IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 9, 2019, accepted June 5, 2019, date of publication June 12, 2019, date of current version June 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922370

A Real-Time Convolutional Neural Network Based Speech Enhancement for Hearing Impaired Listeners Using Smartphone

GAUTAM S. BHAT[1], NIKHIL SHANKAR[1], CHANDAN K. A. REDDY[2], (Student Member, IEEE), ISSA M. S. PANAHI[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson TX-75080, USA
[2]IC3-AI, Microsoft Corporation, Redmond, WA 98052, USA

Corresponding author: Gautam S. Bhat (Gautam.Shreedhar-Bhat@utdallas.edu)

The National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) under award number 1R01DC015430-03 supported this work. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

ABSTRACT This paper presents a Speech Enhancement (SE) technique based on multi-objective learning convolutional neural network to improve the overall quality of speech perceived by Hearing Aid (HA) users. The proposed method is implemented on a smartphone as an application that performs real-time SE. This arrangement works as an assistive tool to HA. A multi-objective learning architecture including primary and secondary features uses a mapping-based convolutional neural network (CNN) model to remove noise from a noisy speech spectrum. The algorithm is computationally fast and has a low processing delay which enables it to operate seamlessly on a smartphone. The steps and the detailed analysis of real-time implementation are discussed. The proposed method is compared with existing conventional and neural network-based SE techniques through speech quality and intelligibility metrics in various noisy speech conditions. The key contribution of this paper includes the realization of CNN SE model on a smartphone processor that works seamlessly with HA. The experimental results demonstrate significant improvements over the state-of-the-art techniques and reflect the usability of the developed SE application in noisy environments.

INDEX TERMS Convolutional neural network (CNN), speech enhancement (SE), hearing aid (HA), smartphone, real-time implementation, log power spectra (LPS).

I. INTRODUCTION

Speech understanding in adverse noisy environments is a major problem even for a normal hearing person and this problem escalates for a hearing impaired listener. Statistics reported by National Institute on Deafness and other Communication Disorders (NIDCD) show that there are more than 360 million people across the world which includes 15% of American adults (37million) aged 18 and over, who suffer from some kind of hearing loss [1]. Researchers have developed viable solutions in the form of individual hearing devices such as hearing aid devices (HADs) and Cochlear Implants (CI). HADs compensate for the loss in audibility due to the modest of sensorineural hearing loss. Although, HA users achieve near-to-normal hearing in controlled acoustic conditions, it becomes extremely challenging to have a normal conversation in many real world noisy environments. This is because, the performance of the HADs degrades in the presence of different background noise [2]. SE algorithms on HADs try to overcome this problem by improving speech quality and intelligibility of the noisy speech. Therefore, SE is a vital block in the HAD signal processing pipeline. HADs have limited computational capabilities due to their size, processor, and power consumption. Due to these limitations, it is impractical to implement complex yet indispensable signal processing algorithms on HADs in order to improve their performance. HA manufacturers have come up with external microphones in the form of a pen or a necklace to improve the performance of a HAD by capturing the signal at higher signal to noise ratios (SNRs). However, the usage of these auxiliary devices are seldom due to their high cost, limited power and portability, especially in formal and social settings. An alternate solution is to use smartphone as an assistive tool for HADs as they are portable, have better battery life,

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

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

ubiquitous as large population possess smartphone and use them almost everywhere. Smartphones are equipped with superior ARM multi-core processors which can run complex algorithms. In this type of setup, the smartphone captures the noisy speech. The SE algorithm running on the processor of the smartphone suppresses the background noise and the enhanced speech is sent from smartphone to HA through wire or wireless connection via low-latency Bluetooth [3]. Recent studies [4], [5] show that, smartphones such as iPhone (iOS) and Pixel(Android) can run SE algorithms on a smartphone to enhance the overall quality of the speech perceived by the users with hearing loss.

SE has been widely studied for decades. We can find extensive studies where SE algorithms are developed to improve the performance of HADs in the presence of background noise [6]. However, the prime challenge in SE is to suppress the background noise without inducing any sort of speech distortion which would degrade the speech perception. SE algorithms like spectral subtraction [7] and statistical model based methods proposed by Ephraim and Malah [8], [9] are studied over years and have been successful in the removal of noise if the noise is approximately stationary and at high SNR levels. There are some computationally efficient alternatives for [8] and [9] which are proposed in [10] and [11] that can be implemented in real time. Recent algorithms such as optimally modified Log-spectral amplitude (OM-LSA) [12] and non-negative matrix factorization (NMF) [13] can achieve better performance at the cost of higher computational complexity. But, these aforementioned algorithms are based on key apriori knowledge or assumptions between the speech and nature of underlying noise. Therefore, due to these assumptions, most of the algorithms induce musical noise and do not acceptably improve performance in complex real world scenarios, especially at low SNRs. Several studies [14], [15] include microphone array and beamforming based SE techniques for better noise suppression. But, as the number of microphones increase, so does the computational power, which makes it practically impossible to implement on HADs or on smartphones which usually allow using a maximum of two microphones. It should be noted that promising results have been observed in [4] and [5] with regard to real-time operation of conventional SE algorithms and their efficient implementations on smartphones using no external hardware of any kind.

Recently, supervised learning approaches based on ideal binary mask (IBM) [16] and ideal ratio mask (IRM) [17] have been proposed for SE in HADs. More recently, SE using deep neural networks (DNNs) have been appeared in literature, as they do not require any such apriori conditions to be met when applying the SE. In DNN based SE approaches [18]–[20], a regression model is trained to predict the estimate of clean speech features from the noisy speech features. Based on the mask estimating DNNs (masking) or direct feature estimating DNNs (mapping), the training targets are selected for different applications [21]. In [22], a mapping based DNN architecture to predict the clean log-power spectra (LPS) features from noisy LPS features was developed. DNN acts as a mapping function to learn the relationship between clean and noisy speech features. Recent studies also point towards the use of convolutional neural network (CNN) for SE as they typically reflect correlations of speech in time and are invariant to translational speech variance [23]. In [24], results show that CNNs can perform better than the fully-connected DNN architecture and recurrent neural networks (RNNs) in SE while utilizing lesser number of learnable parameters. Researchers have also considered CNN-based end-to-end approach to SE that requires just the raw audio data [25]. Linguistic training and testing of SE based on CNN [26] concluded that the performance of monolingual trained models was on par with multilingual models which makes it better than DNNs. Several features like LPS, Mel Frequency cepstral Coefficients (MFCC), Gammatone Frequency cepstral coefficients (GFCCs) and IBM were employed in multi-objective learning for SE with a DNN architecture to improve the performance in terms of quality and intelligibility of speech [27], [28]. However, most of the aforementioned methods are application specific and the approaches have long inference times. This limits their utilization in a real-time frame-based speech processing applications, such as that needed for using a smartphone as a standalone hardware platform. Due to the fact that neural network architectures normally achieve better performance when defined to be as large and as deep as possible, most of the techniques use deep and large networks and do not consider real-time limitations in practice. The speed of the neural networks depends on how many hyper parameters and the layers we have and what operations are run on the nodes. Therefore, in order to achieve smartphone implementation for real-time operations, the architecture of the network, the feature sets and the operations on the network that play a vital role must be revisited and modified as needed.

