Visual Fatigue Detection, Monitoring and Prevention System

Journal:	*KSII Transactions on Internet and Information Systems*
Manuscript ID	TIIS-RP-2016-Jan-0129
Manuscript Type:	Regular Paper
Date Submitted by the Author:	22-Jan-2016
Complete List of Authors:	Rajiv, Pooshkar ; Birla Institute of Technology, Electronics And Communication Engineering Raj, Rohit; National Institute of Technology Patna, Computer Science and Engineering Kumar, Prabhat; National Institute of Technology Patna, Computer Science and Engineering
Keywords of your Paper:	Computer Vision Systems, Visual Fatigue, Eyelid State Detection, Smart Systems, Image Processing

Visual Fatigue Detection, Monitoring and Prevention System

Pooshkar Rajiv[1*], Rohit Raj[2*] and Prabhat Kumar[2*]
[1]Electronics and Communication Engineering, BIT Mesra
Ranchi – India.
[E-mail: pooshkar01@gmail.com]
[2]Computer Science and Engineering, NIT Patna
Patna, Bihar - India
[E-mail: rohit124197@nitp.ac.in]
[E-mail: prabhat@nitp.ac.in]
*Equal Contribution

Abstract

Visual fatigue is a widespread and ever-growing problem affecting all age groups of society. The problem manifolds due to our modern lifestyle which is heavily dependent on use of electronic display devices. Based on year 2000 data of USA, 75% of jobs involved use of computers. As per a study done in University of Benin, Nigeria in 2007 shows that the mean number of hours spent in front of a computer screen by an average individual is 6 ±3 hours/day. Researches show that more than 90% of the computer users experience some form of eye discomfort and can be affected by the Computer Vision Syndrome. These visual problems range from dry eyes, eye strain and headache to even blurred vision. The work in this paper uses cameras of devices to monitor our eyes' blinking pattern and its use duration using proposed algorithms. Novel image processing techniques are used to check the room lighting conditions and timely alerts are generated automatically when required. It further monitors eye distance from screen to detect visual fatigue. This data is processed using proposed computer vision algorithms for giving the user periodic alert to prevent vision problems. The algorithm developed has been implemented for open source operating systems. Further, results obtained in the form of mesh plots and figures have been analyzed and validated through real life simulations.

Keywords: Computer Vision, Visual Fatigue, Eyelid State Detection (ESD), Smart Systems, Image Processing.

1. Introduction

Eyes are one of the most vital organs of our body. They also get fatigued as it works without rest, often unnoticed, during our awake time. The problem manifolds due to our modern lifestyle which is ever increasingly dependent on the use of electronic display devices. Our eyes, despite containing the strongest muscles in our body also gets tired, strained and fatigued when we sit in front of computer, TV, handheld devices etc. for hours at a stretch. Eyes help us to see and visualize things in our everyday life. With the advent of computers, our work has shifted primarily on them.

Visual fatigue is an ever increasing eye problem, that most of the population is unaware of. Computer Vision Syndrome is a widely spreading but largely unknown epidemic among computer users [1]. Even the small percentage of people who are aware of the problem are helpless because modern lifestyle has forced them to spend extended hours in closed range activities in front of visual devices. With the rapid advent of information technology boom, the numbers have risen exponentially. According to the United States Census Bureau [2], the use of computers in household has more than doubled between the periods of 1997 to 2011. It jumped from 36.6% to 75.6% in 2011 [2]. Moreover, the young generations are more prone to visual fatigue as shown in the census data. A total of 82.2% of people in the age group 18-34 years lived in a household with at least a single computer as compared to 61.8% of people in the age group of 65+ years [2]. The education attainment and job nature is also directly proportional to the use of computers. This is shown by the fact that more than in US in the year 2011, 93.1% of people used computers who had a bachelor's degree or higher. This is fairly high when compared to 70.9% of people with only a high school degree.

Prolonged hours of sitting before the computer causes an array of eye problems which are collectively known as Computer Vision Syndrome (CVS). These visual problems range from dry eyes, eye strain and headache to even blurred vision. Even children are spending too much time in front of digital screen and are more prone to the above mentioned problems. They easily adapt to the surroundings and many a times ignore the symptoms of visual fatigue. Dry eyes are one of the major consequences of chronic visual fatigue. Analysis of blink pattern can lead to the identification of various disorders.[3] It has been suggested by Bhelm et.al. [4] that dry eyes are also related with low blink rate. Working for longer periods in front of the computer may result in a lower blink rate and thus inadvertently cause dry eyes. This has resulted in a challenging situation both for researchers as well as display manufacturers for solutions regarding prevention of visual fatigue related issues.

A case study for the evaluation of blink related problems as done in the University of Benin, Nigeria in 2007 by Stella C Chiemeke [5] shows that the mean number of hours spent in front of a computer screen by an average individual is 6 ±3 hours/day. The hours spent will be more in developed economies. CVS results in diverse symptoms which may vary from individual to individual. TABLE 1 shows the percentage of daily users who are suffering from various visual symptoms versus severity levels of the same.

S/N	Symptoms	Severe	Moderate	Mild	Never
1	Eyestrain	21(20.4%)	23(22.3%)	55(53.4%)	04(3.9%)
2	Blurred distance vision	08(7.8%)	39(37.9%)	39(37.9%)	17(16.5%)
3	Headache	04(3.9%)	25(24.3%)	55(53.4%)	19(18.4%)
4	Production of tears	01(1.0%)	12(11.7%)	37(35.9%)	53(51.5%)
5	Redness of eyes	04(3.9%)	26(25.2%)	38(36.9%)	35(34.0%)
6	Inability to deal with messiness of reality	-	11(10.7%)	36(35.0%)	56(54.4%)
7	Double vision	01(1.0%)	29(28.2%)	45(43.7%)	28(27.2%)
8	Letters on screen run together	01(1.0%)	08(7.8%)	27(26.2%)	67(65.0%)
9	Night vision decreased	03(2.9%)	11(10.7%)	31(30.1%)	58(56.3%)
10	Need for visual break	06(5.8%)	30(29.1%)	25(24.3%)	42(40.8%)
11	Circles of light around objects	09(8.7%)	13(12.6%)	34(33.0%)	47(45.6%)

Table 1 : Eye problems and their severity [5]

The above table clearly demonstrates that a large number of people are unknowingly suffering from a number of symptoms of CVS and the numbers are too high for the threat to be ignored. The users, too have limited options as they have no mechanism at their disposal that can inform them about the current status of their eyes and alert them in time for taking remedial measures.

Doctors of USA optometric society have come up with a novel solution called 20-20-20 rule to prevent visual fatigue while working on the display units. They suggests taking a 20 seconds break every 20 minutes and focus on object at 20 feet. However, it may seem an easy solution, implementing it in real time is a challenge in itself. People, particularly children become engrossed in their work and tend to forget to take break while working on their computers. Their parents too have no mechanism of keeping track of the time spent by their wards in front of the digital screens. Even knowledgeable people at present cannot know their eye use duration, blinking pattern, lighting condition and eye distance from screen to detect visual fatigue threshold during work and hence are unable to take preventive measures.The most important causes of CVS are low blink rate, increased corneal exposure to monitor and low lighting in the room while working.

The major causes for eye related problems are prolonged exposure to computer display units and continous visual fatigue. This paper proposes a combination of many novel computer vision algorithms to continously monitor the eye state and ambient environment in which the user is working. Computer vision refers to the branch of computer science which analyses image, accquires data from it and infers meaningful information using this data. The proposed work continously checks the blink rate of the user and informs him if the rate falls below a set threshold. It also measures the ambient lightning conditions of the room in which he is working. We have also proposed an innovative method to check the distance of the users' eyes from the screen and alert him if he is closer than a safe viewing distance.

The program also monitors the total duration for which the user has been active on the computer. This information is also saved in the database and is updated frequently. This helps the program to alert the user if he's been working on the computer for long period of time. This will particularly be helpful for parents who want to see how the status of their children's eyes varied through the day and how much time they spent in front of the computer. In addition, the ambient intensity of light in the room is also measured either by using the concept of image binarization or by using the Y value of the YUV frame and checking if the value is above the pre-set threshold. The user is informed if poor lighting conditions prevail for a long time. It has also been designed to alert the user if the distance between the display devices and eyes is very small. A face to image ratio is calculated to estimate the distance of the user from the computer screen.

The detailed step by step explanation of the used algorithm, its technical implementation and simulation results have been provided in the paper. Section II gives related work. Section III gives an insight on technical background. Section IV deals with a plausible solution of the issues raised. Section V gives an insight into the proposed architecture and the various proposed computer vision algortihms. Section VI and Section VII presents the simulation parameters and results based on our approach; it also analyzes the output of the implemented algorithm. The paper ends with a concluding remarks in section VIII.

2. Related Work

This paper has proposed a comprehensive eye health evaluation, based upon the variation in blink rate and other ambient light detection techniques. The proposed model is able to use the webcams that are in built in laptops or low cost web cameras for continuous capture of video of the user and to provide him with an accurate estimation of his eye health, using the variation in blink rates. Reliable detection of blinks is an integral part of this system and several eye tracking and blink estimation[6] [7] aproaches addressing similar issues have been published, yet none of them address all the mentioned constraints .

For example, Morris et. al. [8] proposed a blink detection system based on variance map calculation and eye corner analysis. They claim that it is a real time system working on 320 x 240 images with 95% true positives. However the use of variance map calculation makes it susceptible to errors caused by head movement leading to sharp decline in it's efficiency.

Sirohey et. al. [9] presented an extensive approach for determining eye blinks by locating eye corners, eyelids or iris and analysizing their movement. Motion estimation is accomplished by using normal optical flow. Head motion is detached from the eye movement by modelling it separately using an affine model, thereby significantly increasing the computations required. Authors have claimed that they can track the motion of eyelids and iris more than 90 percent of the time, though not in real time. The fact that the method does not work in real time and is computationally exhaustive places severe constraints on it.

Chau and Betke [10] have introduced a low computational complexity system which detects eye blinks by corelation with open eye template. For 320 x 240 images, a blink detection accuracy of 95% was reported. The main benefit of this approach is the fact that it is computationally less exhaustive. This model did not use the generated data for comprehensive eye evaluation, distance estimation or lighting estimation i.e the proposed model was limited to only blink detection which severly limited its array of area it could be used.

G.Pan et. al. [11] implemented a booster classifier to detect the degree of eye closure. Hidden Markov Model was used for modelling of changes in eye states. They claim that their method operates in real time detecting more than 96% of eye blinks on 320 x 240 webcam images. Being very specifically trained with the help of typical eye motion, this method lacks in robustness and is ineffective in dealing with non standard eye motions.

S.Han et al [12] have tried to implement a model for eye tracking and blink detction for mobile users, but they do not provide comprehensive evaluation of the use pattern, lighting conditions, visual fatigue alerts etc. Orozco [13] proposed a two appearence based tracker model, where the first one tracks the movement, while the second one tracks the eyelids. With low resolution input video and a simple appearence model, they report that it runs in real time and that too with good trking output. The authors didn't try to detect and utilize eye blinks through this method;however it could have been usd for this purpose. Ran Lui et.al. [14] talked about depth adjustment method for visual fatigue reduaction in 3D systems. The percieved depth from 3D analyzed image is expressed as a function of various parameters and they provide reduction in visual fatigue while watching 3D sterioscopic systems. However, such mechanisms are constrained for only a particula renvironment and does not address the fatigue reduction in general screens.

Most of the approaches mentioned above detect blinks by locating separate eye parts such as iris, eyelids etc. Quality of detection is directly linked to the quality of part extraction, however in our paper it is not so. The YUV colorspace and skin detection algorithnms have been used to create a fail safe skin color threshold based algorithm which gives a binary image of the eye. Here, the ratio of black to white pixels is used for eye state detection. This method is computationally less exhaustive and very accurate.

Some of the proposed works did have a good eye blink evaluation technique [15] [16] [17] but did not provide a detailed evaluation of the eye health and VDT using habits of the individual.

3. Technical background

The blink detection technique as proposed in this paper has been implemented by using the universal threshold for skin component detection i.e. skin lesion and segmentation as proposed in [18] [19].

3.1 YUV Colorspace

YUV format is an 8 bit data colorspace which is used for encoding an RGB image in accordance with human perception, allowing reduced bandwidth for chrominance components. The fact that the masking is done in accordance to human components allows us to use this colorspace for the application of the skin thresholds.

$$\begin{aligned} Y &= \frac{R + 2G + B}{5} \\ U &= R - G \\ V &= B - G \end{aligned} \tag{1}$$

3.2 Skin Segmentation As shown in [18], the universal skin detection threshold for all colours of skin for an image converted into YUV colorspace is given by :-

$$10<U<70 \ AND \ -40<V<11 \tag{2}$$

This universal threshold is found to be very successful and is very effective for the purpose of skin detection. We made use of this universal skin detection threshold in this paper for the purpose of generating a binary image of the eye region, showcasing the skin components with white and non-skin components with black pixels.

4. Proposed Solution and its Architecture

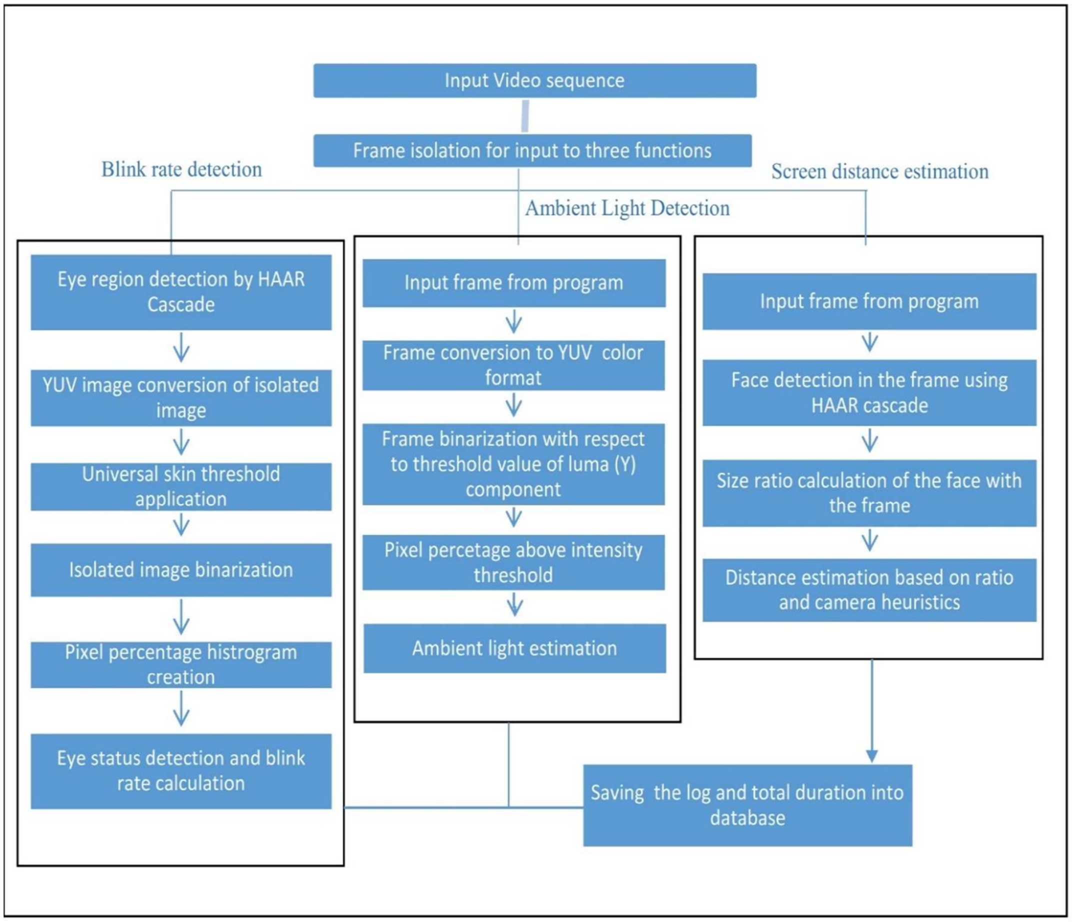

Fig. 1. Proposed Achitecture

The problem of visual fatigue and CVS can be easily avoided if an individual can keep track of his blinking pattern variation, the time he spends in front of a digital display device, the lighting condition of his surroundings and his distance from the computer screen.This paper implements a system which helps in timely identifying and alerting the user of all the above four causes, thus resulting in control of Computer Vision Syndrome. The proposed system was implemented in an UNIX system. It takes continuous video input from the webcam, processes the frames and maintains a database of blink pattern variation for possible identification of CVS and visual fatigue detection. It keeps track of the time spent in front of a VDT (Visual Display Terminal), the lighting condition of the room and the distance from screen. Timely real time alerts and reminders aregiven to the user based upon his activities. The algorithm proposed in this paper has been implemented in python using OpenCV . The blink rate of every computer user is recorded and stored in a database. This provides information about the blink pattern variation of each user of the computer, which can be analysed in detail for identification of possible CVS or visual fatigue (significant decrease in blink rate might be indicative of future CVS problem). These parameters are compared with standard values and suitable preventive alerts are given in real-time for immediate improvements.

The work has used a new innovative method for blink detection based on skin detection threshold instead of using the traditional methods such as Eyelid State Detection (ESD) [20] , Eye contour extraction [21], eye reanimation [22] and image flow [23] etc. This method is computationally less exhaustive than the other currently used methods and thus is more efficient. The Skin threshold of a YUV image helps in image binarization which is used to obtain an image with black pixels for non-skin and white pixel for skin components. This binary image is used for subsequent blink detection and obtaining the blink pattern variation. The input frame is converted into YUV format and the luma value of the frame is used for brightness of picture calculation. If the input frame has a very low luma value, the user is notified of the same and is asked to adjust the brightness. Similarly, the nearness of the user's face to the computer is calculated based on the calculation of the aspect ratio between his face and the entire frame to warn the user of extreme proximity to the screen

The database that stores all of this information can be analysed for improvement of visual behaviour and also for checking for other eye problems. The architecture of the system that has been implemented is shown in **Fig. 1**.
The proposed algorithm for blink detection, was developed using the universal skin threshold for YUV images. A brief explanation of the same has been given in the following section.

5. Proposed Algorithm

The block diagram of proposed work is given in **Fig. 1**. Major parts of the experiment are skin, eyes, blink detection and calculation of blink rate of the individual and its comparison with the preset thresholds. Blink rate detection by the application starts with the isolation of the eye region of the user by the HAAR cascade algorithm which is then modified into a different color space namely YUV. The UV component of this color scheme provides the color component of the image. As the human skin resides within certain range of these components, the filtering of the skin area can be done by applying various thresholds for the U component of the image. This is then followed by the morphological filtering to reduce the chances of a false positive.

5.1 Blink Rate Calculation

5.1.1 Eye Region Detection

Point feature tracking [29] and object detection methods like HAAR feature based classifier as proposed by Paul Viola [24] have been found to be very effective in detecting eyes, face, mouth etc. It is a machine learning approach in which the training of the cascade function is done by using a number of positive and negative images, so that it can be efficiently used for object detection. A python program written with the opencv2 module is used to first load the required XML (Extensible Markup Language) classifiers and the frame image thus obtained is used as the input and the smallest rectangular area that can cover both the eyes is computed. If both the eyes are not detected, the program sleeps for a minute and then tries to detect the eyes again. The sleeping process is reiterated until eyes are detected successfully. The eye region is then separated in a new figure and changed into YUV form.

A sequence of images of the face is used for finding out the skin color histogram. The HAAR cascade classifier provides flexibility to detect both the eyes either as a single entity or as two separate entities. Our work used this classifier to detect both eyes as a single object .The located eyes are then stored into a separate file for further morphological operations. The eye

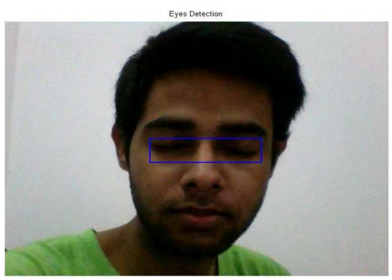

Fig. 2. Closed Eyes Detected

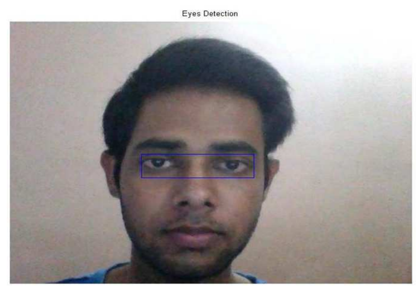

Fig. 3. Open Eye Detection

region after detection using the above mentioned method for a particular frame, have been shown in **Fig. 2** and **Fig. 3**.

The captured frame is then also converted into YUV format separately to find the average luma (brightness) value in the image for estimation of the lighting conditions of the room.

5.1.2 YUV histogram is obtained and Brightness is calculated

The separated image of the eye is converted into YUV format using Algorithm 1 and a pixel probability map is created over the eye region from the histogram back projection of the YUV image of the eyes. Which is then used for determining the current status of the eyes i.e. if they eyes are open or closed. The universally applicable skin threshold is applied on the separated YUV image of the eye region to annotate the skin components present in the area and obtain a binary image of only the eye area. Image indexing according to weighted color histogram and binarization is done. However, our work is quite different from the existing techniques as mentioned in [25] [26] [27] [28].

Algorithm 1: Convert to YUV

```
For i=1 to last pixel of x-axis
    For j=1 to last pixel of y-axis
        Y_value= (r_value + 2*g_value + b_value)/5
        U_value= r_value-g_value
        V_value=b_value-g_value
Return YUV_value
```

Thresholding is used for skin color segmentation, Skin Lesion Segmentation and ESD (Eyelid State Detection). Threshold setting is important to obtain a binary representation of the eye region, i.e. white for representing skin and black for representing non skin components such as eyebrows, eye lids, iris etc. The probability of finding the location of the object (skin) is dictated by the number of white pixels (denoting skin) that are present in the back projected image obtained from the YUV image of the eyes.

5.1.3 Open/closed Eye Detection from Binary Back Projected Image

The U and V values of the YUV image is used to detect the skin pixels and a normalized back projected image histogram is obtained, in which skin components are depicted by white and non-skin components by black pixels. The isolated image of the closed eyes under ideal conditions has non-skin components in the form of eye lashes and thus very little black pixels. Thus, after the threshold of the presence of the skin pixels has been decided, closed eyes are detected as a higher percentage of skin pixels whereas open eyes are estimated as a very low percentage of skin pixels with a high percentage of non-skin components. This estimation

Algorithm 2: Blink rate detection algorithm

```
1.  While webcam is running, do for 5 minutes:
2.      Extract the frame from video
3.      Load XML HAAR Based classifier "EyePairBig"
4.      Select size of rectangles to mark eye area
5.      Separate Eye Image in a JPEG File
6.      Find Size of Eye Image
7.      YUV_eye_Image= ConvertYUV (Size_Of_EyeImage, Eye_Image)
        /*Image Binarization*/
8.      For i,j=1 to last pixel of x and y axis
9.          If (10<U of YUV_Eye_image<70) and (-40<V of YUV_Image<11)
            /*Obtaining Back-projected histogram-white pixels*/
10.             YUV_eye_Image =255 /*Pixel is white*/
11.         Else
            /*Obtaining Back-projected histogram-black pixels*/
12.             YUV_eye_Image=0 /*Pixel is black*/
        /*Blink detection using percentage of black to white pixels*/
13.     For i,j=1 to last pixel of x and y axis
14.         If ( YUV_Eye_Image(I,j,2=255) /*white pixel-No eye*/
15.             counter_white++
16.         Else/*eye present*/
17.             counter_Black++
18.     Set threshold as 20% of size of binary eye image
19.     If (counter_black>threshold)
            /* Eye is open*/
20.         Open_detect = 1
21.     Else
            /* Eye is closed*/
22.         If (Open_detect ==1)
                /*increase blink count if it was closed earlier*/
23.             blink++
24.             Open_detect = 0
        /*average of 5 minutes */
25.     Blink_rate_avg=blink/5
26.     Return blink_rate_avg
```

gives the threshold setting for the current eye state .The threshold decided during the testing phase of the program dictates the percentage of white and black pixels that are present in the binary image.This work has assumed that the eyes are in an open state if the ratio of black to white pixels is greater than twenty percent and closed if the ratio is less than twenty.

Morphological operations such as erode and dilate are used to adjust and amplify the small differences between the back projected images of open and closed eyes. Erode is used to replace the current pixel by the local minimum of the neighborhood. Dilate is used to replace the current pixel by the local maximum. The morphological operations also allow us to get a back projected image that has only white pixels or a very small percentage of black pixels (Non skin colored components) for closed eyes and a higher percentage of black pixels in the eye region for open eyes.

Using this techniques for eye detection gives us a significant improvement over the previously used methods such as ESD etc. as it is computationally less exhaustive and is considerably faster.

5.1.4 Blink Rate Calculation

Open eye detection followed by closed eye detection is quantified as a single blink. A variable is used to update the counter after every consecutive blink detection. This counter runs for a minute and the current blink rate of the user is given as the value of counter at the end of every minute. The average blink rate is calculated continuously and is compared with a predefined threshold value. (As shown in the Algorithm 2). This threshold limit is set to 15 blinks per minute which corresponds to a normal human eye blink rate. If the blink rate of the user is found to be less than the predefined threshold value, the program running in the background will prompt a popup windows encouraging user to blink more.

5.1.5 Database connection with the program

A predefined database has to be created to save the blink rate of the person. In our work, MySQL has been used to maintain the database of the users' blink rate. The table inside the database saves the various useful information about the blink rate of the person. This may include blink rate during the various intervals of the day. As the blink rate may vary while using different applications, the value of the tuples for the current user has to be changed by the python script periodically.

The schema of the database will consist of multiple tables, one for each user of the system. Each table will have three attributes (columns) namely date and the blink rate and the total usage duration. Every value of the tuple (rows) will correspond to the current date of operation, blink rate per minute over the average of past five minutes and the total time the user has logged on in this session. As the tables for every user also stores the data about the usage duration for the user, it will enable proper monitoring of the duration for which the user has worked in a particular day. A new tuple will be added every day to monitor the usage pattern over a longer period of time.

The program automatically updates the database after every five minutes when it receives the new blink rate values. Along with updating the blink rate values, the python program also updates the duration attribute of the tuple which stores the total duration for which the user has worked in this session. This is done in order to warn the user about sustained exposure to the computer screen.

5.2 Ambient Light detection through Luma value

The YUV space is again utilized to detect the ambient light condition of the room. As opposed to chrominance (U, V), the luma (Y) value denotes the brightness of an image. Hence, the Y value of the entire frame denotes the brightness parameter of the image.

Hence, lower the luma value, darker the image. If the light is not sufficient, the image will be darker. In a darker image, the average value of the luma component of the pixel will be lower.

For this detection, the percentage of pixels with Y values which are less than the pre-set threshold are calculated. If the percentage is less than 60 (and has continued to be below 60 for five continuous measuring instances), the system recommends the user to adjust the brightness of his surroundings to an ambient value as it may cause strain on his eyes. This ambient light detection is achieved using Algorithm 3.

Algorithm 3: Ambient light Detection

```
While webcam is recording
    Extract the frame from video
    Find Size of Input Frame
    YUV_Image = ConvertYUV (Size_Of_Image, Input_Frame)
    For i,j=1 to last pixel of x and y axis
        /*check luma of pixels more than threshold value*/
        If ( YUV_Image(I,j,1)) > 125
            Number_good ++
    If (Number_good/ (size of frame)) <0.6
        Return true
    Else
        Return false
```


5.3 Optimal Viewing Distance Calculation

HAAR Cascade classifier for the face object is used to separately mark and store the image of the face in a JPEG file. As the video capture resolution by the webcam is constant, the ratio of face and the whole image will only change if there is a relative movement of the face and the video capturing device. It happens due to perspective i.e. closer objects looks bigger than remote objects. A higher ratio will imply lesser distance between the eyes and the display device.

To estimate the distance between the screen and the face, the ratio of the size of the face to that of the entire image is calculated. If this ratio is greater than 20 percent, user is notified that he has come very close to the computer screen and is causing harm to his eyes.

Alert Mechanism

The script will run in the background and will monitor the blink rate as well as the total duration for which the user has worked. The alerts will however be of three types:

- First, the user will be alerted for his lowered blink rate while working behind the computer.
- Second, the program will also detect the ambient light present in the room to alert if the light present is low.
- Third, by calculating the ratio of face and whole image and estimating the distance of the user from the computer screen, the program will also alert the user if the screen is too close for eye’s comfort.

- Fourth, the total duration the user has worked on will be checked from the database and the user will be alerted if the duration has exceed the prescribed limits.

When the use logs off of his computer, before being closed the total time duration in the database will be again reset to zero by the script.

Algorithm 4: Eye & viewing distance monitoring program

```
1.  While computer is running do
        /*find average blink rate*/
        /*This functions returns value every 5 minutes */
2.      Avg_blink_rate=blink_rate_detect();
3.      If (avg_blink_rate <threshold)
4.          Alert("low Bink Rate")
        /*find ambient light */
5.      Light=Ambient_light_detect()
6.      If (light is false)
7.          Alert ("low light")
8.      If (Viewing Distance is low)
9.          Alert("Too close to monitor")
    /*find and update total duration*/
10. Duration=Database_log_monitor(avg_blink_rate)
11.     If duration > daily time limit
12.         Alert ("Duration has exceeded")
```

The overall eye monitoring and alert mechanism has been shown in Algorithm 4.

6. Simulation Parameters

The simulations were done in *Matlab 2012 v. Ra* . For optimal performance and better fatigue detection, the use of eye accessories such as glasses should be avoided. The following parameters for the simulation of fatigue detection were used:

- A probability map of white and black pixels was created using the histogram back projections. The appropriate threshold value for skin pixels was successfully set in the testing phase of the code. Now, the ratio of black to white pixels (non-skin to skin components) was determined which was used for finding if the eyes were closed or open.
- As the blink rate may drop due to temporary external factors, instead of updating every minute, the database was updated every few minutes as to provide comprehensive and normalized values. The total time duration will also be updated simultaneously to ensure all-round eye health monitoring while using the device.
- For ambient light estimation the total number of pixels in the frame which have the luma component less than 125 i.e. midway between black and white were used obtained. This value is divided by the total pixels to give ratio of white pixels to total pixels.
- To detect lighting condition, a threshold of 60% on this ratio has to be applied to determine if the light in the room is enough. This ensures that the number which are brighter than average are more abundant than their darker counterparts.

- The threshold for the ratio of size of face to size of image was taken as 20 percent, if the ratio was greater than this, the program presumed extreme proximity to the screen and the user was notified that he has come very close to the computer screen.

7. Results and Analysis

The proposed work successfully seperates eye region and detects eye status of open and closed followed by blink rate calculation. The surface plots for the variations in the intensity of the captured image for estimation of lighting conditions was done. The distance from the screen is approximated by comparing the size ratio of face to the frame.

7.1 Blink Rate Calculation

Fig. 4 and Fig. 5 demonstrate the converted YUV image of the eyes using algorithm 1 which is now subjected to the skin threshold. The skin threshold color range is then applied to these two images to find the skin components present in it.

Fig. 4. YUV eye image

Fig. 5. YUV Image of Closed Eyes

In the following figures, the non skin components are shown as black areas and this feature is utilised to detect the eye status using algorithm 2. Fig. 6 has comparitevely large black areas as compared to Fig. 7 and it can easily be concluded that they are open and closed respectively. It happens due to larger non-skin components such as corneal and pupil region in the open eyes whereas in Fig. 7, only eye lashes form the non-skin components along with white areas as the skin components (eyelids in this case). This process checks for eye status every minute but the reports are send to user after 5 minutes to eliminate momentary deviations which needs to be ignored.

Fig. 6. Binary back projected Histogram of Open Eyes

Fig. 7. Binary Back Projection of Closed Eyes

7.2 Intensity Mesh Plots

Fig. 8. Bad lighting image

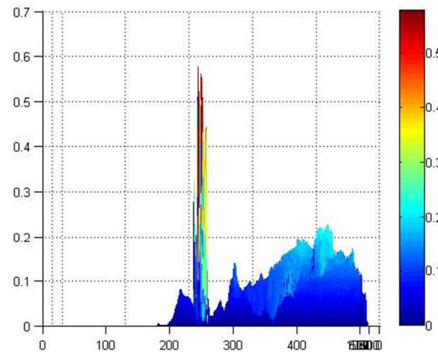

Fig. 9. Intensity Plot-Bad lighting

The frame captured through webcam is analysed to detect room lighting status. This frame is converted black and white for the sake of reduced computation as per algorithm 3. The total number of pixels in the frame which have the luma component less than 125 i.e. midway between black and white are obtained. This value is divided by the total pixels to give ratio of white pixels to total pixels in the frame. A threshold value is then taken to ensure that the brighter pixels are more abundant in the frame than their darker counterparts. For obtaining this threshhold value two images were captures as shown in Fig 8 and Fig 10 with bad room light and good room light respectively. The 3D mesh plots of both the figures were obtained as shown in **Fig. 9** and **Fig. 11** respectively. These two mesh plots provide an interactive visual representation of the underlying contrasting and dynamically different lightning conditions of the room. It can be seen that in **Fig. 11**, the intensity value was constantly above the threshold of 0.6; varying primarily between 0.6 and 0.9. **Fig. 9** i.e. the intensity plot of bad lighting represents a contrasting mesh plot The intensity values were constantly lower than the set threshold of 0.6. In fact, the intensity value did not even once crossed the threshold value and was constantly varied between 0.1 and 0.2. Hence, this work has considered a threshold value of 60% as a measure of good lightning condition. The user is alerted if the ambient lighting conditions goes below this threshold and is requested to change the room lighting

Fig. 10. Good Lighting image

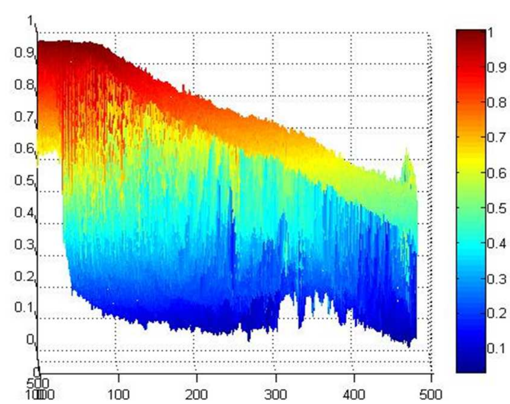

Fig. 11. Intensity Plot- Good Light

environment.

7.3 Viewing distance ratio

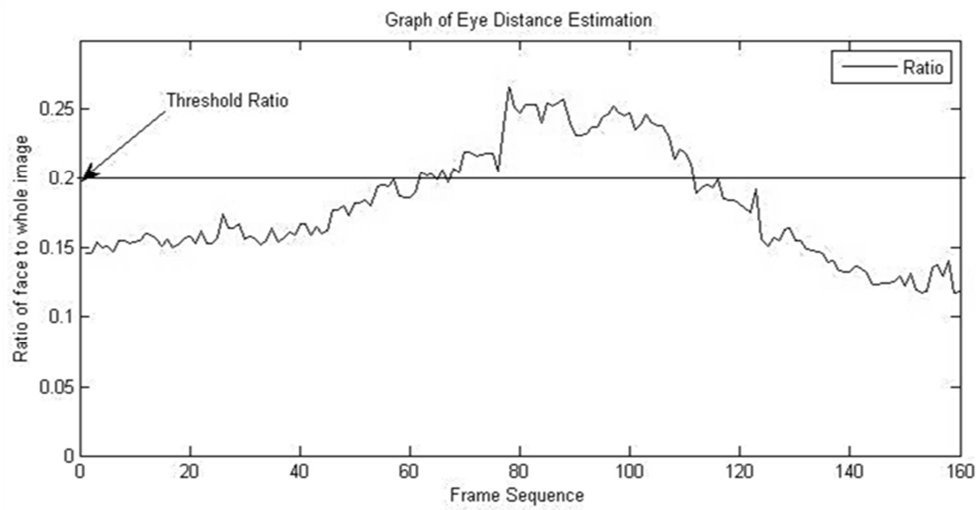

Fig. 12. Viewing distance estimation

The output of the algorithm 4 applied on a normal user for a sequence of 160 frames has been shown in **Fig. 12.** It shows the safe viewing distance estimation as the ratio of face to whole frame captured by the webcam. The distance between the display device and eyes was assessed using this ratio. The x-axis represents the frame number of the video sequence while the y-axis denotes the ratio of the size of face to total image.

The recommended distance between screen and human eye should be in the range of 16 to 30 inches [6]. This may vary depending on the individual, screen size and screen type. This work has kept the threshold for alerting the user to 19 inches which translated to a ratio of 0.2. Different cameras may take images of different sizes and resolutions, however the ratio of face to image will always be approximately same. Hence, a universal ratio for a safe threshold is used.

8. Conclusion and Future Applications

This paper proposed a system which provided a personalised fatigue monitoring, alert and remedy recommendation system using computer vision algorithms. The algorithm used to develop the system is very efficient and robust. The implementation of the algorithm and system development was done in a UNIX based system. However, the same system can also be implemented in Windows, Mac etc. The long term aim is to develop a system which can be integrated with the hardware and help in eye protection for common man. This system has potential to decrease the ever increasing problem of visual fatigueness resulting in an improved optical heath in the society. It may result in less frequent visits to doctors, thus saving a lot of time and resources, both for patients and the doctors. Future work can lead to cloud based solutions which can be integrated by web browsers or E-Mail clients for widespread use and benefit. Depending upon type of content on the screen, the blink pattern of the user may vary. The present work has not taken this aspect into consideration. Future works may include this and further improve the suitability of the algorithm. The timing settings of picture capturing, analysis and reporting has been taken as fixed value in this work, with a provision of manual change by the user. This may be automated in future and intelling AI techniques can be developed which will vary these times depending on user habits resulting in further optimization.

Acknowledgement

The Authors woud like to acknowledge that Pooshkar Rajiv, Rohit Raj and Prabhat Kumar have contributed equally to this work and they should be considered co-first authors. We would also like to show our gratitude to the Professors of the Department of Computer Science, NIT Patna for their guidance and support.

References

[1] Yan, Z., Hu, L., Chen, H., Lu, F. ,"Computer Vision Syndrome: A widely spreading but largely unknown epidemic among computer users", Computers in Human Behavior, vol. 24, no. 5, pp. 2026 – 2042, 2008.

[2] File T, C Ryan. Computer and Internet Use in the United States: 2013. Available from: http://www.census.gov/content/dam/Census/library/publications/2014/acs/acs-28.pdf

[3] Anna Rita Bentivoglio,Susan B. Bressman, Emanuele Cassetta, Donatella Carretta, Pietro Tonali, Alberto Albanese,"Analysis Of Blink Patterns in Normal Subjects",Movement Disorders Vol. 12, No. 6, pp. 1028-1034, 1997

[4] Clayton Blehm, Seema Vishnu, Ashbala Khattak, Shrabanee Mitra, Richard W. Yee, "Computer Vision Syndrome: A Review", Survey of Ophthalmology, Volume 50; Issue 3, pp 253-262, June 2011.

[5] Stella C. Chiemeke, Allen E Akhahowa, Olajire B.Ajayi, "Evaluation of vision related problems amongst computer users : A Case Study in University of Benin "Proceedings of the World Congress on Engineering 2007 Vol I WCE 2007, July 2 - 4, 2007, London, U.K.

[6] Hayes JR, Sheedy JE, Stelmack JA, Heaney CA., "Computer use, symptoms, and quality of life." Optom Vis Sci, vol 84, pp 738-44, Aug 2007.

[7] K. Grauman, M. Betke, J. Gips, and G. R. Bradski, "Communication via eye blinks-detection and duration analysis in real time, " in Proc. IEEE Comput. Soc., vol. 1, pp. 1010-1017, 2001.

[8] Morris, T. – Blenkhorn, P. – Zaidi, F, "Blink detection for real-time eye tracking", J. Netw. Comput. Appl. April 2002, 25, 2, pages 129–143. ISSN 1084-8045. doi: 10.1016/S1084-8045(02)90130-X. Available from: http://dx.doi.org/10.1016/S1084-8045(02)90130-X.

[9] S. Sirohey, A. Rosenfeld, Z. Duric: "A method of detecting and tracking irises and eyelids in video", Pattern Recognition, vol. 35, num. 6, pp. 1389-1401, 2002.

[10] M. Chau, M. Betke: "Real Time Eye Tracking and Blink Detection with USB Cameras", Boston University Computer Science Technical Report No. 2005-12, May 2005.

[11] Gang Pan, Lin Sun, Zhaohui Wu, Shihong Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera", Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1-8, 14-21, Oct. 2007.

[12] S. Han, S. Yang, J. Kim, and M. Gerla, "Eyeguardian: a framework of eye tracking and blink detection for mobile device users." in HotMobile. 13th Workshop on Mobile Computing Systems and Applications 2012, G. Borriello and R. K. Balan, Eds. ACM, 2012, p6.

[13] J. Orozco, et al.: "Real-time gaze tracking with appearance-based models", Machine Vision Applications, 2008.