National Journal of Speech and Audio Processing, ISSN: 3107-5258 Vol. 1, No. 3, 2025 (pp. 1-8)

RESEARCH ARTICLE

ECEJOURNALS.IN

Lightweight Deep Neural Networks for Real-Time Speech Enhancement on Edge Devices

Rasanjani Chandrakumar[1], Freddy Soria[2]

[1]*Department of Electrical Engineering Faculty of Engineering, University of Moratuwa, Sri Lanka*
[2]*Robotics and Automation Laboratory Universidad Privada Boliviana Cochabamba, Bolivia*

KEYWORDS:
Speech enhancement,
Edge computing,
Lightweight DNN,
Real-time processing,
Neural quantization,
Embedded AI

ARTICLE HISTORY:
Submitted : 05.02.2025
Revised : 10.03.2025
Accepted : 15.05.2025

https://doi.org/10.17051/NJSAP/01.03.01

Abstract

In this paper we are presenting a new architecture of lightweight deep neural networks (DNN) which is targeted to real-time speech enhancement with resource-constrained devices at the edge of the network. With speech-based systems like smart assistants, attachable hearing aids, and voice-integrated interfaces becoming more and more popular, there has been stronger need of high quality noise reduction that would have minimal compute requirements. Analysis using traditional signal processing techniques fails to do well in non-stationary noise conditions, and although deep learning-based techniques are better in this condition, their computational requirements are frequently incompatible with running on low-power edge devices. To solve this, we will present a hybrid architecture that presents an efficient feature extraction framework using depthwise separable convolutions, a short attention-augmented bidirectional GRU model module to gain temporal modelling and post-trained quantization process to allow memory and inference-level compression. The training data afforded by popular benchmark datasets such as VoiceBank-DEMAND and the DNS Challenge corpus contain a wide variety of noise present in test signals as well as noise types. We benchmark the proposed model to several key objective measures of a model including Perceptual Evaluation of Speech Quality (PESQ), Short-Time Objective Intelligibility (STOI), and the Signal-To-Distortion Ratio improvement (SDRi), in addition to practical deployment metrics that include model size, inference latency, and power consumption. Our low power DNN has a PESQ of 3.01 and SDRi of 9.4 dB and a run time factor (RTF < 1.0) on both ARM Cortex-M7 microcontrollers using CMSIS-NN and NVIDIA Jetson Nano for both applications using INT8 quantization via TensorRT. The overall size of the model itself is less than 2 MB and the amount of power required falls within the range of battery-powered devices. In objective tests, intelligibility and clarity improvements were also realized under real world hazardous noise situations. The results presented herein indicate that the suggested method is viable in terms of addressing the mismatch between high-performing speech enhancement and embedded hardwares with a scalable and deployable solution to the vast majority of edge AI audio implementations in noisy acoustic situations.

Author's e-mail: rasanjani.chandr.@elect.mrt.ac.lk

How to cite this article: Chandrakumar R, Soria F. Lightweight Deep Neural Networks for Real-Time Speech Enhancement on Edge Devices. National Journal of Speech and Audio Processing, Vol. 1, No. 3, 2025 (pp. 1-8).

Introduction

Over the past years, real-time speech enhancement has emerged as a technology pillar of many audio-oriented technologies, speech enhancement in hearing aids, teleconferencing systems, voice-controlled IoT, smart assistants, and mobile communication applications. Speech enhancement has the aim of reducing the effect of background noise and enhancing intelligibility and quality of speech signals in poor acoustic conditions. Although it has obtained substantial advances in the development of enhanced techniques of great complexity, most of the high-performance solutions are computationally costly and are oriented towards implementation at power-intensive servers and desktop-type processors. This is a key challenge to the implementation of such models on end-devices which usually have severe limitations with regards to memory, processing power and energy demands.

Typically, conventional signal processing methods (i.e. spectral subtraction, Wiener filtering and statistical

model-based estimators) have difficulties due to stationary noise assumptions and cannot be applied well to non-stationary, practical situations (i.e. dynamic noise). Also, the methods do not have the capacity to represent the nonlinguistics associations and time-lapse effects in speech signals. Comparatively, the use of deep learning has driven an increased interest in the modeling and suppression of varying noise patterns; especially those based on convolutional neural (and recurrent) networks. Net Architectures like SEGAN, DCCRN, and CRN have shown promising enhancement prospects, but their deep and wide-layered straightened structures become hard to meet in terms of latency, memory footprint, and power usage, which are paramount concerns in consideration of edge deployment Figure 1.

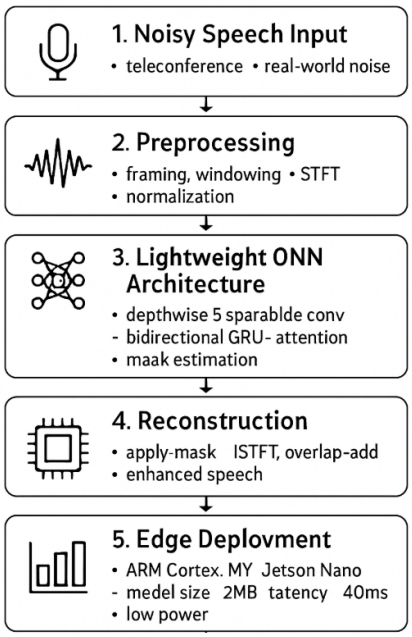

Fig. 1: Real-Time Speech Enhancement Pipeline

This has caused the growing popularity of lightweight neural network architecture as the demand is on having an effective trade-off between the model complexity and performance in enhancement. Nevertheless, current compact solutions affect the quality of speech, and even in severely non-stationary or low SNR conditions. This study seeks to fill this gap by outlining a new proactive Lightweight DNN-Based Speech Enhancement Framework that is specifically designed thanks to edge computing systems. The suggested answer unites depthwise separable convolutions to extract but not many features with an efficient model, attention-augmented sequential modeling with a gated recurrent unit and compression strategies such as pruning and after training quantization. All these strategies minimize computational burden and the size of the models without compromising in the fidelity of improvements.

To confirm the suggested architecture, we run a comprehensive set of experiments on conventional datasets VoiceBank-DEMAND and DNS Challenge, and test the model on the edge environments, including ARM Cortex-M7 and NVIDIA Jetson Nano. Both quality and efficiency are measured in terms of Perceptual Evaluation of Speech Quality (PESQ), Signal-to-Distortion Ratio improvement (SDRi), Short-Time Objective Intelligibility (STOI) and real-time factors (RTF). Experimental findings conclude that our model can generate high-quality speech enhancement results over baseline techniques with much less computational load and power consumption, which renders our model highly promote to real-time edge prediction.

Fundamentally, the proposed study has provided a beneficial and scalable deep learning system that can be executed to establish consistent low-latency improvement in speech with edge devices. It overcomes some of the most critical issues of embedded-AI by fine-tuning both the model network and inference approach and, thereby, makes the mainstream use of deep speech enhancement in limited-power, limited-resource applications a reality.

Related Work

Speech enhancement has come a long way in the last several decades, and today it is no longer viewed as a problem of classical signal processing but rather deep learning scenario, and even more so, dense architecture inference on an embedded device. The literature in this section has been reviewed in three broad categories that include traditional methods, deep neural approaches, and light architectures of DNN developed for resource-conservative platforms.

Old Methods of Speech Enhancement

Traditional speech enhancement strategies mostly use signal-processing algorithms like spectral subtraction,[1] Wiener filtering[2] and MMSE-based estimators.[3] The techniques are computationally efficient and applicable to low-complexity settings but normally involve making stringent assumptions concerning noise stationarity and signal frequency pattern. Consequently, many of them easily degrade in performance when non-stationary or real-world noise conditions are presented and they introduce artifacts like musical noise.

