|| JAI SRI GURUDEV ||

Sri Adichunchanagiri Shikshana Trust®

SJB INSTITUTE OF TECHNOLOGY

AN AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Department of Computer Science and Engineering

Project Presentation On "Advanced Audio Enhancement System" (AAES)"

Presented By:

Sameer Kumar Chaudhary [1JB22CS133]
Sneha Mal [1JB22CS152]
Sristi Priya [1JB22CS160]
Surath Chowdhury [1JB22CS167]

Under the guidance of

Dr. Roopa M J
Assoc Professor,
Dept. Of CSE, SJBIT

INTRODUCTION

Advanced Audio Enhancement Systems use modern signal processing and AI-driven techniques to improve the clarity, quality, and intelligibility of audio. These systems reduce noise, remove distortions, enhance speech, and optimize sound for various real-world environments, enabling better user experiences in communication, media, assistive technologies, and professional audio applications.

LITERATURE SURVEY

Author's	Year	Title	Methodology	Pros/Cons
Thomas Arvanitidis	2024	Spectral Modelling for Transformation and Separation of Audio Signals	Introduces a phase-stability–based modification of STFT using an ensemble of estimates to improve spectral reliability and enhance model-based audio source separation.	Prons:Better discrimination of structured vs. artefact components, improved spectral peak picking, enhanced separation performance. Cons:Higher computational cost, sensitive to parameter settings, requires further validation on broader datasets.
Yang Liu, Mingli Xiao, Yong Tie	2024	A Noise Reduction Method Based on LMS Adaptive Filter of Audio Signals	This paper presents a noise-reduction method that uses LMS/NLMS adaptive filtering to remove noise from audio signals by processing the signal in frames, resulting in improved clarity and higher SNR.	Prons: High stability due to embedded internal spectral models. Cons : Performance decreases for highly non-stationary noise.

Author's	Year	Title	Methodology	Pros/Cons
William Fong, Simon J. Godsill, Arnaud Doucet, Mike West	2024	Monte Carlo Smoothing With Application to Audio Signal Enhancement	Proposes Rao–Blackwellized and block-based particle smoothing methods for nonlinear audio enhancement using time-varying AR/PARCOR models.	Prons: More accurate smoothing than EKF and standard particle filters, reduced variance via analytical marginalization, suitable for long audio signals.. Cons Computationally expensive, dependent on particle count, block-based approach may lose long-range temporal consistency.
Duan Yu, Wang Jinzhen, Su Shaoying, Chen Zengping	2020	Detection of LFM Signals in Low SNR Based on STFT and Wavelet Denoising	Proposes a two-stage method combining Short-Time Fourier Transform (STFT) and wavelet denoising to detect LFM radar signals even at very low SNR (up to –18 dB). STFT extracts a rough time-frequency curve, and wavelet transform removes noise to make LFM features clearer.	Prons:Works effectively at very low SNR (up to –18 dB). Cons:Resolution depends heavily on the STFT window choice.

Author's	Year	Title	Methodology	Pros/Cons
Jean Jiang	2018	Audio Processing with Channel Filtering using DSP Techniques	Develops a DSP-based surround audio system using low-pass, band-pass, and high-pass FIR filters (designed via Remez/Parks–McClellan) and LM386-based multi-channel amplifiers for bass, mid-range, and treble processing..	Pros:Improved sound quality across frequency bands, flexible programmable DSP implementation, effective 3-channel separation. cons: Requires multiple DSP boards, limited power output of LM386, lacks integrated surround-sound processing in current prototype
Dzmitry Saladukha, Ivan Koriabkin, Kanstantsin Artsiom, Aliaksei Rak, Nikita Ryzhikov	2025	IMultichannel Keyword Spotting for Noisy Conditions	Proposes a multichannel KWS system combining adaptive noise cancellation and attention-based channel selection to improve keyword detection in noisy environments.	Prons:Significantly lower false reject rate, effective noise robustness, computationally efficient for on-device use. Cons: ANC may suppress keyword if user speaks before activation, requires multichannel hardware, performance depends on channel quality.

Author's	Year	Title	Methodology	Pros/Cons
Daniel Ogof et al.	2024	Enhancing Audio Comprehension in Large Language Models: Integrating Audio Knowledge	Integrates audio understanding into Mistral LLM using audio encoders and attention fusion..	Pros: Better audio-text performance, multilingua Cons: :High resource usage.
Cassia Valentini-Botinhao, Andrea Lorena Aldana Blanco, Ondrej Klejch	2023	Efficient Intelligibility Evaluation Using Keyword Spotting: A Study on Audio-Visual Speech Enhancement	Proposes a keyword-spotting-based subjective intelligibility test using mined phonetic alternatives and language-model–driven target word selection for evaluating audio-visual speech enhancement	Prons:Faster than transcription tests, less listener fatigue, works with in-the-wild AV material, highly correlated with traditional intelligibility scoring. Cons:Closed-set format may inflate accuracy, requires careful alternative selection, limited control over lexical content in pre-existing recordings.

