Audio Engineering Society

Convention Express Paper 306

Presented at the AES 157th Convention
2024 October 8-10, New York, NY, USA

This Express Paper was selected on the basis of a submitted synopsis that has been peer-reviewed by at least two qualified anonymous reviewers. The complete manuscript was not peer reviewed. This Express Paper has been reproduced from the author's advance manuscript without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. This paper is available in the AES E-Library (http://www.aes.org/e-lib) all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

On-Device Automatic Speech Remastering Solution in Real Time

Dongwoo Kim[1], Inwoo Hwang[2], and Sunmin Kim[3*]

[1] *Sound Laboratory, Visual Display Division, Samsung Electronics, Suwon, South Korea*

[2] *Sound Laboratory, Visual Display Division, Samsung Electronics, Suwon, South Korea*

[3] *Sound Laboratory, Visual Display Division, Samsung Electronics, Suwon, South Korea*

[*] sunmin21.kim@samsung.com

ABSTRACT

With the development of AI technology, there are many attempts to provide new experiences to users by applying AI technology to various multimedia devices. Most of these technologies are provided through server-based AI models due to the large model size. In particular, most of the audio AI technologies are applied through apps and it is server-based in offline AI models. However, there is no doubt that AI technology which can be implemented in real time is important and attractive for streaming service devices such as TVs. This paper introduces an on-device automatic speech remastering solution. The automatic speech remastering solution indicates extracting speech in real-time from the on-device and automatically adjusts the speech level considering the current background sound and volume level of the device. In addition, the automatic speech normalization technique that reduces the variance in speech level for each content is applied. The proposed solution provides users with a high understanding and immersion in the contents by automatically improving the delivery of speech and normalizing speech levels without manually controlling the volume level. There are three key points in this paper. The first is a deep learning speech extraction model that can be implemented in real-time on TV devices, the second is an optimized implementation method using the DSP and NPU, and last is audio signal processing for the speech remastering to improve speech intelligibility.

1 Introduction

Recently, various content has emerged very quickly with the appearance of OTT platforms such as Netflix, YouTube, and SNS. The emergence of such various content provides users with a variety of viewing opportunities, but at the same time, the inconvenience about the content is increasing due to various audio mixing methods for each content. In particular, the most inconvenient thing for users is that the level of the speech in the content is too low compared to the level of the background sound, so they momentarily miss the speech [1] [2]. The speech is the most important element in the context of the content, and if it is not delivered due to the low volume level of the speech, the understanding of the content decreases. To solve this problem, users can directly increase the absolute volume level to improve the delivery of the speech. It can be a useful solution in large spaces such as the theater. However, there are limits to increasing overall volume in a typical home environment.

Another inconvenience is that when watching YouTube or SNS content, the mixing level of the

speech is greatly different depending on the content creator. Since the length of content on YouTube or SNS is very short, users can watch various content in a limited time. Every time users watch content from a new creator like this, even if the volume level of the device is equal, the mixing level of the speech is different, and it makes users reduce immersion in the content.

To solve these inconveniences, this paper introduces an on-device automatic speech remastering solution. The automatic speech remastering solution extracts speech in real-time on the TV device and automatically adjusts the speech level in conjunction with the current background sound and overall volume level. In addition, the automatic speech normalization technique that reduces the variance in speech level for each content is applied to maintain content immersion.

The main contributions of this paper are as follows. 1. Deep learning speech extraction model that can be implemented in real-time on TV devices, 2. Optimized implementation method using the digital signal processor (DSP) and neural processing unit (NPU), and 3. Audio rendering technique for speech remastering.

The proposed extraction model benchmarks the U-Net [3] architecture, which showed excellent performance in audio source separation. Most U-Nets for audio separation models use a convolution kernel not only to the frequency axis but also to the temporal axis. However, the kernel structure is not easy to implement in real-time and on-device since it requires using future frames or many past frames to infer the current frame as input. To overcome this drawback, the proposed architecture is designed based on a 1D convolution layer on the frequency axis and an LSTM layer on the time axis to efficiently separate frequency axis and time axis calculations. As a result, the proposed speech extraction model is based on single-frame input and output, which is optimized for on-device and real-time performance.

