



Speech Enhancement Using Machine Learning

Dr Manisha Mali, Shreyas Thombal, Akshay Gangurde, Sunil Sonu, Jahnvi More
Department of Computer , Vishwakarma Institute of Information Technology , Pune, India

Abstract— Speech enhancement, a crucial part of speech processing, reduces noise, reverberation, and distortions to increase the comprehensibility and clarity of voice signals. Although they have set the foundation, conventional methods such as Wiener filtering and spectral subtraction are frequently constrained in intricate and cacophonous settings. Recent developments in machine learning, especially deep learning, have completely changed this sector by providing more resilient, adaptable models that can deal with a wide range of noise circumstances. This review study looks at different machine learning-based methods for improving speech, with a particular emphasis on neural networks including recurrent and convolutional neural networks (RNNs). Their structures, optimization strategies, and superior performance over conventional approaches are all covered in this study. It also tackles the difficulties of computational efficiency on devices with limited resources, model complexity, and real-time processing.

The study also suggests future lines of inquiry for integrating reinforcement learning, unsupervised learning, and hybrid models to improve performance even more in demanding settings.

Keywords— Recurrent Neural Networks (RNNs), Deep Learning , Noise Reduction , Real-Time Processing , Resource-Constrained Devices

1 . INTRODUCTION

Speech enhancement is one of the many disciplines where revolutionary advancements have been made possible by the rapid growth of artificial intelligence, especially machine learning. Speech enhancement has attracted a lot of attention because of its applications in voice-activated devices, hearing aids, telecommunications, and more. Its goal is to increase the quality and intelligibility of speech signals in loud situations. Statistical models and signal processing techniques were the foundation of conventional voice enhancement approaches [1]. But with the development of machine learning, especially deep learning and Recurrent Neural Networks (RNNs), the way voice augmentation is done has changed dramatically.

Because machine learning models, and RNNs in particular, can retain context across time, they are particularly good at processing sequential input, like

speech signals. This makes them ideal for tasks like noise reduction and enhancing voice intelligibility in real-time applications because they can handle the temporal dependencies found in speech. The use of RNNs for speech augmentation on tiny, resource-constrained devices, such as smartphones and hearing aids, poses a number of difficulties despite its potential. Because of their low memory and processing capability, these devices have trouble efficiently running complicated machine learning models.

The adaption and optimization of RNNs for small, resource-constrained embedded platforms is the main issue this study attempts to solve. It entails specifically figuring out how to make RNNs work well within the constraints of the memory and processing power that these devices naturally have. This includes creating techniques and algorithms that reduce memory usage and computational burden without compromising the precision and performance of RNN-based applications.

The ultimate objective is to maximize the potential of

intelligent applications running on small devices, making sure they function smoothly and effectively, and improving the usefulness and usability of these devices across a range of industries, from consumer electronics to healthcare.

This research paper walks us through following topics:

1. Exploration of Embedded RNN Implementation
2. Identification of Objectives and Challenges
3. Comprehensive Analysis of RNN Components
4. Optimization Techniques
5. Highlighting Trade-offs
6. Encouraging Future Research

Objectives of Research paper

1. Extensive Review: The work offers an extensive and in-depth analysis of previous attempts to apply RNN models on tiny, embedded devices. It is an invaluable resource for anyone interested in this area by bringing together the combined knowledge and experiences of academics and developers in this field.

2. Optimization Strategies: It looks at the several approaches and strategies for optimization that have been used to get RNNs to function well on platforms with limited resources. Model compression, quantization, and algorithmic optimizations are a few examples of these tactics, which might be crucial for obtaining good performance with less resources.

3. Benchmarking and Evaluation: Assessing the present status of RNN implementations on embedded platforms is one of the main goals. This entails evaluating their functionality, memory effectiveness, and real-time skills. The goal of this is to give the paper a starting point for evaluating further advancements.

II. LITERATURE REVIEW

Literature study on research papers that improve speech. In audio signal processing, speech enhancement is an essential field that seeks to increase the fidelity and comprehensibility of spoken signals in noisy settings. Applications for it include voice recognition software, hearing aids, and telecommunications. An overview of various popular methods and pertinent research publications is provided below: Ephraim and companions.

