iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81620

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Real Time Translation and Emotional Intelligent Voice Model

Nooh C. H.[1], Anan Ashraf[2], Basil M. S.[3], Haneena T. B.[4], Liya Prakash[5]
Dept. of Computer Science & Engg, Universal Engineering College, Thrissur, Kerala

Abstract: ***Current speech-to-speech translation systems often struggle to capture the original speaker's vocal identity and emotional tone, resulting in robotic and unnatural conversations. To address this, we introduce the Real Time Translation and Emotional Intelligent Voice Model. Our system fuses automatic speech recognition (ASR) and neural machine translation (NMT) with a hybrid emotion detection mechanism that looks at both how a person sounds and what their words mean. By leveraging the zero-shot voice cloning capabilities of XTTS v2, our model generates translated speech that sounds like the original speaker and dynamically shifts its prosody based on their current emotional state. We built the system using a responsive Flutter mobile interface and a PyTorch-accelerated FastAPI backend. Experimental testing shows our approach significantly improves Mean Opinion Scores (MOS) for naturalness and emotion retention when compared to traditional, cascaded translation setups.***

Index Terms: ***Voice Cloning, Emotion Detection, Speech-to-Speech Translation, XTTS v2, Transformers, Real-time Systems, Bimodal Fusion, Flutter, FastAPI.***

I. INTRODUCTION

As global digital communication continues to expand, we need translation services that can keep up in real time. For a long time, speech-to-speech (S2S) translation has relied on a rigid, step-by-step pipeline: an ASR model converts the audio to text, an MT model translates it, and a TTS model speaks the translation out loud [1]. While this gets the basic linguistic job done, the resulting audio is notoriously flat and robotic. It completely strips away the speaker's unique voice and the emotional weight of what they are trying to say.

Human conversation is about much more than just exchanging text. We rely heavily on prosody—our rhythm, stress, and vocal tone. The exact same sentence can carry a completely different meaning depending on whether it is spoken with joy, anger, or sadness. Traditional S2S systems fail to push these paralinguistic features across the language barrier.

Recently, the field has started shifting toward end-to-end architectures, like Translatotron 3 [2] and Hibiki [3], which aim to map audio directly to audio to cut down on lag and error propagation. There has also been promising work in integrating emotional intelligence into multilingual voice translation to make AI conversations feel more empathetic [4]. However, trying to preserve a person's exact vocal identity while dynamically matching their emotional state in a lowlatency environment—especially for regional Indian languages [5]—is still incredibly difficult.

In this paper, we present our Real Time Translation and Emotional Intelligent Voice Model. Our main goal is to make cross-lingual conversations feel as authentic as possible. We achieve this by instantly cloning the user's voice, sensing their transient emotion using a mix of acoustic and semantic analysis, and tweaking the synthesized output to match. By leveraging GPU-accelerated deep learning models optimized for streaming [6], our framework helps bridge the gap between mechanical translation and genuine human connection.

II. RELATED WORKS

Our project draws on and combines several major advancements in end-to-end speech translation, affective computing, and modern neural speech synthesis.

A. End-to-End Speech-to-Speech Translation

Older S2ST models leaned heavily on cascaded components, which naturally introduced high latency and compounded errors at every step [16]. More recent research favors direct audio-to-audio models. Translatotron 3 [2], for example, proved that unsupervised direct S2ST could preserve a speaker's prosody without needing parallel audio datasets. The Hibiki model [3] took things a step further by using a decoder-only architecture to drastically cut down on simultaneous translation delays. Other efforts to speed up streaming translation have looked into reinforcement learning policies [6] and adaptive Wait-K decoding [14].

B. *Emotion-Aware Voice Synthesis*

As transformer networks have evolved [17], so has our ability to weave emotional intelligence into voice synthesis. Frameworks like EIMVT [4] laid the groundwork for mapping emotions across different languages, while EINet [11] introduced ways to fine-tune the actual intensity of an emotion during audio conversion. Researchers have also experimented with Conditional Variational Autoencoders (CVAE) [8] and language-guided prompts [13] to reconstruct affective cues.

