A Computation-Efficient Neural Network for VAD using Multi-Channel Feature

Runze Wang[1], Iman Moazzen[2], Wei-Ping Zhu[1]
[1]*Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada*
[2]*Castofly Technologies, Vancouver, Canada*
runze.wang@mail.concordia.ca, imanmoaz@gmail.com, weiping@ece.concordia.ca

***Abstract*—Voice activity detection serves as an essential pre-processor in modern speech processing systems. It classifies audio segments into speech and nonspeech. Many state-of-the-art methods have been proposed to increase the detection accuracy. However, there are still significant limitations to retaining high performance while keeping low computation complexity, especially in handling unseen noises. This paper proposes a computation-efficient neural network using a multi-channel audio feature. The audio feature is contextual-aware with positional information and is represented in a three-channel way, similar to RGB pictures, which enables convolutional kernels to capture more information simultaneously. Meanwhile, we introduce channel attention inverted blocks to build a computation-efficient neural network. Our proposed method shows superior performance with extremely few floating point operations as compared with baseline methods.**

***Index Terms*—voice activity detection, channel attention, computation-efficient, deep neural network**

I. Introduction

Voice Activity Detection (VAD) is an essential component of speech processing systems. Its goal is to determine the presence or absence of human speech in audio segments, which can be considered a binary classification problem [1]. Despite decades of research and development [2]–[5], two key challenges remain. The first is the bad performance in dealing with unseen noises at severe signal-to-noise ratio (SNR); the second is numerous calculations that deep learning-based methods will have to execute.

Early VAD methods detect abrupt changes in speech energy level, zero-crossing rate [6] and frequency domain properties such as spectral or cepstral [7]. However, those classifiers only work well when speech power is greater than noise. And their decision-making mechanism is based on predefined thresholds of selected features, which has significant limitations when handling real-world noises with varying SNR levels. As a result, these algorithms cannot maintain stable performance under different circumstances. The statistical signal processing then becomes a trend of VAD research. The statistical model-based method [8] and a machine-learning technique-assisted statistical model [9] further improved the detection accuracy. They can achieve reasonably good performance when handling with stationary noise at high SNR levels; meanwhile, keep low consumption of computation resources [10]. But non-stationary or burst noises from real-world scenarios will drastically degrade the performance.

As computational and data resources have grown, deep learning has been extensively exploited in advanced speech and language processing [11]. It uses large-scale multiple domain data to train a sophisticated model that can map real-world data to desired targets and shows superior generalization ability than earlier statistical models in real-world applications. Recently, many authors conducted deep learning-based VAD research by making use of high-dimension features such as spectrogram [12], Mel-Frequency Cepstral Coefficient (MFCC) [13], Log-Mel spectrograms [14]. The Multi-Resolution Cochleagram (MRCG) feature [15] was first proposed for classification-based speech separation tasks. It delivered the best results among many other features. The authors of [10], [16] and [17] have also demonstrated the superior performance of the MRCG feature with its dynamic information on the VAD task.

Many neural network architectures have been exploited to perform VAD. The LSTM-RNN [18] was employed for VAD due to its ability to encode long short-term contextual information from a sequence of frame features. However, audio usually contains thousands of frames, making LSTM-RNN much slower than other models due to its sequential processing characteristic. The boosted DNN [10] utilized multiple frames features and labels as input and output of DNN in training phase. It uses fully-connected layers with hundreds of neurons in each layer, making the neural network unable to explore deeper hidden features because the computation cost will increase dramatically as more layers added. In [12], a combination of CNN and GRU neural networks was shown to have achieved good performance on challenge tests. The temporal information is still learned by a sequence model.

In previous studies, MRCGs and their dynamic information were arranged in one-dimensional or single-channel two-dimensional formats, resulting in a weak connection between MRCGs and dynamic information. In RGB pictures, every pixel is a combination of different intensities of red, green and blue. We are inspired to propose a multi-channel representation that makes MRCG features provide corresponding dynamic information in the same position to enhance the connectivity. In addition, we add order information to each channel of features explicitly by positional encoding [19]. To overcome the problem of high computation complexity and achieve channel attention, we exploit the MobileNetV2 [20] framework in conjunction with the squeeze-and-extraction modules [21].

(a) 1-D Feature (b) Sigle-Channel 2-D Feature (c) Multi-Channel 2-D Feature and Positional Mask

Fig. 1: Feature Arrangement

II. Proposed Model

The proposed computation-efficient voice activity detector is a frame-level binary classifier that incorporates the contextual information of the current frame to determine whether the frame contains human speech.

