IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2022–10–11.
Submitted on 2022–09–29, accepted on 2022–09–29.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.427

A Presentation and Short Discussion of rVAD-fast, a Fast Voice Activity Detector

Sam Perochon

Université Paris-Saclay, ENS Paris-Saclay, Centre Borelli, France
sam.perochon@ens-paris-saclay.fr

Communicated by Jean-Michel Morel *Demo edited by* Sam Perochon

Abstract

Voice activity detection (VAD) usually refers to the detection of human voices in acoustic signals and is often used as a pre-processing step in numerous audio signal processing tasks. The unsupervised method proposed here was originally developed by Zheng-Hua Tan, Achintya kr. Sarkar and Najim Dehak [Computer Speech & Language, 2020] and consists of a robust segment-based approach. The voice activity detection stage follows two denoising steps. The first one detects high energy segments using a posteriori SNR weighted energy difference, and the second enhances the speech using the MSNE-mod approach. Use cases or downstream tasks include intrusion detection, speech-to-text, speaker diarization, or emotion estimation.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the README.md file of the archive. The rVAD authors' original implementation substantially inspired the provided code[2].
This is an MLBriefs article, the source code has not been reviewed!

Keywords: voice activity detection; a posteriori SNR; spectral flatness; speech enhancement; speaker verification; audio processing

[1] https://doi.org/10.5201/ipol.2022.427

[2] https://github.com/zhenghuatan/rVAD, commit hash: d41f5354317bf13c1d8b31cb4f7ad4bf5112cd34

Sam Perochon, *A Presentation and Short Discussion of rVAD-fast, a Fast Voice Activity Detector*, Image Processing On Line, 12 (2022), pp. 404–419. https://doi.org/10.5201/ipol.2022.427

1 Introduction

Voice activity detection (VAD), also called speech activity detection (SAD), is widely used in real-world speech systems to discard the non-speech part of a signal to reduce the computational cost of downstream processing. The objective is to detect the presence or absence of speech in an audio signal. In downstream tasks, the segments without speech can be discarded to save the device resources. For instance, voice activity detection can precede other tasks such as speech identification, speech separation, speaker recognition or diarization, or emotion estimation. Although much progress has been made in VAD, developing VAD methods that are accurate in both clean and noisy environments and can generalize well under unseen environments is still an unsolved problem. Supervised methods for VAD either train statistical models for speech and non-speech or frame the task as a classification problem, which might require a large amount of data. Although these methods often outperform unsupervised ones, they depend on the quality of the labeled training data, which influences the target domain distribution, preventing them from generalizing to new unseen data. The method proposed here is fully unsupervised and relies on the computation of the *a posteriori signal to noise weighted energy difference* and the spectral flatness of the signal [3, 2]. The former is a metric used for denoising, and the latter is a spectral information used as a proxy for whether a specific segment contains pitch, a fundamental component of speech. A threshold-based stage uses these two quantities to decide on voice activity detection. According to the authors, the advantages of the method compared to other unsupervised approaches are the following:

- A two-stage denoising is proposed to enhance the noise robustness against both rapidly changing, impulsive, and stationary noise.
- The spectral flatness (used as a proxy for pitch) is computed to detect high-energy noise segments and as an anchor to find potential speech segments.
- The VAD stage is conducted in segments, making it easier and more effective to determine a threshold for making decisions since each segment has a certain amount of speech and non-speech.

2 Method

Figure 1 illustrates the different steps of the method. A detailed description of the different operations employed in the algorithms follows. The different steps of the methods are as follows.

1. Pre-processing of the audio signal (e.g. .wav, .mp3).
   - Application of a digital high-pass filter to the loaded signal.
   - Partitioning of the signal into short overlapping frames.
   - Application of a Hamming window to each frame.
2. Identification of the segments containing pitch.
   - Spectral flatness computation as a proxy for pitch in each frame.
   - Classification of each frame as containing pitch or not.
   - Post-processing of the binary pitch mask.
3. First-pass denoising.

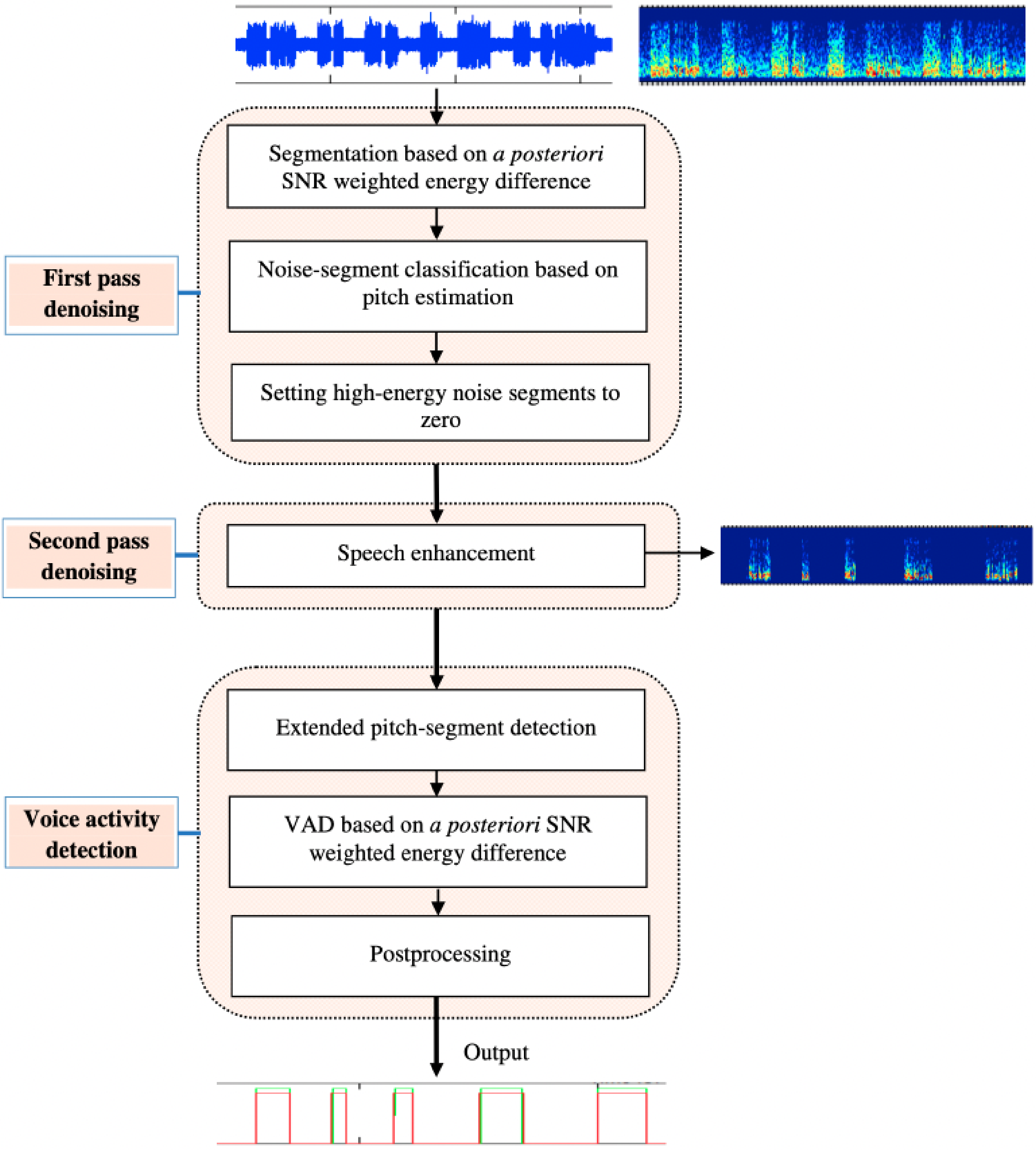

Segmentation based on *a posteriori* SNR weighted energy difference
Noise-segment classification based on pitch estimation
Setting high-energy noise segments to zero
First pass denoising
Second pass denoising
Speech enhancement
Extended pitch-segment detection
VAD based on *a posteriori* SNR weighted energy difference
Postprocessing
Voice activity detection
Output

Figure 1: Block diagram of the proposed method. The illustration is borrowed from the original authors [4].

