Computers and Artificial Intelligence；Vol.2, No.3; 2025
E-ISSN: 3006 -3728 P-ISSN：3006-371X
DOI: https://doi.org/10.70267/cai.25v2n3.4554
Published by: Zeus Press

A Comparative Analysis of FCNN and CNN Architectures for Speech Denoising Across Diverse Noise Frequencies

Xueqing Ma*, and Yuting Cui

College of Information Engineering, Hangzhou Dianzi University, Hangzhou, China

**Corresponding author: Xueqing Ma*

Abstract

Speech denoising remains a critical challenge in audio signal processing, especially under non-stationary noise conditions. While convolutional neural networks (CNNs) have been widely adopted for speech enhancement, the potential of fully connected neural networks (FCNNs) remains underexplored, particularly under frequency-varying noise scenarios. This study presents a systematic comparative analysis of FCNN and CNN architectures for speech denoising across multiple noise frequencies. Using the Common Voice dataset, we introduced diverse noise types at 8 kHz, 16 kHz, and 44 kHz to evaluate the denoising performance of both models. Experimental results demonstrate a frequency-dependent performance disparity: at 8 kHz, both models perform similarly, with CNN showing marginally higher Signal-to-Noise Ratio (SNR) and Root Mean Square Error (RMSE). At 16 kHz, CNN achieves significantly higher SNR albeit with increased RMSE, indicating a trade-off between noise suppression and spectral fidelity. At 44 kHz, CNN comprehensively outperforms FCNN, attaining superior SNR (4.80, +0.04) and lower RMSE (2.6826, –0.1556). These findings underscore the architectural advantages of CNNs in broad-frequency and complex noise environments, while revealing FCNN's applicability in narrowband scenarios. This research highlights the necessity of frequency-aware model selection and provides novel insights into the comparative efficacy of FCNN and CNN in speech denoising.

Keywords

speech denoising, FCNN, CNN, noise frequency analysis, SNR, RMSE, common voice dataset

1. Introduction

Noise pollution has become a global environmental issue, exerting profound impacts on quality of life and public health. In practical scenarios, speech signals are invariably contaminated by environmental noise, channel distortions, reverberation, and other interferences (Hu & Loizou, 2007; Loizou, 2013), leading to severe degradation of speech quality that significantly undermines the performance of communication systems and human-computer interaction. Specifically, industrial settings featuring heavy machinery operation and manufacturing activities generate substantial acoustic interference that disrupts voice communication systems. Military environments present even greater challenges, where gunfire and explosive noises introduce extreme acoustic disturbances that compromise the intelligibility and completeness of speech information transmission. These pressing challenges have made speech denoising an essential research direction in audio signal processing (Wang & Chen, 2018).

The core objective of speech denoising technology is to process human voice-containing audio by suppressing background acoustic waves and various noise components, thereby enhancing speech intelligibility and improving overall signal quality (Azarang & Kehtarnavaz, 2020). Traditional speech denoising approaches, including spectral subtraction (Boll, 1979; Saha et al., 2018) and Wiener filtering (Nuha & Absa, 2022; Shamsa et al., 2016), operate based on linear acoustic assumptions and demonstrate reasonable performance under stationary noise conditions. However, their effectiveness deteriorates rapidly in non-stationary, complex noise environments such as subways and restaurants. Statistical model-based methods, incorporating Gaussian mixture models (Hao et al., 2010) and hidden Markov models (Aroudi et al., 2015; Xiang et al., 2022), remain constrained by manual feature engineering limitations and struggle to capture the high-dimensional nonlinear relationships between noise and clean speech.

The advancement of deep learning has catalyzed exploration in speech enhancement, driving transformative changes in denoising methodologies. Deep neural networks facilitate automated end-to-end feature mapping from noisy to clean speech through multiple nonlinear transformations (Mai & Goetze, 2025; Pascual et al., 2019). Capitalizing on the time-frequency correlation characteristics of speech, recurrent neural networks (RNNs) have proven effective in handling long-term temporal dependencies (Le et al., 2022; Pandey & Wang, 2022), while long short-term memory (LSTM) networks further enhance the modeling of speech dynamics (Huang & Wu, 2023; Tang et al., 2020). Although RNNs demonstrate notable performance in source separation tasks, they remain susceptible to gradient vanishing and explosion problems that significantly impact training efficiency. In recent years, convolutional neural networks (CNNs) have garnered significant attention in speech enhancement research (Lin et al., 2025; Soleymanpour et al., 2023). Prior works, such as Garg and Sahu (2022), have demonstrated the effectiveness of CNNs in speech denoising tasks, while Balasubrahmanyam and Valarmathi (2024) proposed an adaptive residual CNN with encoder-decoder architecture to further improve speech quality. In contrast, research on fully connected neural networks (FCNNs) for denoising remains extremely scarce, with limited comparative studies examining their performance against CNNs under varied acoustic conditions.

