SN Computer Science (2021) 2:10
https://doi.org/10.1007/s42979-020-00395-6

ORIGINAL RESEARCH

# Using Spasmodic Closure Patterns to Simplify Visual Voice Activity Detection

**Ananth Goyal**[1]

Received: 16 June 2020 / Accepted: 9 November 2020 / Published online: 24 November 2020


**Abstract**
While speaking, humans exhibit a number of recognizable patterns; most notably, the repetitive nature of mouth movement from closed to open. The following paper presents a novel method to computationally determine when video data contains a person speaking through the recognition and tally of lip facial closures within a given interval. A combination of Haar-Feature detection and eigenvectors are used to recognize when a target individual is present, but by detecting and quantifying spasmodic lip movements and comparing them to the ranges seen in true positives, we are able to predict when true speech occurs without the need for complex facial mappings. Although the results are within a reasonable accuracy range when compared to current methods, the comprehensibility and simple nature of the approach used can reduce the strenuousness of current techniques and, if paired with synchronous audio recognition methods, can streamline the future of voice activity detection as a whole.

**Keywords** Voice Activity Detection · Computer Vision · Lip Detection

## Introduction

Current voice activity detection (VAD) relies heavily on audio cues. The non-convoluted approach to use auditory benchmarks to predict when speech is occurring has made its way into several recent studies [1–3]. However, VAD struggles to detect true speech when multiple speakers are involved or a strong background noise is present [4], because of the inability effectively to compute and isolate the audio signals from each other.

Visual voice activity detection (VVAD), a subset of VAD, can be used in tandem with auditory techniques or as a standalone method. Given the current progress with intelligent systems, face detection software, and its contemporary subset, facial recognition, have practically become a standard in modern technology [5]. In the past decade alone, several new methods to detect faces and its individual components have surfaced [6, 7]. The usage of automated tracking algorithms, such as active lip shape models [8] and variance-based techniques [9], have made it easy to detect when an individual is speaking. Applications of visual voice activity detection (VVAD) range from automated video extraction, anti-cheating software, speaker recognition in an audio intense situation, machine learning training for complex facial tracking, etc. The following paper presents a comprehensible approach to detect when speech is occurring, along with detecting when a specific individual is speaking among many (conference calls, video chats, ceremonies, lectures, etc.)

Although the proposed method, Lip Closure Quantification (LCQ) is independent, it requires two separate algorithms prior to its activation. The first is face detection and recognition, to ensure that the algorithm will only run when the target speaker is on video. The second component is lip detection, which returns a numerical value for when the lips are closed within an interval as well as the coordinates for their location.

## Related Work

Previous work with VVAD that is relevant to this study will be briefly referenced and summarized.

✉ Ananth Goyal
ananthgoyal@gmail.com

[1] Dougherty Valley High School, San Ramon, CA, USA

### Lip-Based Geometric Approaches

A study done by Sodoyer et al. [10] aimed to establish a relationship between lip activity and speech activity to effectively improve contemporary methods of VAD. Two individuals engaged in a face to face discussion primarily using spontaneous dialogue. By looking at recognizable humanistic patterns and by characterizing lip movements from facial mappings, their audio-visual recognition system flourished in varying environments and situations.

The approach used by Liu et al. [11] involved a novel lip extraction algorithm which combined rotational templates and prior shape constraints, with the introduction of active countours. By amplifying the strength of this technique with audio voice detection and adaboosting, they received low error rates and affirmative results on their tested video clips from the XM2VTS dataset and several youtube videos.

In the study done by Navarthana et al. [12], visual voice activity detection was enhanced with the inclusion of viewpoint variation. They looked at the variance between the speaker's profile and and frontal views on the freely available CUAVE dataset; by using their approach and a Gaussian mixture model-based VVAD framework, their results appear to be useful future work in multi-model human computer interaction.

### Comparison with Other Methods

Due to the extreme variance in approaches and low number of available datasets, it makes it difficult to thoroughly compare the proposed method with existing approaches. To maintain the technical validity of this paper, a set of results were compared to pre-recorded footage on the publicly available LiLir dataset.

