# DETERMINATION OF TIME OFFSET BETWEEN AUDIO AND VIDEO IN ONLINE REPOSITORIES OF VIDEOS (YOUTUBE)

*Vikram Voleti*

# DETERMINATION OF TIME OFFSET BETWEEN AUDIO AND VIDEO IN ONLINE REPOSITORIES OF VIDEOS (YOUTUBE)

*Report submitted to*
*Indian Institute of Technology Kharagpur*
*for the award of the degree*

*of*

**Bachelor of Technology**
**in Electrical Engineering**

*by*

**Vikram Voleti**


**DEPARTMENT OF ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR**
**May 2013**

# DECLARATION

I certify that

a. the work contained in this report is original and has been done by me under the guidance of my supervisor(s).
b. the work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute in preparing the report.
d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
e. whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

……..

Vikram Voleti
IIT Kharagpur

# CERTIFICATE

This is to certify that the report entitled, "**Determination of Time Offset Between Audio And Video in Online Repositories of Videos (Youtube)**" submitted to the Indian Institute of Technology Kharagpur, India, for the award of the degree of Bachelor of Technology by **Mr. "Vikram Voleti**" is a record of bonafide research work carried out by him under my supervision and guidance. The thesis has reached the standard fulfilling the requirements of the regulations related to the degree. The results embodied in the thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

……...……

Dr. Rajiv Sahay
Department of Electrical Engineering
Indian Institute of Technology Kharagpur
Kharagpur 721302 India

Date:

# ACKNOWLEDGMENTS

It is a matter of great privilege for me to be able to express my deep sense of gratitude to my project supervisor Dr. Rajiv Sahay for his unwavering support, insightful suggestions, encouragement and the fruitful discussions throughout the duration of my project. He has always made himself available for any sort of discussion and support. I have learned a great deal in the areas of multimedia, sound processing, computer vision and signal processing over the past year through his uninhibited help and motivation.

I would also like to thank Dr. Sourjya Sarkar, for his exceedingly helpful tips, and the highly informative tutoring sessions and discussions. I would also like to thank Dr. K. S. Rao for his help in kick-starting this project.

I am also grateful to all faculty members of the Department for their help, suggestions and comments during the presentations and throughout the tenure of the work.

Finally, I would like to express my thanks and love to my parents for their support and encouragement.

Vikram Voleti

# CONTENTS


Title Page..... i
Declaration..... ii
Certificate..... iii
Acknowledgement..... iv
Contents..... v
1. Motivation..... 1
2. Abstract..... 1
3. Literature Review..... 1
4. Detection of Bilabial Consonants in Audio..... 2
   4.1 Mel Frequency Cepstral Coefficients..... 2
   4.2 Gaussian Mixture Models..... 3
   4.3 Determining Location of Bilabials in Test Speech Sample..... 4
      4.3.1 Calculation of probability of data point X belonging to a GMM J..... 4
      4.3.2 Advantages..... 4
      4.3.3 Disadvantages..... 4
5. Detection of Lip Closure in Video..... 5
   5.1 The Procedure..... 5
   5.2 Haar Features..... 5
   5.3 Viola-Jones Algorithm to detect face and mouth regions..... 6
   5.4 Lip Detection using Colour Filtering..... 6
   5.5 Advantages..... 6
   5.6 Disadvantages..... 6
6. Results..... 7
   6.1 Detection of Bilabials in Audio..... 7
      6.1.1 Testing on sample belonging to training data..... 7
      6.1.2 Testing on new sample data..... 8
   6.2 Detection of Lip Closure in Video..... 9
      6.2.1 Lips not closed..... 9
      6.2.2 Lips closed..... 10
   6.3 Combining Detection of Bilabials in Audio and Video..... 11
      6.3.1 Eliminating Lip Closure Points with No Voice using Signal Energy..... 11
      6.3.2 Combining Audio and Video..... 11
7. Vowel Detection..... 12
8. Error in Lip Closure Detection in Video..... 13
   8.1 Error due to Viola – Jones Algorithm used to detect face..... 13
   8.2 Error in Colour Detection..... 14
   8.3 Error in Pixel Column..... 15
   8.4 Error Due to Frame Rate..... 15
9. Error in Bilabial Detection in Audio..... 16
   9.1 Error due to MFCC..... 16
   9.2 Error in Training Data..... 16
   9.3 Error in the Gaussian Mixture Model..... 17
      9.3.1 Iterations of the Expectation Maximization Algorithm..... 17
      9.3.2 Over-fitting & Regularization..... 17
      9.3.3 Number of Gaussian distributions in the Mixture..... 17
10. References..... 18

# 1. MOTIVATION

The motivation behind this project is to be able to filter online videos such as those on YouTube, so as to be able to choose between only the best ones. The solution offered by this project would help enable us to analyse the time offset between the audio signal and video signal in the video, thereby enabling us to eliminate those beyond a reasonable threshold. This can help remove various redundant videos that pop up in searches.

# 2. ABSTRACT

In this report, a method to determine the time offset between the audio and video signals, by measuring the points in time where bilabial consonants are detected in the audio and the video, is presented. Bilabial consonants are those consonants which are articulated by both the lips. By noting the times at which they occur, the time offset between the audio and video signals can be approximately calculated. For the video signal, an algorithm has been designed that detects closure of the lips of the speaker. For the audio signal, a Gaussian Mixture Model (GMM) has been trained with audio samples of bilabials uttered by different speakers, and another GMM with audio samples that don't contain bilabials. These GMMs were then used to calculate the probability of the occurrence of bilabials in the audio signal being tested.

# 3. LITERATURE REVIEW

Bilabial consonants are those consonants that are articulated by both the lips. This implies that these sounds require the lips of the speaker to close. For example: the sounds 'pa,' 'ma,' 'ba,' represented as *p*, *b* and *m* respectively in the International Phonetic Alphabet. The initial part of this B. Tech. Project has been to detect occurrence of bilabial consonants in videos. There are two aspects to this problem: the audio, and the video.

In the audio aspect of the problem, we have a speech signal from the speaker in which we need to find patterns that correspond to bilabials. To achieve this Gaussian Mixture Models were used to classify those sounds that are bilabials and those which are not. Support Vector Machines were also implemented and tested, but Gaussian Mixture Models proved to be better tools for the purpose.

The features that were used to train the Gaussian Mixture Models are Mel Frequency Cepstral Coefficients. These features are used to characterize sound modelled on the human auditory response. The Mel Frequency scale is different from the linear frequency scale in that it gives more weightage to the human ear audible frequencies, and lesser to extraneous frequencies. Thus it is a suitable parameter to train the Gaussian Mixture Models. It is globally accepted, and mentioned in countless research papers pertaining to speech recognition, that 13 Mel Frequency Cepstral Coefficients be used. Thus the feature that is input to the GMM is one 13-dimensional vector for every window of speech considered.

In the video aspect of the problem, it is necessary to find out those places where the lips of the speaker close while uttering some speech. This was achieved by designing an algorithm that employs image processing techniques. It was found that the most popular and trusted method for face detection is the Viola-Jones Algorithm or the Haar Classifier. This algorithm uses Haar features to classify whether a given image is a face or not. Both the face and the mouth are detected using Haar features.

