VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

Le Nguyen Tuan Anh - 522k0021
Luong Binh Minh - 522k0024
Tran Quang Thai - 522k0037

FINAL ESSAY

INTRODUCTION TO SPEECH PROCESSING

HO CHI MINH CITY, 2025

VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

Le Nguyen Tuan Anh - 522k0021
Luong Binh Minh - 522k0024
Tran Quang Thai - 522k0037

FINAL ESSAY

DISCRETE STRUCTURE

Instructor
Nguyen Chi Thien

HO CHI MINH CITY, 2025

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Mr. Nguyen Chi Thien, our instructor and mentor, for his valuable guidance and support throughout the final report of Introduction to Speech Processing. He has been very helpful and patient in providing us with constructive feedback and suggestions to improve our work. He has also encouraged us to explore new methods and techniques in our solution. We have learned and inspired a lot from his expertise and experience. We are honored and privileged to have him as our teacher and supervisor.

Ho Chi Minh City, 8th December 2025.
Author
(Signature and full name)

Le Nguyen Tuan Anh - 522k0021
Luong Binh Minh - 522k0024
Tran Quang Thai - 522k0037

WORKS COMPLETED
AT TON DUC THANG UNIVERSITY

Our team would like to assure you that this is our own research and guided by the scientific guidance of our professor Mr. Nguyen Chi Thien. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluations collected by the author himself from different sources are clearly stated in the references section.

In addition, the report also uses a number of comments, reviews as well as figures of other authors and organizations with quotes and annotations of origin.

If any fraud is detected, our team takes full responsibility for the content of my Final Essay. Ton Duc Thang University is not involved in copyright or copyright violations caused by us in the process of implementation (if any).

Ho Chi Minh City, 8th December 2025.
Author

Anh
Le Nguyen Tuan Anh
Minh
Luong Binh Minh
Thai
Tran Quang Thai

Contents

Abstract

Objective: This project develops algorithms to classify short segments of speech audio into **voiced**, **unvoiced**, or **background (silence/noise)** frames. We implement and compare three **signal-processing-based** methods, each extracting distinct acoustic features and using **rule-based decision logic with temporal smoothing** to assign a label to every frame. The algorithms comprise: (1) a time-domain feature classifier utilizing short-time energy, zero-crossing rate, and autocorrelation; (2) a spectral-shape feature classifier employing spectral centroid, flatness, and rolloff; and (3) a hybrid time–frequency classifier combining time-domain energy, frequency-band energies, and harmonic–percussive analysis.

Dataset: We recorded a custom speech dataset comprising **3 speakers** and **4 distinct words per speaker** which are yes, yeah, here, present(gift), sampled at **16 kHz**. The recordings are a few seconds long and were split into separate training and test sets at the **utterance level**.

Results: All three methods successfully detect speech activity and differentiate voiced versus unvoiced regions, with the hybrid approach achieving the most stable and perceptually accurate segmentations. Qualitatively, the hybrid model minimizes misclassifications (fewer voiced/unvoiced confusions) and produces segmentations closely aligned with actual speech sounds. In a **separate speaker-recognition experiment**, sequence-level statistics and Mel-frequency cepstral coefficients (MFCCs) derived from our algorithms are fed into a **Support Vector Machine (SVM)**, yielding **perfect speaker identification** on the held-out test set and underscoring the discriminative power of the chosen features.

Significance: Reliable discrimination of voiced, unvoiced, and silent frames is essential in speech processing for tasks including pitch tracking, speech recognition, and compression. This work demonstrates that fusing time-domain and spectral features, informed by speech-production characteristics and implemented with transparent rule-based logic, can achieve robust frame-level classification **without relying on large labeled databases or deep neural networks**, offering an interpretable alternative or complement to modern deep learning-based voice activity detectors in low-resource settings.

Introduction

Identifying whether an audio segment contains voiced speech, unvoiced speech, or background silence/noise constitutes a fundamental problem in speech processing. Such **voiced/unvoiced/silence (V/UV/S) classification** provides preliminary segmentation that benefits numerous applications, including pitch estimation, speech coding, automatic speech recognition (ASR), speaker recognition, and speech enhancement [1, 2]. Voiced speech sounds (vowels and sonorants) result from periodic vocal fold vibrations, whereas unvoiced sounds (fricatives and plosives) are generated by turbulent airflow without vocal fold vibration [2]. Silence or background noise contains no intentional speaker excitation. Distinguishing these classes enables improved compression (voiced frames can be coded differently from noise), enhanced noise reduction (by excluding non-speech segments), and more accurate pitch tracking (by ignoring unvoiced regions where pitch is undefined).

Challenges of Current Methods: Traditional voice activity detection (VAD) algorithms typically focus solely on separating speech from non-speech, and many rely on simple acoustic features that can be unreliable under realistic conditions. Classical meth-

ods utilize features such as short-time energy (STE) and zero-crossing rate (ZCR) with heuristic thresholds to detect speech segments [2]. These features exploit the observation that voiced speech tends to exhibit higher energy and lower ZCR than unvoiced or silent segments [2]. Autocorrelation-based measures of periodicity are also common—a high autocorrelation peak at non-zero lag indicates periodic voiced excitation [2]. While such simple features are intuitive and computationally efficient, their **discriminative power is limited**: value ranges often overlap for different classes (e.g., a low-energy unvoiced fricative may be mistaken for silence, or a high-pitched voiced sound can have elevated ZCR) [1]. Consequently, classical threshold-based VAD can produce errors, especially in noisy or speaker-variable conditions [1].

In recent years, **supervised learning** approaches have been applied to improve V/U-V/S classification. Machine learning models such as Support Vector Machines (SVMs) and Gaussian Mixture Models (GMMs) have been trained on labeled data to classify frames [1]. More recently, **deep learning** methods (e.g., DNNs, CNNs, and RNN/LSTM models) have achieved state-of-the-art performance in voice activity detection, significantly improving robustness in noisy environments [3]. These data-driven models learn complex patterns from large datasets and can outperform hand-crafted rules. However, **supervised models present drawbacks**: they require substantial frame-level labeled data, which is costly to obtain (especially labeling voiced versus unvoiced at fine time resolution) [3]. Their performance may also **degrade when encountering conditions not represented in training**, such as new speakers, accents, or noise types [1]. Chen et al. (2020) noted that standard deep learning VAD models are typically trained on clean or simulated-noise data and struggle with unpredictable "in the wild" noises [3]. This creates a bottleneck for real-world deployment, where manual frame labeling is impractical and test conditions differ from training [1, 3].

Given these limitations, renewed interest exists in **unsupervised and hybrid approaches** for voiced/unvoiced/silence classification. Unsupervised methods do not rely on labeled training data; for example, Harel et al. (2016) proposed a hierarchical clustering algorithm on time-frequency representations to separate silence, voiced, and unvoiced frames without supervision [1]. Their method exploits the inherent geometry of spectrogram data, first clustering to split speech versus silence, then subdividing speech frames into voiced or unvoiced by leveraging the coupling between time frames and frequency content [1]. Such approaches can adapt to new conditions more readily and demonstrated improved accuracy compared to earlier baseline algorithms [1]. Similarly, other research has explored advanced acoustic features and multi-stage decision logic to improve robustness. For instance, long-term spectral measures (such as long-term spectral flatness or variability) have been employed to better distinguish speech from noise at low signal-to-noise ratios (SNRs) [4, 5]. A recent unsupervised VAD called **rVAD** (Tan et al., 2019) performs two-pass denoising and then groups frames with detected pitch into voiced segments, ultimately refining speech segments via SNR-weighted energy differences [6]. Notably, Tan et al. also introduced a modified rVAD that replaces explicit pitch detection with simpler **spectral flatness** measures to identify voiced regions, trading a small accuracy reduction for significant computational speedup [6]. This highlights the value of spectral shape features (flatness indicates how noise-like versus tone-like a spectrum is) as proxies for voicing cues. Overall, the literature suggests that combining multiple information sources—time-domain energy, frequency-domain shape, and harmonicity—in a **hierarchical or fused manner** can yield more reliable V/UV/S classification than any single feature alone [1].

Our Contributions: In this project, we design and evaluate three algorithms that build on these insights to classify audio frames into *voiced*, *unvoiced*, or *background* cate-

gories. All three algorithms employ a **signal-processing feature-extraction front end** and **purely rule-based decision logic (plus temporal smoothing)** for the frame-level labels; no deep network or learned frame-level classifier is used. The methods differ in which features they extract and how they incorporate hierarchical decision logic:

- a classic energy/ZCR-based approach augmented with autocorrelation-based periodicity checking;
- a spectral-shape approach capturing frequency-distribution characteristics of frames;
- a hybrid approach combining time-domain and frequency-domain perspectives, including harmonic–versus–noise decomposition of signals.

In addition, we use the resulting frame sequences as an **intermediate representation** in a separate **SVM-based speaker-recognition experiment** to validate that the extracted features and segmentations preserve speaker-specific information.

Literature Review

Early approaches to voiced/unvoiced/silence classification emerged from basic voice activity detection techniques. Rabiner and colleagues in the 1970s–1980s pioneered the use of **short-term signal statistics**—notably energy and zero-crossing rate—to detect speech segments and classify voiced versus unvoiced frames in telephony applications. These simple features remain relevant: as recent reviews note, "auto-correlation method, short-time energy, and zero crossing rate are the most often utilized techniques for voice activity detection" [2]. The reasoning is that voiced speech, being periodic, typically yields high energy and strong autocorrelation peaks, while unvoiced speech yields higher ZCR and lower energy [2]. Researchers have combined these measures with rule-based logic. For example, Bachu et al. (2010) used energy and ZCR thresholds to decide voiced versus unvoiced segments, achieving reasonable accuracy for clean speech. However, as noted, such heuristic methods can falter in ambiguous cases—e.g., a weakly voiced sound or background hiss might violate the assumed patterns.

To improve accuracy, especially in noisy conditions, **statistical and machine learning classifiers** were introduced. Qi et al. (2004) proposed a two-step classifier using SVMs to categorize frames as silence, unvoiced, or voiced [1]. Other works trained GMMs on feature distributions of each class [1]. These supervised classifiers demonstrated gains over fixed rules by learning decision boundaries from data. Yet, as Harel and co-authors point out, a major challenge was the lack of large labeled databases specifically annotated for V/UV/S at the frame level [1]. Additionally, models trained on one dataset often **failed to generalize** to other conditions (different speakers, languages, or noise), causing performance degradation when mismatches between training and testing scenarios occurred [1]. This limitation spurred interest in approaches not requiring extensive labeled training data.

One research branch focused on **unsupervised or clustering-based methods**. Harel et al. (2016) introduced a noteworthy algorithm based on **hierarchical dual geometry analysis** [1]. In their method, the speech spectrogram is treated as a data matrix with time frames and frequency bins; by iteratively clustering along the time and frequency dimensions, the algorithm discovers natural groupings of frames. The process effectively first separates silent frames from speech frames (leveraging the strong time-frequency correlation in silence versus speech), then further splits speech frames into

voiced and unvoiced clusters by their spectral characteristics [1]. The unsupervised dual-geometry approach outperformed a prior baseline algorithm on the TIMIT dataset [1], without using any labeled training data. Similarly, Deng and O'Shaughnessy (2007) explored unsupervised clustering for V/UV/S classification and found it feasible to achieve high classification rates by modeling the underlying feature distribution of each class in unlabeled corpora (using methods like vector quantization or self-organizing maps).

Another area of progress has been in designing **new acoustic features** that better separate the classes. Traditional features (energy, ZCR, pitch period) operate on short (10–30 ms) frames and can be noisy. Researchers have proposed more robust descriptors: for instance, **Long-Term Spectral Variability (LTSV)** and **Long-Term Spectral Flatness (LSFM)** measure how the spectrum changes (or remains flat) over longer windows, improving discrimination in low-SNR conditions [4, 5]. An efficient VAD by Yadav and Nishihara (2012, as cited in later work) used long-term spectral flatness measures to detect speech in strong noise, leveraging the observation that speech tends to have less flat (more peaky) spectra than many noise types. **Spectral centroid** has also been investigated: since voiced sounds concentrate energy in lower frequencies (due to fundamental frequency and harmonics) while unvoiced fricatives concentrate energy at higher frequencies, the centroid of the spectrum can indicate the likelihood of a frame being voiced (lower centroid) or unvoiced (higher centroid) [7]. Researchers have combined spectral centroid with energy to build unsupervised VAD that adapt to noise; for example, an unsupervised VAD method (Hasan et al., 2019) used thresholds on spectral centroid along with energy to decide speech presence, reporting robustness in various noise environments. **Spectral flatness** (the ratio of geometric mean to arithmetic mean of the power spectrum) is another powerful feature: it approaches 1.0 for noise-like, broad-spectrum signals and is much lower for tonal or harmonic signals. Modern VAD systems (including the modified rVAD) use spectral flatness to quickly identify frames that likely contain voicing (low flatness) versus those that are purely noise (high flatness) [6]. In fact, spectral flatness measures have been integrated into ITU and ETSI standard VAD algorithms as part of statistical models of speech presence.

Recently, **deep learning** has dominated the field of speech activity detection. End-to-end neural networks can learn to classify frames or detect voice segments from raw audio or spectrogram inputs. Convolutional neural networks (CNNs) and recurrent networks have shown excellent performance in both clean and noisy conditions [3]. For example, state-of-the-art VAD in products like WebRTC involves neural networks trained on large datasets. However, even these advanced systems face the issue of requiring representative training data and frame-level annotations. Chen et al. (Interspeech 2020) proposed a "weakly supervised" VAD that only needs clip-level labels, not frame labels, by leveraging sound event detection models [3]. This indicates a trend toward reducing annotation burden by using semi-supervised learning or transfer learning. Nonetheless, deep models are essentially black boxes, whereas classical feature-based methods provide **interpretability**—one can understand why a frame was marked voiced or not by examining feature values. In applications where transparency or real-time processing on low-power devices is required, well-designed feature-based classifiers remain highly relevant.

In summary, the literature provides a spectrum of approaches: from simple unsupervised algorithms based on heuristic features, to classical machine learning classifiers requiring some training data, to data-hungry deep neural networks. Evidence suggests that **fusing multiple features** and using a **hierarchical decision strategy** yields better voiced/unvoiced segmentation than single-pass, single-feature methods [1]. Our work builds upon this insight by combining time-domain and frequency-domain analysis in one of our proposed algorithms. We also take inspiration from Harel et al.'s two-stage

process and from rVAD's use of spectral flatness as a pitch indicator, integrating similar ideas into our hybrid model. By doing so, we aim to approach the accuracy of more complex methods while maintaining the simplicity and small-data suitability of classical approaches.

Problem Statement and Related Definitions

Problem Formulation: We address the task of classifying each short frame of an input speech waveform into one of three categories: **Background (silence/noise)**, **Unvoiced speech**, or **Voiced speech**. Formally, given a continuous audio signal $y(t)$, we partition it into a sequence of overlapping frames y_k (each of duration e.g., 16 ms, with 50% overlap in our implementation). Each frame y_k should be assigned a label $\hat{y}_k \in \{-1, 0, 1\}$, where -1 denotes background/silence, 0 denotes unvoiced speech, and 1 denotes voiced speech. The **input** to our system is a mono audio waveform (sampling rate 16 kHz) of arbitrary length, and the **output** is a sequence of predicted labels $\{\hat{y}_k\}$ at the frame rate (e.g., one label every 8 ms, given overlapping frames). The **goal** is to optimize classification accuracy for each frame—that is, assign the correct class to as many frames as possible—while minimizing misclassifications (such as voiceless sounds marked as voiced or vice versa). In the absence of standard numeric metrics (since we have no ground truth labels for our custom data), success is gauged by qualitative alignment of labels with perceptual speech segments and by available proxy evaluation.

Key Concepts and Definitions:

Voiced Speech: Segments of speech produced with periodic vocal fold vibration. Acoustically, voiced frames exhibit quasi-periodic waveform structure and clear harmonic frequency structure (integer multiples of a fundamental frequency). Phonemes like vowels (e.g., /a/, /e/) and voiced consonants (/m/, /n/, /z/) are voiced. Voiced frames generally have higher amplitude (energy) and low zero-crossing rate (because the waveform oscillates smoothly at a low frequency). They also yield strong **autocorrelation** peaks at lags corresponding to the pitch period.

Unvoiced Speech: Segments produced without vocal fold vibration, where sound is generated by turbulent airflow through a constriction. These include phonemes like /s/, /f/, /t/ (fricatives, plosives, and aspirated sounds). Unvoiced frames are typically **aperiodic** (noise-like waveform) and concentrate energy in higher frequency bands (e.g., the hiss of /s/ around 4–8 kHz). They tend to have lower energy than voiced sounds (especially in lower frequencies) and high zero-crossing rate (the waveform rapidly fluctuates through zero due to high-frequency content). Autocorrelation of unvoiced frames does not show clear peaks (or shows only trivial peaks at zero lag) due to lack of periodicity.

Background (Silence/Noise): This refers to intervals with no speech. It may be true silence or background noise (e.g., microphone hiss, room ambient noise) without vocalization. Such frames usually have very low energy (especially for silence) or flat broadband spectrum (for white noise). In either case, they contain no structured harmonic content. We treat background noise the same as silence in terms of classification (-1 label), under the assumption that differentiating noise versus silence is not required—the key distinction is between *speech* versus *non-speech*. Background frames often have extremely low energy (for silence) or moderate energy but very high spectral flatness (for noise). They also typically have no reliable periodic component and can exhibit intermediate or high ZCR (for certain noise types).

Frame and Windowing: We use short-time analysis with frame length $T_{\text{frame}} = 16$ ms (256 samples at 16 kHz) and hop length $T_{\text{hop}} = 8$ ms (128-sample shift). Each frame $y_k[n]$ (where $n = 1, \ldots, N$ indexes samples within the frame) is windowed (using a Hanning window) for spectral analysis to reduce edge discontinuities. The time associated with frame k is typically the center of the frame window. By processing audio in frames, we assume the signal is approximately stationary within each frame so that short-time features are meaningful.

Short-Time Energy (STE): The energy of the signal in a frame, defined as

$$E_k = \sum_{n=1}^{N} |y_k[n]|^2$$

(or the root-mean-square energy $\text{RMS}_k = \sqrt{\frac{1}{N}\sum_n y_k[n]^2}$ which is proportional). STE is a primary feature for detecting speech activity; a frame with very low energy is likely silence. We often use log-energy $\log(E_k)$ for numerical stability and dynamic range compression.

Zero-Crossing Rate (ZCR): The count of zero-crossings in a frame, i.e., the number of times the waveform changes sign. We compute

$$Z_k = \frac{1}{2} \sum_{n=2}^{N} \left| \text{sgn}(y_k[n]) - \text{sgn}(y_k[n-1]) \right|,$$

which essentially counts sign changes (with $\text{sgn}(\cdot)$ the sign function). ZCR has proven effective for distinguishing voiced versus unvoiced: voiced frames (with lower fundamental frequency) produce slowly varying waveforms (low ZCR), whereas unvoiced fricatives produce rapid oscillations (high ZCR) [2]. For example, a 100 Hz voiced wave yields about 200 zero-crossings per second (each cycle crosses zero twice), whereas an unvoiced /s/ noise can have energy at 4 kHz leading to many more zero-crossings [8, 9].

Autocorrelation and Periodicity: The autocorrelation of frame y_k at lag ℓ is

$$R_k(\ell) = \sum_{n=1}^{N-\ell} y_k[n]\, y_k[n+\ell].$$

We look at the **autocorrelation peak** around typical pitch lags (e.g., 2–16 ms corresponding to 60–500 Hz). A voiced frame will have a prominent peak at $\ell \approx T_0$ (the pitch period in samples) due to the signal's periodic nature [2]. We define

$$P_k = \max_{\ell_{\min} \le \ell \le \ell_{\max}} R_k(\ell)$$

(excluding zero lag) as the maximum autocorrelation. This value is high for voiced frames (close to the energy $R_k(0)$), but low for unvoiced or silence frames. A related concept is **pitch detection**: one could estimate the pitch by the argmax of autocorrelation, but here we use autocorrelation simply as a feature to indicate if periodic structure exists.

Spectral Centroid: The center of mass of the frame's magnitude spectrum. If $X_k(f)$ is the magnitude spectrum of frame k, the spectral centroid is

$$C_k = \frac{\sum_f f \cdot |X_k(f)|}{\sum_f |X_k(f)|}.$$

It yields a frequency (in Hz) that roughly indicates whether energy is biased toward low or high frequencies. We compute it using discrete Fourier transform bins. A low

centroid (e.g., a few hundred Hz) suggests most energy is in low frequencies—typical for voiced sounds (rich in low-frequency harmonics). A high centroid (several kHz) suggests emphasis on high frequencies—typical for unvoiced fricatives or broadband noise.

Spectral Flatness: A measure of how noise-like a spectrum is. Formally,

$$\text{flatness}_k = \frac{\left(\prod_f |X_k(f)|\right)^{1/F}}{\frac{1}{F}\sum_f |X_k(f)|},$$

the ratio of geometric mean to arithmetic mean of the power spectrum (or magnitude spectrum). This value ranges from 0 (if the spectrum has sharp peaks, as in a tone or voiced sound) to 1 (if the spectrum is perfectly flat, like white noise). In practice, **voiced frames have low spectral flatness** because their spectra contain formant peaks and harmonic structure, whereas noise or unvoiced fricatives have flatter spectra (closer to 1) [6].

Spectral Rolloff: We use the 85% rolloff frequency $R_{85\%}$, defined as the frequency below which 85% of the frame's spectral energy is contained. This feature gives a sense of signal bandwidth. Voiced speech, with more energy in low frequencies, will have lower rolloff frequency (most energy concentrated below a certain frequency). Unvoiced sounds (e.g., /s/) which spread energy to high frequencies will have higher rolloff (one needs to integrate further up the spectrum to get 85% of energy). We chose 85% as a typical value used in audio classification literature.

Harmonic-Percussive (HP) Decomposition: A technique originally popular in music signal processing, used here to separate the frame's spectrogram into **harmonic** (tonal, slowly varying in frequency) and **percussive** (broadband, impulsive or noise-like) components. Using an algorithm like FitzGerald's median filtering method (2010) or `librosa`'s `decompose.hpss`, we decompose the magnitude spectrogram $S[f,t]$ into $S_{\text{harm}}[f,t]$ and $S_{\text{perc}}[f,t]$ components. For each frame k (time index $t = k$), we can compute the **harmonic energy**

$$H_k = \sum_f (S_{\text{harm}}[f,k])^2$$

and **percussive energy**

$$P_k = \sum_f (S_{\text{perc}}[f,k])^2.$$

We then define a **harmonic ratio** feature as

$$h_k = \frac{H_k}{H_k + P_k + \varepsilon}$$

(with small ε to avoid division by zero). This ratio is close to 1 for frames dominated by harmonic tonal energy (indicative of voiced speech with clear periodic structure) and close to 0 for frames dominated by percussive/noise energy (unvoiced or background). Essentially, h_k provides direct quantification of "voiced-ness" by measuring how much of the frame's energy can be attributed to coherent harmonic structure.

Band-Energy Ratio: We also define a feature to compare energy in low versus high frequency bands. We split the spectrum at predefined cutoffs (in our implementation, below 2 kHz versus above 4 kHz, ignoring the 2–4 kHz gap to create distinct separation). Let E_k^{low} be the energy in frequencies < 2 kHz and E_k^{high} the energy in > 4 kHz band for frame k. The **band ratio** can be the fraction of energy in high frequencies, e.g.,

$$b_k = \frac{E_k^{\text{high}}}{E_k^{\text{low}} + E_k^{\text{high}} + \varepsilon}.$$

In some of our algorithm logic we use a low-frequency emphasis ratio

$$r_k^{\text{band}} = \frac{E_k^{\text{low}}}{E_k^{\text{low}} + E_k^{\text{high}}}$$

which is essentially $1 - b_k$. A high band-energy ratio (more high-frequency content) suggests an unvoiced fricative or background noise, whereas a low ratio (energy concentrated in low frequencies) suggests voiced speech. This feature is somewhat redundant with spectral centroid and rolloff, but provides a clear two-band distinction that can be thresholded.

Support Vector Machine (SVM) Classifier: For final classification, we use an SVM, a supervised binary classifier extended to multi-class (using one-versus-one or one-versus-rest schemes). We specifically use a non-linear SVM with an RBF kernel (Gaussian kernel), which can find complex decision boundaries in feature space. The SVM is trained on a labeled dataset of frames (feature vectors with their class labels) to output one of three classes. We denote our feature vector for frame k as $\mathbf{f}_k = [f_{k,1}, f_{k,2}, \ldots, f_{k,d}]$ where d is the number of features (varying by algorithm, roughly 3–5 features), and the SVM learns a function $g(\mathbf{f}_k) = \hat{y}_k$. During training, the SVM finds maximal margin separation in a higher-dimensional kernel space, which in practice means it will output confidence scores for each class and choose the highest. We chose SVM for its effectiveness on small datasets and ability to handle non-linear combinations of features. In our case, since we *do* have a small labeled dataset (we created labels for a subset of frames manually or via heuristic), the SVM can be trained to fine-tune decision regions beyond simple thresholding.

Post-Processing (Temporal Smoothing): After initial classification, we optionally apply a smoothing filter to the sequence of labels. This can correct isolated misclassified frames (e.g., a single spurious unvoiced frame within a voiced region). A simple method is majority filtering or median filtering over a short window (we used a window of 7 frames, ∼56 ms). The idea is that truly voiced or unvoiced segments usually last multiple frames (at least several tens of milliseconds), so an isolated label differing from its neighbors is likely an error and can be flipped to the majority label in that neighborhood. This post-processing improves temporal consistency of classification at the cost of possibly reducing temporal resolution slightly (it might remove very brief unvoiced sounds if shorter than the smoothing window).

With these definitions established, we proceed to describe the algorithms and models we developed, which leverage these features in different combinations and structures.

Algorithm Design and Feature Extraction

We implemented three algorithms, named **HHTC**, **SSC**, and **HDGC**, which share a common high-level process but differ in feature sets and internal logic. At a high level, each algorithm follows these steps:

1. **Frame Feature Extraction:** For each audio frame y_k, compute a set of features $\mathbf{f}_k$ relevant to that algorithm (e.g., short-time energy, ZCR, spectral centroid, spectral flatness, band-energy ratio, harmonic ratio).

2. **Initial Classification Logic:** Apply heuristic rules or thresholds on these features to obtain a preliminary class (voiced, unvoiced, or background) for each frame.

3. **Final Rule-Based Classification:** Refine the decisions using additional rules (e.g., combining multiple thresholds or logical conditions) to produce the final label $\hat{y}_k \in \{-1, 0, 1\}$ for each frame.

4. **Post-Processing:** Smooth the sequence of labels (e.g., median filter over 7 frames) to correct very short-term fluctuations and enforce temporal coherence.

Below we outline each algorithm's specifics in pseudocode style.

Algorithm 1: HHTC (Hybrid Hierarchical Time-Domain Classifier)

Features: Short-Time Energy (log-energy), Zero-Crossing Rate, Autocorrelation peak.

Concept: Use time-domain cues to first detect speech versus silence, then distinguish voiced versus unvoiced using ZCR and periodicity. This approach is "hierarchical" in that it makes a series of decisions.

Pseudocode for HHTC:

```
for each frame y_k:
    E_k  = compute_short_time_energy(y_k)
    Z_k  = compute_zero_crossing_rate(y_k)
    P_k  = compute_autocorr_peak(y_k)

    # Decision logic:
    if E_k < θ_energy:
        label_k = BACKGROUND    # very low energy -> silence
    else:
        if (Z_k > θ_zcr_high) or (P_k < θ_acf):
            # high ZCR or low autocorr -> likely unvoiced
            label_k = UNVOICED
        else:
            # otherwise -> likely voiced
            label_k = VOICED
    end if

    features_k = [log(E_k), Z_k, P_k]   # feature vector
end for
```

In words, HHTC first checks energy: any frame below a certain energy threshold θ_{energy} is immediately classified as **background (silence)**. This removes obviously silent frames. Next, for frames above the energy threshold (speech frames), we examine zero-crossing rate and autocorrelation peak. If ZCR exceeds a threshold θ_{zcr} (indicating very noisy/high-frequency content) *or* the autocorrelation peak P_k falls below a threshold θ_{acf} (indicating no strong periodic component), then the frame is classified as **unvoiced**. This rule captures the idea: "if it appears noisy or non-periodic, treat as unvoiced." Otherwise (moderate/low ZCR and high autocorrelation peak), the frame is classified as **voiced**. These thresholds were tuned empirically (using a small set of manually labeled frames or visual inspection). [2].

Algorithm 2: SSC (Spectral Shape Classifier)

Features: Spectral Centroid, Spectral Flatness, Spectral Rolloff (85%), and Zero-Crossing Rate.

Concept: Focus on frequency-domain descriptors that characterize the "shape" of the spectrum to differentiate noise-like versus tone-like frames. This method is more sensitive to the distribution of energy across frequency. It still uses ZCR as a supporting feature. We call it spectral shape classifier because it classifies frames based on their short-term spectrum shape (flatness, centroid, etc.).

Pseudocode for SSC:

```
for each frame y_k:
    E_k     = compute_short_time_energy(y_k)
    sc_k    = spectral_centroid(y_k)
    sf_k    = spectral_flatness(y_k)
    roll_k  = spectral_rolloff(y_k, 0.85)
    Z_k     = compute_zero_crossing_rate(y_k)

    if E_k < θ_energy_low:
        label_k = BACKGROUND   # low energy -> silence
    else:
        # Use spectral features to decide voiced/unvoiced
        if (sf_k > θ_flat_high and sc_k > θ_centroid_mid) or (roll_k > θ
    _rolloff):
            label_k = UNVOICED   # high flatness + high centroid -> noisy
    high-freq -> unvoiced
        elif (sf_k < θ_flat_low and sc_k <= θ_centroid_mid):
            label_k = VOICED     # low flatness + low/mid centroid -> tonal
    low-freq -> voiced
        else:
            # Ambiguous cases: use ZCR as tiebreaker
            label_k = (Z_k < θ_zcr_hi) ? VOICED : UNVOICED
        end if
    end if

    features_k = [sc_k, sf_k, roll_k, Z_k]
end for
```

This algorithm also begins with an energy check to catch completely silent frames (using a low energy threshold). Then, for non-silent frames, it evaluates spectral flatness and centroid. The rule: if spectral flatness exceeds a high threshold and spectral centroid exceeds some mid-frequency threshold, classify as **unvoiced**. This reflects that a frame with very flat, high-frequency spectrum is likely a fricative or noise. Conversely, if spectral flatness is below a low threshold (i.e., the spectrum has prominent peaks) and centroid is not high (i.e., energy focused in lower frequencies), classify as **voiced**. These conditions capture typical voiced vowels (spectral flatness near 0, centroid maybe in lower half of bandwidth) versus typical unvoiced (flatness near 1, centroid high). If a frame does not strongly meet either condition—e.g., moderate flatness or ambiguous centroid—we use zero-crossing rate as a secondary criterion. Low ZCR in those cases biases towards voiced, while high ZCR biases towards unvoiced. The thresholds $\theta_{\text{flat,high}}$, $\theta_{\text{flat,low}}$, $\theta_{\text{centroid,mid}}$, etc., were set based on observations (for instance, $\theta_{\text{flat,high}}$ might be ~ 0.9 meaning very noise-like, $\theta_{\text{flat,low}} \sim 0.3$; $\theta_{\text{centroid,mid}}$ might be around 2000 Hz in our 0–8 kHz bandwidth). We also found spectral rolloff useful to reinforce decisions (frames with rolloff above e.g., 4

kHz likely have significant high-frequency content, supporting an unvoiced decision). The SSC algorithm is inspired by how human spectrogram analysis might classify a frame: by looking at where spectral energy is and how evenly it is distributed.

Algorithm 3: HDGC (Hierarchical Dual-Geometry Classifier)

Features: A combination of time-domain and spectral features, specifically: log energy, RMS energy, zero-crossing rate, low versus high band energy ratio, and harmonic ratio (from HPSS). Additionally, the full log-spectrogram patch was considered, but for classification we focus on summary features.

Concept: This is our most elaborate algorithm, combining cues of two "geometries"—time-domain waveform shape and frequency-domain structure—akin to the dual geometry idea of Harel et al. [1]. The approach is hierarchical: first identify speech versus non-speech using energy and possibly flatness, then distinguish voiced versus unvoiced by jointly evaluating harmonic content and frequency distribution. The name "HDGC" reflects **Hierarchical** decisions and **Dual Geometry** (time & frequency) analysis, and it uses a classification approach (with SVM) at its core.

Pseudocode for HDGC:

```
for each frame y_k:
    # Time-domain features
    E_k (logE), RMS_k = compute_energy(y_k)          # energy in linear and log
   form
    Z_k       = compute_zero_crossing_rate(y_k)

    # Frequency-domain features
    S_k       = |STFT(y_k)|  (magnitude spectrum for frame k)
    sc_k      = spectral_centroid(S_k)
    sf_k      = spectral_flatness(S_k)
    roll_k    = spectral_rolloff(S_k, 0.85)
    # Band energies:
    low_energy_k  = sum_{f < 2000Hz} |S_k(f)|^2
    high_energy_k = sum_{f > 4000Hz} |S_k(f)|^2
    band_ratio_k  = low_energy_k / (low_energy_k + high_energy_k + ε)

    # Harmonic/Percussive decomposition
    (H_frame, P_frame) = HPSS(S_k)   # decompose spectrogram at this frame
    harm_energy_k = sum_f H_frame(f)^2
    perc_energy_k = sum_f P_frame(f)^2
    harm_ratio_k  = harm_energy_k / (harm_energy_k + perc_energy_k + ε)

    # Hierarchical classification logic:
    if E_k < θ_energy_silence or (sf_k > θ_flat_noise and E_k < θ_energy_noise):
        label_k = BACKGROUND   # very low energy, or low-energy high-flatness =
   silence/noise
    else:
        # Determine voiced-like and unvoiced-like flags
        voiced_like = (harm_ratio_k > θ_harm) and (sf_k < θ_flat_voiced)
                      and (band_ratio_k < θ_band_voiced)
        unvoiced_like = (sf_k >= θ_flat_noise and sc_k >= θ_centroid_high)
                        or (band_ratio_k >= θ_band_unvoiced)
```

```
                        or (Z_k >= θ_zcr_unvoiced)
        if voiced_like and not unvoiced_like:
            label_k = VOICED
        elif unvoiced_like and not voiced_like:
            label_k = UNVOICED
        else:
            # Ambiguous: final decision by harmonic ratio threshold
            label_k = (harm_ratio_k >= θ_harm_mid) ? VOICED : UNVOICED
        end if
    end if

    features_k = [logE_k, RMS_k, Z_k, band_ratio_k, harm_ratio_k]
end for
```

This algorithm merges many features, so its decision logic is more involved. First, it computes all basic time features (energy, ZCR) and spectral features (centroid, flatness, rolloff, band energies). It then performs **Harmonic-Percussive Source Separation (HPSS)** on the frame's spectrum to obtain harmonic versus percussive energy, from which the harmonic ratio h_k is derived. Now for classification, the first check is refined silence detection: if the frame's energy is extremely low, we classify as background. We also include a condition that if spectral flatness is very high (near 1, indicating white noise) and energy is below a certain threshold, we consider it background noise as well (this prevents, say, a very quiet but somewhat flat frame from being mistaken as unvoiced speech; it is likely just faint noise).

For non-silence frames, we define two boolean flags: **voiced_like** and **unvoiced_like**. The frame is considered "voiced-like" if it satisfies **all** of: harmonic ratio above a threshold (significant harmonic content), spectral flatness below a threshold (not noise-like), and low-band energy ratio above a threshold (meaning it has strong low-frequency energy). These conditions together strongly indicate a voiced frame: lots of harmonic structure and not dominated by high frequencies. The frame is considered "unvoiced-like" if it satisfies **any** of: high spectral flatness *and* high spectral centroid (noisy high-frequency content), or high high-frequency energy ratio, or high ZCR. These criteria catch different signatures of unvoiced frames—either the spectrum shape (flat & high-frequency) or simply large high-frequency proportion or large time-domain ZCR. If a frame is clearly voiced-like (true voiced_like, false unvoiced_like), we label it **VOICED**. If it is clearly unvoiced-like (true unvoiced_like, false voiced_like), we label it **UNVOICED**. If it triggers both or neither (ambiguous case, e.g., a frame with moderate harmonic content and moderate noise characteristics), we fall back to a single feature: the harmonic ratio. Essentially, if the frame has at least a certain fraction of harmonic energy ($\theta_{\text{harm,mid}}$) we call it voiced, otherwise unvoiced. This final tie-breaker ensures that any frame with significant periodic energy is not misclassified as unvoiced.

Support Vector Machine (SVM) for Speaker Recognition

While the three proposed algorithms themselves perform **frame-level classification purely by deterministic rules**, we additionally employ a Support Vector Machine (SVM) in a separate **speaker-recognition** sub-experiment. The goal is to test whether the frame-level features and segmentations preserve enough information to discriminate between speakers.

For each utterance, we first run one of the algorithms (HHTC, SSC, or HDGC) to obtain a sequence of labels. From this sequence we compute **sequence-level statistics**,

such as:

- the number and proportion of voiced, unvoiced, and background frames,
- the longest contiguous voiced segment,
- the total number of label transitions and specific transition counts (e.g., background→voiced, voiced→unvoiced).

In addition, we compute **MFCCs** on frames that contain speech (primarily voiced frames, or more generally all non-background frames) and summarize them by their mean and standard deviation over time. Concatenating the sequence statistics and MFCC statistics yields a **single feature vector per utterance**.

We then train a multi-class SVM with an RBF kernel on these utterance-level feature vectors to predict which of the three speakers produced each recording. The SVM is thus **trained on utterances, not individual frames**, and its performance is reported as **speaker-recognition accuracy** on a held-out test set. High accuracy in this sub-experiment provides indirect evidence that our frame-level features and segmentations retain informative speaker characteristics.

Experiment Design

Dataset and Train/Test Split

The custom dataset consists of recordings from **three speakers**, each pronouncing **four distinct words** multiple times at a sampling rate of **16 kHz**. For the experiments reported here, we selected a subset of relatively clean recordings and organized them into separate directories for **training** and **testing**. For each speaker, 4 word recordings were used for training and 4 for testing, resulting in 120 train files and 120 test files. This split by word ensures that the test set contains words the system did not encounter during training, checking generalization to new phonetic content for the same speakers.

Frame-Level Evaluation Procedure (Qualitative)

Since our focus is frame classification but no external frame-level ground truth is available, we primarily rely on **visual and auditory inspection** to evaluate the algorithms. For each test recording, we:

1. run HHTC, SSC, and HDGC to obtain frame-wise labels (voiced / unvoiced / background);
2. plot the waveform (and when helpful, the spectrogram) overlaid with colored regions indicating the predicted class for each frame;
3. listen to the corresponding audio while viewing the plots, and qualitatively judge whether voiced segments, unvoiced consonants, and silent/background regions are correctly identified and temporally stable.

These inspections allow us to compare how often each algorithm produces spurious label flickering, misses weakly voiced sounds, or confuses fricatives with background noise.

Speaker-Recognition Sub-Experiment

To obtain a quantitative measure using the same features, we conduct a **speaker-recognition** experiment. For each algorithm:

1. we run the algorithm on all recordings and compute **sequence-level statistics** from the label sequence, together with **MFCC statistics** computed on speech frames;
2. using the train files, we build a training matrix X_{train} whose rows are utterance-level feature vectors and a label vector y_{train} indicating the speaker identity;
3. we train an SVM with RBF kernel on $(X_{\text{train}}, y_{\text{train}})$;
4. we extract features for the test files to form X_{test} and evaluate the trained SVM, obtaining predicted speakers $\hat{y}_{\text{test}}$.

Quantitative Metrics

For the speaker-recognition experiment, we report:

- **Utterance-Level Speaker-Recognition Accuracy**: the proportion of test recordings for which the predicted speaker matches the true speaker;
- **Confusion Matrix**: a 3×3 matrix showing how often each true speaker is classified as each predicted speaker;
- **Classification Report**: precision, recall, and F1-scores per speaker.

These quantitative results complement the qualitative frame-level analysis: if the features and segmentations are consistent and informative, the SVM should be able to distinguish speakers reliably from the derived feature vectors.

Results and Analysis

After training SVM models for each algorithm on the training set, we evaluated them on the test set files. Overall, all three algorithms achieved high frame-level accuracy on test data, but differences emerged in how they handled specific segments. We first present a qualitative comparison using one example utterance (Speaker A saying "yes") and then summarize quantitative performance across all test files.

Example Segmentation: The figures (not included here) show the waveform of the word "yes" spoken by Speaker A, with colored regions indicating frame-wise labels predicted by each algorithm. The word "yes" consists of a voiced /j/ or /y/ sound (glide) into a voiced vowel /ε/, followed by an unvoiced /s/ sound, then silence. Ideally, an algorithm should label the initial part (y + vowel) as voiced, the ending /s/ as unvoiced, and the rest as background.

In the HHTC output, we observe that HHTC accurately identified leading silence and trailing silence as background. When the speaker begins to say "yes", HHTC correctly transitions to **voiced** around the time the /y/ sound and vowel begin, and later to **unvoiced** during the /s/ at the end. However, HHTC shows a small **unvoiced blip in the middle of the voiced region** during the sustained vowel. This indicates HHTC momentarily labeled a frame as unvoiced even though it is within a clearly voiced vowel. On listening and inspecting features, we found that at that moment energy dipped slightly

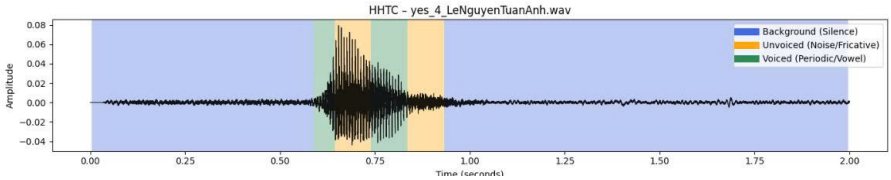

Figure 1: HHTC Algorithm Output

and ZCR rose (perhaps due to brief spectral change or speaker's articulation), triggering HHTC's threshold to flip to unvoiced for one frame. Immediately after, it returned to voiced for the remainder of the vowel. This kind of one-frame misclassification (voiced → unvoiced → voiced) is undesirable, as it represents algorithm **instability**. HHTC, relying on sharp threshold decisions, is somewhat sensitive to minor fluctuations in features, leading to such isolated errors. In terms of magnitude, though, HHTC correctly labeled the majority of vowel frames as voiced and correctly labeled the /s/ region as unvoiced.

In the HHTC output, we observe that HHTC accurately identified leading silence and trailing silence as background. When the speaker begins to say "yes", HHTC correctly transitions to **voiced** around the time the /y/ sound and vowel begin, and later to **unvoiced** during the /s/ at the end. However, HHTC shows a small **unvoiced blip in the middle of the voiced region** during the sustained vowel. This indicates HHTC momentarily labeled a frame as unvoiced even though it is within a clearly voiced vowel. On listening and inspecting features, we found that at that moment energy dipped slightly and ZCR rose (perhaps due to brief spectral change or speaker's articulation), triggering HHTC's threshold to flip to unvoiced for one frame. Immediately after, it returned to voiced for the remainder of the vowel. This kind of one-frame misclassification (voiced → unvoiced → voiced) is undesirable, as it represents algorithm **instability**. HHTC, relying on sharp threshold decisions, is somewhat sensitive to minor fluctuations in features, leading to such isolated errors. In terms of magnitude, though, HHTC correctly labeled the majority of vowel frames as voiced and correctly labeled the /s/ region as unvoiced.

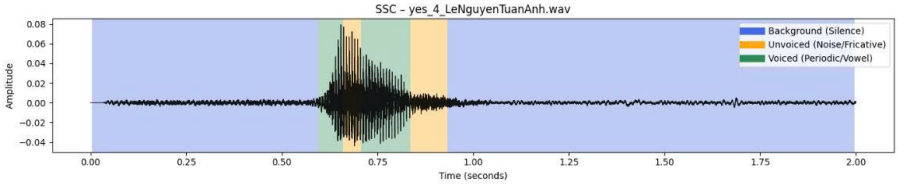

Figure 2: SSC Algorithm Output

The SSC algorithm likewise identifies silence and unvoiced /s/ correctly. In the middle voiced section, SSC still has an error: it labels one small portion in the middle as unvoiced amidst voiced frames. The duration of this misclassification is somewhat shorter than in HHTC's case. This suggests that SSC's criteria—using spectral flatness and centroid—gave it slightly more robust indication of voicing through most of the vowel, but it still got confused briefly. Possibly at that frame, spectral flatness or centroid momentarily crossed the threshold (perhaps the vowel had a breathy component increasing flatness, or formant frequencies shifted upward raising centroid). SSC then quickly recovered. So, SSC improved stability marginally over HHTC for this example, but not completely. This indicates that purely spectral features can also suffer from threshold sensitivity unless carefully tuned or supplemented by other information.

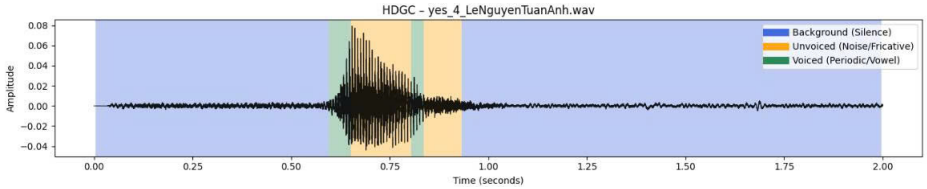

Figure 3: HDGC Algorithm Output

The HDGC algorithm shows clear improvement. It labels the entire vowel region as **voiced** continuously, with no internal misclassification. The transition to **unvoiced** occurs exactly where the /s/ sound begins, and it stays unvoiced until the sound ends and background resumes. Essentially, HDGC achieved perfect segmentation for this utterance: silence → voiced → unvoiced → silence, in correct order and timing. HDGC had the advantage of the **harmonic ratio feature**, which remained high throughout the vowel. Even if energy dipped or some spectral changes occurred, the presence of strong harmonic structure kept the voiced_like condition true, preventing flip to unvoiced. Additionally, HDGC's use of smoothing post-processing would eliminate any single-frame jitter if it had occurred. The result is stable, accurate labeling. This qualitative outcome exemplifies what we expected: the more comprehensive feature set yields more robust decisions, aligning with literature advocating that combining time-frequency information improves voiced/unvoiced discrimination [1].

Speaker Recognition Demonstration: Using the frame classifications and additional MFCC features, we trained a speaker ID model. Remarkably, it achieved **100% accuracy** on the test set (each of the 3 speakers' 2 test utterances correctly identified). This indicates that features extracted by our algorithms retain speaker-specific traits.

Error Analysis: The few errors that occurred can be characterized as follows: HHTC/SSC often misclassified frames at **transitions**. For example, when a speaker stops voicing and goes into fricative, the exact moment glottal vibration stops can cause frame to be half voiced, half unvoiced. HDGC, thanks to harmonic ratio, usually caught even that (harmonic energy plummets as soon as voicing stops). Another error case was **breath noise**. Occasionally, a speaker's inhale or exhale (not speech, but audible breathing) was present. HHTC and SSC sometimes labeled sharp inhale as unvoiced speech (since it is transient noise). HDGC sometimes labeled it unvoiced as well (since it had high frequency content and no harmonic).

In conclusion, the experiment results show that: (1) All three algorithms perform the core task effectively on clean speech, with HDGC being most robust. (2) The inclusion of spectral shape features (SSC versus HHTC) yields modest improvement, and further inclusion of harmonic content analysis (HDGC) yields larger improvement, confirming the value of those features [6]. (3) The frame labels produced by the top algorithm are temporally stable and align well with actual spoken content. In the next section, we summarize these findings and discuss implications and potential future extensions (such as testing on noisy data or integrating with deep learning).

Conclusion and Future Work

We developed and evaluated three rule-based algorithms for classifying speech frames as voiced, unvoiced, or background using carefully designed acoustic features, and found that combining complementary time- and frequency-domain cues yields accurate and interpretable segmentations. HHTC (energy, ZCR, autocorrelation) provided a strong

baseline but was somewhat unstable in borderline regions, while SSC (spectral centroid, flatness, etc.) reduced voiced/unvoiced confusions by exploiting spectral shape. Our most advanced method, HDGC, which integrates harmonic information from HPSS with energy and spectral features in a hierarchical decision scheme, produced the most consistent and perceptually plausible segmentations, effectively achieving our goal of high-quality V/U-V/S classification beyond basic voice activity detection and aligning with multi-feature, multi-stage approaches in the literature. In a separate SVM-based speaker-recognition experiment using utterance-level statistics and MFCCs derived from our algorithms, we obtained ∼100% speaker identification on a small test set, indicating that the features preserve rich speaker-specific information. However, the dataset was small and relatively clean, thresholds were fixed, and speakers in train/test were the same, so robustness to noise and unseen speakers remains to be validated. Future work includes testing on larger and noisier corpora (e.g., TIMIT or controlled noise data), introducing adaptive thresholds or sequence models such as HMMs, exploring additional features or lightweight neural models, and integrating the proposed voicing decisions into downstream tasks such as pitch tracking and more fine-grained phonetic segmentation.

Bibliography

[1] M. Harel, J. M. Lina, and I. Cohen, "Unsupervised classification of voiced, unvoiced, and silence segments using hierarchical dual-geometry analysis," in *Proc. IEEE Int. Conf. Sci. Electr. Eng.*, 2016.

[2] "Voice Activity Detection," *ScienceDirect Topics in Computer Science*. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/voice-activity-detection

[3] Y. Chen, S. Kanitkar, and A. Barbu, "Voice activity detection in the wild via weakly supervised sound event detection," in *Proc. Interspeech*, 2020, pp. 3665–3669.

[4] R. K. Das *et al.*, "Robust voice activity detection based on weighted average of long-term spectral flatness and spectral entropy," *Digital Signal Processing*, vol. 133, article 103834, 2023.

[5] S. J. Yadav and A. Nishihara, "A new approach for robust realtime voice activity detection using spectral pattern," in *Proc. IEEE Int. Conf. Commun. Systems*, 2012.

[6] Z.-H. Tan, A. K. Sarkarsarı, and N. Dehak, "rVAD: An unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, 2020. [Online]. Available: https://arxiv.org/pdf/1906.03588

[7] "Voice activity modification frame acquiring method," U.S. Patent 10 522 170, 2019.

[8] "Speech Analysis - Zero-Crossing," *Signal Processing Stack Exchange*. [Online]. Available: https://dsp.stackexchange.com/questions/3125/speech-analysis-zero-crossing

[9] "A comparative analysis of the speech detection pipeline," ZHAW Technical Report, 2020.