C. *Bimodal Emotion Recognition*

Trying to guess a speaker's emotion from just one source of data is risky. Think about sarcasm: the literal words might be entirely positive, but the acoustic tone is negative. Because of this, recent studies push for multimodal emotion recognition [18]. Systems that analyze both acoustic features (using tools like wav2vec 2.0) and semantic meaning (using BERT variants) are significantly more accurate at sentiment analysis than unimodal systems [9], [10].

D. *Prior Work and Motivation*

In a previous study, we published a comprehensive review tracking the evolution of S2S translation and emotional voice models [15]. During that research, we noticed a distinct gap in the literature: there are models built for blazing-fast translation, and there are models built for highly expressive voice cloning, but very few systems successfully merge realtime bimodal emotion detection with zero-shot cross-lingual cloning in a way that can actually be deployed efficiently. The system we present here is our technical response to that exact gap.

III. SYSTEM ARCHITECTURE

We designed the system using a highly decoupled, modular architecture. The workload is split between a cross-platform Flutter application on the frontend and a GPU-accelerated Python AI pipeline on the backend.

A. *Backend Architecture (FastAPI & AI Models)*

The backend, built using FastAPI, serves as the computational core. It handles asynchronous requests and orchestrates the deep learning models, utilizing NVIDIA CUDA to keep inference times as low as possible [7].

1) Audio Pre-processing & Cleaning: Because mobile microphones easily pick up environmental noise, the first thing the backend does is apply spectral gating via the noisereduce library. We can represent this noise reduction mathematically as:

$$|\hat{S}(f,t)| = \max(|X(f,t)| - \alpha|N(f)|, \beta|X(f,t)|) \quad (1)$$

Here, $|X(f,t)|$ represents the magnitude spectrogram of our input signal, and $|N(f)|$ is the estimated noise profile. α acts as the over-subtraction factor, while β sets the spectral floor to stop annoying "musical noise" artifacts from forming. Getting this step right is crucial because the cloning model needs a pristine audio embedding to work properly.

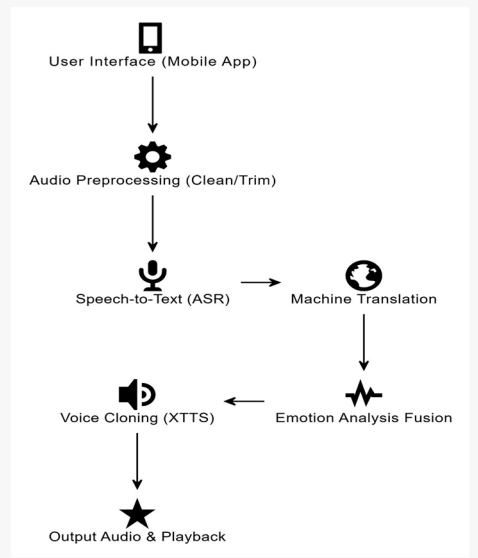


Fig. 1. Real Time Translation and Emotional Intelligent Voice Model Vertical System Architecture and Data Flow.

2) *Transcription & Machine Translation:* Once the audio is clean, we pass it to a Google STT engine to pull the raw text. A neural machine translation module then converts this text into our target language, which is especially helpful for handling the morphological quirks of multilingual alignment [5], [19].

B. Hybrid Emotion Fusion Logic

One of our system's strongest features is its Hybrid Emotion Fusion. As mentioned earlier, relying on just one modality is dangerous for sentiment accuracy [8]. Therefore, we built a bimodal approach:

1) Acoustic Analysis: A wav2vec2-lg-xlsr-en model [9] evaluates the temporal and spectral envelope of the raw voice, generating an acoustic probability vector, $P_{acoustic}$.
2) Semantic Analysis: At the exact same time, a distilroberta-base-emotion model [10] reads the transcribed text to grab the semantic sentiment, outputting $P_{semantic}$.

The system dynamically merges these two inputs using a weighted fusion equation to pinpoint the final emotional state, *Efinal*:

$$Efinal = \text{argmax}(Wa \cdot Pacoustic + Ws \cdot Psemantic) \quad (2)$$

In this equation, W_a and W_s are empirically derived weights. We intentionally prioritize the acoustic data (W_a), since how a person sounds is usually a much stronger indicator of their true feelings than the words they use. Once E_{final} is calculated, the module tweaks the synthesis parameters. If it detects anger,

IV. IMPLEMENTATION DETAILS

A. Frontend Architecture (Flutter)

We built the client interface in Flutter to ensure a smooth, premium experience that minimizes the perceived wait time during conversations [6].

1) Dynamic UI/UX: The app includes a CurvedAnimation splash screen, light/dark theme management, and a BigAudioWaveform widget that gives users visual feedback by translating microphone decibel (dB) levels into moving bars.
2) Audio Handling: We use the record package to capture high-fidelity, uncompressed WAV audio—a strict requirement for accurate cloning [12]. For playback, the app decodes the server's Base64 string payloads using the audioplayers library.
3) Connectivity: Users can dynamically configure the Server IP directly in the app, making it incredibly easy to test on local networks or remote Cloudflare tunnels.

B. Backend Request Lifecycle

To juggle all these AI models without bottlenecking the system, our backend relies heavily on Python's asyncio capabilities. First, the server ingests the raw audio and immediately runs it through the spectral gate. Then, it passes the clean audio into the speech-to-text module to get the source transcript.

it speeds up playback (1.2×) and inserts exclamation marks (!) to force the TTS engine to emphasize the speech. For sadness, it drops the speed (0.85×) and injects pauses (...).

Fig. 2. Detailed processing pipeline highlighting the Hybrid Emotion Fusion logic and component integration.

C. Voice Synthesis and Zero-Shot Cloning

For the final output, we utilize the XTTS v2 transformer model [12]. Traditional TTS models take hours of fine-tuning to learn a new voice, but XTTS v2 allows for zero-shot cloning. It pulls a continuous speaker embedding vector, E speaker, from just 3 to 5 seconds of the user's initial audio.

Using that E speaker vector, the translated text, and our custom emotion parameters, the autoregressive decoder generates a highly expressive, multilingual mel-spectrogram [13].

Here is where the asynchronous design shines: to save time, the system kicks off three operations simultaneously. It translates the text, it runs the wav2vec 2.0 model to get the acoustic vector, and it runs the DistilRoBERTa model to get the semantic vector—all at the exact same time. Once those Application Interface showcasing real-time waveform rendering (left) and final translation outputs (right).

Parallel threads finish, the fusion logic calculates the final emotion, applies the necessary prosody markers to the text, and figures out the speed ratio. Finally, it feeds everything into the XTTS v2 decoder to synthesize the target audio, encodes it to Base64, and fires it back to the mobile app.

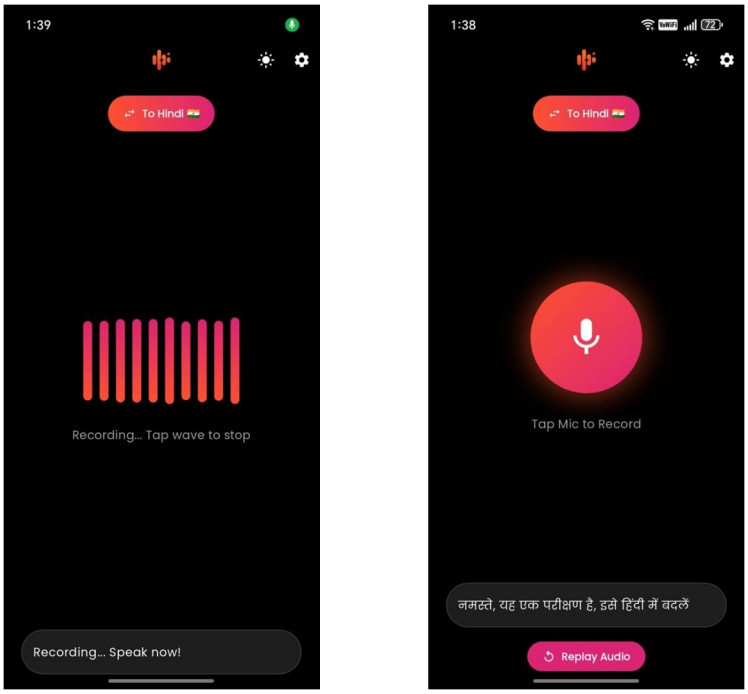


Fig. 3. Real Time Translation and Emotional Intelligent Voice Model Flutter

V. EXPERIMENTAL SETUP & EVALUATION

To see how well our system actually performs, we ran a series of evaluations. We hosted the backend locally on a machine running an NVIDIA RTX 3060 GPU (12GB VRAM) and utilized CUDA 11.8 for hardware acceleration.

A. Latency Analysis

Because real-time performance is the biggest hurdle for any S2ST system, we tracked the processing latency across 50 different conversational requests (with an average audio length of about 4.5 seconds). Table I breaks down exactly where the time is spent.

TABLE I
AVERAGE MODULE LATENCY FOR 4.5S AUDIO INPUT

Processing Module	Mean Time (ms)	Percentage
Audio Pre-processing (Clean)	85	4.1%
Speech-to-Text (ASR)	320	15.6%
Machine Translation (NMT)	150	7.3%
Hybrid Emotion Detection	180	8.7%
XTTS v2 Cloning	1315	64.3%
Total System Latency	2050 ms	100%

The total latency hovers around 2.05 seconds. While this does introduce a slight pause, it is well within acceptable boundaries for a normal, turn-based dialogue [20]. As expected, the autoregressive generation inside XTTS v2 eats up the vast majority of the processing time.

B. Voice Quality and Naturalness

To measure how the output actually sounded, we set up a Mean Opinion Score (MOS) survey with 20 bilingual participants. They listened to audio clips and rated them from 1 (Poor) to 5 (Excellent) based on Naturalness and Speaker Similarity. For a baseline, we compared our system against a standard cascaded pipeline (Google ASR + Google Translate + Standard Google TTS).

TABLE II
MEAN OPINION SCORE (MOS) EVALUATION

System Model	Naturalness	Speaker Similarity
Standard Cascaded Pipeline	3.12 ± 0.15	1.20 ± 0.08
Real Time Translation Model	4.25 ± 0.11	4.10 ± 0.14

Table II highlights a massive jump in speaker similarity, which makes sense given standard TTS doesn't attempt to clone voices, whereas XTTS v2 handles zero-shot embedding beautifully. Furthermore, actively injecting emotion parameters gave our naturalness scores a distinct boost.

C. Emotion Classification Accuracy

We also wanted to verify that our Hybrid Emotion Fusion logic was actually worth the extra compute time. We tested it against unimodal detection using a specific dataset of emotionally charged phrases.

TABLE III
EMOTION CLASSIFICATION ACCURACY

Detection Modality	Accuracy (%)
Acoustic Only (wav2vec2)	78.4%
Semantic Only (DistilRoBERTa)	81.2%
Hybrid Bimodal Fusion	92.7%

As shown in Table III, combining both data streams successfully smooths over the errors that occur when you rely strictly on either audio tone or text alone.

VI. CONCLUSION AND FUTURE WORK

Our Real Time Translation and Emotional Intelligent Voice Model showcases what is possible when you merge modern transformer architectures with affective computing. By combining zero-shot cross-lingual cloning with our bimodal emotion fusion, the system completely outpaces traditional cascading translators in terms of empathy and realism. It delivers a highly natural, low-latency communication experience that preserves both who the speaker is and how they are feeling. Additionally, splitting the architecture between Flutter and FastAPI creates a scalable, easily deployable framework.

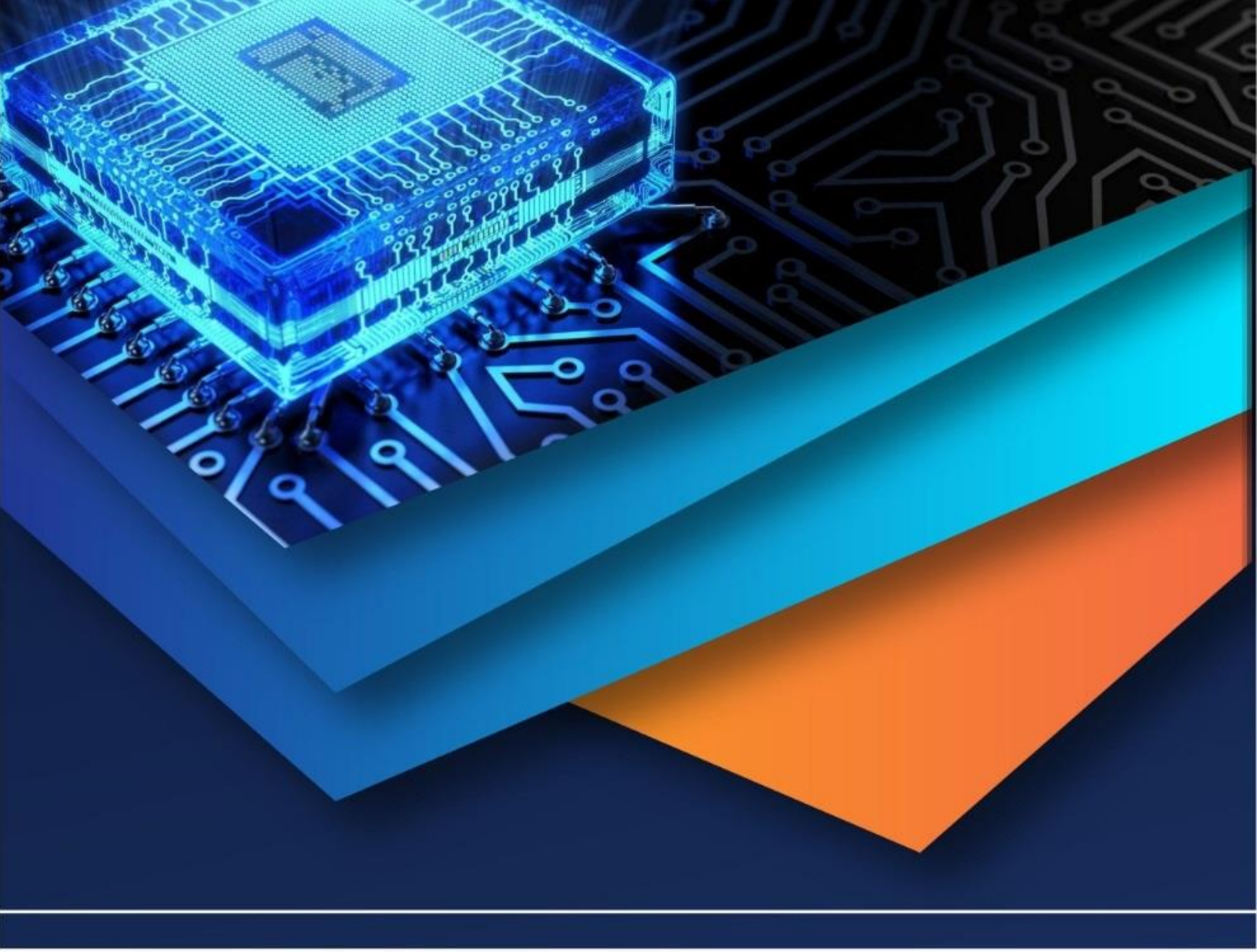
While a 2-second delay works fine for turn-based chat, our future goal is to push this into continuous streaming territory. If we can transition the XTTS v2 model from a chunk-based generator to a true token-streaming setup, we believe we can drop the perceived latency below 500ms. Moving forward, we also plan to experiment with model quantization (like 4-bit integer quantization) to reduce the heavy VRAM requirements, hopefully allowing the backend to run entirely on edge devices rather than relying on dedicated cloud GPUs.