CHALLENGES

1. Non-Stationary & Real-World Noise

Sudden noise from vehicles, wind, or crowd chatter is difficult to suppress without causing speech distortion. Even advanced filters like MMSE and LMS struggle when noise patterns change rapidly..

2. Real-Time Processing & Latency

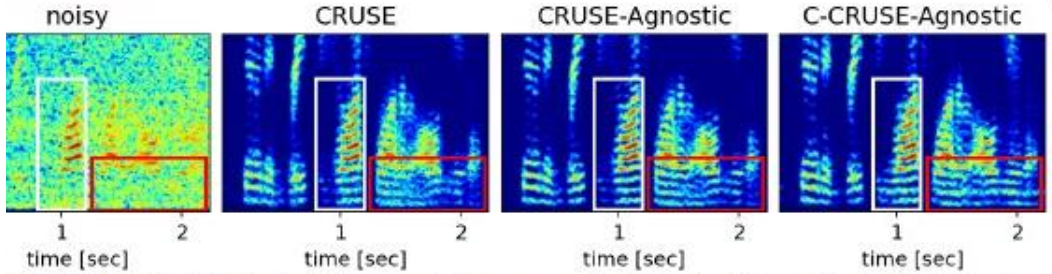
Complex DSP operations such as MMSE-STFT, LMS, STFT analysis, and L-filtering require high computational power. Maintaining latency below 20 ms is challenging, especially on mobile or low-power devices.

3. Speech Distortion & Loss of Naturalness

Over-filtering can remove important speech harmonics, making audio sound robotic or muffled. Ensuring clarity that remains natural and free from artifacts becomes a major challenge.

4. Personalized Processing Complexity

Personalized enhancement requires ear-specific processing such as channel splitting, frequency-based gain, and audiogram mapping, but maintaining a natural Left–Right balance without causing listening fatigue is difficult.



denoised signals by CRUSE and with the proposed phase reconstruction. Noisy signal: Street noise, -2 dB. Note the clearer harmonic nstruction by the proposed method in the highlighted area. C-CRUSE in combination with phase reconstruction even manages to pick tructure (white box) and gives a more continuous harmonic spectrum (red box)

ance, due to matched conditions. We denote gnostic' and 'SE-Matched' in the sequel.
r the compression factor p and the weights γ y grid search. Based on the DNSMOS scores band, quasi-stationary noise) of development challenge [20], the optimal parameters were $= 5]$ for SE-Agnostic, and $[p = 0.5, \gamma =$ ched.

ssion

is evaluated on the DNS2021 synthetic test etrics are given in Tab. 1. Compared to the nt improvement is provided by all approaches, an *upper bound* on achievable performance, where the *oracle, clean* phase is used with the litudes. Comparing CRUSE and C-CRUSE, fit of estimating the phase in the enhancement vious works.
the proposed phase enhancement improves all ured to *both* corresponding baselines (CRUSE STOI is constant for all approaches. This is cement should not affect the speech *envelope*, intelligibility. This *sanity check* ensures we at the cost of intelligibility. We also see that tputs are *comparable* to using oracle phase – shown in Fig. 3 for both subsets. We now see that on *subset a*, SE-Matched has a bigger margin over the SE-Agnostic. When the noise is less stationary and sparse (*subset b*), using SE-Agnostic is better. We reason that in such cases there are fewer contiguous regions where the speech and noise overlap. Then, SE-Agnostic networks, being trained on clean speech, yield more accurate phase estimates. The averaged results in Tab. 1 may indicate only a small achievable improvement by using the proposed phase reconstruction on CRUSE/C-CRUSE. However, Fig. 3 shows that phase reconstruction manages to boost the signal quality in poor SNR conditions – reflected by the decreased spread and higher minimum in the scores!

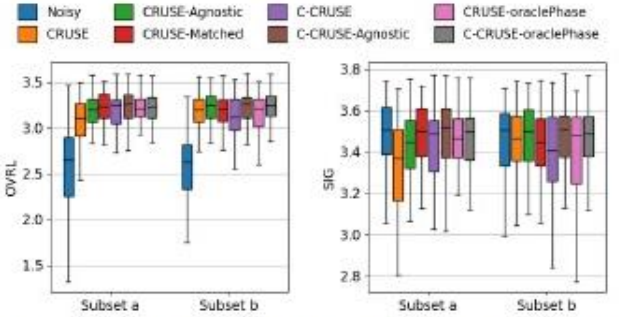


Fig. 3: DNSMOS score distribution, separately on broadband/quasi-stationary and transient/sparse noise subsets from DNS 2021 test set.