The next section is about the optimized implementation method using the DSP and NPU. For efficient latency management, the various processes are allocated to the DSP and the NPU. As a result, a latency of 43.1ms can be obtained, which is a level that does not interfere with real-time when watching content.

The third section is audio signal processing for speech remastering. Speech extracted in real-time is adjusted depending on the background sounds and overall volume level. In addition, a speech normalization technique is proposed to reduce the variance of the speech in each content.

The next section is an evaluation of the proposed solution. This section shows the performance of the speech extraction model and the evaluation results for the overall solution implemented in the TV device.

2 Speech Extraction Model

2.1 U-Net Architecture

To implement automatic speech remastering solutions on various on-devices, it is necessary to impose an appropriate computational burden on the DSP and NPU. The proposed model is designed to minimize the burden of memory and calculation on DSP and NPU. The required input size of the model on NPU is only 1 frame (2048 samples length), so it can be easily implemented even on NPU and DSP with small memory.

The proposed speech extraction model is based on U-Net architecture which is the most commonly used in audio separation models. We use 1D convolution layers, the kernel of which does not extend the temporal axis but the frequency axis, and the Long Short Term Memory (LSTM) layer. The encoder and decoder are composed of seven 1D Convolutional Neural Network (1D CNN) and 1D Transposed CNN (1D TrCNN) blocks, respectively.

In the encoder, the input is compressed from 512 to 4 on the frequency axis using seven depthwise convolution layers having a two-stride size. As an activation function, a rectified linear unit (ReLU) is used. After passing the encoder, 128 channel outputs compressed to 4 sizes on the frequency axis are reshaped to 512 size and used as input of LSTM with 512 hidden dimensions.

The reshaped output of the encoder is passed into a unidirectional LSTM layer to aggregate the information along the temporal axis.

In the decoder, the output from the LSTM layer is reconstructed to the original input size using seven 1D transposed convolution layers having two-stride size. The decoder blocks use two inputs: the previous 1D transposed convolution layer output and the other is a tensor skipped from the encoder blocks at the same hierarchy. All decoder blocks except the last layer use ReLU. The specific configurations of each layer are as follows.

n-th: (k, s, c, p), where n, k, s, c, p represent the index of layer, kernel size, stride size, number of output channel and zero padding respectively.

Encoder = {1st: (6,2,64,2), 2nd: (6,2,128,2), 3rd: (6,2,256,2), 4th: (6,2,256,2), 5th: (6,2,256,2), 6th: (6,2,256,2), 7th: (6,2,128,2)}, and decoder = {1st: (6,2,256,2), 2nd: (6,2,256,2), 3rd: (6,2,256,2), 4th:

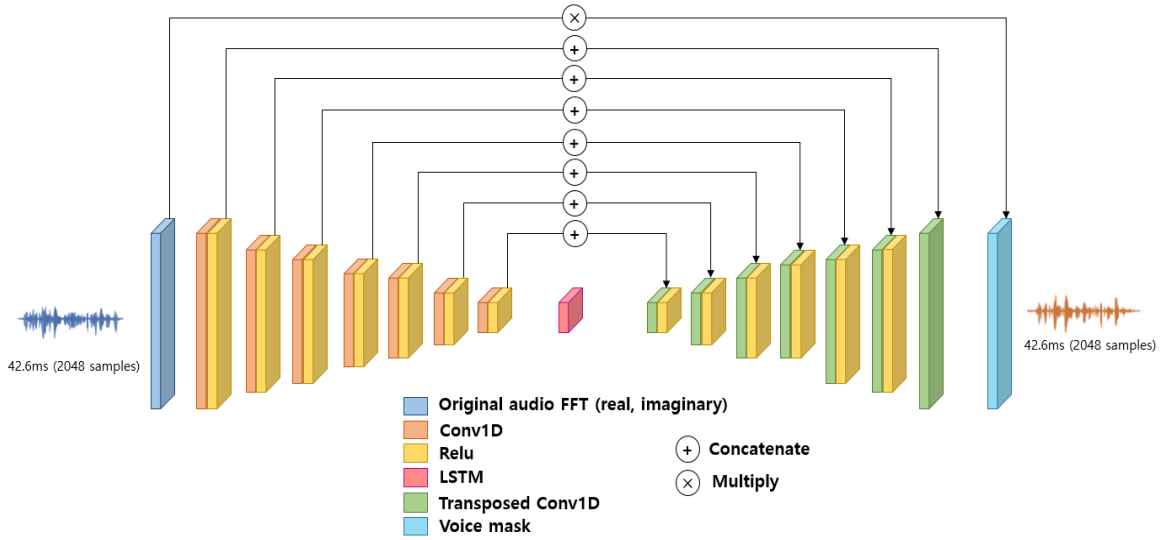


Figure 1. Architecture of speech extraction model.