- Classification of the high-energy frames using a posterior SNR weighted energy difference of two consecutive frames.
- Samples belonging to high-energy segments are set to zero.

4. Second-pass denoising.

   - Speech enhancement using a modified version of the Minimum Statistics Noise Estimation (MSNE) approach.

5. Voice Activity Detection

   - Computation of the a posterior SNR weighted energy difference based on the a posterior weighted energy difference on the pitch segments.

- Classification of speech segments.
- Post-processing.

The noise-corrupted speech signal is modeled using the following additive noise signal model

$$x(n) = s(n) + v(n),$$

where $x(n)$ and $s(n)$ represent the noisy and clean speech sample at time n, respectively, and $v(n)$ the sample of additive noise at time n. No prior assumptions on the type of additive noise are used. The two first denoising steps aim at increasing the overall Signal over Noise Ratio (SNR) of the input signal, before classifying the speech segments. In the following, n indexes the discrete sampling time of the signal, m the frames - resulting from the partitioning of $x(n)$ - and p indexes the super-segments - concatenation of frames. An illustration of these different scales can be found in Figure 2.

2.1 Pre-processing of the Audio Signal

Let's note the audio signal of interest $x(n) \in \mathbb{R}^{N \times C}$, with N its length, C the number of channels, and f_s its sampling frequency. If the audio signal was stored using 16 bits or 32 bits encoding, it is normalized following Equation (1), such that its sample values range between -1 and 1, where b denotes the number of bits used and n the discrete time index.

$$x_{normalized}(n) = \frac{x(n)}{2^{b-1}+1}. \tag{1}$$

Additionally, only the samples from the first channel are considered, such that the signal considered lives in $[-1, 1]^N$.

The signal $x_{normalized}(n)$ is then filtered by a digital Infinite Impulse Response (IIR) high-pass filter to remove the DC component and low frequency noise. A cutoff frequency of 60 Hz is applied to remove low-frequency noise. This step is done using the scipy.signal.lfilter method from the scipy library.

2.2 Creation of the Frames

The signal $x(n)$ (we rename it for convenience after the preprocessing step) is then partitioned into frames of length $l = 25$ms with a frame shift $\Delta l = 10$ms, resulting in a matrix $X^{frame}(m, n) \in \mathbb{R}^{N_f \times n_l}$, with

$N_f = \lceil \frac{N - \lfloor f_s \times l \rfloor + n_{\Delta l}}{n_{\Delta l}} \rceil$ *Number of frames extracted from the audio signal,*

$n_l = \lfloor f_s \times l \rfloor$ *Number of samples per frame,*

$n_{\Delta l}$ *Number of samples between the frames.*

Figure 2 illustrates the partitioning into frames. Note that with these notations, the percentage of overlap between frames is $(n_l - n_{\Delta l})/n_l$. If necessary, the zero-padding approach is used to fill the last frame. Each extracted frame is then multiplied by a Hamming window $W(m, n) \in \mathbb{R}^{N_f \times n_l}$, such that $X(m, n) = X^{frame}(m, n) \odot W(m, n)$, with $\odot$ the Hadamard product of the two matrices. The application of a Hamming window usually precedes the computation of Short-Time Fourier Transform (STFT), as described in what follows.

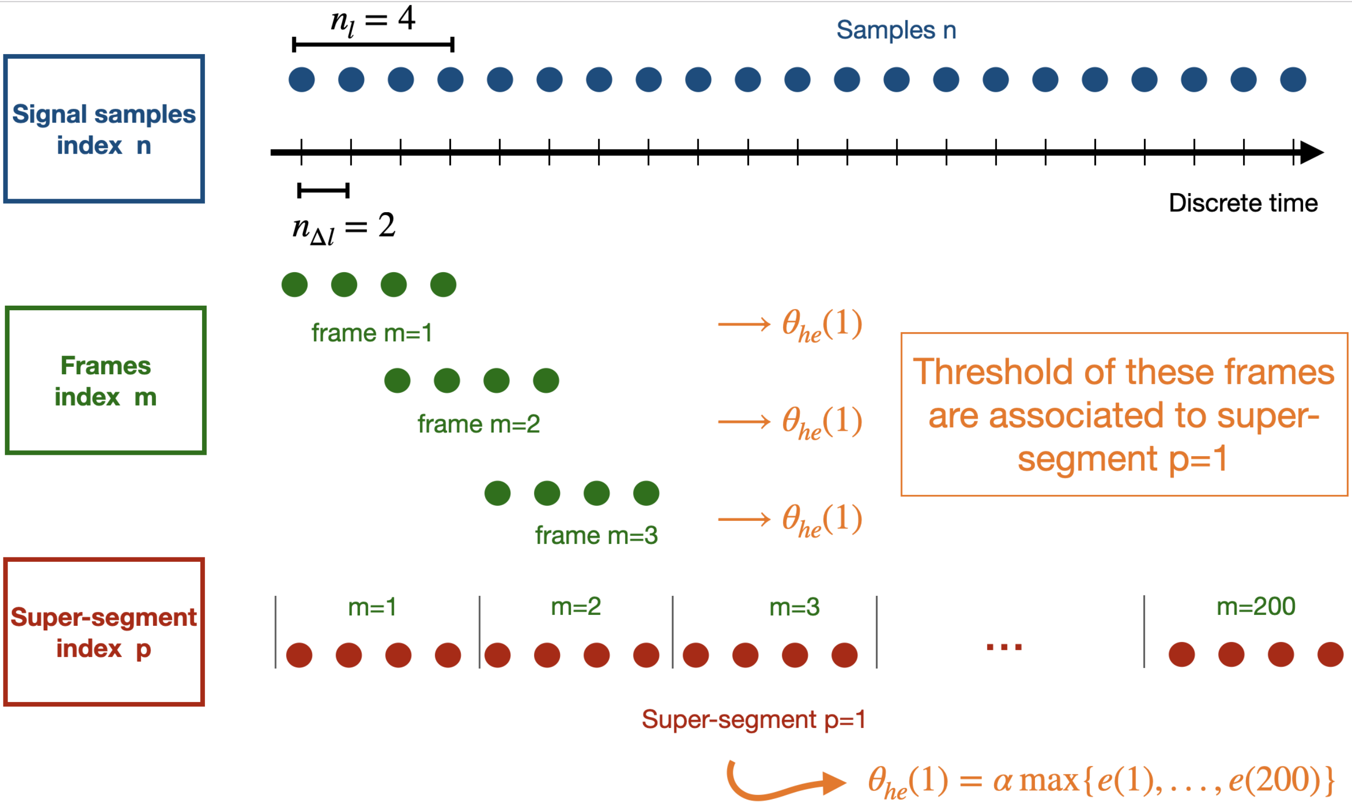

$n_l = 4$
Samples n
Signal samples index n
Discrete time
$n_{\Delta l} = 2$
Frames index m
frame m=1
frame m=2
frame m=3
$\longrightarrow \theta_{he}(1)$
$\longrightarrow \theta_{he}(1)$
$\longrightarrow \theta_{he}(1)$
Threshold of these frames are associated to super-segment p=1
Super-segment index p
m=1
m=2
m=3
...
m=200
Super-segment p=1
$\theta_{he}(1) = \alpha \max\{e(1), \ldots, e(200)\}$

Figure 2: Illustration of the partitioning into frames and super-segments (blue, green, red), and computation of the high-energy frames thresholds (orange).

