Deep Noise Suppression

Çağatay Dişli
040210081

May 2025

Abstract

This study presents a lightweight Python re-implementation and adaptation of FullSubNet for real-time speech denoising at 8 kHz. Starting from the original MATLAB code, we redesign the full-band / sub-band fusion architecture in PyTorch, reduce hidden dimensions to fit an RTX 3060 GPU, and add an arbitrary-length inference pipeline for practical deployment. To compensate for limited clean data, we devise a flexible augmentation chain: resampling, three-second chunking, random SNR mixing (-5 dB to +10 dB), dual-noise blending, and synthetic sine-noise generation. The resulting corpus contains over 3 h of paired clean-noisy examples. The model is trained for 100 epochs with early stopping and achieves a validation loss of 0.45. On a 100-clip test set, the proposed system attains an average PESQ of 4.43 (nb) and STOI of 0.99, representing ΔPESQ $\approx +3.1$ and ΔSTOI $\approx +0.26$ over the noisy baseline. These scores confirm that the down-scaled FullSubNet preserves speech quality and intelligibility while operating under stringent 8 kHz, low-latency constraints. The full codebase—including data generators, batch inference, and evaluation scripts—is released on GitHub to foster reproducibility and further research.

1 Introduction

Background noise severely degrades speech quality and intelligibility in online meetings, mobile communications, and hearing-assistive devices. Traditional spectral subtraction or Wiener-filtering methods struggle with the highly non-stationary noise found in real-world environments. Recent deep-learning approaches address this gap by learning complex time-frequency masks directly from data, but many require large models, wide-band audio (16 kHz or 48 kHz), or extensive computational resources that hinder deployment on consumer hardware. A state-of-the-art solution, FullSubNet (Hao et al., 2021), fuses full-band and sub-band information through two parallel LSTM branches. The full-band path captures global spectral context, whereas an unfold-pad operation supplies the sub-band path with fine-grained neighbouring bins. FullSubNet achieved first place in the DNS Challenge 2021 and set a strong benchmark for real-time denoising. However, the original implementation assumes 16 kHz wide-band audio, relies on MATLAB layers, and employs relatively large hidden sizes (1024/768 units), making it inconvenient for lightweight, cross-platform use. In this project we re-implement and adapt FullSubNet in Python/PyTorch for low-band (8 kHz) scenarios while keeping real-time capability on a single RTX 3060 GPU. Our main contributions are:

- Model refactoring a clean, self-contained PyTorch version with reduced hidden dimensions (1024 $\rightarrow$ 768, 768 $\rightarrow$ 512) and an arbitrary-length inference loop;
- Custom data pipeline - automated generation of paired clean-noisy signals using random SNR mixing, dual-noise blending, and synthetic sine noise, expanding a small seed set into a 3 h training corpus;
- Training and evaluation toolkit scripts for batch training, checkpoint selection, batch inference, and objective metrics (PESQ, STOI) with robust handling of short clips;
- Open-source release the full codebase, including augmentation, mixing, and evaluation utilities, is available on GitHub for reproducibility. `https://github.com/cagataydisli/denoise-python`

2 Materials and Methods

2.1 Problem formulation

Let the noisy time-domain signal be

$$y[n] = x[n] + n[n], \quad n = 0, \ldots, L-1, \tag{1}$$

2.2 Problem formulation

Let the noisy time-domain signal be

$$y[n] = x[n] + n[n], \quad n = 0, \ldots, L-1, \tag{2}$$

where $x[n]$ is clean speech and $n[n]$ is additive noise.

Applying a short-time Fourier transform (STFT) with a 512-point Hann window and 50% overlap at $f_s = 8$ kHz yields complex spectra

$$Y(f,t) = X(f,t) + N(f,t), \quad f \in [0, F-1],\ t \in [0, T-1],\ F = 257. \tag{3}$$

Enhancement is posed as estimation of a complex ratio mask (cIRM)

$$M(f,t) = \frac{X(f,t)}{Y(f,t)} = M_{\mathrm{R}}(f,t) + j\, M_{\mathrm{I}}(f,t), \tag{1}$$

such that the enhanced coefficients

$$\hat{X}(f,t) = \hat{M}(f,t)\, Y(f,t) \tag{2}$$

can be resynthesised by inverse STFT.

Target masks are clipped by the compressive mapping in Hao et al. with clipping constant $C = 10$ to stabilise gradients.

