IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 November 2022, accepted 9 December 2022, date of publication 12 December 2022,
date of current version 20 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228744

RESEARCH ARTICLE

A Compact CNN-Based Speech Enhancement With Adaptive Filter Design Using Gabor Function and Region-Aware Convolution

SALINNA ABDULLAH, (Graduate Student Member, IEEE), MAJID ZAMANI, (Member, IEEE), AND ANDREAS DEMOSTHENOUS, (Fellow, IEEE)
Department of Electronic and Electrical Engineering, University College London (UCL), WC1E 7JE London, U.K.

Corresponding author: Salinna Abdullah (salinna.abdullah.13@ucl.ac.uk)

This work was supported by a Engineering and Physical Sciences Research Council (EPSRC) Industrial Strategy Studentship to Salinna Abdullah under Grant EP/R512400/1.

ABSTRACT Speech enhancement (SE) is used in many applications, such as hearing devices, to improve speech intelligibility and quality. Convolutional neural network-based (CNN-based) SE algorithms in literature often employ generic convolutional filters that are not optimized for SE applications. This paper presents a CNN-based SE algorithm with an adaptive filter design (named 'CNN-AFD') using Gabor function and region-aware convolution. The proposed algorithm incorporates fixed Gabor functions into convolutional filters to model human auditory processing for improved denoising performance. The feature maps obtained from the Gabor-incorporated convolutional layers serve as learnable guided masks (tuned at backpropagation) for generating adaptive custom region-aware filters. The custom filters extract features from speech regions (i.e., 'region-aware') while maintaining translation-invariance. To reduce the high cost of inference of the CNN, skip convolution and activation analysis-wise pruning are explored. Employing skip convolution allowed the training time per epoch to be reduced by close to 40%. Pruning of neurons with high numbers of zero activations complements skip convolution and significantly reduces model parameters by more than 30%. The proposed CNN-AFD outperformed all four CNN-based SE baseline algorithms (i.e., a CNN-based SE employing generic filters, a CNN-based SE without region-aware convolution, a CNN-based SE trained with complex spectrograms and a CNN-based SE processing in the time-domain) with an average of 0.95, 1.82 and 0.82 in short-time objective intelligibility (STOI), perceptual evaluation of speech quality (PESQ) and logarithmic spectral distance (LSD) scores, respectively, when tasked to denoise speech contaminated with NOISEX-92 noises at -5, 0 and 5 dB signal-to-noise ratios (SNRs).

INDEX TERMS Adaptive filter design, activation analysis, convolutional neural network, Gabor filter, pruning, skip convolution, speech enhancement.

I. INTRODUCTION

Speech enhancement (SE) is the task of eliminating or attenuating additive noise from speech signals, commonly used in hearing devices to improve speech intelligibility and quality in noisy environments. In recent years, the adoption of deep-learning approaches for supervised SE tasks has become mainstream since they demonstrated exceptionally improved denoising performance over their non-deep learning-based counterparts (e.g., spectral subtraction [1] and statistical model-based SE [2]). Fully-connected deep neural network (DNN) in particular, has been widely adopted for nonlinear mapping between noisy speech features and enhanced speech in supervised SE (e.g., [3] and [4]). This has been increasingly replaced by convolutional neural networks (CNNs) (e.g., [5] and [6]) which allow greater flexibility in their architectural design and can more effectively characterize local temporal-spectral structures of speech signals.

CNN-based SE algorithms described in literature rarely explore the impact of convolutional filter designs on SE

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

performance. Furthermore, to achieve high accuracy, deeper and wider CNNs are often proposed, resulting in an increased number of parameters and a high cost of inference operations. Complex CNNs are unsuitable for mobile devices, edge computing, embedded systems, and real-time applications, where limitations are imposed on memory, power, and computation speed. For a CNN-based SE algorithm to be suitably implemented in memory, power, and computation speed-constrained systems such as hearing devices, significant model compression and acceleration are required. Approaches for CNN compression can be divided into three categories: network pruning [7], precision reduction [8] and design of compact network architectures [9].

In this paper, a CNN-based SE algorithm with adaptive Gabor-based filter design, named 'CNN-AFD', for more optimal speech denoising performance is proposed. The proposed CNN-AFD investigates the benefit of using customized convolutional filters for SE tasks. In the proposed SE algorithm, Gabor functions are incorporated into the convolutional filters of the network to extract temporal-spectral features based on prior knowledge of auditory models. The extracted features are then used as a learnable guided mask, where the spatial dimension is segmented into several regions to realize high-order separation of the speech and non-speech patterns of the input frames. A filter generator is used to generate custom filters for region-aware 2D convolution execution on the speech segments such that dynamic and optimal filtering is obtained for speech denoising. This leads not only to improved denoising performance but also a significant reduction of the number of parameters to operate on compared to a standard local convolution. This paper additionally employs skip convolution and a variation of pruning to improve the efficiency of the proposed CNN-AFD with minimal compromise on the SE performance. With skip convolution, structured sparsity is enforced in feature maps. This reduces computation and redundant operations while preserving feature representational capability. Pruning is performed via performing activation analysis, where neurons and their connections are removed if they have high percentages of zero activations. The contributions of the paper are summarized as follows:

- Utilization of Gabor incorporated CNN for realizing human auditory-inspired adaptive filter design in speech enhancement tasks;
- Region-aware training and processing on the identified temporal-spectral speech segments by incorporating dynamic and customized filtering to the standard Gabor-based CNN processing pipeline; and
- Skip convolution and a variation of pruning based on activation analysis are used for network compression. The proposed CNN-AFD achieves excellent speech denoising performance compared to its benchmarks including other recent state-of-the-art CNN-based SE algorithms with reduced computation cost.

The rest of this paper is organized as follows. Section II discusses related work employing CNN for SE. Furthermore, the literature utilizing CNN architectures with bespoke filter designs is presented. Section III elaborates the proposed CNN-AFD with adaptive Gabor-based filter design by describing its major functional units in detail. Section IV describes the network compression techniques employed to reduce computational cost and model size, namely skip convolution and activation analysis-based pruning. The datasets, evaluation metrics and baseline models used for assessing the performance of the proposed system are described in Section V. The results are presented in Section VI. This is followed by their discussion in Section VII. Finally, the paper ends with the concluding remarks in Section VIII.

II. RELATED WORK

Table 1 provides a summary of the speech enhancement algorithms mentioned in this section by including a condensed description of the algorithm, and its methods, key results, and limitations. Recent CNN-based SE algorithms have been described in [5], [6], [10], [11], [12], and [13]. In [5], two signal-to-noise ratio-aware (SNR-aware) CNN models are proposed for SE. The first algorithm employs a multi-task framework in which restoring the clean speech given a noisy speech input is formulated as the main task and SNR estimation is given as the second task. In the second algorithm, based on the SNR estimation given by the CNN, a specific SNR-dependent CNN model is selected for denoising from a pool of pre-trained SNR-dependent CNN models. As result, the SNR-aware models outperformed DNN and CNN models without SNR awareness and the SNR-aware models can improve denoising performance even when processing unseen SNR levels. However, the performance of the SNR-aware models relies on accurate SNR estimation.

In [6], a multi-objective learning CNN framework leveraging a smartphone, where log-power spectra and log Mel filterbank energy are used as primary and secondary targets for improved speech denoising is proposed. The experimental results demonstrated improvements over the compared state-of-the-art techniques (i.e., log minimum mean square error-based SE, spectral coherence-based SE, SE methods based on DNN, single channel CNN-based denoising autoencoder and multi-objective DNN-based SE) and validated the usability of the proposed method in the real world under different noisy environments and low SNRs. The multi-objective learning CNN framework makes use of TensorFlow C++ and requires a maximum memory consumption of 308.8 Mb on the smartphone when SE is switched on.

A CNN-based SE for processing speech in the time domain is presented in [10]. SE in the time domain remains desirable due to its capability to jointly enhance both magnitude and phase of speech. To achieve enhancement in the time domain in [10], frequency domain loss is used to train the CNN and an extra operation that converts the time domain representation to the frequency domain representation is added at training. Frequency domain loss gave better SE outcomes than time domain loss, and loss functions considering phase performed better than those that did not. Phase estimation

TABLE 1. Summary of recent CNN-based SE algorithms described in related work.