[1], They employed Wiener filter-based methods in these. The Wiener filter is a basic method for improving speech. By reducing the mean square error between the clean and noisy signals, it estimates the clean speech signal. Boll and associates.

[2], whereby they applied the spectral subtraction technique. It calculates the noise spectrum from the signal's quiet portions and deducts it from the noisy.

[3] In these NMF, or nonnegative matrix factorization, is applied. Using NMF-based methods, a signal's spectrogram is broken down into its basic spectra and activation coefficients. NMF has been used for speech augmentation and source separation. Luo and others. These deep learning techniques are used in

[4]. It offers a thorough synopsis. especially convolutional neural networks (CNNs) and deep neural networks (DNNs), have demonstrated impressive performance in speech augmentation. Pascual and associates.

[5] GANs, or Generative Adversarial Networks. GANs have been applied to voice augmentation, where a discriminator network discerns between real and created clean speech, while a generator network learns to map noisy speech to clean speech. Mesgarani et al.

[6] RNNs, or Recurrent Neural Networks. Speech improvement projects have also used recurrent models such as GRU and LSTM. Srinivas Deepa et al.

[7] A novel approach that subtracts the real and imaginary components independently was proposed by Yi Zhang and Yunxin Zhao. As a result, each component—real and imagined—is processed and improved independently. The reestablished signal performs better. Background noise in the real environment unavoidably affects voice signals, which reduces the clarity of the speech. Speech improvement algorithms reduce background noise in noisy signals to produce clean speech from distorted speech [11].

The research community in speech and audio signal processing has been motivated to switch from standard voice enhancement algorithms [12, 13] to deep neural networks (DNN)-based algorithms [14] because to the positive outcomes of deep learning leads.

These include generative techniques like variation auto-encoder and generative adversarial networks (GANs) [7], as well as discriminative techniques like convolutional neural networks (CNNs) [15, 16], long short-term memory (LSTM), and recurrent neural networks (RNNs) [17].

For speech improvement tasks, GAN-based approaches have proven to be more successful [10] and robust against various noise kinds. In contrast to their discriminating peers. Following Pascual et al.'s groundbreaking work on speech enhancement GAN (SEGAN) [16], there have been a number of advancements made to GAN-based speech enhancement techniques.

III. PROPOSED METHODOLOGY

In order to overcome computational difficulties and neural network problems including vanishing gradient and data processing efficiency, this research focuses on optimizing recurrent neural networks (RNNs) for speech enhancement applications. We provide an extensive examination of RNN components for speech improvement and a methodology that makes use of different processing strategies based on the size of the audio files being processed. For larger files, group processing is suggested, whereas multicore processing is suggested for smaller datasets. We describe the implementation process in depth in this part, emphasizing the mathematical formulations required for network optimization and signal processing.

Dual Processing and Compilation Errors

To handle huge datasets and real-time applications, neural networks developed for voice augmentation frequently need dual processing. If the processes execute simultaneously without synchronization, data corruption may ensue, which could result in compilation issues. In order to solve this problem, we advise utilizing two different processors and making sure that the start and end times of the data are standardized. Three methods are suggested to further handle this:

1. Symmetric Multiprocessing

When using symmetric multiprocessing (SMP), different processors are given an equal quantity of data to process at the same time and share the same memory. This equitable distribution of tasks guarantees effective processing free from bottlenecks.

2. Heterogeneous Computing:

The processing load can be spread according to the capabilities of each type of processor by mixing several processor types, such as CPUs and GPUs. By utilizing GPUs for matrix multiplications in RNNs and CPUs for sequential workloads, this method guarantees a better balanced calculation.

The performance of several computing techniques will be compared in this study with an emphasis on: Processing in bulk (for big audio files) Processing on several cores (for small audio files) Processing on a single core (as a baseline)

Comparison of Processing Techniques

The performance of several computing techniques will be compared in this study with an emphasis on:

- Bunch Processing (for large audio files)
- Multicore Processing (for small audio files)
- Single-core Processing (as a baseline)

A few benefits of bunch computing are its high performance, fault tolerance, scalability, and affordability. It is an excellent choice for managing voice improvement jobs on big datasets because of these features. In order to identify the most efficient strategy, the performance comparison will also take processing time and computing cost into account.