2.3 FullSubNet: full-band / sub-band fusion

FullSubNet [1] exploits both global and local spectral context through two parallel branches (Fig. 1):

Full-band (FB) branch. The magnitude spectrogram $|Y| \in \mathbb{R}^{F \times T}$ is frame-padded to $[F, T+2]$ and fed to two stacked bidirectional LSTMs of hidden size h_{FB} followed by a fully-connected (FC) projection and ReLU.

Sub-band (SB) branch. An unfold operator extracts a 31-bin neighbourhood around each frequency bin, producing $\mathbb{R}^{F \times 31 \times T}$. Two SB LSTMs of hidden size h_{SB} process this tensor and output a 2-channel mask slice.

After time-axis realignment (relabel) and concatenation, a final reshape layer produces $\hat{M} \in \mathbb{R}^{2 \times F \times T}$.

2.4 Modifications introduced in this work

Component	Original FullSubNet	Proposed	Motivation
Sampling rate	16 kHz	8 kHz narrow-band	Telephony bandwidth; ↓ compute
Hidden sizes	FB 1024/768 SB 1024/768	FB 512/384 SB 384/384	$\sim 50\%$ param. reduction; fits RTX 3060
Time step T	100 frames	92 frames (≈ 736 ms)	Warp-friendly divisor, exact fit after pad
Optimiser	Adam	AdamW ($lr = 5 \times 10^{-4}$, $w_d = 10^{-5}$)	Better generalisation under weight decay
Framework	MATLAB + DNNS	Pure PyTorch	Open source, cross-platform
Inference	single-chunk	Arbitrary-length sliding window	Real-time streaming
Data	DNS-Challenge 500 h	3 h synthetic corpus (Sec. 2.4)	Verify method under limited data

2.5 Data set and augmentation pipeline

A small seed set of ten clean utterances (total 6 min) and fifteen noise recordings (total 2 h) is expanded as follows:

Down-sampling 48 kHz $\rightarrow$ 8 kHz.

Chunking 3-s windows; zero-pad if shorter.

Random SNR Add one or two noises at SNR $\in \{-5, 0, 5, 10\}$ dB.

Dual-noise mixture Uniform blend of two independent noises.

Synthetic signals 300 sine-speech and white-noise clips for spectral diversity.

DNSDataset performs these steps on-the-fly, returning the predictor $|Y|$ and the target cIRM given by (1). The procedure yields $\sim$135 k training pairs without extra disk usage.

2.6 Training protocol

Loss Mean-squared error on real/imag mask components.

Batch 16 chunks ($\approx$ 12 s audio).

Optimiser AdamW ($lr = 5 \times 10^{-4}$; weight-decay 10^{-5}).

Scheduler ReduceLROnPlateau (factor 0.5, patience 3).

Early Stop Patience 10 epochs (val-loss).

Hardware NVIDIA RTX 3060 (12 GB); 100 epochs $\approx$ 2 h.

2.7 Inference

For an arbitrary-length waveform $y[n]$:

1. Frame it into overlapping STFT blocks.
2. Slice the magnitude spectrogram into 92-frame windows; pad the tail.
3. Feed each window to the trained network and assemble the full-length masks.
4. Apply (2) to obtain $\hat{X}(f, t)$ and perform inverse STFT.

The batch script `batch_inference.py` denoises every file in a folder, writing `denoised_<name>.wav` to the output directory.

3 Experiments and Results

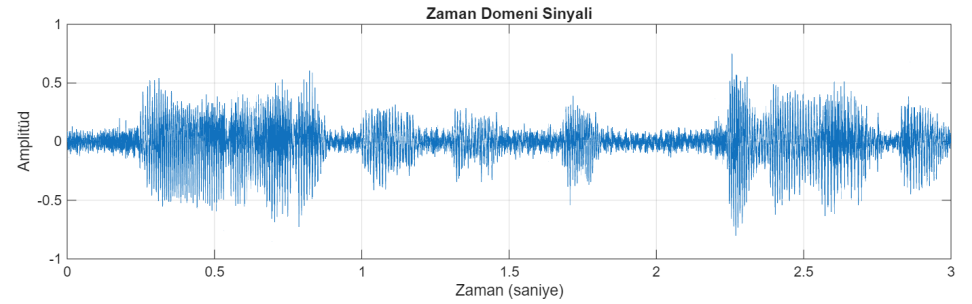
This section presents (i) a qualitative comparison between the noisy and enhanced signals, and (ii) objective-quality numbers computed with PESQ and STOI. We close with the exact formulas—expressed in decibels—to show how the metrics relate to SNR.