MOTIVATION

- Existing audio enhancement methods struggle in noisy, dynamic environments and often distort sound, creating a need for a more adaptive and reliable solution.

- Users have different hearing abilities, but most systems lack personalisation—motivating the development of an intelligent system that delivers clear, natural, and customised audio.

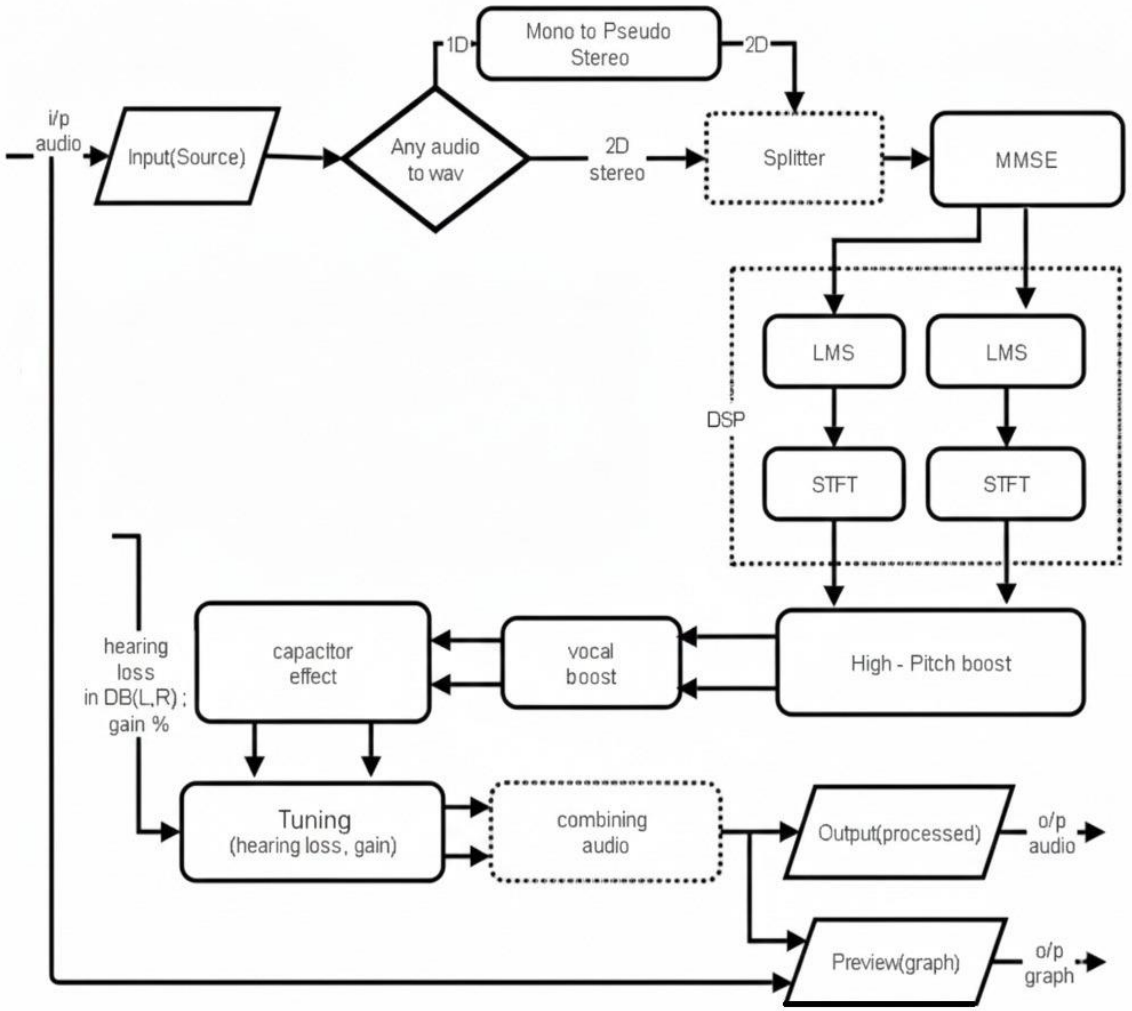
PROBLEM STATEMENT

Audio enhancement systems face challenges in dynamic noise environments, where traditional filters distort sound and fail to adapt. Real-time processing requires high computational power, increasing delay and complexity. Format compatibility issues can introduce additional distortion. The core challenge is to build an adaptive, efficient system that improves audio quality while preserving naturalness.