A. Feature Extraction

First, the original audio is segmented into frames by a 20 ms long sliding window with 10 ms overlap between every two frames. The audio consists of N frames, each with acoustic feature and ground truth label denoted as x_n and y_n respectively, $n = 1, 2, \ldots, N$. The acoustic feature of each frame consists of Multi-Resolution Cochleagrams (MRCGs) [15], and their first and second derivatives, Δ and $\Delta\Delta$. The Δ and $\Delta\Delta$ are employed to capture temporal dynamics information. The dimension of these three feature components is the same and denoted as d. The acoustic feature vector and the ground truth are written as

$$x_n = [MRCG_1, \ldots, MRCG_d, \Delta_1, \ldots, \Delta_d, \Delta\Delta_1, \ldots, \Delta\Delta_d] \quad (1)$$

$$y_n = \begin{cases} 0, & nonspeech \\ 1, & speech \end{cases} \quad (2)$$

To exploit the contextual information, we concatenate j frames before the current frame x_n, and $j-1$ frames after x_n to form a context-aware feature X_n as given below

$$X_n = [x_{n-j}, \ldots, x_{n-1}, x_n, x_{n+1}, \ldots, x_{n+j-1}] \quad (3)$$

where j is a tunable parameter that will be discussed in section V. Next, the above context-aware feature vector will be transformed into a two-dimensional format by vertically stacking the features of different frames. Finally, the proposed feature arrangement separately extracts Δ and $\Delta\Delta$ of all frames as two channels and then stacked with MRCG. The feature arrangement process is shown in Fig. 1.

B. Positional Mask

Despite the addition of neighbouring frames' information, the sequence order is still ignored. The positional mask utilizes Positional Encoding (PE) [19] to assign a unique encoding value for every position in the feature. It takes the advantage of no training parameters required, as opposed to RNNs. Since Δ and $\Delta\Delta$ reflect temporal dynamics of MRCG at the corresponding position, all three channels in the proposed feature share the same positional mask. Also, the size of the positional mask remains the same as the proposed feature, as illustrated in Fig. 1c. The positional mask values are calculated as

$$PM_{(n,2i)} = sin(n/10000^{2i/d}) \quad (4)$$
$$PM_{(n,2i+1)} = cos(n/10000^{2i/d})$$

where n represents frame index and i is the dimension index.

C. Model Architecture

The structure of our proposed computation-efficient neural network for VAD is depicted in Fig. 2. The backbone of the neural network is MobileNetV2 [20] that is usually used in computer vision tasks, such as image classification, object detection and image segmentation. It retains high performance while significantly decreasing the computation complexity and memory consumption.

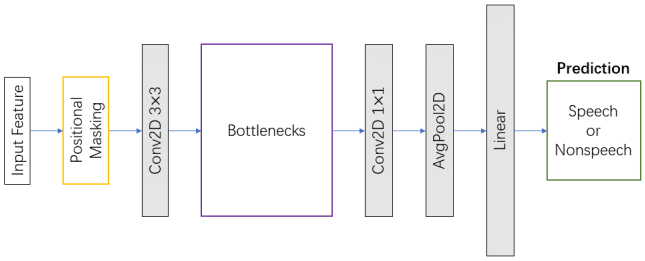
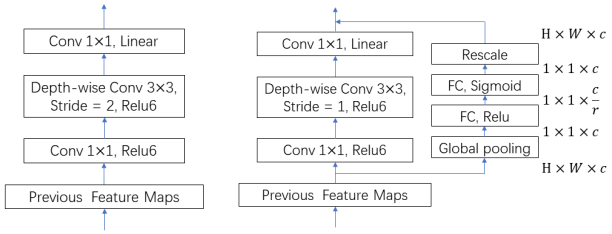


Fig. 2: Model Overview



(a) Inverted Block (b) Channel-Attention Inverted Block

Fig. 3: Inverted Blocks Structure

The proposed multi-channel 2-D feature is input to the model. The positional mask is then applied to the input feature to add sequence order information. Subsequently, the first

convolutional layer employs 32 filters of the kernel size $(3, 3)$ with stride 2. It extracts local details of the input over time and frequency domain and yields output feature maps with half-size of the input feature.