The study done by Qingju Liu et al. [13] attempts to reduce signal interference with Visual voice activity detection. While the study is not entirely focused on the detection itself, the publicly available dataset and results make it easy to compare with the proposed approach.

Additionally, the results found in the work done by Aubrey et al. [14] on VVAD with optical flow, as well as the comparison found in Liu et al.'s with SVM [15] (information about the actual approach is unknown to the author, however its resultant data can be compared with LCQ.)

## Pre-LCQ Setup

OpenCV was used to facilitate a majority of the methods used in this study. While the primary algorithm only involves lip motion detection, it is dependent on the face detection technology to isolate the components of video data that only contain footage of the target speaker.

### Haar-Features and Facial Detection

Prior to facial recognition of the target speaker, a Haar-Feature method, known as the Viola–Jones approach [16, 17], is used to detect the presence of a face. By looking at noticeable differences in hue intensities ($I$), specifically dark and light shadings, patterned edges can be detected [18]. In an ideal situation, the resultant differential would be 1 and only ones and zeros would be produced [19]; however, typically a more error-prone matrix (like the one shown below) would be generated every single time an edge has been detected. Although grayscale pixel intensity levels are between (0, 255) like standard RGB metrics, those values are divided by the outer limit (255) to keep it within 0 and 1.

$$\begin{bmatrix} 0.1 & 0.3 & 0.8 & 0.9 \\ 0.2 & 0.2 & 0.7 & 0.8 \\ 0.1 & 0.1 & 0.6 & 0.6 \\ 0.1 & 0.3 & 0.8 & 0.8 \end{bmatrix} \tag{1}$$

When the algorithm is deciphering whether the edge is a constituent of a face, it calculates the difference ($\Delta$) in pixel intensity [20] between the sets of columns. The closer the difference is to 1, the more likely a Haar feature has been detected [21].

$$\Delta = \frac{1}{n}\sum_{dark}^{n} I(x) - \frac{1}{n}\sum_{white}^{n} I(x). \tag{2}$$

Since the LCQ setup was done through Open CV, the threshold point for where the Haar-Feature differential produces a face was not a known number, but given the results, it is assumed to be between 0.6 and 0.99 (Figs. 1 and 2).

### Eigenvectors and facial recognition

A variant eigenface model is used to recognize the target speaker [22, 23]. Given the mean ($\mu$) face, we are able to accumulate slight differentials of facial structures [23] within a dataset $X$ to improve the model's ability to recognize a specific face and compute a covariance matrix $M$ [24]. By computing every eigenvector and its corresponding eigenvalue, we can then arrange them in decreasing order

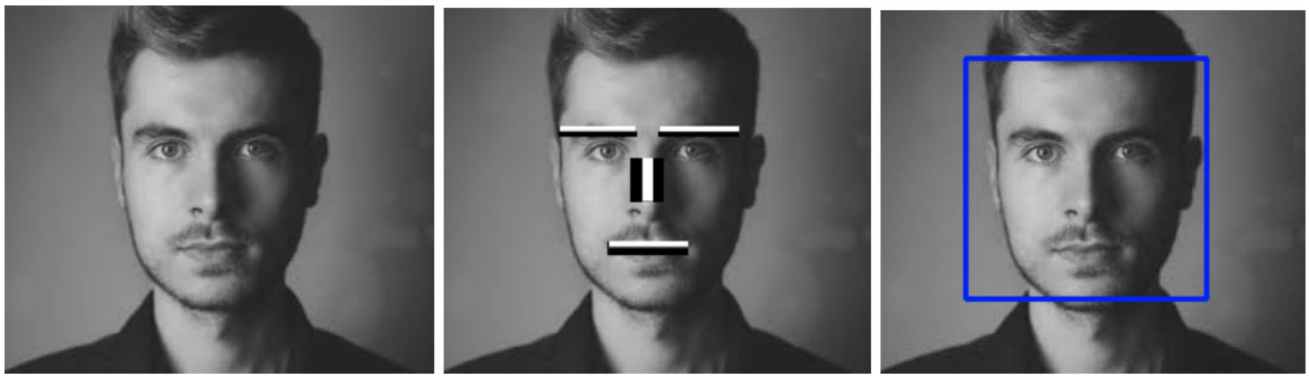

**Fig. 1** Progress from initial Haar-Feature detection from calculated intensity differences to completed facial detection