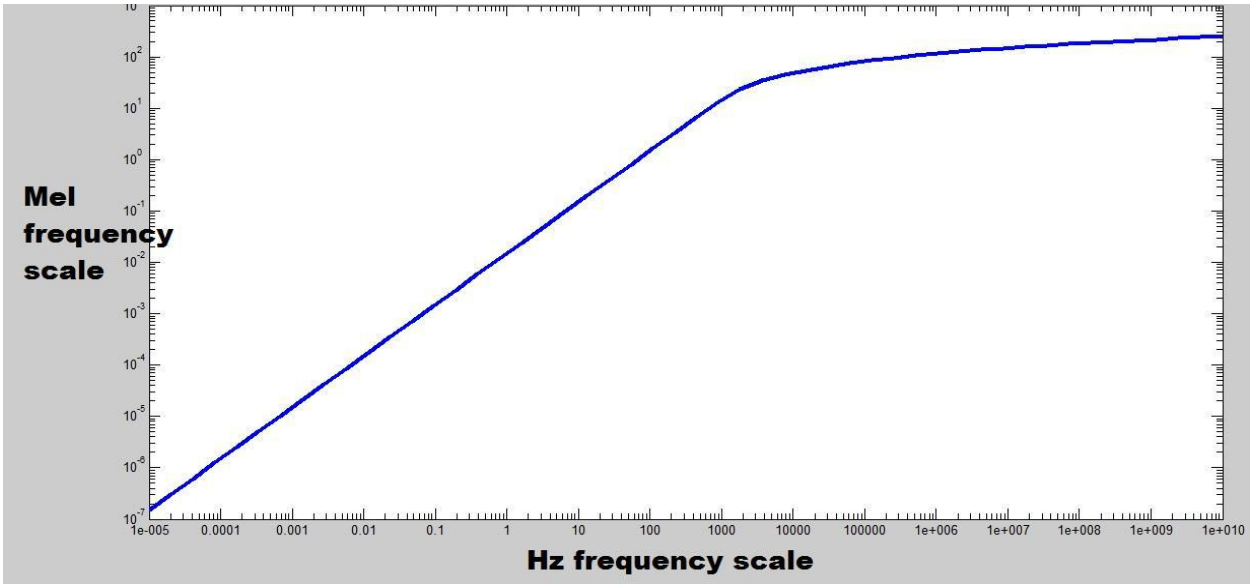
The lips are segmented from the mouth using Colour Filtering, whereby the mouth image is filtered on the basis of the colour range that lips belong to. This method could prove tricky, but gives accurate results when implemented correctly. Finally, lip closure is achieved using a new procedure that was designed during the course of this project. This technique was not found in any research papers that were scanned through.

# 4. DETECTION OF BILABIAL CONSONANTS IN AUDIO

The detection of occurrence of bilabial consonants in audio made use of two very important mathematical tools called Mel frequency cepstral coefficients and Gaussian mixture models.

## 4.1 Mel Frequency Cepstral Coefficients

The mel-frequency scale is a nonlinear frequency scale that is, qualitatively, more spread over those frequencies within the hearing range, and much lesser over higher frequencies. A graph is shown below that represents the relation between the Hertz frequency scale and the Mel frequency scale.

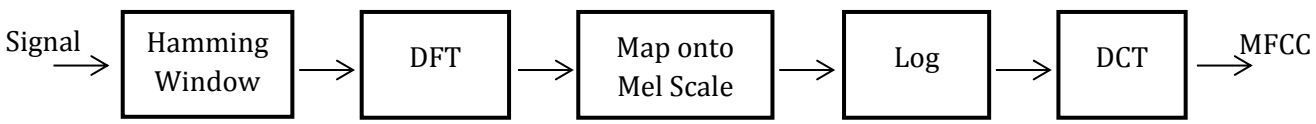



The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel-frequency scale. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip. The frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. Hence, they are well suited to parameterize the human voice. For the task at hand, the MFCCs calculated are 13-dimensional feature vectors.

The procedure for calculating the MFCC's is:

1) Use a Hamming window to limit the 30msec long portion of speech for which MFCC's are being calculated.
2) Find out the Discrete Fourier Transform of that window of speech.
3) Map the square of the DFT, i.e. the Power Spectrum obtained from the linear scale onto the Mel scale for better modelling of the human ear auditory response.
4) Take the Logarithm of the power values at every discrete Mel frequency.
5) Determine the Discrete Cosine Transform of the Log of Mel Powers.

The amplitudes of the resulting spectrum are the Mel Frequency Cepstral Coefficients.



## 4.2 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. For this project, the spectral features that were used to train the GMM are Mel Frequency Cepstral Coefficients.

The two parameters that characterize a Gaussian distribution are the mean and the variance. Since a mixture of Gaussians is being constructed, and additional parameter must be considered: number of centres. The various parameters that are also used to construct GMM are:

- Number of Centres in the model = 16
- Number of dimensions of feature being input, i.e. MFCC = 13
- Number of iterations of the Estimation Maximisation algorithm = 50

Thus, a GMM can be characterized by 4 parameters: Number of Gaussians, Means of the Gaussians, Variances of the Gaussians, and the individual weights associated with the Gaussians. These parameters are calculated by fitting the Gaussians onto a graph of the training data. The iterative algorithm used to determine these parameters is called the Expectation Maximization (EM) algorithm.

The procedure for constructing a GMM is:

1) Initialize the parameters of the GMM using a k-Means algorithm. This is used as a starting point for the EM algorithm, and separates the training data into that number of clusters as the number of centres in the GMM.
2) Use the EM algorithm to determine the parameters of the GMM. This is an iterative algorithm that repeatedly readjusts the values of the parameters based on the posterior probabilities of each cluster with respect to the mixture. The limiting parameter or error term used is the sum of the negative log-likelihood of the activations of every cluster.

Here, the entire training data is considered as the mixture, $\boldsymbol{J}$, and each cluster is represented by $\boldsymbol{X}$. Thus, the posterior probability of each cluster is $\boldsymbol{P(J|X)}$. Every new centre is determined as a proportion of the product of the posterior probability of that cluster and the cluster data.

The activation of a cluster $\boldsymbol{X}$ is the conditional probability of $\boldsymbol{X}$ on the mixture, i.e. $\boldsymbol{P(X|J)}$. This activation, multiplied by the prior of the cluster produces the current likelihood of the cluster $\boldsymbol{X}$. The error term, is the sum of the negatives of the log of this likelihood for every cluster.

$$\boldsymbol{e = - sum( log( prior(X) * P(X|J) ) )}$$

The EM algorithm is iterated until either the error value does not change beyond a threshold value, or the maximum number of iterations is exceeded.

### GMMs for Bilabial and Non-Bilabial Speech Samples

Initially, using speech samples from various speakers, those portions of the speech that comprised a bilabial were manually segmented, and their MFCCs were calculated. Using these MFCCs, a weighted sum of the 13 coefficients of the feature vectors was calculated, and a higher-dimensional Gaussian curve with multiple centres was approximated to fit the different feature vectors on the 13-dimensional plane. GMM parameters were estimated from the training data using the iterative Expectation-Maximization (EM) algorithm. Thus, a GMM was formed using the training data. Let us call this GMM1. In the same way, another GMM was formed using the MFCCs calculated from those speech samples that excluded the bilabials. Let us call this GMM2.

## 4.3 Determining Location of Bilabials in Test Speech Sample

For a test speech sample, a window of 30msecs was taken inside the speech waveform. MFCCs were calculated for the speech within this window, and the probabilities of the MFCC point belonging to either of the two GMMs trained were calculated. The two probabilities were subtracted, and the resulting value was scaled between 0 and 1. The new value is considered to be the probability that the speech in the 30msec window is a bilabial. Thus, such probabilities are calculated throughout the test speech waveform, and a graph of these probabilities is plotted. The complete procedure is:

- Start at time t=0.
- Take a 30msec window of speech, and calculate its MFCCs.
- Calculate the probability of the 30msec speech sample belonging to each of the two GMMs.
- Subtract the probabilities of not being a bilabial to that of being a bilabial, i.e. (<probability from GMM1> - <probability from GMM2>), and store the resulting value.
- Shift the window forward by 10msecs in the speech and repeat the process until end of speech.
- Among the resulting values, consider only the values greater than 0, and scale them to lie between 0 and 1.
- Draw a graph of the resulting values and note the time instances of the peaks of the resulting graph whose values are above a pre-set threshold.

It is decided that the peaks of this graph that cross a certain threshold of probability correspond to a bilabial. Thus, using this graph, we are checking for those 30msec windows in the audio signal which have the maximum probability of being a bilabial.

### 4.3.1 Calculation of probability of data point *X* belonging to a GMM *J*

Given the GMM that has been modelled using the training data, i.e. given the weights, centres and variances of the different Gaussian distributions in the mixture, it is desired that the probability of a new window of speech to belong to the GMM, i.e. $\boldsymbol{P(X|J)}$, be calculated.