The middle part of this model is a sequence of bottleneck layers. Each bottleneck contains m concatenated inverted blocks. The structure of inverted blocks is described in Fig. 3. If the parameter s of the bottleneck layer is 2, the first inverted block has stride 2 in its depth-wise convolution layer, rest $m-1$ inverted blocks have stride 1 and are added with channel attention. Here we call the inverted blocks having channel attention as channel-attention inverted blocks (CAIBs). One inverted block consists of a point-wise convolution layer, a depth-wise convolution layer and a linear point-wise convolution layer. Except for the linear point-wise convolution layer, every convolution layer is followed by batch normalization and non-linear activation ReLU6. The first point-wise convolution layer increases the number of channels by t times, where t is the expansion ratio. After depth-wise convolution, the last linear point-wise convolution layer reduces the number of channels to c. CAIB has the same number of input and output channels. CAIB replaces the shortcut connection of inverted residual block [20] by a squeeze-and-extraction (SE) module [21] to add channel attention. This SE module first conducts global pooling to input feature maps to get a result of $(1, 1, c)$ shape, then two fully connected (FC) layers reduce and increase the number of channels by reduction rate r times, respectively. During this process, two FC layers learn the importance of different channels. Finally, a fully connected layer is employed as the classifier after the average pooling along height and width dimensions. The entire model setting is stated in Table I and the lite version is shown in Table II.

TABLE I: Proposed Model Setting

Input	Operator	t	c	m	s
32×128×3	conv2d	-	32	1	2
16×64×32	bottleneck	1	16	1	1
16×64×16	bottleneck	6	24	2	2
8×32×24	bottleneck	6	32	3	2
4×16×32	bottleneck	6	64	4	2
2×8×64	bottleneck	6	96	3	1
2×8×96	bottleneck	6	128	1	1
2×8×128	conv2d 1×1	-	1280	1	1
2×8×1280	avgpool	-	-	1	-
1×1×1280	linear	-	2	1	-

III. Experiments

A. Experimental Setup

a) Dataset: The clean speech corpus is taken from TIMIT dataset [22], which provides ground truth labels at the word level. From the observation of labels, speech segments take up much higher proportions than nonspeech segments, which may lead to the problem of class imbalance. Therefore,

TABLE II: Proposed Lite Model Setting

Input	Operator	t	c	m	s
32×128×3	conv2d	-	32	1	2
16×64×32	bottleneck	1	16	1	1
16×64×16	bottleneck	6	24	2	2
8×32×24	bottleneck	6	32	3	2
4×16×32	bottleneck	6	64	3	2
2×8×64	bottleneck	6	96	1	1
2×8×96	conv2d 1×1	-	960	1	1
2×8×960	avgpool	-	-	1	-
1×1×960	linear	-	2	1	-

we injected 1-second-long blank segments at the beginning and end of each audio. Correspondingly, ground truth labels were also modified to take into account these blank segments. The training set of TIMIT was augmented with fifteen types of additive noises from NOISEX-92 [23] at five levels of SNR, i.e., -10, -5, 0, 5, 10 dB. We used 90% of the augmented training set for training and the rest 10% for validation.

In the test stage, we use TIMIT test dataset as clean speech, and AURORA [24] noise set containing eight types of unseen noises for data augmentation. The test SNR has five levels ranging from -10 to 10 dB at increment of 5 dB.

b) Baseline Models: For performance comparison purposes, several supervised and unsupervised methods are used as the baseline; they are CNN, 2-D CNN, CRNN, 2-D CRNN [12], rVAD [25]. Both baseline approaches and proposed method use MRCG, Δ and $\Delta\Delta$ as training feature, but they are arranged in different ways to show the effectiveness of incorporating contextual information and fusing dynamic information. Specifically, CNN and CRNN use 1-D feature as shown in Fig. 1a, 2D-CNN and 2D-CRNN use 2D-feature as shown in Fig. 1b and the proposed method use multi-channel 2-D feature depicted in Fig. 1c. For CRNN and 2-D CRNN, we follow the model architecture described in [12]. The structures of CNN and 2-D CNN follow the convolutional part of CRNN and 2-D CRNN, respectively. The rVAD is a state-of-the-art unsupervised statistical method for VAD.

c) Training Setting: All audio segments, including clean speech and noise, are resampled to 16kHz. We pre-extract each frame's feature, and the dimension parameter d is set as 128 for each component. The contextual range parameter j is set to 16. The z-score normalization is only performed on the training set, and the test set remains unprocessed. All baseline and the proposed models are trained with SGD optimizer, using momentum 0.9 and weight decay 0.0001, and cross-entropy is selected as the loss function. The number of total training epochs is 50. The learning rate was adjusted dynamically at the training phase [26]. Specifically, it linearly increases from 0 to the initial learning rate of 0.1 within 10 warm-up epochs. It decreases to the final learning rate of 0.001 in another 30 epochs using cosine decay strategy [27]. The learning rate remains at 0.001 for last 10 epochs. The batch size for training and validation is 64. We employ gradient

clipping with a maximum L2-norm of 5 to avoid gradient explosion. The weights of all convolutional layers and linear layers are initialized by Kaiming normal initialization [28], while the initial weights of batch normalization layers are set as 1. The initial biases of the last linear layer are set to 0.

d) Evaluation Metric: The Area Under the Curve (AUC) is employed as quantitative performance evaluation metric for the proposed and baseline methods.

