E-URES: EFFICIENT USER-CENTRIC RESIDUAL-ECHO SUPPRESSION FRAMEWORK WITH A DATA-DRIVEN APPROACH TO REDUCING COMPUTATIONAL COSTS

Amir Ivry *Israel Cohen*

Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering,
Technion-Israel Institute of Technology, Haifa 3200003, Israel

2024 18th International Workshop on Acoustic Signal Enhancement (IWAENC) | 979-8-3503-6185-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IWAENC61483.2024.10694392

ABSTRACT

The user-centric residual-echo suppression (URES) framework accepts a user-operating point (UOP) comprising two metrics: the residual-echo suppression level (RESL) and the desired-speech maintained level (DSML). It produces several RES-system predictions with different UOP estimates, and the prediction with the highest acoustic-echo cancellation mean-opinion score (AECMOS) within the UOP tolerance becomes the output. Despite showing promising results, its high computational burden limits applicability. This paper introduces an efficient URES (E-URES) framework, which reduces computational costs in the final stage of the URES pipeline by minimizing the number of AECMOS computations. A lightweight neural network learns the relation between the UOP estimates and their corresponding AECMOS values by feeding the network various acoustic signals. During inference, the framework uses the three highest AECMOS predictions within the tolerance limit of the UOP to determine which outcomes to carry the actual AECMOS computations. Using 60 hours of data, average results show that the E-URES reduces 90% of the computational cost with negligible performance reduction.

Index Terms— Residual-echo suppression, user-centric, AECMOS, low compute, deep learning.

1. INTRODUCTION

The popularity of online meetings has brought about the widespread use of hands-free speech communication [1]. This type of communication involves two points of conversation - the far-end and the near-end. In a typical conferencing scenario, the far-end speaker, who is often located in a close-talk setup, sends his speech to the near-end speakers, who are typically sitting in a conference office. At the near-end, the far-end signal is often played by a nonlinear loudspeaker which is close to the near-end microphone [2]. During periods of double-talk, the near-end microphone picks up the desired speech, but also an amplified nonlinear reverberant version of the far-end signal and background noise, which impair the conversation intelligibility [3, 4].

Existing residual-echo suppression (RES) studies focus on benchmark performance, not user inputs. They lack a framework for balancing residual echo and speech distortion, supporting user preferences, and maximizing the AECMOS [5]. This degrades user experience and flexibility. To address that, we recently introduced the URES framework [6]. With a UOP input consisting of the RESL and the DSML values [7] desirable at the URES output, the URES undergoes three stages. Initially, it employs a pre-trained deep RES model with 101 instances. Each instance predicts signals with varying RESL and DSML values, guided by a tunable design parameter [8]. In the next step, each prediction is fed into another pre-trained deep model, which produces the RESL and DSML estimates of its input signal. Subsequently, the predicted RESL and DSML values are compared with the UOP. Predictions with estimates that fall within a pre-set UOP limit have their AECMOS calculated, and the signal with the highest AECMOS is sent to the far-end. The URES prediction offers three distinct benefits: it brings the estimated RESL and DSML of the prediction closer to the UOP, it makes adjustments based on UOP changes in practically real-time, and it maximizes the AECMOS value.

Although the URES has shown potential, its high computational load has hindered widespread use. To address this, we present the efficient-URES (E-URES), a more practical solution that reduces the number of AECMOS computations in the URES pipeline. It employs a streamlined neural network that learns to associate UOP deviations and acoustic signals with their corresponding AECMOS values. During the inference stage, the three highest AECMOS predictions within the UOP tolerance guide the framework to decide which of the 101 outcomes should undergo the actual AECMOS computations. To compare, the URES framework applies as many as 85 AECMOS computations instead of 3. Our experiments, conducted over 60 hours of data from the AEC-challenge [9] and individual recordings, reveal that the E-URES achieves an average reduction of 90% in computational cost with only a slight performance drop. These results consider only double-talk periods, with or without echo path

This research was supported by the Israel Science Foundation (grant no. 1449/23) and the Pazy Research Foundation.

