

A Comparative Analysis of Various Deep-Learning Models for Noise Suppression

Henil Gajjar¹, Trushti Selarka¹, Absar M. Lakdawala², Dhaval B. Shah², and P.N. Kapil^{3,*}

¹ Department of Electronics and Communication Engineering, Institute of Technology, Nirma University, Ahmedabad, India

² Department of Mechanical Engineering, Institute of Technology, Nirma University, Ahmedabad, India

³ Department of Electrical Engineering, Institute of Technology, Nirma University, Ahmedabad, India

Abstract

Excessive noise in speech communication systems is a major issue affecting various fields, including teleconferencing and hearing aid systems. To tackle this issue, various deep-learning models have been proposed, with autoencoder-based models showing remarkable results. In this paper, we present a comparative analysis of four different deep learning based autoencoder models, namely model ‘alpha’, model ‘beta’, model ‘gamma’, and model ‘delta’ for noise suppression in speech signals. The performance of each model was evaluated using objective metric, mean squared error (MSE). Our experimental results showed that the model ‘alpha’ outperformed the other models, achieving a minimum error of 0.0086 and maximum error of 0.0158. The model ‘gamma’ also performed well, with a minimum error of 0.0169 and maximum error of 0.0216. These findings suggest that the proposed models have great potential for enhancing speech communication systems in various fields.

Keywords: CNN, Noise Suppression, Neural Networks, Audio Processing

Received on 02 October 2023, accepted on 22 November 2023, published on 29 November 2023

Copyright © 2023 H. Gajjar *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetiot.4502

1. Introduction

Exposure to high noise levels can pose a significant threat to human health and well-being. Noise pollution, which is an unwanted sound that has a detrimental effect on human health and the environment, is a growing concern in many cities worldwide. According to the World Health Organization, exposure to noise levels above 70 decibels (dB) for an extended period can cause hearing damage, while exposure to levels above 85 dB for an extended period can result in permanent hearing loss [1]. Furthermore, prolonged exposure to high noise levels can lead to stress, cardiovascular disease, and sleep disturbances. In addition to the health risks, high noise levels can also have adverse effects on productivity and cognitive performance. Studies have shown that exposure to high levels of noise can impair memory, reduce concentration, and interfere with communication. This can lead to decreased productivity, impaired academic

performance, and even accidents in the workplace. Reducing noise levels is, therefore, essential to protect human health and well-being, as well as to ensure optimal cognitive and academic performance. One way to achieve this is through noise cancellation technology [2].

Noise cancellation is a technique that involves reducing or eliminating unwanted sounds by generating sound waves that are the opposite of the unwanted sound. This can be achieved using various methods, including active noise control, passive noise control, and adaptive noise control. Machine learning, specifically deep learning techniques such as Convolutional Neural Networks (CNN), can be used to improve noise cancellation by learning the patterns of the input signals and generating a more accurate and effective cancellation signal (Pandey & Wang, 2019). CNN models are trained using large datasets of sound samples and can learn to recognize and cancel out specific types of noise. By using machine learning algorithms, noise cancellation systems can adapt to different environments and noise sources, making them more versatile and effective. Furthermore, machine learning can also

*Corresponding author. Email: pnkapil@nirmauni.ac.in

improve the speed and efficiency of noise cancellation systems by allowing them to process signals in real-time, making them useful in applications such as noise-cancelling headphones, automotive noise reduction, and environmental noise control.

Convolutional Neural Networks (CNN) have emerged as a powerful tool for pattern recognition in various fields, including speech recognition and image processing. Their ability to reduce the number of parameters in Artificial Neural Networks (ANN) has been a significant factor in their success. As a result, researchers and programmers have been exploring the use of larger models to tackle more complex tasks that were previously deemed impossible using traditional ANNs. One key assumption of CNN is that the problems they handle should not have spatially dependent properties. Additionally, CNN's ability to learn abstract features as the input moves through deeper layers is another critical aspect [3]. For example, in image classification, the first layer may recognize edges, the second layer may identify simpler shapes, and the third layer may recognize higher-level characteristics, such as faces [4].

Autoencoder models are a type of artificial neural network that can learn to compress data into a low-dimensional representation and then reconstruct it back into its original form [5]. Ghosh et al. (2023) embarked on a comprehensive study to assess water quality through predictive machine learning. Their research underscored the potential of machine learning models in effectively assessing and classifying water quality. The dataset used for this purpose included parameters like pH, dissolved oxygen, BOD, and TDS. Among the various models they employed, the Random Forest model emerged as the most accurate, achieving a commendable accuracy rate of 78.96%. In contrast, the SVM model lagged behind, registering the lowest accuracy of 68.29% [17].