(6,2,256,2), 5th: (6,2,128,2), 6th: (6,2,64,2), 7th: (6,2,2,2)}.

The final output of the decoder is multiplied by the original input spectrogram to obtain the spectrogram of the speech. The overview of the speech extraction model is shown in Figure 1.

2.2 Dataset

Unlike the vocal extraction model, which requires only vocals and musical instruments as training datasets, this model for a speech remastering solution that should be operated in various contents requires various types of training datasets. In other words, not only vocals & music but also general speech & effect sounds in movies and dramas have to be included in the dataset. We used a self-produced dataset for vocals, general speech, instruments, and background sound.

All training datasets are resampled to 48 kHz and cut to a length of 3 seconds (sec). Since the kernels of our 1D CNN and TrCNN do not extend the temporal axis, the length of the training samples affects the training LSTM layer.

The total number of data used for training is 49,254 samples each of speech and background sound, which are randomly mixed for each epoch to create new input audio. The batch size is set to 10, and the data augmentations such as the Gaussian noise adding, time stretch, and pitch shift are made to a 30% probability for robust training.

2.3 Preprocess of Data

The length of the training data is 3sec, but in the real-time implementation, the length of the input is only 42.6ms (2048 samples) to minimize the burden of DSP and NPU memory size. The length of 42.6ms indicates a 2048-point Fast Fourier Transform (FFT) size in the 48 kHz sample rate, which means that the inference input is 1 frame in terms of the frequency domain. Therefore, when preprocessing data, a normalization method considering the previous frame already passed is required. Per Channel Energy Normalization (PCEN) suitable for real-time implementation is applied [4], [5]. In this work, the PCEN is not a trainable version and the equation is as follows,

$$M_t = (1-s)M_{t-1} + sX_t, \quad (1)$$

$$PCEN_t = \sqrt{\frac{X_t}{M_t+\epsilon} + 2} - \sqrt{2}, \quad (2)$$

where X_t is the original FFT magnitude at t time and M_t is defined as the last state of t time. s and ϵ are constant values and set to 0.025 and 1e-6 respectively.

The input of this model is coefficients of complexity converted to a frequency domain, not magnitude. Therefore, we obtain the normalized input as follows,

$$input_t = (\frac{PCEN_t}{X_t}r, \frac{PCEN_t}{X_t}i), \quad (3)$$

where r and i represent coefficients of real and imaginary FFT values.

The normalized real and imaginary coefficients are filled in 1024 frequency bins (excluding the 1st bin) for each frame, and the resolution is about 23.4 Hz for each frequency bin.

This system needs to optimize the size of the model for on-devices. Therefore, we adjust the resolution to reduce the number of frequency bins for the input of the NPU model in a high-frequency band with little speech. From more than the 428th bin, which indicates more than 10 kHz, it is mapped with the following equation, and the number of input sizes is reduced from 1024 to 512.

$$y = \left\lfloor \frac{p^{(x+q)}}{r} \right\rfloor, \quad (4)$$

where $\lfloor \cdot \rfloor$ defines descending operation. p and q are constants that depend on where the frequency bins are mapped, and in this solution, since the mapping point starts from the 428th bin, these are set as 1.0104 and 463, respectively. r represents the resolution value of the frequency bin and is set to 23.4 (48kHz & 2048-point FFT). x is converted frequency bin and ranges from 428 to 512. y denotes the original frequency bin. Figure 2 shows the mapping process of the newly converted frequency bin with the original bin. Reducing the input size through this method can efficiently reduce the length of path memory between NPU and DSP, which is important in on-device implementation. In addition, since the frequency bands at which resolution deteriorates due to this mapping are the bands above 10 kHz where there are almost no speech sources, speech extraction performance does not deteriorate.