B. Results and discussion

Table III shows the AUC comparison results of different methods at various SNR levels. For every SNR level, eight types of noises are tested and the results are then averaged to get the final AUC value of each method. As seen from Table III, the proposed method and its lite version outperform other methods over 4 SNR levels. In Fig. 4, we select two noise scenarios, restaurant and street for indoor and outdoor environments, to show more details. The overall performance of all methods in the restaurant noise is much worse than in the street noise, as their worst AUCs at -10 dB SNR are about 55% and 70%, respectively. Because in a restaurant environment, noise is more concentrated and from the babble sound. Even so, our method outperforms other methods at the whole range of SNRs.

TABLE III: AUC(%) Comparison

SNR	rVAD	CNN	2D-CNN	CRNN	2D-CRNN	Proposed	Proposed (lite)
10	93.64	91.55	90.06	93.05	91.62	96.44	**96.81**
5	91.97	88.57	88.98	91.01	88.87	**95.00**	93.89
0	84.22	84.03	86.57	85.43	83.73	**91.53**	87.24
-5	75.05	81.58	84.80	80.81	79.71	**86.47**	79.75
-10	68.83	79.41	**83.18**	77.69	77.83	82.49	74.03

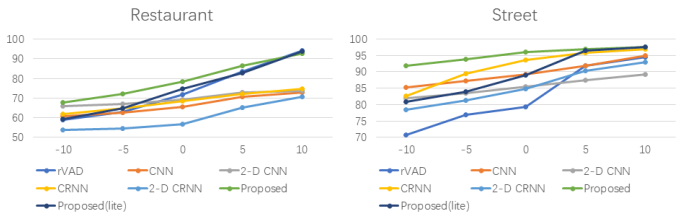


Fig. 4: AUC(%) Under Difference Scenarios

To investigate how the number of neighbour frames will influence the model performance, we conducted four experiments with 40, 32, 24, 16 neighbour frames with the current frame included. The parameter j is set as 20, 16, 12, 8, respectively. We tested the four cases with eight types of noise in the test set at 10 dB SNR and then averaged them to get the final AUC result. Table IV describes test results that indicate the positive correlation between the number of neighbor frames and FLoating Point Operations (FLOPs). Here, we used a public package in [29] to count the number of FLOPs. However, the performance does not follow the same rule, where 16 frames even can be comparable to 40 frames. But it shows a great improvement after changing from 16 to 32 frames. Because the feature map size is always reduced by half and it will lead to information loss when the number is not the power of 2. We finally chose 32 neighbour frames as a performance-FLOPs trade-off solution.

TABLE IV: Discussion on Different Number of Neighbour Frames

# of neighbor frames	40	32	24	16
FLOPs(M)	66	48	42	24
AUC(%)	93.22	96.44	90.80	93.19

Table V shows performance and FLOPs comparison between baseline deep learning methods and our proposed method. We averaged AUC results in Table III for each method to get an overall AUC performance. From the observation of test results, our proposed method has a more robust performance than other baseline methods. In addition, the lite version of our proposed method has better performance even if it requires 21 times fewer FLOPs than the CRNN methods. Also, the best performance of 90.39% AUC is achieved by our method. The results in the table also demonstrated that our methods require moderate computation while leading to better performance.

TABLE V: Computation Cost and Performance Comparison Among Deep Learning-Based Models

Model # of param	CNN (329K)	2D-CNN (399K)	CRNN (935K)	2D-CRNN (732K)	Proposed (854K)	Proposed(lite) (349K)
FLOPs(M)	294	120	642	324	48	**30**
AUC(%)	85.03	86.72	85.60	84.35	**90.39**	86.34

IV. Conclusion

In this paper, we have proposed a new approach of feature arrangement that allows convolutional kernels to capture and fuse multi-channel information. We have also employed channel attention to help the neural network to learn the importance of different feature maps. Experiments demonstrated that our proposed method provides a better performance in dealing with a variety of unseen noises. Meanwhile, for some specific noises, it yields a significant performance improvement as compared to reference methods. We have also shown that our method requires a lower computational cost.

