COMPARISION OF VAD AND VVAD

Nimcy Lisiya D[1], Nivedha R[2], Padma J P[3], Mr.Balasubramanian Sivasubramanian[4]

[1]Student, Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India
[2]Student, Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India
[3]Student, Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India
[4]Associate Professor, Department of Electronics and Communication Engineering, Meenakshi Sundararajan Engineering College, Chennai, India

---***---

Abstract- *The quality of speech signal is essential in wide variety of applications. Majority of them require the voice activity detection (VAD) as an important part of the pre processing stage. For speech based application VAD place a major role involving the usage of only the audio cues from the input. Eventhough VAD is more effective but it result in false estimation due to acoustic disturbances or background noise which also includes many persons speaking around. This false detection arise as VAD use only the audio cues. There are several visual cues but the proposed design uses the lip movement features using opencv which helps to define whether the speaker has opened or closed the mouth. The benefits of employing the video approach for voice activity detection are evident when the audio has a poor SNR (which would produce bad results if using a solely audio based VAD). The separation of vocal activity periods and non speech periods can also be used to isolate and estimate the noise, during the utterance periods. As, the visual cues are independent of acoustic disturbances or background noises resulting in high quality speech even in low snr. This proves the beneficial use of visual cues over audio cues for better quality audio*

Key Words: VAD(voice activity detection), audio cues, VVAD(visual voice activity detection),visual cues, opencv, SNR(signal to noise ratio).

1. INTRODUCTION

Human speech comprises two processes namely, the auditory and the visual one. Many researchers have emphasized the close connection between the two (audio and visual). A human cannot produce auditory speech without also exhibiting visual cues such as lip, head or eyebrow movements, and these may provide supplementary information to various applications involving automatic speech recognition (ASR, [2]), audio surveillance and monitoring, language identification [3], speech coding [1], speech enhancement, or speaker. For many of these applications it is important to be capable to detect when a person is speaking. Hence, as a solution voice activity detection (VAD) serves as an essential component for a variety of speech processing applications. It has been observed that performances of various speech based tasks are very much dependent on the efficiency of the VAD

2. VAD

Voice activity detection is also called as speech activity detection or speech detection. This technique is used in speech processing to detect the presence or absence of human speech. VAD is an important enabling technology for a variety of speech-based applications such as speech coding , speech recognition and it can also be used to deactivate some processes during non-utterance section of an audio i.e, it can avoid unnecessary coding/transmission of voiceless packets in Voice over Internet Protocol applications thereby saving on computation and on network bandwidth. Voice activity detection is usually language independent.

2.1. VAD ALGORITHM

The Voice Activity Detector implements the algorithm described in [6],which detects the presence or absence of speech. It is thus a binary decision. A pertinent errand is to determine the *probability* that an input signal contains speech or not, referred to as the *speech presence probability* (SPP). SPP is an intermediate step in voice activity detection and it range from 0 to 1.The voice activity classification is obtained by thresholding the output of the speech presence probability estimator. VAD makes more complex tasks simple. For example, speech recognition need to be applied only when speech is present. Similarly, in speech coding, we can transmit only the speech part and thereby we can reduce bitrate in the absence of speech.

2.2. PROBLEM FORMULATION

Consider an input signal *x*. Our objective is to determine whether it is speech or not. We express the VAD algorithm as a function y=*VAD(x)*, where the desired target output is

$$y^* := \begin{cases} 0, & x \text{ is not speech,} \\ 1, & x \text{ is speech.} \end{cases}$$

Correspondingly, the speech existence probability is the probability that *x* is speech, *SPP(x)=P(x is speech).* A possible definition for the VAD is then,

$$VAD(x) := \begin{cases} 0, & SPP(x) < \theta \\ 1, & SPP(x) \geq \theta, \end{cases}$$

where θ is a scalar threshold.