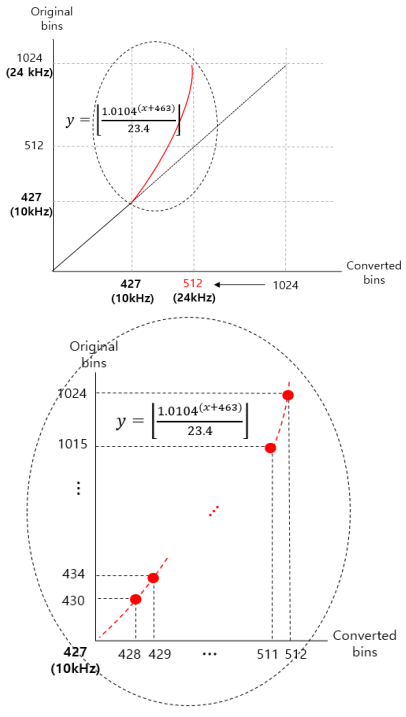


Figure 2. Mapping process of input. Above is the entire process and below is an enlarged figure.

3 Model Implementation

3.1 Implementation Architecture

The implementation of the proposed solution is optimized in DSP and NPU, which are essential hardware for real-time audio devices. In the DSP, the pre-processing of the input of the model and the post-processing of converting the speech mask extracted from the NPU into actual speech audio is performed. In the NPU, the speech extraction model is compiled and the speech mask is extracted in real time. The overall implementation architecture is shown in Figure 3. First, audio samples in the time domain are converted to the frequency domain through a 2048-point FFT. Then, the input is normalized using PCEN, and a mapping process is performed to reduce the number of frequency bins by half. The pre-processed input is passed to the NPU. The speech mask is generated in the NPU and passed back to the DSP. It is reconstructed to the original frequency bin size of 1024, and the speech spectrogram is obtained by multiplying the reconstructed mask with the original input spectrogram. Finally, the speech sources are obtained by inverse FFT.

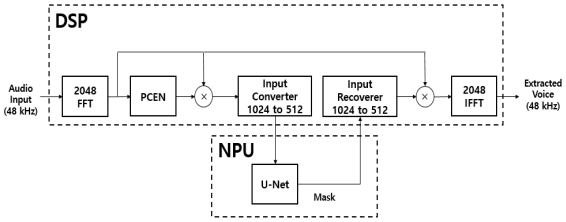


Figure 3. Overall implementation architecture.

3.2 On-device Implementation Latency

The audio input of the TV to which this solution was applied is streamed every 5.3ms (256 samples) with a 48 kHz sampling rate. The 256 samples are stacked in 2048-size memory in the DSP and pre-processed as described in section 3.1. However, the pre-processed values are passed on the NPU input after two frames, not one. In that, when 512 samples are pre-processed, they enter the NPU for speech extraction. Due to this DSP and NPU operating cycle, 256 samples (5.3ms) latency first occurs.

Next, latency is for model operations in the NPU. The proposed model is compiled in the TV NPU

