INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Dual Stage Learning Based Dynamic Time-Frequency Mask Generation For Audio Event Classification

Donghyeon Kim[1], *Jaihyun Park*[1], *David K. Han*[2], *Hanseok Ko*[1]

[1]Korea University, South Korea
[2]Army Research Lab, USA

dhkim_2@ispl.korea.ac.kr, jhpark@ispl.korea.ac.kr, ctmkhan@gmail.com, hsko@korea.ac.kr

Abstract

Audio based event recognition becomes quite challenging in real world noisy environments. To alleviate the noise issue, time-frequency mask based feature enhancement methods have been proposed. While these methods with fixed filter settings have been shown to be effective in familiar noise backgrounds, they become brittle when exposed to unexpected noise. To address the unknown noise problem, we develop an approach based on dynamic filter generation learning. In particular, we propose a dual stage dynamic filter generator networks that can be trained to generate a time-frequency mask specifically created for each input audio. Two alternative approaches of training the mask generator network are developed for feature enhancements in high noise environments. Our proposed method shows improved performance and robustness in both clean and unseen noise environments.

Index Terms: dynamic filter network, feature enhancement, dual stage, audio recognition

1. Introduction

Sound event recognition is an important area in audio analysis with many monitoring and surveillance applications [1, 2, 3]. In real audio streaming recognition environment, most of audio streams contain noise resulting a degradation in recognition performance. To alleviate this, various noise reduction methods have been developed. As a statistical approach, Wiener filter [4] and MMSE spectral amplitude estimator [5] were employed for noise reduction. As the algorithms focused on discovering statistical differences between target audio and noise, a high order statistical method based on kernel functions was proposed [6]. While demonstrating reasonable performance, this method required manually designed kernel functions. However, with the advent of deep learning framework, high order information can be automatically learned by multi layer perceptrons and can be applied to noise reduction. Sum *et al.* [7] introduced a separable auto encoder to estimate unseen noise by separating target audio and noise components with the prior knowledge about differences between target and noise. Badi *et al.* [8] proposed a skip-connection denoising auto-encoder to estimate enhanced acoustics features by applying Correlation Distance Measure as a penalty term to increase dependency between the clean and enhanced features. Also, an iterative model based on Griffin-Lim algorithm [9] and a generative model-based approach [10, 11] have been applied for audio feature enhancement. However, as these deep learning methods require prior knowledge about noise, their performance varied depending on the noise model assumption. To mitigate this limitation, robust feature enhancement approaches have been proposed for implementation in deep learning [12, 13]. In the proposed methods, a spectral mask is extracted by a neural architecture and elementwise multiplication is performed between the spectral mask and the input feature on the front end of the network. As the neural architecture is optimized by using the classifier loss function, the spectral mask is trained to enhance salient components for robust and improved downstream classification process. Although the spectral mask approach showed improved performance without any prior knowledge of noise, fixed parameters of the spectral mask upon completion of the training process may hinder its ability to generalize effectively over a variety of input noise types. We propose to mitigate the fixed parameter problem by using a Dynamic Filter Network (DFN) [14]. In the DFN framework, filter parameters of a neural network for generating spectral mask are produced by another neural network called "Filter Generator." Through supervised training, the Filter Generator is trained, but the network weights of the spectral mask are varied as output of the Filter Generator produces filters conditioned on input features. Thus the network dynamically adjusts the spectral mask conditioned on each input to enhance salient features for improving the classification performance.