To address this research gap, this study makes several key contributions through a comprehensive comparative analysis of FCNN and CNN models across multiple noise frequency bands. We introduce a novel evaluation framework incorporating diverse noise types at 8 kHz, 16 kHz, and 44 kHz, enabling detailed analysis of model behavior under different spectral characteristics. Furthermore, our work provides new insights into the fundamental trade-offs between noise suppression and signal fidelity through combined objective metrics (SNR, RMSE) and time-frequency analyses. By leveraging the Common Voice dataset and employing rigorous experimental protocols, this work not only clarifies the relative strengths of FCNN and CNN architectures but also offers practical guidance for model selection in real-world speech denoising applications.

2. Methodology

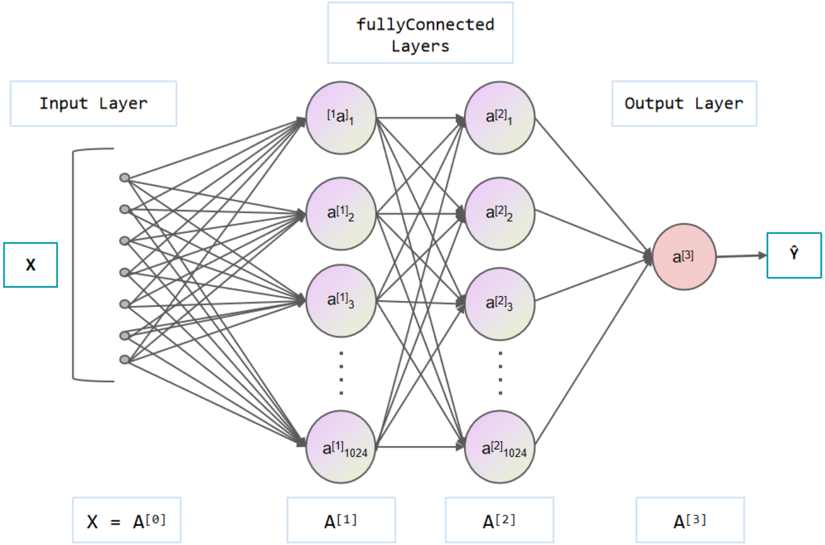
2.1 FCNN for Speech Denoising

The Fully Connected Neural Network (FCNN) adopts a typical multilayer perceptron structure (Reddy et al., 2022), featuring a fully interconnected topology between layers with computational coupling. A standard FCNN architecture consists of an input layer, one or more fully connected layers, and an output layer, as illustrated in Figure 1.

The operation of an FCNN involves the forward propagation of input values through its hierarchical structure. Processing within a single layer is performed in two stages. The first stage performs a linear transformation by computing a weighted sum of the outputs from the previous layer, along with the addition of a bias term. Denoting the input to the m-th layer as $a^{(m-1)}$, its output is given by:

$$z^{(m)}=W^{(m)}*a^{(m-1)}+b^{(m)} . \tag{1}$$

Figure 1: FCNN architecture



Where w is the weight matrix and b is the bias vector. The second stage applies a nonlinear activation function σ(·) to produce the activation value:

$$a^{(m)=}\sigma(z^{(m)}) \ . \tag{2}$$

The introduction of the activation function provides nonlinearity, enabling the network to approximate complex functional mappings. The raw input is denoted as a(0)=x and the final activation value a(M) constitutes the network's prediction.

The prediction is then compared against the ground truth to compute an error. The mean squared error (MSE) loss is commonly used for regression tasks, while cross-entropy loss is typically employed for classification tasks. The gradients of the loss with respect to the model parameters are computed via the chain rule, where ⊙ denotes element-wise multiplication and η represents the learning rate:

$$\delta^{(M)}=\frac{\partial M}{\partial a(M)}\odot\sigma'(z^{(M)}) \ . \tag{3}$$

$$((W^{(m+1)})^{T}*\delta^{(m+1)})\odot\sigma'(z^{(m)}) \ . \tag{4}$$

$$\frac{\partial M}{\partial W}=\delta^{(m)}*(a^{(m-1)})^{T},{}^{W}=W^{(m)} \ . \tag{5}$$

$$\frac{\partial M}{\partial b}=\delta^{(m)},b=b^{(m)} \ . \tag{6}$$

Subsequently, the weights and biases are updated using gradient descent:

