

# Establishing the Blink Cycle of the Eye using OTSU Method and Gaussian Filter

Dominic Asamoah  
Department of Computer  
Science, KNUST, Ghana

Peter Amoako-  
Yirenkyi  
Department of  
Mathematics, KNUST,  
Ghana

Stephen Opoku  
Oppong  
Faculty of Informatics,  
GTUC, Ghana

Nuku Atta Kordzo  
Abiew  
Faculty of Informatics,  
GTUC, Ghana

## ABSTRACT

Strong and efficient algorithm in real time eye tracking system has been an ultimate and thought-provoking problem for computer vision. This so because most studies have tried to characterize eye using mainly pupil and iris. These features need the full cooperation of the individual making computing information impractical. Secondly, computing information using these features is subjective and also depends on the race. All these methods do not consider the individual making it general as the individual has blink cycle and for that matter different levels of fatigue rendering previous works inaccurate, hence this study. In this paper, a methodology for establishing the blink cycle of the eye is presented. The paper employs a method, where individual's face is captured by a camera by receiving video sequence which are streamed into frames and then transformed into RGB. Haar classifiers are used to detect eyes region and eyelid feature. The eyes are detected to be either open or closed at a particular period by using thresholding and equations regarding the symmetry of human face. The eye region is processed to ascertain certain attributes of eyelid movement.

## General Terms

Image Processing, Algorithms, Computer Vision

## Keywords

Blink Cycle, Haar Classifiers, Eyelid movement, Gaussian Filters, Otsu Method

## 1. INTRODUCTION

Eye tracking as a tool is now more reachable than ever, and is growing in popularity amongst researchers from a whole host of different disciplines [1]. The technique is used in cognitive science, psychology, human-computer interaction, advertising, medical research, and other areas. Human eye-gaze, blinking and eye movement can be recorded with relatively high dependability by modest techniques. Though, there are relatively few techniques proposed for the active scene where the head and the camera move independently and the eye moves freely in all directions independently of the face. Though, care must be taken, that eye-gaze tracking data is used in a sensible way, since the nature of human eye movements is a combination of several voluntary and involuntary cognitive processes. Normally, eye tracking is performed on two dimensions to measure the horizontal and vertical motions of the eye.

## 2. LITERATURE REVIEW

Lalonde et al [2] used Optical flow in tracking eyelid movements to detect eye blinks. Another method based on optical flow estimation was also presented by Divjak and Bischof [3]. In this method, eyes and face position were located by 3 different classifiers. This algorithm is successful only when the head is directly facing the camera.

Liting et al [4] use a deformable model - Active Shape Model represented by several landmarks as the eye contour shape. Model learns the appearance around each landmark and fits it in the actual frame to obtain new eye shape. Blinks are detected by the distance measurement between upper and lower eyelid. Ayudhaya et al [5] detect blinks by the Eyelid's State Detecting (ESD) value calculation. It increases the threshold until the resulting image has at least one black pixel after applying median blur filtering. This threshold value (ESD) differs while user blinks.

Khilari [6] proposed an iris based method which based on a projection function to detect the user's eye blinks, analyse their patterns and duration and interpreting them. Mardiyanto and Arai [7] proposed another blinking detection method which was based on distance between the two arcs of the eye (the upper and the lower part). The arcs were detected by applying Gabor filter onto the image.

Arai et al [8] present Gabor filter-based method for blink detection. Gabor filter is a linear filter for extracting contours within the eye. After applying the filter, the distance between detected top and bottom arc in eye region is measured. Different distance indicates closed or opened eye. The problem of arc extraction arises while the person is looking down.

Pimplaskar et al [1] presented a method for detection of eye blinking robustly in real time environment by the use of connected component technique and centroid method to track and blinking of eyes on OpenCV platform.

Galab et.al [9] proposed a system for detecting eye blinks without any restriction on the background and the user does not have to wear any sensor or marks. This system automatically classifies the eye as either open or closed.

Salehian and Far [10] proposed a blink detection system algorithm that serves as a visual cue that may be correlated to the state of fatigue using a complimentary but independent approach, shape analysis and histogram to detect blinking.

Akshatha [11] presented a tracking algorithm which was designed to find the position of the interested regions in image sequence which can ultimately be used to search pose and actions of the eye. This system automatically learns specified eye appearances which look strongly related to the blink detection.

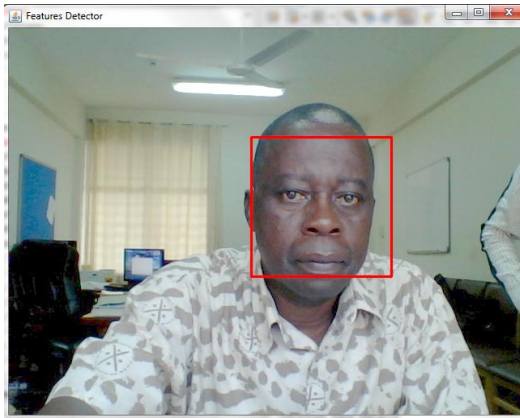
## 3. METHODOLOGY

### 3.1 Video Acquisition

The initial stage of the proposed system is the video acquisition stage. In the video acquisition stage, the video sequence was captured using a Dell Inspiron webcam. The sample videos were collected under different intensities and illuminations over a specified time interval. The input videos

were then enhanced using image processing techniques by filtering or removing the noise. The acquired video stream is then split into individual frames for use in face detection. It is suitable to initially identify the face in the image before applying digitizing the image because it reduces the area of interest as compared to the entire image and thus improves the response time.

The Haar Cascade classifier is used to identify the face. Once the face region is identified it is marked with a rectangle as shown in Fig. 1.

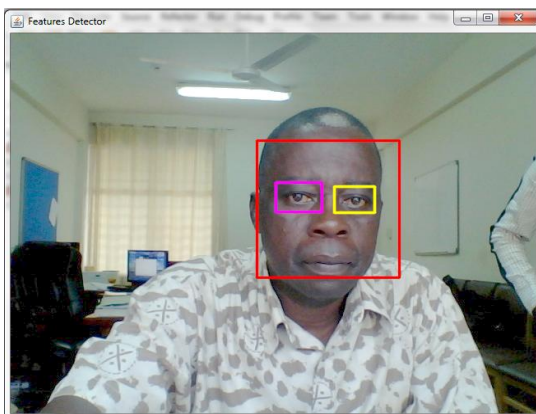


**Fig. 1: Detected face**

Finding the face narrows down the region of the eye portions and thereby makes it simpler to computationally focus on the position of the eye.

### 3.2 Eye Region Detection

The Haar classifier cascades is first trained to be able to locate the human facial features, such as the mouth, eyes, and nose, as shown in fig. 2. The Ada Boost algorithm and the Haar feature algorithms are employed in order to train the classifiers. The classifier is implemented using the OpenCV library.



**Fig. 1: Detecting Facial features**

After identifying the face region, the eyes are detected using the Haar cascade classifiers. The object detector includes the detected eye in the list of facial features and draws a rectangle over the eye region as shown in Fig. 2.

Blinking (eye lid movement) is observed constantly and continuously by the present state of the eye as to whether it is open or closed in each frame. The eye is not detected in the closed state; it is identified only in the open state when using the Haar classifier.

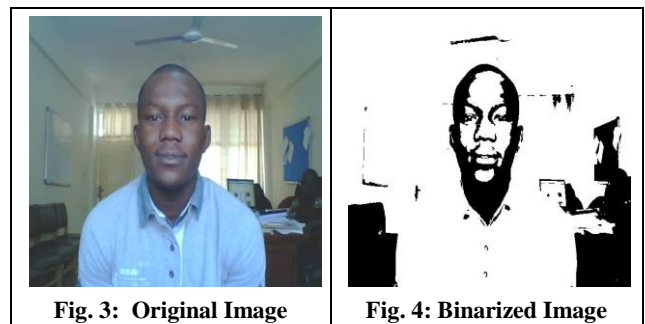
After successful detection of the eye, the eye sub image is created and stored in a buffer for further processing.