However, directly applying DFN with a single pass spectral mask generation might not be adequate for sufficient improvement in downstream classification performance. The effectiveness of the Filter Generator and the spectral mask may get significantly degraded when input features contain high noise content. Some research efforts [15, 16] showed that an iterative mask estimation (IME) framework improve the performance of time-frequency mask. Therefore, we propose a dual stage dynamic feature enhancement method based on iterative learning [16, 9] and cascade learning [17] to further enhance effectiveness of the spectral mask. Our proposed method is tested against the Urbansound8k [18] dataset with seven noise types [19] at five levels of SNRs. The results show that our method achieves not only 5% improvement in clean environment but also attains superior results in most of the noisy environments over the prominent methods compared. The followings are the key contributions of our effort:

1. We propose a novel time-frequency based feature enhancement architecture based on the concept of dynamic filter generation.
2. We develop a novel structure of dynamic filter generation by integrating multiple filter generators and associated training strategies to significantly improve performance in noisy acoustic environments.

The remainder of this paper is organized as follows. The related work is described in Section 2. The proposed method, experimental process and conclusion are described in Sections 3, 4 and 5 respectively.

http://dx.doi.org/10.21437/Interspeech.2020-2152

2. Related work

2.1. Spectral masking

Spectral mask is a type of filter that can be applied to spectral representations [20] and we apply it here for enhancing noise invariant signals from a time-frequency feature prior to the classification process. Li *et al.* [12] introduced a mask estimation based on a fully-connected network which adopted Linear Input Network (LIN) to reduce mismatch between train and test data. Shon *et al.* [13] tried to separate an enhancement model and a classification model by using a pre-trained structure. Upon completion of pre-training, the parameters of the pre-trained classification model are fixed and only the parameters of the enhancement model are optimized by using a cross entropy loss [21] between prediction and label. In the output layer of the enhancement model, a sigmoid activation function was employed to perform attentive normalization with respect to the loss function. Although those methods showed performance improvements in noisy environment, static filters fixed from training may limit the performance in test stage when the input contains high noise content. For improving classification performance in distorted domain, we adopt DFN to handle unseen noise in the input.

2.2. Dynamic filter network

DFN is a type of filter adaptation method for deep neural network [14]. Instead of learning the filter parameters, they are generated by another neural network model to mitigate the fixed parameter problem in test stage. DFN consists of a filter generation model trained from training data and a dynamic filter layer which is the output of the generation model. Upon training, therefore, the filter generation model generates a dynamic filter per each audio input. Sharma *et al.* [22] tried to enhance an image feature in the dynamic framework to improve the classification performance. Instead of directly using handcrafted pre-processing technique, they generated dynamic filters which transformed YCbCr features to target pre-processing features. A CNN-FC model generates dynamic filters and a CNN based dynamic layer transformed YCbCr features. Classification was then performed in an end-to-end fashion. The entire learning parameters were optimized by a reconstruction loss which was based on Mean Square Error (target reference and represented feature) and a cross entropy loss (label and prediction) function.

Our approach is inspired by the dynamic filter development network by Jia *et al.* and others [14, 22], and the mask learning architecture by Shon *et al.* [13]. We propose a dual stage dynamic mask generator based feature enhancement concepts by considering both a cascade dual stage learning and an end-to-end shared generator weight learning to improve performance in noisy environments. To the best of our knowledge, our approach is the first to apply DFN in constructing the spectral mask for noise mitigation in audio classification.

3. Proposed method

In our proposed Dynamic Feature Enhancement (DFE) model, a filter generation network generates a dynamic filter for the dynamic filter layer. Then, the dynamic mask is multiplied with input T-F features and the results are fed to a pre-trained classifier. The key feature of our method that enables it to perform well in noisy condition is by using a dual stage filter generation.

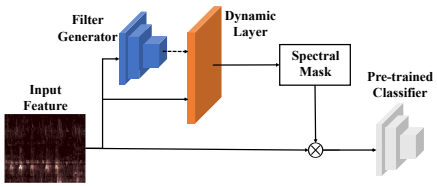


Figure 1: *The process of dynamic filter enhancement model*

3.1. Dynamic feature enhancement

Our proposed model has two main parts: a filter generator and a dynamic layer. The process of filter generation network is as follows:

$$w_t^* = \sigma(f(\cdots, (f(\sigma(f(x_t, w_1)), w_2)) \cdots, w_N), \quad (1)$$