OBJECTIVES

- Improve Audio Clarity:Reduce background noise while preserving speech harmonics and naturalness.
- Achieve Real-Time Processing:Optimize DSP methods (MMSE, LMS, STFT, filtering) for low-latency enhancement suitable for live use.
- Provide Personalized Enhancement :Use audiograms, channel splitting, and frequency-based adjustments to meet individual hearing needs.
- Maintain Natural Sound Quality:Avoid robotic, harsh, or over-filtered audio output.
- Improve Robustness Across Environments:Ensure consistent performance across different acoustic conditions, microphones, rooms, and noise types.

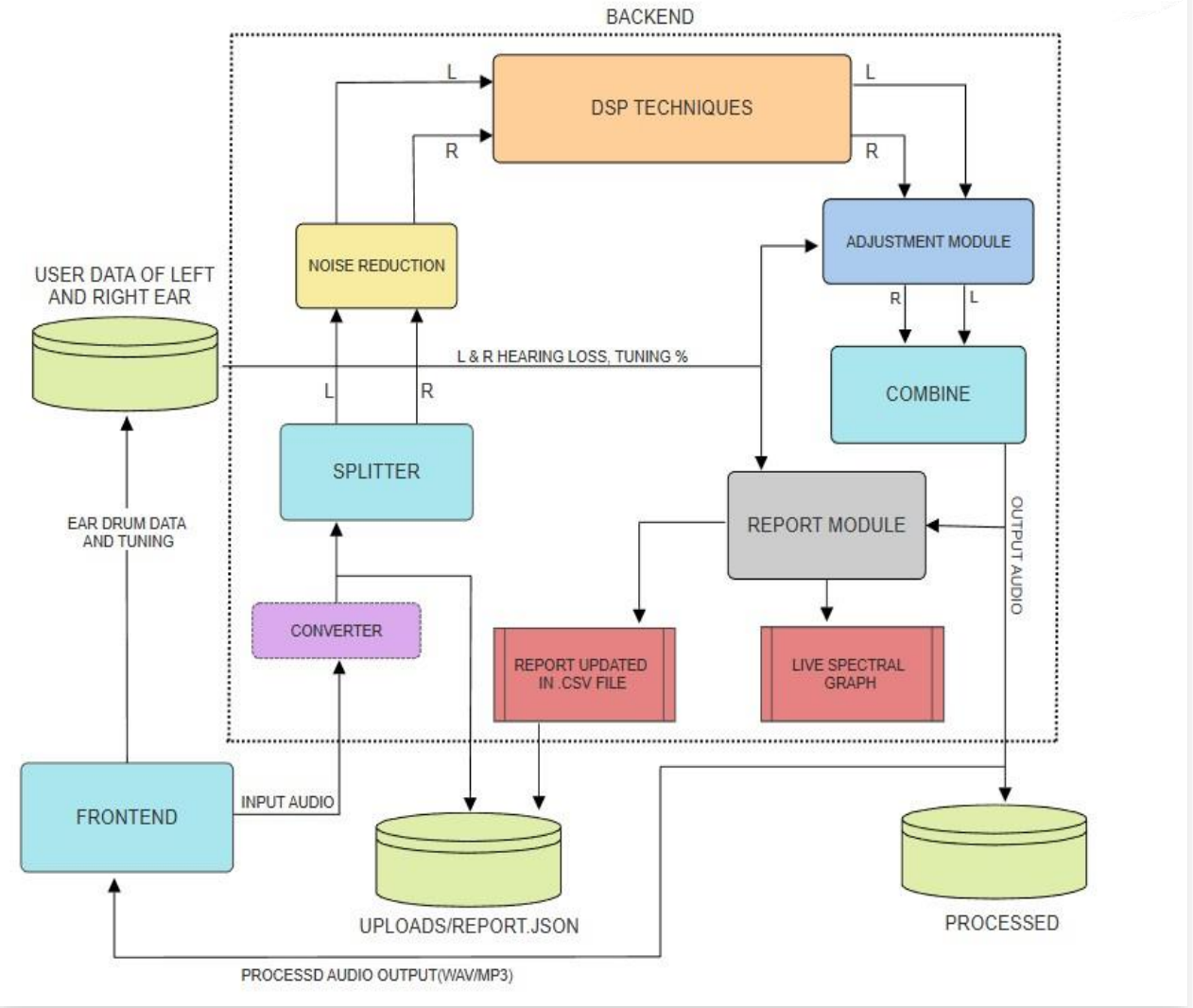
METHODOLOGY



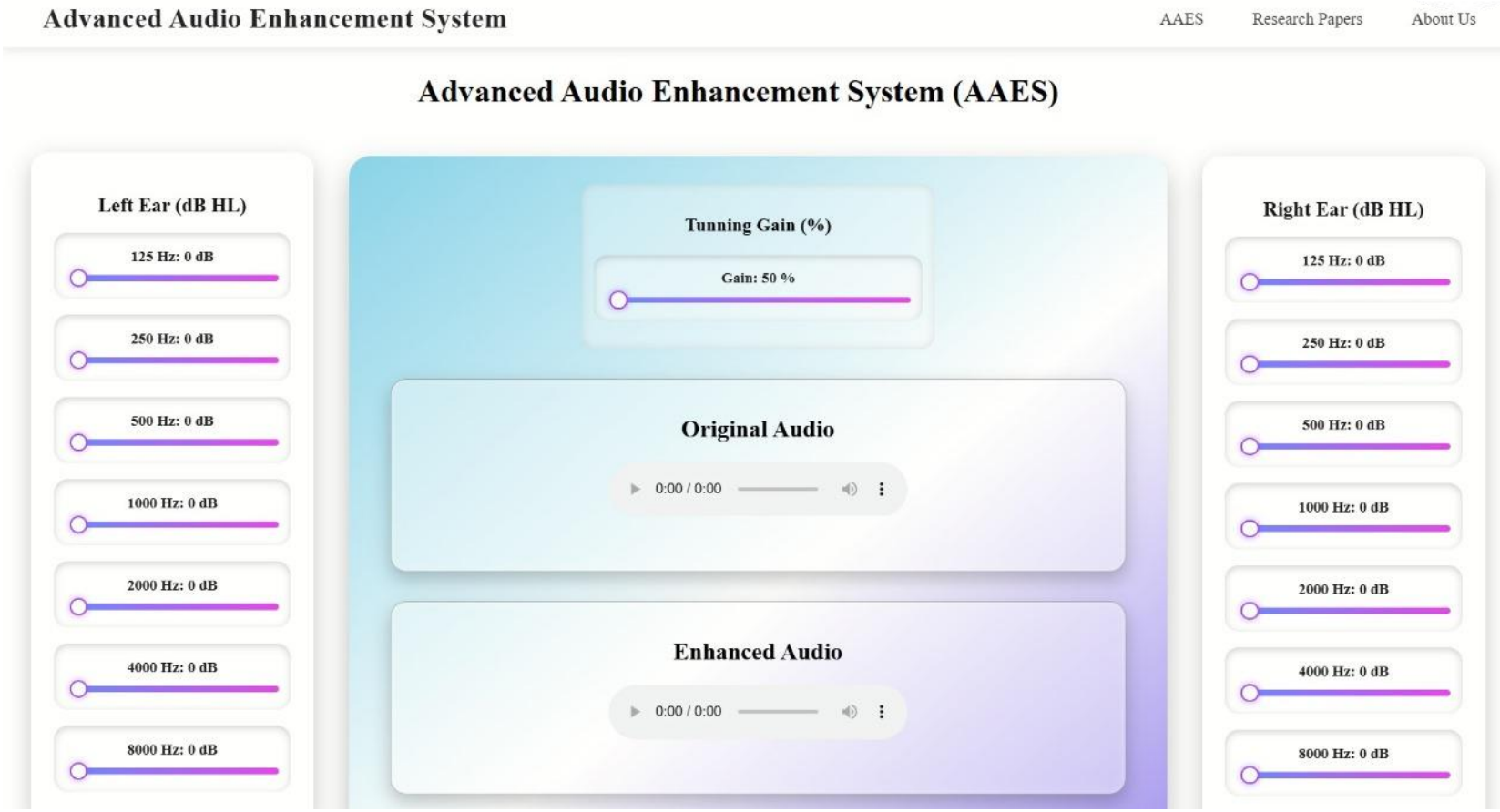
- **User-Specific Customization**:Tailors audio using left/right ear data, hearing loss profiles, and gain settings for personalized enhancement.
- **Stereo Channel Simulation**:Converts mono input to pseudo-stereo to enable spatial audio processing.
- **Stereo Channel Splitting**:Splits audio into L and R channels for ear-specific signal treatment.
- **Dedicated DSP Techniques**:Applies LMS filtering, STFT analysis, pitch/vocal boosts, and capacitor effects to each channel.
- **Tuning and Personalization**:Adjusts processed signals based on hearing loss and gain percentage for optimal clarity.
- **Signal Recombination**:Merges enhanced L and R channels into a unified stereo output.
- **Advanced Noise Reduction**: Reduces background noise post-processing to improve speech and overall clarity.
- **Format Flexibility**:Supports WAV/MP3 output for compatibility across devices and platforms.
- **Persistent Audio Storage**:Stores raw input in Uploads and enhanced output in Processed; logs reports and graphs.
- **Frontend-Backend Architecture**:Separates user interaction (frontend) from audio processing and enhancement (backend).

