Dilated-Gated Convolutional Neural Network with A New Loss Function on Sound Event Detection

Ke-Xin He*, Wei-Qiang Zhang*, Jia Liu*, Yao Liu†
* Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
E-mail: hekexinchn@163.com, wqzhang@tsinghua.edu.cn, liuj@tsinghua.edu.cn
† China General Technology Research Institute
E-mail: liuyao88@mail.ustc.edu.cn

***Abstract*—In this paper, we propose a new method for rare sound event detection. Compared with conventional Convolutional Recurrent Neural Network (CRNN), we devise a Dilated-Gated Convolutional Neural Network (DGCNN) to improve the detection accuracy as well as computational efficiency. Furthermore, we propose a new loss function. Since frame-level predictions will be post processed to get final prediction, continuous false alarm frames will lead to more insertion errors than single false alarm frame. So we adopt a discriminative penalty term to the loss function to reduce insertion errors. Our method is tested on the dataset of Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Challenge task 2. Our model can achieve an F-score of 91.3% and error rate of 0.16 on the evaluation dataset while baseline achieves an F-score of 87.5% and error rate of 0.23.**

I. Introduction

Recently, sound event detection (SED) has become increasingly popular in the field of acoustic signal processing. The goal of sound event detection is to detect the sound event and the time boundaries.

Several challenges have been organized on the topic of SED. The first Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge is organized by Queen Mary University in 2013, creating an opening into the sphere of public evaluations for everyday sounds. DCASE 2017 Challenge consists of four tasks and our research is relevant to task 2 — detection of rare sound events [1].

Typical neural networks used in SED task include Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNNs are able to extract higher level features. RNNs have shown strong performance in learning the longer term temporal context. A method combining CNN with RNN called Convolutional Recurrent Neural Network (CRNN) has been proposed and shown further improvements [2], [3], [4]. In the task 2 of DCASE 2017 Challenge, Lim et al. [2] and Cakir et al. [3] utilized CRNN and they won the top two. There are also some variants and developments of CRNN in this task. For instance, Kao et al. [5] proposed a regional-based CRNN architecture and Shen et al. [6] proposed a CRNN with temporal-frequential attention mechanism. RNN plays an essential part in all of the above work.

However, in RNN, the next output depends on the previous hidden state which does not allow parallelization. Training or inference of RNN model will cost a lot of time. Our research is SED based on CNN which can be more efficient since they allow parallelization over sequential tokens.

In deep learning, the choice of loss function is also important. Phan et al. proposed weighted and multi-task loss [7] for SED. In [8], the authors proposed a focal loss in object detection, and it has shown great power to reduce data imbalance in object detection. The dataset in SED is also highly unbalanced. Inspired by [8], we propose a new loss function to solve this problem.

Besides, we find that the most common error in SED is not deletion error but insertion error, which means system classifies the background sound as target sound event. So a discriminative penalty term is devised to avoid such errors.

Our key contributions in this paper are as follows:

1) We replace currently popular CRNN with Dilated-Gated Convolutional Neural Network (DGCNN) to enhance the ability to capture context. Meanwhile, DGCNN is based on convolutional neural network, leading to faster computation at both training and test time. So proposed method can achieve more efficient performance without accuracy degrading.
2) We propose a new loss function for SED to reduce insertion errors. Our elaborate loss function can significantly improve the performance of SED.

II. Methods

A. System Overview

The overview of SED system is illustrated in Figure 1. The input of our system is log filter bank (fbank) feature. We utilize deep learning model based on DGCNN as classifier. The neural network will output a set of scores, denoting the presence probabilities of sound event in each frame. After post-processing including threshold method and median filtering, scores will be converted into the start time and end

The corresponding author is Wei-Qiang Zhang.

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219 and the National Key R&D Program of China.