References

[1] A. Sholokhov, M. Sahidullah, and T. Kinnunen, "Semi-supervised speech activity detection with an application to automatic speaker verification," Computer Speech & Language, vol. 47, pp. 132–156, 2018.

[2] 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.

[3] D. Malah, R. V. Cox, and A. J. Accardi, "Tracking speech presence uncertainty to improve speech enhancement in non-stationary noise environments," in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP), vol. 2. IEEE, 1999, pp. 789–792.

[4] T. Gerkmann, C. Breithaupt, and R. Martin, "Improved a posterior speech presence probability estimation based on a likelihood ratio with fixed priors," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 5, pp. 910–919, 2008.

[5] R. Martin and I. Cohen, “Single-channel speech presence probability estimation and noise tracking,” in Audio Source Separation and Speech Enhancement. Wiley, 2018, ch. 6, pp. 87–106.

[6] F. Xie S. Van Gerven, “A comparative study of speech detection methods.,” in Eurospeech, 1997, vol. 97.

[7] J. Haigh and J. Mason, “Robust voice activity detection using cepstral features,” in IEEE Region 10 International Conference on Computers, Communications and Automation, vol. 3, 1993, pp. 321–324.

[8] J. Sohn, N. Kim, and W. Sung, “A statistical model-based voice activity detection,” IEEE Signal Processing Letters, 1999.

[9] J. W. Shin, J.-H. Chang, and N. S. Kim, “Voice activity detection based on statistical models and machine learning approaches,” Comput. Speech Lang., vol. 24, no. 3, pp. 515–530, 2010.

[10] X.-L. Zhang and D. Wang, “Boosting contextual information for deep neural network based voice activity detection,” IEEE/ACM Trans Audio, Speech, and Language Processing, vol. 24, no. 2, pp. 252–264, 2016.

[11] T. Dutoit, Martinn-Vide Carlos, and G. Pironkov, Statistical language and Speech Processing 6th International Conference, SLSP 2018, Mons, Belgium, October 15–16, 2018, Proceedings. Cham: Springer International Publishing, 2018.

[12] A. Vafeiadis et al., “Two-dimensional convolutional recurrent neural networks for speech activity detection,” in Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, Sep. 2019, pp. 2045–2049.

[13] A. Ivry, B. Berdugo and I. Cohen, ”Voice activity detection for transient noisy environment based on diffusion nets”, IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 254-264, 2019.

[14] Y. Lee, J. Min, D. K. Han and H. Ko, ”Spectro-temporal attention-based voice activity detection”, IEEE Signal Process. Lett., vol. 27, pp. 131-135, 2020.

[15] J. Chen, Y. Wang, and D. Wang, “A feature study for classification-based speech separation at low signal-to-noise ratios,” IEEE/ACM TASLP, vol. 22, no. 12, pp. 1993–2002, 2014.

[16] J. Chen, Y. Wang, and D. Wang, “A feature study for classification-based speech separation at low signal-to-noise ratios,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 22, no. 12, pp. 1993–2002, Dec. 2014.

[17] Y. R. Jo, Y. K. Moon, W. I. Cho, en G. S. Jo, “Self-Attentive VAD: Context-Aware Detection of Voice from Noise”, in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, bll 6808–6812.

[18] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, “Real-life voice activity detection with LSTM recurrent neural networks and an application to Hollywood movies,” inProc. Int. Conf. Acoust., Speech, Signal Process., 2013, pp. 483–487.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems 30, 2017, pp. 5998–6008.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4510–4520

[21] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7132–7141.

[22] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, “TIMIT acoustic-phonetic continuous speech corpus LDC93S1,” Philadelphia, PA, USA: Linguistic Data Consortium, 1993.

[23] 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 Commun., vol. 12, no. 3, pp. 247–251, Jul. 1993.

[24] H.-G. Hirsch and D. Pearce, “The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in Proc. Autom. Speech Recognit.: Challenges Millenium ISCA Tut. Res. Workshop, 2000, pp. 181–188.

[25] Z.-H. Tan, A. k. Sarkar, and N. Dehak, “rVAD: An unsupervised segmentbased robust voice activity detection method,” Comput. Speech Lang., vol. 59, pp. 1–21, Jan. 2020.

[26] J. Xie, T. He, Z. Zhang, Z. Zhang, Z. Zhang, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 1097–1105.

[27] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. CoRR, abs/1608.03983, 2016.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1026–1034.

[29] V. Sovrasov, “flops-counter.pytorch”, Github Repository, 2019. [Online], Available: https://github.com/sovrasov/flops-counter.pytorch