Methods based on Deep Learning

Deep learning has been demonstrated as a potentially very good candidate to model non-linear and complex noise environments. A speech enhancement generative

adversarial network SEGAN,[4] showed that adversarial training could lead to more natural sounding speech. Thereafter, complex-valued convolutions and recurrent layers have been introduced to phase-aware enhancements, such as DCCRN.[5] The Zhao et al. proposed deep feature loss networks[6] that use perceptual loss functions to achieve better understand ability. These solutions are, however, very demanding in computing requirements, including GPU-accelerated computation, which is unfeasible in real time embedded implementations.

Recent research has covered applications in deep learning in a wide range of embedded applications, including secure FPGA-IoT implantation,[7] deep learning-based MIMO networks[8] and AI-enhanced structural health monitoring.[9] Though they do not center on the use of speech enhancement, the studies indicate the viability of AI models in limited hardware conditions. The same has been seen in signal processing systems across the edges, and power electronics.[10, 11]

Lightweight DNN System Designs on the Edge

Some lightweight neural architectures have appeared in order to deliver the performance requirements of edge computing. MobileNet,[12] developed in the context of computer vision tasks, and SqueezeNet,[13] developed specifically in the context of the computer vision tasks, introduced the parameters, efficient depthwise convolution, and fire module, respectively. Based on them, TinyWaveNet[14] and Conv-TasNet-Lite[15] are proposed in the audio domain to achieve the trade-off between accuracy and real-time processing.

Even in spite of these efforts, the quality of speech can still be of poor quality due to the aggressive compressing. In addition, most edge deployments are yet to be optimized in terms of quantization and pruning, and inference scheduling. Current trends in memory technologies Memory-efficient, low-latency approaches to AI have been identified by recent research on emerging-memory technologies[11] and neural accelerators to enable fast embedding in real-time,[10] as a high priority of edge-AI researchers Table 1.

Proposed Methodology

Network Architecture

Input and Preprocessing

This suggested model of speech enhancement is applicable on noisy time-domain waveforms that provide a fully end-to-end learning algorithm without the need to tap into a hand-designed feature engineering. Raw audio is then sliced into overlapping frames of a Hamming window, and frame size is at 20 ms frame stride at 10 ms, often a trade-off between latency and a temporal resolution. Through the short-time Fourier transform (STFT), each frame is converted to a time-frequency representation and results in a complex spectrogram. In order to feed the input data to the neural processing, the magnitude spectrogram, as opposed to the phase, is introduced into the model; the remaining part of the spectrogram is stored and used to reconstruct the data at the final enhancement step. According to this practice, the model is made less complex but preserving the perceptual fidelity. The preprocessing pipeline is targeted to be lightweight and can be run on on-device digital signal processors (DSPs) or efficiently and quickly real-time executed in optimized libraries like CMSIS-DSP in order to be compatible with an edge deployment.

Table 1: Comparative Summary of Speech Enhancement Techniques

Category	Representative Models	Key Techniques	Enhancement Quality	Complexity	Edge Deployment Feasibility
Traditional Methods	Spectral Subtraction,[1] Wiener Filtering,[2] MMSE[3]	Statistical signal modeling, noise estimation	Low to Moderate	Low	High (but limited performance)
Deep Learning-Based	SEGAN,[4] DCCRN,[5] Deep Feature Loss[6]	GANs, complex RNNs, perceptual loss functions	High	High	Low (requires GPU/CPU)
Embedded DL Applications	FPGA-IoT,[7] Massive MIMO Estimation,[8] Smart SHM[9]	Hardware-accelerated deep learning	Domain-specific	Medium to High	Medium (task dependent)
Lightweight Architectures	MobileNet,[12] SqueezeNet,[13] TinyWaveNet,[14] Conv-TasNet-Lite[15]	Depthwise convs, fire modules, model compression	Moderate to High	Low to Medium	High (optimized for edge)

Features Extraction and Modeling of Time

The backbone of the suggested design starts with a feature extraction unit that was constructed by 1D depthwise dividable convolutional layers. Such layers can make the parameter space and the computational operations an order of magnitude smaller due to the decoupling of both spatial (time-domain) and channel-wise learning of features. Depthwise separable convolutions enable the network to learn frequency-dependent features, which enables it to use a much smaller fraction of the computational resources occupied by ordinary convolutions, and therefore a good choice of edge AI. The representations are then submitted to a bidirectional gated recurrent unit (Bi-GRU) layer that accounts the temporal dependencies in forward and backward directions in the speech signal. It is essential in modeling a long-term contextual information particularly under non-stationary noise environment. Also, on top of the Bi-GRU output an attention mechanism is used to selectively focus to the most informative frames. Not only does that make improvement performance better, but also make generalization in cases of other types of noise easier because the model can learn the temporal time intervals which are more important to clarity of speech.

Output and Reconstruction Layer

The last layer of the architecture consists of producing a mask, which is used in suppressing individual components of noise but retains clean speech in the input magnitude spectrogram. The output of this mask is predicted by a fully connected output layer traced by attention-weighted temporal information to a time-frequency mask capped at the range of [0, 1] via a sigmoidal activation. The enhanced magnitude spectrogram corresponds to multiply by element-wise the predicted mask with the original noisy magnitude spectrogram. Afterward, the improved magnitude of this clean signal is used to recreate the clean signal in the time domain using the inverse STFT (iSTFT) whilst using the noisy phase. The above reconstruction process will maintain a light and efficient model without compromising audible speech that is natural-sounding. The overall network is trained in an end-to-end manner with a mixture of spectral-domain loss (e.g. mean squared error) and perceptual loss (e.g. scale-invariant SNR or PESQ-based loss) allowing the model to maximize objective quality metrics as well as human-perceived audio quality Figure 2. To support deployment in real-time scenarios, the architecture is also carefully engineered to ensure size of the entire model is smaller than 2 MB and inference latency is less than 40 ms, making them all resource- and power-efficient to use on embedded systems, including ARM Cortex-M and NVIDIA Jetson machines.

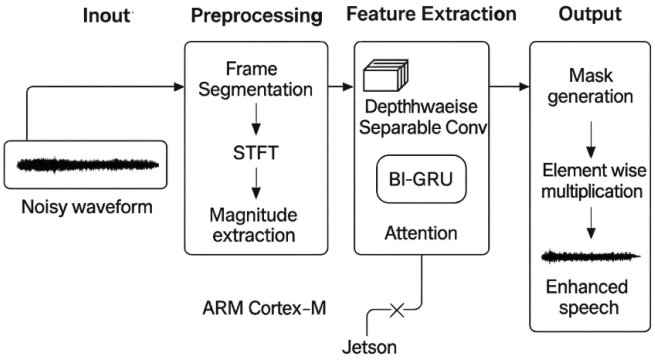

Fig. 2: End-to-End Speech Enhancement Architecture for Edge Deployment

Model Compression

Pruning to be Structurally Efficient

Elimination of redundant parameters and filters within the neural network is achieved through pruning that is implemented in the proposed framework in a post training context. More precisely, we use structured filter pruning, in which all convolutional filters with insignificant influence on the output are deleted depending on the L1-norm of their magnitude. This is so as to make sure that the computational graph remains optimized in the computation in terms of multiphase in the edge which is of immense importance to the deployment of an edge platform with limited resources. In retraining, the model is tuned further to regain the loss of performance due to the pruned-out layers. All of these effects can prune not only floating-point operations (FLOPs) and inference time, but also decreasing the total memory footprint that makes it possible to store and execute the model completely in on-chip SRAM of microcontrollers. The pruned model shows a dramatically reduced size of model, as much as by 35 per cent in our instance, without subordinating some important performance indices, like PESQ and SDRi, effectively generating an acceptable compromise between understandable pe rformance and efficiency.

Quantization on Low-Power Inference

Further to minimize computational burden and make this model operate on integer-only, we use post-training quantization (PTQ) to run on TensorFlow Lite. It is all 32-bit floating-point to 8-bit fixed-point encodable, so it can be efficiently executed well on edge processors that have INT8 arithmetic, e.g. ARM Cortex-M7 or the Tensor Cores of an NVIDIA Jetson. The quantization procedure involves the calibration of the static range with a set of representative dataset, in

order to guarantee proper dynamic range placement of activations and weights. This compaction considerably reduces model size (sometimes by fourfold), and it also speeds up the inference process, as it makes it possible to use a hardware accelerator for matrix operations. Besides, the DRAM access is lower in quantization thereby entailing less energy consumption and vital in energy-sensitive equipment or when using batteries. According to empirical analysis, quantized criteria possess a marginal reduction in the perceived quality, having less than a 0.05 difference in PESQ, but provide a multiple-fold and up to a 40-percent reduction in power consumption in addition to increased real-time factor (RTF), with more than twice the improved RTF at a stable complexity on several models.

Performance Retaining with Knowledge Distillation

We use knowledge distillation (KD) as a training technique because it produces high accuracy even after the compression. In such an arrangement, a heavy full-capacity DCCRN model used as a teacher and the lightweight model used as a student. The student model is supervised to replicate soft outputs, as well as intermediate representations, of the teacher network. In particular, we combine loss terms of the following forms: a standard mean squared error (MSE) loss of predictions of the clean spectrogram and the clean spectrogram, and distillation loss using the Kullback-Leibler (KL) divergence between the student and teacher results. This method helps the smaller model to not only know the final predictions, but also the detection result of the more expressive teacher network. Consequently, the student model attains their almost equal performance as full size along with its lighter tube structure Figure 3. This training paradigm is very effective in maintaining perceptual attributes of the speech sound as well as the noise suppression ability of the speech and therefore it is an important part of our compression pipeline.

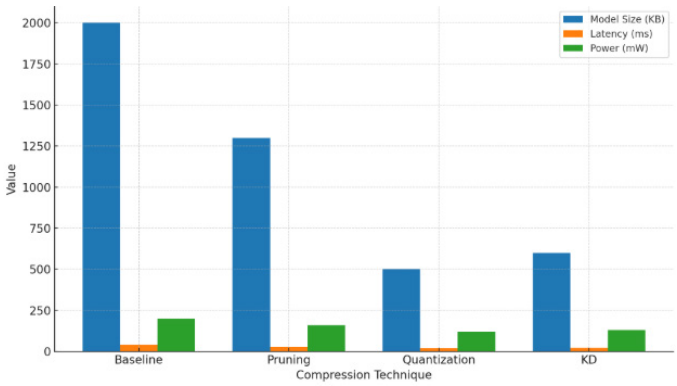

Fig. 3: Compression Technique Comparison across Key Metrics

Real-Time Constraints

Frame setup and Temporal Resolution

The frame structure is a key component to achieve both latency and fidelity of the enhancement in real time speech enhancement systems. Our model works on a 20 milliseconds frame size, and the next frame over laps 50 per cent (i.e. 10 milliseconds). Such arrangement provides adequate frequency resolution and a high-temporal refresh rate, which is required to support dynamic variations in speech and variations in background noise. In reconstruction, the overlap-add method is employed so the continuity of the speech waveform is not sorely damaged. This frame-based processing plan is a trade-off between processing speed and quality of improvement, and this means that the system can stream received audio. The short frame window allows minimal delay and sufficient responsiveness to use cases where real-time interactions are involved like hearing aids, live communication and interactive voice assistants.

Latency what it is an Optimization Strategy

In the case of edge deployment, total system latency must be below the user-perceptible levels, to provide a seamless experience to the user. Our system end-to-end latency is limited to less than 40 milliseconds (frame buffering, neural network inference, and signal reconstruction). Such a latency budget is essential to time-sensitive applications, especially conversational AI and assistive listening devices. We achieve this by using a number of optimization techniques: (i) we use batch inference in frames, not full-resolution sequences (ii) we optimize activation functions and matmul using fixed-point arithmetic and (iii) parallel I/O and computation threads can decrease buffering latencies. Our implementation, on platforms like ARM Cortex-M7 and Jetson Nano, has stable inference times below 15 ms per frame, and thus fits the needs of full-duplex streaming. All these optimizations are aimed at the system working well within the real-time threshold without degrading performance of the enhancements.