2.3 Identification of the Frames Containing Pitch using the Spectral Flatness

As the computational complexity of pitch detection is relatively high, the authors investigated an alternative measure of pitch, the spectral flatness (SFT) [2]. Replacing the pitch detector by a simple SFT-based voiced/unvoiced speech detector lead the authors to propose a more computationally efficient algorithm, called rVAD-fast. The computation of the spectral flatness of each frame will be used to compute a binary mask, $Z(m) \in \{0,1\}^{N_f}$, which indicates whether the frame m contains pitch. This binary mask $Z(m)$ is used in the first-pass denoising and for the final speech detection part.

Computation of the Spectral Flatness of each Frame

Figure 3 illustrates the computation of the binary mask $Z(m)$. For $m \in [1, N_f]$, we note $f_m(n)$ the frame m. The one-dimensional STFT of $f_m(n)$ is computed using $K = 512$ points, which represents the frame in the spectral domain as

$$F_m(n,k) = S_m(n,k) + V_m(n,k).$$

Thereafter, the spectral flatness SFT_m of frame m is calculated following Equation (2)

$$SFT_m = \frac{\exp\left(\frac{1}{K}\sum_{k=0}^{K-1} \log|F_m(n,k)|\right)}{\frac{1}{K}\sum_{k=0}^{K-1} |F_m(n,k)|}, \tag{2}$$

where $|F_m(n,k)|$ denotes the magnitude spectrum of the k^{th} frequency bin for the m^{th} frame, and K is the total number of frequency bins.

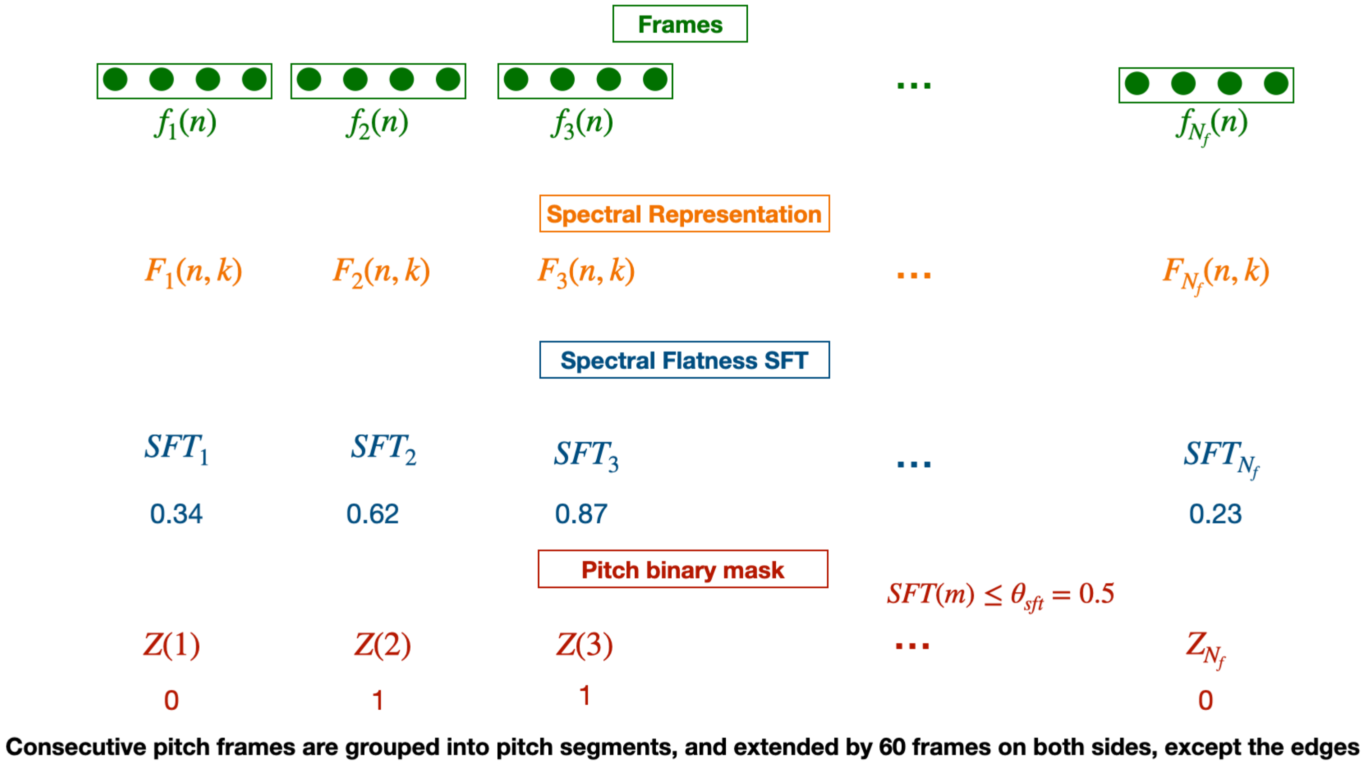


Figure 3: Illustration of the computation of the pitch binary mask $Z(m)$.

Classification of the Voiced and Unvoiced Frames

As SFT is used as a replacement for pitch, SFT values are compared against a predefined threshold θ_{sft} to decide whether their corresponding frame is voiced or unvoiced. Figures 4 illustrates spectrogram, pitch labels (1 for pitch and 0 for no pitch) and SFT values of a speech signal from TIMIT (clean) and those of a signal from the NIST 2016 SRE evaluation (noisy), respectively.