Algorithm	Methods	Key Results	Limitations
SNR-aware CNN-based SE [5]	- Two SNR-aware CNN models: (1) Multi-task framework, where clean speech estimation is the primary task and SNR estimation is the secondary task. (2) SNR-dependent CNN models selected for denoising from a pool of pre-trained SNR-dependent CNN models.	- Outperformed DNN and CNN models without SNR awareness and improved denoising performance even for unseen SNR levels. - SNR adaptive denoising (i.e., using a pool of pre-trained SNR-dependent CNN models) was more computationally expensive than the CNN model with multi-task learning.	- Generic convolutional filter design. - SNR estimation was not always accurate and can be improved.
Real-time CNN-based SE using a smartphone [6]	- Multi-objective learning CNN framework leveraging a smartphone that works seamlessly with hearing aids. - More accurate clean speech log-power spectra are obtained by adding log Mel filterbank energy as the secondary target for SE.	- Improvements over the compared state-of-the-art techniques (i.e., log minimum mean square error-based SE, spectral coherence-based SE, SE methods based on DNN, single channel CNN-based denoising autoencoder and multi-objective DNN-based SE) - Demonstrated real-time SE functionality with low audio latency on a smartphone. - Objectively and subjectively validated the usability of the proposed method in the real world under different noisy environments and low SNRs.	- Generic convolutional filter design. - Dependent on a smartphone, makes use of TensorFlow C++ API and requires maximum memory consumption 308.8 MB when SE is switched on – not suitable for resource-constrained implementations.
Time-domain CNN-based SE [10]	- Investigated different types of loss functions in the frequency domain. - Frequency domain loss is used to train the CNN and an extra operation that converts the time domain to the frequency domain is added at training.	- Frequency domain loss gave better SE outcomes than time domain loss. - Loss functions using the real and imaginary part of the short-time Fourier transform (STFT) did not perform as well as the loss based on the STFT magnitude. - The highest improvement was achieved using a mean absolute error loss computed on STFT magnitudes defined using the L_1-norm. - Loss functions considering phase performed better than those that did not.	- Generic convolutional filter design. - Significant performance drop when input and output lengths were reduced to one frame from four frames. - Improvement can be made to phase estimation (estimated phase not yet comparable with clean phase).
Dense CNN-based SE with self-attention [11]	- Dense convolutional neural network with skip connections and self-attention for SE in the time domain. - Each layer in the encoder and decoder comprises a dense block and an attention module. A dense block promotes feature reuse in a deeper network and the self-attention module is used for utterance-level context aggregation. - New phase constrained loss that combines STFT magnitude losses of the enhanced speech and predicted noise.	- Attention mechanism in conjunction with a normal convolution with a small receptive field (i.e., no dilation) was helpful for time-domain enhancement. - Demonstrated superiority of dense convolutional neural network over baselines employing T-F masking, spectral mapping, complex spectral mapping and temporal mapping. - The proposed model showed highly effective SE in the time domain even without dilation and attention.	- Generic convolutional filter design. - Computationally expensive and did not outperform baselines in terms of real-time performance.
Wavenet-based SE [12]	- Wavenet is used to directly estimate clean speech in the time domain.	- Wavenet is highly parallelizable during both training and inference. - Ability to predict large target fields instead of single samples and supports denoising variable-length audio. - Both computational and perceptual evaluations indicated that the Wavenet model is preferential over Wiener filtering.	- Generic convolutional filter design. - Inability to deal with sudden inferences like honks in city traffic.
U-Net CNN-based SE [13]	- U-Net-based CNN SE architecture with skip connections and focus on real-time applications.	- 27% and 11% PESQ improvements over two generative adversarial network-based baselines operating on spectral and temporal domain baselines respectively. - Demonstrated real-time operation under extremely low latency conditions while maintaining performance quality to some extent.	- Generic convolutional filter design. - Degradation in speech quality when operating in low latency duration. - The real-time factor was compromised when the processing window is reduced but led to improvements in speech intelligibility and quality.
CNN-based SE with selective kernel network [14]	- Convolutional encoder-decoder architecture with selective kernel convolution through adaptive receptive field size.	- Experimental results suggest that different receptive fields are required under various noise conditions. - The dynamic receptive field size led to improved SE performance in both seen and unseen conditions when compared to baselines with fixed receptive field size.	- The proposed model gave comparable or slightly worse results in STOI compared with RTNet-3.
Automatic searching of the optimal kernel shapes for stripe-wise network pruning [7]	- Coefficient matrices regularized by a variety of regularization terms are introduced to locate important kernel positions which remove redundant parameters in convolution kernels in the process.	- Embedding the searched kernels into VGG-16 increased the accuracy from 93.53% to 94.26% on CIFAR-10 dataset, while pruning 59.27% model parameters and reducing 27.07% inference latency.	- The proposed strip-wise pruning is not suitable for compressing compact networks.
Adaptive convolutional kernels [16]	- Convolutional network with a dynamic filter that changes its weights depending on the input image.	- The use of adaptive kernels decreased the number of epochs required for training by 2x and the number of activation function computations. - A 66x reduction of parameters was achieved compared to LeNet when evaluated with the MNIST dataset while maintaining >99% accuracy.	- Did not outperform some baselines in terms of accuracy as achieving high efficiency with comparable accuracy performance was the focus.

in [10] has room for improvement as the estimated phase was yet to be comparable with the clean phase. Another SE algorithm leveraging time domain processing is proposed in [11]. In this work, a dense convolutional network with self-attention is employed, where each layer in the encoder and decoder comprises a dense block and an attention module. The dense block promotes feature reuse in a deeper network while the self-attention module is used for utterance-level context aggregation. The dense convolutional neural network model demonstrated superiority in SE over baselines employing time-frequency (T-F) masking, spectral mapping, complex spectral mapping, and temporal mapping. However, it is more computationally expensive and did not outperform baselines in terms of real-time performance.

In [12], a CNN for speech synthesis called Wavenet is used to directly estimate clean speech in the time domain. The Wavenet model is highly parallelizable during both training and inference. Both computational and perceptual evaluations indicated that the Wavenet model is preferential over Wiener filtering. The Wavenet model, however, displayed limitations in dealing with sudden interferences like honks in city traffic.

A U-Net-based CNN SE architecture with skip connections and a focus on real-time applications is proposed in [13]. The U-Net CNN model provided 27% and 11% perceptual evaluation of speech quality (PESQ) improvement over a spectral-domain and temporal-domain baseline system respectively, as well as significantly lower latency. The real-time factor is compromised when the processing window length employed is reduced but this leads to improvements in speech intelligibility and quality.

All the aforementioned CNN-based SE algorithms employ generic convolutional filters (or kernel functions) for speech denoising. Recent studies on CNN architectures suggest more optimally designed filters can lead to improved performance, better convergence behaviour and reduced hyperparameter sensitivity. For example, in [14] a selective kernel convolution scheme is introduced into a convolutional encoder-decoder architecture for SE by employing adaptive receptive field size. The selective kernel convolution takes into consideration varying spectrograms arising from different noises, speakers, and contents of speeches. As result, the implementation of dynamic receptive field size in the convolutional layer of the encoder led to improved SE performance in both seen and unseen conditions when compared to baselines with fixed receptive field size.

In [7], a framework is developed for searching optimal filter shapes for stripe-wise network pruning. The optimal filter shapes not only provided appropriate receptive fields for each convolution layer but also removed redundant parameters in convolution filters. When embedded in the VGG-16 CNN [15] for the task of object recognition, the optimal filter shapes increased recognition accuracy from 93.53% to 94.26%, in addition to providing model compression and processing speed-up.

Adaptive convolution kernels are explored in [16] for computer vision tasks. The adaptive kernel is defined by a dynamic filter that changes its weights by itself depending on the input image. The adaptive convolutional kernels significantly reduced the number of epochs required for training, number of activation function computations and number of parameters needed in the model whilst maintaining >99% accuracy. These studies make a compelling case for investigating bespoke filter designs for SE applications. Therefore, this work explores and demonstrates the benefits of using adaptive convolutional filter design in SE.

CNN is also widely used in other speech processing methods besides SE. In recent years, speech emotion recognition and classification have garnered much research interest. An example CNN-based speech emotion recognition system is the novel hybrid architecture based on acoustic and deep features proposed in [17]. Deep features are obtained by feeding spectrogram images of the original sound signal to pre-trained CNN architectures such as VGG-16 and ResNet-101 [18]. The use of a hybrid feature vector containing deep and acoustic features showed superior classification accuracy and efficiency compared to previous approaches described in the literature (e.g., [19]). In [20], CNN is used for real-time speech source localization that is robust to realistic background acoustic conditions (noise and reverberation). A combination of the imaginary-real coefficients of STFT and spectral flux with delay-and-sum beamforming is used as the input feature. The CNN model is trained using noisy speech recordings collected from different rooms. The proposed CNN-based approach trained with STFT coefficients and spectral flux with beamforming provided successful real-time inferencing with low latency (21 ms per frame with a frame length of 30 ms) and high accuracy (i.e., 89.68% under babble noise condition at 5 dB SNR).

III. CNN-BASED SPEECH ENHANCEMENT WITH ADAPTIVE FILTER DESIGN

In a CNN-based SE, the extracted features from the audio input are represented as an image and the mapping from noisy speech to enhanced speech forms a regression problem. Through performing convolution on the audio input with a series of weighted learnable filters, a feature map which describes simple image features of the audio input is generated. The obtained feature maps are often fed to a max-pooling layer for dimensionality or resolution reduction. Oftentimes, the max-pooled output is connected to another convolution layer as more complex feature learning can be achieved with the addition of more convolution layers. The outputs of convolution or max-pooling layers are flattened and fed to fully-connected layers before the final regression output (enhanced speech) is obtained. The enhanced speech output is given at the nonlinear output layer which comes after the fully-connected layers. From training, the CNN learns to identify important T-F auditory features such as formants. Since each part of the feature map is convoluted with the same filter, the CNN is invariant to translational variance. This allows the network to be robust in processing speech from different genders, and possibly languages and accents,