$$W^{(m)} \leftarrow W^{(m)} -\eta*\frac{\partial M}{\partial W}. \tag{7}$$

$$b^{(m)} \leftarrow b^{(m)} -\eta*\frac{\partial M}{\partial b} \ . \tag{8}$$

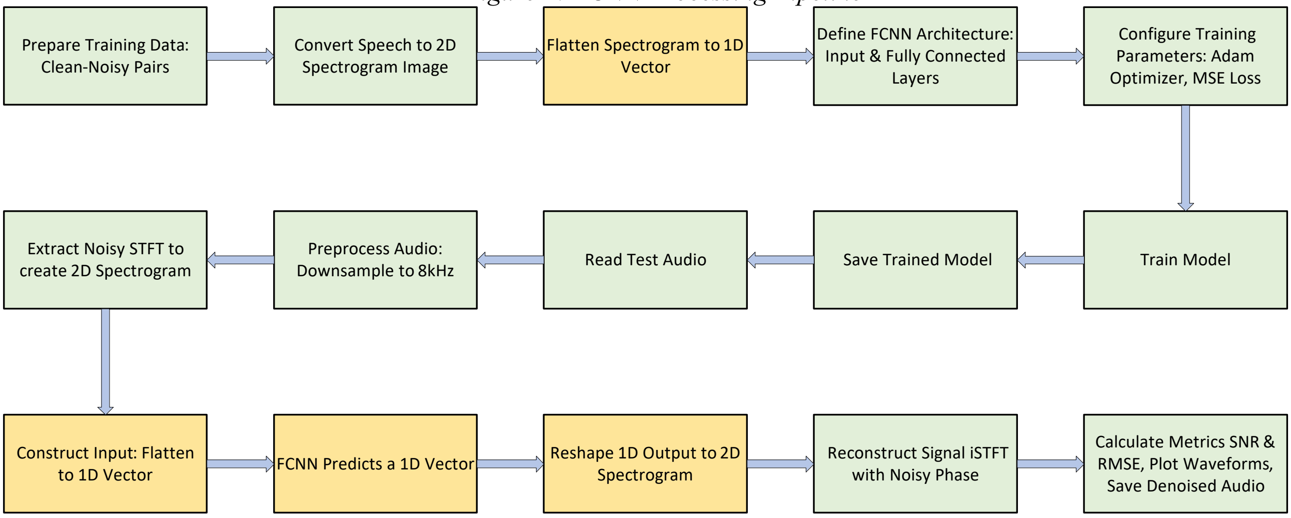
The FCNN is employed for prediction using the following structure: the network comprises two fully connected layers, each containing 1024 neurons, followed by a regression layer that computes the mean squared error (MSE) loss. The input and output dimensions are designed to be identical. Training parameters are configured with the Adam optimizer over three epochs, with a batch size of 128. To achieve rapid convergence, a relatively high learning rate is used in the initial phase of training, followed by a step-wise decay strategy adjusted periodically. Model training is performed either from scratch or via loading a pre-trained model. The total number of weights is calculated to quantify model complexity and verify the correctness of the network architecture.

The speech denoising pipeline based on a Fully Connected Neural Network (FCNN) operates as follows. The process initiates with data preparation, where a dataset comprising parallel clean and noisy audio utterances is compiled and partitioned into training and testing sets. Each audio signal is converted into a two-dimensional time-frequency representation via the Short-Time Fourier Transform (STFT), where the time and frequency axes constitute the dimensions of an image-like structure. A critical preprocessing step involves flattening this 2D spectrogram into a one-dimensional feature vector, thereby discarding the inherent spatial structure of the time-frequency representation. The input dimensions are consequently adjusted to a 1D vector for each sample.

The FCNN architecture is subsequently defined, featuring an input layer sized to match the flattened vector, multiple hidden fully-connected layers with non-linear activation functions (e.g., ReLU), and an output layer with identical dimensions to the input layer to perform spectrum estimation. The network is trained to learn a mapping from noisy input vectors to their corresponding clean target vectors. The training configuration typically employs the Adam optimizer and a Mean Squared Error (MSE) loss function, often with a scheduled learning rate decay.

For inference, a test noisy audio signal is preprocessed (downsampled to 8 kHz), and its STFT magnitude is computed and flattened into a 1D vector. This vector is fed into the trained FCNN, which predicts a 1D output vector. This output is then reshaped back into a 2D spectrogram. The final denoised time-domain signal is reconstructed by applying the inverse STFT (iSTFT) using the predicted magnitude spectrum and the phase information from the original noisy input. The performance of the system is quantitatively evaluated using metrics such as Signal-to-Noise Ratio (SNR) and Root Mean Square Error (RMSE), alongside qualitative waveform comparisons. The overall procedure is summarized in Figure 2.