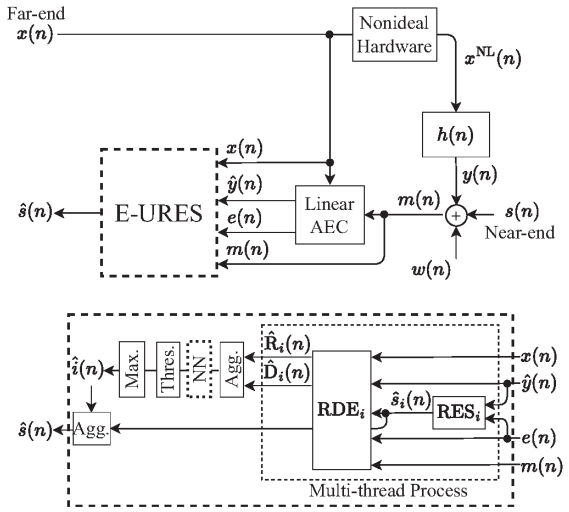

Fig. 1. Proposed E-URES framework. Top: an overview of the general RES acoustic setup. Bottom: The E-URES framework with the neural network (NN) in dashed line. The NN is what differentiates the E-URES from the URES framework.

changes. We consider various signal-to-noise-ratios (SNRs) and signal-to-echo-ratios (SERs) and show that the E-URES is able to maintain its original performance through challenging acoustic conditions.

2. PROBLEM FORMULATION

The E-URES system is illustrated in Fig. 1. We use italics to represent scalars and boldface to represent column vectors. At a given time index $n \in \mathbb{Z}$, the near-end microphone is:

$$m\left(n\right) = s\left(n\right) + w\left(n\right) + y\left(n\right), \tag{1}$$

where $s\left(n\right)$ denotes the desired near-end speech, $w\left(n\right)$ includes noise from the environment and system, and $y\left(n\right)$ is the reverberant far-end echo. The signal $y\left(n\right)$ can be expressed as:

$$y\left(n\right) = \mathbf{h}^{T}\left(n\right)\mathbf{x}_{\text{NL}}\left(n\right), \tag{2}$$

where $\mathbf{x}_{\text{NL}}\left(n\right) \in \mathbb{R}^{L}$ represents the L latest samples of the nonlinearly-distorted far-end signal, and $\mathbf{h}\left(n\right) \in \mathbb{R}^{L}$ is a filter with a finite impulse response, characterized by L coefficients, which describe the echo path extending from the loudspeaker to the microphone. We utilize an adaptive linear AEC that uses $m\left(n\right)$ as an input and the L latest samples of the far-end signal, $\mathbf{x}\left(n\right) \in \mathbb{R}^{L}$, as reference. This process results in the generation of the echo-path estimate $\widehat{\mathbf{h}}\left(n\right) \in \mathbb{R}^{L}$.

The acoustic signals $\mathbf{x}\left(n\right)$, $\widehat{y}\left(n\right)$, $e\left(n\right)$, and $m\left(n\right)$ are inserted into the E-URES, which generates the estimated desired-speech $\widehat{s}\left(n\right)$ before the far-end receives it. The U-RES' objective is to make sure that $\widehat{s}\left(n\right)$ complies with a UOP and attains the highest possible AECMOS value, while minimizing the number of AECMOS computations.

3. A DATA-DRIVEN E-URES FRAMEWORK

3.1. Overview of the URES Framework

The time index n is removed for brevity. Let the UOP be denoted by (R, D), where $\mathrm{R} \in \mathbb{R}$ is the RESL and $\mathrm{D} \in \mathbb{R}$ is the DSML [7]. Supported values are $15 \leq \mathrm{R} \leq 30$ and $7.5 \leq \mathrm{D} \leq 15$, in dB. From [8], the RES system is adopted. It receives the linear AEC's outcomes in the STFT domain [10]. The non-negative tradeoff parameter, $\alpha \in \mathbb{R}$, governs the balance between echo suppression and speech distortion during training, by regularizing the objective function $J(\alpha)$ at the RES system output:

$$J(\alpha) = \left\|\widehat{\mathbf{S}} - \mathbf{S}\right\|_{2}^{2} + \alpha \cdot \left\|\widehat{\mathbf{S}}\right\|_{2}^{2} + \sigma_{\widehat{\mathbf{S}}}^{2} \cdot \mathbb{I}_{\alpha>0}\,. \tag{3}$$