In this paper, a multi-objective learning framework which improves the performance of the CNN is developed. LPS and Log Mel-filterbank energy are used as primary and the secondary targets for CNN. The joint optimization of these two features improves the performance of the proposed CNN architecture in predicting the primary target LPS. The developed neural network architecture is small with low inference time, allowing it to implement on low-latency smartphone devices. The proposed method works in real-time as an application on a smartphone which can be used as an assistive tool for hearing impaired listeners. The novel contribution made in this work lies in the development of a practical CNN architecture for SE and its real-time operation as an app running on smartphone platforms with no additional or external hardware. It also includes the use of secondary targets to learn and improve the performance of the CNN architecture. The proposed SE method is computationally efficient, and implementable in practice. Objective and subjective evaluations of the proposed method show substantial improvement in quality and intelligibility of enhanced speech in various

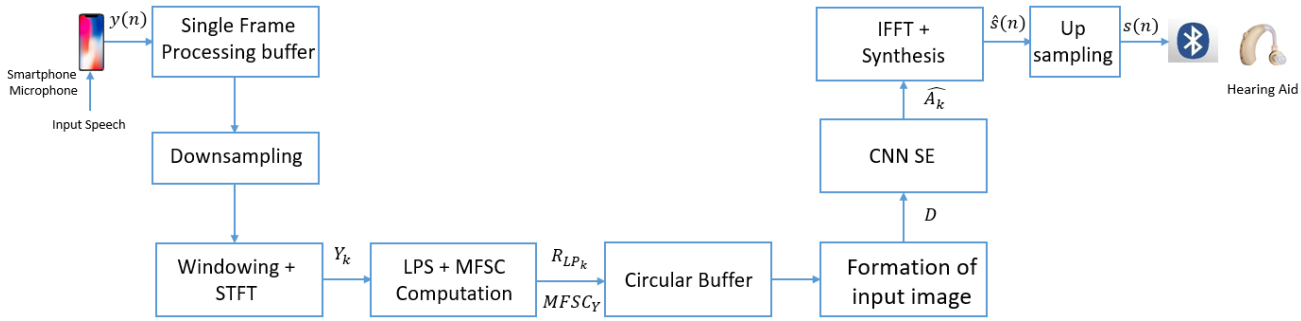

FIGURE 1. Block diagram of the proposed method and illustration of the real time processing modules used in the developed CNN-based SE application.

noisy environment reveal the overall benefits and usability of the developed CNN-based SE algorithm for the end-users.

The remainder of this paper is organized as follows. In Section II, we describe the signal model, the features utilized in the proposed algorithm and the CNN architecture for de-noising the speech signal. Section III describes the real-time implementation of the developed CNN-based SE method on smartphone. Analysis over real time characteristics and experimental results are presented in Section IV. Conclusion is in Section V.

II. PROPOSED SPEECH ENHANCEMENT METHOD

In this section, we describe the developed speech processing method. Fig. 1 shows the block diagram of the proposed method. (Function of each block is discussed further in the Section II and III).

A. PROBLEM FORMULATION

Speech enhancement (SE) techniques often consider the additive mixture model for noisy speech $y(n)$, with clean speech $s(n)$ and noise $v(n)$, as

$$y(n) = s(n) + v(n) \tag{1}$$

The noisy k^{th} short time fourier transform (STFT) coefficient of $y(n)$ for frame λ is given by,

$$Y_k(\lambda) = S_k(\lambda) + V_k(\lambda) \tag{2}$$

where S and V are the clean speech, and noise STFT coefficients respectively. In polar coordinates, (2) can be written as,

$$R_k(\lambda)\, e^{j\theta_{Yk}(\lambda)} = A_k(\lambda)\, e^{j\theta_{S_k}(\lambda)} + B_k(\lambda)\, e^{j\theta_{Vk}(\lambda)} \tag{3}$$

where $R_k(\lambda)$, $A_k(\lambda)$, $B_k(\lambda)$ are magnitude spectra of noisy speech, clean speech, and noise respectively. $\theta_{Yk}(\lambda)$, $\theta_{S_k}(\lambda)$, $\theta_{Vk}(\lambda)$ are the phase spectra of noisy speech, clean speech and noise respectively. The goal of a speech enhancement is to determine an estimate of the speech spectrum $\widehat{A_k}$. As we know from the literature that the phase is perceptually unimportant [30], we consider the noisy phase for reconstruction. Hence the estimate of clean speech after reconstruction can be written as,

$$\widehat{S_k}(\lambda) = \widehat{A_k}(\lambda)\, e^{j\theta_{Yk}(\lambda)} \tag{4}$$

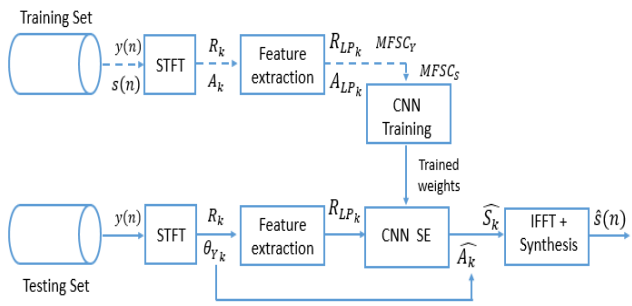

FIGURE 2. Speech Enhancement system used in the entire experiment. The dashed lines indicate the training phase and the solid lines indicate the testing phase.

B. PRIMARY FEATURES – LOG POWER SPECTRA

The proposed CNN system is similar to the one in [19] and the overall working of the method can be seen in Fig. 2. We consider processing only the magnitude part of the spectrum, because phase is of less importance for speech intelligibility [29]. Therefore, as a mapping based CNN network, the log power features of noisy and clean recordings are computed in our approach for training. Only the first half of LPS is considered as the primary target in the proposed CNN technique since the magnitude spectrum is symmetric. Thus,

$$R_{LP_k}(\lambda) = \ln[R_k(\lambda)^2] \tag{5}$$

$$A_{LP_k}(\lambda) = \ln[A_k(\lambda)^2] \tag{6}$$

where, $R_{LP_k}(\lambda)$ and $A_{LP_k}(\lambda)$ are the log power spectra of noisy and the clean speech at every k^{th} frequency bin for current time frame λ. Here the $A_{LP_k}(\lambda)$ can be visualized as target vectors and $R_{LP_k}(\lambda)$ as inputs for training and testing.

C. SECONDARY FEATURES – LOG MEL - FILTERBANK ENERGY

The Log Mel-filterbank energy is considered as the secondary features for the CNN. Experiments show that the human ear follows the Mel frequency scale for the perception of the frequency components in the speech which can be seen as a linear frequency spacing below 1kHz and logarithmic spacing above 1kHz [30]. Therefore, linearly spaced filters at low frequency and logarithmic at high frequencies are used in our approach to capture the phonetically important characteristics (voiced and unvoiced) of the speech. In [31] and [32], the Log

Mel-filterbank have shown promising results in increasing the classification accuracy when treated as input to the neural network. The Mel-scaled STFT spectrograms, in which the audio data represented in image format, consistently performed well as inputs to CNNs for audio classification tasks [33]. The performance of CNN with Mel-scaled STFT spectrogram as an input feature [33] was better when compared to the CNNs with input features like linear-scaled STFT spectrograms, Constant-Q transform (CQT) spectrogram, continuous Wavelet transform (CWT) scalogram and MFCC cepstrogram.

In addition, the Log Mel-filterbank energy used here is computationally efficient for real-time implementation on a smartphone. Since we already have LPS as our primary feature, we cannot afford the secondary feature to be computationally complex. This is because, it can lead to an added delay when combined with the primary feature for real time application on smartphone. The Log Mel energy spectrum represents the short term energy of an audio signal in the Mel-frequency scale made up of Mel-frequency spectral coefficients (MFSC).

Mel-frequency analysis of speech is based on human perception, the commonly used formula to approximately reflect the relation between the Mel-frequency and the physical frequency is [34],

$$M = 1125\, ln\,(1 + f/700) \tag{7}$$

$$\hat{f}(n) = 700\,(\exp(M(n)/1125) - 1 \tag{8}$$

Here, the (7) computes the Mel-frequency coefficient from frequency f in Hertz. Eq. (8) is the inverse to go from Mel domain back to physical frequency, where $M(n)$ is the nth frequency coefficient in Mel domain.

Typically, to compute the MFSCs of a speech signal, the signal is split into 20-40 ms frames. This is because, the shorter frames do not provide enough samples for accurate spectral estimation and if it is longer; the signal may change too much within the frame. The frames are 50% overlapped frames and are windowed (e.g. Hamming window) to counteract the spectral leakage that could occur with DFT computation. Fast Fourier transform (FFT) is applied to the windowed audio frame to compute the frequency spectrum. Then the power spectrum is computed and only the first half is utilized while the magnitude spectrum is mirrored in frequency.