Figure 2: FCNN Processing Pipeline


Prepare Training Data: Clean-Noisy Pairs
Convert Speech to 2D Spectrogram Image
Flatten Spectrogram to 1D Vector
Define FCNN Architecture: Input & Fully Connected Layers
Configure Training Parameters: Adam Optimizer, MSE Loss
Train Model
Save Trained Model
Read Test Audio
Preprocess Audio: Downsample to 8kHz
Extract Noisy STFT to create 2D Spectrogram
Construct Input: Flatten to 1D Vector
FCNN Predicts a 1D Vector
Reshape 1D Output to 2D Spectrogram
Reconstruct Signal iSTFT with Noisy Phase
Calculate Metrics SNR & RMSE, Plot Waveforms, Save Denoised Audio

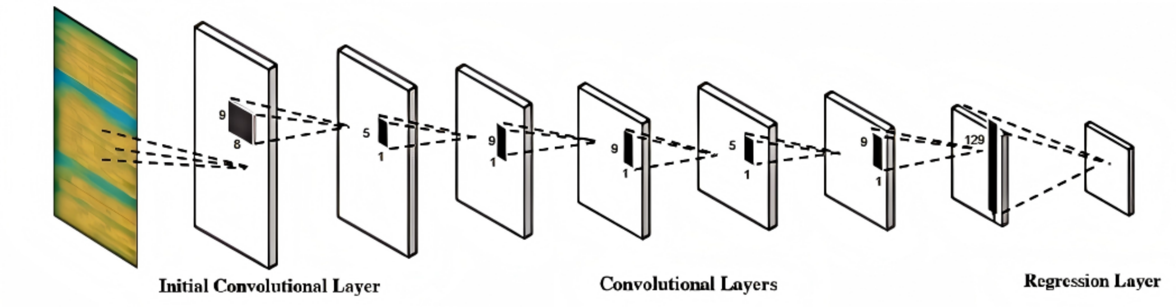
2.2 CNN for Speech Denoising

A Convolutional Neural Network (CNN) is typically composed of an input layer, convolutional layers, activation layers, pooling layers, fully-connected layers, and an output layer. The convolutional layers perform feature extraction and mapping through convolution operations, which can be categorized into one-dimensional and multi-dimensional convolutions. The activation layers introduce non-linear units to enhance the network's capability to approximate non-linear patterns. The pooling layers perform down-sampling and sparsification on the features, primarily using max pooling or average pooling. The fully-connected layers are usually placed in the later part of the convolutional model to perform secondary fitting of features and reduce loss.

A speech segment is converted into an image-like representation as input, where each frame undergoes a Fourier transform, and the time and frequency axes form the two dimensions of the image. After data preprocessing, the CNN architecture is designed as follows: the input layer is configured to receive the segmented time-series signal. The initial convolutional layer uses a large kernel of size 9×8 along the vertical (feature) dimension with a stride of 1 to preserve feature information. Along the horizontal (time) dimension, a large stride of 100 is applied to significantly compress the temporal dimension, effectively replacing the pooling layer for down-sampling and reducing computational load.

This is followed by a module repeated four times, each containing three convolutional layers. Each convolutional layer is followed by batch normalization and a ReLU activation function, progressively extracting features at different scales (with filter sizes increasing from 30 to 8 to 18) to enhance non-linear representation capacity. Stacking these modules deepens the network, enabling it to handle complex tasks. After processing through the four modules, the features become highly abstract but may contain redundancy or noise. To prevent overfitting, subsequent convolutional layers are designed with kernel sizes of 5×1 and 30 filters, and 9×1 with 8 filters. The smaller receptive field (5 time steps) performs fine-grained calibration to eliminate potential local interference introduced by the repeated modules, while the larger receptive field (9 time steps) integrates information across time steps to capture residual long-term dependencies. These layers reduce the channel count from 18 to 8, providing low-dimensional input to the final very-large-kernel convolutional layer (layer 7 with a 129×1 kernel). This two-step convolutional process achieves a “wide-to-narrow” transition, effectively removing irrelevant features. The final convolutional layer uses a kernel size of [129×1], covering the entire feature dimension, and outputs a single channel suitable for regression tasks. The overall CNN architecture and the processing pipeline are illustrated in Figure 3and Figure 4 separately.