Alenezi et al. (2021) developed a novel Convolutional Neural Network (CNN) integrated with a block-greedy algorithm to enhance underwater image dehazing. The method addresses color channel attenuation and optimizes local and global pixel values. By employing a unique Markov random field, the approach refines image edges. Performance evaluations, using metrics like UCIQE and UIQM, demonstrated the superiority of this method over existing techniques, resulting in sharper, clearer, and more colorful underwater images [18]. Sharma et al. (2020) presented a comprehensive study on the impact of COVID-19 on global financial indicators, emphasizing its swift and significant disruption. The research highlighted the massive economic downturn, with global markets losing over US \$6 trillion in a week in February 2020. Their multivariate analysis provided insights into the influence of containment policies on various financial metrics. The study underscores the profound effects of the pandemic on economic activities and the potential of using advanced algorithms for detection and analysis [19].

This process is done by training the model to minimize the difference between the input data and the reconstructed data. Autoencoders can be used for various tasks, including data compression, feature extraction, and denoising. In the context of noise cancellation, autoencoder models are particularly useful. One way to use autoencoders for noise cancellation is

to train them to learn a mapping between a noisy input signal and a clean output signal. During training, the model is fed noisy input data and is trained to generate a clean output that is as close as possible to the original clean signal. Once the model is trained, it can be used to denoise new input data by passing it through the autoencoder and obtaining the reconstructed clean output. The decoder is responsible for generating the reconstructed clean output, and its architecture is designed to extract useful features from the input signal. By training the decoder to reconstruct clean signals from noisy inputs, the model learns to extract features that are robust to noise. This allows the decoder to effectively denoise input signals by removing the noise while retaining the underlying information. Autoencoder models have been shown to be effective in various noise cancellation applications, including speech and image denoising.

This paper compares and showcases four distinct kinds of autoencoder model architectures. Section 2 talks about the research done in the field of noise cancellation and suppression. While Section 3 and Section 4 speak about the methodology and the result and comparison of the proposed models, Section V concludes the paper highlighting all the key inferences from the results.

2. Related Works

Table 1. compares the results from different models. Various models including CNN model, CRN, LSTM, RNN, R-CED, and DeepClean Architecture were implemented, and the performances were analyzed. A common pattern was observed among the research. All the proposed models were a little bit computationally extensive.

Table 1. Model Comparisons from various papers

Reference	Model	Results
[6]	A Custom CNN	The training set accuracy was 95.82% and the corresponding loss was 0.13. The validation set accuracy was 73.59% and the loss was 1.02. Different noise removal methods showed varying performance on different types of noise. The RNN method was found to be more effective in removing stationery and street noise compared to the CNN method, whereas the CNN method performed better in removing noise from music and voiced background.
[7]	Custom CNN and RNN	
[8]	Redundant Convolutional Encoder Decoder (R-CED)	Achieved 8.79 SDR on R-CED on a 20Conv model
[9]	Blind CNN enhanced by an iterative	0.984 accuracy on INRNet and 0.955 on ROLD (both on 10% noise density)

[10]	post-processing A mask-based long short-term memory (LSTM) network	Average STOI value of 0.75 on seen noise types and 0.78 on unseen noise types
[11]	A recurrent neural network with bidirectional long short-term memory (BLSTM)	Compared to the AES+DNN method, the BLSTM approach showed superior performance, exhibiting a 5.4 dB increase in ERLE and a 0.5 improvement in PESQ.
[12]	A custom DeepClean model	The signal-to-noise ratio (SNR) of the injected signal is increased by 21.6% The proposed SAES based on CRN performs better than the conventional Wiener and NLMS algorithms, particularly in low Signal-to-Error Ratio (SER) conditions and high Reverberation Time (RT60) conditions.
[13]	A custom Convolved Recurrent Network (CRN)	The subjective listening tests revealed that the proposed techniques were preferred by the listeners over 60% of the time.
[14]	Three postfiltering strategies based on MMSE noise PSD estimation method were proposed	The subjective listening tests revealed that the proposed techniques were preferred by the listeners over 60% of the time.

assessing their performance on the testing dataset, utilizing appropriate evaluation metrics for noise suppression tasks. The results obtained during evaluation provide insights into the models' capabilities in reducing noise and preserving the clarity of the desired audio signals. Lastly, the best-performing deep-learning model from the comparative analysis is implemented. This implementation stage aims to showcase the practical application of the selected model, potentially serving as a basis for further research and development in noise suppression techniques. Figure 1 shows the flowchart for the algorithm.