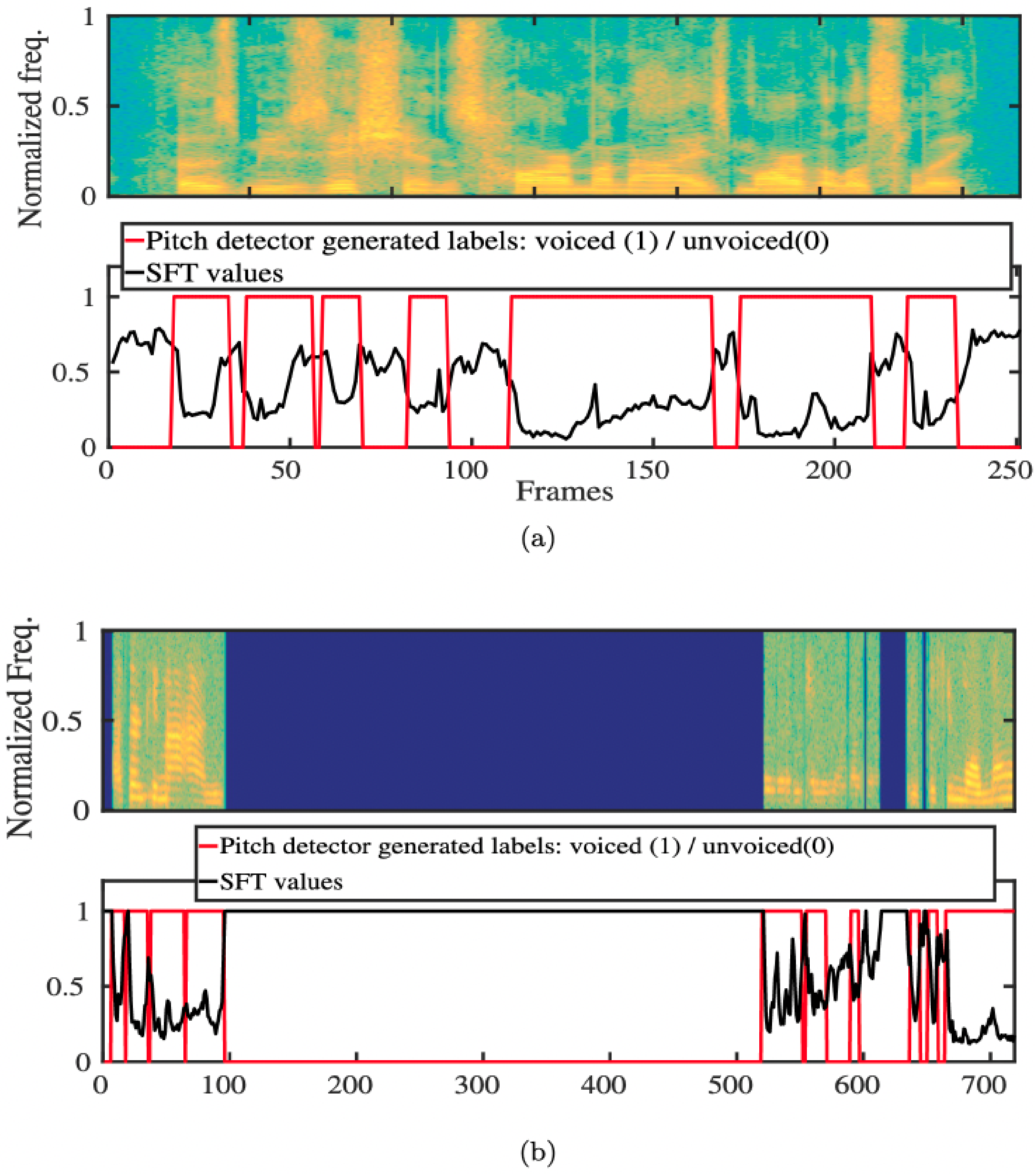
Figure 4 shows that if we choose a threshold value θ_{sft} of 0.5 (i.e. if $SFT_m \leq 0.5$, the frame m is said to contain pitch), the labels generated by SFT are close to those generated by the pitch detector. The authors claimed that they extensively studied the effect of different threshold values θ_{sft} on the performance of SFT as a replacement of the pitch detector, which resulted in setting the default value of θ_{sft} to 0.5.

This thresholding of each SFT value result in the binary pitch mask, $Z(m) \in \{0,1\}^{N_f}$, indicates whether each frame contains or not pitch, which is a proxy for whether they contain speech signal.

Extension of the Voiced Segments

A fundamental assumption made by the authors is that all speech segments should contain several speech frames with pitch. In their algorithm, pitch frames are first grouped into pitch segments, which are then extended on both ends by 60 frames (600ms; which is based on speech statistics) to include voiced sounds, unvoiced sounds, and potentially non-speech parts. This is illustrated in Figure 3, and results in the pitch binary mask $Z \in \{0,1\}^{N_f}$ (renamed for convenience after extending the edges of each pitch segment).

This strategy is taken for the following reasons: 1) pitch information is already extracted in the previous steps, 2) pitch is used as an anchor to trigger the VAD process, and 3) many frames can be potentially discarded if an utterance contains a large portion of non-speech segments, which are also non-pitch segments.



(a)

(b)

Figure 4: Spectrogram (top) and the associated spectral flatness (bottom) of two speech signals, one clean for the TIMIT dataset (a) and a noisy speech signal from the NIST 2016 SRE evaluation set (b). We can see on both cases the pitch labels, with 1 when the signal contains pitch, 0 otherwise. The illustration is borrowed from the original authors [4].

2.4 First-pass Denoising

Computation of the A Posteriori SNR Weighted Energy Difference

In the first pass, high-energy segments are detected using an *a posteriori SNR weighted energy difference* measure [3]. To compute this metric, we first need to compute the energy associated to a frame $m \in [1, N_f]$, following Equation (3).

$$e(m) = \sum_{n=1}^{n_l} X_{mn}^2. \tag{3}$$

The next step to compute the *a posteriori SNR weighted energy difference* measure is to compute the a posteriori SNR of a single frame m, denoted $SNR_{post}(m)$, following Equation (4)

$$SNR_{post}(m) = 10\log_{10} \tfrac{e(m)}{\tilde{e}_v(m)}, \tag{4}$$

where the estimated noise energy of frame m, $\tilde{e}_v(m)$ is computed using the following scheme.

First, the filtered signal x is partitioned into multiple super-segments of size $N_S = 200$ frames each (about $2s$), without overlap between the super-segments, which results in the super-segment matrix $X^{super-segment} \in \mathbb{R}^{P\times 200}$, with P the number of super-segments. Figure 2 illustrates the partitioning into frames and super-segments.

Then, the energy of the noise of a super-segment p, $e_v(p)$ is calculated as the energy of the frame ranked at 10% of lowest energy within the super-segment, as described in Algorithm 1.

Algorithm 1: Computation of the noise energy of the super-segment p, $e_v(p)$.

```
Input $X^{super-segment}_{\bullet p}$: Filtered samples of the super-segment $p$
Output $e_v(p)$: Estimated noise energy of super-segment $p$
$e(1), \dots, e(200) \coloneqq$ compute_energy($X^{super-segment}_{\bullet p}$)
$\tilde{e}(1), \dots, \tilde{e}(200) \coloneqq$ sort_increasing_order($e(1), \dots, e(200)$);
$e_v(p) \coloneqq \tilde{e}(20)$        # 10% lower energy frame
return $e_v(p)$
```

Thereafter, the final estimated noise energy $\tilde{e}_v(p)$ is calculated as the smoothed version of $e_v(p)$ with a forgetting factor of 0.9 as follows

$$\tilde{e}_v(p) = 0.9 \times \tilde{e}_v(p-1) + 0.1 \times e_v(p).$$

The noise energy of the m^{th} frame, $\tilde{e}_v(m)$, takes the energy value $\tilde{e}_v(p)$ of the p^{th} super-segment which the m^{th} frame belongs to.

Finally, we can compute the *a posteriori SNR weighted energy difference* between the frames m and $m-1$, following Equation (5). At the edges of the signal, the first value is repeated 18 times at the beginning, and the last value is repeated 18 times at the end.

$$d(m) = \sqrt{|e(m) - e(m-1)| \times \max(SNR_{post}(m), 0)}. \tag{5}$$

In Equation (5), the square root is taken to reduce the dynamic range, which differs from [3] where the square root is not applied.