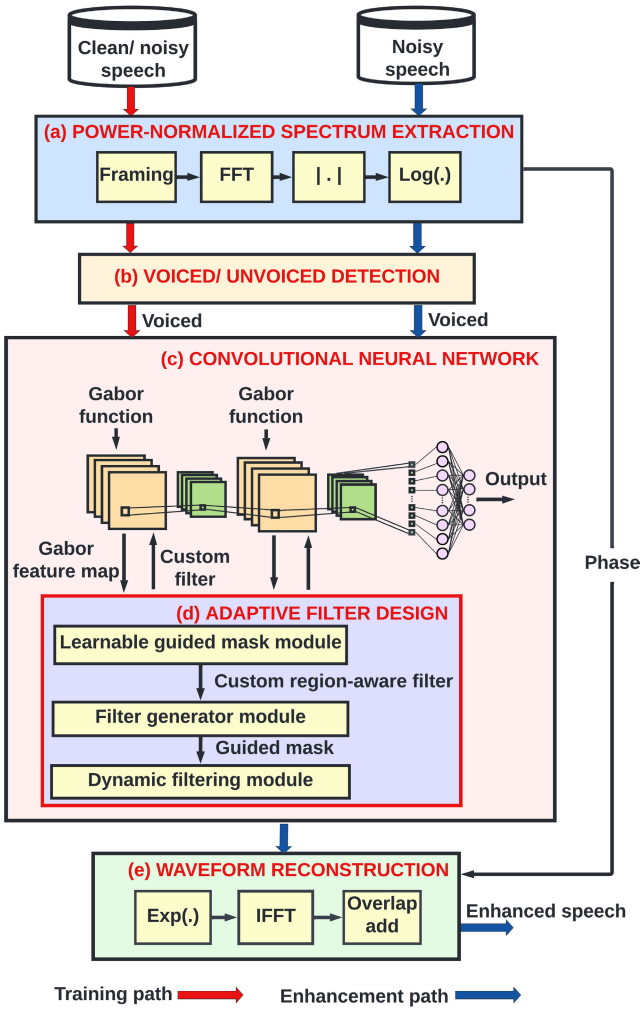


FIGURE 1. System overview of the proposed CNN-based SE method with adaptive filter design (CNN-AFD). The adaptive filter design is conducted via a three-step process shown in (c) which involves: (1) extracting the learnable guided mask from the Gabor feature map; (2) generating the custom region-aware filter based on the extracted learnable guided mask; and (3) performing convolution operation using the generated custom filter.

where different pitches (related to fundamental frequencies) are presented.

Fig. 1 illustrates the system-level overview of the proposed CNN-AFD, where the CNN filters are generated dynamically, conditioned on the input noisy speech frames. In the training stage (denoted by the red arrows in Fig. 1, representing the training path), the CNN model is trained with pairs of 64-frequency channel power-normalized spectrum (PNS) [21] features extracted from the noisy speech and the corresponding clean speech. The PNS is obtained using the fast Fourier transform as shown in Fig. 1(a). Subsequently, the logarithmic function is performed on the normalized energy magnitude. Note that the PNS features form a single-channel 2D image unlike RGB images in computer vision tasks which contain three channels defining red, green and blue color components. Skip convolutions are performed when the examination of the extracted PNS suggests an absence of speech (unvoiced). This process is depicted in Fig. 1(b). In the enhancement stage (denoted by the blue arrows in Fig. 1, representing the enhancement path), the trained CNN-AFD model is fed with the PNS features of noisy speech from the test set. Through forward propagation within the trained model, the enhanced PNS features would be given as the output. The enhanced speech is then combined with the phase information extracted from the input noisy speech **for** waveform reconstruction. At reconstruction, the enhanced speech features undergo inverse fast Fourier transform. This is followed by performing overlap-add to synthesize the enhanced speech in the time domain (Fig. 1(e)). The proposed design consists of three major components as shown in Fig. 1(d) to achieve a CNN-based SE with adaptive filter design and filtering: (i) a learnable guided mask module, where convolution with the Gabor filter is performed to generate the speech/non-speech region patterns according to the input frames; (ii) filter generator module, where a customized filter is extracted from the speech region patterns and (iii) dynamic filtering module, where region-aware convolution is performed using the customized filter. The parameters within the filter generator and dynamic filtering modules are not fixed after training like regular deep-learning model parameters as they are made to adapt according to the calculated learnable guided mask, which is in contrast, fixed and optimized at training.

A. ADAPTIVE FILTER DESIGN

1) LEARNABLE GUIDED MASK MODULE

The learnable guided mask is calculated based on the output from the Gabor filter and it indicates the regions containing speech and non-speech components within each input noisy speech utterances. The Gabor filter [22], popular for its invariance in scale, rotation, and translation, captures localized regions of temporal-spectral information over broader time intervals. Neurophysiological evidence [23] from several animals shows that the 2D temporal-spectral Gabor filterbank resembles the temporal-spectral receptive fields of auditory cortical cells. Due to this and its other inherent benefits such as noise robustness, it is widely used in speech processing applications. The two-dimensional Gabor function G implemented in this paper is defined by a complex sinusoidal signal modulated by the Gaussian envelope. Only the real part of the Gabor filter equation is used [22]:

$$G = \exp\left(-\frac{\hat{u}^2 + \gamma\hat{v}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\hat{u}}{\lambda} + \psi\right), \tag{1}$$

where

$$\hat{u} = (u - u_0)\cos\theta + (v - v_0)\sin\theta, \tag{2}$$

$$\hat{v} = -(u - u_0)\sin\theta + (v - v_0)\cos\theta, \tag{3}$$

λ is the wavelength of the sinusoidal factor, θ is the orientation of the Gabor filter, ψ is the phase offset, σ represents the standard deviation of the Gaussian envelope, γ denotes the spatial aspect ratio and finally, (u_0, v_0) is the location of the center.

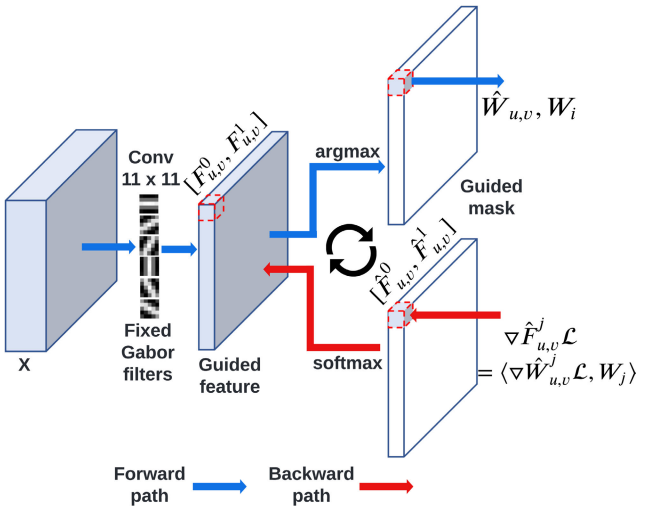


FIGURE 2. Learnable guided mask acquisition and optimization process. The forward propagation involves calculating the guided mask by applying the *argmax*(.) function on the Gabor guided feature. The backpropagation process requires the introduction of an intermediary term $\hat{F}^j_{u,v}$ which enforces the guided feature to closely approximate the guided mask. Calculating the error gradient for $\hat{F}^j_{u,v}$ will allow the estimation of the error gradient for the guided feature $F^j_{u,v}$, which is needed to enable successful backpropagation training.

The ψ, σ and λ were set to the following ranges: $\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4},\}$ $\{1, 2, 3, 4, 5\}$ and $\{0.2, 0.4, 0.6, 0.8, 1\}$ respectively. The parameters leading to the best validation accuracy are subsequently chosen in the Gabor filter construction. θ and (u_0, v_0) were kept controlled at 0° and $(0.5k, 0.5k)$, respectively, where k is the filter size, at the time of the preliminary test. Adjusting θ tunes the Gabor function to a particular direction of temporal-spectral modulation. Only a subset of the possible filter combinations is used to avoid high similarities (leading to redundancies) between the feature components. Narrow Gabor filters capture rapid time-varying detail of the temporal-spectral input, while coarse (or dilated) filters highlight the coarse representations from the speech signal. Similarly, tall, and short filters each capture different spectral dynamics.

The input to the Gabor filter can be denoted as $X \in \mathbb{R}^{U\times V}$, where U and V are the height and width of the 2D PNS matrix $\mathbb{R}$, respectively. When convolved with a Gabor filter G, the corresponding feature map output $F \in \mathbb{R}^{U\times V}$ is given by:

$$F_{u,v} = X_{u,v} * G(u_0, v_0) \in S, \tag{4}$$

where S represents the spatial dimension ($S \in \mathbb{R}^{U\times V}$) and $*$ is the 2D convolution operation. (u_0, v_0) represents a T-F unit within the PNS matrix that corresponds to the center of the filter when the filter size is larger than 1×1. From the feature map output, a guided mask $M = \{S_0, \ldots, S_{m-1}\}$ which represents the different regions (i.e., regions dominated by speech and regions dominated by noise or silence) within the spatial dimension is obtained.

The learning process of the guided mask is shown in Fig. 2. The guided mask $M \in \mathbb{R}^{U\times V}$ is computed based on the Gabor feature map $F \in \mathbb{R}^{U\times V}$, by executing the following expression [24] in the forward propagation (shown by blue arrows in Fig. 2):

$$M_{u,v} = argmax(F^0_{u,v}, F^1_{u,v}), \tag{5}$$

where *argmax* (.) gives the index (0 or 1) at which the maximum value is found when $F^0_{u,v}$ and $F^1_{u,v}$ are compared, $F^0_{u,v}$ is the Gabor guided feature for non-speech-dominated regions and $F^1_{u,v}$ is the guided feature for speech-dominated regions. Therefore, the guided mask varies between 0 and 1, which indicates the index of the guided feature that should be used in the corresponding position. From the guided mask, a filter $\hat{W}_{u,v}$ for each position (u, v) can be obtained as follows:

$$\hat{W}_{u,v} = W_{M_{u,v}} M_{u,v} \in [0, 1] = W * M_{u,v} \tag{6}$$

where $W_{M_{u,v}}$ is one of the filters $[W_0, W_1]$ generated in the learnable guided mask module. $*$ denotes 2D convolution operation.