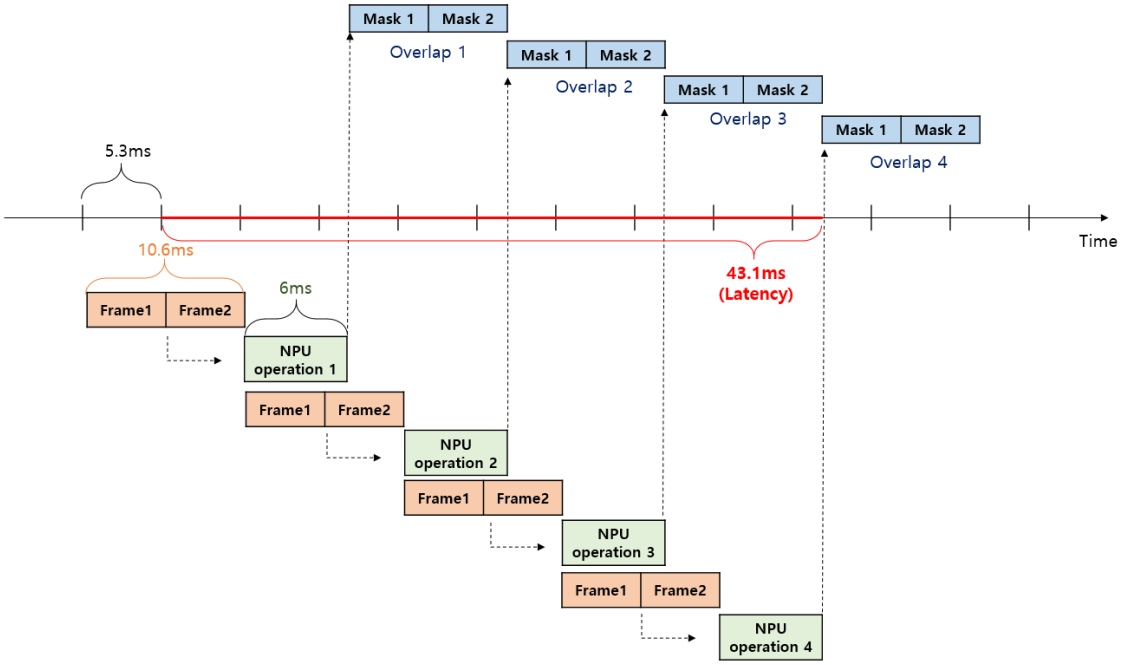


Figure 4. Implementation latency with 4 overlaps

(840MHz supporting Open-CL1.2 and Open-VX1.1) with INT16 type. The operating latency is about 6ms.

The third latency of this implementation is caused by the process of overlapping the obtained speech sources using a window filter. The overlaps are to improve the quality of the obtained speech source, however since the number of overlaps and latency is a trade-off, too many overlaps cannot be used. We found that the number of overlaps that maintain high-quality speech without compromising real-time is 4. The latency for 4 overlaps is 31.8ms.

With efficient scheduling of the DSP and NPU based on the above three latency values, the implementation latency can be optimized to 43.1ms, which is a small latency that the user cannot recognize the different synchronization between video and audio. The implementation latency is explained in Figure 4.

4 Speech Remastering

We found two main scenarios that degrade speech intelligibility through tests simulating the home environment. One scenario is that the speech is masked by a loud background sound. It has been studied for the relation between speech intelligibility and background noise [6]. The other scenario is that the speech is temporarily mixed smaller than the average power due to whispering scenes or poor recording environments.

To prevent the degradation caused by the two scenarios, the proposed speech remastering is set up with two audio processing blocks corresponding to each scenario. One is a speech amplifier based on the Speech-to-Background sound Ratio (SBR) and the other is a speech normalizer.

4.1 Speech and Background Sound Power Measurement

Before processing the speech remastering, it is required to measure the power of speech and background sound.

Input audio signal x can be divided into speech s and background sound b with the speech extraction model.

$$x(n) = s(n) + b(n), \quad (5)$$

where n is sample index.

The power of speech P_s and background sound P_b can be measured with mean square or ITU-R standard for audio loudness [7]. Each power is measured using a sample buffer every frame. And the buffer is updated with data from the current streaming point to the past N seconds. For the speech normalizer, average speech power P_{s_avg} is also required. It can be measured with a sample buffer of sufficient length or a hash table [8].

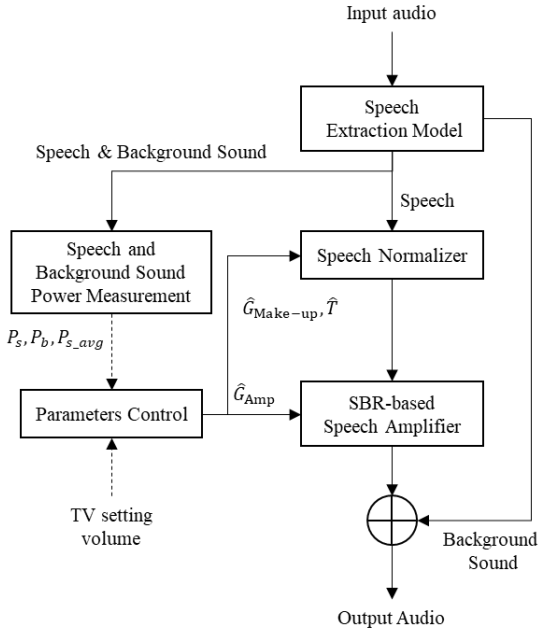


Figure 5. Flow chart of the speech remastering

4.2 SBR-based Speech Amplifier

SBR can be calculated with the power of speech and background sound as follows.