**Fig. 2** Progress from initial Haar-Feature detection from calculated intensity differences to completed lip closure detection

and reconstruct the original dataset given the calculated vector set $W$ [25].

$$M = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T, \tag{3}$$

$$X = WW^T(x - \mu) + \mu. \tag{4}$$

The uniqueness of any particular face is stored as its relative variance between the mean or average face [26]. When the target individual is present in the video data, the eigenvalues are compared and matched to justify the presence of the correct speaker.

### Lip Detection by Using Segments of Haar-Features

Using Haar-Features, we can focus solely on lip detection, which is the basis for the current study. When the lips are closed, the algorithm will look for a strong horizontal differential in the matrix such that it will fit the requirements to be considered a set of lips.

The structure that appears around the lips is based on the matrix produced from the Haar-Feature. Similar to facial detection ("Haar-Features and Facial Detection"), lips are detected by the differential between white and dark pixels, except with horizontal calculations rather than vertical (Shown in Fig. 2). If the difference is greater than the minimal condition, the algorithm will assume that lips are present in the video data.

## Lip Closure Quantification

The primary proposed technique, Lip Closure Quantification (LCQ), is a pattern analysis algorithm used to determine whether the target individual is speaking by counting the number of closure occurrences in an established time interval as a basis for its prediction. When the total number of closure occurrences ($\sigma_n$) within an interval fit within an approximated range of false and surplus detections, ($\sigma_n$) will be inserted into a dataset ($\sigma$). Both ($\sigma_{\min}$) and ($\sigma_{\max}$) are predetermined bounds, but are constantly adjusting (mentioned in Sect. 4.2) to reduce the margin of error.

If the condition is not met, there are two possibilities of negatives: the first being that $\sigma_n$ exceeds the range, i.e., the mouth is closed. The second is that $\sigma_n$ fails to meet the sufficient

requirement for it to be considered a positive; this could be a result of a resting open mouth, sporadic facial alignment with the camera, false lip detections, etc. The initial time interval is repeated; if the number of occurrences within such interval falls within the detection range, then the whole time interval $t_n$ is marked as positive and inserted into the dataset ($t$). For data accumulation, a two-dimensional matrix (T) contains every subsequent dataset $t$ (referenced in "Adjusting Frequential Bounds"). The difference between a single mouth closure and a continued closure is the difference between a positive and negative. When the mouth is closed throughout the interval, $\sigma_{\max}$ will naturally be exceeded. An example is shown below.

## Dealing with False Negatives

A false negative is defined as a situation when the target individual is speaking but it is not recognized. In an entirely positive scenario the standard interval time $\alpha$ will define every interval until the first false negative.

$$\alpha = \frac{1}{(x-1)} \sum_{n=1}^{x} \left[ t_n - t_{(n-1)} \right]. \tag{5}$$

The difference between the latest time stamp and the next positive will no longer be $\alpha$. Typically if the speaker moves their head, takes a brief pause, or wipes their face within an interval, the entire stretch will be disregarded as a negative and excluded from the set. To effectively reduce the number of false negatives, $t$ is scanned for minimal discrepancies and compared to a set threshold ($\lambda$). The most effective way to differentiate between a true negative and false negative is to calculate the number of gaps within two positive intervals. When the number of gaps is greater than the threshold value, it is assumed that the individual was not speaking in that time frame. However, if the number of gaps is less than or equivalent to the threshold value (shown below), speech was undetected but was still occurring,and should be inserted into the dataset (Fig. 3).

$$\lambda \geq \frac{(t_x - t_{x-1})(x-2)}{\sum_{n=1}^{x-1} \left[ t_n - t_{(n-1)} \right]}. \tag{6}$$

## Analyzing Direct Accuracy

The direct accuracy (DA) is an accumulation of instantaneous Boolean data. Each binary data point is represented as $\xi$.