At the backward propagation process (shown by red arrows in Fig. 2), the error gradients of the weights which produce the guided feature are approximated. The gradient is estimated from comparing the performance given by the guided mask with the intended outcome. It is subsequently backpropagated to the guided feature to form a training loop which in turn optimizes the guided mask. To calculate the gradient, a new term, $\hat{F}^j_{u,v}$, which closely approximates the guided mask is introduced. $\hat{F}^j_{u,v}$ is obtained by applying a softmax function to the guided feature $F^j_{u,v}$ as follows:

$$\hat{F}^j_{u,v} = \frac{e^{F^j_{u,v}}}{\sum_{n=0}^{1} e^{F^n_{u,v}}} j \in [0, 1], \tag{7}$$

Eq. (7) enforces the guided feature to be as close to 0 or 1 and as a result, reduces the gap between the guided feature and the guided mask. The gradient of $\hat{F}^j_{u,v}$ is computed as follows:

$$\nabla_{\hat{F}^j_{u,v}} \mathcal{L} = \langle \Delta_{\hat{W}_{u,v}} \mathcal{L}, W_j \rangle \quad j \in [0, 1], \tag{8}$$

where $\nabla_{\cdot}\mathcal{L}$ represents the tensor's gradient with respect to the loss function and $\langle , \rangle$ denotes dot product. From the gradient of $\hat{F}^j_{u,v}$, the gradient of the guided feature $F^j_{u,v}$ finally can be obtained:

$$\nabla_{F_{u,v}} \mathcal{L} = \hat{F}^j_{u,v} \odot (\nabla_{\hat{F}^j_{u,v}} \mathcal{L} - 1), \langle F_{u,v}, \Delta_{\hat{F}^j_{u,v}} \mathcal{L} \rangle \tag{9}$$

where $\odot$ is element-by-element multiplication. The approximate backpropagation is needed to ensure that the stochastic gradient descent algorithm (used for backpropagation training) is successful at parameter optimization since the *argmax* (.) function is non-differentiable, leading to the ceasing of gradient propagation. Using the softmax function allows the transfer of the gradient to the guided feature, ensuring the guided mask is trainable and optimized.

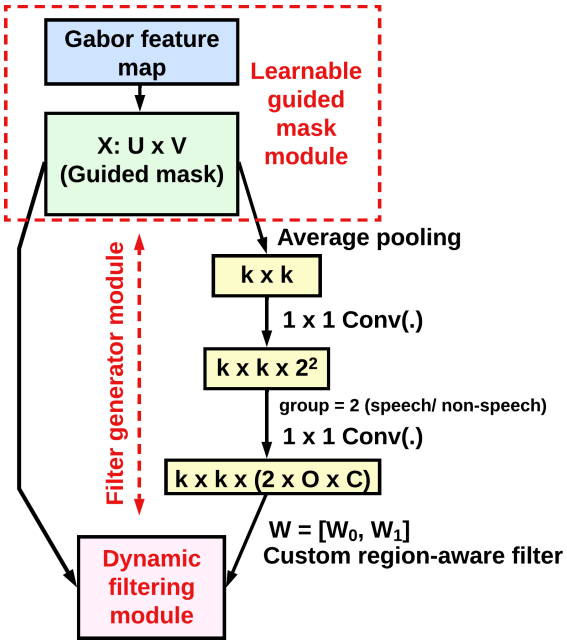


FIGURE 3. Filter generator module where the custom region-aware filter is obtained from the guided mask through a series of downsampling. Subsequently, the convolution of the noisy PNS input with the calculated custom filter is performed in the dynamic filtering module.

2) FILTER GENERATOR MODULE

The filter generator module computes the custom filters for the speech regions detected by the learnable guided mask module. The execution process to adaptively obtain the custom filter within the filter generator module is illustrated in Fig. 3. As shown in the figure, the input to the filter generator module $X \in \mathbb{R}^{U \times V}$ (output of the learnable guided mask module) is average pooled such that it is downsampled to filter size $k \times k$. The custom filter size $k \times k$ is made to be the same as the Gabor filter size which is 11×11. The downsampling process consists of applying two 1×1 convolution layers consecutively: the first involves the use of a sigmoid activation function and the second does not employ any activation function. The filter generator module adaptively generates custom filters based on the obtained guided mask to more optimally capture characteristics of the speech components given a noisy speech frame. This is beneficial since the T-F characteristics of different speech sounds, such as vowels, stops, weak fricatives, strong fricatives, and nasals, are very different [25]. Therefore, using filters whose design can adapt based on the constantly evolving T-F speech shapes would lead to improved mapping from noisy speech to enhanced speech.

3) DYNAMIC FILTERING MODULE

The dynamic filtering module is the final step within the adaptive filter design pipeline. It basically involves performing convolution on the PNS feature map using the computed custom region-aware filter. The dynamic filtering module takes feature maps as inputs and gives the filtered results as outputs. It is similar to a traditional convolutional layer, where the same filter is applied at every position of the 2D PNS input. In the dynamic filtering module, however, the filter parameters are dynamically generated by the filter generator module. The filters are sample-specific and conditioned on the input frame. The dynamic filtering module is executed after the filter generator module as shown in Fig. 3.

B. CONVOLUTIONAL NEURAL NETWORK

The proposed CNN architecture is illustrated in Fig. 4. As shown in the figure, the CNN has 5 hidden layers, 2 convolutional layers with a max-pooling layer in between, and 2 fully-connected layers. In a convolutional layer, a neuron is connected to a local subset of frequency bands, where a set of neurons with receptive fields shifted in frequency share the same filter weights. The activation of each neuron is computed by multiplying a local receptive field with the network weights before adding a bias, and then finally applying a nonlinear function:

$$h_m(n,c) = \alpha\left[\sum_{i=-N}^{N}\sum_{j=-C}^{C} W_m(i,j)\cdot x(n+i,c+j) + b_m\right] = \alpha\left[W_m(-n,-c) * x(n,c) + b_m\right], \quad (10)$$

where $h_m(n,c)$ represents the neuron of the mth feature map, whose receptive field (centered at $x(n,c)$) encompasses $2C+1$ frequency bands and $2N+1$ time frames. W_m, b_m and $\alpha(.)$ are the network weights, bias and sigmoid function, respectively. To model Gabor filtering in CNNs, Gabor filter coefficients are incorporated into the initial weights W_m. In addition, the nonlinear function $\alpha(.)$ is replaced with a linear one and the bias term b_m is enforced to be zero. This change to Eq. (10) means that the neurons of the convolutional layers simply compute the filter outputs of the receptive field. Since Gabor features consist of filters with different time and frequency band supports, the receptive field size is modified to give the same supports of the Gabor filters rather than remaining fixed.

In the proposed architecture, the first convolutional layer is initialized with 6 fixed Gabor filters, equally spaced in orientation (i.e., $\theta = 0°, 30°, 60°, 90°, 120°$ and $150°$). The fixed Gabor filters are then used to generate the learnable guided masks which are in turn used to obtain the custom region-aware filters. Although weight initialization is performed by employing fixed Gabor functions in the convolutional layers, adaptivity and awareness are incorporated by expanding on the Gabor feature map to perform guided mask construction, filter generation and eventually dynamic filtering as discussed in Section III.A. Unlike the fixed Gabor filters, the custom region-aware filters are trainable at backpropagation. The second convolutional layer contains 12 fixed Gabor filters for each of the 6 output feature maps of the first convolutional layer, giving a total of 72 filters. The 12 fixed Gabor filters are also equally spaced in orientation

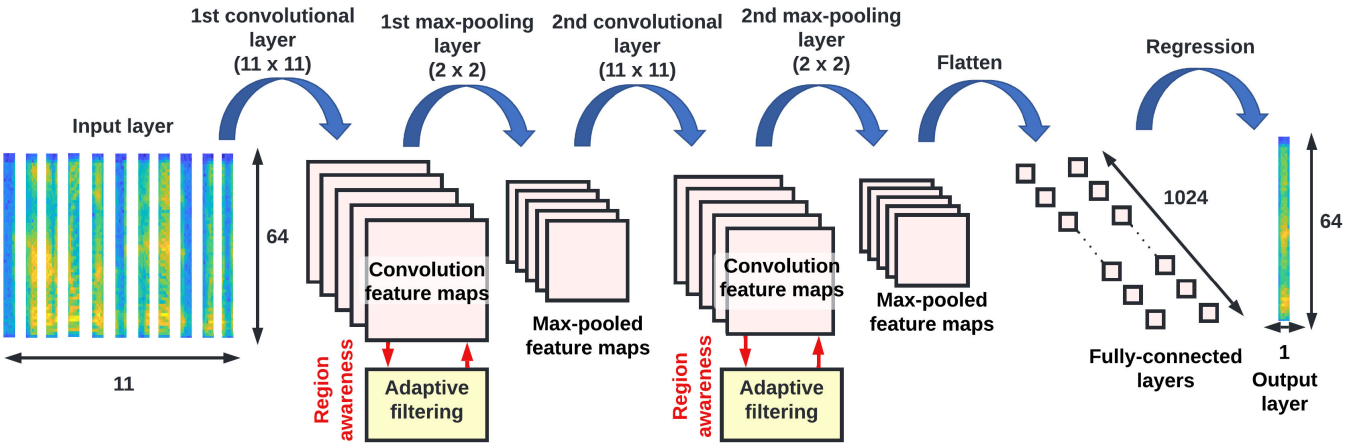


FIGURE 4. The employed CNN architecture with two convolutional, max-pooling and fully-connected layers. Adaptive filter design is embedded in each convolutional layer.