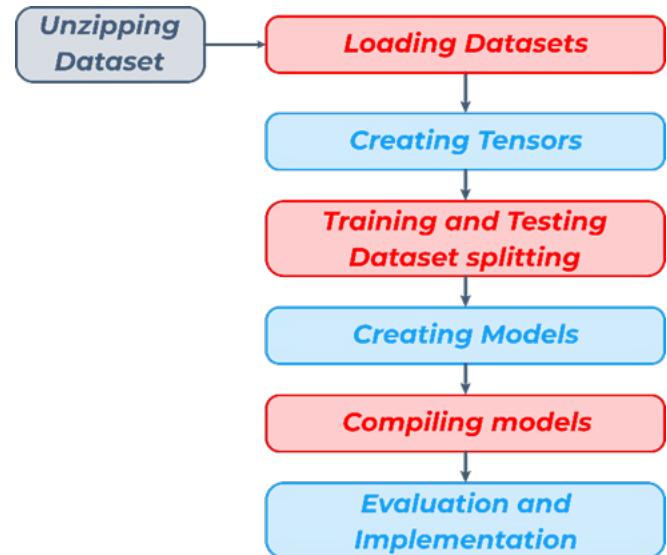


Figure 1. Flowchart for the Algorithm

3. Methodology

The first step of the study involves unzipping the dataset, which contains a diverse collection of noisy sounds. These sounds include speeches embedded with different types of background noises. Additionally, the dataset includes corresponding clean signals, which serve as references for noise-free audio. Upon unzipping, the dataset is loaded into memory for further processing. To facilitate data manipulation and model training, the loaded data set are converted into tensors, which are mathematical representations of the audio signals. This conversion enables efficient computation and manipulation of the data during subsequent stages. Next, the dataset is divided into training and testing subsets. This step ensures that the performance of the deep-learning models can be evaluated on unseen data, thereby providing a realistic assessment of their effectiveness in noise suppression tasks. Then comes emphasis on the creation and compilation of various deep-learning models. These models, specifically designed for noise suppression, employ different architectural configurations and parameters [15]. The purpose of this diversity is to enable a comprehensive comparative analysis, shedding light on the strengths and weaknesses of each model. Once the models are created and compiled, they undergo evaluation. This involves

3.1. Creating and loading the dataset

The dataset utilized in this study comprises a diverse collection of audio recordings. These recordings consist of speeches embedded with various background noises, simulating real-world acoustic environments. Additionally, the dataset includes corresponding clean signals, which serve as references for the noise-free versions of the speeches. Careful curation of the dataset ensures a wide range of noise types, including environmental, mechanical, and human-generated sounds. The collection covers different scenarios and contexts to provide a comprehensive representation of the challenges encountered in noise suppression tasks. The dataset's size and composition helps to train and evaluate autoencoder models effectively while ensuring the privacy and anonymity of the individuals involved in the recordings.

The goal is to train a deep learning model to remove noise from the noisy speech sounds and obtain the clean speech sounds. The data is first read from files using TensorFlow's audio decoding function. The clean sounds and noisy sounds are stored in separate lists. These lists are then concatenated to create two large tensors representing the clean and noisy sounds respectively. Next, the clean and noisy audio files are decoded using the decode_wav function from the TensorFlow library. The decoded clean and noisy audio files are

concatenated into two separate numpy arrays, `clean_sounds_list` and `noisy_sounds_list`. To train the deep learning model, we need to divide the data into training and testing sets. We use 80% of the data for training and 20% for testing [16]. The data is then split into smaller batches, with each batch having a fixed size of 12000 sound samples. This is done to make the training process more efficient and to avoid running out of memory during training.

Next, we created two TensorFlow datasets: one for training and another for testing. The dataset is created using the `tensorflow` function. This function takes the clean and noisy tensors as input and returns a dataset object that can be used for training the model. The dataset is then shuffled and batched, with a batch size of 64. We also drop any incomplete batches at the end of each epoch to ensure that all batches are the same size. Figure 2 and Figure 3 shows a sample clean waveform and a sample noisy waveform.

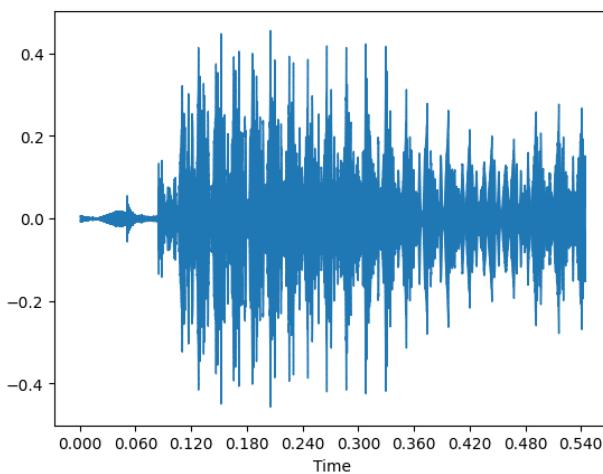


Figure 2. Clean Sound Sample waveform

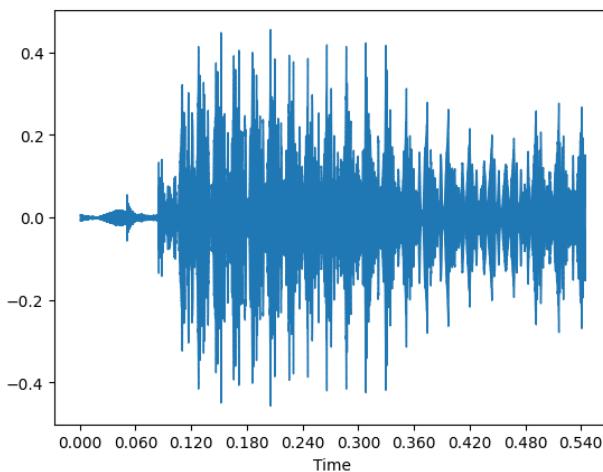


Figure 3. Noisy Sound Sample waveform

The resulting dataset is then used to train a deep learning model to remove noise from the noisy speech sounds. The model architecture used in this code is a convolutional neural

network (CNN) with an encoder-decoder architecture. The encoder part of the model consists of several 1D convolutional layers that extract relevant features from the input sound. The decoder part of the model consists of several 1D transposed convolutional layers that reconstruct the clean sound from the extracted features. The model is trained using the noisy speech sounds as input and the clean speech sounds as output.