$$\sum_{i=1}^{n} (5\xi - \xi^2 - 2). \tag{7}$$

(DA) is determined using the binary output while inserted as $d \in D$ such that $a = n_{\max}$ and $c$ is the true binary value of that respective iteration.$\frac{c_i+1}{d_i+1}$ is inputted as $\xi$.

$$\sum_{i=1}^{n} \left( 5 \left[ \frac{c_i+1}{d_i+1} \right] - \left[ \frac{c_i+1}{d_i+1} \right]^2 - 2 \right). \tag{8}$$

## Adjusting Frequential Bounds

Given the varying frequencies of occurrences in any $t_n$, the quality of the (LCQ) outputs is dependant on the accuracy of the bounds $\sigma_{\min}$ and $\sigma_{\max}$. To enhance the computational accuracy of the algorithm, an intelligible self-learning method utilizing $\sigma$ is employed. By setting a compensation factor $\omega$ (initially a random value within close proximity of the original bounds), we are able to reduce the margin of error by equally and continually adjusting the bounds of ($\sigma_{\min}$, $\sigma_{\max}$) accordingly.

$$\left( \frac{(1-\omega)}{y} \sum_{n=1}^{y} \sigma_n, \frac{(1+\omega)}{y} \sum_{n=1}^{y} \sigma_n \right). \tag{9}$$

Once the new frequential bounds are computed (if the need to do so arises), they are updated for future comparisons. If a true negative arises, then the bounds are not updated and will maintain their previous value until more timestamps are inserted into $t$.

## Accumulating Video Data After (LCQ)

Once the subsequent calculation is completed and $t$ is returned from (LCQ), it is appended into the following two-dimensional dataset $T$, which, when completed, contains all the stored positive intervals in which the target individual was speaking.

$$T = \begin{bmatrix} t_{1_1} & t_{1_2} & t_{1_3} & \dots & t_{1_w} \\ t_{2_1} & t_{2_2} & t_{2_3} & \dots & t_{2_x} \\ t_{\dots_1} & t_{\dots_2} & t_{\dots_3} & \dots & t_{\dots_y} \\ t_{n_1} & t_{n_2} & t_{n_3} & \dots & t_{n_z} \end{bmatrix} \tag{10}$$

Because of the inclusion of $\lambda$ to bypass false negatives, the only important values in the datasets within $T$ are the initial and final elements. These time bounds can be used to accumulate the total amount of video data that will be extracted ($V$), given the length of the dataset ($z$).