The new data point $\boldsymbol{X}$ is the 13-dimensional MFCC of the new 20msec window of speech. The probability of this point belonging to the GMM is:

$$\boldsymbol{P(X|J)} = \sum_{i=1}^{n} \mathbf{w}_i \frac{e^{\frac{\left|\mathbf{X}-\bar{\mathbf{X}}_i\right|}{2\boldsymbol{\sigma}_i}}}{\left(2\pi\boldsymbol{\sigma}_i\right)^{n/2}}$$, where n=16 centres in the GMM.

### 4.3.2 Advantages

- Successfully uses MFCCs to compute a probability with good success rate.
- Is very flexible; parameters of the GMM can be easily changed to better fit the model.

### 4.3.3 Disadvantages

- This method can only provide us with a probabilistic determination of bilabials, rather than a classification.
- There can be many correct sets of parameters for the GMM to fit the model.
- Requires a lot of training data.

The program for this task was written in MATLAB.

# 5. DETECTION OF LIP CLOSURE IN VIDEO

The task of detecting bilabials in the video has been achieved using image processing tools. The main clue to detecting a bilabial is identifying that when a bilabial occurs, the lips of the speaker close. Thus, the problem is reduced to that of detecting when the lips of the speaker are closed.

## 5.1 The Procedure

The algorithm devised to detect where the lips of the speaker are closed is:

- The face is detected. This was achieved using the Viola-Jones Algorithm or Haar Classifier, a standard procedure used in various applications.
- From the face region, the mouth region is detected, also using Haar Classifiers.
- Within the mouth region, the lips are detected using colour filtering. A suitable colour range was selected to detect lips, and a binary image of the lip region was formed by selecting only those pixels within the colour range.
- A set of conditions were devised to detect whether the lips are closed or not:
    - On the lips, along a 3-pixel wide vertical column of pixels, the values of the pixels are checked.
    - The change in values of the pixels can follow a particular order if the lips are closed and another order if they are open. If the first order of change in pixels is detected, then it is said that the lips are closed.

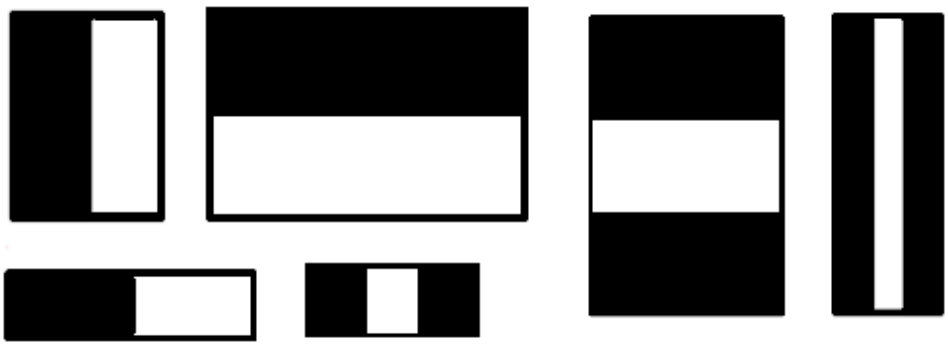
Thus, all the points in the video where the lips are closed are detected.

One of the problems faced was that once a lip closure has been detected that at some point in time, it had not been verified whether the closure of the lips is due to a bilabial being uttered, or due to the speaker not speaking anything. In order to solve this problem, a check was made to distinguish between the two cases by calculating the signal energy in the speech audio signal at all points. Based on the values of the signal energy, the periods of time where no speech is being uttered could be identified, and the lip closures occurring within those time durations were eliminated.

## 5.2 Haar Features

Haar features, in simple terms, are a number of rectangular designs of white and black. The presence of a Haar feature in an image is determined by calculating the difference between the average pixel values in the given image in the white region and in the black region of the Haar feature being considered.

There are various kinds of Haar features depending on the kind of design they make. For example, Edge features are those that contain only one of white and black rectangles. Line features are those that contain three rectangles, either a white sandwiched between two black rectangles, or vice versa. Some examples of Haar features are:



## 5.3 Viola-Jones Algorithm to detect face and mouth regions

The Viola-Jones algorithm uses Haar-like features to determine whether a given image is a face or not. The steps involved are:

1) **Detect Haar features:**
   This is done by iterating through the entire image with all Haar features. As seen, every Haar feature contains a white region and a black region. In the given image, the value of the difference of the average pixel value in the regions defined by the white region and the black region in the Haar feature is calculated. For every Haar feature, if this number is above a certain threshold, then it is said that that particular Haar feature is present.

2) **Check if Haar feature is present:**
   Going through every Haar feature one after another, at every Haar feature check if the value calculated above is greater than a certain threshold value. If it is, then it is said that the Haar feature is present. The threshold values for a face, or a mouth, have been recorded previously and stored in the OpenCV code for finding Haar features. Only if all the Haar features are present is a face, or a mouth, said to be detected.

3) **Use the AdaBoost Algorithm:**
   During the above step, it can so happen that some of the Haar features are not so significant to the detection of a face. These features are called "Weak Classifiers." But since such a stringent method is applied as described above, unnecessary amount of importance is given to these weak classifiers. In order to make the algorithm faster and reduce the importance of weak classifiers, a number of weak classifiers are combined into one "Strong Classifier." This combination is done by the AdaBoost Algorithm.

## 5.4 Lip Detection using Colour Filtering

Lip Detection is carried out using Colour Filtering, i.e. by finding those pixels that carry the colours of the lips, as compared to those of the skin or the inside of a mouth. These were experimentally found to lie in:-

Hue (on a scale of 0 to 180): 0 – 12 & 142 – 180,
Saturation (on a scale of 0 to 255): 25 – 100,
Intensity (on a scale of 0 to 255): 25 – 190

## 5.5 Advantages

- Checking for lip-closure in video is more accurate than checking for bilabials in audio.
- This method can work for videos with low quality, as are very often found in online repositories such as YouTube.
- No training required.
- Can be easily implemented in OpenCV rather than MATLAB for faster results.

## 5.6 Disadvantages

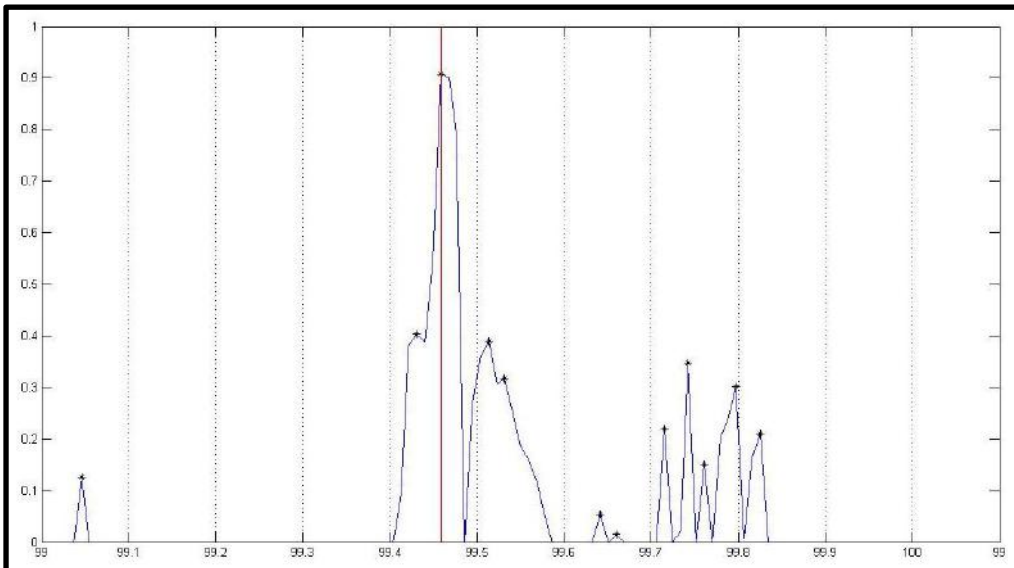
- Frame rate of the video affects the accuracy of the time at which lip closure is detected.
- Colour range chosen to detect lips may need to be dynamically varied depending on the gender and ethnicity of the speaker in the test case.
- Cannot detect face and lip regions if head is tilted beyond a certain angle (only in rare cases).