Once computed, the *a posteriori SNR weighted energy difference* of each frame is smoothed using a moving average window of length $D = 18$ frames, following Equation (6).

$$\bar{d}(m) = \tfrac{1}{2D+1} \sum_{i=-D}^{D} d(m+i). \tag{6}$$

Classification of the High-energy Frames

A frame is then classified as being high-energy if $\bar{d}(m) \geq \theta_{he}(m)$. For each super-segment p (containing 200 frames), $\theta_{he}(p)$ is computed following Equation (7). Figure 2 illustrates the computation of these thresholds.

$$\theta_{he}(p) = \alpha \max\{e((p-1) \times 200 + 1), \dots, e(200 \times p)\}, \tag{7}$$

where $\alpha = 0.25$. Then $\theta_{he}(m)$ takes the threshold value $\theta_{he}(p)$ of the p^{th} super-segment which the m^{th} frame belongs to. Consecutive high-energy frames are grouped together to form high-energy segments.

Denoising of the High-energy segments

For each high-energy segment, as defined above, the pitch binary mask $Z(m)$ is used to decide whether the high-energy segment is noise, or has a chance to be speech. Within a high-energy segment, **if no more than two pitch frames are found**, the segment is classified as noise, and the samples of all the frames in that segment are set to zero. The high-energy segments containing at least 2 pitch frames are labeled **extended pitch segments**.

According to the authors, the motivations of this pass is two-fold: first, to avoid overestimating noise due to the burst-like noise when applying a noise estimator in the second pass denoising and, secondly, to detect and denoise high-energy non-speech parts, which are otherwise difficult for conventional denoising and VAD methods to deal with.

2.5 Second-pass Denoising

The second-pass denoising is a speech enhancement technique that relies on noise estimation. The authors resorted to a modification of the Minimum Statistics Nise Estimation (MSNE) approach [1].

In their proposed MSNE-mod approach, using the spectral representation of each frame $F_m(n,k)$, as denoted in Section 2.3, if more than half of the energy is located within the first 7 frequency bins ($f < 217Hz$), the values of the 7 frequency bins are set to zero to further remove low-frequency noise in addition to the use of the first-order high-pass filter mentioned earlier. This step is claimed by the author to remove remaining stationary noise from the input signal.

2.6 Voice Activity Detection

The VAD algorithm is based on the denoised signal and the pitch information generated from the previous steps. The *a posteriori SNR weighted energy difference* is computed for each **extended pitch segment**.

Computation of the SNR Weighted Energy Difference on the Extended Pitch Segments

First, the *a posteriori SNR weighted energy difference*, $d'(m)$, is computed for each frame m following Equation (5). To calculate the $SNR_{post}(m)$ component following Equation (4), $\tilde{e}_v(m)$ is estimated as the energy ranked at 10% of lowest frame energy **within the extended pitch segment**, which differs from the computation of the $SNR_{post}(p)$ in the first-pass denoising, which considered the energy of super-segments to compute the noise energy $\tilde{e}_v(p)$ of each super-segment p.

Central-smoothing of the SNR Weighted Energy Difference

Then, the a posteriori SNR_weighted energy difference is central-smoothed with $D = 18$ as in Equation (6), resulting in $d'\bar{(}m)$. At the edges of the signal, the first value is repeated 18 times at the beginning, and the last value is repeated 18 times at the end.

Classification of Speech Frames

A frame is then classified as containing speech if its *a posteriori SNR weighted energy difference* $d'\bar(m)$ is greater than the threshold θ_{VAD}.

$$\theta_{\text{VAD}} = \beta\tfrac{1}{L}\sum_{j=1}^{L} d'\bar(j),$$

where L is the total number of frames with pitch in the extended pitch segment, and the default value for β is set to 0.4 by the authors. This results in the speech binary mask $SPEECH(m)$, containing 1's for each frame contains speech, 0 otherwise.

Post-processing

Finally, a post-processing step on the binary speech mask $SPEECH(m)$ is applied. The authors' assumption are as follows, and are derived in the rules below.

- Speech frames should not be too far away from their closest pitch frame
- Within a speech segment, there should be a certain number of speech frames without pitch.

First, frames that are 33 frames away from the pitch segment to the left and 47 frames away to the right are classified as non-speech, regardless of the VAD results above, which covers 95% of the cases based on a few speech files. On the other hand, frames within 5 frames to the left and 12 frames to the right of the pitch segments are classified as speech, again regardless of the VAD results, which leaves out 5% of the cases based on a few speech files. According to the authors, this hangover schemes appear to be frequent in other VAD methods. Furthermore, segments with energy below 0.05 times the overall energy are removed.

3 Experiments

This section presents some qualitative experiments on the algorithm based on its performance on the audio samples associated with the demo. While those experiments are mostly user-oriented, as they depict the strength and limitations of the approach on a few enlightening examples, interested readers can find in the original paper a few experiments on the hyperparameters of the method.

An Optimal Audio Sample

Figure 5 illustrates the final speech segmentation of the method on a well-defined clean audio sample. Indeed, this example features an audio sample with a clear alternation of speech and silence, which are indeed detected.

Figure 6 shows a reasonably tricky example of a noisy audio sample with speech and no-speech segments. Although burst and stationary noise corrupted the original audio signal, the algorithm can still correctly label the speech segments.

Limits of the Method

One limit of this approach is that it is barely robust to white noise, as illustrated in Figure 7. This is due to the use of the spectral flatness as a proxy for whether or not there is pitch, for audio samples severely corrupted with white noise.

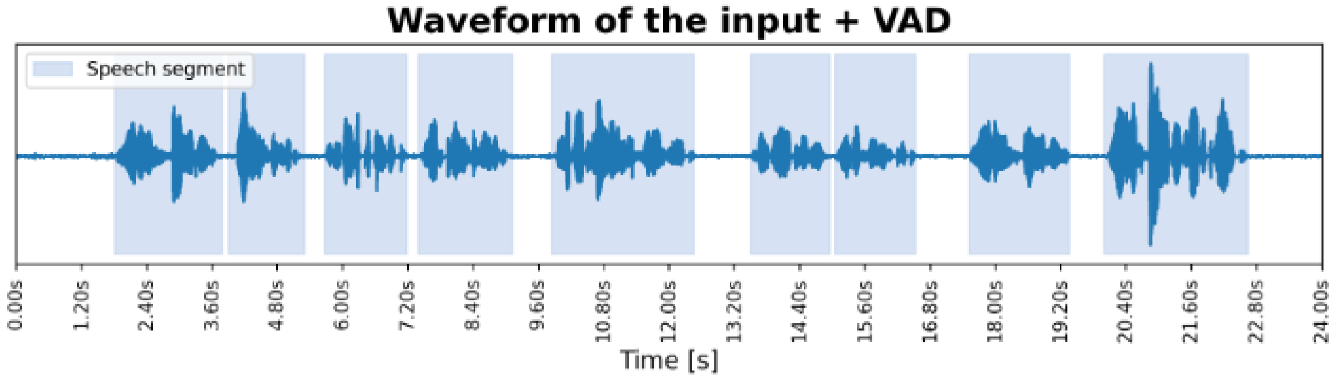


Figure 5: Waveform of the 'Speech' audio sample of the demo, overlapped with the outputted speech segmentation of rVAD-fast. As we can see in this example, the speech parts of the input are detected as such by the algorithm.

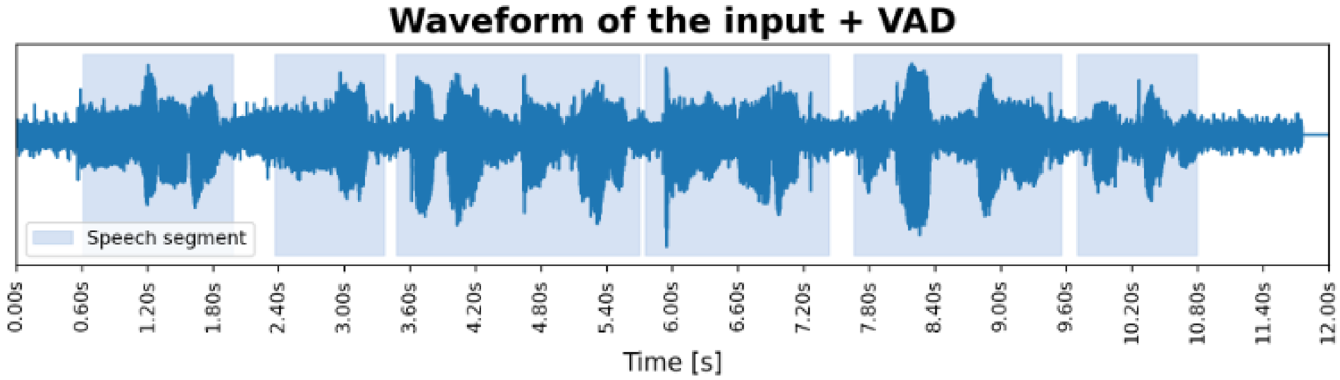


Figure 6: Waveform of the 'a capella' audio sample of the demo, overlapped with the outputted speech segmentation of rVAD-fast. Audio sample with both speech and no-speech segments, corrupted by impulsive (or burst) noise.

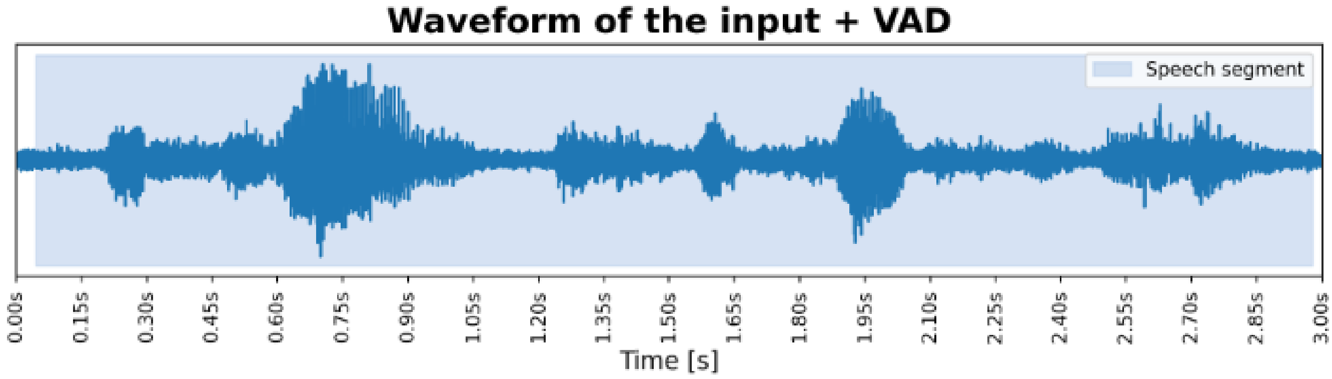


Figure 7: Waveform of the 'People talking - White noise' audio sample of the demo, overlapped with the outputted speech segmentation of rVAD-fast. This audio signal features speech overlapped with white noise. As we can see, the algorithm fails to label the non-speech part as such, and the whole sample is incorrectly labeled as speech.

Another major limit of the algorithm is that it fails on extreme audio signal with no speech. For audio signal featuring only music or sounds, the addressed task is closer to the one of a silence detector than voice activity detection. Figure 8 illustrates such example for an audio sample featuring instrumental jazz music only, not corrupted by noise. Except for a short silence in the beginning, the rest of the audio signal is incorrectly labeled as speech. The approach being unsupervised, it appears necessary that the input signal features both speech and non-speech segments so that a contrast between both appears in the spectral signature of the input, revealed when computing the spectral flatness of the signal.

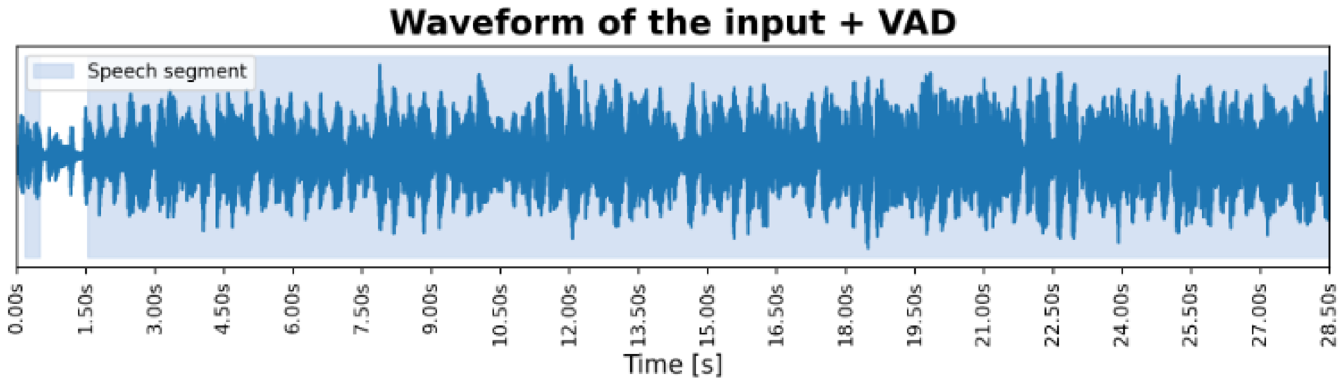


Figure 8: Waveform of the 'Jazz song - no speech' audio sample of the demo, overlapped with the outputted speech segmentation of rVAD-fast. This audio signal features an extreme case with no speech, with instrumental jazz music only. As we can see, except a short silence in the beginning, the rest of the audio signal is incorrectly labeled as speech.

4 Demos

The demo[3] associated to this algorithm is composed of a few components:

- An input section. See Figure 9.
- A parameter section, followed by the RUN button. See Figure 9.
- The output section, which appears after the algorithm has finished to run, and features the different outputs of the algorithm. See Figure 10.