$$V = \sum_{n=1}^{z} \sum_{i=1}^{x} [T_{ni} - T_{ni-1}]. \tag{11}$$

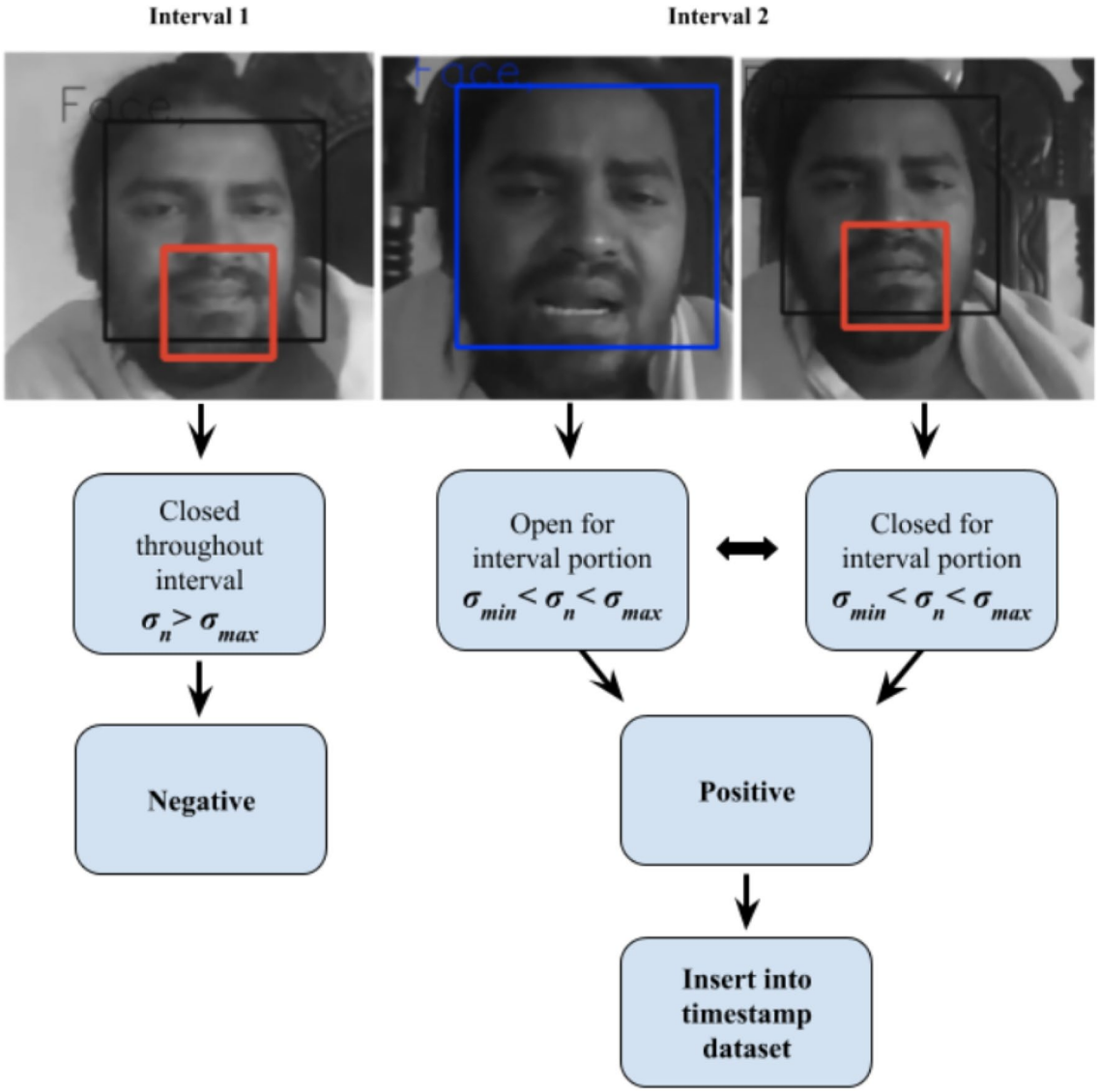


**Fig. 3** Differences in interval types (negative and positive). Negative: closed face maintained throughout the duration of the interval. Positive: both types of facial detections for LCQ ($\sigma_n$ and $\sigma_n + 1$)

## Results

The results are distinguished into individual segments. The algorithm's pre-LCQ ability was tested once with the stock face images made available from the LFW dataset. Its LCQ performance was tested twice, first with a sample video conference lecture to test the algorithm's ability to use pre-LCQ and LCQ simultaneously and, second, with the video clips available from the LiLir video dataset for thorough comparisons to other methods.

### Pre-LCQ Results

The LFW dataset contains 13233 sample images of 5749 people's faces. The dataset is alphabetically organized by name; 100 images were used from each letter. The following data shows the pre-LCQ's accuracy in detecting both the face and lips of the sample individual (Table 1).

**Table 1** Calculated scores of Pre-LCQ on the LFW Image dataset by letter

| Segment | Mouth | Face | AVG |
|---|---|---|---|
| A–G | 0.95 | 1.00 | **0.975** |
| | 0.97 | 0.98 | **0.975** |
| | 1.00 | 1.00 | **1.000** |
| | 1.00 | 0.95 | **0.975** |
| | 0.96 | 0.98 | **0.970** |
| | 0.96 | 0.97 | **0.965** |
| | 0.95 | 0.97 | **0.960** |
| H–N | 1.00 | 1.00 | **1.000** |
| | 0.97 | 0.97 | **0.970** |
| | 0.94 | 0.97 | **0.955** |
| | 1.00 | 0.95 | **0.975** |
| | 1.00 | 0.98 | **0.990** |
| | 0.94 | 1.00 | **0.970** |
| O–T | 0.93 | 0.98 | **0.955** |
| | 0.97 | 1.00 | **0.985** |
| | 0.98 | 1.00 | **0.990** |
| | 1.00 | 1.00 | **1.000** |
| | 1.00 | 0.97 | **0.985** |
| | 1.00 | 1.00 | **1.000** |
| U–Z | 0.97 | 1.00 | **0.985** |
| | 1.00 | 0.98 | **0.990** |
| | 1.00 | 1.00 | **1.000** |
| | 1.00 | 1.00 | **1.000** |
| | 0.97 | 0.99 | **0.980** |
| | 1.00 | 1.00 | **1.000** |

The important components of the data; either a final value, or an average are in bold

**Table 2** Continually adjusted direct accuracy (DA), with varying values in $\sigma$ for every fourth time stamp. The final value is the accuracy after the removal of false negatives

| Time stamps | $\sigma_n$ | DA |
|---|---|---|
| 13.79 | 51 | 1 |
| 65.39 | 72 | 0.8 |
| 145.63 | 52 | 0.66 |
| 189.35 | 56 | 0.75 |
| 286.69 | 42 | 0.64 |
| 345.77 | 43 | 0.67 |
| 407.61 | 37 | 0.68 |
| 473.53 | 46 | 0.67 |
| 510.67 | 36 | 0.70 |
| – | – | **0.98** |

The important components of the data; either a final value, or an average are in bold

## Pre-LCQ and LCQ Integrated Results

A sample 510 second conference call video was inputted into the algorithm with the lecturer recognized as the target speaker. The training method for recognizing the particular face used Haar-Feature for the detection and eigenvectors for the recognition like previously noted. $\alpha$ was set at 10 s; the number of closure motion counts within each interval is shown below (For simplistic purposes, we arbitrarily chose to show every fourth interval; however the complete progression is referenced in the graphs at the end of this section) (Table 2).

The direct accuracy will continually fluctuate as each false negative accumulates. As time increases, the direct accuracy will approach a certain number, in this case 0.7, indicating an estimate of the average frequency of success to error in that particular interval set. The completed accuracy, 0.98, is the final accuracy after the detected false negatives have been calculated and accounted for. In real time, the false negatives cannot be recognized, as their detection requires the new data available in the next interval. However, if LCQ is being used on pre-recorded footage or for analytical work, not in the real time, then its final accuracy would be a stronger indicator of its performance.

## LCQ Comparisons

When testing with the LiLir dataset, which consisted of several speech utterances (with emphasis on the visual data), results were compared to the findings from Liu et al., Aubrey et al., and SVM methods. We define $\epsilon_n$ as the false negative rate, $\epsilon_p$ as the false positive rate, and $\epsilon$ as the total error rate.

The SVM comparison approach presented in Liu et al.'s work had the strongest false positive rate, yet also the highest false negative rate. Liu et al.'s approach had the strongest results overall, but also involved a more convoluted methodology involving adaboosting and interference removal.

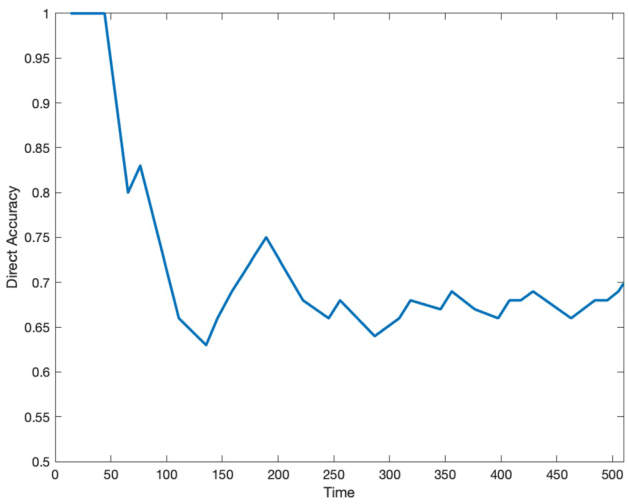


**Fig. 4** Direct accuracy (prior to false negative removal) with respect to time

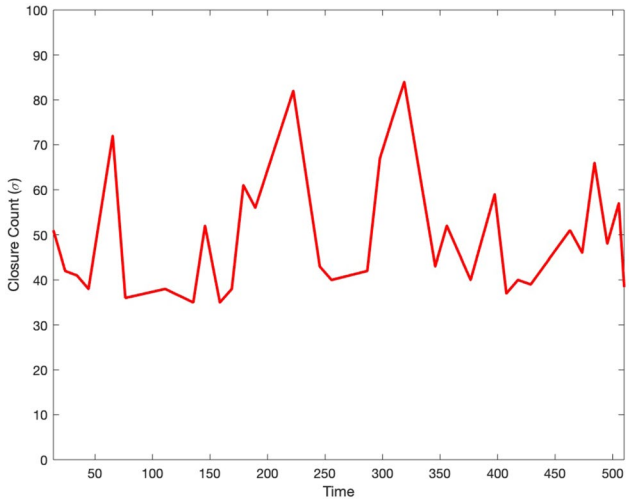


**Fig. 5** Closure counts ($\sigma$) (prior to false negative removal) with respect to time

**Table 3** Calculated scores compared with Liu et al. and Aubrey et al.) with the LiLir dataset

| Method | $\epsilon_n$ | $\epsilon_p$ | $\epsilon$ |
|---|---|---|---|
| Liu et al. | 0.25 | 0.03 | **0.11** |
| Aubrey et al. | 0.32 | 0.01 | **0.12** |
| SVM | 0.72 | 0.01 | **0.27** |
| LCQ | 0.31 | 0.02 | **0.12** |

The important components of the data; either a final value, or an average are in bold

When compared to the three VAD methods shown above, LCQ performs in the high performance range with an error rate of 0.12 (Figs. 4 and 5, Table 3).

## Conclusion and General Discussion

In this paper, we presented a novel approach to enhance visual voice activity detection without the need for audio data or complex facial mappings. When interpreting the results, it is important to consider all aspects ("Pre-LCQ Results", "Pre-LCQ and LCQ Integrated Results", "LCQ Comparisons") and the overall comprehensibility of the approach used.

With the Pre-LCQ results, we were able to test the efficacy of the concurrent face and lip detection algorithm. On average, the variance between the accuracy on both the lips and face detection was within (0.965, 1.000) indicating a strong performance. With the integrated results, we found LCQ's real-time ability to be less accurate than its post-LCQ ability, however, still reasonable nonetheless. When considering the usage of LCQ in future technologies and research, it is likely to be more successful with analytic work than real-time tasks. While the ability to automatically account for false negatives after interval data has been stored is useful, it can reduce the frequency of highly accurate results until post-analytic work is done. With regard to LCQ comparisons, it performed on par with other methods, superior in most aspects. With a total error score of 0.12, it is slightly less effective than Liu et al.'s method; however, its simple nature has many benefits, such as less CPU strain and easier development.

Although the current findings cover a wide variety of effectiveness, from pre-LCQ, LCQ, and integrated results, they are unable to predict its performance in more variable heavy situations involving large crowds, situations with several faces within a single frame, or instances when simultaneous audio-visual voice activity detection is required. These cases, most significantly the latter, given that LCQ is solely a visual-based approach, will make way for future research. Testing LCQ's or a similiar VAD method's performance with audio signals could potentially improve its overall performance and thus reduce the current error rate. While current work in VAD is beginning to digress to more complex approaches such as complete facial mappings, lip reading, and speaker recognition, it is important to establish a strong and effective basis to further enhance the results found in such methods.

**Acknowledgements** I would like to thank the editorial board and review team from the SN Computer Science Research Journal for their kind and constructive feedback. I would also like to thank Professor Jeffery Ullman, Mr. Sudhir Kamath, Mr. Robert Gendron, and Mrs. Katie MacDougall for their continual support throughout my research work.