Acoustic Signal Processing

After the audio signal is captured and digitalized, speech enhancement begins. The analog signal is recorded by a microphone and subsequently sampled at a rate determined by the Nyquist Theorem:

$$f_s \geq 2f_{\max}$$

Where f_s is the sampling rate and $f_{\max}$ is the maximum frequency of the signal. According to this theorem, the sampling rate must be at least twice the maximum frequency to avoid **aliasing**, which leads to distortion and loss of information. Aliasing occurs when the sampling rate f_{s_fs} is insufficient:

$$fa = |fs - fn|$$

where f_a represents the aliasing frequency and f_n is the signal's frequency. Ensuring that the sampling rate meets the Nyquist criterion is essential for accurate speech recognition and enhancement.

Signal Filtering

A technique for signal processing called signal filtering takes out undesired elements from a signal while keeping the information that is sought.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$$

where N is the total number of samples, k is the frequency index, $x(n)$ is the time-domain signal, and $X(k)$ is the frequency domain representation. The clarity of the speech signal is improved by removing the extraneous components.

RNN Optimization and Activation Functions

Tanh and sigmoid activation functions are commonly used in classical speech enhancement models, yet they provide difficulties because of the vanishing gradient problem. These are nonlinear activation functions.

Tanh Function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Sigmoid Function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Both functions provide diminishing gradients as the input value increases away from zero, which makes it challenging for the neural network to learn efficiently even though they are helpful for capturing nonlinear relationships. We suggest using unidirectional RNNs to address this, as they are easier to train, require less data, and do not have many of the complexity that come with bidirectional RNNs or traditional speech improvement models.

RNN Activation Function for Speech Enhancement

In the case of unidirectional RNNs, the hidden states are updated based on the input, previous hidden state, and weights:

Forward Hidden State Update:

$$h_t^{(f)} = \tanh(W_{xh}x_t + W_{hh}h_{t-1}^{(f)} + b_h^{(f)})$$

Backward Hidden State Update:

$$h_t^{(b)} = \tanh(W_{xh}x_t + W_{hh}h_{t+1}^{(b)} + b_h^{(b)})$$

Final Output:

$$y_t = W_{hy}h_t + b_y$$

The suggested methodology optimizes RNNs for voice augmentation while integrating multiple processing techniques. Through the use of multicore and bunch processing, this research tackles issues like the vanishing gradient problem and offers a comprehensive method for enhancing RNN performance in speech processing systems. In order to improve speech quality and processing speed, future research will examine different architectures and additional activation function improvements.

IV. TECHNOLOGY INVOLVED IN PROPOSED METHODOLOGY

Achieving a precise lung cancer prediction is the main goal of the proposed system. It is anticipated that this proposed design will accurately forecast the lung cancer accuracy score. A thorough explanation of the step-by-step process will be given in the section that follows.

4.1. Dataset collection

Data Processing

There are two types of data processing methods for speech enhancement with RNNs.

Noise estimation:

In order to do this, the speech signal must be used to estimate the noise signal. For this, a statistical model of the noise signal is typically used.

Speech enhancement:-

This is about using the estimated noise to improve the speech signal quality. For this task, a variety of methods can be used, including subtraction, Wiener filtering, and deep learning. Here are a few data processing methods for RNN-based speech improvement.

Preprocessing

Frame Segmentation: To produce sequences of frames, the continuous voice signals are split into brief, overlapping frames (20–30 ms). Frames that overlap aid in capturing context.

1) **Feature extraction:** Mel coefficients (MFCCs), log mel spectrograms, or spectrogram frames are among the pertinent auditory features that are recovered from every frame. The RNN uses these features as input.

2) **Optional Data Augmentation:** You can use data augmentation strategies to broaden the diversity of training data. Adding noise, adjusting pitch, modifying tempo, or incorporating room impulse reactions into speech are examples of common augmentations.

3) **Label Creation:** Labeling your data is a necessary step in learning. Typically, these names stand for the clean speech frames that each frame is associated with.

4) **Sequence Padding:** In order to maintain consistency, it may be required to modify the length of frames by either padding them with zeros or truncating them. This is because RNNs, or recurrent neural networks, require input sequences of a given length. Using zeros as padding is one method.