Filterbank is computed by applying L triangular filters (typically 26-40) [30], [34] on Mel-scale to calculate MFSCs. A lower and upper frequency is specified to limit spectrogram. Ideal values are 300Hz for the lower and 8000Hz for the upper frequency. Of course, if the speech is sampled at 8000Hz our upper frequency is limited to 4000Hz. Using the lower and upper frequencies, the lower and the upper Mel-frequency coefficients (Mels) are calculated from (7). L linearly spaced frequencies are obtained between lower and upper Mels in the Mel-domain. These linearly spaced frequency coefficients in the Mel domain are then converted

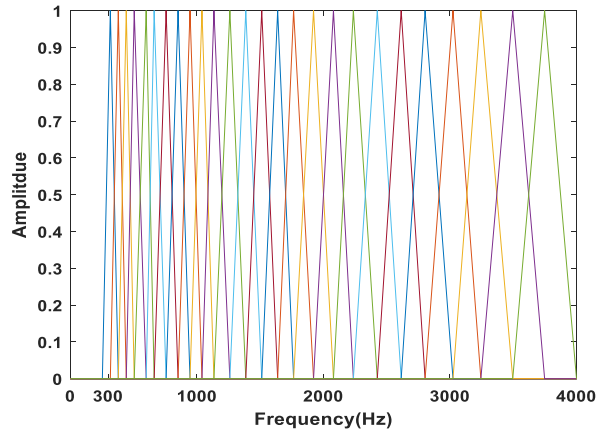

FIGURE 3. The figure demonstrates Mel-filterbank consisting 26 triangular filters. These filters are linearly spaced in Mel domain and non-linearly spaced in frequency domain.

to the non-linearly spaced frequencies in frequency domain using (8). The non-linearly spaced frequencies $\hat{f}(n)$ act as center frequencies for L triangular filters. To calculate the FFT bins corresponding to the center frequencies, we need to know the FFT size (N) and the sampling frequency (F_s), given by (9). Finally, the Mel-spaced filterbank is then created by (10),

$$h(n) = \frac{(N+1)\hat{f}(n)}{F_s}, \quad n = 0, 1 \ldots L+1 \tag{9}$$

$$G_n(k) = \begin{cases} 0 & k < h(n-1) \\ \dfrac{(k - h(n-1))}{h(n) - h(n-1)} & h(n-1) < k \le h(n) \\ \dfrac{h(n+1) - k}{h(n+1) - h(n)} & h(n) < k \le h(n+1) \\ 0 & k > h(n+1) \end{cases}$$
$$k = 1, 2 \ldots N/2$$
$$n = 1, 2 \ldots L \tag{10}$$

Here, $G_n(k)$ represents the amplitude n^{th} filter at k^{th} frequency bin. $h(n)$ denotes the list of frequency bin values of $L + 2$ Mel-spaced frequencies. The triangular filters of the filterbank observed in the frequency domain is illustrated in Fig 3.

The filterbank is then multiplied with the computed power spectrum of the noisy speech and the clean speech to generate inputs for training and testing and the target vectors respectively. The logarithmic summation of each individual product is taken to compute the Log Mel energy spectrum, given by (11-12),

$$MFSC_Y(\lambda, n) = \log\left(\sum_{k=0}^{N/2} G_n(k)\, R_k(\lambda)^2\right), \quad n = 1, 2 \ldots L \tag{11}$$

$$MFSC_S(\lambda, n) = \log\left(\sum_{k=0}^{N/2} G_n(k)\, A_k(\lambda)^2\right), \quad n = 1, 2 \ldots L \tag{12}$$

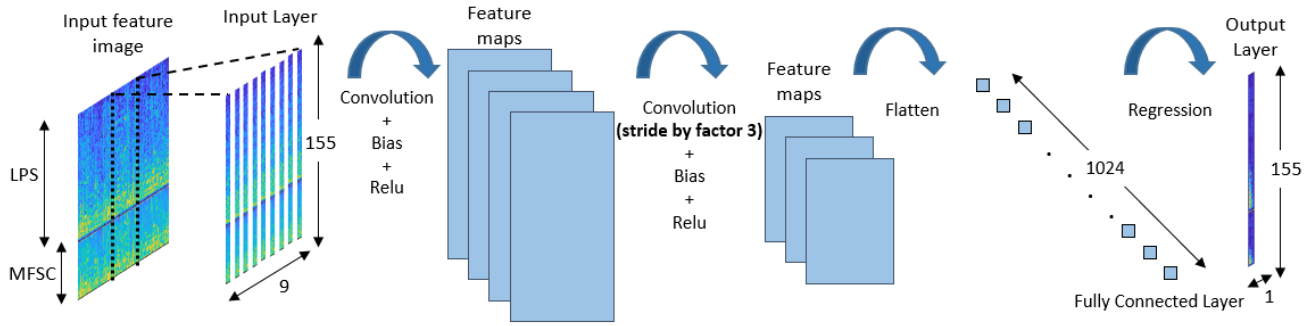

FIGURE 4. Schema of the proposed CNN-based SE model architecture. The CNN consists 3 components. Input image is first convolved with kernels. The final convolutional layer obtained by stride of 3 is flattened and fed into fully connected layer. Finally, output is fed to the linear layer.

The noisy and the clean speech Log Mel-filterbank energy features per time frame λ are given by (11) and (12). Here, $R_k(\lambda)^2$ and $A_k(\lambda)^2$ denote the noisy and the clean speech power spectrum estimate. The $MFSC_S(\lambda, n)$ can be visualized as target vectors and $MFSC_Y(\lambda, n)$ as inputs for training and testing.

After finding the LPS features and Log Mel-filterbank energy features per time frame, they are concatenated across the time frames to create an image. This image is then fed as an input to the CNN during the training and testing phase. We believe that the estimate of clean LPS features would be better predicted with a MFSC constraint imposed at the output layer. The MFSC estimates were used to improve the overall performance of the CNN network and were not used for signal reconstruction. The reconstructed enhanced speech was obtained from the clean LPS estimate $\widehat{A_k}(\lambda)$ and the noisy speech phase information as shown in (4).

Similar work in [27] showed that to improve the performance of SE based on DNN, we could make use of different input features. However, the motivation of our work is multi-objective learning with a different neural network, novel architecture in input, hidden and output layers, and also its smartphone real-time implementation.

D. CNN FOR SPEECH ENHANCEMENT

A regression based decision which estimates the LPS and MFSC of the clean speech is done by the proposed convolutional neural network (CNN). CNNs use a matrix or an image as an input and they are processed by their hidden layers performing convolution and pooling functions. Finally, before inferring the decision, they are connected with fully connected layers similar to traditional DNNs which use back propagation methods.

Typically, in CNNs, a small region of an image can be convoluted by a series of weighted learnable filters (also known as kernels) which can form a convolutional layer. These kernels are repeated over the entire input space. After every forward pass, each kernel generates a feature map that describes simple features of an image. These feature maps are fed to the maxpooling layer for dimensionality reduction or to reduce the resolution. Furthermore, based on the application, the maxpooled output can be connected to convolutional layers as adding more convolutional layers enables the network to learn complex features of an image. The output from convolutional or maxpool layers are then flattened and connected to fully connected layers. Thus, the classification or regression output is obtained by connecting the fully connected layer to a non-linear output layer.

When used for speech enhancement, the overall network remains similar. However, the extracted features from the audio input is represented as an image and the output is treated as a regression problem and not classification. Here, the network learns to recognize crucial time-frequency auditory features, such as formants. Since each part of the feature map is processed by the same kernel, the CNN is invariant to translational variance. This enables the network to learn and process speech from different gender and language which usually have different pitch and therefore different fundamental frequencies.

The proposed CNN structure illustrated in Fig. 4 consists combined LPS and MFSC features as input to the network. The input vector consists of $\frac{N}{2}+1$ LPS features and L MFSC features per time frame, then they are concatenated to create $D \times T$ image, where $D = \left(\frac{N}{2}+1+L\right)$ and T is the total number of frames considered in a spectrum. Multiple frame expansion [19] is considered with T being 9 consecutive noisy frames; 8 preceding plus the current frame. The target vector is the current clean frame. Usually T is considered to be a large number for capturing the temporal details. However, in order to reduce computational complexity especially for real time implementation, T is taken to be 9 frames. Network consists of convolutional layers, and a fully-connected layer. The convolutional kernels extract the local patterns from input image in both time and frequency.

Let z be a convolutional kernel of size $a \times b$. We define a feature map u_z to be the convolution of the input with kernel z, followed by an elementwise nonlinear mapping.

$$u_{\mathbf{z}}(\boldsymbol{Q}) = \sigma(\mathbf{Q} * \mathbf{z})$$

For brevity, the $D \times T$ image is considered to be $\boldsymbol{Q}.*$ is the convolution operation.