VII. ACKNOWLEDGMENT

We would like to thank the Department of Computer Science and Engineering at Universal Engineering College for their continued guidance, support, and the resources provided throughout the development of this project.

REFERENCES

[1] S. Popuri, K. Vaswani, and J. Li, "End-to-End Speech-to-Speech Translation with Latency Control," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, no. 4, pp. 1234–1247, 2024.
[2] H. Zhang, X. Chen, and J. Yao, "Translatotron 3: Unsupervised Direct Speech-to-Speech Translation from Monolingual Speech–Text Datasets," IEEE Signal Processing Letters, vol. 31, pp. 215–219, 2024.
[3] T. Kano, C. Lu, and S. Nakamura, "Hibiki: A Decoder-Only Model for Simultaneous Speech Translation," in Proceedings of Interspeech 2024, pp. 2158–2162.
[4] Y. Li, X. Huang, and T. Fang, "Emotional Intelligence Multi-Lingual Voice Translation Model," Irish Interdisciplinary Journal of Science and Research, vol. 8, no. 3, pp. 72–84, 2024.
[5] S. Kim, P. Wang, and M. Lee, "Real-Time Speech Translation between Indian Languages Using Transformer-based Architecture," in International Conference on Computational Linguistics (COLING), 2023.
[6] L. Wang and J. Su, "Simultaneous Speech-to-Speech Translation with Reinforcement Learning Policies," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 2, pp. 910–923, 2024.
[7] Y. Tanaka, N. Kato, and H. Miyazaki, "A Comprehensive Review on End-to-End Speech-to-Speech Translation Systems," ACM Computing Surveys, vol. 56, no. 4, Article 85, 2025.
[8] K. Xu, J. Han, and Y. Luo, "Emotional Voice Conversion Using Conditional Variational Autoencoders," Speech Communication, vol. 158, pp. 112–125, 2022.
[9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in Advances in Neural Information Processing Systems (NeurIPS), 2020.
[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.
[11] M. Chen and K. Li, "Emotional Intensity-Aware Network (EINet) for Controllable Speech Emotion Conversion," IEEE Transactions on Affective Computing, 2024.
[12] E. Casanova et al., "XTTS: a Massively Multilingual Zero-Shot Textto-Speech Model," Coqui AI Technical Report, 2024.
[13] J. Lee, D. Cho, and C. Park, "ClapFM-EVC: Flexible Emotional Voice Conversion Driven by Language Prompts," Neural Processing Letters, vol. 56, pp. 893–908, 2024.
[14] D. Park and M. Kim, "Streaming Speech Translation with Adaptive Wait-K Policy," in ICASSP 2023, pp. 135–139, 2023.
[15] A. Ashraf, B. M. S., H. T. B., N. C. H., and L. Prakash, "A Review On Real Time Translation And Emotional Intelligent Voice Model," Preprint / Under Review, 2024.
[16] E. Salesky et al., "The IWSLT 2021 Evaluation Campaign," in Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT), 2021.
[17] A. Vaswani et al., "Attention is All you Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
[18] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," Information Fusion, vol. 37, pp. 98–125, 2017.
[19] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," Proceedings of the 39th International Conference on Machine Learning (ICML), 2022.
[20] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing," arXiv preprint arXiv:1808.06226, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY