Real Time Eye Blink Detection Interface

Rajana Nayak and Zhen Liu

Department of Computer Science

Lawrence Tech University, Fall 2018

Faculty Advisor: Dr. CJ Chung

Abstract

Robustness and accuracy in real time eye blink detection has been one of the most challenging problems in computer vision, and it has many applications such as driver fatigue analysis and face live detection. The existing methods toward eye blink detection includes traditional image processing method for eye blink computation, in which thresholds are specified to find the whites of the eyes and if the whites decrease for a period of time, an eye blink is calculated. In this paper, we present a method based on automatic tracking of facial landmarks to localize the eyes and eyelid contours. Automatic facial landmarks detectors are trained on an on iBUG 300-W dataset and show an outstanding performance in varying lighting conditions, facial expressions, or head orientation. The proposed method estimates the facial landmark positions and extracts the ratio between horizontal and vertical distance between eye contour – eye aspect ratio – for each video frame to examine the open and closed state of eyelids. To test the robustness of the algorithm, a testing interface has been applied for obtaining the benchmark data in tracking accuracy evaluation.

1. Introduction

As one of the most salient features of human face, the eyes play a crucial role in interpreting a person's intension and attention. Eye blinks are considered to be one of the most robust and intrinsic properties of human face. Psychological research reported that eye blinks can be used to interpret a person's intention [2]. For example, we tend to blink faster when feeling excited and bat our eyelids when speaking in public as well as when lying. Eye blink detection can be utilized in various fields such as in face live detection to distinguish photo from real person, in driver fatigue system to detect drivers' fatigue state, or in disease detection to capture altered blink rate, which may be an alert to some abnormal pathological cases [2]. Specifically, driver fatigue is a significant factor in lots of car accidents, and development of techniques to detect and prevent drowsiness at the wheel is a foremost challenge in the field of accident avoidance system. Due to the hazard that drowsiness presents on the road, we need methods to detect eye blinks to counteract the effects.

However, there are so many difficulties in eye blink estimation: eye patterns have large variation in appearance due to various factors, such as size, shape, pose, rotation, facial movement dynamics, the closed and open eyes, illumination conditions, or the occlusion by hairs. Given the complexity of the task, many efforts have been devoted: some build a generic eye model based on eye shapes, while others use template matching, where templates with open or closed eyes are learned and a normalized correlation coefficient is computed to search the images for eyes. However, these methods are sensitive to image resolution, illumination, or facial movement dynamics in real time performance.

Recently, robust facial landmark detectors have been proposed to track most of characteristic points on a human face. These detectors are trained on iBUG 300-W dataset and their precision and sturdiness are evaluated to varying illumination, various facial expressions, and head rotation [14]. Based on one of these facial landmark detectors, in this paper, we propose a simple and efficient technique to detect eye blinks: a scalar quantity, namely EAR, is derived from the eye landmark positions to reflect a level of eye openness or closeness by measuring the ratio of the horizontal and vertical distance between eyelids, and the rapid distance ratio changes in eyelids are considered as blinks [14]. Such eye blink detection algorithm is a coarse-to-fine method: firstly, we detect and locate eyes position; secondly, eye contour is estimated by 12 landmarks and therefore it can describe both open and closed eyes; thirdly, EAR is calculated and used for determining the eye states based on whether the value is above or below a certain threshold, and finally an interface is designed to test the robustness of the system and to gather data to calculate the accuracy of the system.

This paper is structured as follows, after related eye blink detection works, proposed algorithm to detect eye blinks are detailed in the 3rd section, testing results analysis is presented in 4th section, and finally conclusion is given for future work.

2. Related Work

Many researches have been devoted to this work. For face detection, Haar feature-based Cascade Classifier is used to identify the face region of the image in real time [18]. Camshift algorithm is used to combine colors represented as Hue from HSV color model for face tracking [5]. For eye detection, a cascade of classifiers based on Haar-like features is built by two training datasets: positive and negative samples [1]. Adaptive Boost, a learning algorithm, is used to build a strong classifier from weak classifier [10]. For eye tracking, Kalman filter algorithm uses finite-difference method to calculate partial derivatives of nonlinear functions [9].

