0e - 2, 5.0e - 2]$, respectively. We first perform model selection on the validation set to find the hyper-parameters and then evaluate the selected models on the test set.

3.3. Implementation Details

Preprocessing All the raw audios are resampled to 44.1kHz and then fixed to the certain length by zero-padding or truncating (*i.e.* 4s for the Audio-MNIST, 5s for the ESC-50, 2s for the TESS and 10s for the TUT). The short time Fourier transform (STFT) is then applied on the audio signals to calculate spectrograms, with a window size of 40ms and a hop size of 20ms. 40 mel filter banks are applied on the spectrograms followed by a logarithmic operation to extract the MFCC features.

Training details In the training phase, the Adam algorithm [33] is employed as the optimizer with the default parameters. The model is trained end-to-end with the initial learning rate of 0.001 and the exponential decay rate of 0.98 for each epoch. Batch size is set to 8 and training epoch is set to 20 for TUT and 40 for other datasets.

4. Results and Analysis

4.1. Overfitting Suppression

Here we exhibit that VIB can suppress overfitting during model training. Fig. 2 summarizes the loss and accuracy on training and validation sets versus the number of training epochs. Fig. 2a and 2b present the performance on the ESC-50 and TUT datasets, respectively. In Fig. 2a, the left plot illustrates the training and validation losses of CNN and CNN+VIB versus training epochs. The blue dotted and solid lines represent the training and validation losses of the baseline CNN model, while the red dotted and solid lines indicate training and validation losses for our CNN+VIB approach. As the training epoch increases, the baseline CNN training loss (the blue dotted line) decreases quickly and then stabilizes after around 15 epochs. However, the baseline validation loss (the blue solid line) falls rapidly in the beginning, reaches its minimum at around 10 epochs and then increases speedily for the rest of training epochs. This is a clear sign that the baseline CNN model has overfitted the ESC-50 dataset. By contrast, the validation loss from CNN+VIB (red solid line) exhibit a gradually decreasing trend as the increase of training epochs. This is the empirical evidence of VIB reducing overfitting, which also appears in other datasets like the TUT in Fig. 2b.

The central-left plot of Fig. 2 displays the classification accuracy on training and validation sets of ESC-50. The blue dotted and solid lines represent the training and validation accuracy of the baseline CNN model, while the red dotted and solid lines indicate training and validation accuracy for our CNN+VIB approach. As the training epoch increases, the baseline CNN training accuracy (the blue dotted line) increases quickly and then stabilizes after around 15 epochs. However, the baseline validation accuracy (the blue solid line) grows rapidly in the beginning, reaches its maximum at around 15 epochs and then decreases slightly later. Fig. 2b shows the loss and accuracy of CNN and CNN+VIB on the TUT dataset, which also exhibits the similar pattern.

Table 3 displays the performance (in terms of accuracy and F1 score) of four methods on audio-MNIST, ESC-50, TESS and TUT datasets under multiple low-resource settings. We create low-resource settings by subsampling the original training data with a certain percentage. The main findings of this table are i.) CNN+VIB can always outperform baseline methods in all settings. ii.) in extremely low-resource settings like only using 5% training data, CNN+VIB outperforms the baseline methods by a significant margin and the margin decreases as the training data size increases.

4.2. Ablation Study

To analyze the effect of VIB on reducing overfitting, we investigate the effect of the β on training and validation cross-entropy (CE) losses since β controls the trade-off between removing information from the audio features (high β) and keeping information that is predictive of the target y (low β). In Fig. 3, we fix the bottleneck size ($K = 20$) and train CNN+VIB on two datasets (ESC-50 and TUT) for varying values of β and plot the validation (red solid) and training (blue dotted) CE losses. Fig. 3a displays a typical pattern of how training and validation losses evolve with the increase of β . For small values of β (left side of Fig. 3a), where VIB has little effect, the validation loss is substantially higher than the training loss, which indicates overfitting. This is because the network learns to be more deterministic ($\Sigma \approx 0$), thereby retaining too much irrele-

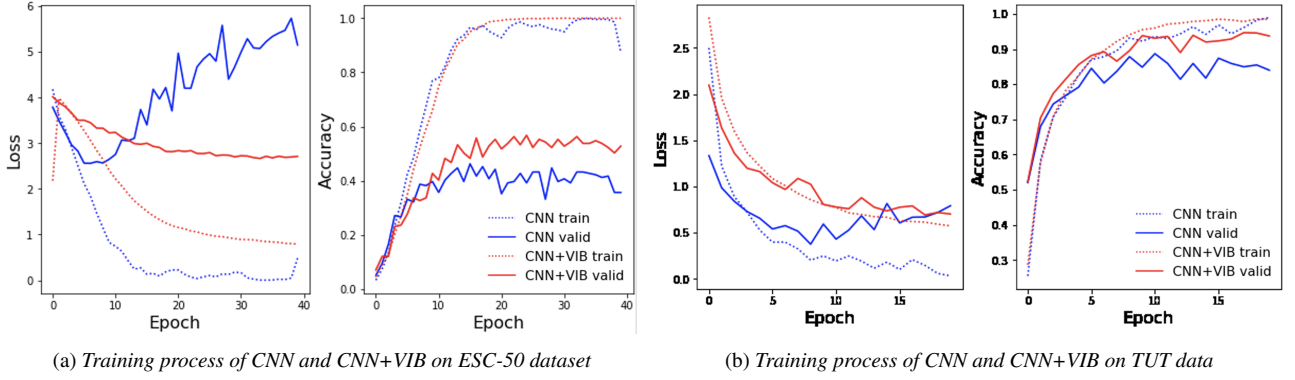


Figure 2: The loss and accuracy of CNN+VIB (Red lines) versus CNN (Blue lines) on two datasets: ESC-50 and TUT. The dashed lines present the loss and accuracy on training sets, whereas the solid lines exhibit loss and accuracy on validation sets.

Table 3: The performance (accuracy and F1 score) of four methods on audio-MNIST, ESC-50, TESS and TUT datasets under multiple low-resource settings that are characterized by varying the percentage of training data. Bold values indicate the best ones. The hyper-parameters are tuned by grid search.

Data	Model	5%		10%		30%		50%		100%	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
audio-MNIST	CNN	0.543	0.541	0.842	0.835	0.953	0.951	0.991	0.990	0.991	0.993
	+Dropout	0.572	0.572	0.824	0.821	0.952	0.951	0.991	0.991	0.990	0.990
	+Weight Decay	0.578	0.573	0.844	0.833	0.954	0.949	0.990	0.992	0.993	0.991
	CNN+VIB	0.587	0.584	0.868	0.869	0.975	0.975	0.997	0.997	1.000	1.000
ESC-50	CNN	0.159	0.155	0.192	0.190	0.303	0.303	0.396	0.392	0.552	0.507
	+Dropout	0.162	0.161	0.213	0.207	0.295	0.287	0.502	0.436	0.551	0.503
	+Weight Decay	0.178	0.176	0.210	0.198	0.358	0.357	0.421	0.420	0.568	0.563
	CNN+VIB	0.224	0.184	0.239	0.210	0.363	0.336	0.532	0.528	0.581	0.589
TESS	CNN	0.936	0.934	0.952	0.952	0.921	0.916	0.998	0.998	0.997	0.997
	+Dropout	0.938	0.937	0.967	0.964	0.991	0.991	0.993	0.993	0.997	0.997
	+Weight Decay	0.927	0.920	0.946	0.947	0.925	0.920	0.993	0.992	0.998	0.998
	CNN+VIB	0.954	0.957	0.971	0.968	0.997	0.997	0.998	0.998	0.998	0.998
TUT	CNN	0.411	0.373	0.652	0.650	0.811	0.803	0.867	0.862	0.891	0.892
	+Dropout	0.551	0.542	0.611	0.601	0.815	0.810	0.903	0.902	0.893	0.892
	+Weight Decay	0.571	0.553	0.623	0.617	0.834	0.831	0.869	0.864	0.927	0.925
	CNN+VIB	0.607	0.595	0.674	0.670	0.855	0.857	0.912	0.913	0.942	0.943