IMPLEMENTATION

- **Input**: Audio and user-specific ear data (left and right ear drum response) are received through the Frontend. This data is stored in the User Data of Left and Right Ear for personalized processing.
- **Conversion**: The input audio is optionally passed through a **Converter**.
- **Splitting**: The audio is then split into **left (L)** and **right (R)** channels using a **Splitter**, which utilizes the user data for personalized processing.
- **DSP Processing**: Each channel undergoes **DSP Techniques** (Digital Signal Processing) in the **Backend** for enhancement.
- **Noise Reduction**: The combined signal is passed through **Noise Reduction.**
- **Adjustment**:The Adjustment Module uses the user's **L & R Hearing Loss data** and **Tuning %** to fine-tune the combined audio, ensuring optimal personalization.
- **Combining**: The processed L and R channels are **combined**.
- **Output**: The final enhanced audio is stored in the **Processed** database and can be downloaded as **WAV/MP3**.
- **Uploads**: Original input audio is also stored for reference or reprocessing.



FRONTEND


Advanced Audio Enhancement System
AAES
Research Papers
About Us
Advanced Audio Enhancement System (AAES)
Left Ear (dB HL)
125 Hz: 0 dB
250 Hz: 0 dB
500 Hz: 0 dB
1000 Hz: 0 dB
2000 Hz: 0 dB
4000 Hz: 0 dB
8000 Hz: 0 dB
Tunning Gain (%)
Gain: 50 %
Original Audio
0:00 / 0:00
Enhanced Audio
0:00 / 0:00
Right Ear (dB HL)
125 Hz: 0 dB
250 Hz: 0 dB
500 Hz: 0 dB
1000 Hz: 0 dB
2000 Hz: 0 dB
4000 Hz: 0 dB
8000 Hz: 0 dB

Research Papers for References

Browse and read the IEEE research papers:

Research Paper 1: MMSE (noise reduction)

Research Paper 2: LMS (dsp,noise/error)

Research Paper 3: STFT (dsp)

Research Paper 4: L-filter (filter,dsp)

Research Paper 5: Tuning (channel,dB gain)

About Us

Description

AAES is a digital signal processing (DSP) based solution designed to enhance audio quality by reducing noise, improving stereo separation, and optimizing frequency balance. Built with a React frontend and a Python backend, AAES offers a seamless and efficient user experience. The system leverages advanced algorithms for real-time audio enhancement, making it suitable for various applications like music streaming, VoIP communication, gaming, and assistive technologies.

Tech Stack

Frontend

React, CSS, HTML

Backend

Flask / FastAPI – API & Request Handling

Audio Processing

NumPy, SciPy – Signal Processing
PyDub – Format Conversion & Manipulation
FFmpeg – Encoding / Decoding
Librosa – Feature Extraction
NoiseReduce – LMS Filter & MMSE Filter
Soundfile – WAV / FLAC I/O

Team members

Sameer Kumar Choudhury
Email: sameer133@gmail.com
Sneha Mal
Email: snehamal@gmail.com
Shristi Priya
Email: shristipriya@gmail.com
Surath Choudhury
Email: surath172003@gmail.com

BACKEND

```python
def process_audio_with_full_chain(input_path, output_path
    # 9. Match Lengths
    left_m, right_m = match_audio_lengths(left_cap, r

    # 10. Apply Audiogram Tuning
    left_tuned, right_tuned = apply_audiogram(
        left_m, right_m, sr, hearing_loss, tuning_gai
    )

    # 11. Final Volume Boost
    final_audio = np.vstack((left_tuned, right_tuned)
    final_audio = final_volume_boost(final_audio)

    # 12. SAVE OUTPUT
    sf.write(output_path, final_audio.T, sr)

    end_time = time.time()
    latency_ms = (end_time - start_time) * 1000.0

    # ---- GENERATE METRIC REPORT ----
    report_data = generate_report(
        original_audio=original_stereo,
        enhanced_audio=final_audio,
        sr=sr,
        hearing_loss=hearing_loss,
        tuning_gain=tuning_gain,
        latency_ms=latency_ms,
    )
    report_json_path = output_path.replace(".wav", "_
    with open(report_json_path, "w") as f:
        json.dump(report_data, f, indent=4)

    append_report_to_excel(report_data)

    print(f"📄 Report generated at: {report_json_path
    print(f"✅ Processing complete: {output_path}")
```

```python
def process_audio_with_full_chain(input_path, output_path,
    """
    Full AAES DSP pipeline + reporting hooks.
    """
    try:
        start_time = time.time()

        print(f"🎧 Processing audio (full chain): {input_path

        # 1. Convert to WAV if needed
        if not input_path.lower().endswith(".wav"):
            input_path = convert_to_wav(input_path)

        # 2. Load audio
        left, right, sr = load_audio(input_path)

        # FIX: Save ORIGINAL stereo audio for SNR/PESQ/STOI
        original_stereo = np.vstack((left, right))

        # 3. Combine channels for INITIAL noise reduction
        print("🔰 Initial Noise Reduction (MMSE at beginning)"
        stereo_raw = np.vstack((left, right))
        mmse_initial = mmse_filter(stereo_raw, sr)

        # Extract denoised channels
        left_clean, right_clean = mmse_initial[0], mmse_initia

        # 4. LMS filter
        left_lms = apply_lms_filter(left_clean)
        right_lms = apply_lms_filter(right_clean)

        # 5. STFT denoising (light)
        left_den = stft_denoising(left_lms, sr)
        right_den = stft_denoising(right_lms, sr)
```

```
---------- Utility Functions ----------
compute_snr(signal, noisy):
noise = noisy - signal
snr = 10 * np.log10(
    (np.sum(signal ** 2) + 1e-12) / (np.sum(noise
)
return float(snr)

compute_lsd(clean, enhanced):
eps = 1e-8
stft_clean = np.abs(librosa.stft(clean)) + eps
stft_enhanced = np.abs(librosa.stft(enhanced)) + e
lsd = np.mean(np.sqrt(np.mean((20*np.log10(stft_cl
return float(lsd)

stereo_energy_balance(left, right):
L_energy = np.sum(left ** 2)
R_energy = np.sum(right ** 2)
if R_energy == 0:
    return "Undefined"
balance_db = 10 * np.log10(L_energy / R_energy)
return float(balance_db)
```

```
86  # ---------- Main Report Generator ----------
87  def generate_report(original_audio, enhanced_audio, sr, hearing_loss, tuni
88
89      orig_mono = np.mean(original_audio, axis=0)
90      enh_mono = np.mean(enhanced_audio, axis=0)
91
92      # 1. SNR
93      snr_before = compute_snr(orig_mono, orig_mono + 1e-6)   # no 0-noise
94      snr_after = compute_snr(orig_mono, enh_mono)
95      delta_snr = snr_after - snr_before
96
97      # 2. PESQ
98      if pesq:
99          try:
100             pesq_score = pesq(sr, orig_mono, enh_mono, 'wb')
101         except:
102             pesq_score = None
103     else:
104         pesq_score = None
105
106
107     # 3. STOI
108     if stoi:
109         try:
110             stoi_score = stoi(orig_mono, enh_mono, sr, extended=False)
111         except:
112             stoi_score = None
113     else:
114         stoi_score = None
115
116
117     # 4. LSD
118     lsd_value = compute_lsd(orig_mono, enh_mono)
119
120     # 5. Stereo balance
121     seb = stereo_energy_balance(enhanced_audio[0], enhanced_audio[1])
```