$$\text{SBR} = 10\log_{10}\left({}^{P_s}/_{P_b}\right), \quad (6)$$

The parameter for speech amplification gain is obtained with the calculated SBR.

$$G_{\text{Amp}} = \begin{cases} 10^{\left(\frac{\Delta dB_{\text{Amp}}}{20}\right)}, & \text{if } \Delta dB_{\text{Amp}} > 1, \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

where the ΔdB_{Amp} is defined as

$$\Delta dB_{\text{Amp}} = \text{SBR}_{\text{Target}} - \text{SBR}, \quad (8)$$

where $\text{SBR}_{\text{Target}}$ is the target level of SBR.

4.3 Speech Normalizer

Dynamic Range Compressor (DRC) is adopted for the speech normalization. The main parameters of the DRC are calculated depending on the estimated energy as follows.

Make-up Gain:

$$G_{\text{Make-up}} = \begin{cases} {}^{P_{s_avg}}/_{P_s}, & \text{if } P_{s_avg} > P_s, \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

Threshold (decibel):

$$T = 10\log_{10} P_{s_avg} + T_{\text{offset}}, \quad (10)$$

where T_{offset} is the offset decibel (dB) for the threshold.

The DRC increases the speech smaller than the average level to the make-up gain, while the speech larger than the average level is normalized to the threshold.

4.4 Parameters Control with Volume

Since the sound pressure level (SPL) of speech also affects speech intelligibility [9], the parameters of the speech remastering were adjusted according to the TV setting volume.

Control factors for the speech remastering parameters consisting of amplification gain, makeup gain, and threshold are defined as follows.

$$\boldsymbol{\alpha} = \{\alpha_{G_{\text{Amp}}}, \alpha_{G_{\text{Make-up}}}, \alpha_T\}, \quad (11)$$

The speech remastering parameters are adjusted by the control factors as follows.

$$\begin{aligned} \hat{G}_{\text{Amp}} &= \alpha_{G_{\text{Amp}}} G_{\text{Amp}} + \left(1 - \alpha_{G_{\text{Amp}}}\right), \\ \hat{G}_{\text{Make-up}} &= \alpha_{G_{\text{Make-up}}} G_{\text{Make-up}} \\ &\quad + \left(1 - \alpha_{G_{\text{Make-up}}}\right), \\ \hat{T} &= \alpha_T T + (1 - \alpha_T), \end{aligned} \quad (12)$$

As the volume increases, the degradation of speech intelligibility due to the two scenarios is reduced, so the control factor is gradually adjusted to 0 to output the original sound.

Figure 5 shows the flowchart of the speech remastering. First of all, the voice extraction model extracts voice and background sound from the input audio. Based on the measured powers, the speech normalizer is first processed, and then the speech is amplified by multiplication with $\hat{G}_{\text{Amp}}$ in the SBR-based speech amplifier processing block.

5 Evaluation

5.1 Experimental Environment

We evaluated the speech remastering solution in two parts. First, the performance of the speech extraction model was evaluated. Our model was compared with the most popular speech extraction models, ConvTasNet and Hybrid Transformer (HT) Demucs models, in terms of extraction performance and computations. The extraction performance was evaluated using signal-to-distortion ratio improvement (SDRi), which measures how much the SDR of the extracted speech signal has improved compared to the SDR of the original sound. The test samples were set to 5 movies, 5 dramas, and 5 daily content of about 2 minutes in length.

Secondly, the perceptual subjective metric of the mean opinion score is used to show that our system improves the viewing experience of users in real-time. The experiments were done with a TV set with its mainboard whose CPU is 1.66 GHz with 4 × Cortex-A76, the DSP is 810 MHz with 3 × Cadence HiFi4, and the NPU is 840 MHz supporting Open-CL 1.2 and Open-VX 1.1.

5.2 Evaluation of Speech Extraction Model

The evaluation of the speech extraction model was largely conducted from three perspectives. The first measures how much the SDR of the speech separated from the original sound has improved. This is about the speech extraction performance itself. The second is the number of parameters. Lastly, the most important thing in this paper is whether compilation is possible on a TV NPU and then, how long latency it takes on the NPU.