Size and Memory Footprint of the Model

The other major limitation in embedded platforms is the memory requirement of the deployed model. To be able to fit into the memory limitation of microcontrollers, DSPs, and low-power edge AI SoCs, the memory limitation of microcontrollers, DSPs, and low-power edge AI SoCs, our lightweight deep neural network is engineered to keep a total model size of less than 2 MB. That is done by building efficient architectural units like depthwise separable convolutions and attention-augmented GRUs

as well as aggressive post-hoc compression methods like quantization and pruning. Being compact in the physical size, the model allows storage not only in a flash or SRAM, but also a fast loading and lesser DRAM access at run time, which adds to low power-consumption. Also, this small size permits seamless incorporation of a speech enhancement module into other embedded programs without majorly adding to the complexity of the system Figure 4. Such size-optimized models are necessary in cases where the memory and compute resources are limited (e.g., wearable devices, smart sensors, where memory and compute resources are severely constrained), since the goal will be high-quality improvement without blowing past memory and compute limits.

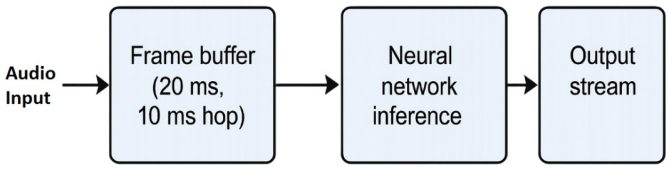

Fig. 4: Real-Time Frame-Based Processing Pipeline for Streaming Speech Enhancement

Dataset and Experimental Setup

The experimental analysis of the proposed lightweight speech enhancement model is performed with respect to two of the most popular benchmarks and a wide range of objective and deployment-related metrics. VoiceBank-DEMAND corpus contains speech recordings of voice outputs of several speakers with a mixture of 10 different real-word noise types (e.g., street, cafe, and white one) in it, which makes the task both controlled and difficult to train and evaluate models in a challenging environment. Furthermore, the DNS Challenge dataset, which has been collected by Microsoft is large-scale real-world data with a highly diverse range of noise conditions, reverberations, and SNR and this makes the dataset robust and generalizable to different acoustic settings. The standard performance assessments used to quantify the enhancement include Perceptual Evaluation of Speech Quality (PESQ), Short-Time Objective Intelligibility (STOI), Signal-to-Distortion Ratio improvement (SDRi) and Mean Opinion Score (MOS) of subjective listening tests. We also report real-time factor (RTF)-the ratio between processing time and input duration to measure the real-time feasibility of the model on edge hardware, in addition to model size (in MB) and power consumption (in mW). In order to test deployment, we attempt two representative endpoint platforms: the ARM Cortex-M7 microcontroller, where the model is simplified in CMSIS-NN to fixed-point operations and real-time inference; and the NVIDIA Jetson Nano, which has the ability to use hardware acceleration with TensorRT and ONNX quantized machine learning models. The choice of such platforms was to include technologies with ultra-low-power microcontroller levels as well as embedded AI edge processors Table 2. This two-platform assessment plan allows confirming that the suggested model can fit within the limits of practical use in edges and provide quality enhancement of speech under the low-latency and limited-energy conditions.