5) **Data Splitting:** Divide your dataset into separate portions that include test, validation, and training sets. Typically, partition ratios would be 10% for validation, 10% for training, and 10% more for testing.

6) **Standardization:** To get a mean of 0 and a variation of 1, standardize your input features. This makes it possible for the model's training to proceed with faster convergence.

7) **Architectural Blueprint:** Create an RNN-based structural blueprint for your speech improvement model. Typical design factors include:

Recurrent Layers: To capture the inherent temporal dependencies in the voice data, use one or more strata of RNN cells, such as LSTM or GRU.

Encoder-Decoder: Use an encoder-decoder system in which the encoder extracts characteristics from loud speech and the decoder creates enriched features.

Bidirectional RNNs: Use bidirectional RNNs [15], which go through sequences in both directions and

assimilate context from previous and subsequent frames. **Attention Mechanisms:** Integrate attention mechanisms into the architecture of the model to enable targeted examination of particular segments of the input sequence.

Deep stacking: To increase model capacity and representation, layer several levels of RNN cells.

Training: - Use the assigned training dataset to carry out RNN training. Remain aware of any loss that is seen in the validation set in order to prevent overfitting. Think about implementing strategies such as early halting in order to prevent overfitting.

Hyperparameter Tuning:- Engage in the exploration of diverse hyperparameters (e.g., learning rate, batch size, model architecture) to fine-tune the model's performance.

1) **Training:** Using the designated training dataset, conduct RNN training. To prevent any overfitting, keep a close check on the loss experienced in the validation set. In this context, deliberate application of techniques like early halting can be helpful.

2) **Evaluation:** - When applying the trained model to the test dataset, evaluate it based on relevant evaluation criteria. These could include measurements like mean opinion score (MOS), perceptual assessment of speech quality (PESQ), or signal-to-noise ratio (SNR) augmentation.

3) **Real-time Application:** To provide improved voice output, use the trained model to process noisy speech inputs in real-time or in batches.

4) **Optional Post-processing:** Use optional post-processing techniques to boost the speech's quality even more. These methods could include, for example, dereverberation or dynamic range compression.

5) **Deployment Phase:** Insert the trained model into the intended application, such as improving speech recognition systems, reducing background noise in voice communications, or any other use cases that make sense.

4.3 RNN Model

The significance of social media platforms for identifying different themes, conducting sentiment analysis, and gauging public opinion has increased. This study proposes a method for tweet classification using TensorFlow and Keras. It makes use of a Recurrent Neural Network (RNN) architecture, with an emphasis on Long Short-Term Memory (LSTM) units in particular [14].

This technique improves the representation of tweets by combining word embeddings with additional information. With this improvement, the model can now accurately classify tweets into designated groups and capture contextual relationships. Our RNN-LSTM model has proven to perform better than baseline techniques after extensive testing and evaluation, confirming its applicability for tweet categorization jobs. We use extra characteristics like emotion ratings and hashtags together with pre-trained word embeddings to improve the model's comprehension of tweet content.

With the use of this method, deep learning approaches in tweet categorization can be effectively utilized for applications such as sentiment analysis, subject detection, and opinion mining. By illuminating the possibilities of RNN-LSTM models in the context of understanding and interpreting social media data, this research advances the discipline.

The results highlight how important it is to take temporal dynamics and contextual information into account while working on tweet classification tasks. Prospective research avenues could encompass investigating substitute pre-trained embeddings, investigating sophisticated RNN structures, and tackling issues associated with noisy and skewed Twitter data. All things considered, this study makes it easier to analyze and comprehend the vast amount of data that is posted on social media sites like Twitter. One kind of recurrent neural network (RNN) that analyzes input sequences bidirectionally—from start to finish and from end to start—is the bidirectional recurrent neural network (BiRNN). The network can efficiently learn temporal relationships within the input sequence thanks to its bidirectional processing. Because BiRNNs can capture such temporal relationships more effectively than unidirectional RNNs and are applicable to a wider range of tasks. They also exhibit greater resilience to noise compared to unidirectional RNNs.

To sum up, DNNs work better at tasks requiring fixed-size input data, like tabular or picture data, but BRNNs are superior at modeling sequential data. Depending on the particular application and network architecture, several performance evaluation criteria are chosen.

Speech enhancement makes use of a variety of estimating approaches, some of the most popular ones being:

Spectral Subtraction:

Even though it's simple, spectral subtraction is nevertheless a powerful method. The way it works is that the noisy speech spectrum is subtracted from the estimated noise spectrum. Using a sliding window technique, the noise spectrum is calculated from an independent noise sample. The clear speech spectrum is approximated by this subtraction.

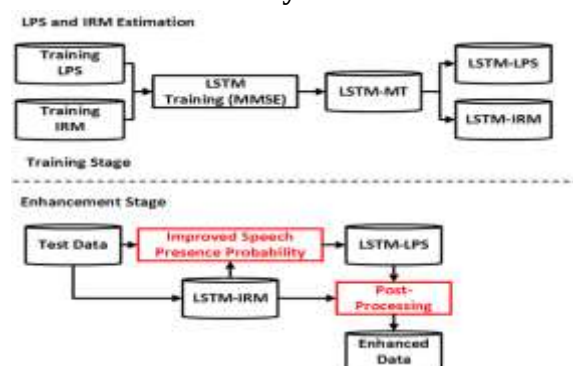
Wiener Filtering:

An much more sophisticated technique, Wiener filtering, uses a statistical model that includes both speech and noise data. This model often makes the assumption that noise and speech are uncorrelated. As a linear filter, the Wiener filter reduces the mean square error between the real and estimated speech signals.

Minimum Mean Square Error (MMSE) Estimation:

Correlation between speech and noise signals is accommodated by MMSE estimation, which extends Wiener filtering. In contrast to the Wiener filter, the MMSE estimator aims to minimize the mean square error between the estimated and genuine speech, enabling more complex modeling. In certain cases, this results in better performance.

Fig 3.1 Block Diagram on Application of Long Short Term Memory.



A speech enhancement system utilizing Long Short-Term Memory (LSTM) models is depicted in this block diagram. During the training phase, an LSTM model for feature estimation is trained using LPS (Log Power Spectrum) and IRM (Ideal Ratio Mask) data. Test data is run through LSTM models to estimate LPS and IRM during the enhancement step. Post-processing is then used to improve speech presence probability, resulting in enhanced data.

Deep Learning:

The usefulness of deep learning for speech improvement has been demonstrated by recent developments. Neural networks are often used in these techniques to determine the statistical relationships between clean and noisy speech.

Although deep learning can produce cutting-edge results, its training frequently involves significant computational demands. Moreover, the quality of the noise estimate has a major impact on how well the selected estimation technique performs. Attaining optimal performance requires a precise estimation of the noise.

Different mathematical models are used in voice enhancement algorithms; the specifics change depending on the algorithm. Still, a few of widely used mathematical models include:

Speech and Noise Models:

These models serve to depict the statistical attributes characterizing speech and noise signals. Typically, the speech model assumes non-stationary properties with a high-pitched spectrum, while the noise model presupposes a stationary nature with a flat spectrum.

Estimation Algorithms:

When estimating clear voice signals from noisy equivalents, these algorithms are used. These methods aim to minimize the mean square error between the genuine speech signal and the estimated speech signal by utilizing statistical models of speech and noise signals. One of the most important tools for describing the statistical characteristics found in speech and noise signals is the Speech and Noise Model Algorithm, which is basically a statistical model.

While the noise model usually assumes stationary characteristics with a flat spectrum, the speech model commonly depends on the assumption of non-stationary features with a high-pitched spectrum. The speech and noise model is applied to the task of

estimating clear speech signals from noisy variations. The estimating procedure is based on statistical speech models.

The stepwise progression within the Speech and Noise Model Algorithm comprises:

Noise Estimation:

During this stage, the noise signals are estimated using independent noise samples that were recorded apart from the speech signal.

Speech and Noise Modeling:

Statistical models are used to model noise and voice signals, providing a complete representation.

Estimation of Clean Speech Signal:

By using statistical models that regulate speech and noise signals, the program ultimately estimates the clean speech signal from the noisy speech signal.

There are other ways to apply the speech and noise model, such as:

Parametric methods:-

These methods use a parametric model of the speech and noise signals. The training data are estimated from this model.

Non-parametric methods:

A parametric model of the speech and noise signals is not used in these techniques. Rather, they estimate the statistical characteristics of the signals using a non-parametric method.