The program for this task was written in OpenCV, and the results (the time points where bilabials occurred) were exported onto a text file, to be imported by MATLAB for further use.

# 6. RESULTS

## 6.1 Detection of Bilabials in Audio

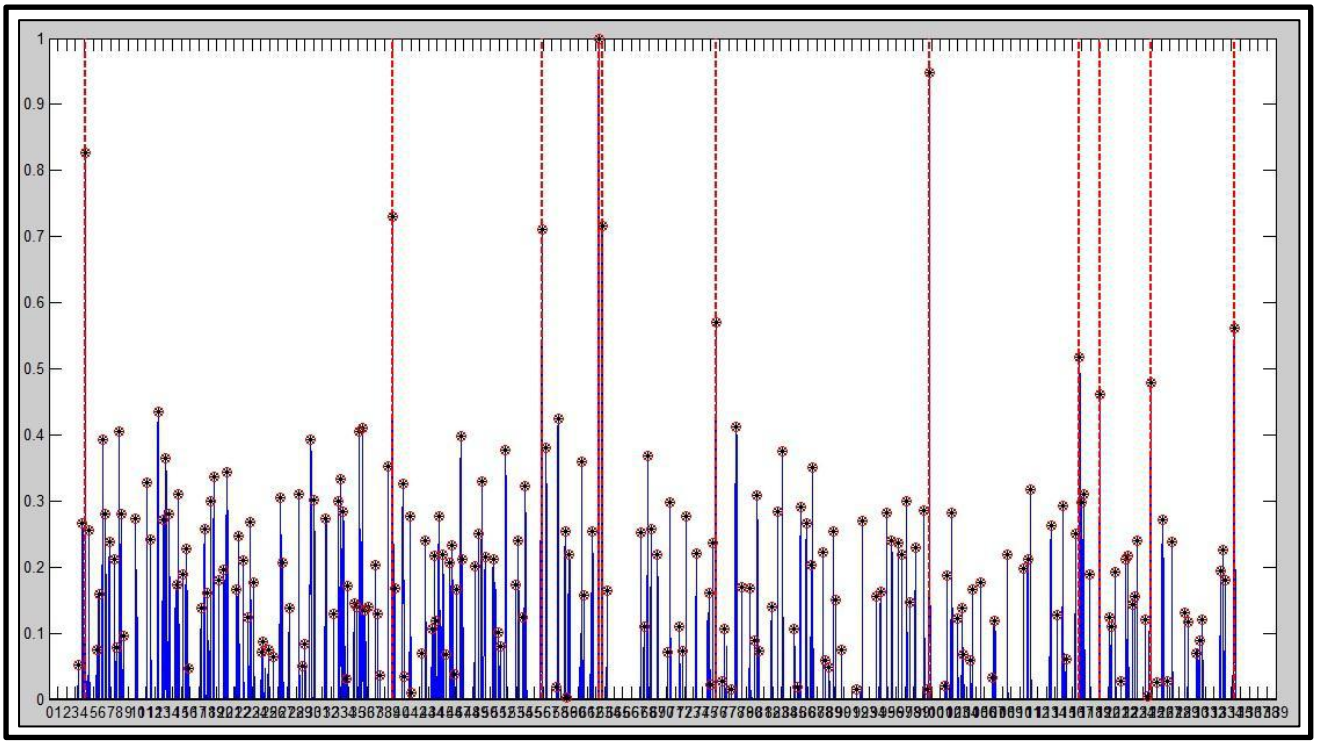
An example of such a bilabial detection in audio is given below:



The peaks in the graph have been identified with a '+' symbol. A bilabial exists at the point where the red vertical line is drawn. Taking a probability threshold of .5, we can see that the graph clearly and accurately identifies the bilabial in this example.

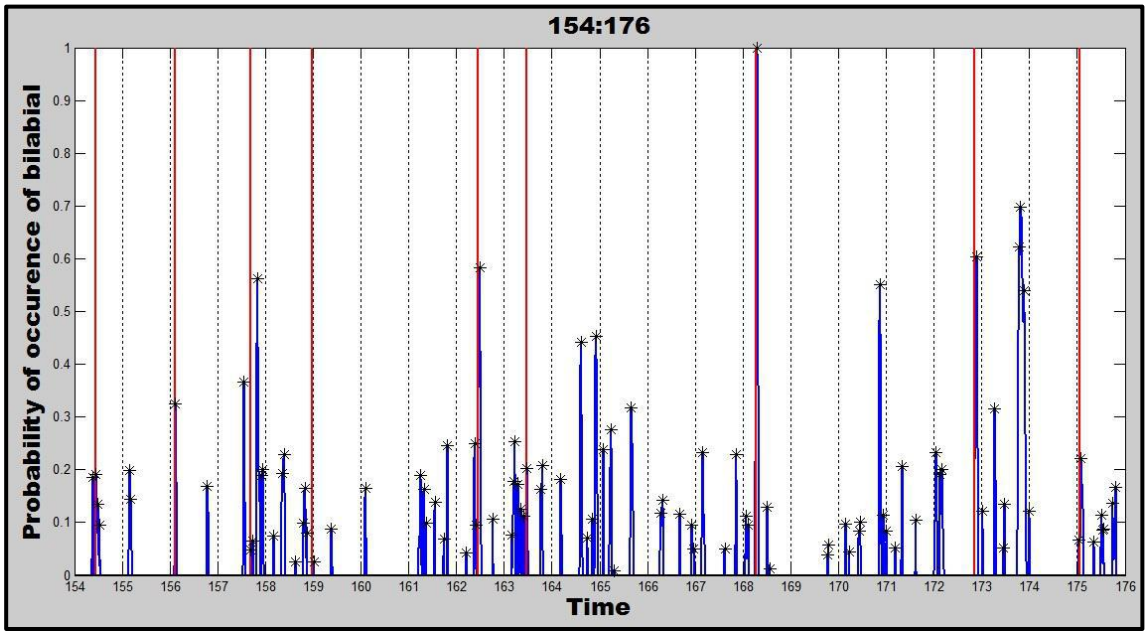
### 6.1.1 Testing on sample belonging to training data

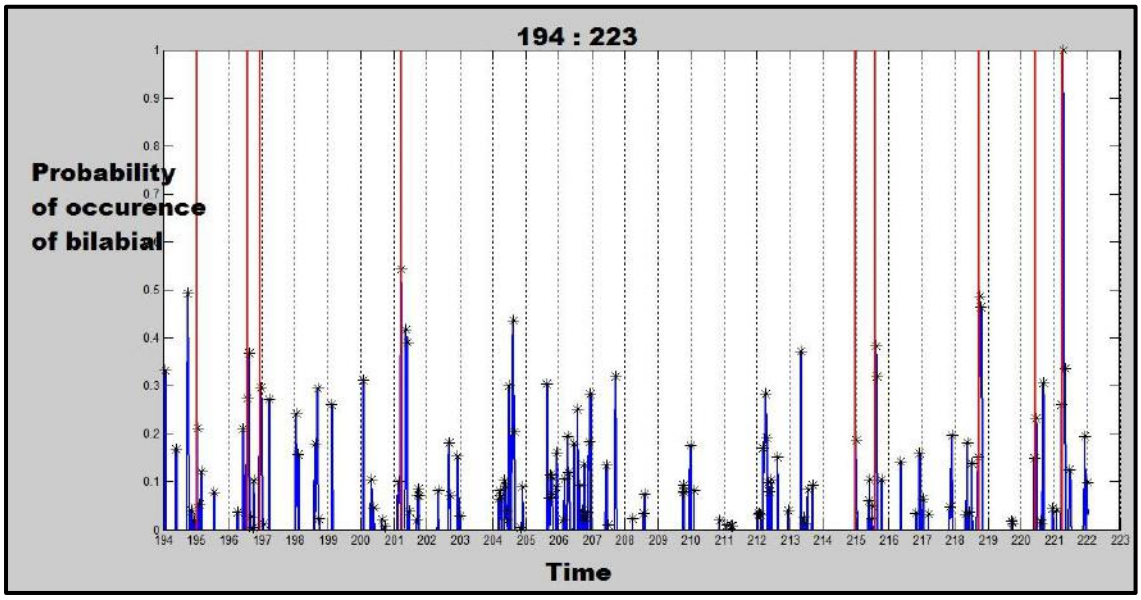
In the following example, the dotted lines are the points where bilabials occur. It can be seen that these points can be clearly distinguished from the rest of the peaks in the graph using a suitable threshold.