(i.e., $\theta = 0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°$ and $165°$). Since the same set of 12 fixed Gabor filters is convolved with each of the 6 output feature maps of the first convolutional layer at every training cycle, the 6 Gabor filters of the first convolutional layer are also present in the second convolutional layer. This allows the integrity of the previous layer to be carried forward to the following layer.

According to [26], the window length, window shift and normal length of vowels are approximately 32, 16 and 99 ms, respectively. This led to the decision of setting the Gabor filters in the time axis to 11 frames in all convolutional layers. This results in the filters covering approximately 192 ms, which is around twice the vowel length. Therefore, the filters are 11×11 to ensure symmetry such that both temporal and spectral patterns are equally considered. The signal frames are obtained using a Hanning window with 256 points and an overlap interval of 128 points. All data were sampled at 8 kHz. Max-pooling is used to reduce the size of the convolutional layer and remove variance along the convolutional bands. It involves taking the maximum value from a window of activations (i.e., region of the feature map covered by the filter) in a convolutional layer. This leads to the extraction of the most prominent features from the previous feature map. The window size is referred to as the pooling size. A pooling size of 2×2 with a stride of 2, giving a dimensionality reduction by a factor of 2, is employed for the max-pooling layers. Two fully-connected layers are added after the last max-pooling layer to integrate the pooled features. Fully-connected layers are an essential component of CNNs as they are responsible for learning the relationship between the features and the desired output. The fully-connected layers of the proposed CNN-AFD contain 1024 neurons and employ the ReLU activation function. Restricted Boltzman machine pre-training [27] is used to initialize the parameters of the fully-connected layers. Thereafter, the CNN is trained for 50 epochs with the stochastic gradient descent algorithm. The learning rate is initially set at 0.015. This reduces by factors of 2 when the 5-fold cross-validation loss reaches convergence and continues to decay until the validation loss shows no further change. The mean-square error is set as the error criterion.

IV. CONVOLUTIONAL NETWORK OPTIMIZATION

One of the main drawbacks of CNN is the demand for large computational and storage overhead, which constitutes a challenge in deployment on devices with limited computing resources (e.g., implantable devices). Within a CNN, significant resource consumption resides in the convolutional and fully-connected layers. This is because the convolution operation involves computationally expensive repetitions of multiplications and summations over iterations of sliding windows. Similarly, fully-connected layers suffer from great redundancy. In this paper, skip convolution and an activation analysis-based pruning method are explored for CNN optimization.

A. SKIP CONVOLUTION

CNNs are highly redundant in terms of computation due to the presence of spatial redundancy. Each pixel or in this case, each T-F unit on the PNS is surrounded by very similar T-F units in the neighbourhood. Greater efficiency can be achieved by exploiting spatial redundancy in feature maps to bypass extra computations. Skip convolution [28] can be used to attain improved efficiency by skipping convolution operations in the nearby T-F units (horizontally or vertically) while ensuring feature representational capability is maintained. It is motivated by the Nyquist sampling theorem, which states that a signal can be reconstructed without loss of information when a signal is sampled at twice the highest frequency. Besides this, a speech spectrogram usually depicts continuous signals in certain frequency bands and has recurring unvoiced intervals. The structured sparsity presents a suitable scenario for sampling alternate rows or columns. Structured sparsity can also be introduced to the output feature maps by

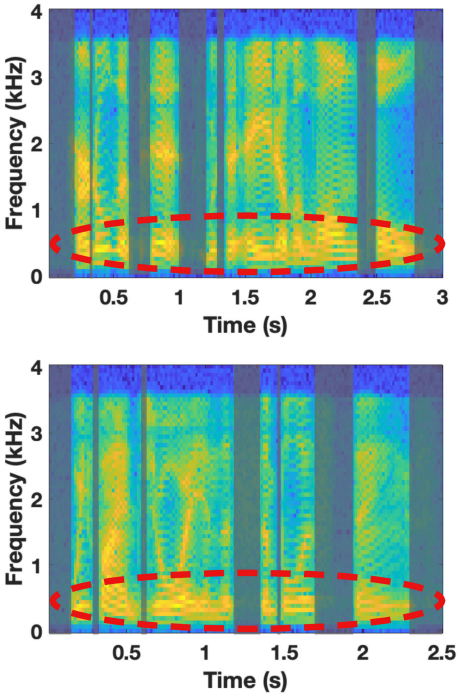


FIGURE 5. Skip convolution criterion. Convolution operations are omitted for unvoiced frames (example areas where skip convolution is carried out are shown by the greyed boxes). A frame is deemed unvoiced when the lower frequency bands (red dashed circles) in the PNC have significantly less energy than that of the preceding frame.

intentionally skipping corresponding rows or columns of the convolutional filters. This subsequently establishes sparsity in the CNN model parameters during training.

In this work, skipping convolution introduced in [28] is adapted for the proposed SE method. Instead of skipping convolution operations on alternate rows or columns of pixels, this work uses skipping convolution to omit performing convolutions on unvoiced intervals since the number of PNS frequency bands (rows) employed is already limited. Unvoiced intervals refer to frames (columns) where speech is not present (e.g., pauses between words or syllables). Since speech signals are often shown to possess high energy in lower frequencies as illustrated by the dashed circles in Fig. 5, it can be assumed that a frame is unvoiced when the combined first 15 bands (around a quarter of the 64 total bands) of a single PNS frame contains lesser than a tenth of the energy of the combined first 15 bands of the preceding frame. The greyed areas shown in Fig. 5 depict the segments on the PNS feature map where the first 15 frequency bands are significantly lower in energy, corresponding to columns (i.e., frames) where convolutions can be skipped. A constraint is imposed on the skip convolution decision such that no two subsequent frames are skipped to satisfy the Nyquist sampling theorem. In addition, the latest calculated first-15-band energy of a voiced frame is stored in memory to enable the identification of longer pauses or silence (in this case, the energy of the current unvoiced frame will not be significantly lesser than that of the preceding frame's which was also unvoiced). The proposed approach of ignoring unvoiced frames to achieve skip convolutions provided comparable performance to a traditional voice activity detection proposed in [29] even though little time was spent on perfecting the voice activity detection capability of the proposed approach. The proposed approach was used instead of the traditional voice activity detection due to its simplicity. Furthermore, it utilizes already existing components (i.e., PNS extraction) within the algorithm to perform voiced/ unvoiced frame identification, making it a more efficient approach. This paper explores the impact of performing the proposed skipping convolution with or without max-pooling layers. The results are reported as part of the ablation study given in Section VI.A.

B. ACTIVATION ANALYSIS PRUNING

In this work, network pruning is used to prune lesser important filters, feature maps, as well as neurons in the fully-connected layers to achieve model compression while ensuring speech denoising capability is not compromised. The baseline CNN is pruned via activation analysis to remove neurons that do not contribute to the regression output. Structured pruning is enforced by encouraging frequency band-wise or time frame-wise pruning instead of pruning individual neurons. The proposed activation analysis pruning benefits from the structured sparsity introduced by skip convolution. Since skip convolution is performed on the unvoiced frames as explained in Section IV.A, many column-wise neurons and their associated connections will be pruned. The activation analysis evaluates the importance of any neuron by calculating the average percentage of zeros (APoZ) [30] from the activation output of each neuron at training. For a given convolutional layer CL, the importance of the custom filter given by the filter generator module is calculated as follows:

$$CL = \frac{\sum_{i=1}^{O}\sum_{j=1}^{S} f(\mathrm{A} = 0 || \mathrm{B} \geq 2\sigma)}{O \times S}, \tag{11}$$

where $\mathrm{A} = CL_r(k)$ and $\mathrm{B} = \frac{1}{5}\sum_{r-2}^{r+2} CL_{r-1}^{r+1}(k) \geq 2\sigma$. r denotes the neuron index within the kth convolutional layer. S and O are the total numbers of training samples and the dimension of the output feature map respectively. $f(.)$ is computed as follows:

$$f(.) = \begin{cases} 0, & if\ \mathrm{A} = 0\ or\ \mathrm{B} \geq 2\sigma \\ 1, & else. \end{cases} \tag{12}$$

The pruning process begins by calculating the neuron's importance in the convolutional layers using Eq. (11). A neuron and its connections are removed if the APoZ of the neuron is zero or if a group of five consecutive row or column-wise neurons possesses an average APoZ of larger than twice the standard deviation of the overall average APoZ in the same layer. Once the pruning of both convolutional layers is completed, the network is fine-tuned to recover the original accuracy using the baseline model's training configuration described in Section III.B.

TABLE 2. Summary on the development of the train and test set used for evaluating the SE algorithms.

Dataset	Setup
Train Set	• 1500 random TIMIT utterances. • Random cuts of the first 2 minutes of NOISEX-92 noise (i.e., babble, factory, pink, Volvo and white noise). • -5, 0 and 5 dB SNR.
Test Set	• 192 TIMIT utterances that were not used in the train set. • Random cuts of the last 2 minutes of NOISEX-92 noise (i.e., babble, factory, pink, Volvo, white, f16 and factory2 noise). • -10, -5, 0 ,3 and 5 dB SNR.

V. EXPERIMENTAL SETTINGS

A. DATASETS

The proposed CNN-AFD was evaluated using the TIMIT [31] corpus and NOISEX-92 [32] database. The TIMIT corpus contains 6300 clean phonetically-rich English utterances that include 8 major American English dialects. The corpus was developed by recording 630 participating speakers (male and female), where each spoke 10 sentences. For the development of the training set, the dialect sentences were removed, and 1500 utterances were randomly selected for training. The training utterances were mixed with five types of noises (babble, factory, pink, Volvo (car) and white noise) from the NOISEX-92 database at three SNR levels (-5, 0, 5 dB). For the development of the test set, the TIMIT core test set containing 192 utterances that were not part of the training set were used. Similarly, they were combined with the same five types of noises from the NOISEX-92 database at -5, 0 and 5 dB SNR. To avoid using the exact same frames of noise in both training and testing, random cuts of the first 2 minutes of the noise recordings were used for training. The test set consisted of random cuts of the last 2 minutes of the noise recordings. To evaluate the generalization performance of the proposed approach, two unseen (untrained) types of noises ('f16' and 'factory2' noise) and two other SNR values (-10 and 3 dB) were added to the test set. Table 2 summarizes the development of the training and test set.

B. EVALUATION METRICS

For speech denoising performance evaluation, the short-time objective intelligibility (STOI) [33], perceptual evaluation of speech quality (PESQ) [34] and logarithmic spectral distance (LSD) [35] were employed. STOI was found to be positively related to subjective speech intelligibility. It is calculated using the short-time (386 ms) temporal envelopes of the clean speech and the estimated speech. STOI varies between 0 to 1, with 1 indicating the absence of speech distortion (i.e., best achievable STOI). STOI is mathematically calculated as follows [33]:

$$STOI = \frac{1}{U, V} \sum_{U,V} \frac{\left(x_{u,v} - \mu_x\right)^T \left(\hat{x}_{u,v} - \mu_{\hat{x}}\right)}{\left\|x_{u,v} - \mu_x\right\| \left\|\hat{x}_{u,v} - \mu_{\hat{x}}\right\|}, \quad (13)$$

where u and v are time and frequency indexes respectively. $\|.\|$ denotes the Euclidean ℓ^2-norm. $x_{u,v}$ and $\hat{x}_{u,v}$ represent the short-time temporal envelope of the clean and enhanced speech, respectively. μ is the sample average of the corresponding vector ($x_{u,v}$ or $\hat{x}_{u,v}$).

PESQ was recommended by the ITU-T for objective speech quality assessment. It is computed using a linear combination of disturbance parameters to predict the subjective mean opinion score. Two parameters, symmetric disturbance (d_{SYM}) and asymmetric disturbance (d_{ASYM}), are combined to predict the speech quality as follows [34]:

$$PESQ = 4.5 - 0.1 d_{SYM} - 0.0309 d_{ASYM}, \quad (14)$$

where d_{SYM} and d_{ASYM} are calculated by the disturbance processing model in PESQ [30]. PESQ falls within the range -0.5 and 4.5, where the higher PESQ score represents better perceptual speech quality. PESQ may fall below 1 in extremely high distortion conditions, but this is usually uncommon in the real-world.

LSD measures the logarithmic spectral distance (averaged over all frames) between two speech samples (in this case, clean and enhanced speech) as follows [35]:

$$LSD\left(x, \hat{x}\right) = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\frac{1}{F} \sum_{f=1}^{F} \left(x_{t,f} - \hat{x}_{t,f}\right)^2}, \quad (15)$$

where $x_{t,f}$ and $\hat{x}_{t,f}$ represent the clean and enhanced speech respectively. T and F denote the number of time frames and frequency bins of the samples, respectively. In contrast to STOI and PESQ, lower LSD is desirable as it represents lower speech distortion. When LSD is zero, this suggests the signal is clean speech.

C. BASELINE MODELS

The speech denoising performance of the proposed CNN-AFD is compared against six baseline models, four of which are other CNN-based SE methods and two are DNN-based SE methods. The first fully-connected DNN model is known as 'DNN-IRM'. It employs 4 hidden layers, each with 1024 neurons, and is trained to estimate the ideal ratio mask [36] from a noisy PNS input. The second DNN-based SE approach is the 'DNN-QCM' proposed in [3]. The DNN-QCM uses correlation information between clean speech/ noise and noisy speech to fine-tune the ideal ratio mask. Similarly, the DNN-QCM is trained to process noisy PNS input for consistency. The first CNN baseline employs two convolutional layers, each followed by a max-pooling layer, and then two fully-connected layers. This is similar to the topology used for the proposed CNN-based SE method described in Section III.B. However, 64 fixed convolution filters of size 3×3 are used in each of the convolutional layers instead. This CNN baseline is referred to as 'CNN-Norm'. The second CNN baseline has the same two convolutional layers, two max-pooling layers and two fully-connected layers topology but only employs fixed Gabor filters (i.e., similar to the proposed approach but without adaptive filter design).

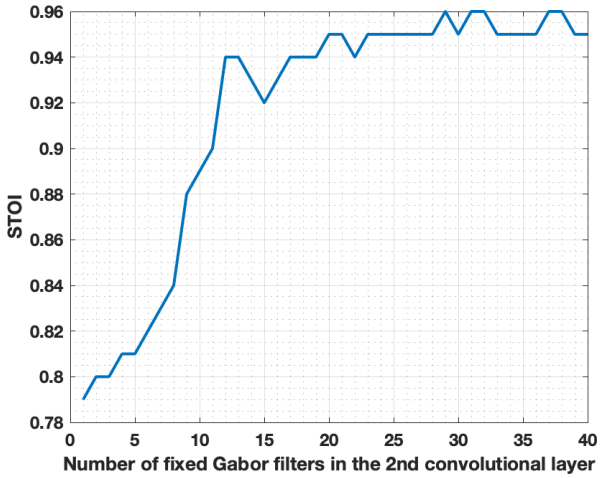


FIGURE 6. STOI performance with varying numbers of fixed Gabor filters employed for the 2nd convolutional layer. Saturation in STOI performance is observed with a higher number of fixed Gabor filters.

TABLE 3. STOI performance with different max-pooling and skip convolution configurations.

CNN Configuration	STOI
Without max-pooling and skip convolution	0.85
With max-pooling only	0.92
With skip convolution only	0.94
With max-pooling and skip convolution	0.89

TABLE 4. Average training per epoch with and without skip convolution and average inference time when processing an input frame with and without skip convolution.

CNN Configuration	Ave. Training Time (s)	Ave. Inference Time (s)
Without skip convolution	262	0.019
With skip convolution	159	0.007

This is referred to as 'CNN-Gabor' in the results provided. The third CNN baseline is named 'CNN-Dilated' in this paper and it represents an SE approach involving complex spectrogram processing using frequency-dilated filters proposed by [37]. Finally, the last CNN baseline is the 'TCNN' described in [38]. TCNN is a temporal convolutional neural network proposed for real-time SE in the time domain.

VI. RESULTS

A. ABLATION STUDY

Fig. 6 depicts the STOI score achieved when varying numbers of Gabor filters were employed in the second convolutional layer. The performance of the CNN-based SE with and without max-pooling, with and without skip convolution, and with and without combining max-pooling and skip convolution

TABLE 5. Percentage parameter reduction achieved on every CNN layer with activation analysis pruning.

CNN Layer	% Parameter Reduction with Act. Analysis Pruning Only	% Parameter Reduction with Skip Conv. + Act. Analysis Pruning
1st conv. layer	16.12%	36.21%
2nd conv. layer	22.31%	41.53%
1st fully-con. layer	18.91%	33.01%
2nd fully-con. layer	18.03%	37.44%

was assessed. When investigating this, activation analysis pruning was not implemented. The performance was evaluated using STOI and the results are shown in Table 3. Table 4 presents the average training time required to train an epoch and the average inference time required to process an input frame of 35 ms with or without skip convolution. The percentage of parameters pruned at each layer using activation analysis pruning was assessed by comparing the trained proposed method with and without pruning. The efficacy of activation analysis pruning was assessed when it was used on its own as well as when it was combined with skip convolution. The results are shown in Table 5.

B. SPEECH ENHANCEMENT PERFORMANCE

Table 6 lists the denoising performance of the proposed CNN-AFD (optimized with activation analysis pruning and skip convolution) and the other considered SE algorithms, evaluated using mean STOI, PESQ and LSD scores. The optimal values are marked in bold.

C. GENERALIZATION CAPABILITY

For supervised deep-learning algorithms, generalization ability is an important aspect of performance evaluation. The generalization of the proposed approach is mainly evaluated from two perspectives: noise and SNR generalization ability. Fig. 7 shows the performance of the proposed CNN-AFD and the baselines on unseen noises (i.e., 'factory2' and 'f16' noises) and SNRs (-10 and 3 dB).

VII. DISCUSSION

The STOI score generally improved with an increasing number of Gabor filters. However, using more filters also incurred higher computational costs. As shown in Fig. 6, it was found that using 12 Gabor filters in the second layer led to the optimal performance before the enhancement performance begins to saturate with a higher number of filters due to increasing redundancy. From Table 3, it was observed that the CNN with skip convolution without max-pooling led to the best STOI score. The CNN with skip convolution without max-pooling led to the best STOI score likely because it was harder to perform skip convolution on unvoiced

TABLE 6. Speech denoising performance (STOI, PESQ and LSD) comparison of the various deep learning-based SE algorithms on different noise types and SNR conditions.

SE	Babble			Factory			Pink			Volvo			White		
	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB	-5 dB	0 dB	5 dB
	STOI														
Unprocessed	0.56	0.67	0.73	0.57	0.70	0.82	0.58	0.68	0.84	0.84	0.96	0.96	0.58	0.69	0.82
DNN-IRM	0.75	0.82	0.84	0.77	0.78	0.87	0.80	0.83	0.86	0.93	0.95	0.97	0.79	0.88	0.91
DNN-QCM	0.84	0.88	0.89	0.87	0.90	0.91	0.87	0.90	0.93	0.92	0.97	**0.98**	0.85	0.93	096
CNN-Norm	0.77	0.84	0.86	0.80	0.83	0.88	0.82	0.86	0.92	0.93	0.97	0.97	0.82	0.90	0.93
CNN-Gabor	0.79	0.84	0.89	0.83	0.86	0.90	0.85	0.88	0.94	0.94	0.97	**0.98**	0.85	0.90	0.94
CNN-Dilated	0.87	0.90	0.92	0.90	0.92	0.94	0.89	0.92	0.95	**0.95**	**0.98**	**0.98**	0.89	0.94	0.97
TCNN	0.85	0.88	0.90	0.89	0.92	0.93	0.85	0.88	0.94	0.94	**0.98**	**0.98**	0.89	0.93	0.97
CNN-AFD	**0.90**	**0.93**	**0.95**	**0.92**	**0.95**	**0.97**	**0.91**	**0.94**	**0.97**	**0.95**	**0.98**	**0.98**	**0.91**	**0.96**	**0.98**
	PESQ														
Unprocessed	1.07	1.09	1.45	1.03	1.06	1.58	1.05	1.08	1.61	1.35	1.53	1.57	1.06	1.07	1.52
DNN-IRM	1.13	1.32	1.89	1.11	1.15	1.82	1.24	1.32	1.83	1.85	1.83	2.10	1.14	1.31	1.83
DNN-QCM	1.32	1.77	1.97	1.41	1.46	1.90	1.39	1.55	1.93	1.93	2.07	2.14	1.20	1.58	1.93
CNN-Norm	1.23	1.41	1.92	1.25	1.32	1.87	1.32	1.40	1.89	1.89	1.93	2.16	1.19	1.39	1.92
CNN-Gabor	1.34	1.46	1.96	1.31	1.37	1.92	1.39	1.44	1.92	1.93	1.98	2.20	1.24	1.42	1.93
CNN-Dilated	1.40	1.66	2.00	1.41	1.52	1.98	1.47	1.56	1.96	2.01	2.10	2.23	1.30	1.59	2.04
TCNN	1.38	1.79	1.98	1.38	1.46	1.93	1.43	1.55	1.96	1.96	2.04	2.19	1.28	1.54	1.98
CNN-AFD	**1.43**	**1.82**	**2.09**	**1.47**	**1.66**	**2.05**	**1.53**	**1.61**	**2.03**	**2.05**	**2.19**	**2.26**	**1.38**	**1.66**	**2.12**
	LSD (dB)														
Unprocessed	1.72	1.61	0.98	2.98	2.62	2.51	2.82	1.94	1.65	0.89	0.83	0.79	2.56	2.28	1.93
DNN-IRM	1.04	0.98	0.70	1.76	1.51	1.19	1.73	1.02	0.95	0.74	0.58	0.55	1.37	1.14	1.05
DNN-QCM	0.96	0.85	0.62	1.44	1.03	0.86	1.44	0.89	0.84	0.55	0.50	0.48	1.10	0.94	0.84
CNN-Norm	0.99	0.93	0.64	1.64	1.32	1.09	1.69	0.97	0.89	0.68	0.51	0.51	1.30	1.02	0.95
CNN-Gabor	0.98	0.90	0.62	1.60	1.27	1.00	1.63	0.93	0.85	0.64	0.46	0.43	1.26	0.96	0.91
CNN-Dilated	0.93	0.83	0.62	1.48	0.97	0.84	1.32	0.88	0.75	**0.51**	0.44	0.41	1.09	0.89	0.82
TCNN	0.95	0.86	0.61	1.53	1.01	0.90	1.40	0.90	0.79	**0.51**	**0.43**	0.41	1.15	0.93	0.85
CNN-AFD	**0.90**	**0.80**	**0.60**	**1.40**	**0.92**	**0.82**	**1.23**	**0.84**	**0.74**	**0.51**	**0.43**	**0.40**	**1.07**	**0.83**	**0.80**

intervals after downsampling via max-pooling especially since unvoiced segments occur within short time frames. The computational savings achieved from skip convolution depends on how frequently unvoiced intervals are present within a sentence. Table 4 depicts that close to 40% reduction in training time could be achieved since short speech pauses often occur after every English syllable. Performing activation analysis pruning on top of skip convolution led to at least 30% parameter reduction in each layer according to Table 5.

Improvements across all metrics were obtained when the SE algorithms were employed for denoising. This demonstrates the effectiveness of the SE operations. Table 6 shows that the CNN-based systems demonstrated superior SE capability over DNN-IRM as they achieved better scores than DNN-IRM across all metrics and in all conditions. This is likely contributed by the sophisticated feature extraction at multiple temporal and/ or spectral resolutions embedded in the CNN approaches. There were many instances where DNN-QCM provided comparable performance with TCNN. This displays the importance of convolutional filter choice in CNN-based SE algorithms and suggests that DNN-based can still outperform CNN-based methods given the right architectural and training configuration. The evaluation metric scores obtained from all SE methods when processing Volvo noise were all almost equally high in terms of STOI as the Volvo noise is more stationary than the other noise types, making it less challenging to denoise. The performance of all SE systems deteriorated with increasing noise dominance (diminishing SNR). Overall, the performance of the proposed model is better than the other SE models. The proposed CNN-AFD achieved an average improvement of 2.05%, 4.35% and 4.03% in terms of STOI, PESQ and LSD respectively when compared to the next best-performing SE model, which is the CNN-dilated.

A trend similar to that in the seen conditions can be observed when the proposed SE approach was used to process noise types that were not part of the training dataset. The proposed method continued to show outstanding denoising performance and outperformed the baselines. This is followed by CNN-dilated. It was much more challenging for the SE systems to process noisy speech at -10 dB SNR as greater STOI, PESQ and LSD improvements can be observed when processing noisy speech at 3 dB SNR. Nevertheless, the SE systems continued to show effective denoising as refined STOI, PESQ and LSD scores were attained. The proposed method provided the largest percentage of STOI (1.30%), PESQ (4.00%) and LSD (3.76%) improvements across both unseen SNRs, demonstrating excellent SNR generalization capability as well as noise generalization capability.

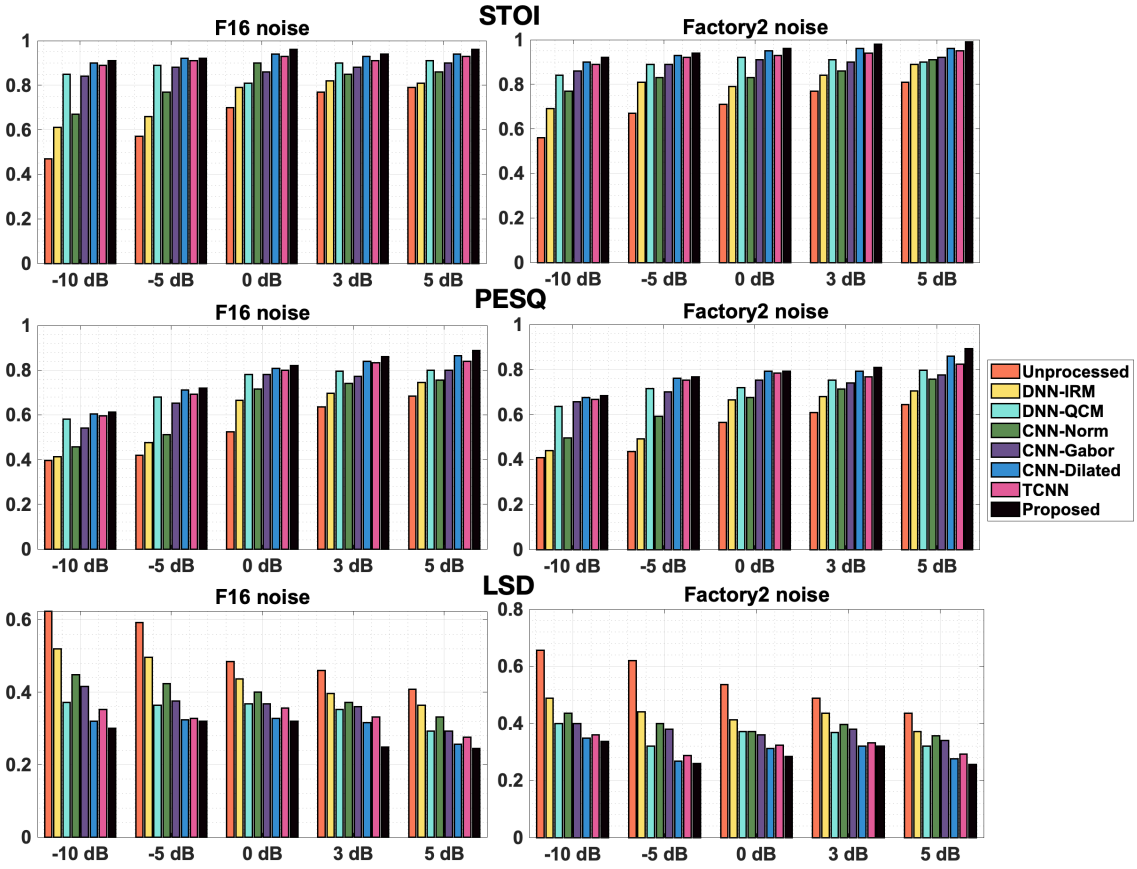


FIGURE 7. Evaluation performance comparisons of the various SE algorithms on untrained noise types ('f16' and 'factory 2' noise) and untrained SNRs (-10 and 3 dB).

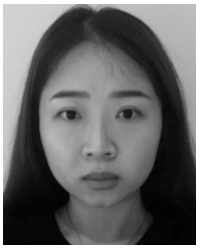
VIII. CONCLUSION

In this paper, a CNN-based SE with an adaptive filter design named CNN-AFD was presented. The CNN of the proposed SE approach was incorporated with fixed Gabor functions to extract human auditory model-inspired features in multiple temporal and spectral resolutions. The obtained Gabor feature map output was used as the basis for generating adaptive region-aware filters. A learnable module was used to predict guided masks for assigning similar feature region patterns on the Gabor feature map to the same filter. Subsequently, the back propagation-optimized guided masks were used in a filter generator module to produce specialized filters given a noisy speech input. Skip convolution and activation analysis-based pruning were explored for model compression and computation reduction. The optimization techniques employed demonstrated complementary compression performance and led to a significant reduction in model parameters and processing time while maintaining excellent speech denoising outcomes. Comparison with other deep-learning-based SE algorithms showed that the proposed approach outperformed other SE methods employing fixed convolutional filters. The proposed CNN-AFD provided the best denoising performance with average STOI, PESQ and LSD scores of 0.95, 1.82 and 0.82, respectively. Furthermore, it displayed good noise and SNR generalization capability. The proposed CNN-AFD possesses some limitations: it ignores the significance of phase information on SE performance (i.e., noisy speech phase is used at waveform reconstruction) and the adaptivity of the filter design can be an overkill in situations where the noise and speech characteristics remain constant (e.g., listening to the same speaker in the same noise condition for an extended period). Furthermore, the proposed CNN-AFD has not been designed and evaluated to consider reverberations. Future work will address these limitations and investigate the performance of the proposed CNN-AFD when presented with combinative noise. The viability of deploying CNN-AFD in hearing devices will also be examined through hardware implementation. Subjective evaluation of the CNN-AFD will be performed through human hearing tests to understand the real-life speech intelligibility and quality benefit achievable by the CNN-AFD processor.

REFERENCES

[1] L.-P. Yang and Q.-J. Fu, "Spectral subtraction-based speech enhancement for cochlear implant patients in background noise," *J. Acoust. Soc. Amer.*, vol. 117, no. 3, pp. 1001–1004, Mar. 2005.

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 6, pp. 1109–1121, Dec. 1984.
[3] S. Abdullah, M. Zamani, and A. Demosthenous, "Towards more efficient DNN-based speech enhancement using quantized correlation mask," *IEEE Access*, vol. 9, pp. 24350–24362, 2021.
[4] Y. Xu, J. Du, L.-R. R. Dai, and C.-H. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE Trans. Audio. Speech. Lang. Process.*, vol. 23, no. 1, pp. 7–19, May 2015.
[5] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement," in *Proc. Interspeech*, Sep. 2016, pp. 3768–3772.
[6] G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, "A real-time convolutional neural network based speech enhancement for hearing impaired listeners using smartphone," *IEEE Access*, vol. 7, pp. 78421–78433, 2019.
[7] G. Liu, K. Zhang, and M. Lv, "SOKS: Automatic searching of the optimal kernel shapes for stripe-wise network pruning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 12, 2022, doi: 10.1109/TNNLS.2022.3162067.
[8] W. Nogami, T. Ikegami, S.-I. O'uchi, R. Takano, and T. Kudoh, "Optimizing weight value quantization for CNN inference," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[9] Z. Shao, X. Chen, L. Du, L. Chen, Y. Du, W. Zhuang, H. Wei, C. Xie, and Z. Wang, "Memory-efficient CNN accelerator based on interlayer feature map compression," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 668–681, Feb. 2022.
[10] A. Pandey and D. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 7, pp. 1179–1188, Jul. 2019.
[11] A. Pandey and D. Wang, "Dense CNN with self-attention for time-domain speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1270–1279, 2021.
[12] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5069–5073.
[13] A. E. Bulut and K. Koishida, "Low-latency single channel speech enhancement using U-Net convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6214–6218.
[14] C. Chen, Y. Lyu, Y. Wang, J. Li, and T. Lan, "Selective kernel network with intermediate supervision loss for monaural speech enhancement," in *Proc. IEEE 21st Int. Conf. Commun. Technol. (ICCT)*, Oct. 2021, pp. 1330–1334.
[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Apr. 2014, p. 1–15.
[16] J. Z. Esquivel, A. C. Vargas, P. L. Meyer, and O. Tickoo, "Adaptive convolutional kernels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1998–2005.
[17] M. B. Er, "A novel approach for classification of speech emotions based on deep and acoustic features," *IEEE Access*, vol. 8, pp. 221640–221653, 2020.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[19] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomed. Signal Process. Control*, vol. 47, pp. 312–323, Jan. 2019.
[20] Y. Hao, A. Kucuk, A. Ganguly, and I. M. S. Panahi, "Spectral flux-based convolutional neural network architecture for speech source localization and its real-time implementation," *IEEE Access*, vol. 8, pp. 197047–197058, 2020.
[21] S.-Y. Chang, B. T. Meyer, and N. Morgan, "Spectro–temporal features for noise-robust speech recognition using power-law nonlinearity and power-bias subtraction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7063–7067.
[22] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Electr. Eng., III, Radio Commun. Eng.*, vol. 93, no. 26, pp. 429–441, Nov. 1946.
[23] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *J. Acoust. Soc. Amer.*, vol. 118, no. 2, pp. 887–906, Aug. 2005.
[24] J. Chen, X. Wang, Z. Guo, X. Zhang, and J. Sun, "Dynamic region-aware convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8064–8073.
[25] P. Scanlon, D. P. W. Ellis, and R. B. Reilly, "Using broad phonetic group experts for improved speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 3, pp. 803–812, Mar. 2007.
[26] H. Kuwabara, "Acoustic properties of phonemes in continuous speech for different speaking rate," in *Proc. 4th Int. Conf. Spoken Lang. Process.*, vol. 4, Oct. 1996, pp. 2435–2438.
[27] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.
[28] P. Singh and V. P. Namboodiri, "SkipConv: Skip convolution for computationally efficient deep CNNs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[29] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 1998, pp. 365–368.
[30] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," 2016, *arXiv:1607.03250*.
[31] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallet, and N. L. Dahlgren, "DARPA-TIMIT: Acoustic-phonetic continuous speech corpus," U.S. Department of Commerce, Washington, DC, USA, Tech. Rep. 4930, 1993.
[32] A. Varga and H. J. M. 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.
[33] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio. Speech. Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Dec. 2011.
[34] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2, May 2001, pp. 749–752.
[35] P. J. Wolfe and S. J. Godsill, "Towards a perceptually optimal spectral amplitude estimator for audio signal enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, Aug. 2000, pp. 821–824.
[36] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7092–7096.
[37] Z. Ouyang, H. Yu, W.-P. Zhu, and B. Champagne, "A fully convolutional neural network for complex spectrogram processing in speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5756–5760.
[38] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jul. 2019, pp. 6875–6879.



SALINNA ABDULLAH (Graduate Student Member, IEEE) received the M.Eng. degree in electronic engineering with computer science from University College London (UCL), London, U.K., in 2017, where she is currently pursuing the Ph.D. degree with the Bioelectronics Group, Department of Electrical and Electronic Engineering. She was a recipient of an EPSRC Industrial Strategy Studentship. Her research interests include FPGA design, efficient application of speech enhancement, image processing, and deep-learning methods in wearable devices. She was awarded the Cisco Prize for Most Outstanding Female Engineer during her final year of undergraduate study.

MAJID ZAMANI (Member, IEEE) received the M.Sc. degree in microelectronics from Islamic Azad University Science and Research Branch, Tehran, Iran, in 2011, and the Ph.D. degree from University College London (UCL), London, U.K., in 2017. He is currently a Research Associate with the Bioelectronics Group, UCL. His research interests include design and fabrication of advanced and energy-efficient computational systems utilizing pattern recognition, machine learning, and computer vision algorithms, especially for wearable and implantable biomedical applications. He was a recipient of the Overseas Research Scholarship and the UCL Graduate Research Scholarship to pursue his Ph.D. degree. He was also a recipient of the Best Researcher M.Sc. Student Award.

ANDREAS DEMOSTHENOUS (Fellow, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 1992, 1994, and 1998, respectively. He is currently a Professor with the Department of Electronic and Electrical Engineering, UCL, where he leads the Bioelectronics Group. He has made outstanding contributions to improving safety and performance in integrated circuit design for active medical devices, such as spinal cord and brain stimulators. He has numerous collaborations for cross-disciplinary research, both within the U.K. and internationally. He has authored over 350 articles in journals and international conference proceedings, several book chapters, and holds several patents. His research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal processing applications. He is a fellow of the Institution of Engineering and Technology and a Chartered Engineer. He was a co-recipient of a number of best paper awards and has graduated many Ph.D. students. He was an Associate Editor, from 2006 to 2007 and the Deputy Editor-in-Chief, from 2014 to 2015 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS; and an Associate Editor, from 2008 to 2009 and the Editor-in-Chief, from 2016 to 2019 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He is an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS. He serves on the International Advisory Board for *Physiological Measurement*. He has served on the technical committees for a number of international conferences, including the European Solid-State Circuits Conference and the International Symposium on Circuits and Systems.