HT-Demucs had excellent extraction performance. However, HT-Demucs is a non-causal model that requires future frames as input, and considering it, the performance of our model was also excellent. The number of parameters was small enough for both our model and ConvTasNet to be implemented on TV, but the extraction performance of ConvTasNet was much lower than our model. As for the latency of the models in the actual TV NPU, our model had the lowest latency of 6ms. Non-causal models with excellent performance were not feasible for NPU. Table 1 shows these results.

Model	SDRi (dB)	Nb. Param. [M]	Latency on TV NPU
Ours (Causal)	15.31	**6.98**	**6ms**
ConvTasNet (Causal)	8.46	8.71	8.4ms
ConvTasNet (Non-causal)	15.51	8.71	Not feasibility
HT-Demucs (Non-causal)	**16.15**	41.2	Not feasibility

Table 1. Evaluation of speech extraction model.

5.3 Listening Test Results

Since the proposed speech remastering amplifies and normalizes extracted speech automatically while preserving other sounds, existing objective evaluation methods such as short-time objective intelligibility [10] used in noise reduction are not suitable. Therefore, as shown in Table 2, a Comparative Mean Score (CMOS) was employed to score subjective listening experiences for speech understanding compared to unprocessed audio.

For the evaluation, 10 participants scored the CMOS with 12 clips of various commercial contents including movies. The clips have an average SBR of -2.5 dB from -8.5 dB to 4.5 dB.

The experiment was conducted on a 65-inch TV at a recommended viewing distance of 2m in a room simulating the home environment. The TV volume was set to the average viewing volume of 15 from our own survey.

The overall average CMOS is 4.5, indicating that the listening experience for speech understanding has progressed significantly.

CMOS	Description
5	Much Better
4	Better
3	About the Same
2	Worse
1	Much Worse

Table 2.Comparative mean opinion scores and their meanings.

6 Conclusion

This paper proposed an on-device automatic speech remastering solution. The automatic speech remastering solution extracts speech in real-time on the TV device and automatically adjusts the speech level in conjunction with the current background sound and overall volume level. In addition, the automatic speech normalization technique that reduces the variance in speech level for each content was applied to maintain content immersion. The proposed solution was implemented on a TV device and showed excellent performance and latency fast enough for use in real-time content. This solution will be able to be applied to a variety of multimedia devices to provide improved speech quality.

References

[1] Jaeyoun Cho, Sunmin Kim, and Inwoo Hwang. "Active Voice Amplifier: On-Device Noisy Environment-Aware Solution for Dialogue Enhancement in Real Time." *Journal of the Audio Engineering Society* 71, no. 3 (2023): 129—137.

[2] Inwoo Hwang, Kibeom Kim, and Sunmin Kim. "On-Device Intelligence for Real-Time Audio Classification and Enhancement." *Journal of the Audio Engineering Society* 71, no. 10 (2023): 719—728.

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234—241. Springer International Publishing, 2015.

[4] Lostanlen, Vincent, Justin Salamon, Mark Cartwright, Brian McFee, Andrew Farnsworth, Steve Kelling, and Juan Pablo Bello. "Per-channel energy normalization: Why and how." *IEEE Signal Processing Letters* 26, no. 1 (2018): 39—43.

[5] Hyeong-Seok Choi, Sungjin Park, Jie Hwan Lee, Hoon Heo, Dongsuk Jeon, and Kyogu Lee. "Real-time denoising and dereverberation wtih tiny recurrent u-net." In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5789—5793. IEEE, 2021.

[6] ANSI, "American National Standard: Methods for the Calculation of the Speech Intelligibility Index" *ANSI S3.5* (1997).

[7] ITU-R, "Algorithms to measure audio program loudness and true-peak audio level" *ITU Radiocommunication BS.1770-3* (2012).

[8] K. Mehlhorn and P. Sanders, "Hash Tables and Associative Arrays" *Algorithms and Data Structures* pp. 81—98 (2008)

[9] P. Jianxin, "Chinese speech intelligibility at different speech sound pressure levels and signal-to-noise ratios in simulated classrooms." *Applied Acoustics* vol. 71, no. 4, pp. 386—390. (2010)

[10] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech" *2010 IEEE international conference on acoustics, speech and signal processing*, pp. 4214—4217 (2010).