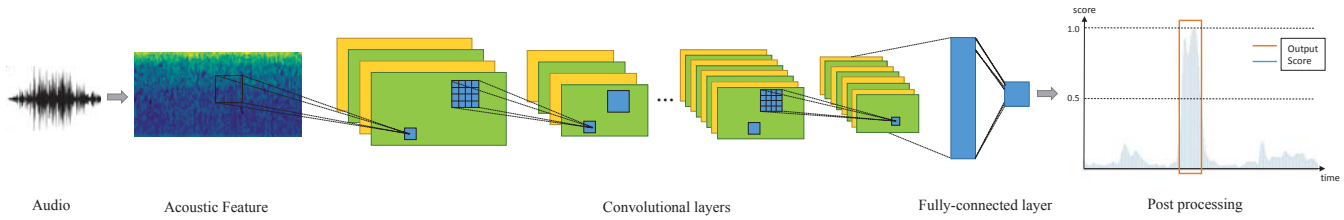

Fig. 1. Overall architecture of proposed system.

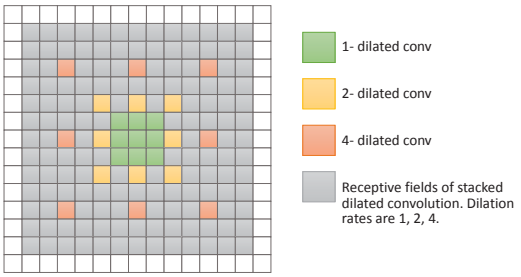

Fig. 2. Illustration of 3×3 convolution kernels with different dilation rates as 1, 2 and 4.

time of the target events. During training, a new loss function is applied to enhance the classification capability of our model.

B. Dilated-gated Convolutional Neural Network

In this subsection, we will introduce dilated convolution, gated convolution and their combination.

Dilated convolution is devised to exponentially expand the receptive field with linearly increasing number of parameters. It has been demonstrated in context aggregation with significant improvement of accuracy [9]. The specific formula of dilated convolution is:

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r*i, n + r*j)\, w(i,j) \qquad (1)$$

where $x(m,n)$ is a 2-D signal, $y(m,n)$ is the output of dilated convolution with the dilation rate r and a filter $w(i,j)$. Dilated convolution is equivalent to a normal convolution when $r = 1$.

Dilated convolution works by introducing "holes" [10] in the kernels. It is equivalent to a convolution with a larger filter derived from the original filter by dilating it with zeros. Compared with normal convolution, dilated convolution is powerful in learning longer term temporal context. For example, three stacked normal convolution layers with 3×3 kernel size only have 7×7 receptive fields. However, as shown in Figure 2, three stacked dilated convolution layers with dilation rate 1, 2 and 4 have 15×15 receptive fields. The larger receptive fields, the stronger ability to learn longer term temporal context.

Gated Convolutional Neural Network (GCNN) is proposed by Dauphin in [11] to train language model. It utilizes gated

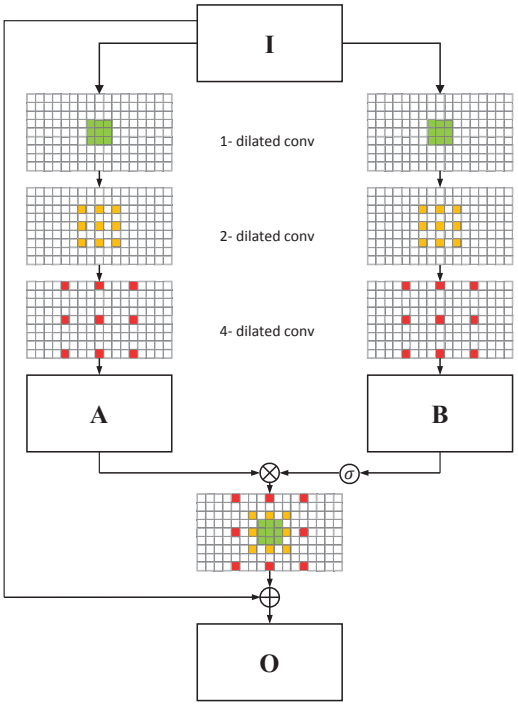

Fig. 3. Illustration of Dilated-Gated convolutional block.

units which are similar to RNN to control information flow to the next layer. But compared with RNN, it greatly improves the computational efficiency of networks. In GCNN, the output of the convolutional layer is divided into A and B. A is modulated with gated weights $\sigma(B)$, where $\sigma(\cdot)$ is sigmoid activation.

