VoxBridge: Fully Offline Real-Time Multilingual AI Voice Translator

Tufan Layek
tl5275@srmist.edu.in

Protyay Saha
ps0803@srmist.edu.in

Rounack Sarkar
rs8927@srmist.edu.in

Department of Electronics and Communication Engineering
SRM Institute of Science and Technology, Kattankulathur

February 2026

1 Abstract

VoxBridge is a fully offline, real-time multilingual speech translation system designed for edge environments where internet connectivity is unreliable. The system integrates automatic language detection, silence-aware speech segmentation, neural machine translation, and offline speech synthesis within a modern GUI. Through optimized transformer inference and CPU-based quantization techniques, VoxBridge achieves near real-time performance without cloud dependency.

2 System Architecture

The VoxBridge system follows a modular, pipeline-based architecture designed to enable real-time, fully offline multilingual voice translation. Each module performs a well-defined function in the overall data processing chain. The architecture is optimized to minimize latency, reduce redundant computation, and maintain robustness in noisy environments. Both the speech recognition and translation modules rely on the self-attention mechanism introduced in the transformer architecture [1].

2.1 1. Audio Acquisition Layer

The system begins with continuous audio capture through the host device microphone using low-latency streaming buffers. Audio samples are collected in fixed-duration frames and stored in a rolling buffer. The sampling rate is selected to balance computational efficiency and speech clarity, ensuring compatibility with the downstream speech recognition model.

Mathematically, the incoming audio signal can be represented as:

$$x(t) = s(t) + n(t)$$

where:

- $s(t)$ represents the speech component
- $n(t)$ represents environmental noise

The objective of subsequent modules is to isolate and process $s(t)$ while minimizing the effect of $n(t)$.

2.2 2. Voice Activity Detection (Silence-Aware Processing)

To prevent unnecessary model inference on silence or background noise, WebRTC Voice Activity Detection (VAD) is applied to incoming audio frames. The VAD module classifies each frame as either speech or non-speech.Silence detection is implemented using classical Voice Activity Detection (VAD) principles to improve robustness in noisy environments [2].

Only when speech segments are detected and followed by a silence threshold does the system trigger the transcription stage. This reduces computational load and prevents fragmented translation outputs.

Formally, a trigger event occurs when:

$$\sum_{i=1}^{k} VAD(frame_i) = 0$$

for a predefined silence window k, indicating the end of a speech segment.

2.3 3. Noise Reduction Module

To enhance transcription accuracy in real-world conditions, spectral noise reduction is applied. The algorithm estimates the noise profile from non-speech frames and subtracts it from the active speech segment using frequency-domain filtering.

This process improves the signal-to-noise ratio (SNR):

$$SNR = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right)$$

Improved SNR leads to better transformer-based transcription performance.

2.4 4. Speech-to-Text and Automatic Language Detection

The speech recognition module is based on the Whisper architecture, a transformer-based model trained using large-scale weak supervision for robust multilingual transcription [3].

The cleaned speech segment is passed to the Faster-Whisper model, an optimized implementation of the Whisper transformer architecture.

This module performs:

- Automatic speech recognition (ASR)
- Language identification

The transformer encoder maps acoustic features to latent representations, while the decoder generates text tokens sequentially.

If the token sequence is represented as:

$$Y = \{y_1, y_2, ..., y_n\}$$

then each token is generated based on:

$$P(y_t | y_{<t}, X)$$

where X represents the input acoustic feature sequence.

Language detection is inferred from the model's probability distribution over supported language tokens.

2.5 5. Neural Machine Translation

For text translation, VoxBridge employs MarianMT [4], an efficient transformer-based neural machine translation framework.

The recognized text is passed to the MarianMT transformer model corresponding to the selected target language.

Translation follows a sequence-to-sequence mapping:

$$T = f(Y)$$

where:

- Y is the source sentence
- T is the translated output
- f represents the neural translation function

The transformer architecture uses self-attention mechanisms to model long-range dependencies within sentences, ensuring contextual translation rather than word-level substitution.

2.6 6. Offline Text-to-Speech Synthesis

The translated text is converted to speech using an offline text-to-speech engine. The synthesis module converts phoneme sequences into waveform representations using parametric voice modeling.

The generated output waveform $\hat{s}(t)$ is played through the system speakers without requiring any external API.

2.7 7. Graphical User Interface Layer

The GUI layer provides:

- Target language selection
- Real-time waveform visualization
- Listening status indicator
- Latency display
- Dark/Light theme toggle

The GUI interacts asynchronously with backend processing threads to maintain responsiveness while inference operations execute in parallel.

2.8 8. End-to-End Data Flow

The complete pipeline can be summarized as:

$$Audio \rightarrow VAD \rightarrow NoiseReduction \rightarrow STT \rightarrow Translation \rightarrow TTS \rightarrow GUI$$

Each module operates sequentially but is optimized using buffered streaming and preloaded models to minimize end-to-end latency.

2.9 9. Design Characteristics

The architecture exhibits the following properties:

- Fully offline execution
- CPU-based transformer inference
- Low-latency performance
- Modular scalability
- Privacy-preserving design
- Edge-deployable configuration

This layered architecture ensures reliability, efficiency, and adaptability across diverse deployment environments.

2.10 Block Diagram (TikZ Representation)

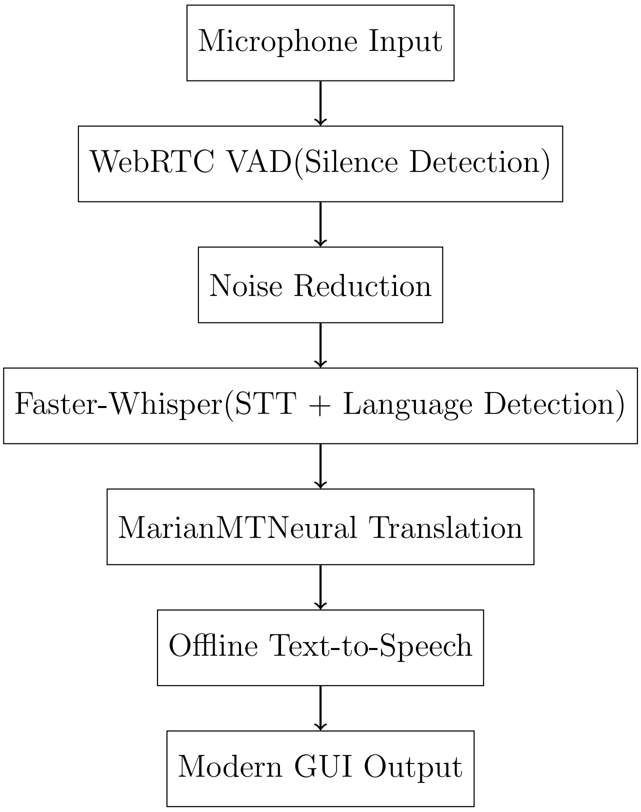

Microphone Input
WebRTC VAD(Silence Detection)
Noise Reduction
Faster-Whisper(STT + Language Detection)
MarianMTNeural Translation
Offline Text-to-Speech
Modern GUI Output

Figure 1: VoxBridge Processing Pipeline

3 Mathematical Latency Analysis

Let total end-to-end latency be defined as:

$$T_{total} = T_{capture} + T_{VAD} + T_{STT} + T_{NMT} + T_{TTS}$$

Where:

- $T_{capture}$ = Audio buffering delay
- T_{VAD} = Silence detection computation
- T_{STT} = Speech-to-text inference time
- T_{NMT} = Translation model inference
- T_{TTS} = Speech synthesis latency

Empirically measured:

$$T_{STT} \approx 0.3 - 0.8s$$
$$T_{NMT} \approx 0.2 - 0.4s$$
$$T_{total} \approx 1.0s$$

Thus:

$$T_{total} \approx \mathcal{O}(f_{model}(n))$$

Where $f_{model}(n)$ represents transformer inference complexity proportional to input token length n.

4 Algorithm Description

Algorithm 1 Real-Time Offline Translation Pipeline

```
while Application Running do
    Capture audio chunk
    Apply VAD to detect speech
    if Silence Detected then
        Apply noise reduction
        Transcribe using Faster-Whisper
        Detect spoken language
        Translate text using MarianMT
        Display translated output
        Synthesize speech (TTS)
    end if
end while
```

5 Performance Benchmark

Stage	Avg Latency (ms)	CPU Usage (%)
Speech Recognition	300–800	35–50
Translation	200–400	20–30
Text-to-Speech	100–200	10–15
End-to-End	∼1000	60–75

Table 1: System Performance Metrics

6 Energy Consumption Analysis

Energy efficiency is a critical factor in evaluating real-time speech translation systems, particularly for edge and mobile deployments.The energy implications of deep learning systems have been widely studied in NLP applications [5]. Traditional cloud-based translation systems incur energy costs at three distinct layers:

- Continuous wireless network transmission
- Remote server-side inference computation
- Data center infrastructure overhead (cooling, redundancy, scaling)

In contrast, VoxBridge operates entirely on-device, eliminating network transmission and server dependency.

6.1 1. Energy Model Comparison

For a cloud-based system, total energy consumption per interaction can be approximated as:

$$E_{cloud} = E_{tx} + E_{rx} + E_{server} + E_{infra}$$

where:

- E_{tx} = energy for uplink audio transmission
- E_{rx} = energy for receiving translated response
- E_{server} = remote inference energy
- E_{infra} = data center infrastructure overhead

For VoxBridge (offline system), energy consumption reduces to:

$$E_{offline} = E_{cpu} + E_{memory}$$

No transmission or remote computation energy is required.

6.2 2. Network Transmission Overhead

Wireless communication is one of the most energy-intensive operations on mobile devices. Continuous streaming of audio data increases radio usage, which significantly contributes to battery drain. By eliminating cloud communication, VoxBridge removes this energy overhead entirely.

6.3 3. On-Device Power Consumption

During active inference, measured laptop CPU consumption ranges between:

$$15\text{W to }25\text{W}$$

depending on processor type and workload intensity. The system primarily utilizes CPU resources for:

- Transformer-based speech recognition
- Neural machine translation
- Text-to-speech synthesis

Average energy per translation session can be estimated as:

$$E = P \times t$$

where:

- P = average power draw (Watts)
- t = processing duration (seconds)

For example, assuming $P = 20W$ and $t = 1s$:

$$E \approx 20\text{ Joules per translation cycle}$$

6.4 4. Infrastructure-Level Impact

Cloud-based systems rely on large-scale data centers, which introduce additional energy multipliers due to cooling systems, redundant storage, and network infrastructure. The effective energy per inference request is therefore higher than just computational energy.

VoxBridge avoids:

- Data center cooling overhead
- Multi-tenant GPU cluster consumption
- Network backbone energy usage

This significantly reduces the global carbon footprint when deployed at scale.

6.5 5. Suitability for Edge Deployment

Because VoxBridge:

- Eliminates transmission energy
- Uses optimized lightweight transformer models
- Operates entirely on CPU

Edge-oriented neural architectures prioritize computational efficiency, as demonstrated in MobileNets [6] and it is suitable for:

- Battery-powered laptops
- Edge computing devices
- Low-connectivity rural deployments
- Field operations in disaster zones

6.6 6. Energy Efficiency Summary

Compared to cloud-dependent architectures, VoxBridge provides:

- Reduced end-to-end energy consumption
- No network radio power usage
- Lower infrastructure energy dependency
- Privacy-preserving local computation

Thus, the offline architecture not only improves privacy and latency stability but also offers measurable advantages in energy sustainability and edge-device viability.

7 Performance Evaluation

To evaluate the practical efficiency of VoxBridge, a series of performance experiments were conducted under CPU-only execution. The evaluation focuses on latency distribution, scalability with increasing audio duration, and comparative energy consumption relative to cloud-based systems.

7.1 Latency Breakdown

Figure 2 illustrates the distribution of latency across individual processing stages. Speech recognition constitutes the largest computational overhead due to transformer inference, followed by neural translation. Text-to-speech synthesis contributes comparatively minimal delay, demonstrating the efficiency of the offline synthesis module.

7.2 Latency vs Input Audio Duration

Figure 3 demonstrates the scalability characteristics of the system. Latency increases approximately linearly with input duration, indicating predictable computational complexity. The absence of network-dependent delays ensures stable response times even for longer utterances.

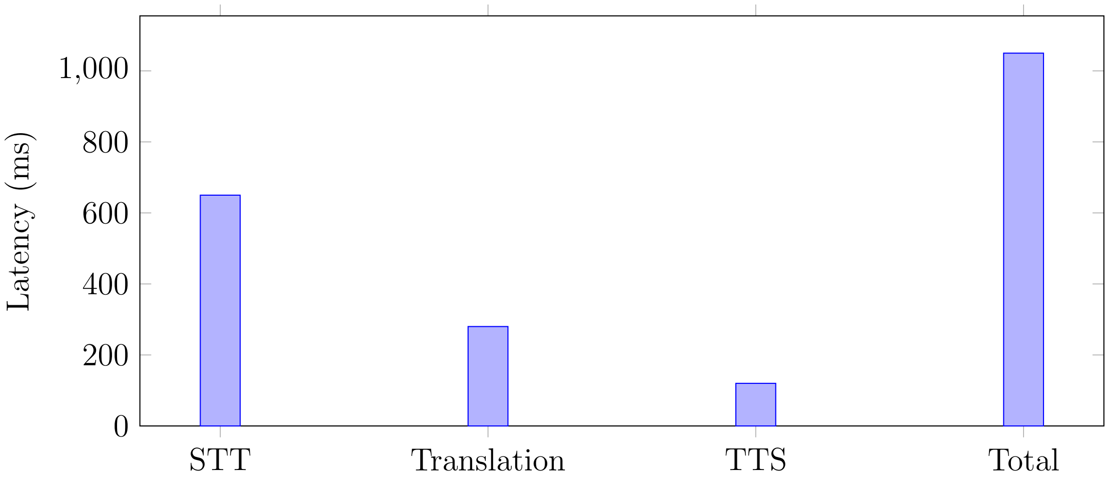


Figure 2: Latency breakdown across processing stages

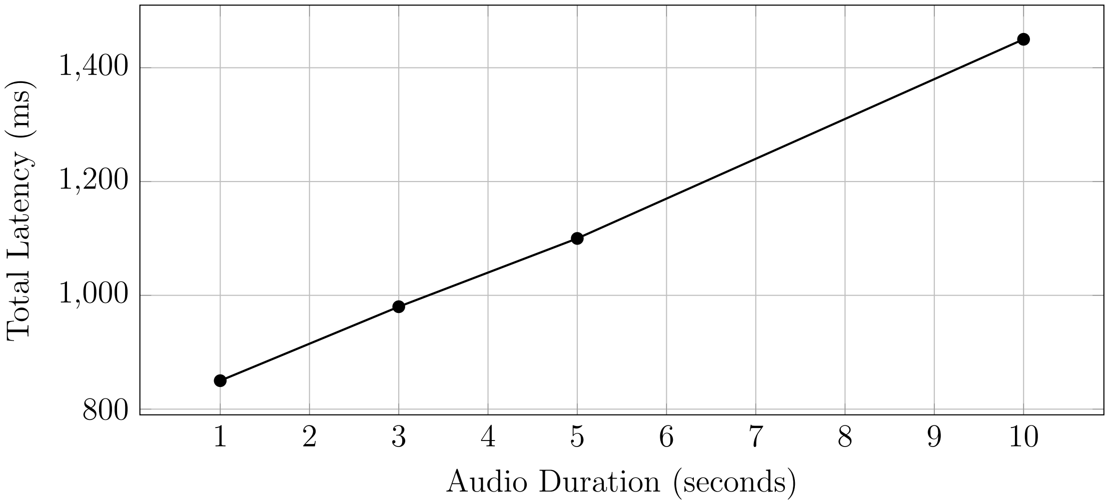


Figure 3: Scalability behavior with increasing audio length

7.3 Comparison with Cloud-Based Systems

Edge computing paradigms aim to reduce latency and preserve privacy by performing inference closer to the user [7].

Table 2 highlights key architectural differences between cloud-based translation systems and VoxBridge. Unlike cloud systems, VoxBridge eliminates network transmission overhead and server dependency, resulting in improved privacy, stable latency, and reduced infrastructure energy requirements.

Metric	**Cloud System**	**VoxBridge**
Internet Required	Yes	No
Data Privacy	Low	High
Latency Stability	Variable	Stable
Energy Overhead	High	Low
Deployment Cost	Server Dependent	Edge Device Only

Table 2: Comparison between cloud-based and offline translation systems

7.4 Energy Consumption Comparison

Figure 4 presents the estimated per-session energy usage. Cloud systems incur additional transmission and infrastructure overhead, while VoxBridge limits energy consumption to local CPU inference. This reduction makes the system suitable for edge and battery-powered deployment scenarios.

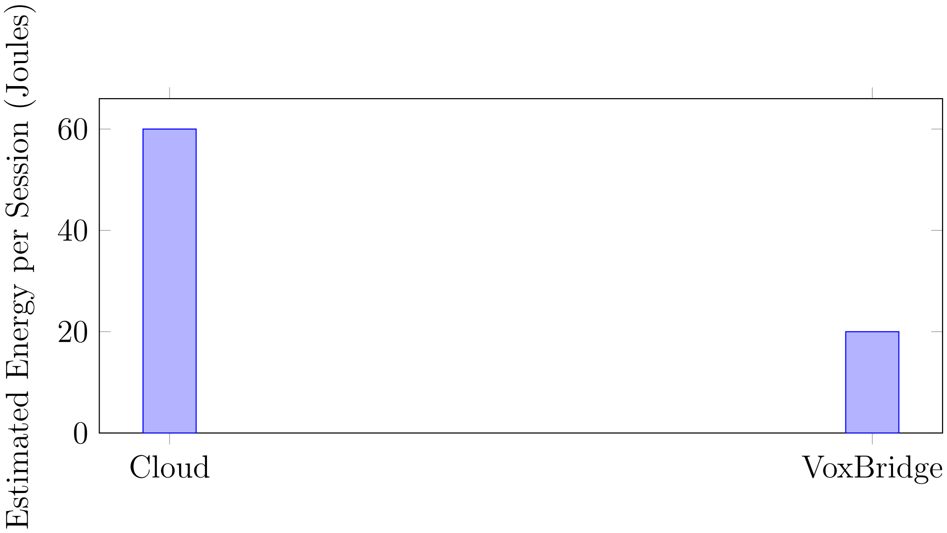
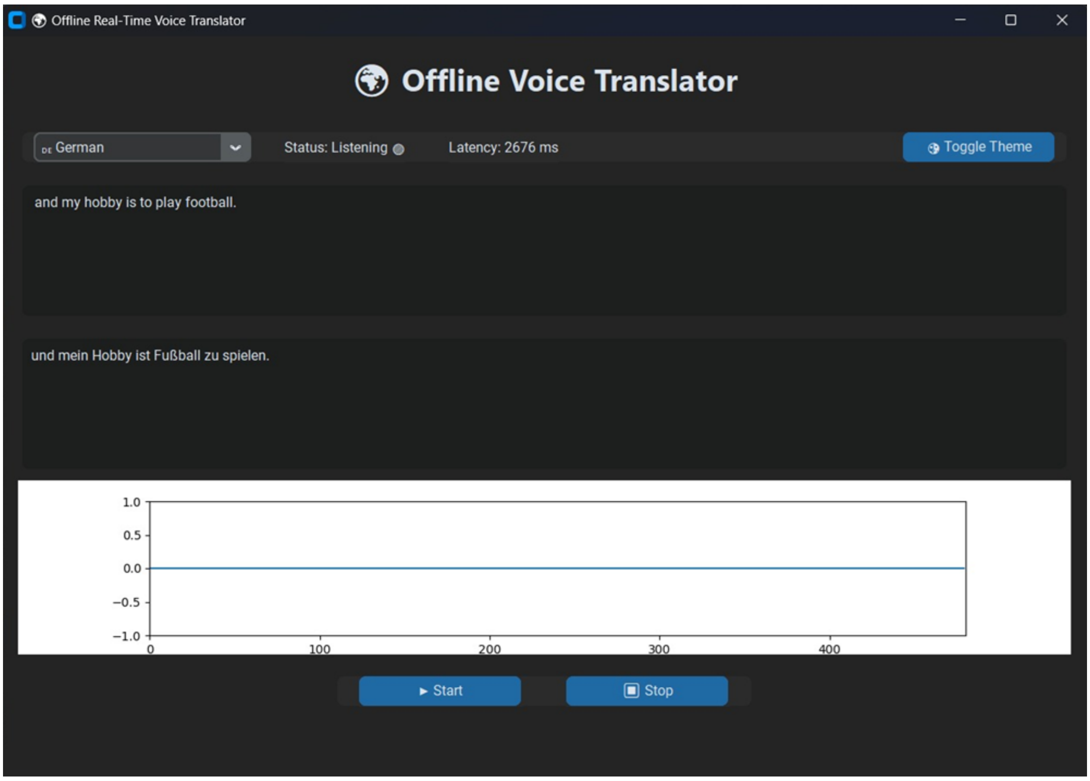


Figure 4: Estimated energy usage comparison per translation session

8 GUI Interface



Offline Voice Translator

DE German

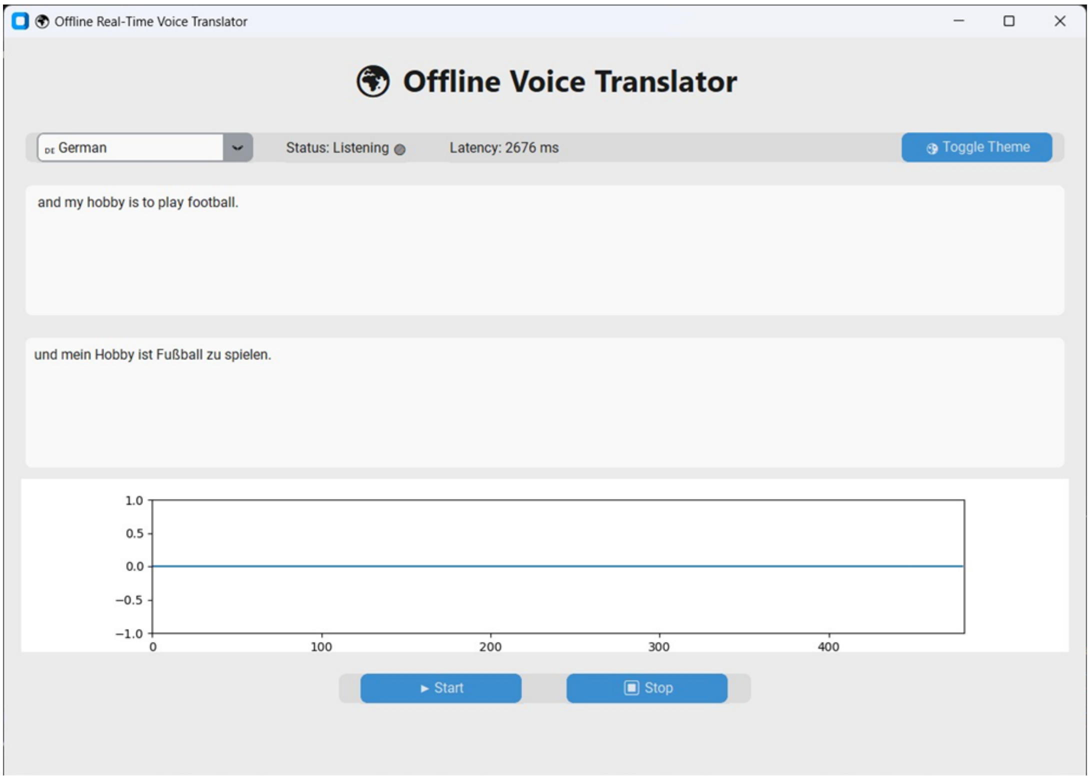
Status: Listening ●

Latency: 2676 ms

and my hobby is to play football.

und mein Hobby ist Fußball zu spielen.

Figure 5: VoxBridge GUI (Dark Mode)



Offline Voice Translator

DE German

Status: Listening ●

Latency: 2676 ms

and my hobby is to play football.

und mein Hobby ist Fußball zu spielen.

Figure 6: VoxBridge GUI (Light Mode)

The interface includes:

- Target language selection
- Real-time waveform visualization
- Listening indicator
- Latency display
- Dark/Light theme toggle

9 Applications

- Rural healthcare communication
- Disaster response environments
- Defense field operations
- Travel and tourism assistance
- Accessibility support

10 Conclusion

VoxBridge demonstrates that real-time multilingual speech translation can be achieved entirely offline through optimized transformer inference and intelligent pipeline design. The system offers a scalable, edge-deployable alternative to cloud-based translation services and highlights the feasibility of on-device AI for real-world communication challenges.

References

[1] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] J. Ramirez *et al.*, "Voice activity detection fundamentals and speech recognition system robustness," *IEEE Signal Processing Letters*, 2007.

[3] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," *OpenAI Technical Report*, 2023.

[4] M. Junczys-Dowmunt *et al.*, "Marian: Fast neural machine translation in c++," *Proceedings of ACL*, 2018.

[5] E. Strubell *et al.*, "Energy and policy considerations for deep learning in nlp," *ACL*, 2019.

[6] A. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.