**Funding** No funding was recieved to support and conduct research. Ananth Goyal authored the entirety of this paper.

**Data Availability** All datasets used are publicly available: the LFW Image dataset and LiLir lip tracking dataset.

## Compliance with Ethical Standards

**Conflict of Interest** The authors declare no competing interests.

## References

1. Chang J-H, Kim NS, Mitra SK. Voice activity detection based on multiple statistical models. IEEE Trans Signal Process. 2006;54(6):1965–76.
2. Ghosh PK, Tsiartas A, Narayanan S. Robust voice activity detection using long-term signal variability. IEEE Trans Audio Speech Lang Process. 2010;19(3):600–13.
3. Ramırez J, Segura JC, Benıtez C, De La Torre A, Rubio A. Efficient voice activity detection algorithms using long-term speech information. Speech Commun. 2004;42(3–4):271–87.

4. Joosten B, Postma E, Krahmer E. Visual voice activity detection at different speeds. Auditory-Visual Speech Processing (AVSP) 2013, 2013.
5. Dang K, Sharma S. Review and comparison of face detection algorithms. In: International Conference on Cloud Computing, Data Science & Engineering-Confluence. 2017;7:629–33.
6. Yang S, Luo P, Loy C-C, Tang X. "Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5525–5533, 2016.
7. Li H, Lin Z, Shen X, Brandt J, Hua G. A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5325–5334, 2015.
8. Caplier A. Lip detection and tracking. In: Proceedings 11th International Conference on Image Analysis and Processing. pp 8–13, 2001.
9. Wang L, Wang X, Xu J. Lip detection and tracking using variance based haar-like features and kalman filter. In: 2010 Fifth International Conference on Frontier of Computer Science and Technology. pp 608–612, 2010.
10. Sodoyer D, Rivet B, Girin L, Savariaux C, Schwartz J-L, Jutten C. A study of lip movements during spontaneous dialog and its application to voice activity detection. J Acoust Soc Am. 2009;125(2):1184–96.
11. Liu Q, Wang W, Jackson P. A visual voice activity detection method with adaboosting, 2011.
12. Navarathna R, Dean D, Sridharan S, Fookes C, Lucey P. Visual voice activity detection using frontal versus profile views. In: 2011 International Conference on Digital Image Computing: Techniques and Applications, pp 134–139, 2011.
13. Liu Q, Aubrey AJ, Wang W. Interference reduction in reverberant speech separation with visual voice activity detection. IEEE Trans Multimed. 2014;16(6):1610–23.
14. Aubrey A, Hicks YA, Chambers J. Visual voice activity detection with optical flow. IET Image Process. 2010;4(6):463–72.
15. Platt J. Sequential minimal optimization: a fast algorithm for training support vector machines, 1998.
16. Vikram K, Padmavathi S. Facial parts detection using viola jones algorithm. In: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp 1–4, IEEE, 2017.
17. Gupta A, Tiwari R. Face detection using modified viola jones algorithm. Int J Recent Res Math Comput Sci Inform Technol. 2014;1(2):59–66.
18. Kolsch M, Turk M. Analysis of rotational robustness of hand detection with a viola-jones detector. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol 3, pp 107–110, IEEE, 2004.
19. Castrillón M, Déniz O, Hernández D, Lorenzo J. A comparison of face and facial feature detectors based on the viola-jones general object detection framework. Mach Vis Appl. 2011;22(3):481–94.
20. Wang Y-Q. An analysis of the viola-jones face detection algorithm. Image Process Line. 2014;4:128–48.
21. Jensen OH. Implementing the viola-jones face detection algorithm. Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Denmark: Lyngby; 2008.
22. Turk M, Pentland A. Face recognition using eigenfaces. In: Proceedings of 1991 IEEE computer society conference on computer vision and pattern recognition, pp 586–587, 1991.
23. Yang M.-H, Ahuja N, Kriegman D. Face recognition using kernel eigenfaces. In: Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 1, pp 37–40, IEEE, 2000.
24. Barnouti NH, Al-Dabbagh SSM, Matti WE, Naser MAS. Face detection and recognition using viola-jones with PCA-LDA and square Euclidean distance. Int J Adv Comput Sci Appl (IJACSA). 2016;7(5):371–7.
25. Duda RO, Hart PE, Stork DG. Pattern classification. Amsterdam: Wiley; 2012.
26. Kshirsagar V, Baviskar M, Gaikwad M. Face recognition using eigenfaces. In: 2011 3rd International Conference on Computer Research and Development, vol 2, pp 302–306, IEEE, 2011.