The features and particular application of the noise will determine which method is chosen. The speech and noise model algorithm's output is commonly assessed by objective metrics like the signal-to-noise ratio (SNR) and the perceptual evaluation of speech quality (PESQ). These are a few of the most often used models for speech and noise:

Gaussian model:

This is a simple but effective model that assumes that the speech & noise signals are both Gaussian distributed.

Non-Gaussian model:

This model relaxes the assumption of Gaussian distribution & allows for more flexibility in modeling the speech & noise signals.

Hidden Markov model (HMM):

This is a more modern model that can capture the temporal dependencies in the speech & noise signals.

Deep learning model:

This is a more recent innovation that has demonstrated successful speech and noise modeling. Without making any assumptions about the distribution of the signals, deep learning algorithms are able to determine the statistical relationship between speech and noise signals. The exact application and the noise's properties will determine which speech and noise model is best.

For example, a Gaussian model can be sufficient if the noise is steady. In case the noise is not consistent, a non-Gaussian model can be required. The quantity of training data that is available will also affect the speech and noise model's performance. Better performance is generally correlated with more training data.

Fig 5.2 Graph of Comparision between LSTM , DTW and CNN on Memory Usage.

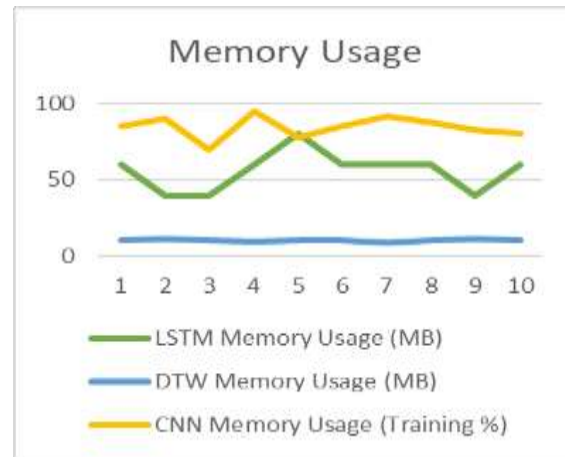
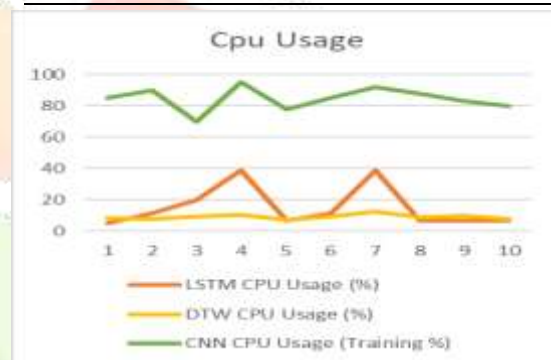


Fig 5.3 Graph of Comparision between LSTM , DTW CNN on Memory Usage



V. RESULT AND DISCUSSION

Fig 5.1 Graph of Comparision between LSTM , DTW and CNN on Accuracy.

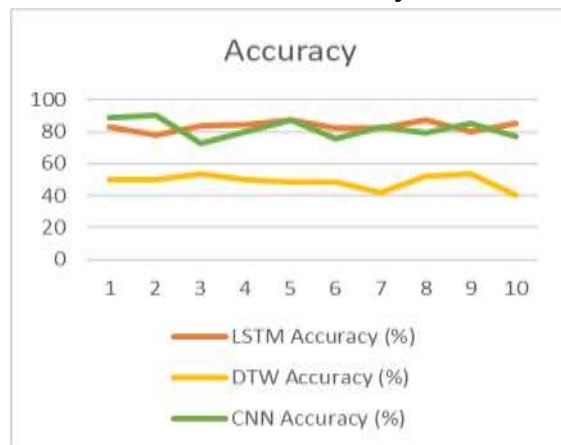


Fig 5.4 Table of Comparision between LSTM , DTW , CNN on Execution Time , Memory Usage , CPU Usage , Accuracy

Method	Accuracy (%)	CPU Usage (%)	Memory Usage (MB)	Execution Time (seconds)
LSTM	47.73	1.82	83.07	15.49
DTW	10.23	0.36	0.03	10.24
CNN	82	4.24	65.84	5.6