2.3. ENERGY THRESHOLDING

A speech signal is not a static signal. Sometimes we speak energetically and sometimes we do not .Thus the presence of speech can be detected by signal energy.

$$VAD(x) := \begin{cases} 0, & \sigma^2(x) < \theta_{SILENCE} \\ 1, & \sigma^2(x) \geq \theta_{SILENCE}. \end{cases}$$

As speech adds energy to the signal, high-energy regions of the signal are likely speech. For example, we can set a threshold θ SILENCE such that when the energy of the signal σ2(x) is above the threshold, the VAD indicates speech activity. For example: To implement this approach, we first apply windowing to the input signal with 20 ms windows and 50 % overlap. For each window, we calculate signal energy as

$$\sigma^2(x) := \|x\|^2 = \sum_{k=0}^{N-1} x_k^2.$$

To choose a suitable threshold, in the fig-1, we plot the energy over a speech signal σ2(x). We can notice that areas in the speech signal with less activity have an energy below 17 dB, whereby we can set the threshold at θSILENCE:=17dB. The resulting voice activity estimate is illustrated in the lowest pane.

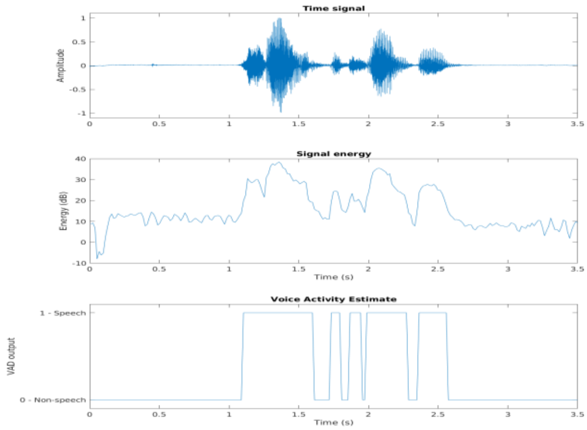

Fig -1: VAD example with energy thresholding

2.4PERFORMANCE CRITERIA

The function of voice activity detection (VAD) is unequivocal. But it is difficult to evaluate the performance. The performance criteria depend on the application;

- During speech transmission(speech coding) we could turn off the transmission when there is no speech but this may lead to degradation when false detection happens.

VAD is employed in speech enhancement algorithm to estimate the noise statics in speech segment. In speech segments, the algorithm can remove everything which seems like noise. If speech is present within the segment where we estimate the characteristics of noise, then we might remove features which appear during noise segments. This may cause degradation of the desired speech signal.

3. VVAD

VVAD is nothing but Visual Voice Activity Detection which involves in observing all the input data in the form of image. Initially VVAD involves in a step of identifying the face in the given input. First the input video is splitted into frames and for each frame is there a face or not is recognized. For recognizing there are various methods involved. With the help of all the facial features and facial landmarks we can able to recognize the face in all the frames.

As computer vision engineers and researchers we've been trying to grasp the human face. The foremost obvious application of facial analysis is Face Recognition. But to be ready to identify an individual in a picture we first have to find where the face is there. Facial feature detection is additionally cited as “facial landmark detection”, “facial keypoint detection” and “face alignment”. The most important algorithm in recognizing the face is that the Viola-Jones Algorithm.

The Viola–Jones object detection framework is that the first object detection framework to supply competitive object detection rates in real-time which is proposed in 2001 by Paul Viola and Michael Jones. Although it may be trained to detect a range of object classes, it absolutely was motivated primarily by the problem of face detection. The problem to be solved is detection of faces in a picture. An individual's can try this easily, but a computer needs precise instructions and constraints. To create the task more manageable, Viola–Jones requires full view frontal upright faces. Thus so as to be detected, the complete face must point towards the camera and may not be tilted to either side.

Even though Viola-Jones Algorithm is efficient in identifying the object (i.e face). But it don’t have the accurate result when we do with matlab software. So, for the efficient output we have chosen OpenCV python implementation in our proposed system.

3.1. OpenCV

OpenCV is the foremost popular library for computer vision. Originally written in C/C++, it now provides bindings for Python. OpenCV uses machine learning algorithms to explore for faces within a picture. Because faces are so complicated, there isn't one simple test which will tell you if it found a face or not.

For identifying a face there are 6,000 or more classifiers, all of which must match for a face to be detected. But therein lies the problem: for face detection, the algorithm starts at the best left of a picture and moves down across small blocks of data, observing each block, constantly checking whether it matches with the features listed for deciding it as a face. Since there are 6,000 or more tests per block, you'd possibly have legion calculations to undertake to, which may grind your computer to a halt.

To get around this, OpenCV uses cascades. The OpenCV cascade breaks the matter of detecting faces into multiple stages. For each block, it does a awfully rough and quick test. If that passes, it does a rather more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the image will return a negative during the first few stages, which suggests the algorithm uses time efficiently by not testing all 6,000 features on it ie., instead of taking hours, face detection can now be done in real time.

3.2. OpenCV FEATURE DETECTION

Video features describe the lip motion of a possible speaker. During this work, geometrical lip features are used, therefore the features describe the form of a mouth instead of raw intensity values within the mouth region. The extraction of such features consists of several steps, namely face detection, facial landmarks localization and geometrical features extraction. Also, the features should be normalized so they're invariant to the face size and head pose. However, we suppose that each one analyzed speakers face the camera; no profile views are considered for simplicity. Lip-based approaches employ geometrical models supported the lips. The geometrical models typically include a versatile mesh formed by landmarks, or connected fiducial points surrounding the lips, flexible active contours that are automatically fitted to the lip region.

Aubrey et al [4] employed a geometrical lip model for VVAD that consisted of landmarks. Given a video sequence of a speaking and dummy, the task was to differentiate speech from non-speech. Their landmarks (constituting the lip model) were fitted to the video data of a speaking person by means of a vigorous Appearance Model (AAM) [5]. For every frame, the 2 standard geometric features, i.e., the width and height of the mouth, were extracted from the positions of the landmarks and submitted to a Hidden Markov Model.

3.3. FACIAL LANDMARKS

Face landmark detection is that the process of finding points of interest in a picture of a person's face. The Chehra detector [7] is used for facial landmark localization in all experiments. It's recently seen ascent within the computer vision community because it's many compelling applications. For instance, we've shown the flexibility to detect emotion through facial gestures, estimating gaze direction, changing facial appearance (face swap), augmenting faces with graphics, and puppeteering of virtual characters.

The Face Detection API doesn't use landmarks for detecting a face, but rather detects a face in its entirety before trying to find landmarks. There are twelve landmarks that are possible to find left and right eye, left and right ear, left and right ear tip, base of the nose, left and right cheek, left and right corner of the mouth, base of the mouth. The landmarks that are available rely on the angle of the face detected.

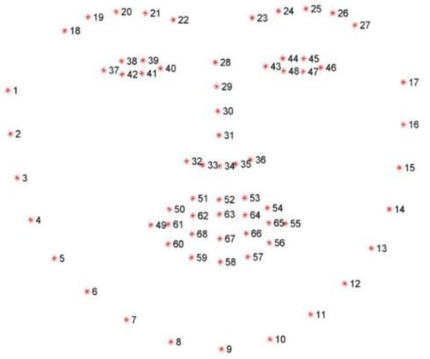

Fig-2: Visualizing the 68 facial landmark coordinates

4. OpenCv LIBRARIES FOR FACIAL LANDMARKS

4.1. DLIB

Dlib is also a recent C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to unravel globe problems. It's employed in both industry and academia during a good selection of domains including robotics, embedded devices, mobile phones, and enormous high performance computing environments. Dlib's open source licensing permits its usage in any application.

The pre-trained facial landmark detector inside the dlib library is used to evaluate the locality of 68 (x, y)-coordinates that map to facial structures on the face.

4.2. PILLOW

Pillow may be a fork of PIL (Python Image Library), started and maintained by Alex Clark and Contributors. It had been supported the PIL code, so evolved to a much better, modern and more friendly version of PIL. It adds support for opening, manipulating, and saving many alternative image file formats. Plenty of things work the identical way because the original PIL. The modules of PIL plays major role in determining the coordinates of facial parts.

Thus, with the help of all the openCV libraries we can able to detect the face in all the frames and can able to draw all the facial features with the ability to locate the facial landmarks. Thus, VVAD is performed more efficiently with the help of openCV than using Viola Jones algorithm.