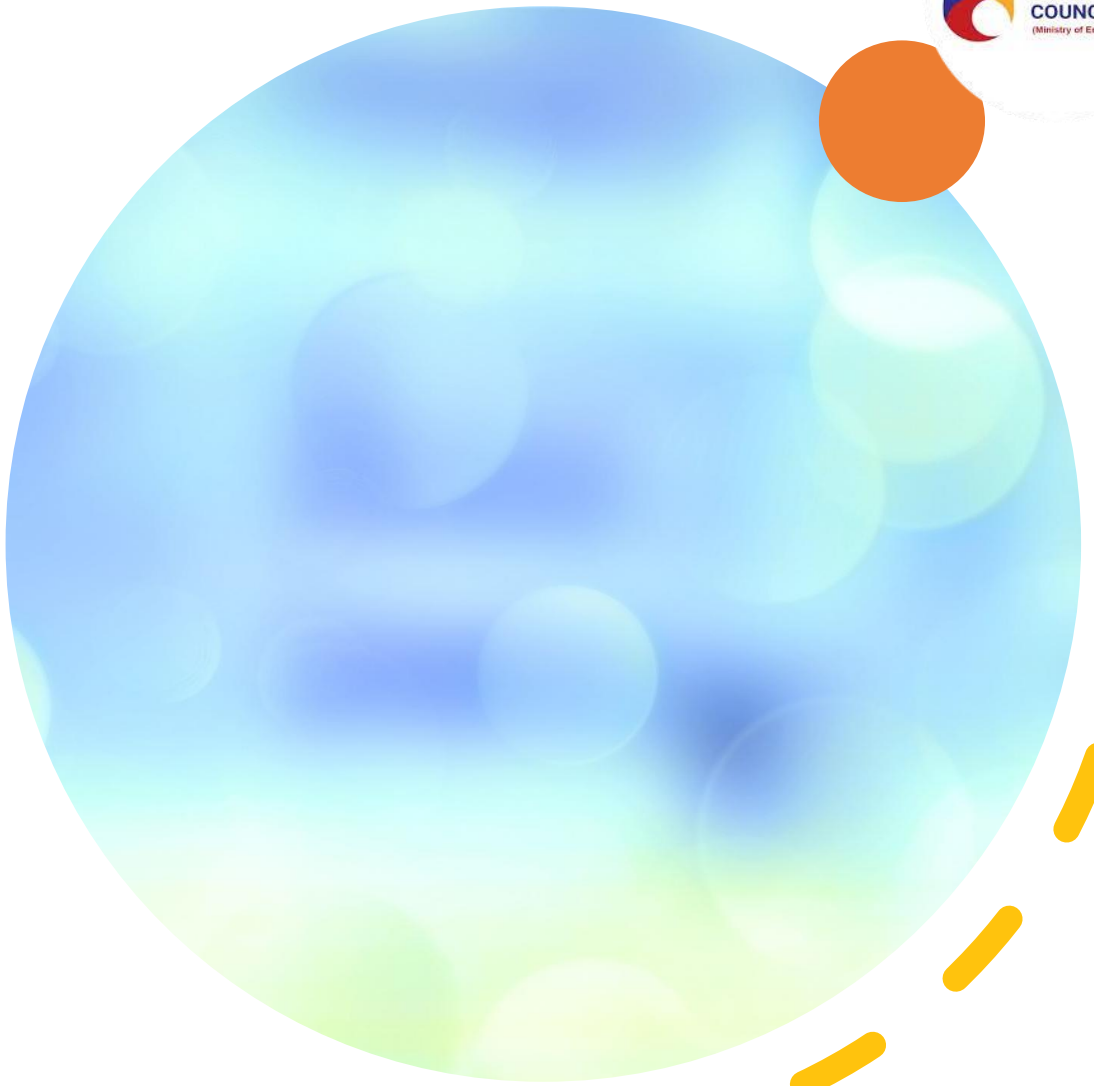
DATASET FOR PROJECT

Audiogram Dataset :

1,018 JPG images (638×664 px, 45–50 KB) split into Left and Right Ear Charts, providing balanced hearing profiles for ML integration.

Speech Audio Dataset :

12,162 labelled .wav samples (45–50 KB each) from CREMA-D, RAVDESS, TESS, and SAVEE, covering diverse speakers, accents, emotions, and utterances for robust speech recognition training.



CONCLUSION

The Advanced Audio Enhancement System (AAES) demonstrates that a modular DSP-driven framework can successfully personalize audio processing by integrating techniques such as LMS adaptive filtering, STFT analysis, capacitor-effect smoothing, and MMSE noise reduction. Together, these methods enhance clarity, reduce noise, and preserve stereo fidelity across diverse audio categories including speech, music, and environmental recordings. Experimental results confirm that AAES delivers a natural, balanced, and perceptually improved auditory experience, making it suitable for both general audio enhancement and assistive hearing applications. Looking ahead, the system aims to achieve faster processing on low-power devices, incorporate AI/ML for smarter adaptation, expand into mobile platforms, and ensure broader compatibility across playback environments.

REFRENCES

- [1] https://ieeexplore.ieee.org/document/6516079
- [2] https://www.atlantis-press.com/proceedings/icmt-13/10475
- [3] https://ieeexplore.ieee.org/document/7009929
- [4] https://etheses.whiterose.ac.uk/id/eprint/9070/1/thesis.pdf
- [5] https://www.isca-archive.org/eurospeech_2003/paliwal03b_eurospeech.pdf
- [6] https://ieeexplore.ieee.org/abstract/document/8301696
- [7] https://ieeexplore.ieee.org/document/10096479
- [8] https://www.isca-archive.org/interspeech_2025/saladukha25_interspeech.pdf
- [9] https://www.kaggle.com/datasets/danielasgo/dataset-image-audiograms-categorized
- [10] https://www.kaggle.com/code/chaluvadiravindra/speech-recognition-lstm-tensorflow