3.2. Model Formation

This paper introduces 4 distinct autoencoder architectures for better understanding and comparison of training loss and validation losses. The models are named ‘alpha’, ‘beta’, ‘gamma’, and ‘delta’ for a clear distinction between them. The models are all created using the Keras library in python. We picked the sequential model from the Keras library since our model development was focused on a stack of layers with only one input and output for each layer. The architecture of model ‘alpha’ consists of a series of convolutional layers followed by a series of transpose convolutional layers. The convolutional layers downsample the input data to a lower dimensional feature representation, while the transpose convolutional layers upsample the features to reconstruct the original input data. The network starts with an input layer of size (16000, 1). The first convolutional layer (c1) has 2 filters with a kernel size of 32 and a stride of 2. The activation function used is ReLu. The following convolutional layers (c2, c3, c4, c5) have 4, 8, 16, and 32 filters respectively with the same kernel size and stride as c1 and the ReLu activation function. The first transpose convolutional layer (dc1) has 32 filters, kernel size of 32, and a stride of 1 with padding set to ‘same’. The Concatenate layer merges the output of the dc1 layer with the output of c5. The resulting tensor is passed through the next transpose convolutional layer (dc2) which has 16 filters, kernel size of 32, and a stride of 2. Again, the output is concatenated with the output of the previous convolutional layer (c4) and passed through the next transpose convolutional layer (dc3) with 8 filters, kernel size of 32, and a stride of 2. This process is repeated until the last transpose convolutional layer (dc7), which has 1 filter and a kernel size of 32. Finally, the output of the last concatenation layer is passed through a linear activation function and produces the output of the autoencoder. Figure 4 displays the input shape and the output form following each layer.

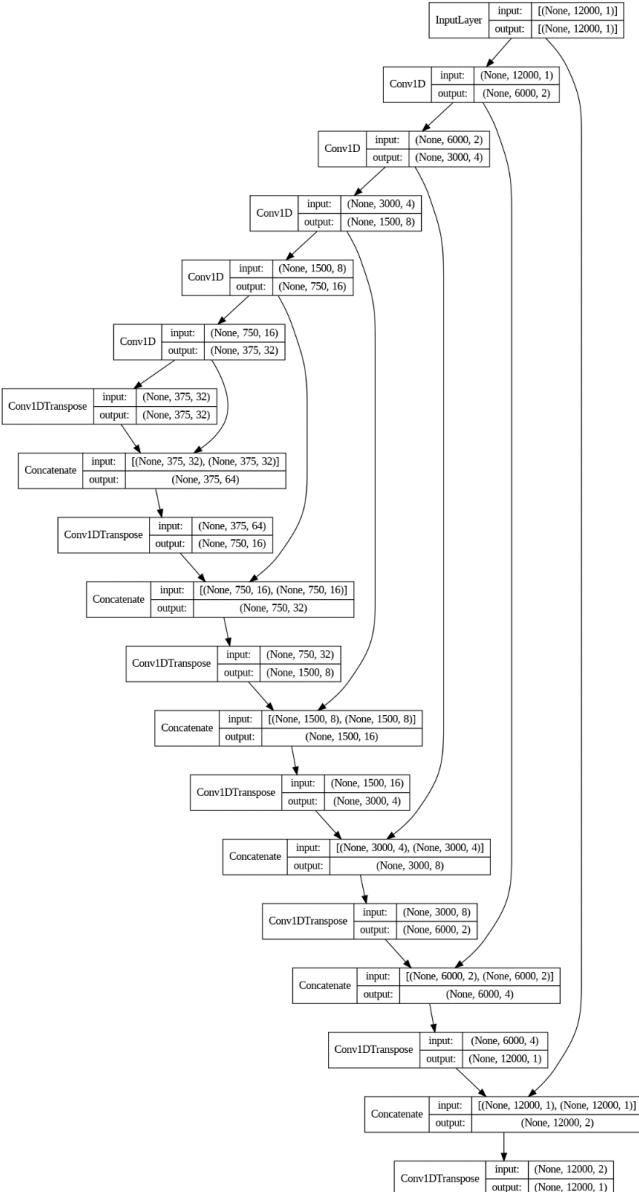


Figure 4. Model ‘Alpha’ Architecture

The second model ‘beta’ is comparatively less complicated than alpha model. The encoder consists of four Conv1D layers with 32, 16, 8, and 4 filters, respectively. Each Conv1D layer has a kernel size of 3 and uses the ReLU activation function. Padding is set to ‘same’ to ensure that the dimensions of the output match the input. The output of the encoder, called encoded, is then passed through the decoder. The decoder also consists of four Conv1D layers with 8, 16, 32, and 1 filters, respectively. Again, each Conv1D layer has a kernel size of 3 and uses the ReLU activation function. Padding is set to ‘same’. The output of the final Conv1D layer in the decoder, called decoded, has a single filter and uses a linear activation function. Figure 5 displays the model beta architecture.