In order to aid the network to learn complicated and nonlinear functional mapping between the inputs and the output labels, activation functions are used in each hidden

layers. As it has been verified to be effective in solving the gradient vanishing problem in practice [35], we choose rectified linear unit (ReLU) as activation function, $\sigma(.)$, i.e.

$$\sigma(a) = \max\{a, 0\}$$

Each convolutional kernel $\mathbf{z}$ produces a 2D feature map, and we apply η separate convolutional kernels to the input image $\boldsymbol{Q}$ leading to a collection of 2D feature maps $u_{\mathbf{z}}^{\eta}(\boldsymbol{Q})$. This locality approach allows the CNN to learn the characteristics of the spectrum. Also, the convolutional kernels can map the local temporal structure of the utterances, which generates effective temporal behavior. To obtain ideal reconstruction of speech signal, we have to ensure that the input and the final prediction of the model to have same length in time dimension. Zero padding is applied to the input $\boldsymbol{Q}$ before convolution. Zero padding also assures that the feature maps u_z has same dimension as that of the input.

The maxpool layer can be applied due to the fact that the spectrograms replicate local similarities in adjacent frequency bins. This design will reduce the number of parameters and the computations needed for the further layers. It is also worth pointing out that instead of pooling layer, we can use a stride of size equal to the pooling factor along the frequency domain while convolving the input $\boldsymbol{Q}$ and kernel z. In the proposed method, we make use of stride of size 3 to make the network computationally efficient without losing much of prediction accuracy. The complexity in real-time also decreases as there are no additional maxpool layers. Finally, before the fully connected layer, a second convolution layer with $\eta/3$ number of convolutional kernels is applied as the dimension is reduced by using the stride of size 3. The output layer has D neurons and uses a linear activation function to map the predicted output features. (CNN SE architecture is explained in Section IV.)

III. REAL–TIME IMPLEMENTATION ON SMARTPHONE

A. TOOLS AND STEPS INVOLVED FOR SMARTPHONE IMPLEMENTATION

The proposed CNN-based SE algorithm was trained and tested using the input image data. The LPS and MFSC features for noisy and the clean speech were extracted and the input images were created using MATLAB. The CNN modeling and the training which are offline procedures, were performed using Tensorflow in Python [36]. The main reason behind using Tensorflow was that it offered a special tool called TensorFlow Lite for running deep learning models on mobile and embedded platforms [37]. The C++ API provided by TensorFlow Lite can be used on smartphones to run the inference only part of the CNN. The TensorFlow Lite converter and the TensorFlow Lite interpreter libraries have been provided to facilitate model deployment on a mobile device and using the compressed model for on-device inference. Google cloud Platform (GCP) and Chameleon Cloud Computing were used to train the CNN-based SE. All the trained weights had to be converted to constants to deploy the trained model on smartphone. The trained weights were saved to a file (.pb extension) by freezing the model. Freezing removed training, backpropagation and regularization layers and inference only structure was taken by saving all the required weights. This model can be used for implementing on a smartphone and operating in real-time. The Tensorflow libraries mentioned above were used to run the inference part of the frozen model. The size of the frozen model can be varied from several bytes to gigabytes depending on the model architecture.

The LPS and MFSC feature extraction and image formation for smartphone application were coded in C. Xcode [38] was used for coding and debugging of the SE algorithm. For deployment on iPhone, the GUI was coded in Objective C and to carry out input/output (i/o) handling, Core Audio [39], an open source library from Apple, was used.

B. REAL – TIME SE PROCESSING

In this work, iPhone 7 and iPhone X smartphones running iOS 12.0.1 operating system have been considered, offering the smartphone platform as an assistive device to HA. Though smartphones come with 2 or 3 built-in microphones, manufacturers only allow default microphone (Fig. 6) on iPhone to capture the audio data. The audio signal is processed and the enhanced signal is transmitted to the HADs. The HADs can be connected to smartphone either by a wire or wirelessly via Bluetooth. The smartphone device considered for implementation had an M3, T4 HA Compatibility rating and meets the requirements set by Federal Communications Commission (FCC).

The optimal parameters for the proposed SE constitute 8 kHz sampling frequency with a processing frame size of 32 ms with 50% overlap. To achieve the lowest latency audio setup on iOS smartphones, it is required to read and write audio data samples at a sampling rate of 48 kHz. This latency is related to i/o of the smartphone. Therefore, for real-time processing, the input is captured at 48 kHz giving 1536 samples for each 32ms frame. The frames are then down sampled to 8 kHz by low-pass filtering and decimation factor of 6. This produces a down sampled frame of size 256 samples which is again 32ms in time. A 256-point FFT is computed for each frame. Considering only the first half of magnitude spectra, 129 LPS features are generated. 26 MFSC features are generated by considering the 26 triangular filters of filterbank. We note that the FFT size and the filter number considered here are set to be as small as possible to reduce the computation and preserve the efficiency of the proposed method. Therefore, we get a total of 155 features per frame out of which 129 are LPS and 26 are MFSC. Once these features are generated, a circular buffer is created to collect the features from 9 processing frames (i.e. 8 preceding frames plus current frame) to process to process. This creates an input processing image. The circular buffer works in a First in first out (FIFO) manner as shown in Fig. 5. This process continues for the entire signal duration. Thus, we continuously get a processing image as an input to the CNN. Finally, we get a processed output frame of size 155×1. The predicted

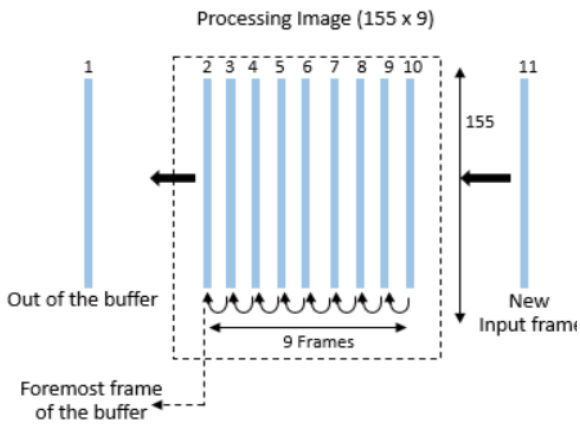

FIGURE 5. Buffer used to create input processing image. Here the frame 10 is the current processing frame. 2-9 are the preceding frames and 1 has left the buffer. When frame 11 comes in, frame 2 leaves the buffer.

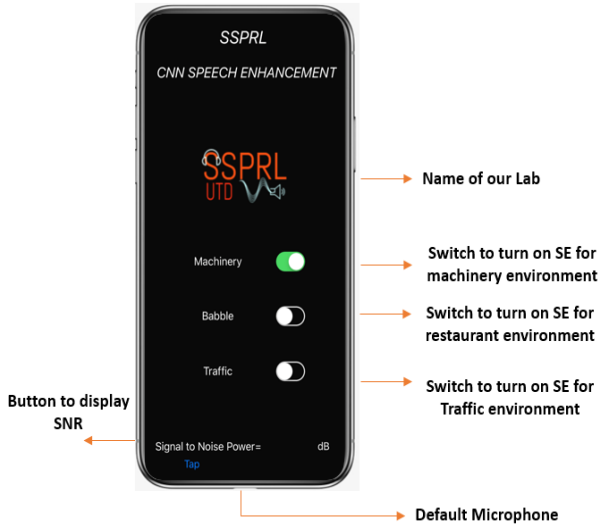

FIGURE 6. Snapshot of developed CNN-based SE application running on a smartphone.

LPS and the phase of the noisy input frame is utilized to reconstruct the signal.

The GUI of the app is displayed in Fig. 6. When the switch buttons shown are in 'OFF' mode, the application merely plays back the audio through the smartphone without processing it. Switching 'ON' the button on the touch screen panel of the smartphone enables SE module to process the incoming audio stream by applying the proposed noise suppression algorithm.

There are three models implemented on the smartphone based on three different real world noisy environments; machinery, restaurant/babble/crowd, and traffic background noise types. Smartphone user can select the either of the three buttons to turn on the SE based on the nearest environment they are in. We have to note that we have considered mainly the outdoor noise and not indoor noises like fan, Vacuum cleaner etc. Once the CNN-based SE app is activated, the enhanced signal is played back through the HADs. Initially when the switch is turned on, no processing takes place

TABLE 1. Architecture of the proposed CNN-based SE. Illustration of the number of layers and the number of nodes in each layer.

Layers	Number	Number of nodes/kernels
Hidden Layers	3	
Input	1	155 x 9
Convolutional	2	129 , 43
Fully Connected	1	1024
Output	1	155

TABLE 2. Illustration of the effect of dropout on the proposed CNN-based SE model. Speech data with Machinery noise at 0db SNR was considered.

Dropout (%)	PESQ	LSD	SegSNR
0	2.1855	11.9056	6.7691
10	2.1684	11.6541	6.7420
20	2.1062	11.9294	6.6951
30	2.0050	12.2047	6.6240

for about 2 seconds to create the circular buffer. After that, the algorithm runs seamlessly on a smartphone in real time. We use live listen device [40] to stream the data from iPhone to the HAD. The audio streaming is encoded for Bluetooth Low Energy (BLE) consumption. The processing time for a frame of 32 ms is 14ms. The computationally efficiency of the proposed algorithm is discussed in the experimental analysis in Section IV.

IV. EXPERIMENTAL ANALYSIS, RESULTS AND DISCUSSION

A. CNN ARCHITECTURE

The proposed CNN architecture has 3 hidden layers, 2 convolutional layers, and 1 fully connected layer. The same network has been used for online and offline evaluations in this paper.

There exists 129 LPS features and 26 MFSC features per time frame, and since we need 9 consecutive frames, the input layer is structured as 9 feature maps of size 155 x 1. Therefore, the input image size is 155x9 ($\boldsymbol{Q} = D \times T$). The first convolutional layer uses 129 feature maps, η, to avail superior learning of 129 LPS features. The second convolutional layer uses 43 feature maps $\eta/3$. As the second convolutional layer connects to a fully connected layer, we use less number of feature maps in order to reduce the computational complexity. The kernel for both convolution layers has a size 5×1 ($a\times b$). The convolution is done with a stride of 1 for the first convolution layer and the stride size of 3 for the latter. The fully-connected layer has 1024 neurons. As discussed before, all the activation functions are ReLU and the output layer with linear activation function has 155 neurons. The resulting network has around 9 million learnable parameters The CNN architecture utilized is given in Table 1. Adam optimization algorithm [41] was used with mean squared error as the loss

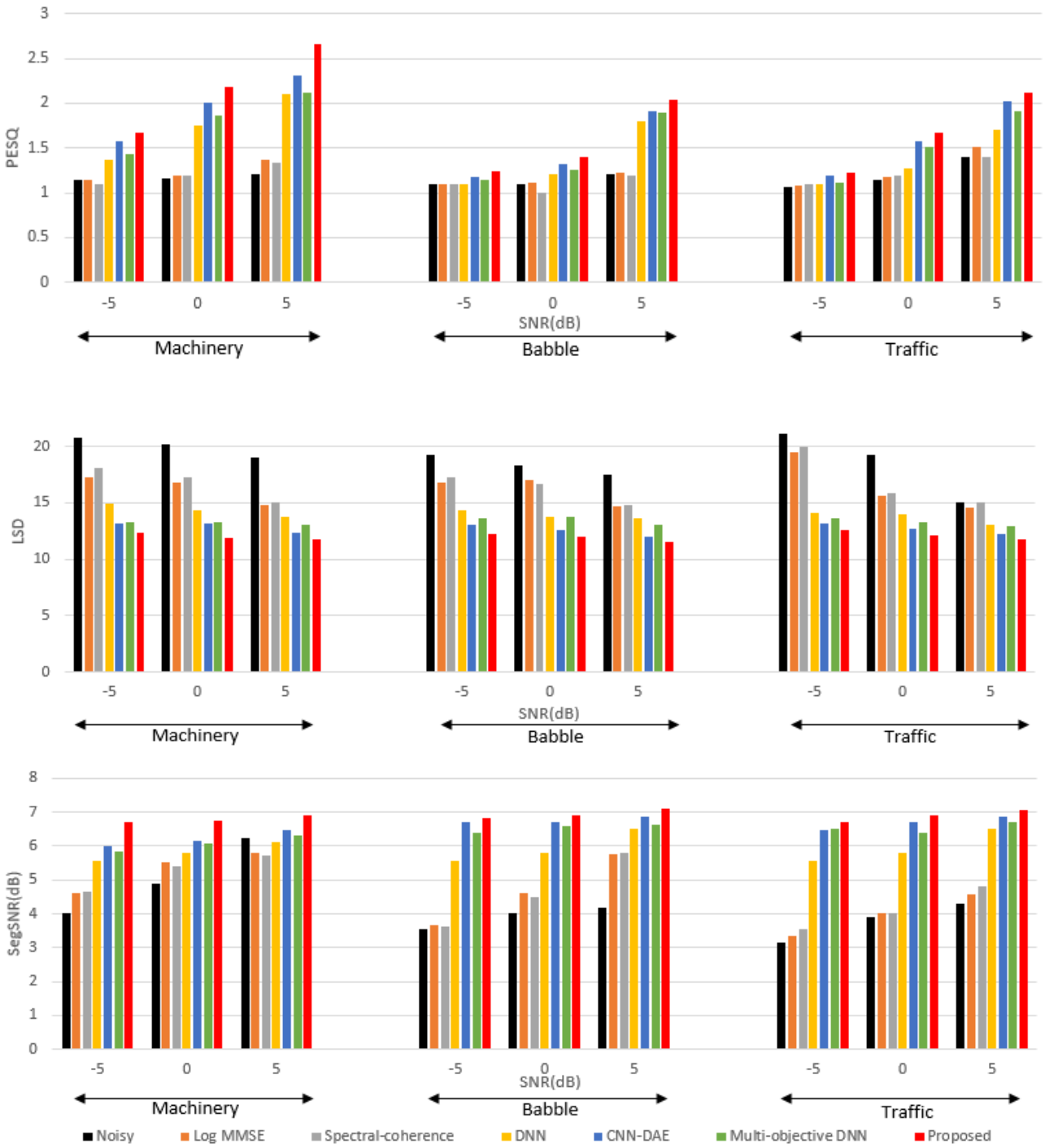

FIGURE 7. Comparison of average performance of PESQ, LSD, and SegSNR on seen and unseen data for different SE methods. Machinery, multi-talker babble and traffic noise at SNR levels of −5, 0 and +5dB are considered. The results show that the proposed CNN-based SE method outperforms the other SE methods.

function to train the CNN model. All the training vectors which include weights and biases for all the nodes and kernels were initialized with a truncated normal distribution with zero mean and a standard deviation of 0.05. The model was trained for 20 epochs with 1125 iterations per epoch.

The learning rates were set to 10^{-4} for first 10 epochs and 10^{-5} for the rest. A 10-fold cross-validation was used with a single fold left-out for testing and the rest used for training.

B. OFFLINE OBJECTIVE EVALUATION

To train and evaluate the developed CNN-based SE, the noisy speech files were created by adding speech and noise at 3 different SNR levels of −5,0 and +5 dB. The speech corpus used for evaluation was the combination of HINT, TIMIT and LibriSpeech corpus [42]. The noise dataset used was the DCASE 2017 challenge dataset [43] that consists of 15 different background noise environments. Here, all the

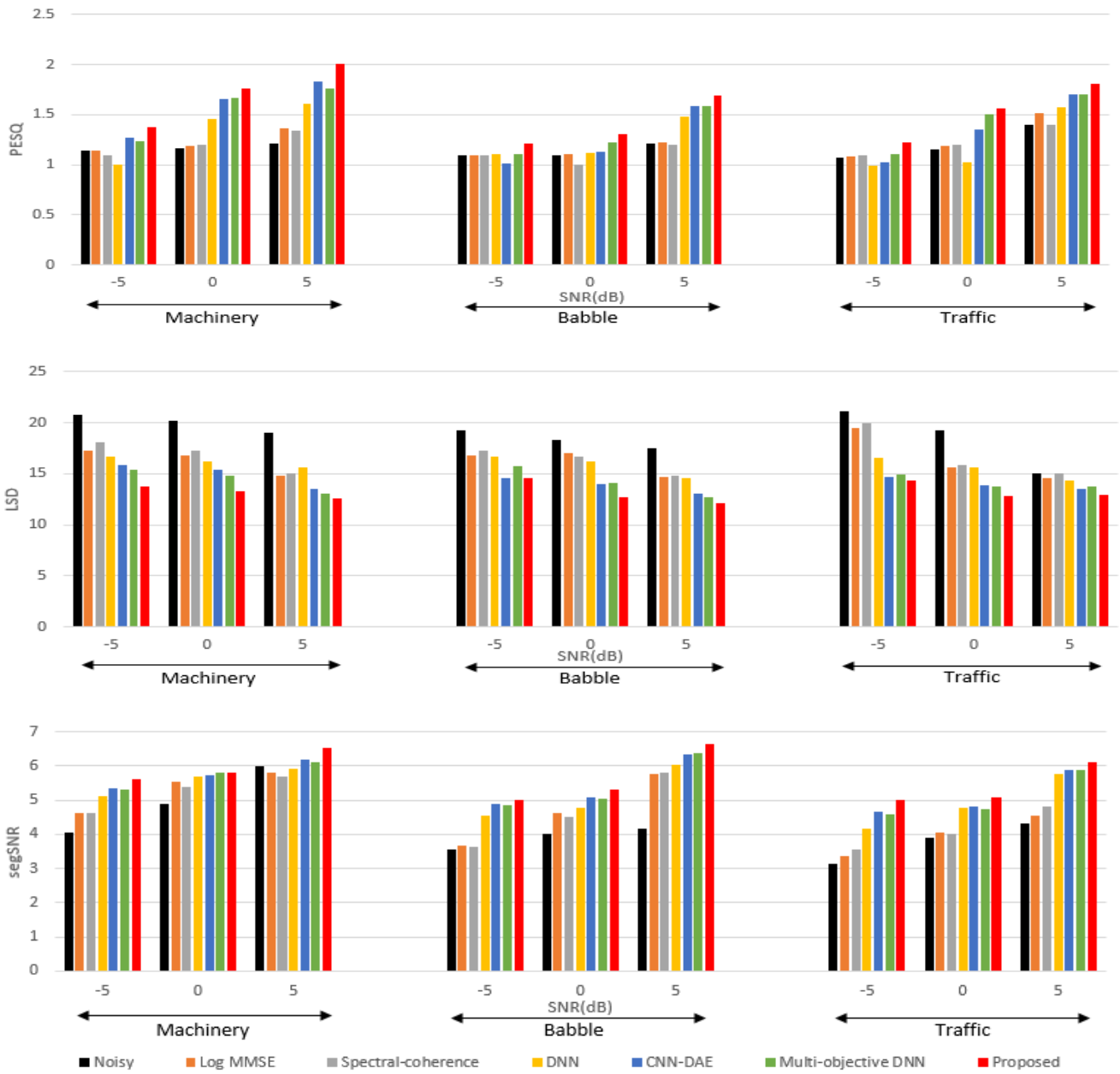

FIGURE 8. Comparison of average performance of PESQ, LSD, and SegSNR on smartphone recorded unseen data for different SE methods. Machinery, multi-talker babble and traffic noise at SNR levels of −5, 0 and +5dB are considered. The results show that the proposed CNN-based SE method outperforms the other SE methods.

15 different noise types were categorized into 3 major types of outdoor noise namely multi-talker babble, machinery and traffic noise. This division was made because in real world acoustic conditions, we often come across these noise types in outdoors. In addition, more than 30 smartphone recorded realistic noise for each noise type were collected and used for training. Speech files and noise files from different corpus were chosen to diversify the data which is important for real-time application. The duration of the resulting dataset was almost 33 hours. The noisy speech was considered at the sampling rate of 8 kHz for experimental results. All the other parameters are considered to be the same as explained in Section III. As objective evaluation criteria, we choose the perceptual evaluation of speech quality (PESQ) [44], Log Spectral Distance (LSD), segmental SNR (SegSNR). PESQ measure usually has a correlation with subjective tests compared to other objective measures. It ranges between 0.5 and 4.5, with 4.5 being high perceptual quality. LSD, which is the measure of logarithmic distance in spectra, provides information about signal similarity. Lower the LSD means higher the SNR and also do not contain considerable speech distortion. The amount of suppression of background noise and the musical noise reduction is generally measured using SegSNR. The proposed method which is the CNN-based SE

TABLE 3. Performance evaluation in unseen SNR condition for speech degraded by machinery noise. The neural network models were trained at −5, 0 and+5dB SNR and tested at −2.5,+2.5 and+7.5dB respectively.

Objective Measure	Method	SNR= -2.5 (dB)	SNR=2.5(dB)	SNR=7.5(dB)
PESQ	Noisy speech	1.1496	1.1819	1.2468
	CNN-DAE	1.19	1.6437	1.6883
	Proposed	**1.2679**	**1.7711**	**1.8441**
LSD	Noisy speech	18.5031	17.6985	16.3696
	CNN-DAE	14.9512	14.4864	14.3056
	Proposed	**14.0192**	**13.7728**	**13.6160**
SegSNR	Noisy speech	4.3988	5.4868	7.1029
	CNN-DAE	6.0734	6.8436	7.2104
	Proposed	**6.4612**	**6.9063**	**7.2141**

method is compared to noisy speech, conventional single channel SE based on Log-MMSE [9] and dual-microphone method like spectral coherence [45]. SE methods based on DNN [19], single channel CNN based denoising auto encoder (CNN-DAE) [26], and Multi-Objective Learning-based DNN SE [27] methods are implemented and included for comparison. The deep learning-based SE methods were trained on the same datasets as that of proposed method. The experimental evaluations are performed for 3 different noise types; machinery, multi-talker babble, and traffic noise. The reported results in Fig. 7 are the average over 20 sentences in 3 seen and 3 unseen noise conditions. Objective measures show significant improvements over conventional and deep learning method methods for all three noisy types considered. In neural network based methods, the data knowledge plays a vital role in the performance. Objective evaluation of smartphone collected unseen speech and noise data, shown in Fig 8, was carried out in order to analyze the performance variation of these methods on smartphone collected real data. SegSNR result at machinery noise condition at 0 dB SNR was on par with the CNN-DAE method. Other than that, the proposed method outperformed the conventional and neural network based methods. Objective measures shown in Fig. 7 and Fig. 8 reemphasize the fact that the proposed method achieves comparatively more noise suppression without distorting speech.

C. EFFECT OF DROPOUT AND UNSEEN SNR

1) DROPOUT EFFECT

In deep learning literature, the dropout is a well-known regularization technique for reducing overfitting in neural networks. The term 'dropout' refers to dropping out some of the units in a neural network during training phase.

In this experiment, we analyze the effect of dropout in the proposed model. Dropout usually correlates with how the loss function as well as the performance of the objective measures change. In most of the neural network methods, if the model over fits, the objective measures show significant variations when dropout is applied. The objective measures for the proposed method with dropout of 0%, 10%, 20% and 30% were evaluated as shown in Table 2. Clean speech degraded by machinery noise at 0dB SNR were considered to evaluate the effect of dropout for the proposed CNN-based SE model.

Even when dropout was applied, the performance variation of PESQ, LSD, and SegSNR were not significant, as shown in Table 2. Therefore, we can say that the current model does not over fit with the current training data. We see a slight reduction in terms of performance as more dropout is applied. This is due to the drastic decrease in the number of learnable parameters in the network. Therefore, dropout of 20% was used for real-time implementation in order to reduce computations.

2) EFFECT OF UNSEEN SNR

In this experiment, we evaluated the effect of unseen SNR on the proposed model. Sometimes, in the real- world noisy conditions, the changes in the SNR is quite common and rapid. Even the conventional SE techniques which are unsupervised can fail to maintain the performance during those changes. Therefore, we need a stable model which can overcome these changes in real time. To examine the effect of unseen SNR, three different models were trained at −5, 0 and +5dB SNRs and tested with the noisy speech at −2.5, +2.5 and +7.5dB respectively. Table 3. shows the performance evaluation of the proposed method versus unprocessed noisy speech and CNN-DAE method. Since, proposed method and CNN-DAE method use convolutional neural networks, this was used for comparison.

Clean speech degraded by machinery noise was considered to evaluate the performance of the proposed method in unseen SNR conditions. The neural network models of the proposed and the CNN-DAE methods were trained using noisy speech at −5dB SNR and were tested using the noisy speech data at −2.5dB SNR which is an unseen SNR. Similar experiments were conducted where the networks were trained at 0 and +5dB SNRs and tested at +2.5 and +7.5dB. From Table 3, we can see that the proposed method outperforms other methods in unseen SNR conditions. The trends were similar with other noise types like multi-talker babble and traffic noise. These results shown in Table 3 suggest that, the model can be used in unseen SNR conditions and also can be trained using different SNRs with a small trade off with the performance.

TABLE 4. Objective evaluation for unseen noisy condition on smartphone platform and on PC platform. The speech degraded machinery noise at 0 dB was used for real-time (smartphone) and offline (PC) evaluation.

Testing Platform	PESQ	LSD	SegSNR
PC(offline)	1.6876	12.7072	5.9085
Smartphone	1.6024	12.0219	5.8420

D. SMARTPHONE TESTING

Most of the methods perform extremely well in offline conditions with a controlled environment. However, their performance degrades when tested in real-time and in varying acoustic conditions especially on smartphone platforms. To the best of our knowledge, there are no published neural network based SE algorithms that are implemented on a smartphone as an application. Therefore, to evaluate the real-time operation of our proposed CNN-based SE application on smartphone, known speech sentences were played in machinery noise conditions at 0 dB SNR and were enhanced on a smartphone and also offline on a PC. The outcomes of the smartphone app running proposed CNN-based SE were stored to compare with the offline method. Both real-time and offline evaluations were tested on unseen data. Table 4 shows the performance evaluation of the proposed method in offline (PC) and in real-time (smartphone) conditions. The results shown in Table 4 show that, the real-time smartphone tested results are in par with the offline method. This experiment demonstrates that the model performs well when tested on smartphone platform.

E. SUBJECTIVE EVALUATIONS

Objective results provide useful information during the development phase of the proposed method. However, the practical usability of proposed method can be assessed by subjective tests. We performed mean opinion scores (MOS) [46] tests on 9 normal hearing adults including both male and female subjects. The subjects were presented with the noisy speech and enhanced speech using the Log-MMSE, Coherence, DNN SE, single channel CNN, multi-objective DNN SE and the proposed CNN-based SE methods at the SNR levels of 0 dB for 3 different noise types. Each subject was instructed to score in the range of 1 to 5 for the different audio files based on the following criteria; 5 being excellent speech quality and imperceptible level of distortion, 4 for good speech quality with perceptible level of distortion, 3 stood for fair speech quality with mediocre level of distortion, 2 for poor speech quality with lot of disturbances, causing uneven distortions, and 1 having the least quality of speech and intolerable level of distortion. Subjective test results are shown in Fig. 9 illustrating the effectiveness of the proposed method in various background noise, while simultaneously evaluating the speech quality and intelligibility perceptually. Detailed description of the scoring procedure is explained in [46]. The sample audio files enhanced using proposed method can be found in [47]. We can also observe that the PESQ plot for different noise types at 0dB SNR (shown in Fig.7) has similar response as that of the MOS shown in Fig. 9.

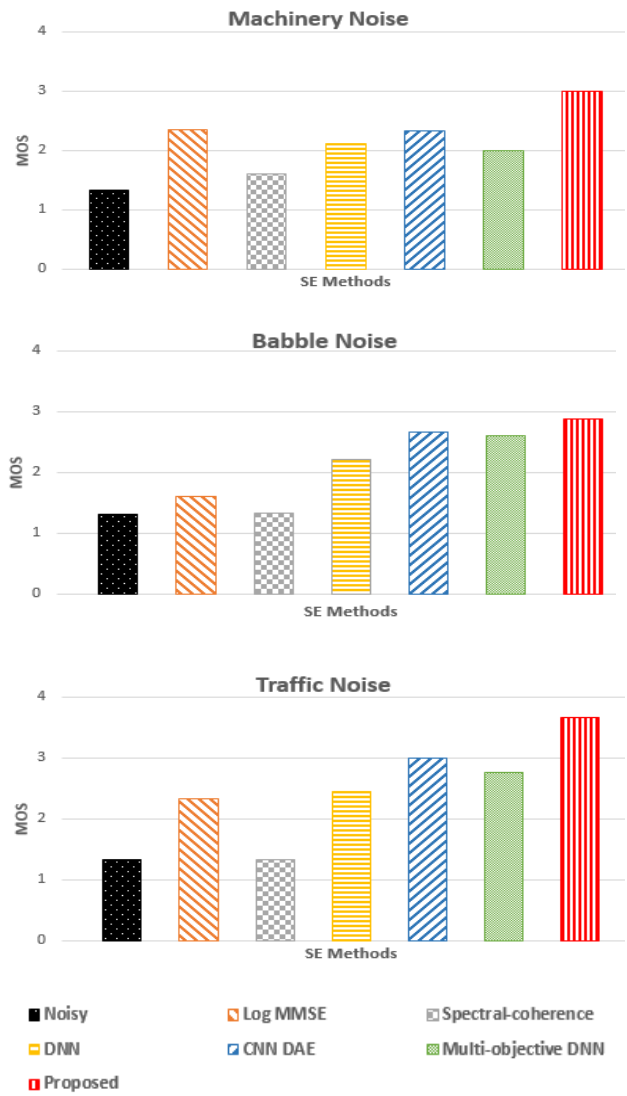

FIGURE 9. Subjective MOS test results evaluated for Babble, Machinery and Traffic noise types at 0dB SNR. The results show that the subjects preferred the proposed CNN-based SE method over other methods in terms of speech quality and intelligibility.

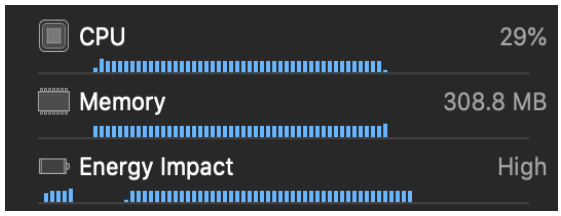

FIGURE 10. CPU and memory consumption of the iOS version of CNN-based SE application.

F. COMPUTATIONAL COMPLEXITY

This subsection provides the information regarding the computational complexity of the developed real-time

CNN-based SE app. To test the app, an iPhone X smartphone was used. The audio latency for these devices was measured to be 12 to 14 ms for iPhone X. Even some of the conventional methods [4], add a processing delay around 4-7ms considering a frame length of 32ms. Ideally, all processing should take place within the frame size of the audio input-output (i/o) frame to run smoothly at the lowest audio latency without skipping any frames. Since the processing time is less than frame length, i.e. 32ms, the proposed SE application is computationally efficient and runs seamlessly on a smartphone. Based on our experiments, the implemented application runs for approximately 5 hours on a completely charged iPhone X with a 2716 mAh battery.

The CPU, memory and battery usage of the app is also shown in Fig. 10 for the iOS smartphone used. The CPU consumption of the app is quite low. Since the app makes use of tensorflow C++ API, the CPU usage is around 29%. The maximum memory consumption of the app after switching on the SE is around 308.8 MB. The frozen model (explained in Section III) which includes all the weights that is used for smartphone implementation is about 57.3 MB, this means the actual memory footprint for the algorithm is around 251 MB. Since majority of the smartphones in the modern market come with processors with a memory of a minimum 12-16 GB, the memory consumption of the app is around 2.56% which is quite low. This shows that the app does not crowd the CPU and the memory resources of smartphones. It is also worth noting the point that even though both the energy and memory consumptions are quiet low, it is better to use simple networks rather than deeper/ wider networks. This is because the deeper networks usually have high inference time that can add higher latency in real-time.

V. CONCLUSION

In this paper, we proposed a multi-objective learning based convolutional neural network speech enhancement (SE). More accurate measurement of clean speech LPS estimate was obtained by adding the secondary feature sets MFSCs. The proposed CNN-based SE method is computationally efficient and is implemented on a smartphone as an application to perform real-time speech enhancement with low audio latency. The CNN-based SE architecture is optimized to allow input audio data frames to be processed on smartphone in real-time. Characteristics and performance of the proposed SE method and its implementation on smartphone are analyzed in detail. The objective and subjective test results validate the functionality of the proposed method as well as its practical application in real world under different noisy environments and low SNRs. The smartphone plus the proposed SE algorithm offer an affordable portable platform that can be used by people with hearing loss, as well as by audiologists and researchers for improving hearing study.

Acknowledgment

The authors would like to thank the National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) for their support. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

REFERENCES

[1] (2016). *Quick Statistics*. [Online]. Available: https://www.nidcd.nih.gov/health/771statistics/quick-statistics-hearing
[2] G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, "Formant frequency-based speech enhancement technique to improve intelligibility for hearing aid users with smartphone as an assistive device," in *Proc. IEEE Healthcare Innov. Point Care Technol. (HI-POCT)*, Bethesda, MD, USA, Nov. 2017, pp. 32–35.
[3] Apple. (2017). *Hearing Accessibility-iPhone-Apple*. [Online]. Available: https://www.apple.com/accessibility/iphone/hearing
[4] C. K. A. Reddy, N. Shankar, G. S. Bhat, R. Charan, and I. Panahi, "An individualized super-Gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device," in *IEEE Signal Process. Lett.*, vol. 24, no. 11, pp. 1601–1605, Nov. 2017.
[5] N. Shankar, A. Küçük, C. K. A. Reddy, G. S. Bhat, and I. M. S. Panahi, "Influence of MVDR beamformer on a speech enhancement based smartphone application for hearing aids," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Honolulu, HI, USA, Jul. 2018, pp. 417–420.
[6] B. Cornelis, M. Moonen, and J. Wouters, "Performance analysis of multichannel Wiener filter-based noise reduction in hearing aids under second order statistics estimation errors," in *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 5, pp. 1368–1381, Jul. 2011.
[7] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[8] 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.
[9] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 33, no. 2, pp. 443–445, Apr. 1985.
[10] P. J. Wolfe and S. J. Godsill, "Efficient alternatives to the Ephraim and Malah suppression rule for audio signal enhancement," *EURASIP J. Applied Signal Process.*, vol. 2003, no. 10, pp. 1043–1051, 2003.
[11] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 7, pp. 1110–1126, 2005.
[12] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Process.*, vol. 81, no. 11, pp. 2403–2418, 2001.
[13] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2008, pp. 4029–4032.
[14] E. A. P. Habets, "Single- and multi-microphone speech dereverberation using spectral enhancement," *Diss. Abstr. Int.*, vol. 68, no. 4, p. 257, 2007.
[15] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.
[16] E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners.," *J. Acoust. Soc. Amer.*, vol. 134, no. 4, pp. 3029–3038, Oct. 2013.
[17] E. W. Healy, S. E. Yoho, J. Chen, Y. Wang, and D. Wang, "An algorithm to increase speech intelligibility for hearing-impaired listeners in novel segments of the same noise type," *J. Acoust. Soc. Amer.*, vol. 138, no. 3, pp. 1660–1669, 2015.
[18] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.
[19] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.
[20] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1–5.
[21] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.
[22] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Proc. 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008, pp. 1–4.

[23] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. R. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, Apr. 2015.

[24] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," 2016, *arXiv:1609.07132*. [Online]. Available: https://arxiv.org/abs/1609.07132

[25] S.-W. Fu, T.-W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 9, pp. 1570–1584, Sep. 2018.

[26] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *Proc. Electron., Control, Meas., Signals Appl. Mechatronics (ECMSM)*, Donostia-San Sebastian, Spain, May 2017, pp. 1–5.

[27] Y. Xu, J. Du, Z. Huang, L. R. Dai, and C. H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *Proc. Interspeech*, 2015, pp. 1508–1512.

[28] Q. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A multiobjective learning and ensembling approach to high-performance speech enhancement with compact neural network architectures," in *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 7, pp. 1185–1197, Jul. 2018.

[29] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 4, pp. 679–681, Aug. 1982.

[30] J. K. Shah, A. N. Iyer, B. Y. Smolenski, and R. E. Yantorno, "Robust voiced/unvoiced classification using novel features and Gaussian mixture model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, pp. 17–21.

[31] Y. Obuchi, "Framewise speech-nonspeech classication by neural networks for voice activity detection with statistical noise suppression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar.*, vol. 2016, pp. 5715–5719.

[32] A. Sehgal and N. Kehtarnavaz, "A convolutional neural network smartphone app for real-time voice activity detection," *IEEE Access*, vol. 6, pp. 9017–9026, 2018. doi: 10.1109/ACCESS.2018.2800728.

[33] M. Huzaifah, "Comparison of time-frequency representations for environmental sound classification using convolutional neural networks," Jun. 2017, *arXiv:1706.07156*. [Online]. Available: https://arxiv.org/abs/1706.07156

[34] D. Liu, X. Wang, J. Zhang, and X. Huang, "Feature extraction using Mel frequency cepstral coefficients for hyperspectral image classification," *Appl. Opt.*, vol. 49, no. 14, pp. 2670–2675, 2010.

[35] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013, pp. 1–6.

[36] Google. (2018). *TensorFlow Lite*. [Online]. Available: https://www.tensorfow.org/

[37] Google. (2018). *TensorFlow*. [Online]. Available: https://www.tensorflow.org/lite/

[38] (2017). *Apple*. [Online]. Available: https://developer.apple.com/xcode/

[39] (2017). *Apple*. [Online]. Available: https://developer.apple.com/library/content/documentation/MusicAudio/Conceptual/CoreAudioOverview/WhatisCoreAudio/WhatisCoreAudio.html

[40] (2014). *Starkey*. [Online]. Available: http://www.starkey.com/blog/2014/04/7-halo-features-that-will-enhanse-every-listening-experience

[41] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–15.

[42] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 5206–5210.

[43] A. Mesaros, T. Heittola, and T. Virtanen. (Jan. 1, 2017). *TUT Acoustic Scenes 2017*. [Online]. Available: https://zenodo.org/record/400515

[44] 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.

[45] C. K. A. Reddy, Y. Hao, and I. Panahi, "Two microphones spectral-coherence based speech enhancement for hearing aids using smartphone as an assistive device," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, Oct. 2016, pp. 3670–3673.

[46] *Subjective Performance Assessment of Telephone-Band and Wideband Digital Codecs*, document ITU-Rec.P.830, 1996.

[47] (2019). *SSPRL*. [Online]. Available: https://utdallas.edu/ssprl/cnn-based-speech-enhancement/

GAUTAM S. BHAT received the B.E. degree in electronics and communication from Visvesvaraya Technological University, Belgaum, India, in 2016, and the M.S. degree in electrical engineering from the University of Texas at Dallas, Richardson, TX, USA, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include development of real-time speech enhancement and blind source separation algorithms using statistical signal processing methods, and machine learning-based approaches.

NIKHIL SHANKAR received the B.E. degree in electronics and communication from Visvesvaraya Technological University, Belgaum, India, in 2016, and the M.S. degree in electrical engineering from the University of Texas at Dallas, Richardson, TX, USA, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include beamforming, machine learning, and smartphone implementation of signal processing algorithms.

CHANDAN K. A. REDDY received the M.S. and Ph.D. degrees in electrical and electronics engineering from the University of Texas at Dallas. He is currently a Machine Learning Scientist with Microsoft Corporation. His research was focused on developing speech enhancement techniques for hearing aid applications. During his studies, he interned at Amazon Lab126 and Zounds Hearing, where he developed interesting algorithms in the area of speech processing and machine learning. His current research interest includes deep learning applied to various audio and computer vision problems.

ISSA M. S. PANAHI (S'84–M'88–SM'07) received the Ph.D. degree in electrical engineering from the University of Colorado at Boulder, in 1988. He is a Professor with the Department of Electrical and Computer Engineering (ECE) and also an Affiliate Professor with the Department of Bioengineering, University of Texas at Dallas (UTD), where he is the Founding Director of the Statistical Signal Processing Research Laboratory (SSPRL) and Audio/Acoustic/Speech Research Laboratory (UTAL) with the ECE Department. He joined the Faculty of UTD after working in research centers and industry for many years. Before joining UTD in 2001, he was the DSP Chief Architect, the Chief Technology Officer, an Advance Systems Development Manager, and the World Wide Application Manager with the Embedded DSP Systems Business Unit, Texas Instruments (TI) Inc. He holds U.S. patent and is author/co-author of four books and over 160 published conference, journal, and technical papers. His research interests include audio/acoustic/speech signal processing, noise and interference cancellation, signal detection and estimation, sensor array, source separation, and system identification. He founded and was the Vice Chair of the IEEE-Dallas Chapter of EMBS. He is the Chair of the IEEE Dallas Chapter of SPS. He was a member of organizing committee and the Chair of the Plenary Sessions at IEEE ICASSP-2010. He has been an Organizer and the Chair of many signal processing invited and regular sessions and an associate editor of several IEEE international conferences since 2006. He received the 2005 and 2011 Outstanding Service Award from the Dallas Section of IEEE and the ETRI Best Paper of 2013.