Here, $\widehat{\mathbf{S}}$ and $\mathbf{S}$ are the respective STFT magnitudes of $\widehat{\mathbf{s}}$ and $\mathbf{s}$, $\|\cdot\|_2$ is the ℓ_2 norm, $\sigma_{(\cdot)}^2$ is the variance, and $\mathbb{I}_{\alpha>0}$ is the indicator function. In [7], we showed that the RESL increases and DSML decreases on average as α rises, and conversely, so adjusting α can align the measured RESL and DSML with the UOP. We pre-train 101 identical RES system instances, using α values in the set $\{0, 0.01, \ldots, 1\}$. The index i in $A = \{0, 1, \ldots, 100\}$ lists each pre-trained instance of the RES system and its prediction, i.e., RES_i and $\widehat{\mathbf{s}}_i$, respectively. We take $\alpha_i = i/100$ to pre-train RES_i. Given that the RESL and DSML necessitate the target speech $\mathbf{s}$, it is not possible to transform the RES system's prediction to its corresponding RESL and DSML during the inference process. To address this, we have designed a deep learning model, termed as the RESL-DSML Estimator (RDE), estimating the relationship between accessible acoustic signals and the RESL and DSML of $\widehat{\mathbf{s}}_i$ for every i separately. We use 101 identical RDE instances, and each instance, RDE_i, is fed with the waveform signals $\mathbf{x}$, $\widehat{\mathbf{y}}$, $\mathbf{e}$, $\mathbf{m}$, and $\widehat{\mathbf{s}}_i$. The predicted RESL and DSML of $\widehat{\mathbf{s}}_i$ by RES_i are $\widehat{\mathrm{R}}_i \in \mathbb{R}$ and $\widehat{\mathrm{D}}_i \in \mathbb{R}$, respectively. During training, the ℓ_2 distance between the estimates $\widehat{\mathrm{R}}_i$ and $\widehat{\mathrm{D}}_i$, and the real RESL and DSML values is minimized. The UOP is compared with $\left(\widehat{\mathrm{R}}_i, \widehat{\mathrm{D}}_i\right)$ for each i. The allowed deviations from the UOP are defined by non-negative tolerance thresholds $\mathrm{TH}_{\mathrm{R}} \in \mathbb{R}$ and $\mathrm{TH}_{\mathrm{D}} \in \mathbb{R}$, in dB. The subset $A^{\mathrm{TH}} \subseteq A$ contains indices that meet (4):

$$\left|\widehat{\mathrm{R}}_i - \mathrm{R}\right| < \mathrm{TH}_{\mathrm{R}}, \quad \left|\widehat{\mathrm{D}}_i - \mathrm{D}\right| < \mathrm{TH}_{\mathrm{D}}. \tag{4}$$

All predictions $\widehat{\mathbf{s}}_i$ for $i \in A^{\mathrm{TH}}$ are batched, and the far-end receives the prediction with the highest AECMOS, $\widehat{\mathbf{s}}_{\widehat{i}}$.

3.2. The E-URES Framework

The purpose of the E-URES framework is to minimize the number of AECMOS calculations, so we replace the last stage of the URES pipeline with a data-driven solution using a nerual network. The main requirements of the neural network are to have low computational burden to improve the current limitation of the framework for general availability, and low latency to retain the near-real-time abilities of the framework. So, we utilize the standard fully-connected neural network and operate in the waveform domain. The inputs to the network are the acoustic signals $\mathbf{x}$, $\widehat{\mathbf{y}}$, $\mathbf{e}$, $\mathbf{m}$ and $\widehat{\mathbf{s}}_i$, and the pairs $\left(\widehat{\mathrm{R}}_i, \widehat{\mathrm{D}}_i\right)$ for all i. During training, the corresponding 101 AECMOS values for all i are used as ground truth and the ℓ_2 distance between them and the network's prediction is minimized. During inference, the network predicts the corresponding 101 AECMOS values. For all indices $i \in A^{\mathrm{TH}}$, the three maximal AECMOS predictions are considered. Let us denote the indices associated with these predictions as $i_p^1, i_p^2, i_p^3 \in A^{\mathrm{TH}}$. Then, the actual AECMOS calculations are carried for $\widehat{\mathbf{s}}_{i_p^1}, \widehat{\mathbf{s}}_{i_p^2}, \widehat{\mathbf{s}}_{i_p^3}$. Out of these, the prediction associated with the maximal AECMOS value, $\widehat{\mathbf{s}}_{\widehat{i}}$, is communicated to the far-end. Experimental results in [6] show it is extremely rare for A^{TH} to have less than 3 values. In that case, we consider all indices in A^{TH} and apply the same methodology.

4. EXPERIMENTAL SETTINGS

We use 10 hours of synthetic data and 40 hours of real recordings of data from the AEC-challenge database [9], considering only double-talk periods. The AEC-challenge corpus contains segments with and without echo-path changes. Scenarios without echo-path changes are those when the near-end speakers and loudspeaker are still. On the other hand, scenarios with echo-path changes involve regular movement of either the speakers or the devices at the near-end, which cause frequent re-convergence of the linear AEC system [3, 9].

We also utilize 10 hours of independent real lab recordings [8] with double-talk only. These recordings use the TIMIT [11] and Librispeech [12] corpora and include segments without echo-path changes. A mouth simulator and a loudspeaker placed in several positions in the near-end were used to generate the near-end speech and echo, respectively. This batch contained extremely high levels of echo to assert the operational envelope of the framework.

The SER levels were distributed in $[-20, 10]$ dB and SNR levels were distributed in $[0, 40]$ dB. The SER and SNR levels are defined as SER=$10\log_{10}\left[\|s\left(n\right)\|_2^2/\|y\left(n\right)\|_2^2\right]$ in dB and SNR=$10\log_{10}\left[\|s\left(n\right)\|_2^2/\|w\left(n\right)\|_2^2\right]$ in dB, respectively. Both corpora were recorded at 16 KHz.

The training dataset comprises 45 hours of data from the AEC challenge, which includes 35 hours of actual recordings and 10 hours of simulated data, supplemented by 5 hours of independent real recordings. The test dataset is authentic and includes 5 hours from both the AEC challenge and independent recordings. Both sets are balanced as per guidelines in [8], ensuring equal representation of genders, not placing speakers at both conversation ends, and more. The data was divided into 10-second segments, leading to frequent and realistic behavior where the linear AEC filter continuously reconverges [9]. The linear AEC filter used was a sign-error normalized least mean square (SNLMS) adaptive filter with a length of 150 ms [13, 14].

Echo-paths undergo sudden alterations every t seconds, where t follows a uniform distribution on $[4, 10]$, a trait typical of real-world conversations. Waveforms are handled in 20 ms time frames with a 50% overlap. The neural network was trained using back-propagation with a learning rate of 10^{-4}, which reduces by 10^{-6} every 5 epochs. The training used a mini-batch size of 40 ms and was conducted over 20 epochs, employing the Adam optimizer [15]. During training, the UOP estimates at the input and the AECMOS values at the output are normalized. During inference, normalization uses the training set statistics [16].

The neural network has an input layer, two hidden layers with 1024 neurons each, and an output layer, each is followed by a dropout [17] layer with ratio 0.5 and a ReLU [18] activation function. The input layer of the neural network has nearly 34000 neurons. In total, the network has 35.76 million parameters, 7.1 giga FLOPS, and requires 71.5 mega bytes of memory for allocation and instructions, considering we use 16-bit floating-point precision. Training the neural network took roughly 25 hours on an Intel Core i7-8700K CPU @ 3.7 GHz with two GPUs of type Nvidia GeForce RTX 2080 Ti.

We utilize the AECMOS, version 4, from Microsoft's API [5]. There are two AECMOS categories, and we report the first, which anticipates the subjective response of human listeners to the question "How would you rate the echo degradation?". The AECMOS is calculated with long time frames of 15 seconds, and we adhere to this as done in the URES case [6]. The AECMOS is dimensionless and operates on a scale from 1 to 5, with 5 being the highest score.

The next set of metrics we use report the computational burden and latency of the E-URES system components, including the neural network. For that, we utilize floating point operations per second (FLOPS), the number of bytes needed for the allocations and instructions, the number of parameters, system latency, and the real-time factor [19].

5. RESULTS

During inference, each utterance is inferred with a random UOP pair where the RESL value is uniformly drawn from $[15, 30]$ dB and the DSML value is uniformly drawn from $[7.5, 15]$ dB. The AECMOS results are reported using mean and standard deviation values across the entire test set.

In Table 1, we focus on the computational complexity and the timing constraints of the E-URES and compare it with

Table 1. Computational load and timing requirements of the E-URES and URES.

Measure	Parameters	FLOPS	Latency	RTF
E-URES	22×10^6	1.4×10^{12}	20.2 ms	1
URES	43.7×10^6	12.9×10^{12}	38.4 ms	1.9

Table 2. AECMOS results of the E-URES and URES in scenarios with and without echo-path change.

Scenario	Echo-path change	No echo-path change
E-URES	3.15 ± 0.45	3.5 ± 0.4
URES	3.3 ± 0.45	3.6 ± 0.35

Table 3. AECMOS results of the E-URES and URES in various SER levels with no echo-path change.

SER	-20 [dB]	-10 [dB]	0 [dB]	10 [dB]
E-URES	3.0	3.4	3.75	3.95
URES	3.1	3.5	3.8	4

Table 4. AECMOS results of the E-URES and URES in various SNR levels with no echo-path change.

SNR	0 [dB]	10 [dB]	25 [dB]	40 [dB]
E-URES	2.85	3.3	3.65	3.9
URES	3	3.4	3.7	3.95

the ones of the URES framework. The E-URES pipeline maintains a fixed number of model instances. Conversely, in the URES, the quantity of AECMOS model instances is contingent on the size of the set A^{TH}, which in turn is data-dependent. As demonstrated in [6], the maximum number of AECMOS model instances is 85, a figure we consider when presenting the URES framework's values.

It can be seen that the E-URES framework is more efficient than the URES across all measures. The parameter count is halved, FLOPS are reduced by 90%, and both buffering and inference times, i.e., latency, are cut down by 48%. Furthermore, the real-time factor (RTF) is also reduced by 48%, achieving a real-time value of 1 [19]. The latency and RTF are calculated using the specifications of the NVIDIA Jetson Xavier NX system-on-module (SoM) [20], a module specifically designed to facilitate speech processing via neural networks. Another improvement occurs in memory, where the E-URES requires 1.3 giga bytes for memory and allocation, and the URES requires 2 giga bytes, a drop of 35%. The E-URES not only significantly reduces computational and timing resources but also enhances efficiency to a level that enables the framework to run on a common processor used by personal end-users, such as the 11th Gen Intel CoreTM i7-11850H @ 2.50 GHz processor. The E-URES is easily accessible, unlike URES, which requires dedicated hardware.

In Table 2, we investigate whether the enhanced efficiency of the E-URES framework has compromised its performance compared to the URES. Initially, we separately evaluate the average AECMOS value in scenarios with and without echo-path change. We notice that the AECMOS has decreased by 0.1 points in echo-path change scenarios and by 0.15 points in no echo-path change scenarios compared to the URES. The standard deviation values remain approximately constant. In Tables 3 and 4, we persist in our examination of the average AECMOS of the E-URES, this time concentrating on different levels of SERs and SNRs in scenarios with no echo-path change. Once again, we observe only a slight decrease in the average AECMOS across all echo and noise levels, with the most significant gap being 0.15 points when SNR $= 0$ dB. For all echo levels, the largest reported gap is 0.1 points. To the subjective human listener at the far-end, the average differences between the E-URES and URES are too small to be noticeable [6] and do not indicate a meaningful degradation in the E-URES performance.

The neural network we developed and integrated into the E-URES at the expense of a large number of AECMOS computations has proven to be dramatically more efficient while preserving roughly the same performance, on average, as the URES framework. This holds true for scenarios both with and without echo-path change, which indicates the generalization ability of the neural network in scenarios of echo-path re-convergence, and for low and high noise and echo levels, which implies the high robustness of the neural network.

6. CONCLUSIONS

The E-URES framework, an extension of the URES framework, has been introduced to address the high computational demand of the URES. The E-URES employs a lightweight neural network that learns the association between UOP deviations, acoustic signals, and their corresponding AECMOS values. This learning enables the E-URES to reduce the number of AECMOS computations in the final stage of the URES pipeline. Our experiments reveal that the E-URES achieves a reduction of 90% in computational cost with only a slight performance drop. Therefore, the E-URES framework presents a more practical and efficient solution for residual-echo suppression, enhancing user experience in realistic acoustic setups. Efforts will continue to streamline other aspects of the URES framework, paving the way for its availability in offices, homes, and mobile phones.

7. REFERENCES

[1] M. Schmidtner, C. Doering, and H. Timinger, "Agile working during COVID-19 pandemic," *IEEE Engineering Management Review*, vol. 49, no. 2, pp. 18–32, 2021.

[2] K. Sridhar, R. Cutler, A. Saabas, T. Parnamaa, M. Loide, H. Gamper, et al., "ICASSP 2021 acoustic echo cancellation challenge: Datasets, testing framework, and results," in *Proc. ICASSP*. IEEE, 2021, pp. 151–155.

[3] Eberhard Hnsler and Gerhard Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*, Wiley-IEEE Press, 2004.

[4] M. M. Sondhi, D. R. Morgan, and J. L. Hall, "Stereophonic acoustic echo cancellation-an overview of the fundamental problem," *IEEE Signal Processing Letters*, vol. 2, no. 8, pp. 148–151, 1995.

[5] M. Purin, S. Sootla, M. Sponza, A. Saabas, and R. Cutler, "AECMOS: A speech quality assessment metric for echo impairment," in *Proc. ICASSP*. IEEE, 2022, pp. 901–905.

[6] A. Ivry, I. Cohen, and B. Berdugo, "A user-centric approach for deep residual-echo suppression in double-talk," *IEEE/ACM Transations on Audio, Speech, and Language Processing*, 2024.

[7] A. Ivry, I. Cohen, and B. Berdugo, "Objective metrics to evaluate residual-echo suppression during double-talk," in *Proc. WASPAA*. IEEE, 2021, pp. 101–105.

[8] A. Ivry, I. Cohen, and B. Berdugo, "Deep residual echo suppression with a tunable tradeoff between signal distortion and echo suppression," in *Proc. ICASSP*. IEEE, 2021, pp. 126–130.

[9] R. Cutler, A. Saabas, T. Pärnamaa, M. Purin, E. Indenbom, N. C. Ristea, et al., "ICASSP 2023 acoustic echo cancellation challenge," *IEEE Open Journal of Signal Processing*, pp. 1–10, 2024.

[10] H. Zhivomirov, "On the development of STFT-analysis and ISTFT-synthesis routines and their practical implementation," *Technology, Education, Management, Informatics (TEM) Journal*, vol. 8, no. 1, pp. 56–64, 2019.

[11] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report*, vol. 93, pp. 27403, 1993.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[13] A. Ivry, I. Cohen, and B. Berdugo, "Deep adaptation control for acoustic echo cancellation," in *Proc. ICASSP*. IEEE, 2022, pp. 741–745.

[14] N. L. Freire and S. C. Douglas, "Adaptive cancellation of geomagnetic background noise using a sign-error normalized LMS algorithm," in *Proc. ICASSP*. IEEE, 1993, vol. 3, pp. 523–526.

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

[16] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *preprint arXiv:1803.08375*, 2018.

[19] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457, 2021.

[20] NVIDIA, "Jetson Xavier NX: The world's smallest AI supercomputer," `https://developer.nvidia.com/blog`, 2023.