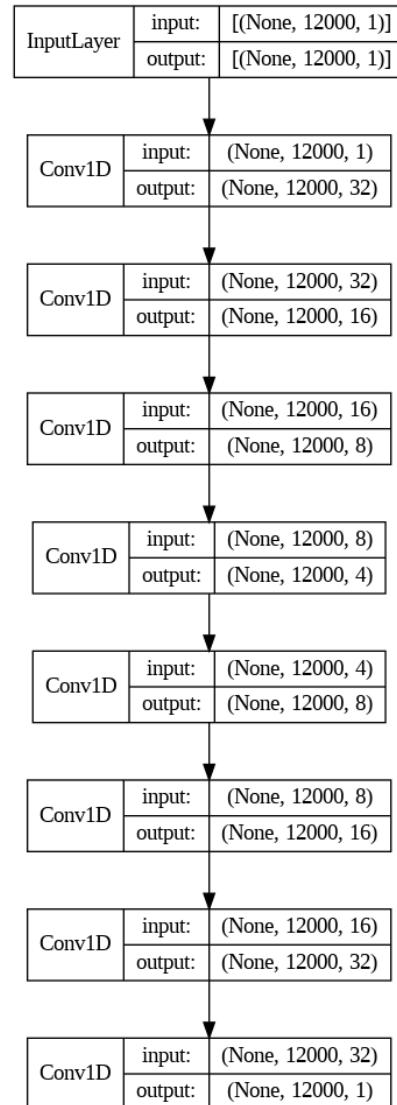


Figure 5. Model ‘Beta’ Architecture

Model ‘gamma’ is similar to model ‘beta’. The encoder consists of two Conv1D layers with 32 and 16 filters, respectively, each with a kernel size of 3 and ReLU activation function. The padding is set to ‘same’. After each Conv1D layer, a MaxPooling1D layer with a pool size of 2 is added, which halves the size of the input along the time axis. The output of the encoder, called encoded, is then passed through the decoder. The decoder consists of two Conv1D layers with 16 and 32 filters, respectively, each with a kernel size of 3 and ReLU activation function. The padding is set to ‘same’. After each Conv1D layer, an UpSampling1D layer with a size of 2 is added, which doubles the size of the input along the time axis. The output of the final Conv1D layer in the decoder, called decoded, has a single filter and uses a linear activation function. Both the models have mean squared error as their loss functions. Adam optimizers are used with a

learning rate of 0.0001 to reduce loss functions. Figure 6 displays the model gamma architecture.

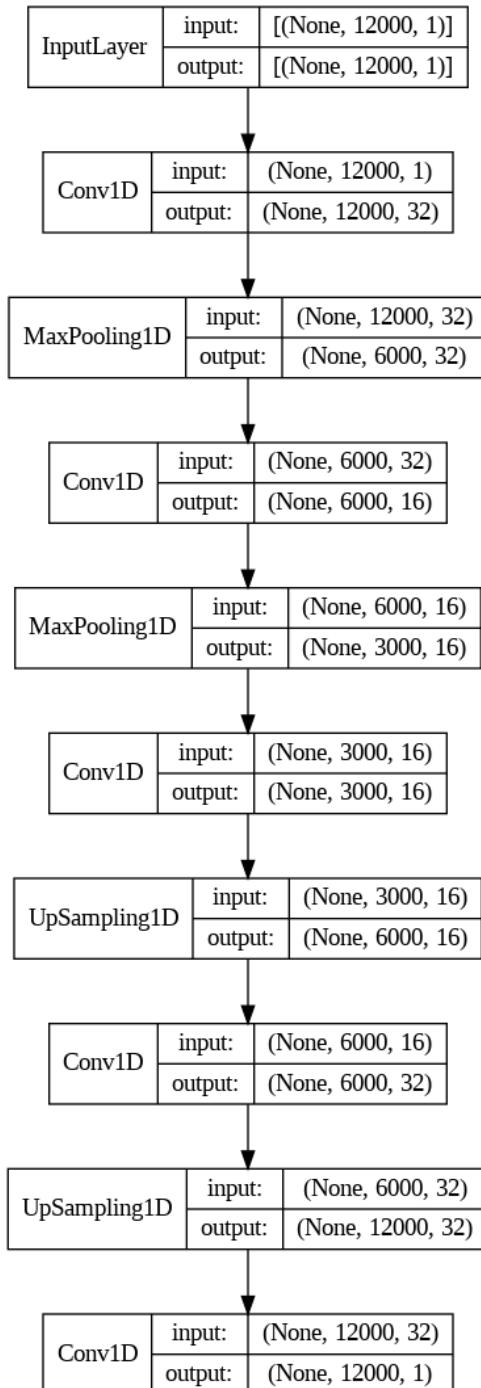


Figure 6. Model 'Gamma' Architecture

Model 'delta' is also similar to model 'gamma' and model 'beta'. The encoder part consists of three 1D convolutional layers. Each layer has 16, 8, and 4 filters respectively. The convolutional layers use the ReLu activation function and

padding is set to 'same' to ensure that the output shape of the layers is the same as the input shape. The decoder part is defined using three 1D transposed convolutional layers. The transposed convolutional layers are used to upsample the encoded input back to its original shape. The layers have 8, 16, and 1 filter respectively, with the same padding and activation function as the encoder. Figure 7 displays the model delta architecture.

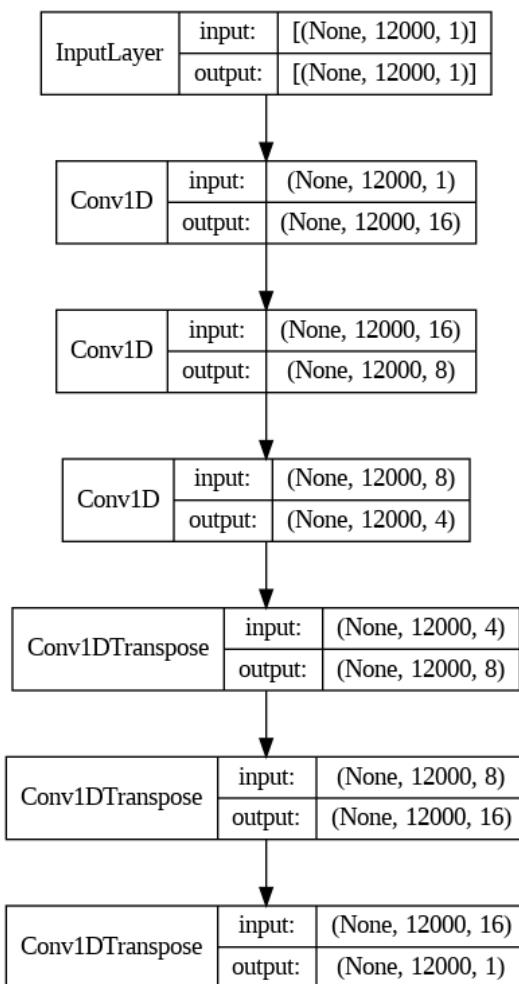


Figure 7. Model 'Delta' Architecture

All the models have mean squared error as their loss functions. Adam optimizers are used with a learning rate of 0.002 to reduce loss functions.

4. Results and Discussions

All the models are tested and validated on a set of parameters for each noise file i.e., batches, steps per epoch, validation steps, and the number of epochs. 6630 samples are used in training and 1658 samples are used for validating the model.

Table 2 depicts the results obtained from all the models comparing their minimum and maximum errors. It is observed that model alpha has the lowest minimum and maximum loss and consistency as compared to other models with model gamma being the second best, followed by model beta and model delta.

Table 2. Model Error Comparison

Model	Min. Error	Max. Error
Model Alpha	0.0086	0.0158
Model Beta	0.0177	0.0229
Model Gamma	0.0169	0.0216
Model Delta	0.0188	0.0230

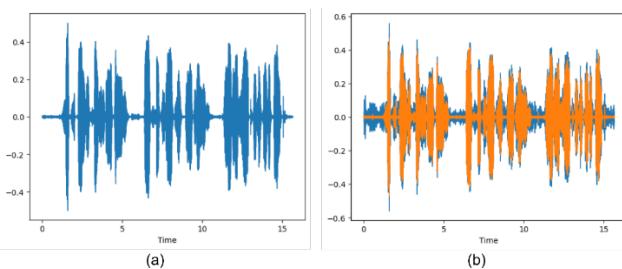


Figure 8. (a) Corresponding Clean Audio Signal (b) Results of Model Alpha

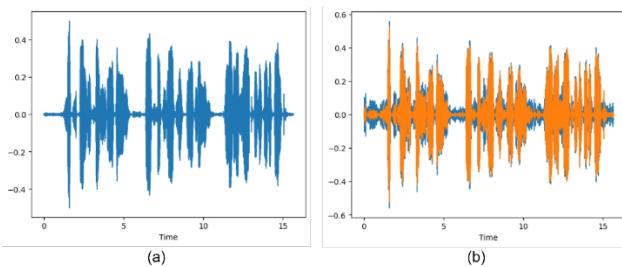


Figure 9. (a) Corresponding Clean Audio Signal (b) Results of Model Beta

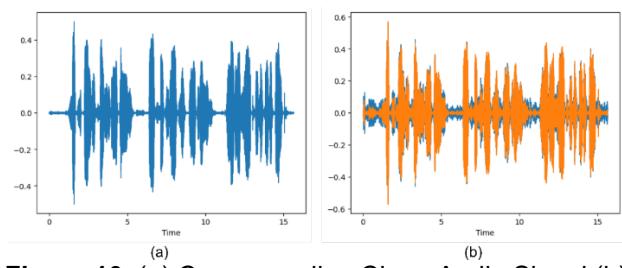


Figure 10. (a) Corresponding Clean Audio Signal (b) Results of Model Gamma

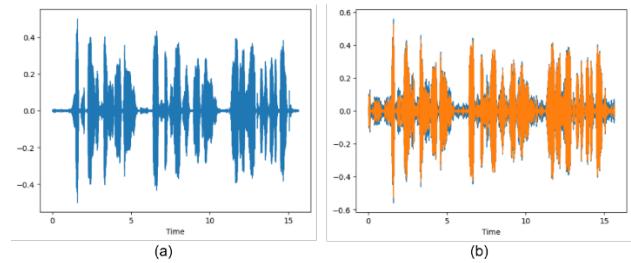


Figure 11. (a) Corresponding Clean Audio Signal (b) Results of Model Delta

Figure 8, Figure 9, Figure 10, and Figure 11 shows the waveform result of various models. It can be seen from these figures that model alpha has the closest results as compared to the other models followed by model gamma, beta and then delta.

5. Future Scope

In considering future research directions for this topic, it is crucial to explore advanced techniques for noise suppression in deep-learning models without triggering detection algorithms. One potential area of investigation involves the development of novel architectures that incorporate attention mechanisms. Attention mechanisms enable the models to focus on relevant audio features while suppressing unwanted noise, leading to enhanced speech quality and improved noise reduction performance. Another promising avenue for future research lies in leveraging generative adversarial networks (GANs) for noise suppression. GANs have shown remarkable potential in generating realistic data, and their application to noise suppression could lead to more robust and accurate models. By training GANs on large-scale datasets containing diverse noise patterns, researchers can aim to achieve superior noise reduction performance and generalize well to unseen noisy environments. Additionally, investigating the effectiveness of transfer learning approaches in noise suppression is another intriguing research direction. By leveraging pre-trained models on related tasks, researchers can explore the transferability of learned features and weights to enhance noise suppression performance. This approach could potentially reduce the amount of labeled data required for training, making noise suppression models more accessible and adaptable to various real-world scenarios. Moreover, exploring the combination of deep-learning models with other signal processing techniques, such as adaptive filtering or spectral enhancement, holds promise for further improving noise suppression performance. Integrating these techniques into the existing deep-learning frameworks could result in more comprehensive and effective noise reduction algorithms.

Furthermore, the investigation of real-time noise suppression systems for practical applications is an important direction for future research. Developing efficient and low-latency deep-learning models capable of processing audio signals in real-time scenarios can have significant implications for industries such as telecommunications, voice

assistants, and audio-conferencing systems. Also, it is crucial to consider the ethical implications and potential bias in noise suppression algorithms. Future research should focus on developing fair and unbiased models that do not inadvertently discriminate against certain speech characteristics or demographic groups. By pursuing these future research directions, the field of deep-learning-based noise suppression can continue to advance, leading to more accurate, robust, and practical solutions for enhancing speech communication systems in various domains.

6. Conclusion

In this paper, the primary aim was to compare and make the best noise suppression algorithm that can help in various modern-day devices. For that, four autoencoder models were built from scratch and were compared on the basis of the fixed set of input parameters. Model ‘alpha’ was an excellent performer with the lowest minimum loss of 0.0086. While model ‘beta’, ‘gamma’, and ‘delta’ obtained minimum error of 0.0177, 0.0169, and 0.0188 respectively. The proposed autoencoder model for noise suppression holds significant promise in effectively handling adaptive noises in a wide range of environments. By leveraging its inherent capacity to learn robust representations of audio signals, the autoencoder can adaptively capture and model the complex dynamics of different noise types. This adaptability enables the model to effectively suppress adaptive noises that exhibit time-varying characteristics or exhibit non-stationary patterns. The autoencoder’s ability to learn latent representations from both noisy and clean signals allows it to identify and extract relevant features that are essential for differentiating between the desired speech and adaptive noises. Consequently, the model can dynamically adjust its noise suppression parameters and adapt its filtering mechanisms to accommodate the changing nature of the noises in real-time. This capability of the autoencoder model makes it a valuable tool in applications where adaptive noises are prevalent, such as communication systems, voice assistants, and audio recording devices. By effectively suppressing adaptive noises, the model enhances the intelligibility and quality of the desired speech signal, improving the overall user experience in various practical scenarios.

Acknowledgements.

The authors acknowledge the financial support provided by Institute of Technology, Nirma University for the publication of the paper.

References

- [1] A. Gupta, A. Gupta, K. Jain, and S. Gupta, “Noise Pollution and Impact on Children Health,” *Indian J. Pediatr.*, vol. 85, no. 4, pp. 300–306, Apr. 2018, doi: 10.1007/s12098-017-2579-7.
- [2] B. R. C. Molesworth, M. Burgess, and D. Kwon, “The use of noise cancelling headphones to improve concurrent task performance in a noisy environment,” *Appl. Acoust.*, vol. 74, no. 1, pp. 110–115, Jan. 2013, doi: 10.1016/j.apacoust.2012.06.015.
- [3] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
- [4] D. Suryani, P. Doetsch, and H. Ney, “On the Benefits of Convolutional Neural Network Combinations in Offline Handwriting Recognition,” in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct. 2016, pp. 193–198. doi: 10.1109/ICFHR.2016.0046.
- [5] W. Wang, Y. Huang, Y. Wang, and L. Wang, “Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 490–497. Accessed: Jun. 01, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2014/W15/html/Wang_Generalized_Autoencoder_A_2014_CVPR_paper.html
- [6] M. Dogra, S. Borwankar, and J. Domala, “Noise Removal from Audio Using CNN and Denoiser,” in *Advances in Speech and Music Technology*, vol. 1320, A. Biswas, E. Wenckes, T.-P. Hong, and A. Wieczorkowska, Eds., in *Advances in Intelligent Systems and Computing*, vol. 1320, , Singapore: Springer Singapore, 2021, pp. 37–48. doi: 10.1007/978-981-33-6881-1_4.
- [7] N. Boyko and A. Hrynyshyn, “Using Recurrent Neural Network to Noise Absorption from Audio Files”.
- [8] S. R. Park and J. Lee, “A Fully Convolutional Neural Network for Speech Enhancement.” arXiv, Sep. 22, 2016. Accessed: Apr. 30, 2023. [Online]. Available: <http://arxiv.org/abs/1609.07132>
- [9] S. Sadrizadeh, H. Otroshi-Shahreza, and F. Marvasti, “Impulsive noise removal via a blind CNN enhanced by an iterative post-processing,” *Signal Process.*, vol. 192, p. 108378, Mar. 2022, doi: 10.1016/j.sigpro.2021.108378.
- [10] M. Strake, B. Defraene, K. Fluyt, W. Tirry, and T. Fingscheidt, “Speech enhancement by LSTM-based noise suppression followed by CNN-based speech restoration,” *EURASIP J. Adv. Signal Process.*, vol. 2020, no. 1, p. 49, Dec. 2020, doi: 10.1186/s13634-020-00707-1.
- [11] H. Zhang and D. Wang, “Deep Learning for Acoustic Echo Cancellation in Noisy and Double-Talk Scenarios,” in *Interspeech 2018*, ISCA, Sep. 2018, pp. 3239–3243. doi: 10.21437/Interspeech.2018-1484.
- [12] R. Ormiston, T. Nguyen, M. Coughlin, R. X. Adhikari, and E. Katsavounidis, “Noise reduction in gravitational-wave data via deep learning,” *Phys. Rev. Res.*, vol. 2, no. 3, p. 033066, Jul. 2020, doi: 10.1103/PhysRevResearch.2.033066.
- [13] L. Cheng, R. Peng, A. Li, C. Zheng, and X. Li, “Deep learning-based stereophonic acoustic echo suppression without decorrelation,” *J. Acoust. Soc. Am.*, vol. 150, no. 2, pp. 816–829, Aug. 2021, doi: 10.1121/10.0005757.
- [14] Y. Ke, A. Li, C. Zheng, R. Peng, and X. Li, “Low-complexity artificial noise suppression methods for deep learning-based speech enhancement algorithms,” *EURASIP J. Audio Speech Music Process.*, vol. 2021, no. 1, p. 17, Apr. 2021, doi: 10.1186/s13636-021-00204-9.
- [15] W.-H. Lee, M. Ozger, U. Challita, and K. W. Sung, “Noise Learning-Based Denoising Autoencoder,” *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 2983–2987, Sep. 2021, doi: 10.1109/LCOMM.2021.3091800.

- [16] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A critical look at the current train/test split in machine learning." arXiv, Jun. 08, 2021. doi: 10.48550/arXiv.2106.04525.
- [17] Ghosh, H., Tusher, M.A., Rahat, I.S., Khasim, S., Mohanty, S.N. (2023). Water Quality Assessment Through Predictive Machine Learning. In: Intelligent Computing and Networking. IC-ICN 2023. Lecture Notes in Networks and Systems, vol 699. Springer, Singapore. https://doi.org/10.1007/978-981-99-3177-4_6
- [18] Alenezi, F.; Armghan, A.; Mohanty, S.N.; Jhaveri, R.H.; Tiwari, P. Block-Greedy and CNN Based Underwater Image Dehazing for Novel Depth Estimation and Optimal Ambient Light. *Water* 2021, 13, 3470. <https://doi.org/10.3390/w13233470>
- [19] G. P. Rout and S. N. Mohanty, "A Hybrid Approach for Network Intrusion Detection," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 614-617, doi: 10.1109/CSNT.2015.76.