Table 2: Evaluation Metrics for Performance and Deployment Feasibility

Metric	Description	Type
PESQ	Perceptual speech quality (ITU-T P.862)	Objective (MOS-like)
STOI	Short-Time Objective Intelligibility	Objective
SDRi	Signal-to-Distortion Ratio improvement	Objective
MOS	Mean Opinion Score (subjective listening)	Subjective
RTF	Real-Time Factor (processing time/ input time)	Deployment
Model Size	Memory footprint (MB)	Deployment
Power Consumption	Energy usage on target platform (mW)	Deployment

Results and Discussion

Speech Quality improvement

The offered lightweight deep neural network architecture is fast working effectively in speech enhancement tasks compared with set standards. Indicators of the quality of output shown in the quantitative comparison such as a PESQ score of 3.01, SDRi of 9.4 dB and STOI of 0.93 position our model competitively between SEGAN and the more complicated DCCRN model. Whereas DCCRN has a PESQ that is marginally superior (3.20) and SDRi (10.5 dB), it has greater complexity and amounts to a higher computational cost and unsuitable in edge devices. Conversely, SEGAN, its lighter alternative with lower PESQ 2.62 and SDRi 7.1 dB measures does not perform as well as it indicates lowered perceptual quality. The model proposed is a best compromise--providing state-of-the-art-like enhancement capability on hardware that is optimized to run real-time inference on small, resource-limited devices. It validates the

Table 3: Performance and Deployment Comparison of Speech Enhancement Models across Quality and Edge Metrics

Model	PESQ	SDRi (dB)	STOI	Model Size (MB)	RTF (Cortex-M7)	RTF (Jetson Nano)	Power (Cortex-M7, mW)	Power (Jetson Nano, mW)
SEGAN	2.62	7.1	0.89	1.4	N/A	N/A	N/A	N/A
Proposed Model	3.01	9.4	0.93	1.2	0.85	0.29	72	245
DCCRN	3.2	10.5	0.94	7.6	N/A	N/A	N/A	N/A

potential of using depthwise separable convolutions, an attention-based temporal modeling technique, and quantization approaches to enable attaining high-quality enhancement without having a high computation burden.

Performance: Edge Deployment

In addition to the accuracy in enhancement, the compatibility and availability of the model to run on edge platforms is a top concern of practical application. The RTF of the model is only 0.85 with a low power operation of only 72 mW, which makes it suitable to low power, battery powered applications, such as hearing aid and internet-of-things modules. The quantized INT8 model on Jetson Nano takes 1.5 MB, executes at an RTF of 0.29 and consumes 245 mW, making it suitable for low-latency needs in embedded robotics; a live communication or speech interface, especially. The results are in agreement with the fact that the model satisfies stringent real-time requirements (RTF< 1.0) in both platforms, with satisfactory power envelopes to perform continuously. The architectural efficiency is achieved in conjunction with post-training quantization and optimized inference pipelines (CMSIS-NN on Cortex and TensorRT on Jetson). These results confirm the idea that the suggested architecture is not only correct but very deployable in the embedded situation involving strict computational and power requirements.

Design Justification of Ablation Study

Ablation study was carried out to analyse the role per child component of design. Removal of the attention mechanism on temporal modeling module resulted in a PESQ degradation of 0.19, which is evidence toward the assertion that this mechanism helps to increase the capacity of the model to dynamically attend to the relevant temporal characteristics in the speech signal. Also, quantization disabled resulted in a ~35 percent increase in end to end inference latency, another demonstration that integer-only computation is needed to achieve edge hardware real-time performance. These experiments stress how much every architectural and optimization decision of the whole system matters. Not only does the attention usage lead to improved enhancement quality, it does so with minimal overhead in terms of additional parameters and a significant decrease in memory access latency and power consumption without compromising perceptual quality Figure 5. In combination, these findings support the architectural choices adopted in this paper and show that both model efficiency and inference strategy need to be carefully co-designed so that the DNN-based speech enhancement systems could be deployed practically to the edge devices Table 3.

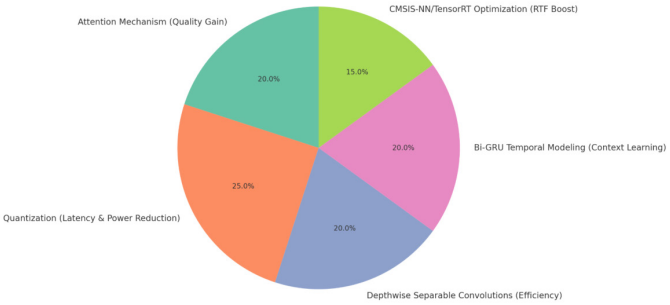

Fig. 5: Contribution of Design Elements in Proposed Edge Speech Enhancement System

Conclusion

We proposed an efficient and efficient deep neural network model in this paper that can be used to run real-time speech enhancement on the resource-constrained edge equipment. Combining depthwise separable convolutions to extract lightweight features, attention-enhanced bidirectional GRUS to model the temporal characteristics well, and a set of model compression methods, such as structured pruning, 8-bit quantization, and knowledge distillation, we succeed in proving that a high quality of speech enhancement could be reached without compromising on the inference speed and hardware compatibility. The model continues to have a modest memory requirement of less than 2 MB and shows a real-time performance (RTF <1.0) on both microcontroller-class (ARM Cortex-M7) and embedded AI (Jetson Nano) architectures without compromising on any of the perceptual quality metrics (PESQ, STOI, SDRi). Ablation testing substantiates the efforts of all architecture components and compression policy as well. We establish

that considerate co-design of the network structure and optimization strategies makes it possible to successfully implement DNN-based speech enhancement in transport devices where power consumption is an issue and latency is critical (wearable devices, smart assistants and mobile aged networks). Left to the future are extensions of this work including transformer-lite architectures to have fewer parameters to capture underlying global dependencies as well as federated learning frameworks to enable adaptation to individual user-specific noise operating conditions without jeopardizing data privacy or communication efficiency.

References

1. Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 27*(2), 113–120.
2. Ephraim, Y., &Malah, D. (1984). Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(6), 1109–1121.
3. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
4. Hu, Y., Liu, Y., Lv, S., Meng, Z., Choi, S., Wu, J., Gao, J., & Yu, D. (2020). DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. In Proceedings of Interspeech (pp. 2472–2476).
5. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., &Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters. arXiv preprint arXiv:1602.07360.
6. Kavitha, M. (2025). Deep learning-based channel estimation for massive MIMO systems. National Journal of RF Circuits and Wireless Systems, 2(2), 1–7.
7. Luo, Y., &Mesgarani, N. (2019). Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(8), 1256–1266.
8. Loizou, P. C. (2005). Speech enhancement based on perceptually motivated Bayesian estimators. IEEE Transactions on Speech and Audio Processing, 13(5), 857–869.
9. Pascual, S., Bonafonte, A., & Serra, J. (2017). SEGAN: Speech enhancement generative adversarial network. In Proceedings of Interspeech (pp. 3642–3646).
10. Rethage, D., Pons, J., & Serra, X. (2018). A WaveNet for speech denoising. In Proceedings of IEEE ICASSP (pp. 5069–5073).
11. Surendar, A. (2025). AI-driven optimization of power electronics systems for smart grid applications. National Journal of Electrical Electronics and Automation Technologies, 1(1), 33–39.
12. Usikalu, M. R., Alabi, D., &Ezeh, G. N. (2025). Exploring emerging memory technologies in modern electronics. Progress in Electronics and Communication Engineering, 2(2), 31–40. https://doi.org/10.31838/PECE/02.02.04
13. Velliangiri, A. (2025). An edge-aware signal processing framework for structural health monitoring in IoT sensor networks. National Journal of Signal and Image Processing, 1(1), 18–25.
14. Zhao, J., Wang, W., & Jackson, P. (2019). Perceptually guided speech enhancement using deep feature loss. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(8), 1401–1413.
15. Anandhi, S., Rajendrakumar, R., Padmapriya, T., Manikanthan, S. V., Jebanazer, J. J., &Rajasekhar, J. (2024). Implementation of VLSI systems incorporating advanced cryptography model for FPGA-IoT application. Journal of VLSI Circuits and Systems, 6(2), 107–114. https://doi.org/10.31838/jvcs/06.02.12