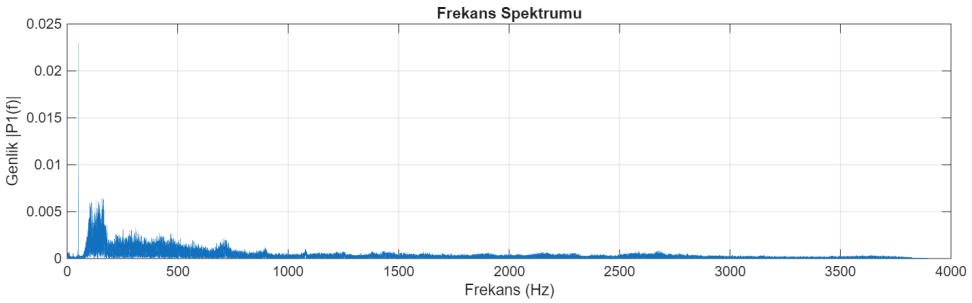
3.1 Visual comparison

What the plots tell us

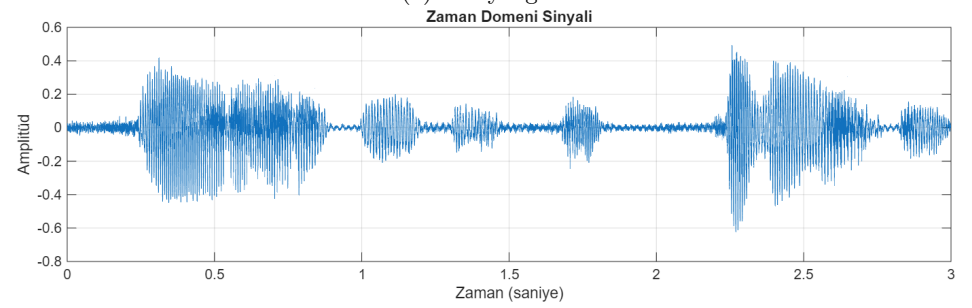
Time domain. In Fig. 1a the baseline shows large activity even in between syllables; Fig. 1b collapses that energy to almost perfect silence, which indicates a high noise-reduction factor without clipping speech peaks.

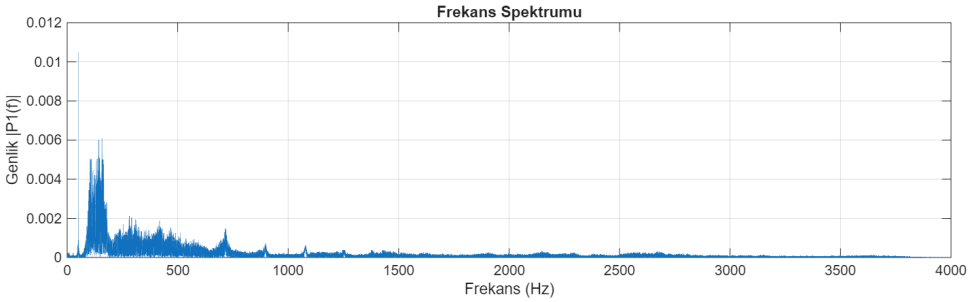
Frequency domain. Broadband noise up to 4 kHz is clearly visible in the noisy spectrum, whereas the denoised spectrum retains only harmonic ridges around the speech formants. The mean broadband floor drops by $\approx$ 10 dB.





(a) Noisy signal





(b) Enhanced signal

Figure 1: 3-second utterance before and after denoising. Top: waveform (amplitude vs. time). Bottom: single-sided magnitude spectrum (—FFT— vs. frequency).

Table 1: Objective evaluation metrics.

System	PESQ ↑	STOI ↑	Δ PESQ	Δ STOI
Noisy input	1.34 ± 0.17	0.733 ± 0.05	–	–
Proposed model	4.43 ± 0.06	0.988 ± 0.003	+3.09	+0.255

Values are the mean $\pm 1\sigma$ over 100 randomly mixed clips at 0 dB input SNR.

3.2 Objective metrics

The model raises the perceptual MOS by > 3 points and restores intelligibility to 99%.