where x_t presents t-th data instance in mini-batch and w_i's are learning parameters. The notation of N presents the number of layers. $f(\cdot)$ and σ represent each CNN layer and nonlinear activation respectively. CNN layers extract features by using x_t and w_i, then feature averaging is performed. This process makes output feature map as a fix length of a dynamic vector which is time independent. After obtaining the vector, a FC layer is implemented to generate the dynamic filter (w_t^*) which follows similar enhancement structure as in [13], and the architecture is as follows:

$$m_t = \sigma(f(\cdots, \sigma(f(\sigma(f(x_1, w_{t1}^*)), w_{t2}^*)) \cdots, w_{tN}^*), \quad (2)$$

where w_{ti}^*'s are the dynamic weights of the CNN. We apply dilated CNN layers in both the filter generator and the dynamic filter layer for extracting a high dimensional feature map, which is then reduced to one dimensional feature space by using a 1D CNN and sigmoid function. The reduced feature map (i.e. spectral mask) is multiplied with an input feature, thereby enhancing input features as shown by the following expression.

$$x_t^* = x_t \odot m_t, \quad (3)$$

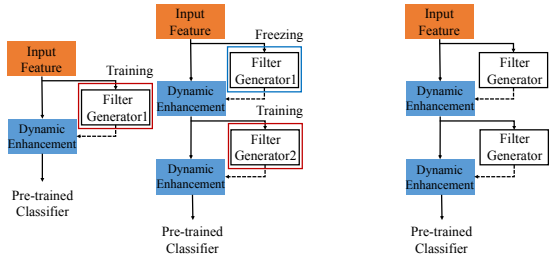
where notation $\odot$ indicates element-wise multiplication. Finally, the enhanced feature (x_t^*) is fed to the pre-trained classifier. For the model training, DFE architecture is trained by a cross entropy loss function which is the same loss function for training the classifier, thus the process would ensure consistency of classification.

3.2. Feature normalization

Due to the dynamic filtering proposed here, batch normalization [23] may not be effective in our model as the adaptation parameters for test phase should reduce a flexibility of the DFE process. Therefore, we follow Layer Normalization [24] framework in the dynamic layer. A zero mean unit variance normalization is performed per each input, and scale and bias parameters are obtained from the filter generation model.

3.3. Multi operation of DFE

Feature enhancement can be further improved by repeating the enhancement process via an updated mask as salient features may remain invariant in the enhancement process while noise can be further reduced. We consider some alternatives in this idea, and proceed with two following options.



(a) *DCL* (b) *DSL*

Figure 2: *Pipeline of multi operational DFE*

a. Dual staged Cascade Learning (DCL) is a pipeline which optimizes a shallow architecture by using a layer-wise learning process [17] done in two stages as shown in Figure 2(a). Once Filter Generator 1 is trained, the whole network cascades to the next stage training of Filter Generator 2 while the weights of Filter Generator 1 is frozen. Since only one filter generator is trained at a time, the learning process is more stable.
b. Dual staged Shared weight Learning (DSL) adopts an end-to-end iterative learning strategy [9], by training two filter generators with shared weights done simultaneously as shown in Figure 2(b). The training process is more efficient compared to DCL since it is an end-to-end model.

4. Experimental process

Database and segmentation: Urbansound8K contains 10 classes of sound events with each class containing 10 sets of data. We use first to 8th set as training datasets, 9th set as development dataset and 10th set as test dataset. All audio clips are uniformly converted to a mono type and at 16KHz sampling rate. Also, 4 second segmentation is performed by using zero padding. Then, spectrograms are generated in a sliding window fashion using Hamming window with 512 points for 256 steps. This process gives an output spectrogram of size 257×248 and a Mel spectrum of size 40×248 is obtained by using a filterbank process. After that, Min-Max normalization is performed. The normalized Mel spectrum is utilized as an input feature in our experiment. For noise dataset [19], 7 types of noise audios with 5 different SNRs are utilized for the test environment. The noise audios are artificially combined with only test dataset to evaluate the performance of unseen noise condition.
Computational setup: All our experiments are done by exploiting Tensorflow Deep learning package, and training is performed on RTX 2080-Ti GPU. In training process, we use 64 batch size, 100 epochs and ADAM optimizer [25] with 0.001 learning rate for parameters training.

4.1. Implementation detail

Filter generation model: The filter generation model consists of 3 dilated CNN layers and 10 FC layers. Each FC layer generates different parameters of the dynamic filter layers and the output dimension of FC is the dynamic filer size. At the end of every layer, batch normalization, Max Pooling and ReLU activation [26] are performed. All the CNN layers have a 3×3 kernel and stride of 1×1. The dimension of kernel is sequentially designed by [12, 72, 256] manner and the dilation size is sequentially designed by [6, 4, 1] manner. Except the first Max Pooling layer, which has 3×3 kernel and stride of 2×2, all the Max Pooling layers have a 3x4 kernel and stride of 2×3. After that, feature averaging is performed to obtain 256 dimensional feature vector. First through fifth FC layers (FC 1 to FC 5) generate CNN parameters in the dynamic filter layer while sixth through tenth FC layers (FC 6 to FC 10) generate scale and bias factors for the feature normalization in the dynamic filter layer.
Dynamic filter layer: The dynamic filter layer consists of 5 dilated CNN layers. The shape of dynamic filters is converted from a vector which is an output feature map of the neural network to a two dimensional tensor for CNN implementation. The output of FC 1 to FC 5 in the filter generation model are assigned to weights of CNN layers in the dynamic filter layer and the output of FC 6 to FC 10 in the filter generation model are assigned to weights of an affine transform for the feature normalization respectively. At the end of every layer, the feature normalization and a ReLU activation are performed. At the last layer of the dynamic filter, a sigmoid activation [21] is performed. All the CNN layers have a 3×3 kernel and stride of 1×1. The dimension of kernel is sequentially designed by [24, 96, 96, 96, 1] manner and the dilation size is sequentially designed by [1, 2, 4, 8, 1] manner. The dimension of an affine transform follows the dimension size of the CNN kernel.

4.2. Baseline method

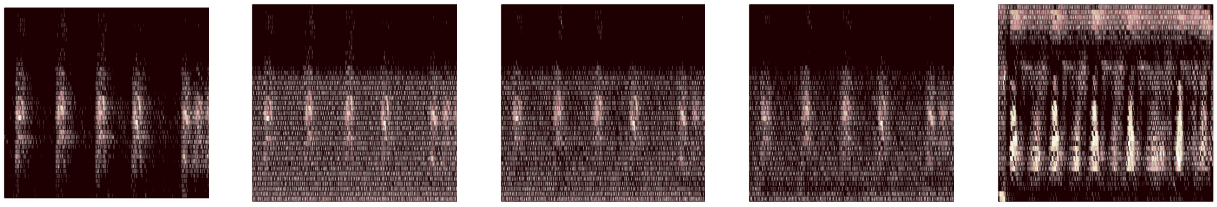
For baseline models for comparison to our proposed model, we employ CNN-FC and Statistic Feature Enhancement (SFE).
CNN-FC: CNN-FC is composed of 5 CNN layers with 3×3 size of filters, 3 times of max pooling and FC layer with softmax normalization for finial decision. In CNN layer, the dimension of CNN kernels are designed by [12, 36, 72, 96, 96] manner. In max pooling layer, excepting the 1th layer which has 3×3 kernel and stride of 2×2, 3×4 kernel and stride of 2×3 filter is utilized into the 3 and 5th layer. This structure is considered as a pre-trained model for the other feature enhancement methods.
Statistic Feature Enhancement: SFE is a feature enhancement method which is based on [13]. The enhancement model follows the same structure as described in dynamic filter layer.

4.3. Result and discussion

The experiments are designed to demonstrate a performance of our proposed method in both clean and noisy environments. The classifier is first trained and validated by the development dataset. The classifier weights are then frozen for the rest of the training process. Figure 3 shows the Mel spectrum features of (a) clean input, (b) noisy input (clean + 5dB pink noise), (c) SFE enhanced features, (d) DCL enhanced features, and (e) DSL enhanced features respectively. When comparing SFE and DCL features, DCL method shows better noise reduction effect. Although the DSL enhanced features seem to show different Mel spectrum structures compared to the clean input, the method results in consistently better classification performances in most cases as shown in Tables 1 and 2. It can be interpreted that

Table 1: *Performance comparison in classification rate among various implementations of dynamic filter layer and baseline models*

	CNN-FC	SFE	DFE	DCL	DSL
Dev	76.47	**78.79**	76.34	76.34	78.55
Test	73.84	74.79	76.94	77.65	**78.85**
Difference	2.63	4.0	0.6	1.29	**0.3**



(a) *Clean Feature* (b) *Noisy Feature* (c) *SFE Feature* (d) *DCL Feature* (e) *DSL Feature*

Figure 3: *Comparison of enhanced feature*

Table 2: *Performance comparison in classification rate over various noisy environments*

SNR	Method	Noise						
		Babble	**F16**	**Factory1**	**Factory2**	**Leopard**	**Pink**	**Destroyer**
20	**CNN-FC**	74.07	73.48	73	73.48	73.36	73.72	73.84
	SFE	75.03	74.55	74.79	75.03	74.79	75.15	74.79
	DFE	76.94	76.58	77.18	76.94	77.54	76.82	77.54
	DCL	76.94	76.82	76.94	77.9	78.49	77.42	77.66
	DSL	**79.33**	**78.73**	**78.97**	**79.09**	**78.97**	**79.09**	**79.09**
15	**CNN-FC**	73.72	73.84	72.64	72.64	73.72	72.64	73.95
	SFE	74.07	74.43	73.95	73.95	74.79	74.07	74.12
	DFE	76.7	77.3	76.46	76.46	76.7	76.7	75.87
	DCL	76.58	77.3	76.82	77.3	77.18	77.18	76.58
	DSL	**78.97**	**78.85**	**78.97**	**78.38**	**79.09**	**79.57**	**79.45**
10	**CNN-FC**	71.45	73.6	71.45	72.28	72.88	69.89	71.21
	SFE	74.07	73.84	73.84	74.31	73.48	73.48	73.72
	DFE	75.27	75.75	75.15	77.18	75.87	75.15	75.51
	DCL	75.39	76.34	74.91	76.58	76.34	75.51	76.11
	DSL	**79.33**	**77.54**	**77.9**	**79.21**	**78.02**	**78.38**	**79.21**
5	**CNN-FC**	71.32	70.61	68.82	70.37	72.88	63.32	66.67
	SFE	73.36	70.97	68.70	71.68	72.76	64.40	67.26
	DFE	70.97	**73.3**	69.18	73	75.75	67.26	70.49
	DCL	71.21	72.76	68.58	73.24	76.22	67.98	70.49
	DSL	**76.22**	72.4	**73.12**	**78.26**	**78.73**	**75.03**	**76.7**
0	**CNN-FC**	67.98	60.81	61.41	69.41	70.25	55.44	61.29
	SFE	**68.46**	62.72	61.17	69.30	69.41	55.20	60.33
	DFE	68.1	**64.52**	64.52	70.13	71.18	**62.27**	64.16
	DCL	**68.46**	**64.52**	64.76	70.01	72.16	61.89	64.16
	DSL	68.45	64.16	**65.35**	**72.52**	**72.88**	60.93	**66.43**

the method effectively highlights salient features crucial in the classification process better than the other methods.

Table 1 compares the performance among the learning methods based on the developmental and the test datasets in clean environment. Our proposed methods show improved results in test conditions over the other state-of-the-art methods. Additionally, our models show some signs of generalization capabilities considering that both models performed better for the testset. Table 2 presents the performance over various noise types and SNR levels. Except for few cases, it shows that our represented features are more robust to unseen noise over the state-of-the-art methods. DSL shows the overall best improvement among the methods considered. Since DCL is trained by adding an additional training stage to DFE, the added step by the cascade training can be attributed to its improved performance over DFE. In summary, the DFN approach for feature enhancement we propose here shows robustness to domain change due to unseen noise and also exhibits generalization properties. In addition, the dual stage operations are shown to achieve performance improvement over the single DFE methods.

5. Conclusions

The performance of most of the current audio enhancement methods falter when input streams contain unseen noise. These are primarily due to fixed parameters in their enhancement process, and we addressed the issue by proposing a dynamic filter learning approach. The proposed method demonstrated that it not only improves classification performance but also exhibits less sensitivity to domain change. A key insight shown in our effort was that the dual stage learning improves in highlighting salient features for event classification in noisy condition.

6. Acknowledgements

This work was supported by the Korea Environmental Industry & Technology Institute (KEITI) through the Public Technology Program based on environmental policy funded by the Korean Ministry of Environment (MOE; 2017000210001), and the contribution of David Han was supported by the US Army Research Laboratory.

7. References

[1] P. Maijala, Z. Shuyang, T. Heittola, and T. Virtanen, "Environmental noise monitoring using source classification in sensors," *Applied Acoustics*, vol. 129, pp. 258–267, 2018.

[2] N. Lin, H. Sun, and X.-P. Zhang, "Overlapping animal sound classification using sparse representation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2156–2160.

[3] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 1, pp. 279–288, 2015.

[4] P. Scalart *et al.*, "Speech enhancement based on a priori signal to noise estimation," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2. IEEE, 1996, pp. 629–632.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[6] X. Lu, M. Unoki, S. Matsuda, C. Hori, and H. Kashioka, "Controlling tradeoff between approximation accuracy and complexity of a smooth function in a reproducing kernel hilbert space for noise reduction," *IEEE transactions on signal processing*, vol. 61, no. 3, pp. 601–610, 2012.

[7] M. Sun, X. Zhang, T. F. Zheng *et al.*, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 93–104, 2015.

[8] A. Badi, S. Park, D. K. Han, and H. Ko, "Correlation distance skip connection denoising autoencoder (cdsk-dae) for speech feature enhancement," *Applied Acoustics*, vol. 163, p. 107213, 2020.

[9] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep griffin–lim iteration," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 61–65.

[10] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[11] C. Y. Kao, S. Park, A. Badi, D. K. Han, and H. Ko, "Orthogonal gradient penalty for fast training of wasserstein gan based multi-task autoencoder toward robust speech recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 103, no. 5, pp. 1195–1198, 2020.

[12] B. Li and K. C. Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 8, pp. 1296–1305, 2014.

[13] S. Shon, H. Tang, and J. Glass, "Voiceid loss: Speech enhancement for speaker verification," *arXiv preprint arXiv:1904.03601*, 2019.

[14] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 667–675.

[15] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and H. Li, "On time-frequency mask estimation for mvdr beamforming with application in robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3246–3250.

[16] Y.-H. Tu, J. Du, L. Sun, F. Ma, H.-K. Wang, J.-D. Chen, and C.-H. Lee, "An iterative mask estimation approach to deep learning based multi-channel speech recognition," *Speech Communication*, vol. 106, pp. 31–43, 2019.

[17] E. S. Marquez, J. S. Hare, and M. Niranjan, "Deep cascade learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 11, pp. 5475–5485, 2018.

[18] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[19] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[20] R. Lyon, "A computational model of binaural localization and separation," in *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8. IEEE, 1983, pp. 1148–1151.

[21] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[22] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen, "Classification-driven dynamic image enhancement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4033–4041.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

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

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.