Figure 3: CNN architecture

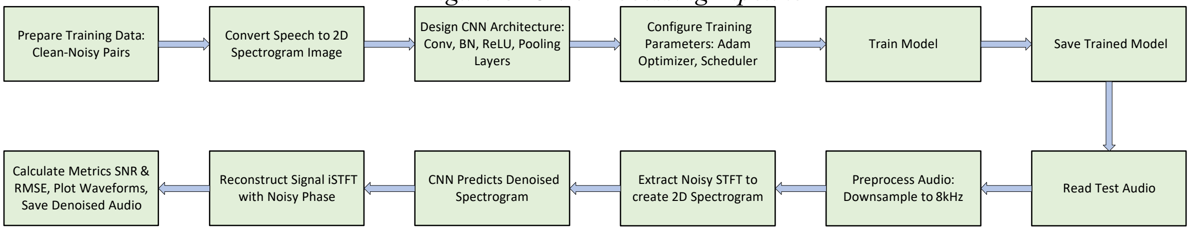


The speech denoising pipeline utilizing a Convolutional Neural Network (CNN) is designed to explicitly leverage the spatial structure of the time-frequency representation. The initial data preparation phase is similar to the FCNN approach, involving the creation of a paired dataset of clean and noisy audio signals and their conversion into 2D spectrograms via the STFT. However, a fundamental distinction is that the spectrogram is processed directly as a two-dimensional image, preserving the local correlations in time and frequency.

The CNN architecture is meticulously designed for this input structure. Initial layers often employ large convolutional kernels and strategic stride settings to perform aggressive downsampling on the time axis while preserving resolution on the frequency axis, effectively substituting for pooling layers. This is typically followed by a deep stack of convolutional modules, each consisting of multiple convolutional layers, batch normalization, and ReLU activations. These modules hierarchically extract features at various scales and levels of abstraction. The latter stages of the network often employ 1D convolutional kernels oriented along the time dimension to integrate long-term contextual information and refine the feature maps before the final output layer, which produces a spectrogram of the same dimensions as the input for regression. The training configuration employs the Adam optimizer with an MSE loss and a piecewise learning rate schedule.

For denoising, the test noisy audio is preprocessed, and its STFT magnitude is extracted to form a 2D input image for the network. The trained CNN processes this image directly, leveraging its convolutional layers to identify and enhance salient features while suppressing noise. The output of the network is a 2D spectrogram estimate of the clean speech. The time-domain signal is subsequently reconstructed via the iSTFT, using the predicted magnitude and the phase from the noisy input. The enhancement performance is rigorously evaluated using objective measures like SNR and RMSE, providing a quantitative assessment of the model's efficacy.

Figure 4: CNN Processing Pipeline


Prepare Training Data: Clean-Noisy Pairs
Convert Speech to 2D Spectrogram Image
Design CNN Architecture: Conv, BN, ReLU, Pooling Layers
Configure Training Parameters: Adam Optimizer, Scheduler
Train Model
Save Trained Model
Read Test Audio
Preprocess Audio: Downsample to 8kHz
Extract Noisy STFT to create 2D Spectrogram
CNN Predicts Denoised Spectrogram
Reconstruct Signal iSTFT with Noisy Phase
Calculate Metrics SNR & RMSE, Plot Waveforms, Save Denoised Audio

2.3 Dataset

The Common Voice dataset, pioneered by Mozilla, stands as a benchmark for large-scale, open-source speech data. Its primary distinction lies in its crowd-sourced methodology, which fosters exceptional linguistic diversity and demographic variety. A critical feature of its design is the collection of speaker-donated metadata, where contributors can optionally, and anonymously, disclose attributes such as age, gender, accent, and dialect. This metadata is exceptionally valuable for training and evaluating robust, fair speech technology models that perform well across diverse populations.

As of its latest iterations (e.g., Version 16.0 or 17.0), the corpus has surpassed 29,000 hours of validated speech across more than 120 languages. Its commitment to language inclusivity extends beyond widely spoken languages to include underrepresented and endangered ones, such as Tigre (a Semitic language spoken in Eritrea and Sudan), Meadow Mari (a Uralic language from Russia), and Toki Pona (a constructed philosophical language). It also encompasses major regional dialects and languages, including Bengali (Bangla), Min Nan Chinese (Southern Min), and Yue Chinese (Cantonese). All data is released under the Creative Commons CC-0 license, placing it in the public domain with minimal restrictions.

For this experiment, a subset of the Common Voice dataset is employed to train and evaluate the speech denoising performance of both Fully Connected Neural Network (FCNN) and Convolutional Neural Network (CNN) models. The open and diverse nature of the dataset ensures that the models are tested under a variety of acoustic conditions and speaker characteristics, providing a robust assessment of their generalization capabilities.

2.4 Evaluation Metrics

Signal-to-Noise Ratio (SNR) is defined as the ratio of the power of a signal to the power of background noise. It is widely used in communications and electronic engineering to quantify the level of desired signal relative to unwanted noise, with a higher SNR indicating better performance. The ratio is commonly expressed in decibels (dB) and calculated using the following formula:

$$\mathrm{SNR}=10\log_{10}(\frac{\mathrm{P_s}}{\mathrm{P_n}})\ . \tag{9}$$

where P_s denotes the power of the signal and P_n represents the power of the noise.

The Root Mean Square Error (RMSE) serves as a standardized measure for quantifying prediction errors by evaluating the square root of the cumulative squared deviations between predicted and actual values. It provides an aggregated indication of prediction accuracy, where a lower RMSE value corresponds to reduced error and hence improved predictive performance. RMSE is widely employed in regression tasks to assess model quality and support precision control during practical implementation. The metric is mathematically defined as:

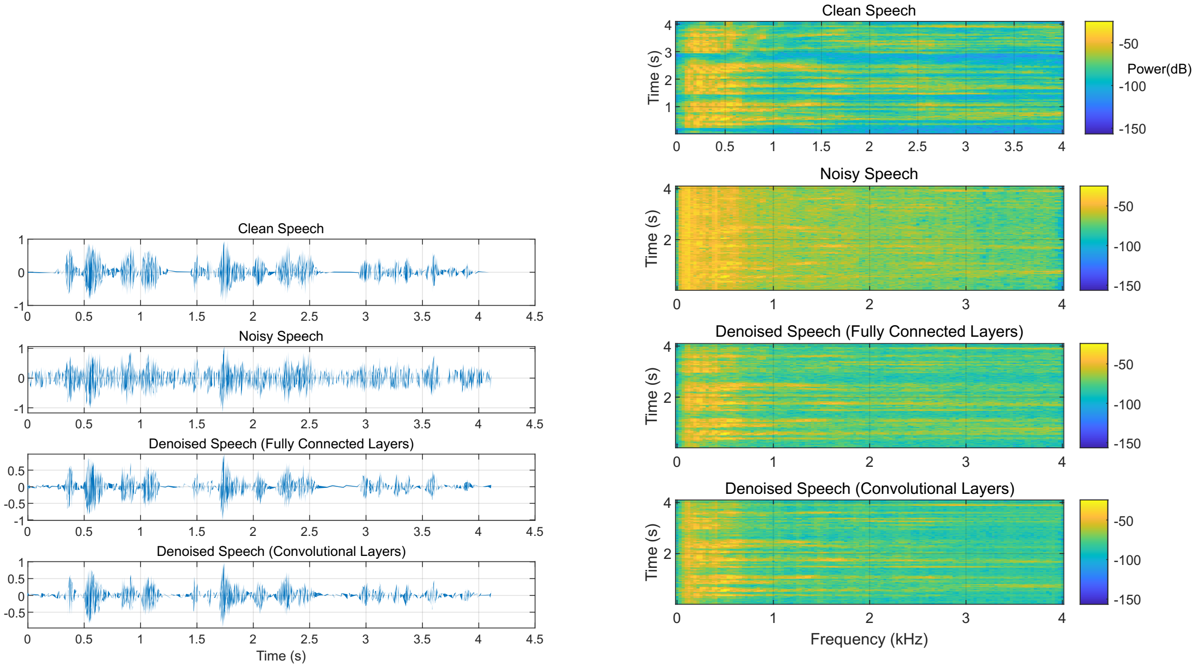
$$\mathrm{RMSE}=\sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}}(y_i\text{-}\hat{y}_i)^2}\ . \tag{10}$$

where n is the number of observations, y_i represents the true value, and $\hat{y}_i$ denotes the predicted value.

3. Results and Discussion

The generalization capability of the Fully Connected Neural Network (FCNN) and Convolutional Neural Network (CNN) models was rigorously evaluated using a set of typical noises not encountered during the training phase. These included WashingMachine noise at 8 kHz, MainStreetOne noise at 16 kHz, a mixture of multipleSounds at 16 kHz, and ChurchImpulseResponse interference at 44 kHz. A comparative analysis of the processing effects of both models on the noisy speech signals was conducted. The Signal-to-Noise Ratio (SNR) and Root Mean Square Error (RMSE) metrics for the denoised signals are summarized in Table 1. Figure 5(a) and (b) present the waveform diagrams and corresponding spectrograms of the original clean speech signal, the speech signal corrupted with 8 kHz WashingMachine noise, and the signals denoised by the FCNN and CNN models, respectively.

Figure 5: Waveform and spectrogram of signals corresponding to 8 kHz WashingMachine noise



(a)

(b)

The experimental results revealed a nuanced performance differential between the FCNN and CNN models across varying noise frequencies. When an 8 kHz noise signal was introduced to the clean speech, the disparity in performance metrics between the two models was minimal. The CNN achieved a marginally higher SNR, exceeding the FCNN by only 0.03, while its RMSE was also slightly higher by 0.016. In contrast, a more substantial performance gap was observed with the introduction of two distinct types of 16 kHz noise. For these scenarios, the CNN's SNR was markedly higher than that of the FCNN by 0.65 and 0.94, respectively; concurrently, its RMSE was also higher by 0.4834 and 0.3143. At the highest tested frequency of 44 kHz, the CNN demonstrated superior performance on both metrics, achieving an optimal SNR of 4.80 (0.04 higher than the FCNN) and a lower RMSE of 2.6826 (0.1556 lower than the FCNN).

Table 1: SNR and RMSE for noisy signals of different frequencies processed by FCNN and CNN

Noise category	Noise frequency	SNR (dB)		RMSE	
	(Hz)	FCNN	CNN	FCNN	CNN
WashingMachine	8k	7.55	7.58	3.0987	3.115
MainStreetOne	16k	4.39	5.04	3.3014	3.7848
multipleSounds	16k	4.75	5.69	3.5111	3.8254
ChurchImpulseResponse	44k	4.76	4.80	2.8382	2.6826

The divergence in model performance can be attributed to their fundamental architectural principles. The minimal difference at 8 kHz suggests that for narrowband, low-frequency noise, the simple denoising mapping learned by the FCNN, despite its structural limitations, can be almost as effective as the CNN's approach. The FCNN's tendency to produce an over-smoothed output may suffice for this less complex task, resulting in comparable point-wise error (RMSE) and perceptual quality (SNR). The FCNN, operating on a flattened input, lacks the spatial awareness to perform precise local operations. Its denoising strategy likely tends towards over-smoothing, producing a spectrogram that is globally similar to the target but lacks sharp details. This smooth output may have a lower overall squared error against the smooth contours of the clean speech but fails to aggressively remove all noise, leaving a diffuse noise floor that harms SNR.

The significantly larger performance gap at 16 kHz underscores the CNN's strength in handling more complex, structured interference. The 16 kHz bandwidth presents a richer, more detailed spectrogram where the CNN's ability to exploit local spatial patterns and hierarchical features becomes critically advantageous. Its convolutional layers can more effectively discriminate between the finer textures of speech and diverse noise types, leading to vastly superior noise suppression (higher SNR). The associated increase in RMSE for

CNN indicates that its more aggressive, localized filtering might introduce specific, concentrated errors not present in the FCNN's globally smoothed output, a trade-off that is acceptable given the substantial gain in perceptual clarity.

The CNN's unequivocal superiority at 44 kHz, excelling in both higher SNR and lower RMSE, represents the culmination of its architectural advantages. The full-bandwidth, high-resolution spectrogram provides a complex data landscape perfectly suited for the CNN's processing paradigm. The network leverages its hierarchical structure to perform precise feature extraction and reconstruction across the entire frequency range, effectively suppressing wideband noise while faithfully preserving the speech signal's details. This results in an output that is both perceptually cleaner and spectrally more accurate, minimizing both residual noise power and point-wise reconstruction error. The FCNN, overwhelmed by the high dimensionality and unable to capture the intricate local correlations, fails to compete on either metric. This result confirms that for high-fidelity, wideband audio processing, the inductive biases embedded in the CNN architecture are not merely beneficial but essential for optimal performance.

The CNN's consistent SNR advantage across all frequencies stems from its inherent architectural inductive biases. The convolutional layers and weight-sharing mechanism excel at identifying and isolating local patterns in the spectrogram domain, enabling exceptional discrimination between structured speech components and stochastic noise patterns. In summary, the observed performance patterns highlight the fundamental operational differences between the two architectures: the FCNN performs conservative, global denoising that minimizes gross point-wise error but yields perceptually noisier outputs (lower SNR), while the CNN implements aggressive, local denoising that is highly effective at improving perceptual metrics (SNR) despite potentially introducing small, specific reconstruction errors in limited-bandwidth scenarios. With sufficient spectral complexity, the CNN's approach becomes unambiguously superior, optimizing both perceptual quality (SNR) and spectral accuracy (RMSE).

References

Aroudi, A., Veisi, H., & Sameti, H. (2015). Hidden markov model-based speech enhancement using multivariate laplace and gaussian distributions. *IET Signal Processing, 9*(2), 177-185. https://doi.org/10.1049/IET-SPR.2014.0032

Azarang, A., & Kehtarnavaz, N. (2020). A review of multi-objective deep learning speech denoising methods. *Speech Communication, 122*, 1-10. https://doi.org/10.1016/J.SPECOM.2020.04.002

Balasubrahmanyam, M., & Valarmathi, R. S. (2024). An intelligent speech enhancement model using enhanced heuristic-based residual convolutional neural network with encoder-decoder architecture. *International Journal of Speech Technology, 27*(3), 637-656. https://doi.org/10.1007/S10772-024-10127-3

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 27*(2), 113-120. https://doi.org/10.1109/TASSP.1979.1163209

Garg, A., & Sahu, O. P. (2022). Deep convolutional neural network-based speech signal enhancement using extensive speech features. *International Journal of Computational Methods, 19*(8), Article 1420056. https://doi.org/10.1142/S0219876221420056

Hao, J., Lee, T. W., & Sejnowski, T. J. (2010). Speech enhancement using Gaussian scale mixture models. *IEEE Transactions on Audio, Speech and Language Processing, 18*(6), 1127-1136. https://doi.org/10.1109/TASL.2009.2030012

Hu, Y., & Loizou, P. C. (2007). Subjective comparison and evaluation of speech enhancement algorithms. *Speech Communication, 49*(7-8), 588-601. https://doi.org/10.1016/J.SPECOM.2006.12.006

Huang, P., & Wu, Y. (2023). Teacher-student training approach using an adaptive gain mask for LSTM-based speech enhancement in the airborne noise environment. *Chinese Journal of Electronics, 32*(4), 882-895. https://doi.org/10.23919/CJE.2022.00.307

Le, X., Lei, T., Chen, K., & Lu, J. (2022). Inference skipping for more efficient real-time speech enhancement with parallel RNNs. *IEEE/ACM Transactions on Audio Speech and Language Processing, 30*, 2411-2421. https://doi.org/10.1109/TASLP.2022.3190738

Lin, Z., Wang, J., Li, R., Shen, F., & Xuan, X. (2025). *PrimeK-net: Multi-scale spectral learning via group prime-kernel convolutional neural networks for single channel speech enhancement* [Paper presentation]. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Hyderabad, India.

Loizou, P. C. (2013). *Speech enhancement: Theory and practice*. CRC Press. https://doi.org/10.1201/B14529

Mai, Y., & Goetze, S. (2025). *MetricGAN+KAN: Kolmogorov-arnold networks in metric-driven speech enhancement systems* [Paper presentation]. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Hyderabad, India.

Nuha, H. H., & Absa, A. A. (2022). *Noise reduction and speech enhancement using wiener filter* [Paper presentation]. 2022 International Conference on Data Science and Its Applications, ICoDSA 2022, Bandung, Indonesia.

Pandey, A., & Wang, D. L. (2022). Self-attending RNN for speech enhancement to improve cross-corpus generalization. *IEEE/ACM Transactions on Audio Speech and Language Processing, 30*, 1374-1385. https://doi.org/10.1109/TASLP.2022.3161143

Pascual, S., Serrà, J., & Bonafonte, A. (2019). Time-domain speech enhancement using generative adversarial networks. *Speech Communication, 114*, 10-21. https://doi.org/10.1016/J.SPECOM.2019.09.001

Reddy, H., Kar, A., & Østergaard, J. (2022). Performance analysis of low complexity fully connected neural networks for monaural speech enhancement. *Applied Acoustics, 190*, Article 108627. https://doi.org/10.1016/J.APACOUST.2022.108627

Saha, B., Khan, S., Shahnaz, C., Fattah, S. A., Islam, M. T., & Khan, A. I. (2018). *Configurable digital hearing aid system with reduction of noise for speech enhancement using spectral subtraction method and frequency dependent amplification* [Paper presentation]. IEEE Region 10 Annual International Conference, Proceedings/TENCON, Jeju, Korea.

Shamsa, A., Ghorshi, S., & Joorabchi, M. (2016). *Noise reduction using multi-channel FIR warped Wiener filter* [Paper presentation]. 13th International Multi-Conference on Systems, Signals and Devices, SSD 2016, Leipzig, Germany.

Soleymanpour, R., Soleymanpour, M., Brammer, A. J., Johnson, M. T., & Kim, I. (2023). Speech enhancement algorithm based on a convolutional neural network reconstruction of the temporal envelope of speech in noisy environments. *IEEE Access, 11*, 5328-5336. https://doi.org/10.1109/ACCESS.2023.3236242

Tang, X., Du, J., Chai, L., Wang, Y., Wang, Q., & Lee, C. H. (2020). *Geometry constrained progressive learning for LSTM-based speech enhancement* [Paper presentation]. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Barcelona, Spain.

Wang, D., & Chen, J. (2018). Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio Speech and Language Processing, 26*(10), 1702-1726. https://doi.org/10.1109/TASLP.2018.2842159

Xiang, Y., Shi, L., Højvang, J. L., Rasmussen, M. H., & Christensen, M. G. (2022). A speech enhancement algorithm based on a non-negative hidden Markov model and Kullback-Leibler divergence. *Eurasip Journal on Audio, Speech, and Music Processing, 2022*(1), Article 22. https://doi.org/10.1186/S13636-022-00256-5

Funding

This research was funded by Hangzhou Dianzi University Information Engineering College Research Cultivation Fund, grant number KYP0323019.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights