GATED RECURRENT NETWORKS APPLIED TO ACOUSTIC SCENE CLASSIFICATION AND ACOUSTIC EVENT DETECTION

Matthias Zöhrer and Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria
matthias.zoehrer@tugraz.at, pernkopf@tugraz.at

ABSTRACT

We present two resource efficient frameworks for acoustic scene classification and acoustic event detection. In particular, we combine gated recurrent neural networks (GRNNs) and linear discriminant analysis (LDA) for efficiently classifying environmental sound scenes of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016). Our system reaches an overall accuracy of 79.1% on DCASE 2016 task 1 development data, resulting in a relative improvement of 8.34% compared to the baseline GMM system. By applying GRNNs on DCASE2016 real event detection data using a MSE objective, we obtain a segment-based error rate (ER) score of 0.73 – which is a relative improvement of 19.8% compared to the baseline GMM system. We further investigate semi-supervised learning applied to acoustic scene analysis. In particular, we evaluate the effects of a hybrid, i.e. generative-discriminative, objective function.

Index Terms— Acoustic Scene Labeling, Gated Recurrent Networks, Deep Linear Discriminant Analysis, Semi-Supervised Learning

1. INTRODUCTION

In acoustic scene classification the acoustic environment is labeled. Many different features, representing the scene, and models have been suggested in a recent acoustic scene classification challenge, summarized in [1]. One of the most popular baseline models are Gaussian mixture models (GMMs) [2] or hidden Markov models (HMMs) [3, 4] using mel-frequency cepstral coefficients (MFCCs). Interestingly, various deep architectures have not been applied in [1]. Recent work however, shows that deep neural networks (DNNs) boost the classification accuracy when applied to audio data [5]. In particular, Cakir et al. [6, 7] proposed a DNN architecture for acoustic scene classification. In [8], long-short-term memory networks (LSTMs), i.e. DNNs capable of modeling temporal dependencies, were applied to acoustic keyword spotting. Performance in recognition comes at the expense of computational complexity and the size of labeled data available. LSTMs have a relatively high model complexity. Furthermore, parameter tuning for LSTMs is not always simple.

This work was supported by the Austrian Science Fund (FWF) under the project number P27803-N15 and the K-Project ASD. The K-Project ASD is funded in the context of COMET Competence Centers for Excellent Technologies by BMVIT, BMWFJ, Styrian Business Promotion Agency (SFG), the Province of Styria - Government of Styria and the Technology Agency of the City of Vienna (ZIT). The program COMET is conducted by Austrian Research Promotion Agency (FFG). Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

Due to the great success of deep recurrent networks for sequence modeling [9, 10], we advocate gated recurrent neural networks (GRNNs) [11, 12, 13] for acoustic scene and event classification. GRNNs are a temporal deep neural network with reduced computational complexity compared to LSTMs. We evaluate GRNNs on environmental sound scenes of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016) [14]. GRNNs proof themselves in practice through fast and stable convergence rates. We obtain an overall accuracy of 79.1% on development data, i.e. a relative improvement of 8.34% compared to the baseline GMM system, using GRNNs and linear discriminant analysis (LDA). Furthermore, we used GRNNs for acoustic event detection, i.e. task 3 in DCASE2016. For this task we obtain a segment-based error rate (ER) of 0.82 and 0.63 for the scene categories *home* and *residential area*, respectively.

This work is structured as follows: Firstly, we introduce GRNNs in Section 3. In Section 4 we discuss various regularizers for GRNNs. Finally, we show experimental results for the challenge data in Section 7 and draw a conclusion in Section 8, respectively.

2. ACOUSTIC ANALYSIS FRAMEWORKS

2.1. Scene Classification Framework

Figure 1 shows the processing pipeline of our acoustic scene analysis framework. We extract sequences of features frames x_f, where $x_f \in \mathbb{R}^D$. In particular, we derive *MFCCs* or *log-magnitude spectrograms*, given the raw audio data x_t. We feed frequency domain features into the GRNN and estimate a class label for every frame f. Finally, we compute a histogram over all classified frames of the audio segment, where the maximum value determines the final scene class.

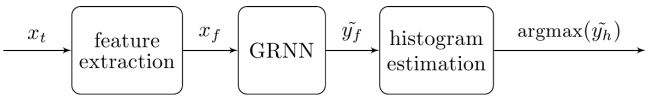


Figure 1: DCASE2016 task 1: GRNN scene classification system.

2.2. Event Detection Framework

Figure 2 shows the processing pipeline of our acoustic event detection framework. Similar as above, we extract sequences of feature frames x_f of *MFCCs* or *log-magnitude spectrograms*, given the raw audio data x_t. These feature frames are processed by a GRNN and

class labels are determined by applying individual thresholds on the real-valued output of the GRNN. Similar as in [14], we post-process the events by detecting contiguous regions neglecting events smaller than 0.1 seconds as well as ignoring consecutive events with a gap smaller than 0.1 seconds.

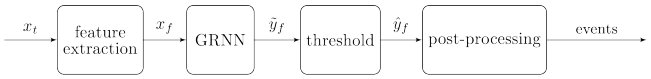


Figure 2: DCASE2016 task 1: GRNN event detection system.

3. DISCRIMINATIVE GRNNS

GRNNs are recurrent neural networks (RNNs) using blocks of gated recurrent units. GRNNs are a simple alternative to LSTMs, reaching comparable performance, but having fewer parameters. They only use *reset*- and *update*-gates. These switches couple static and temporal information allowing the network to learn temporal information. In particular, the *update*-gate z decides to re-new the current state of the model, whenever an important event is happening, i.e. some relevant information is fed into the model at step f. The *reset*-gate r is able to delete the current state of the model, allowing the network to forget the previously computed information. Figure 3 shows the corresponding flow diagram of a GRNN layer, respectively. It gives a visual interpretation how the *update*- and *reset*-gates, i.e. z and r, govern the information in the network.

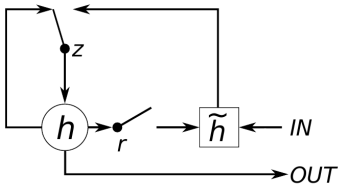


Figure 3: Flow graph of one GRNN layer [12].

The Equations (1-4) model the network behavior, mathematically. Starting at the output state h_f^l of layer l, the network uses the *update*-state z_f^l to compute a linear interpolation between past state h_{f-1}^l and current information $\tilde{h}_f^l$ in (1). In particular, the *update*-state z_f^l decides how much the unit updates its content; z_f^l is computed as sigmoid function of input x_f^l and the past hidden state h_{f-1}^l in Equation (2). The weights and bias terms in the model are denoted as W and b, respectively.

$$h_f^l = (1 - z_f^l)h_{f-1}^l + z_f^l \tilde{h}_f^l \quad (1)$$

$$z_f^l = \sigma(W_z^l x_f^l + W_{hz}^l h_{f-1}^l + b_z^l) \quad (2)$$

$$\tilde{h}_f^l = g(W_x^l x_f^l + W_{hh}^l (r_f^l \cdot h_{f-1}^l) + b_h^l) \quad (3)$$

$$r_f^l = \sigma(W_r^l x_f^l + W_{hr}^l h_{f-1}^l + b_r^l) \quad (4)$$

The state $\tilde{h}_f^l$ of the network is computed by applying a non-linear function g to the affine transformed input and previous hidden state h_{f-1}^l in (3). This is similar to *vanilla* RNNs. However, an additional *reset*-state, i.e. r_f^l, is introduced in GRNNs. In particular, an element-wise multiplication is applied between r_f^l and h_{f-1}^l. In (4), the reset state is computed based on the current input frame x_f and the provided hidden state h_{f-1}^l. Multiple GRNN layers can be stacked, forming a deep neural network.

4. DISCRIMINATIVE-GENERATIVE GRNNS

Recent advances in the field of semi-supervised learning combines discriminative learning objectives with generative cost terms [15, 16, 17, 18]. In particular, by modeling the data frame x_f using unlabeled examples, discriminative training objectives are regularized to prevent overfitting. These so called *hybrid* architectures outperform pure discriminative models if little labeled information is available. In order to exploit this regularization constraint, a reconstruction $\tilde{x}_f$ of the input frame x_f is computed by routing the network's output $\tilde{y}_f$ back to the bottom layer. This is done in any auto-encoder network by default [19], but could be also achieved via a separate *decoder*-network, visualized in Figure 4.

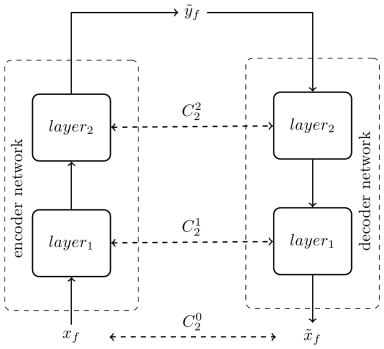


Figure 4: Flow graph of 2-layer hybrid GRNN network.

Following the idea of [15], we add an additional GRNN *decoder* network to the model. In particular, we use a noisy version of the input, i.e. $x_f + \mathcal{N}(\mu{=}0, \sigma{=}1)$, compute the output activation and feed the network's output $\tilde{y}_f$ back into the *decoder* network, which passes the information layer-by-layer down to the bottom and compute a reconstruction $\tilde{x}_f$ of the input. Next, the MSE between every hidden states h^l of a clean encoder and noisy decoder is computed. Adding this generative regularization term to the network's objective leads to the following hybrid objective function:

$$C = \underbrace{C_1(\tilde{y}_f, y_f)}_{\textit{discriminative}} + \lambda \cdot \underbrace{\frac{1}{L}\sum_{l=0}^{L} C_2^l(h_e^l, h_d^l)}_{\textit{generative}}, \quad (5)$$

where C_1 and C_2 are specific cost functions, such as the MSE criteria. The variable $\tilde{y}_f$ is the network's output and y_f is the current target label. The states h_e^l denote the hidden states of the *encoder* and h_d^l the hidden states of the *decoder*, respectively. The variable λ determines the tradeoff between the generative and discriminative objective.

5. VIRTUAL ADVERSARIAL TRAINING

Virtual adversarial training (VAT) [20, 17, 21] regularizes discriminative learners by generating *adversarial* training examples. Given

a clean training example x_f, an input noise pattern $\tilde{n}$ is generated by maximizing the KL-divergence between $P(y_f|x_f)$ and $P(y_f|x_f + n)$ using a softmax output layer, where the noise n is limited to $||n||_2 < \epsilon$, i.e. to the sphere of radius ϵ located around x_f. This means that the perturbed $x_f + \tilde{n}$ maximally changes the KL divergence between the posterior distributions, i.e the virtual adversarial example is most sensitive with respect to the KL-divergence. The newly obtained adversarial sample $\tilde{x}_f = x_f + \tilde{n}$ is used as additional training example. VAT can be used as a semi-supervised learning criteria. In this case a contractive cost term is applied on unlabeled data, scaled by a parameter λ. Further details can be found in [20].

6. DEEP LINEAR DISCRIMINANT ANALYSIS

Deep linear discriminant analysis (DLDA) [22] combines neural networks with the linear discriminant analysis (LDA). LDA is a discriminative learning criterion minimizing the inner class variance and maximizing the between class variance. Due to the lack of representational power the LDA criterion is usually not applied to high dimensional data. However, if combined with a non-linear system acting as a frontend, the LDA boosts classification performance to a certain extend. Following [22], we aim to maximize the eigenvalues v_i of the generalized eigenvalue problem:

$$S_b e_i = v_i(S_w + \lambda I)e_i, \tag{6}$$

where I is the identity matrix, S_b is the between class scatter matrix and S_w is the within class scatter matrix extracted from the network's output given the target labels, respectively. Details are in [22]. The eigenvalues $\{v_i, ..., v_k\}$ reflect the separation in the corresponding eigenvector space $\{e_1, ..., e_k\}$. In [22], they propose to optimize the smallest of all $C - 1$ eigenvalues. This leads to the following discriminative optimization criterion:

$$\underset{\theta}{\operatorname{argmax}} \frac{1}{k} \sum_{i=1}^{k} v_i, \tag{7}$$

where $\{v_1, ..., v_k\} = \{v_i | v_i < \min\{v_1, ..., v_{C-1}\} + \epsilon\}$, and ϵ acts as a threshold pushing variance to all $C - 1$ feature dimensions. This prevents the network from maximizing the distance to classes where good separation have already been found, and forces the model to concentrate on potentially non-separated examples instead. The cost function is differentiable, therefore, any neural network trainable with backpropagation can act as a frontend. The parameters θ are the network's weights and bias, respectively.

7. EXPERIMENTS

7.1. Experimental Setup: Acoustic Scene Classification

We pre-processed all DCASE2016 utterances with a STFT using a hamming window with window-size 40ms and 50% overlap. Next, MFCCs including Δ- and Δ^2-features were computed. All features were normalized to zero-mean unit variance using the training corpus. For the experiments we used either MFCCs + Δ + Δ^2 features, resulting in a 60-bin vector per frame as in [14], or raw 1025-bin log magnitude spectrograms. In order to guarantee a stable stochastic optimization, all observations need to be randomized. We implemented a variant of *on-the-fly* shuffling proposed in [23]. In particular, we processed batches of 500 randomly indexed, time-aligned utterance-chunks, cropped to a fixed length of 100 frames, in each optimization step. By doing so, we ensure proper randomization, preserving the sequential ordering of each utterance. The final classification score was obtained by computing a majority vote over all classified frames of the acoustic scene signal.

We put much effort in designing a solid machine learning framework which is also runnable on an embedded system. Therefore, we did not make use of ensemble or boosting methods [24], which usually, increases the classification performance. We built a single multi-label classification system instead. In particular, we used 3-layer GRNNs initialized with orthogonal weights [25] and rectifier activation functions. A linear output gate was used as a top layer. All networks have 200 neurons per layer. ADAM [26] was used for optimizing either the MSE or LDA objective.

7.2. Experimental Database: Acoustic Scene Classification

The DCASE2016 task 1 scene dataset is divided into a training and test set consisting of 1170 and 290 scene recordings, respectively. K-fold cross-validation was used for training all networks. In particular, we split the training corpus into 4 folds including 880 training utterances and 290 validation scenes, respectively. We report the average classification accuracy for all 4-folds of the training set. The labels of the test set are not published yet. More details about the data and the evaluation setup are in [14].

7.3. Experimental Results: Acoustic Scene Classification

Table 1 shows the overall scores of the DCASE2016 task 1, i.e. acoustic scene classification. We compared different feature representation using a 3-layer GRNN. Feeding raw spectrograms into the network slightly improves the classification performance, compared to MFCCs. This is consistent with the findings of [27].

Model	Features	Objective	Accuracy
baseline	MFCC	MLE	72.5%
GRNN	MFCC	MSE	74.0%
GRNN	spectrogram	MSE	76.1%
GRNN	MFCC	LDA	78.2%
GRNN	spectrogram	LDA	**79.1**%

Table 1: DCASE2016 task 1: Comparing MSE and LDA objectives using a 3-layer GRNN on different input feature representations.

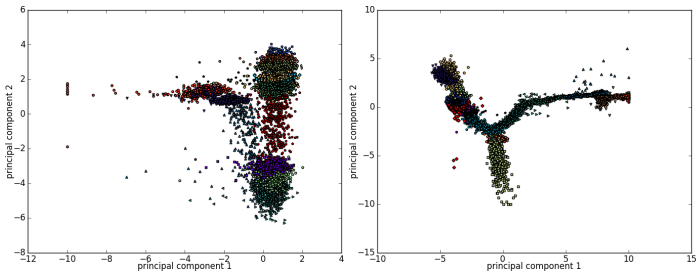
The use of a temporal model boosts recognition results in general. Most interestingly, the use of the LDA criterion achieved the best overall result, i.e. 79.1%, which leads to a relative improvement of 8.34% compared to the GMM baseline.

Table 2 shows the overall accuracy of GRNNs using a VAT regularized objective including the evaluation and test data as a semi-supervised data set. Furthermore, results for semi-supervised discriminative-generative GRNNs using the MSE objective are reported. VAT slightly improves the classification performance when using MFCC features. However, the result is slightly worse compared to the LDA criterion. The use of an additional generative cost function slightly improves the results. In particular, the *hybrid* learning criterion, i.e. Equation 5, achieves a relative improvement of 3.8% compared to the baseline system. However, VAT still

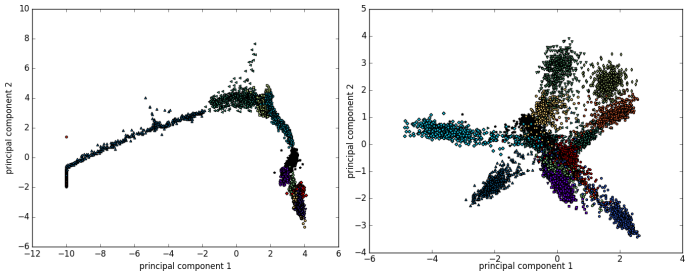
Model	Regularizer	Features	Objective	Accuracy
GRNN	VAT	MFCC	MSE	**77.8**%
GRNN	VAT	spectrogram	MSE	77.4%
GRNN	MSE (Eq. 5)	MFCC	MSE	75.4%
GRNN	MSE (Eq. 5)	spectrogram	MSE	**76.7**%

Table 2: DCASE2016 task 1: Semi-supervised training with a 3-layer GRNN using a VAT regularizer ($\lambda = 0.1, \epsilon = 0.25, I_p = 1$) and hybrid MSE objective (λ =1e-4).

outperformed the *hybrid* MSE objective.



(a) supervised (MSE) (b) semi-supervised (MSE)



(c) semi-supervised (MSE+VAT) (d) supervised (LDA)

Figure 5: Juxtaposition of supervised and semi-supervised training using a 3-layer GRNN and a subset of DCASE2016 scene spectrograms. Figure 5a-5d show the 1^{st} and 2^{nd} principal component of the activations generated from the last hidden layer using different optimization criteria.

Visual interpretations of the hidden activations provide some insights into neural networks, which are treated as *blackbox* models. Figure 5 shows the first two principal components of the activations the last hidden layer of a 3-layer GRNN, using a subset of DCASE2016 task 1 data. Starting with a pure discriminative learning criterion in Figure 5a, we see that a non-regularized MSE objective produces slightly overlapping clusters. By adding a *generative* cost function, i.e. *hybrid* MSE optimization criterion in (5), as well as a VAT regularizer (see Section 5) the inner class variance is lowered, improving the overall class margins in the end. In both, MSE and VAT objectives, the between class variance is not maximized. In Figure 5d however, we clearly see that the LDA criterion in (6) produces more separated class projections. In this case, the within class variance is minimized, whereas the between class variance is maximized.

7.4. Experimental Database: Acoustic Event Detection

The DCASE2016 task 3 acoustic event dataset is divided into a training and test set containing 22 recordings. The dataset has two scene categories, i.e. *home* and *residential area*. The *home* training corpus contains 7 event classes with 563 events, whereas the *residential area* training corpus contains 11 event classes including 906 events. Similar as in Section 7.2 K-fold cross validation, using 4 folds, was applied. More details about the data and the evaluation setup are in [14].

7.5. Experimental Setup: Acoustic Event Detection

We applied the same pre-processing routines, i.e. STFT calculation, MFCC and log-magnitude spectrogram extraction, as in Section 7.1 using data of the DCASE2016 sound event detection in real life audio challenge (task 3). Regarding the training procedure, we extended the *on-the-fly* shuffling routine in two ways: We drop frames with a probability of 50% and use smaller permuted sequence batches. By doing so, we increase the data size by introducing slight permutations and variations of the training sequences. Frames with multiple event labels were removed in the training corpus, forcing the model to extract class specific features. Apart from that, an additional *blank* label was introduced. The model sizes and configuration parameters are kept the same as in Section 7.1.

7.6. Experimental Results: Acoustic Event Detection

Model	Features	Objective	ER	F [%]
baseline	MFCC	MLE	0.90	37.3
GRNN	MFCC	MSE	0.74	42.3
GRNN	spectrogram	MSE	**0.73**	**47.6**

Table 3: DCASE2016 task 3: Classification results GRNNs using MFCCs or log-magnitude spectrograms and a MSE objective.

Model	Acoustic Scene	Segment-based		Event-based	
		ER	F [%]	ER	F [%]
GRNN	home	0.82	37.3	1.55	2.9
GRNN	residential area	0.63	57.9	4.64	0.9
GRNN	average	0.73	47.6	3.9	1.9

Table 4: DCASE2016 task 3: Detailed classification results with a 3-layer GRNN using a MSE objective and log-magnitude spectrograms.

Table 3 shows the results of the DCASE2016 task3 real audio event detection task using GRNNs. We did not apply a LDA due to overlapping events in the test set. GRNNs trained on log-magnitude spectrograms achieved an overall segment-based error rate (ER) of 0.73 and an F-score of 47.6%. This results in a relative improvement of 19.8% and 51.1% compared to the baseline GMM model for the ER- and F-scores, respectively. The error measures are specified in detail in [14]. In Table 4 detailed segment- and event-based results for both, *home* and *residential area* are reported.

8. CONCLUSION

We applied gated recurrent neural networks (GRNNs) to acoustic scene and event classification. In particular, we trained a 3-layer GRNN on environmental sounds of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016) (task 1 and task 3). The use of virtual adversarial training (VAT) slightly improves the model performance using MFCC features. For scene classification, models trained with a deep linear discriminant objective (LDA) using log-magnitude spectrogram representations outperformed VAT regularized networks. For acoustic event detection we use a multi-label GRNN. For both tasks we outperform the GMM baseline system significantly.

9. REFERENCES

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[2] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *18th European Signal Processing Conference*, 2010, pp. 1267–1271.

[3] J. Keshet and S. Bengio, "Discriminative keyword spotting," in *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*. Wiley Publishing, 2009, pp. 173–194.

[4] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4087–4091.

[5] M. Zhrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. combined single-label sound event detection with deep neural networks," in *23rd European Signal Processing Conference 2015 (EUSIPCO 2015)*, 2015.

[7] E. Cakir, T. Heittola, H.Huttunen, and T.Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[8] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5236–5240.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3104–3112.

[10] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.

[11] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014.

[12] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *CoRR*, vol. abs/1502.02367, 2015.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[15] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3546–3554.

[16] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3581–3589.

[17] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," *CoRR*, vol. abs/1511.05644, 2015.

[18] M. Zöhrer and F. Pernkopf, "General stochastic networks for classification," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2015–2023.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[20] T. Miyato, S. Shin-ichi Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing by virtual adversarial examples." *CoRR*, vol. abs/1507.00677, 2015.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[22] M. Dorfer, R. Kelz, and G. Widmer, "Deep linear discriminant analysis," *International Conference of Learning Representations (ICLR)*, vol. abs/1511.04707, 2015.

[23] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[24] H. Drucker, R. Schapire, and P. Simard, "Improving performance in neural networks using a boosting algorithm," in *Advances in Neural Information Processing Systems*, 1993, pp. 42–49.

[25] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *International Conference of Learning Representations (ICLR)*, 2014.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[27] M. Espi, M. Fujimoto, and T. Nakatani, "Acoustic event detection in speech overlapping scenarios based on high-resolution spectral input and deep learning." *IEICE Transactions on Information and Systems E98D*, pp. 1799–1807, 2015.