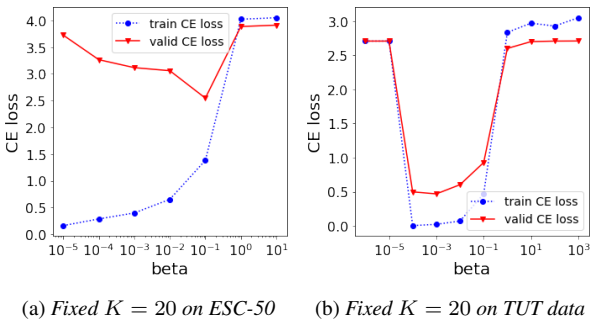


Figure 3: Validation and training cross-entropy losses of CNN+VIB for varying β with K fixed at 20.

vant information. As we increase β , where VIB has an effect, we observe better generalization performance with less overfitting. As β becomes too large (right side of the plot), both the training and validation CE losses shoot up because the amount of preserved information is insufficient to differentiate between the classes. Fig. 3b shows a slightly different case. When β is

too small, the model performs poorly on both the training and validation sets. As β rises, both training and validation losses reduce greatly. Therefore, a proper β is helpful for reaching a good local minima.

5. Conclusion

We present a deep VIB framework to address overfitting when training CNN classifiers on low-resource audio datasets. We conduct extensive experiments to verify its effectiveness in removing redundant information. Although we implement VIB method on CNNs in this paper, it is also suitable with other deep learning feature extractors. Therefore, our approach has good potential in audio classification tasks.

6. Acknowledgement

This work is supported by National Key Research and Development Program of China under grant No.2018YFB0204403, No.2017YFB1401202 and No.2018YFB1003500. Corresponding author is Jianzong Wang from Ping An Technology (Shenzhen) Co., Ltd.

7. References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.
- [4] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 131–135.
- [5] J. Cho, R. Pappagari, P. Kulkarni, J. Villalba, Y. Carmiel, and N. Dehak, "Deep neural networks for emotion recognition combining audio and transcripts," *Proc. Interspeech 2018*, pp. 247–251, 2018.
- [6] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [7] J. Guo, H. He, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha *et al.*, "Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing." *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.
- [8] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational intelligence magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [9] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [10] J. Nam, K. Choi, J. Lee, S.-Y. Chou, and Y.-H. Yang, "Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach," *IEEE signal processing magazine*, vol. 36, no. 1, pp. 41–51, 2018.
- [11] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.
- [12] M. Massoudi, S. Verma, and R. Jain, "Urban sound classification using cnn," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 583–589.
- [13] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6074–6078.
- [14] Z. Bai, X.-L. Zhang, and J. Chen, "Partial auc optimization based deep speaker embeddings with class-center learning for text-independent speaker verification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6819–6823.
- [15] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 81–85.
- [16] P. Georgiev, S. Bhattacharya, N. D. Lane, and C. Mascolo, "Low-resource multi-task audio sensing for mobile and embedded devices via shared deep neural network representations," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–19, 2017.
- [17] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 16–20.
- [18] R. K. mahabadi, Y. Belinkov, and J. Henderson, "Deep variational information bottleneck," in *International Conference on Learning Representations*, 2017.
- [19] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *The 37th annual Allerton Conf. on Communication, Control, and Computing*, 1999.
- [20] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 2015, pp. 1–5.
- [21] M. Chalk, O. Marre, and G. Tkacik, "Relevant sparse codes with variational information bottleneck," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.
- [22] R. K. mahabadi, Y. Belinkov, and J. Henderson, "Variational information bottleneck for effective low-resource fine-tuning," in *International Conference on Learning Representations*, 2021.
- [23] B. Dai, C. Zhu, B. Guo, and D. Wipf, "Compressing neural networks using the variational information bottleneck," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 1135–1144. [Online]. Available: <http://proceedings.mlr.press/v80/dai18d.html>
- [24] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-y. Lee, "Audio bert: A lite bert for self-supervised learning of audio representation," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 344–350.
- [25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: <http://arxiv.org/abs/1312.6114>
- [26] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.
- [27] M. K. Pichora-Fuller and K. Dupuis, "Toronto emotional speech set (TESS)," 2020. [Online]. Available: <https://doi.org/10.5683/SP2/E8H2MF>
- [28] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 85–92.
- [29] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R. P. Lippmann, Eds., vol. 4. Morgan-Kaufmann, 1992.
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://www.aclweb.org/anthology/N19-1423>
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.