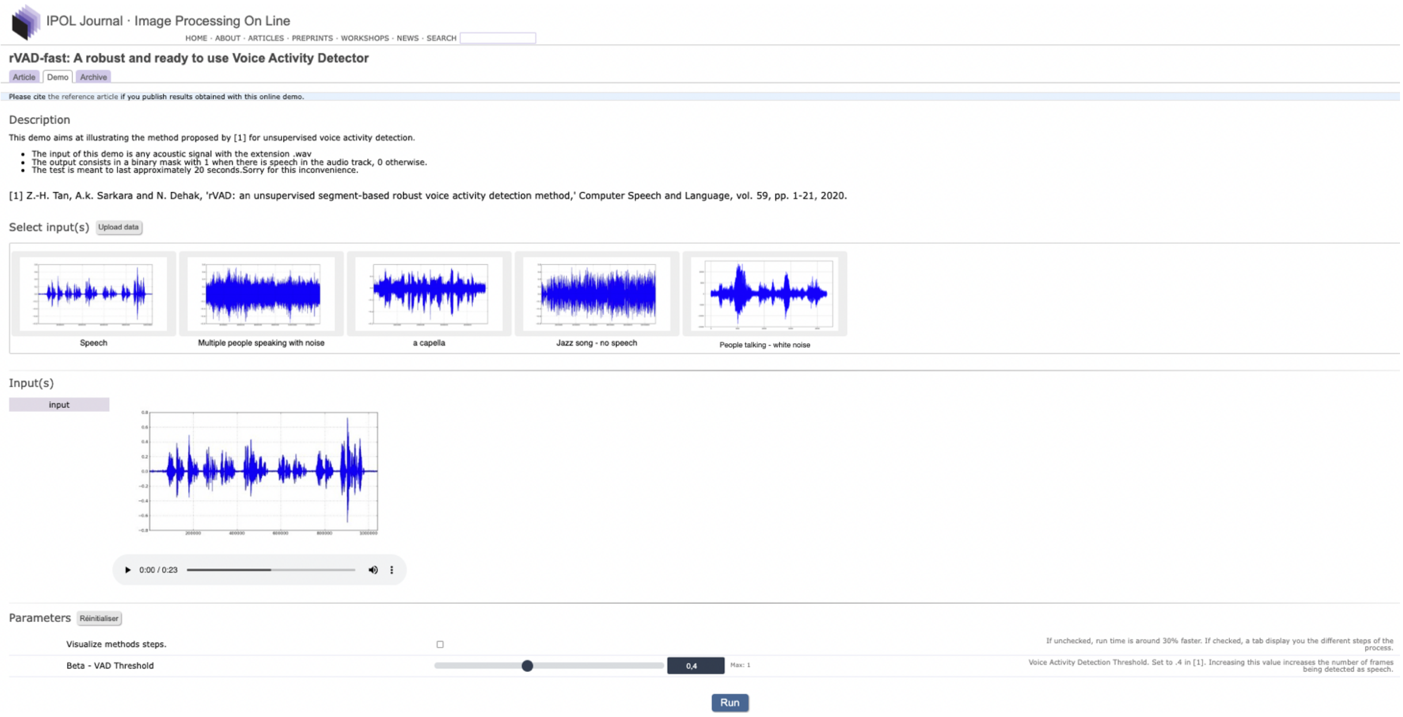
The Input Section

This part allows the user to upload its audio sample. In this case, the user is invited to browse a .wav file that should not exceed around 300 Mb or a few minutes duration. **Otherwise, the running time of the algorithm might exceed a minute.** The user can also test the algorithm by selecting one of the five pre-selected audio samples. Those provided examples represent a large variety of use cases, illustrative of the strength and weaknesses of the method: Speech, Multiple people speaking with noise, a capella, Jazz song - no speech, or People talking - white noise.

The Parameter Section

- Visualize method steps. Checkbox which allows to visualize a few steps of the method if checked. If unchecked, the run time is around 30% faster. This parameter is unchecked by default.
- β - VAD threshold. Final threshold of the voice activity detection described in Section 2.6. The default value is $\beta = 0.4$, as encouraged by the authors. The higher the threshold, the less likely the signal frames are to be labeled as speech.

[3] https://doi.org/10.5201/ipol.2022.427



IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

rVAD-fast: A robust and ready to use Voice Activity Detector

Article | Demo | Archive

Please cite the reference article if you publish results obtained with this online demo.

Description

This demo aims at illustrating the method proposed by [1] for unsupervised voice activity detection.

- The input of this demo is any acoustic signal with the extension .wav
- The output consists in a binary mask with 1 when there is speech in the audio track, 0 otherwise.
- The test is meant to last approximately 20 seconds.Sorry for this inconvenience.

[1] Z.-H. Tan, A.k. Sarkara and N. Dehak, 'rVAD: an unsupervised segment-based robust voice activity detection method,' Computer Speech and Language, vol. 59, pp. 1-21, 2020.

Select input(s) Upload data

Speech | Multiple people speaking with noise | a capella | Jazz song - no speech | People talking - white noise

Input(s)

input

0:00 / 0:23

Parameters Réinitialiser

Visualize methods steps. □ — If unchecked, run time is around 30% faster. If checked, a tab display you the different steps of the process.

Beta - VAD Threshold 0,4 Max: 1 — Voice Activity Detection Threshold. Set to .4 in [1]. Increasing this value increases the number of frames being detected as speech.

Run

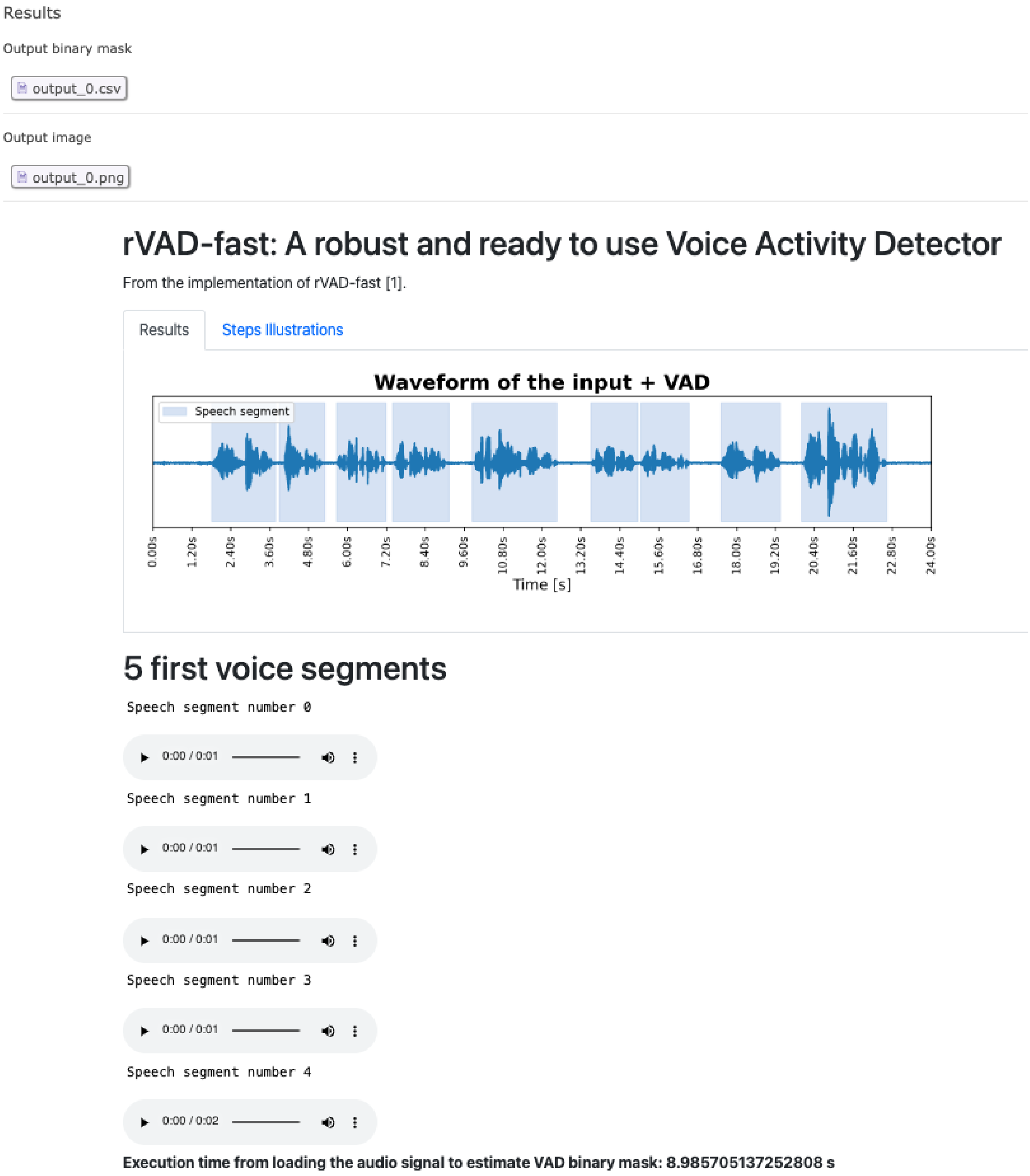
Figure 9: Input and parameters sections of the demo.

The Output Section

The output section is comprised of:

- A .csv file containing the output segmentation, with a time column, and the final binary mask, with 1 for segments with speech, 0 otherwise.
- A .png image containing the waveform of the input signal overlapped with the speech segments.
- A tab panel containing the image presented above on one tab, and a few graphs showing the different steps of the algorithms on the other tab (see Figure 11).
- A section with the 5 first speech segments detected, which can be directly listened by the user.

The execution time is also displayed at the end of the output section.



Results

Output binary mask

output_0.csv

Output image

output_0.png

rVAD-fast: A robust and ready to use Voice Activity Detector

From the implementation of rVAD-fast [1].

5 first voice segments

Speech segment number 0

Speech segment number 1

Speech segment number 2

Speech segment number 3

Speech segment number 4

Execution time from loading the audio signal to estimate VAD binary mask: 8.985705137252808 s

Figure 10: Output of the demo when using the provided audio sample 'Speech'.

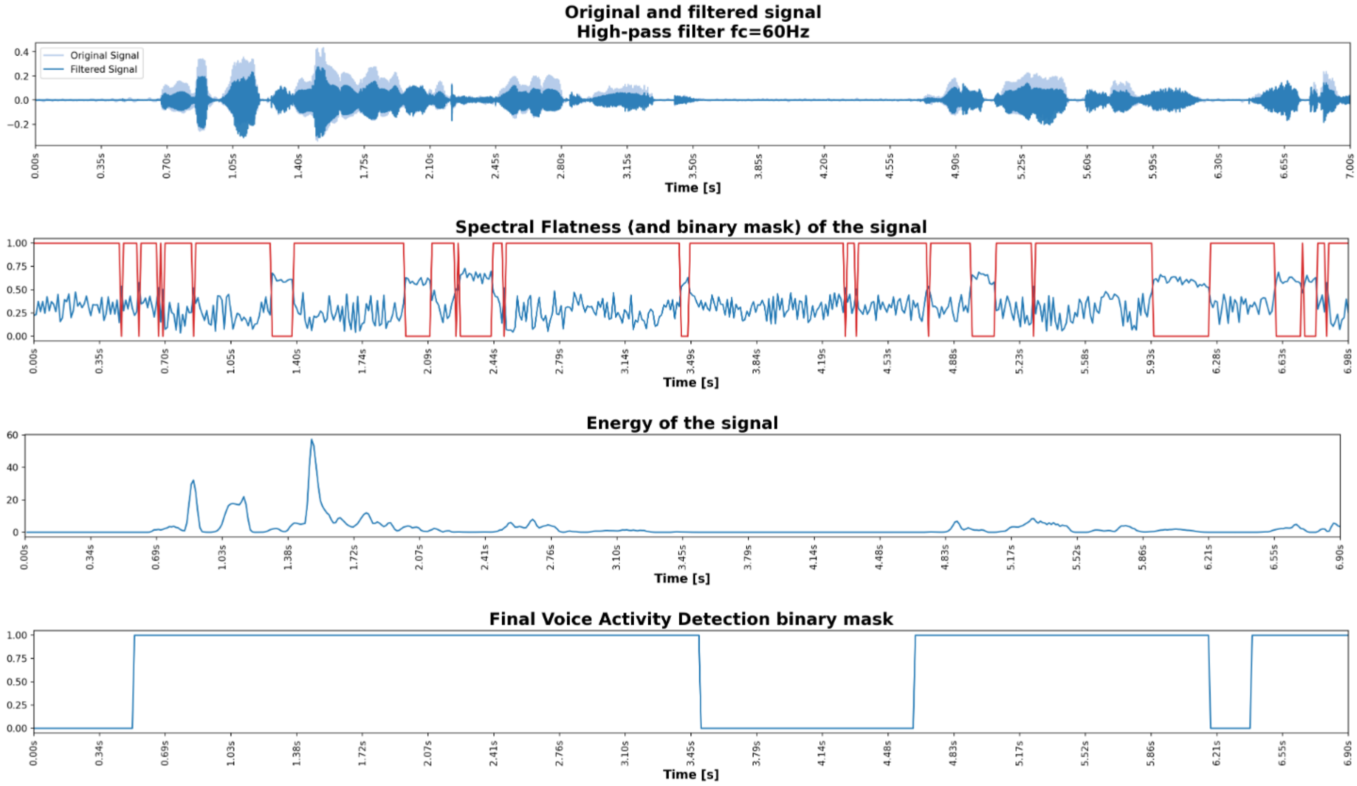


Figure 11: Visualization of different steps of the process. From top to bottom, these graphs represent the original and high-pass filtered audio sample, the computed spectral flatness, overlapped with a binary mask showing potential voiced segments, the energy of the denoised filtered audio sample, and the final binary speech segmentation of the method.

5 Conclusion

The algorithm performs well on audio signals containing speech segments alternated with clean silences. It is robust to rapidly changing, burst, and stationary noise. The authors have also shown that it works well under babble, market, and car noise under clean conditions. However, it is not robust to white noise and performs as a silence detector in audio signals without speech segments.

Acknowledgment

This paper was supported by the Borelli Center. We thank the members of the IPOL team and organizers of the MLBrief first edition, hold in March 2022, Tina Nikoukhah, Roger Marí, Quentin Bammey, Jérémy Anger , Charles Hessel, Gregory Randall, Jean-Michel Morel, Thibaud Ehret, Enric Meinhardt-Llopis, Rafael grompone, Gabriele Facciolo, Miguel Colom, Charles Truong, Ioannis Bargiotas, Nicolas Vayatis, Laurent Oudre, and Mathilde Mougeot, for their incredible assistance in all parts of the demo creation process, and Laurent Oudre for his supervision and comments that greatly improved the manuscript. We would also like to show our gratitude to the original authors, Zheng-Hua Tan, Achintya kr. Sarkar, and Najim Dehak.

References

[1] R. Martin, *Noise power spectral density estimation based on optimal smoothing and minimum*

statistics, IEEE Transactions on Speech and Audio Processing, (2001), pp. 504–512. https://doi.org/10.1109/89.928915.

[2] G. Peeters and X. Rodet, *A large set of audio feature for sound description (similarity and classification) in the CUIDADO project*, tech. report, Ircam, Analysis/Synthesis Team, Paris, France, 2004.

[3] Z-H. Tan and B. Lindberg, *Low-complexity variable frame rate analysis for speech recognition and voice activity detection*, IEEE Journal of Selected Topics in Signal Processing, 4 (2010), pp. 798–807. https://doi.org/10.1109/JSTSP.2010.2057192.

[4] Z-H. Tan, A.kr. Sarkar, and N. Dehak, *rVAD: An unsupervised segment-based robust voice activity detection method*, Computer Speech & Language, 59 (2020), pp. 1–21. https://doi.org/10.1016/j.csl.2019.06.005.