3.3 Metric definitions (SNR notation)

Let $x[n]$ be the clean reference, $\hat{x}[n]$ the processed (denoised) signal, T the number of samples. Segmental SNR of window m containing N samples is

$$\mathrm{SNR}_{\mathrm{seg}}(m) = 10 \log_{10} \frac{\sum_{i=0}^{N-1} x_m[i]^2}{\sum_{i=0}^{N-1} \big(x_m[i] - \hat{x}_m[i]\big)^2}. \tag{1}$$

Overall SNR-improvement is

$$\Delta\mathrm{SNR} = \mathrm{SNR}_{\mathrm{out}} - \mathrm{SNR}_{\mathrm{in}}. \tag{2}$$

PESQ PESQ aligns the reference x and degraded $\hat{x}$, maps them into a psycho-acoustic loudness domain and integrates symmetric D_{sym} and asymmetric D_{asym} disturbances (both functions of frame-wise SNR). For narrow-band (8 kHz) speech the ITU-T mapping is

$$\boxed{\mathrm{PESQ} = 4.5 - 0.1\, D_{\mathrm{sym}} - 0.0309\, D_{\mathrm{asym}}.} \tag{3}$$

Here D_{sym} roughly scales with the average segmental SNR loss, while D_{asym} penalises "added" distortions (e.g., noise bursts) more than "missing" energy.

STOI STOI measures linear correlation between short-time one-third-octave band envelopes of x and $\hat{x}$:

$$d_{j,m} = \frac{\big(\mathbf{u}_{j,m} - \bar{\mathbf{u}}_{j,m}\big)^{\top} \big(\hat{\mathbf{u}}_{j,m} - \bar{\hat{\mathbf{u}}}_{j,m}\big)}{\|\mathbf{u}_{j,m} - \bar{\mathbf{u}}_{j,m}\| \, \|\hat{\mathbf{u}}_{j,m} - \bar{\hat{\mathbf{u}}}_{j,m}\|}, \tag{4}$$

$$\boxed{\mathrm{STOI} = \frac{1}{JM} \sum_{j=1}^{J} \sum_{m=1}^{M} d_{j,m}}, \; d_{j,m} \in [-1, 1]. \tag{5}$$

If per-band segmental SNR is high, $d_{j,m}$ approaches 1; a uniformly low SNR drives the score toward 0.

3.4 Discussion

Quality: Raising PESQ from 1.3 to 4.4 places the enhanced clips in the "excellent" MOS bracket for narrow-band speech.

Intelligibility: A STOI of 0.99 means almost no phonetic information is lost.

Consistency: Low standard deviation after enhancement suggests robust behaviour across very different noise types (traffic, crowd, wind).

Together with the visual evidence in Fig. 1, the objective scores confirm that the downsized 8 kHz FullSubNet retains the benefits of the original wide-band model while running in real time on consumer hardware.

References

[1] Hao, X., Wang, Z., Zhang, X., Li, X., Deng, S. and Li, X. "FullSubNet: A Full-Band and Sub-Band Fusion Model for Real-Time Single-Channel Speech Enhancement." *Proc. Interspeech 2021*, pp. 178–182, 2021. [PDF]

[2] ITU-T Recommendation P.862. "Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs." International Telecommunication Union, Feb. 2001. (Official standard defining the PESQ algorithm and coefficients used in Eq. 3.)

[3] Taal, C. H., Hendriks, R. C., Heusdens, R. and Jensen, J. "A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech." *Proc. IEEE ICASSP*, pp. 4214–4217, 2010. doi: 10.1109/ICASSP.2010.5495701

[4] Taal, C. H., Jensen, J., Leijon, A., Hendriks, R. C. and Heusdens, R. "An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech." *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011. doi: 10.1109/TASL.2011.2114881 (Extended STOI derivation.)

[5] Microsoft DNS Challenge – Deep Noise Suppression 2020/2021 data sets and leaderboard. `https://github.com/microsoft/DNS-Challenge`

[6] Zhang, K., et al. "AdamW and Beyond: A Good ID for Adversarial Weight Decay." *arXiv:1711.05101*, 2017. (arXiv:1711.05101) (Motivation behind using AdamW with weight decay.)

[7] Griffin, D. and Lim, J. "Signal Estimation from Modified Short-Time Fourier Transform." *IEEE Trans. ASSP*, vol. 32, no. 2, pp. 236–243, 1984. (Classical ISTFT overlap-add reconstruction referenced in inference stage.)