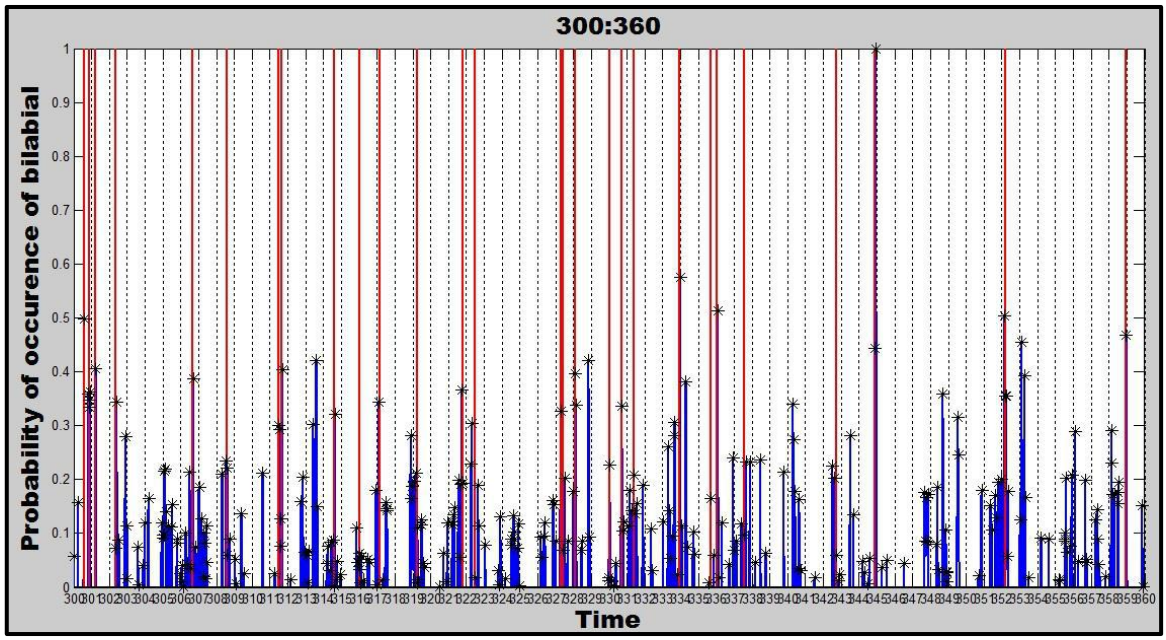


These results were obtained by testing the algorithms designed on the training set.

### 6.1.2 Testing on new sample data



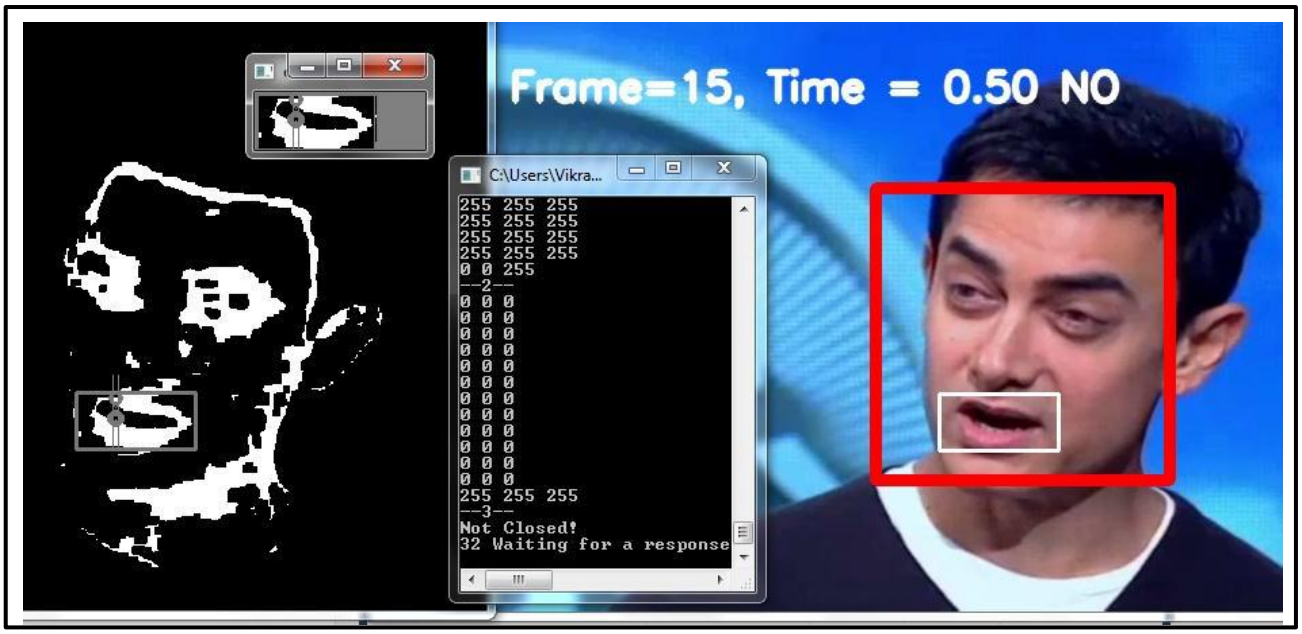



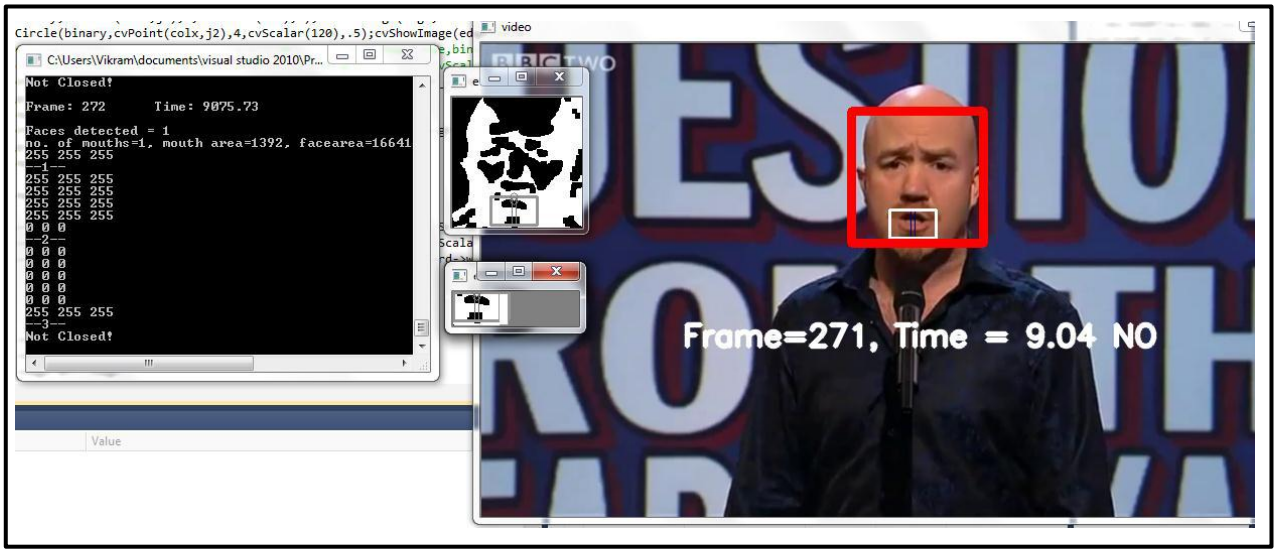
## 6.2 Detection of Lip CLosure in Video

### 6.2.1 Lips not closed

Below are examples of cases where the lips are not closed. As can be seen, in the lip region, along the vertical column shown by the grey lines, three regions of alternate white and black can be seen. This is characteristic of the lip region when the lips are open.

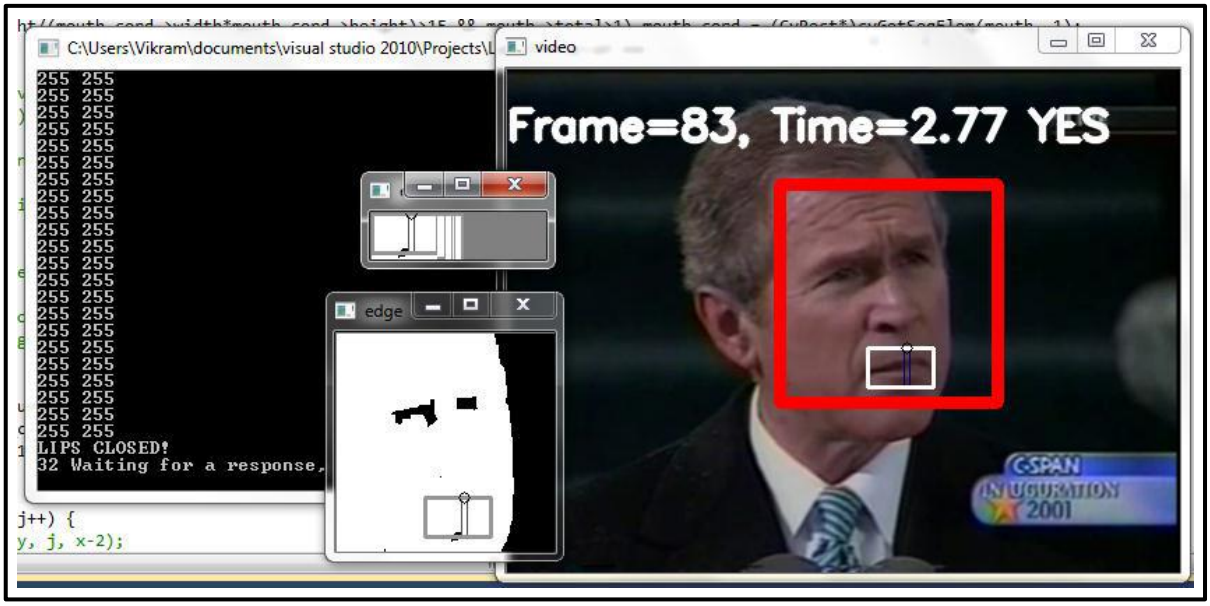


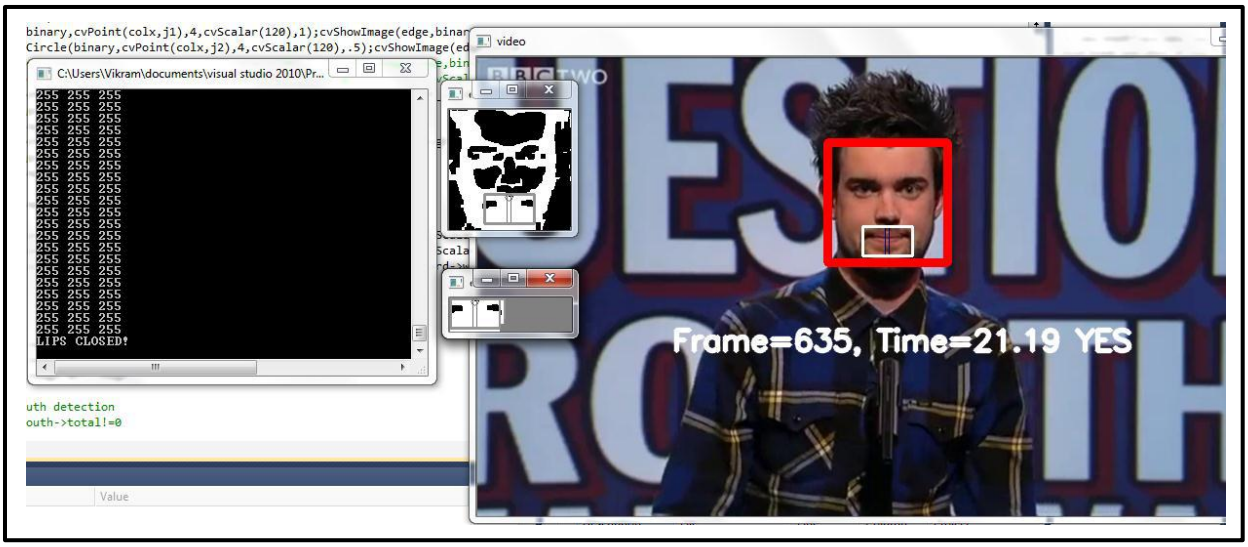




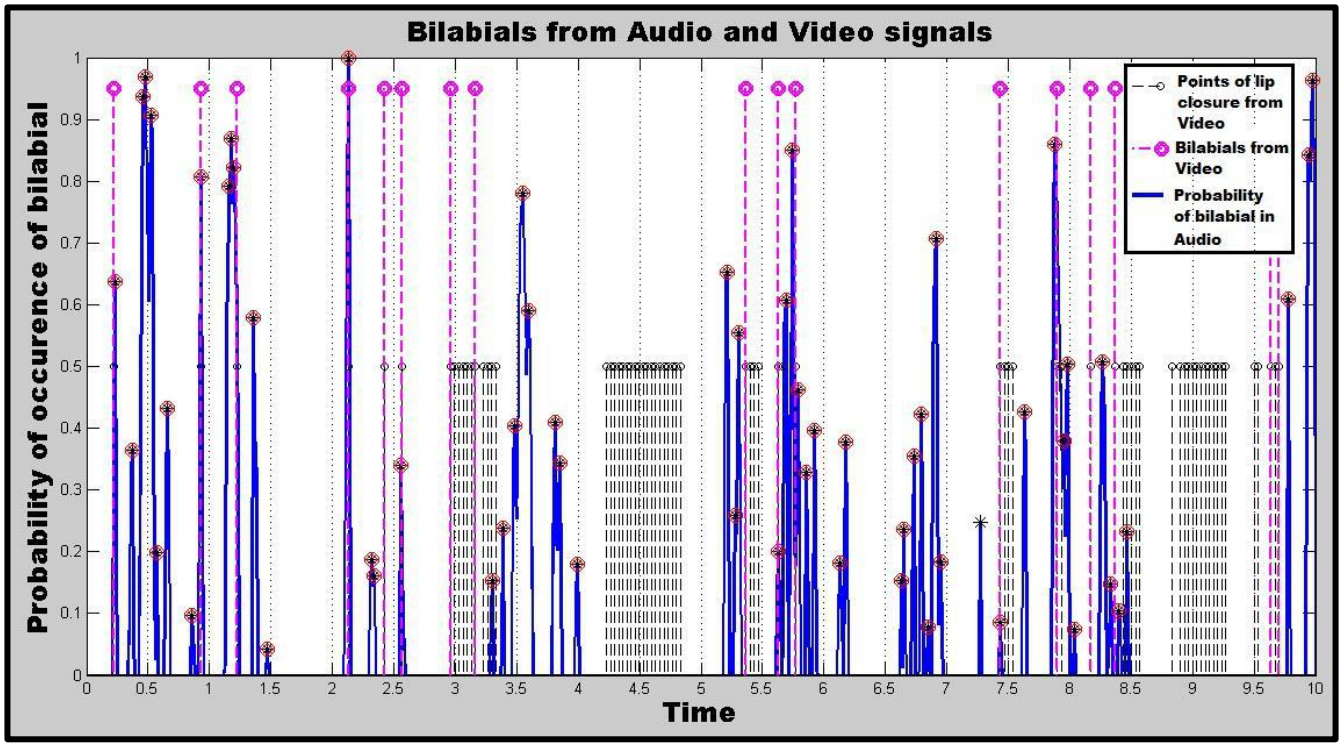
### 6.2.2 Lips closed

In the figure below, it can be seen that lip closure has been detected using the algorithm described above. As can be seen, the lip region can be split into two regions, one of white and the other of black. This is characteristic of the lip region with the lips closed.

# 6.3 Combining Detection of Bilabials in Audio and Video




## 6.3.1 Eliminating Lip Closure Points with No Voice using Signal Energy

In the figure above, the black stem lines represent those points in time that have been identified where the lips of the speaker close. By calculating the signal energy in the audio, those time durations that have no speech in them have been eliminated, leaving only those points in time which represent bilabials. Thus, the stem lines in magenta represent these bilabials identified in the video.

As an example of the elimination described in the previous paragraph, consider the time duration between 4 and 5 seconds. Many black stem lines can be seen in that duration, which indicates that the lips are closed within that time. However, there are no magenta lines in that duration. This tells us that in that duration, the lips of the speaker were closed because he was not speaking anything. The fact that there was no speech in that duration was derived from the signal energy calculated for the audio signal.

## 6.3.2 Combining Audio and Video

As seen before, the plot in blue represents the probability of occurrence of bilabials in the audio. The peaks of this plot have been decided upon to represent the points where bilabials occur. These peaks have been highlighted in the plot with red circles.

As can be seen, there are some bilabials that have been identified in the video, and some that have been identified in the audio. For the purpose of the project, even though some bilabials have been missed from the video, and some from the audio, it is necessary and sufficient to record those points in time that have been identified as bilabials in both video and audio.

# 7. VOWEL DETECTION

Vowel Detection has been implemented to try to decrease the number of false positives detected. It is known that a bilabial consonant always follows a vowel. Thus, if vowel sounds are detected in the given speech, the points of vowel sounds and points of bilabial consonants can be considered together to determine more accurately where a bilabial consonant is occurring.

The algorithm devised to detect vowels in the audio signal makes use of the simple concept of a Formant in speech analysis. A Formant is a peak in the power spectrum of a sound signal. Every vowel that is emitted by the human voice has its spectral peaks at certain frequencies that are different for different vowels. Thus, if we find the formants for a window of speech, and compare them with the formants that belong to a vowel, a vowel can be detected. The algorithm is:

- Iterate through the speech signal, considering speech samples of 30msecs.
- The speech sample is zero-centred, by subtracting the speech values by their mean, and normalized, by dividing all values with the absolute maximum sample value.
- The Power Spectrum of the normalized zero-centred speech sample is determined using the Yule-Walker method.
- The frequencies where the first three peaks occur in the Power Spectrum are noted as the Formants of the speech signal.
- The three frequencies, combined with their respective weights, are compared to pre-determined formant frequency values for the different vowel sounds.
- If the error between the formant frequencies of the speech sample and any one of the vowel sounds is less than a threshold value, then it is claimed that the speech sample in consideration contains a vowel.

The first three formant frequency values of vowel sounds have been experimentally found to be:

| Vowel Symbol | Example Word | F1 (Hz) | F2 (Hz) | F3 (Hz) |
|---|---|---|---|---|
| /ow/ | Caught | 570 | 840 | 2410 |
| /oo/ | Loot | 300 | 870 | 2240 |
| /u/ | Put | 440 | 1020 | 2240 |
| /a/ | Hot | 730 | 1090 | 2440 |
| /uh/ | Cut | 520 | 1190 | 2390 |
| /e(r)/ | Bird | 490 | 1350 | 1690 |
| /ae/ | Bat | 660 | 1720 | 2410 |
| /e/ | Bet | 530 | 1840 | 2480 |
| /i/ | Bit | 390 | 1990 | 2550 |
| /iy/ | Beat | 270 | 2290 | 3010 |

## Yule-Walker Method:

The Power Spectrum of the input signal vector is estimated using the Yule-Walker Auto-Regressive method. This method, also called the autocorrelation or windowed method, fits an autoregressive (AR) linear prediction filter model to the input signal by minimizing the forward prediction error (based on all observations of the input sequence) in the least squares sense. This formulation leads to the Yule-Walker equations, which are solved by the Levinson-Durbin recursion. The AR model thus obtained is used to determine the Power Spectrum of the voice signal by calculating the squared magnitude of the frequency response, i.e. the Discrete Time Fourier Transform of the auto-covariance of this AR model.

Vowel Detection was carried out over various speech samples, but unfortunately its advantages turned out to be negligible. Hence, this has been made redundant in the current code.

# 8. ERROR IN LIP CLOSURE DETECTION IN VIDEO

## 8.1 Error due to Viola – Jones Algorithm used to detect face

1) Detect Haar-like features
2) Check if each feature is above a threshold
3) Use the AdaBoost Algorithm to combine weak classifiers and attach weights to them.

The Haar Features used only work within limits of horizontal, vertical and angular face tilts. Also, the thresholds recorded to detect the feature are only experimentally theorized values, and may not be 100% accurate. Thus, it is possible that faces or mouths in certain frames are not detected, or are erroneously detected.

For example:

In this image, it can be seen that due to excessive head tilt, the face could not be detected.

In this image, it can be seen the mouth has been erroneously detected because of limitations in the Haar Classifier, which led to the false conclusion that the lips are closed.


## 8.2 Error in Colour Detection

The original image, extracted as a frame of the video, is in the format BGR in OpenCV. It is then converted to the HSV – Hue, Saturation, Intensity – format so as to simplify filtering on the basis of colour. The essential part of lip detection is Colour Filtering, whereby a binary image of the lips is formed by examining the pixels of the mouth and classifying them on the basis of their HSV values. Thus, what is essentially being implemented is a Classification operation on the basis of colour, into "lip" and "non-lip" regions. The HSV values for different lips are different, but they generally fall under the bracket of:

Hue (on a scale of 0 to 180): 0 – 12 & 142 – 180

Saturation (on a scale of 0 to 255): 25 – 100

Intensity (on a scale of 0 to 255): 25 – 190

This process of colour filtering could be erroneous when the lips of the speaker fall beyond the above brackets, or when the skin (outside the lips) or inside-of-mouth region have HSV values in the above bracket.

This can occur due to two reasons:

1) Lighting of the room renders the colours of the skin and lips to change.
2) The resolution of the video is not very high. This leads to higher pixilation of the video, i.e. larger pixel size.

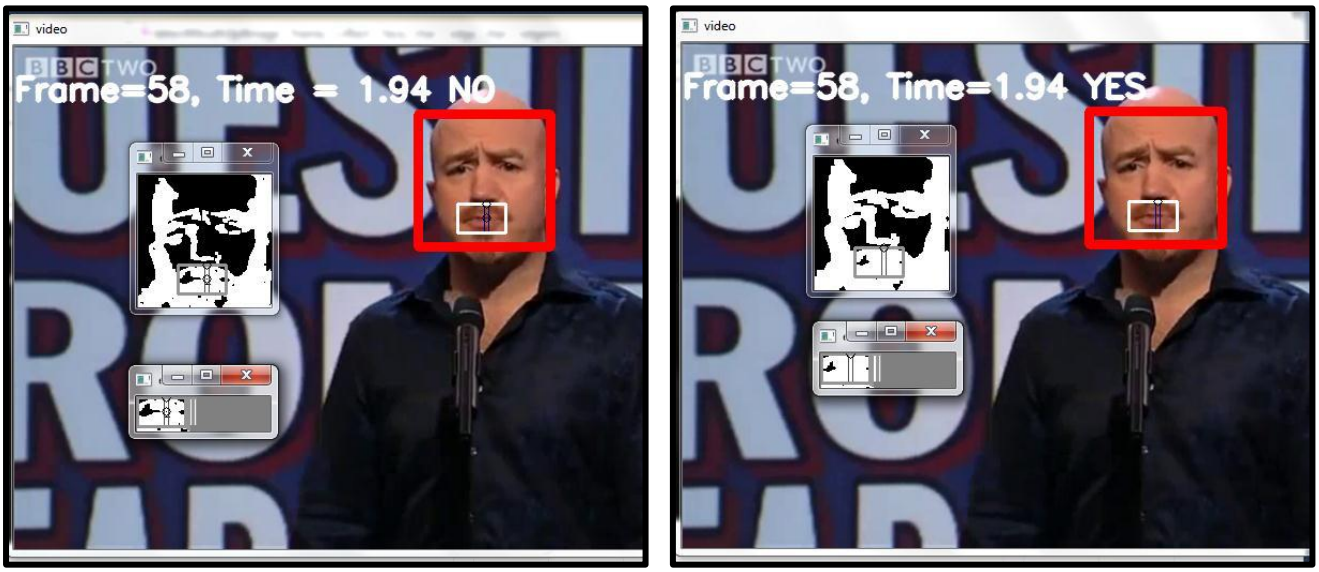
Reason 1) can be partially taken care of by mean-centring the image of every frame, i.e. by subtracting the mean of all pixels BGR or HSV values from every pixel. This shall be able to cancel out the effect of varied illumination.

Reason 2) is harder to take care of. When the pixel size is small, a better gradation of colours through the lips and mouth occurs. This means that if the inside-of-mouth region is black, it would show up as black for the number of pixel rows that represent it. However, if the video quality is deteriorated, then pixel size would be greater, leading to the BGR or HSV value of the bigger pixels to be the mean of the original smaller pixels which the bigger pixel replaces (similar to mean filtering). This can lead to the HSV value of the pixel to lie within the above bracket, thereby tampering with the classification.

Correcting Options:

1) Depending on the skin tone and lip redness of the speaker, the Hue, Saturation and Intensity value brackets can be suitably changed manually. This shall lead to better classification of lip and non-lip regions. This has been implemented in the code.
2) In addition to colour filtering, an additional contiguous operation of Erosion and Dilation of the binary image can carried out. These shall lead to removal of granular noise in the image. By changing the number of times Erosion and Dilation are each carried out, different results can be obtained on the binary image, which can lead to better classification of lip and non-lip regions. This has been implemented in the code.

In the following example, such correcting options as explained above have been employed. Observe how the lip closure was not detected before, while after the corrections were made, lip closure has been detected in the same frame:



## 8.3 Error in Pixel Column

This is a direct consequence of the error in colour filtering. The 3-pixel-wide column that is to be examined could be positioned anywhere across the lips. Due to errors in colour filtering, different positions of the pixel column could lead to different results. Generally the column is positioned at half the width of the mouth from the left edge of the mouth. Different positions that have been tried are at one-third mouth width, two-fifth mouth width, three-fifth mouth width and two-third mouth width.

This is a relatively rare error, and need not be bothered with frequently.

## 8.4 Error due to Frame Rate

The frame rate of a typical video is 29fps. This amounts to a gap of almost 35msecs between frames. This gap between frames could lead a lip closure's exact time to not be captured with a margin of error of 35msecs, or worse, missed entirely if spoken too soon. There is no way to rectify this since this is an error that is set during the recording of the video itself.

# 9. ERROR IN BILABIAL DETECTION IN AUDIO

## 9.1 Error due to MFCC

The Mel Frequency Cepstrum Coefficients have been chosen as the features to train the Gaussian Mixture Models upon because:

1) these coefficients are relatively gender-independent in modelling vocal sounds,
2) these coefficients are very good at modelling human ear auditory response, as the Mel frequency scale is a good approximation to the frequency response that the human ear provides,

Though the MFCC features are very good approximations of auditory response of the human ear, they are nonetheless only approximations. Thus, some error does arise from the conversion from linear scale to Mel scale of the Discrete Fourier Transform of the input signal.

## 9.2 Error in Training Data

The training data fed to the GMM is the manually segmented windows of speech. Speech of 300 seconds each of 8 speakers has been manually scanned through to look for bilabial sounds. The start time and end time of each of those bilabial consonants has been manually noted down. Thus, this method is prone to human error in noting down the bilabial sounds. This can lead to erroneous modelling of the GMM due to:

1) inclusion of non-bilabial points into the GMM,
2) exclusion of bilabial points from the GMM.

## 9.3 Error in the Gaussian Mixture Model

A major portion of the error in the procedure arises from the training of the GMMs. The training data fed into the GMMs is the different 13-dimensional MFCC vectors that arise from the various 20msec windows of speech that was manually labelled as bilabial.

Consider all of these 13-dimensional vectors as points on a 13-dimensional plane. The GMM is a weighted sum of (in this case) 16 number of 13-dimensional Gaussian distributions. The training of this GMM occurs by trying to fit the parameters of every Gaussian such that the probability of each point to belong to the GMM is maximized. It has been mentioned that this is done by minimizing the error term, which is the negative log-likelihood of the training points to belong to all the 16 Gaussian distributions in the GMM.

The points where errors can occur in the GMM are:

### 9.3.1 Iterations of the Expectation Maximization Algorithm

The EM Algorithm iterates itself repeatedly and tries to find the best fit possible for the GMM by adjusting the weights, centres and variances of the different Gaussian distributions. The maximum number of iterations allowed in the current program is 50. It is possible that the number of iterations could be increased to get a better fitting GMM. But the more the number of iterations, the more the time consumed by the program.

### 9.3.2 Over-fitting & Regularization

Over-fitting is a common problem in many Machine Learning concepts. It refers to the excess of training data that can lead to an extravagant fitting of the GMM. An excess of training data can occur due to many data points that are very close to each other, thereby occurring in a relatively small area. Thus, excess of training data is not a good idea as the GMM would over-work itself to try to conform to the more minute variations in the training data points in a relatively small area.

Regularization refers to the concept of trying to identify such close-by points and try to depict them with a single point so as not to over-stress the EM algorithm. Currently, no provision for regularization has been provided in the code.

It is fair to mention that care has been taken to try to avoid over-fitting. Some of the bilabial points of the same sound from the same speaker have not been included in the training data, so as to avoid repetition. Also, it has been experimentally found that the sound 'ma' occurs more frequently that the other bilabial consonants 'pa' and 'ba.' This could lead to an over-fitting of the GMM with respect to the 'ma' sound. Efforts have been made to include an equal number of training points from all three sounds 'ma,' 'pa,' and 'ba,' in order to regularize the training.

### 9.3.3 Number of Gaussian distributions in the Mixture

The number of Gaussian distributions used in the GMM affects the accuracy of the model. More the number of distributions, more is the flexibility of the model to be able to conform to the graph of the training data. However, an excess of distributions could lead to over-fitting. Thus, the right number of distributions has to be selected. It has been found during literature survey that in general a power of 2 number of distributions is preferred. Experiments have been conducted with 8, 16 and 32 distributions, and 16 distributions were selected. However, 32 distributions could also be used, keeping in mind that more the number of distributions, more the time consumed by the EM algorithm to fit them in the training data.

# 10. REFERENCES

[1] D. Reynolds, and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models". *IEEE Transactions on Speech and Audio Processing 1995*, vol. 3, pp.72–83, January 1995.

[2] A. Albiol, L. Torres, E. J. Delp, "Optimum color spaces for skin detection". *In Proceeding of the International Conference on Image Processing 2001,* vol. 1, pp.122-124, October 2001.

[3] M. Marolt, "Gaussian mixture models for extraction of melodic lines from audio recordings". *In Proceeding of the 2004 International Conference on Music Information Retrieval,* October 2004.

[4] V. Robert, B. Wrobel-Dautcourt, Y. Laprie and A. Bonneau, "Inter speaker variability of labial co-articulation". *Auditory-Visual Speech Processing Workshop 2005*, July 2005.

[5] S. Stillittano, V. Girondel, A. Caplier, "Inner and outer lip contour tracking". *VISAPP 2008 - International Conference on Computer Vision Theory and Applications*, January 2008.

[6] A. Philar, A. Akerkar, "Vowel recognition through formant analysis". *IEEE Transactions on Speech and Audio Processing 2010*, vol. 1, pp.17–35, May 2010.

[7] V. Tiwari, "MFCC and its applications in speaker recognition", *International Journal on Emerging Technologies,* vol. 1, pp. 19-22, February 2010

[8] S. Davis and P. Merlmestein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". *IEEE Transactions on Audio-Specific Signal Processing 1980,* pp. 357-366, August 1980.