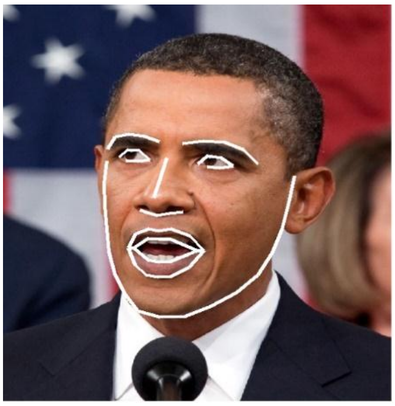

Fig-3:Implementation of opencv libraries on open mouthed image

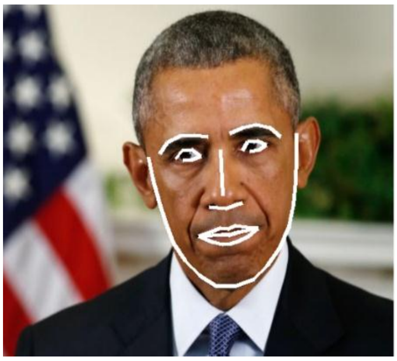

Fig-4: Implementation of opencv libraries on close mouthed image

5. SNR ANALYSIS

A signal-to-noise ratio compares a strength of signal power to a strength of noise power. As long as the incoming signal is strong and well above the noise floor, then the audio will be able to maintain a higher quality which forms the necessity for snr analysis. Hence, in this paper the snr of input signal with low snr value has been improved with visual cues .The improvement in snr with visual cues is higher when compared with snr improvement using audio cues alone. The average improvement in snr for output speech is shown in the following table.

INPUT SNR (db)	OUTPUT SNR(db)				IMPROVEMENT IN db			
	Babble noise	Factory noise	Engine noise	Speech Shaped noise	Babble noise	Factory noise	Engine noise	Speech Shaped noise
10	-7.8809	-5.8103	-7.6948	-7.7684	17.8809	15.8809	17.6948	17.7684
5	-2.9824	-0.9767	-2.8026	-2.874	7.9824	5.9767	7.8026	7.874
0	1.7096	3.538	1.8738	1.8082	1.7096	3.538	1.8738	1.8082
5	5.8559	7.2882	5.986	5.9335	0.8559	2.2882	0.986	0.9335
10	8.9099	9.7607	8.9876	8.9556	1.0901	0.2393	1.0124	1.0444
15	10.5992	10.9694	10.6313	10.6174	4.4008	4.0306	4.3687	4.3826
AVERAGE IMPROVEMENT IN db					5.6532	5.3256	5.6230	5.6351

Table 1-Status of improvement in snr for visual cues

6. CONCLUSIONS

The main goal of this paper is to compare whether high quality speech is produced with help of audio cues or with video cues. This paper proves that as visual cues are not affected by background noise the improvement in quality of received speech is higher when compared with audio cues. This is because audio cues gets easily affected by the acoustic disturbances.

Though this method has improved the snr of received speech, it has some drawbacks. As lip detection is influenced by many condition such as poor lighting condition, if the face deviates from frontal pose and when the face is too far from the camera, the lip feature extraction will become a very difficult task. Thus, this causes the proposed method to perform less efficiently.

REFERENCES

[1] A. Benyassine, E. Shlomot, H. Su, D. Massaloux, C. Lamblin, J. Petit: ITU-T recommendations G.729 Annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications, IEEE Commun. Mag., vol. 35, pp. 64-73, 1997.

[2] D. Valj, B. Kotnik, B. Horvat, Z. Kacic: A Computationally Efficient MelFilter Bank VAD Algorithm for Distributed Speech Recognition

Systems, Eurasip J. Appl. Signal Processing, no. 4, pp. 487-497, 2005.

[3] I. McCowan, D. Dean, M. McLaren, R. Vogt, S. Sridharan: The DeltaPhase Spectrum With Application to Voice Activity Detection and Speaker Recognition, IEEE. trans. Audio Speech Lang. ., vol. 19, pp. 2026-2038, 2011.

[4] Aubrey A, Rivet B, Hicks Y, Girin L, Chambers J, Jutten C (2007) Two novel visual voice activity detectors based on appearance models and retinal fillltering. In: Proceedings of the 15th European Signal Processing Conference, EUSIPCO-2007. pp 2409–2413

[5] Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. Pattern Anal Mach Intell IEEE Trans 23.

[6] Sohn, Jongseo., Nam Soo Kim, and Wonyong Sung. "A Statistical Model-Based Voice Activity Detection." *Signal Processing Letters IEEE*. Vol. 6, No. 1, 1999.

[7] Asthana et al. “Incremental Face Alignment in the Wild”. In: Conference on Computer Vision and Pattern Recognition (2014)