We apply the dilated convolution to gated unit in GCNN. We called this new network DGCNN, which supports exponential expansion of the receptive field without loss of resolution or coverage.

The structure of dilated-gated convolutional block is illustrated in Figure 3. The input of this structure I will pass 2 architectures of 3 dilated convolution layers, turning to two tensors A and B. The dilation rate of the three layers are 1, 2, 4 respectively. Then B passes through sigmoid activation function and multiplies with A by element-wise. Meanwhile, in order to enable stronger performance, we add residual connections from the input I to the output of DGCNN. Residual network is introduced to avoid vanishing gradient problem [12].

The specific formula of DGCNN is:

$$A = I * W_1 + b_1 \tag{2}$$

$$B = I * W_2 + b_2 \tag{3}$$

$$O = I + A \otimes \sigma(B) \tag{4}$$

where W_1, W_2 represent convolutional kernel values, and b_1, b_2 represent biases. $\otimes$ represents element-wise production. $\sigma(\cdot)$ is a sigmoid function.

C. A New Loss Function

In SED task, our system will output a set of scores. After post-processing, scores will be converted into the start time and end time. We use the threshold method to get the prediction of each frame. Then the median filtering on predictions is applied to reduce interference from background noise. The longest continuous positive sequence is considered as the target event.

If false alarm occurs in a single frame, this mistake can be avoided with a filter. But if false alarms occur in a continuous sequence, it may be difficult to avoid with a filter. Sometimes the system may detect a long sequence of false alarms as sound event. So we think that a single-frame false alarm should be treated differently with a continuous long sequence of false alarms.

To tackle this problem, we devise a discriminative penalty term to avoid continuous misclassifications. In our architecture, each frame will output a vector through convolutional layers. This vector can be considered as the feature vector of this frame. We set a parameter n, to count the Jaccard similarity coefficient [13] of false alarm frame with adjacent n frames as our penalty term. In an ideal case, there are only two impulses in the outputs of our system: the beginning and the end of target events. So we should try to smooth the outputs of our system. The specific formula of the discriminative penalty term is as follows:

$$C = \sigma\left(\sum_{i=0}^{N}\sum_{j=x-n}^{x+n} J\left(O_x, O_j\right)\left(1 - y^{(i)}\right)\hat{y}^{(i)}\right) \tag{5}$$

$$J\left(O_1, O_2\right) = \frac{\sum_i \min\left(O_{1i}, O_{2i}\right)}{\sum_i \max\left(O_{1i}, O_{2i}\right)} \tag{6}$$

where O_x represents the output of DGCNN, O_{1i} and O_{2i} are the i-th element of O_1 and O_2. Jaccard similarity coefficient $J \in [0, 1]$. The larger the jaccard coefficient value, the higher the sample similarity. $y^{(i)}$ represents label, and $\hat{y}^{(i)}$ represents prediction value. In experiments, the value of n is 10. Adding C to loss function can avoid most continuous false alarms.

Futhermore, SED is a task with unbalanced dataset. To mitigate data imbalance, the common method is to set different weight coefficients to different classes. While weight coefficients balance the importance of positive and negative samples, it does not differentiate between easy and hard samples. Due to data imbalance, negative samples contribute a lot to the loss function. But most of those negative samples are easily classified and cannot provide enough information for classification. So we should focus more on the negative samples that can not give a correct prediction easily (hard sample). We utilize a power function y^γ as weight coefficient to mitigate data imbalance. This loss function is similar to focal loss [8]. The specific formula of loss function is as follows:

$$\begin{aligned} L = -\frac{1}{N}\sum_{i=1}^{N}[w_p y^{(i)} \log(\hat{y}^{(i)}) + \\ (1 - y^{(i)})(\hat{y}^{(i)})^2 \log(1 - \hat{y}^{(i)})] \end{aligned} \tag{7}$$

where $y^{(i)}$ represents label, and $\hat{y}^{(i)}$ represents prediction value. w_p is the weight for positive samples. The value of w_p is 5. $(\hat{y}^{(i)})^2$ means greatly reducing the loss from easy negative samples.

In SED task, the positive samples are usually labeled 1 and negative samples are labeled 0. However, this method of notation cannot reflect the weight of each positive samples accurately. Because the samples close to target event center and those far from target event center are treated equally. But in fact, sound events almost appear with blurred boundaries, and it is hard to distinguish them from the background clutters. This issue is more pronounced for short sound event. So we should focus more on the center of target events and pay less attention to the margin. In order to implement this idea, we give larger weights to those frames around the event center in the loss function. The improved loss function is:

$$\begin{aligned} L = -\frac{1}{N}\sum_{i=1}^{N}[w_p y^{(i)} \log(\hat{y}^{(i)}) + \\ (1 - y^{(i)})(\hat{y}^{(i)})^2 \log(1 - \hat{y}^{(i)})] * \lambda_{ca} \end{aligned} \tag{8}$$

$$\lambda_{ca}\left(I\right) = \frac{k}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(I - \mu_c\right)^2}{2\sigma^2}\right) \tag{9}$$

where $y^{(i)}$ represents label, and $\hat{y}^{(i)}$ represents prediction value. I is the index of current frame, μ_c represents the index of central frame of events. We set σ to 10. λ_{ca} makes network focus more on the center of target events.

III. Experiments

A. Dataset

We demonstrate our proposed system on the dataset [14] provided by DCASE 2017 Challenge task 2. The dataset is divided into development dataset and evaluation dataset. We use a subset of the development dataset to train and another subset to optimize our model. Finally we evaluate our system on the evaluation dataset. Dataset consists of isolated sound events for each target class and recordings of everyday acoustic scenes to serve as background. Target sound events include

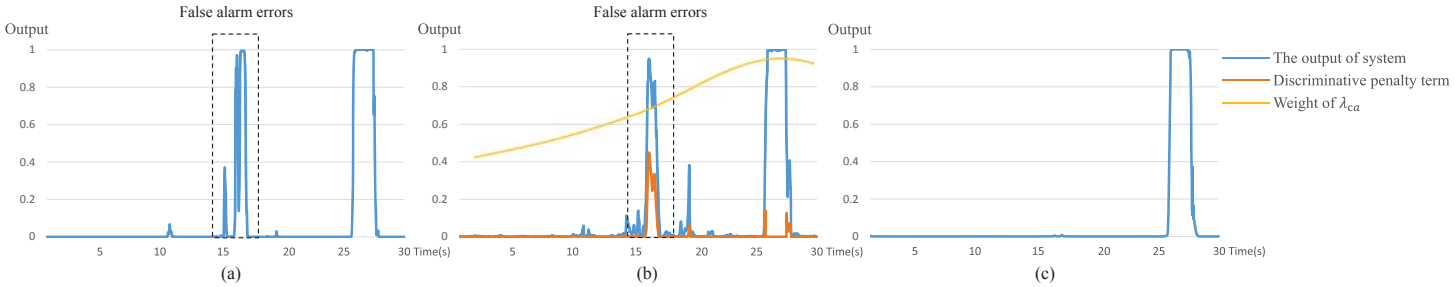

Fig. 4. Illustration of how our methods work. (a) The output of GCNN. (b) The output of DGCNN. (c) The output of DGCNN with new loss function. False alarm errors occur in the dotted box, and the discriminative penalty term is high in this area.

baby crying, glass break and gunshot. The background audio material consists of recordings from 15 different audio scenes.

The synthesizer is provided by DCASE challenge, and we use it to generate the training set. The mixing event-to-background ratios (EBR) are -6, 0 and 6 dB. The event occurrence probability is set to 0.9. The generated training set has 3000 monaural mixed audios with 44,100 Hz and 24 bits for each target class, and each mixture contains one target event or no events.

B. Experiment Setup

We use fbank as the acoustic feature. It has been widely used in SED with deep neural networks and has good performance [15], [4]. Each audio sample is divided into 40 ms frames with 50% overlap and 128 log mel-band energy features are extracted from the magnitude spectrum of each frame. Finally, each feature is normalized to zero mean and unit standard deviation.

TABLE I
MODEL STRUCTURE AND PARAMETERS OF PROPOSED NETWORK.

Input 128×1500×1	Output size
Conv (kernel: [5, 5, 32])	128,1500,32
BN-ReLU-Dropout(0.2)-Maxpooling(4×1)	32,1500,32
Conv (kernel: [3, 3, 64])	32,1500,64
BN-ReLU-Dropout(0.2)-Maxpooling(4×1)	8,1500,64
Dilated-Gated Conv (kernel: [3, 3, 64])	8,1500,64
BN-ReLU-Dropout(0.2)-Maxpooling(4×1)	2,1500,64
Gated Conv (kernel: [3, 20, 64])	2,1500,64
BN-ReLU-Dropout(0.2)-Maxpooling(2×1)	1,1500,64
Fully-connected(unit num: 64) -ReLU-Dropout(0.2)	1500,64
Fully-connected(unit num: 1)	1500,1

The proposed network consists of two main parts: convolutional layers and fully-connected layers. Convolutional layers consist of two normal CNNs, a dilated-gated convolutional block and a gated convolutional layer. The output of each convolutional layer is followed by batch normalization [16], a ReLU activation unit [17] and a dropout layer [18]. Then a max-pooling layer is applied to keep some important features. Then two fully-connected layers are used to combine extracted features and output a set of scores. The structure of proposed network is shown in Table 1 along with parameters.

TABLE II
PERFORMANCE ON DEVELOPMENT DATASET, IN TERMS OF DELETION ERRORS AND INSERTION ERRORS. (1) DGCNN: DILATED-GATED CONVOLUTIONAL NEURAL NETWORK. (2) PROPOSED: DGCNN WITH NEW LOSS FUNCTION.

	babycry			glassbreak			gunshot		
	Del.	Ins.	ER	Del.	Ins.	ER	Del.	Ins.	ER
DGCNN	0.08	0.09	0.16	0.02	0.03	0.05	0.12	0.12	0.23
Proposed	0.07	0.06	0.13	0.02	0.02	0.04	0.11	0.09	0.20

TABLE III
PERFORMANCE ON DEVELOPMENT DATASET, IN TERMS OF ER AND F1. (1) GCNN: GATED CONVOLUTIONAL NEURAL NETWORK. (2) DGCNN: DILATED-GATED CONVOLUTIONAL NEURAL NETWORK. (3) PROPOSED: DGCNN WITH NEW LOSS FUNCTION.

	babycry		glassbreak		gunshot		average	
	ER	F1	ER	F1	ER	F1	ER	F1
GCNN [a]	0.19	90.4	0.06	97.0	0.27	86.1	0.17	91.2
DGCNN	0.16	91.8	0.05	97.4	0.23	87.3	0.15	92.2
Proposed	0.13	93.3	0.04	97.7	0.20	89.2	0.12	93.4

[a] In GCNN, we replace dilated-gated convolutional layer in DGCNN with a gated convolutional layer.

Adam [19] is adopted for gradient based optimization. The initial learning rate is 0.001 and the batch size is 64. We train the classifiers for 100 epochs.

C. Evaluation Metrics

The evaluation metrics for the task are event-based error rate (ER) and F1-score calculated using onset-only condition with a collar of 500 ms. ER is the sum of deletion and insertion error. F1-score is the harmonic average of precision and recall. Sed_eval toolbox is provided by the challenge to evaluate our system. More details of the evaluation metrics can be found in [20].

D. Experimental Results

We compare our proposed system with DGCNN on development dataset, in terms of deletion errors and insertion errors in Table 2. Compared with DGCNN, the insertion errors of proposed system decrease greatly.

The results of our methods and other methods, in terms of ER and F1-score, are given in Table 3 and Table 4. Our best system outperforms most systems except the method ranked 1st in the challenge. We think the reasons leading to this results

TABLE IV
PERFORMANCE ON EVALUATION DATASET, IN TERMS OF ER AND F1. (1) BASELINE: OFFICIAL BASELINE PROVIDED BY DCASE COMMITTEE. (2) 1D-CRNN: DCASE 1ST PLACE MODEL. (3) CRNN: DCASE 2ND PLACE MODEL. (4) AED-NET: DCASE 3RD PLACE MODEL.

	babycry		glassbreak		gunshot		average	
	ER	F1	ER	F1	ER	F1	ER	F1
GCNN	0.27	86.0	0.10	94.9	0.31	81.6	0.23	87.5
DGCNN	0.22	88.7	0.08	95.9	0.27	85.5	0.19	90.0
Proposed	**0.17**	**90.3**	**0.08**	**95.9**	**0.24**	**87.6**	**0.16**	**91.3**
Baseline	0.80	66.8	0.38	79.1	0.73	46.5	0.64	64.1
1d-CRNN [2]	0.15	92.2	0.05	97.6	0.19	89.6	0.13	93.1
CRNN [3]	0.18	90.8	0.10	94.7	0.23	87.4	0.17	91.0
AED-Net [7]	0.23	88.4	0.11	94.3	0.32	82.1	0.22	88.2

TABLE V
RUNTIME COMPARISON BETWEEN DGCNN AND CRNN.

	DGCNN	CRNN
train time	0.56 hours	5.22 hours
train speedup	9.3×	1×
test time per 30s audio	0.004 seconds	0.5 seconds
test speedup	125×	1×

are that our method is completely based on convolutional neural network. Admittedly, RNN has stronger capability of processing sequential data in spite of its slow computation. We can achieve comparable performance with RNN only using the combination of DGCNN and a new loss function. The runtime comparison between DGCNN and CRNN model based on Tesla P100 is shown in Table 5. We use the CRNN model in [2], which consists of one convolution layers and two LSTM layers. The efficiency of our system is much better than CRNN.

Figure 4 is illustration of how our methods work. In Figure 4 (a), the blue curve denotes the outputs of GCNN. There is a long continuous sequence of false alarm frames from 15 s to 16 s, which may probably lead to insertion error. In Figure 4 (b), there is still a serious misjudgment from 15 s to 16 s, but improvement has been made compared with GCNN. In addition, there are some fluctuations on the boundary of target event. This is possibly the result of the blurred boundaries of target events. The yellow curve represents weight of λ_{ca}. It indicates that our system focuses more on the occurrence of target events. The red curve is discriminative penalty term for insertion error. Bigger penalty coefficients are given where continuous false alarms occur, which can reduce insertion error. Shown in Figure 4 (c) is the result of DGCNN with new loss function. And the output curve of this system matches well with ground-truth labels.

IV. CONCLUSION

In this paper, we propose a dilated-gated convolutional neural network and a new loss function for sound event detection. We demonstrate our model on task 2 of the DCASE 2017 Challenge, and achieve competitive performance. Compared with CRNN, the speed of our model has been greatly improved. Furthermore, our new loss function can effectively reduce insertion errors. Experiments on task 2 of the DCASE 2017 Challenge demonstrate the effectiveness and efficiency of the proposed methods.

REFERENCES

[1] 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*, 2017, pp. 85–92.

[2] H. Lim, J. Park, K. Lee, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 80–84.

[3] E. Cakir and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 803–806.

[4] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[5] C. C. Kao, W. Wang, M. Sun, and C. Wang, "R-crnn: Region-based convolutional recurrent neural network for audio event detection," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[6] Y.-H. Shen, K.-X. He, and W.-Q. Zhang, "Learning how to listen: A temporal-frequential attention model for sound event detection," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.

[7] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, "Weighted and multi-task loss for rare audio event detection," in *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 336–340.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[9] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[11] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] L. Hamers *et al.*, "Similarity measures in scientometric research: The jaccard index versus salton's cosine formula." *Information Processing and Management*, vol. 25, no. 3, pp. 315–18, 1989.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Signal Processing Conference*, 2016, pp. 1128–1132.

[15] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1744–1748.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of The 32nd International Conference on Machine Learning*, 2015.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.