### 3.3 Binarization process

Before a digital image is obtained, an image being made up of continuum intensity values is quantized. At this stage of the process, the extracted eye images are binarized (digitized). A digital image which has only two possible values for each pixel is termed to be a binary image.

In binary form, it is easier to differentiate the objects from the background. The greyscale image is converted to a binary image through a process called Thresholding. The converted binary image has a value 0 for black which represents luminance less than a given threshold level for all pixels and 1 for white in the original image for any other pixels. Thresholds are often given depending on certain conditions such as surrounding lighting conditions and the complexion of the human face. It was found out that it is not ideal to choose a fixed threshold value after looking at a lot of images of different faces under different lighting conditions. Hence the Otsu method of calculating the threshold value based on the selected image is adapted for use in this study. The condition used in determining the accurate threshold was centered on the notion that, in the binary image, the face should have a majority of white to allow a few black blobs for the eyes, nose and lips.

The first step in localizing the eyes is to binarize or digitize the picture. The process of converting an analog image to a binary one is called Binarization. The table below shows examples of images and their respective binarized counterparts.



### 3.4 Thresholding

Detecting the sub images that denotes objects is one of the most significant difficulties in vision system. Partitioning an image into regions is called segmentation and a partition denotes an object or part of an object. Generally, segmentation is said to be a method to partition image,  $M [i, j]$ , into sub images, called regions, that may range from  $P_1$ , through  $P_k$ , such that each sub image is a candidate for accommodating an object.

Segmentation and thresholding are synonymous for the case of binary vision. Thresholding is a process used to convert a grayscale image to a binary image so that objects of interest are segregated from the background.

In a fixed thresholding scheme, the characteristics of the intensity is used to determine threshold value. Assuming that a binary image  $D[i,j]$  is the equal to a threshold gray image  $M_T[i, j]$  which is achieved using a threshold  $T$  for the original gray image  $M[i, j]$ , then

$$D[i, j] = M_T[i, j] \quad (1)$$

For a darker object on a lighter background

$$M_T[i, j] = \begin{cases} 1 & \text{if } M[i, j] \leq T \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

If it is known that the object intensity values are a range  $[T_1, T_2]$ , then we may use

$$M_T[i, j] = \begin{cases} 1 & \text{if } T_1 \leq M[i, j] \leq T_2 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Where intensity levels for an object come from several disjoint intervals, a general threshold scheme may be represented as

$$M_T[i, j] = \begin{cases} 1 & \text{if } M[i, j] \in Z \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where Z is a set of intensity values for object components.

Knowledge about the application domain is incorporated into the thresholding algorithm. The knowledge has been tailored for the domain and therefore the same threshold values may not work in a new domain.

To binarize an image, the image would first have to be converted to grayscale using the following steps as showed in Fig. 5

After the image has been converted to grayscale, a threshold that can be used in the binarization process has to be set. Otsu's method which this is just one of the methods that can be used to binarize an image, provides the required binary threshold. The Otsu method iterates through all the likely threshold values and calculates a measure of spread for the pixel levels on either side of the threshold, that is, the pixels that either fall in the

Step 1.	Declare variables a, r, g, b, p
Step 2.	Read in the source images
Step 3.	Loop through the source image and get the pixel values at each position.
	a = alpha pixel value
	r = red pixel value
	g = green pixel value
	b = blue pixel value
	let p = colorToRGB(a,r,r,r)

Fig. 5: Gray Scaling Algorithm

foreground or the background. Finding the threshold value where the sum of foreground and background spreads is minimum is the main aim of this process.

### 3.5 The Mathematics behind the Otsu Method

Let  $r_1$  and  $r_2$  represent the estimate of class probabilities defined as:

$$r_1(t) = \sum_{i=1}^t S(x) \quad (5)$$

$$r_2(t) = \sum_{i=t+1}^I S(x) \quad (6)$$

Sigmams are the individual class variances defined as:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{S(i)}{r_1(t)} \quad (7)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{S(i)}{r_2(t)} \quad (8)$$

The class means are:

$$\mu_1(t) = \sum_{i=1}^t \frac{iS(i)}{r_1(t)} \quad (9)$$

$$\mu_2(t) = \sum_{i=t+1}^I \frac{iS(i)}{r_2(t)} \quad (10)$$

where S signifies the image histogram. The problem of minimizing within class variances can be written as a maximization problem of the between class variances. This can be given as a difference of total variances and within class variances:

$$\begin{aligned} \sigma_b^2 &= \sigma^2 - \sigma_w^2(t) \\ &= r_1(t)[1 - r_1(t)][\mu_1(t) - \mu_2(t)]^2 \end{aligned} \quad (11)$$

This expression can then be safely maximized and the solution is t that is maximizing  $\sigma_b^2(t)$ .

To get the binary threshold using Otsu's method, the above mathematical model is coded in java. The threshold is then used to get a binarized image. Thus a call to Otsu's methods return a threshold, and the threshold value is used to create a new Buffered Image object exhibiting the same attributes as the original image. Because it is gray scaled, only the first pixel value is checked after going through all the pixels of the image. A binarized image is created by setting the pixel value that exceeds the threshold to 255, otherwise 0.

### 3.6 Histogram Equalization

Histogram Equalization is performed on the binarized image. Locating images in which every intensity value fall within a small range is very common if the images have unevenly distributed gray values. Histogram equalization redistributes the gray values uniformly by stretching their contrast. This makes the selection process of threshold more efficient. Generally, histogram equalization improves an image's quality subjectively and is beneficial for an observer who will view it.

A simple illustration of histogram modification is image scaling: the pixels in the range  $[c, d]$  are expanded to fill the range  $[V_1, V_k]$ . The formula for mapping a pixel value z in the original range into a pixel value z' in the new range is:

$$V' = \frac{V_k - V_1}{d - c} (v - c) + V_1 \quad (12)$$

$$= \frac{v_k - v_1}{d - c} v + \frac{v_1 d - v_k c}{d - c} \quad (13)$$

Using this formula with this scheme, the resulting histogram shows gaps between bins when the histogram is stretched. The histogram is stretched while filling all bins in the output histogram with other methods which are better. the following method may be used if the desired gray value distribution is known a priori. We assume that  $p_i$  is the number of pixels at level  $v_i$  in the original histogram and  $q_i$  is the number of pixels at level  $v_i$  in the desired histogram. We start at the left end of the original histogram and select the value  $k_1$  such that

$$\sum_{i=1}^{k_1-1} S_i \leq r_1 < \sum_{i=1}^{k_1} S_i \quad (14)$$

The maximum or highest intensity values of each binarized eye image is computed and stored in a buffer.

Appendix A shows sample images extracted from video and Appendix B shows the highest intensity values of the segmented eye images

### 3.7 Gaussian Filters

Gaussian filters are then applied to produce smoothing of the data to determine whether the eye is close or open.

Gaussian filters belong to a family of linear smoothing with filters weights selected in relation to the shape of the Gaussian function. By applying the Gaussian smoothing filter, it provides a convenient way of removing noise inherent in a normal distribution. It is a mere coincidence that the filter weights are selected from a Gaussian distribution and also the noise is also distributed as a Gaussian function.

In one dimension, the Gaussian function with mean zero is represented as:

$$k(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (15)$$

where  $\sigma$  is the Gaussian spread parameter. This defines the width of the Gaussian function [1].

The discrete Gaussian function with mean zero in two dimensions is used in image processing as a smoothing filter

$$k[i, j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (16)$$

## 4. ANALYSIS

### 4.1 Video Acquisition

Seventy video clips were acquired and processed. These video clips were recorded using thirty-five individuals and were used to study facial movements. A sample image from each video clip is presented in Appendix A. In the video acquisition experiments, frames were extracted from each video sequence. These frames constitute the basis of the analysis. The extracted images were analysed by identifying whether faces can be detected in them or not. The computer technology in this study identifies both the human face location and size in any given digital images and also identifies only features of the face and disregards anything other thing, such as buildings, trees, cars and others. This makes it a specific case of an object-class detection system. The results of the analysis is represented in Table 1 below.

Table 1: Facial Detection Analysis

Video index	No. of images extracted	No. of detected face images	Time(s)	Accuracy	RF
1	891	887	60	99.55	0
2	889	888	60	99.89	0
3	888	887	60	99.89	0
4	891	891	60	100	0
5	890	890	60	100	0

#### 4.1.1. Interpretation of Parameters

- **Time:** The time captured for the acquisition of the individual videos is measured in seconds. Thirty (30) frames per seconds were recorded for each video capture.
- **Accuracy:** The accuracy of the images retrieved from the sample video sequences is given by the ratio of the number of detected face images to the extracted images. This is expressed as a percentage.

$$= \frac{\text{Number of detected face images}}{\text{Number of extracted images}} \times 100 \quad (17)$$

where the Number of Extracted Images is the addition of number of detected face images and the number of undetected face images.

Therefore, the accuracy depicts the relevance of the sample videos used during the experimental process.

- **Redundancy Factor (RF):** The Redundancy Factor (RF) is a measure that provides the extent of undetected face images upon completion of the experimental process for the face detection. It is one area which has been largely neglected by most researchers during statistical analysis. The RF is expressed as:

$$\frac{(\text{Total of images retrieved}) - (\text{Total of detected face images})}{\text{Total of detected face images}} \quad (18)$$

The ideal RF is 0, implying that faces are detected in all the extracted images. An RF greater than 0 means that, the detected face images are less than the extracted images. This means that the systems are not performing to the maximum expected level. On the other hand, an RF less than 0, depicts a system performing abysmally which can be considered an error.

The extracted images derived from the video clips are shown in Appendix A. Most of the video clips recorded 100% face detection. The reasons for the few clips that were not 100% in the face detection process are as follows:

- Most of the faces that could not be detected can be attributed to the face's size in relation to the image captured. This can be avoided by adjusting the focus of the camera.
- The target face is partly covered. There are occasions when the candidate's face is partly covered by clothing, eye glasses or hair.

### 4.2 Eye Detection Analysis

The Haar cascade classifier is used to detect the facial features such as the mouth, eyes, and nose. The classifier is trained using the Ada Boost algorithm and Haar feature algorithms. The results of the eye detection process is shown in Table 2.

**Table 2: Eye Detection Analysis**

Video index	No. of images extracted	No. of detected face images	No. of detected eye images	Time (s)	Accuracy	RF
1	891	887	887	60	100	0
2	889	888	888	60	100	0
3	888	887	887	60	100	0
4	891	891	890	60	99.89	0.0011236
5	890	890	889	60	99.89	0.00112486

### 4.3 Eyelid Movement Analysis (Based on Open and Closure)

The eye movement analysis is based on how often the eyelid state change from being open to close and the vice versa.

A cycle is defined as moving from the open state to the closed state to the open state. Each Individual has its maximum blink value. Beyond this indicates a fatigue state. A person is said to be in the normal state if it has a blink rate that is equal to the threshold of the maximum blink as shown in Table 3 below.

**Table 3: Eye Movement Analysis**

Video index	No. of detected face images	No. of detected eye images	Continuous detection		Freq	Rate	Fatigue
			Open Eye(OE)	Closed Eye(CE)			
1	887	887	300	150	10	15	F
2	888	888	400	80	10	8	W
3	887	887	510	56	10	5.6	W
4	891	890	300	150	10	15	F
5	890	889	462	86	10	8.6	W

**W = Widely Awake**  
**N = normal**  
**F = Fatigue**

#### Rate:

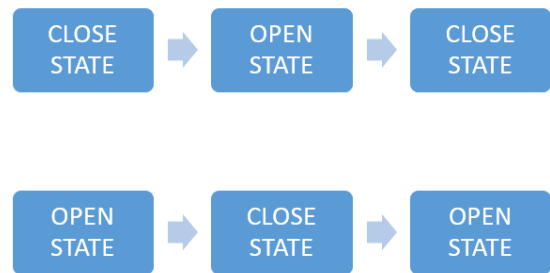
The rate, as used in the eyelid movement analysis, is calculated based on the number of closed eye frames found within ten consecutive frames. The rate is given by

$$Rate = \frac{Closed\ Eyes}{maximum\ blinks} \quad (19)$$

In the experiment, seventy (70) sample videos were used. Fatigue was detected in 23 of the seventy samples resulting in 34.28%. The duration of the video clips were considered by counting the number of closed and opened eyes within the continuous frames. Closed and opened eyes were detected in adverse conditions such as with and without eye glasses and also under different hair styles.

#### 4.3.1. Cycle Definition

The blink cycle could be defined in two different ways as shown in Fig. 5 below;



**Fig. 5: Cycle state definition**

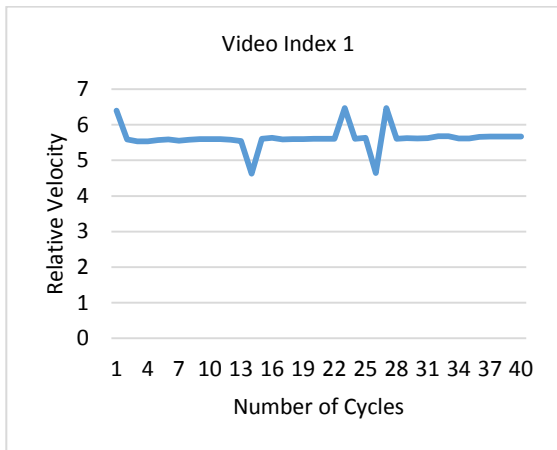
The above figure gives the general trend of the eyelid movement that shows whether a person is in fatigue state or not. However, because of differences in the eyelid feature, eye ball and eye blink rate, there is the need to establish a blink cycle and its relative velocity.

The table below shows the number of cycles found within each of the sample videos.

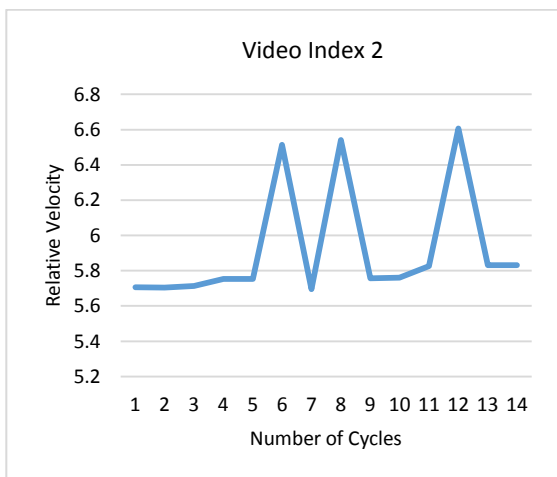
**Table 4: Cycle Definition Table**

Video index	No. of detected face images	No. of detected eye images	No. of Cycles	Continuous detection	
				Open Eye(OE)	Closed Eye(CE)
1	887	887	40	300	150
2	888	888	14	400	80
3	887	887	8	510	56
4	891	890	25	300	150
5	890	889	4	462	86

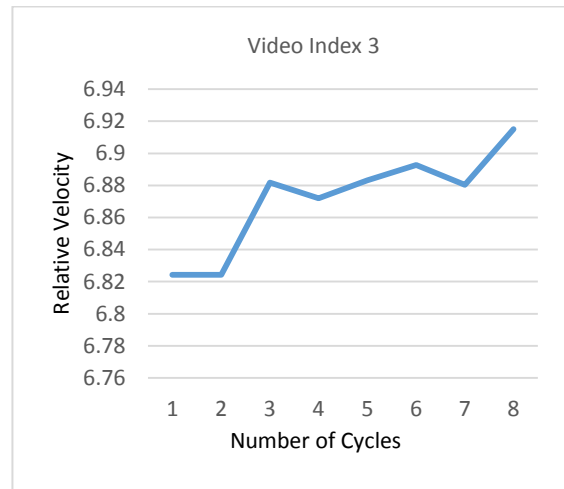
Fig. 6 below shows sample relative velocity of some cycles within the individual videos which is calculated by the distance of one cycle it travelled with respect to time.



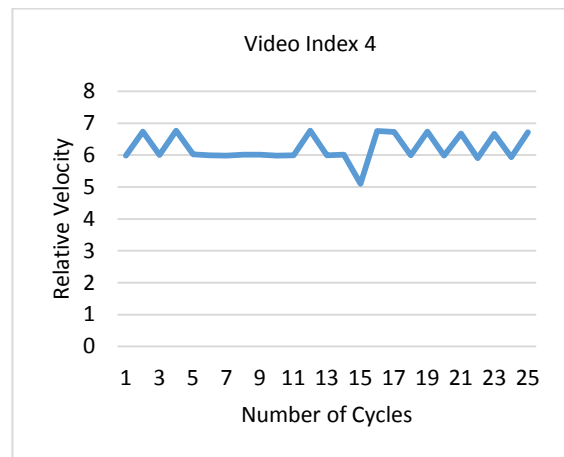
**Fig. 6: Relative eye velocity cycle graph of Video Index 1**



**Fig. 7: Relative eye velocity cycle graph of Video Index 2**



**Fig. 8: Relative eye velocity cycle graph of Video Index 3**



**Fig. 9: Relative eye velocity cycle graph of Video Index 4**

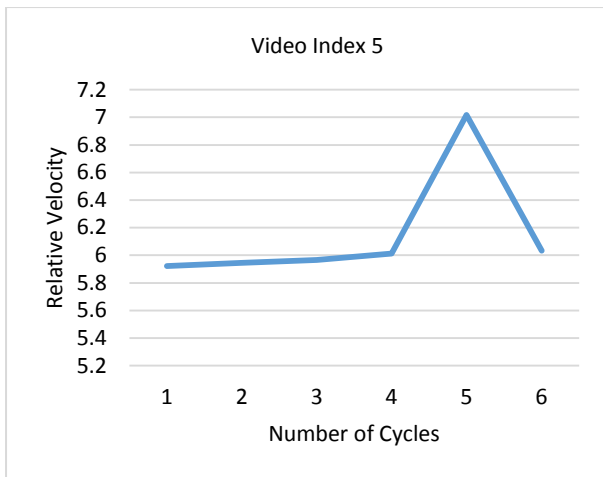


Fig. 10: Relative eye velocity cycle graph of Video Index 5

## 5. CONCLUSION

The establishment of a cycle between the open and closed eye states was performed by using the sample data from the dataset. The state of the eyes (whether opened or closed) is decided by the distance between the first two intensity changes found. The distance between the y – coordinates of the intensity changes is larger when the eyes are closed as compared to when the eyes are opened. When there are 5 consecutive frames that find the eye closed, the previous state before the closure of the eye begins to constitute a cycle.

The speed or the velocity of the eyelid movement can therefore be used to determine when an eye is opened or closed hence estimate or predict various fatigue levels.

In general, the normal blink rate of the eye is 10 blinks per second. If the eye is closed in more than ten (10) continuous frames, then the person is considered to be fatigued. The analysis gives the general trend of the eyelid movement that shows whether a person is in fatigue state or not, a person is said to be in fatigue state if the number of cycle per minute is less than thirty (30) otherwise the person is not in the fatigue state and maybe considered to be widely awake.

The Kalman filter used together with the blink cycle established would be a strong thresholding application that can be used in tracking and predicting the blinking rate of the eyelid.

## 6. REFERENCES

- [1] Pimplaskar D., Nagmode M.S., Borkar A., (2013), Real Time Eye Blinking Detection and Tracking Using OpencvDhaval Pimplaskar et al Int. Journal of Engineering Research and Applications www.ijera.com
- [2] Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N., Laurendeau, D.(2007): Real-time eye blink detection with GPU-based SIFT tracking. In: Proceedings of the Fourth Canadian Conference on Computer and Robot Vision. CRV '07, Washington, DC, USA, IEEE Computer Society, 2007, pp. 481–487.
- [3] Divjak, M., Bischof, H.(2009): Eye blink based fatigue detection for prevention of Computer Vision Syndrome. In: IAPR Conference on Machine Vision Applications (MVA 2009), 2009, pp.350–353.
- [4] Liting,W., Xiaoqing, D., Changsong, L.,Wang, K.(2009): Eye Blink Detection Based on Eye Contour extraction. In: Image Processing: Algorithms and Systems, SPIE Electronics Imaging, 2009, p. 72450.
- [5] Ayudhaya, C., Srinark, T.(2009): A method for a real time eye blink detection and its applications. In: The 6th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2009, pp. 25 – 30.
- [6] Khilari R, (2010). “Iris Tracking and Blink Detection for Human Computer Interaction Using a Low Resolution Webcam.”
- [7] Mardiyanto R., Arai K., (2010) “Real Time Blinking Detection Based on Gabor filter”.International Journal of recent Trends in Human Computer Interactions (HCI) Vol 1: Issue 3, Dec 2010
- [8] Arai, K., Mardiyanto, R (2011): Comparative Study on Blink Detection and Gaze Estimation Methods for HCI, in Particular, Gabor Filter Utilized Blink Detection Method. In: Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations. ITNG '11, Washington, DC, USA, IEEE Computer Society, 2011, pp. 441–446.
- [9] Galab M.K., Abdalkader H.M., Zayed. H.H., “Adaptive Real Time Eye-Blink Detection System” International Journal of Computer Application (0975-8887) Vol. 99. - No. 5 August 2014
- [10] Salehian S., Far B., (2015) “Embedded Real Time Blink Detection System for Driver Fatigue Monitoring.” Department of Electrical and Computer Engineering University of Calgary. 2015.
- [11] Akshatha .S., (2016) “Eye Blink Detection Using Adaboost Approach and Morphological Operation” International journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. Vol 5, Issue 4, April 2016.














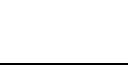



## 7. APPENDIX

### A. SAMPLE FACE IMAGES



### APPENDIX B. SEGMENTED EYE INTENSITY VALUES

S/N	BINARIZED IMAGE	INTENSITY VALUES	S/N	BINARIZED IMAGE	INTENSITY VALUES
1		0.377358490566037	9		0.312030075187969
2		0.415637860082304	10		0.344863731656184
3		0.375786163522012	11		0.332304526748971
4		0.329729729729729	12		0.334876543209876
5		0.322911051212938	13		0.328301886792452
6		0.300838574423480	14		0.323989218328841
7		0.400181488203266	15		0.326519916142557
8		0.382749326145552			