For eye blink detection, many methods are applied. Some use *correlation coefficient* to measure the similarity between closed eye and open eye image: as someone closes eyes when blinking, correlation coefficient will decrease [7]. Other approaches include *optical flow* to track eyelid movement, in which method detection is based on matching SIFT (scale-invariant feature transform) descriptors computed on GPU [11]. This method uses threshold frame difference inside eye region to locate motion region, which will be used to calculate the optical flow. While users blink, eyelids move up and down so as to trigger the motion in vertical direction. Later the eyelid movements are estimated by *normal flow* and deterministic finite machine with three states – steady state, open state and close state – instead of optical flow to calculate eye blink characteristics [8]. *Variance map* specifies distribution of intensities from the mean value in a video frame sequence; the intensity of pixels located in eye region changes during blink [13]. A deformable model – *Active Shape Model* – is represented by several landmarks as the eye contour shape; the model learns the appearance around each landmark and fits it in the actual frame to obtain the eye shape [12]. Blinks are detected by the distance measurement between upper and lower eyelid. *Eyelid's Status Detecting (ESD) Value* calculation can also be used to detect blinks: it increases the threshold until the resulting image has at least one black pixel after applying median blur filtering, and the value is different when users open or close their eyes [4].

3. Eye Blink Detection Algorithm

Blinking is a natural eye motion defined as the rapid closing and opening of eyelids. The algorithm we apply consists of four steps, as shown in figure 1. The method first utilizes the facial landmark detector included in the dlib library, which is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees by Kazemi and Sullivan (2014) [9], to obtain the face bounding box. Then, eyes and eyelid contours are localized and drawn inside the bounding box in each video frame. Next, from the eye landmark coordinates detected, eye aspect ratio (EAR) is derived to estimate the eye-opening state. Since each individual has a little bit different patterns for blinks: some people will close and open their eyes in a more frequent manner; some people will squeeze their eyes harder when blinking; and some will blink in a longer duration, a certain threshold is defined and specified for each individual. This trial-and-error experiment for threshold is achieved through a user-friendly testing interface, in which the actual blinks calculated by user key inputs are compared with the blink counts calculated by the algorithm.

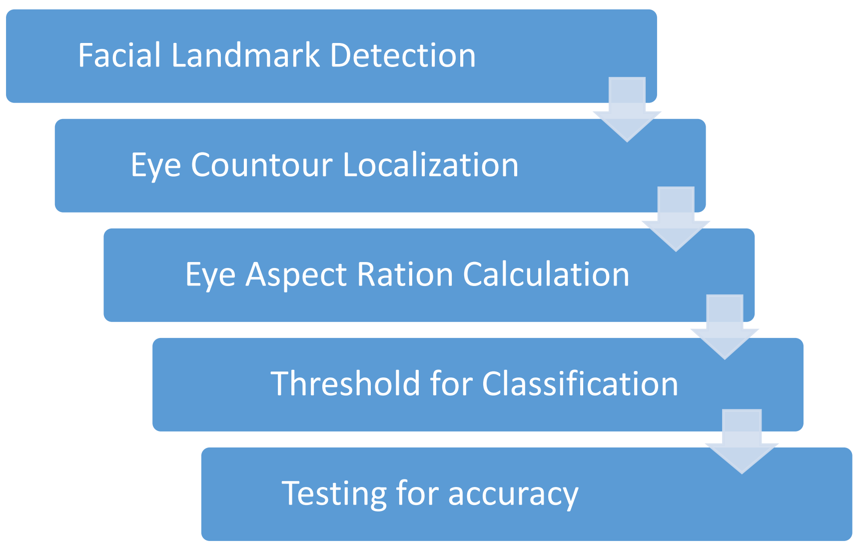

Figure 1: Algorithm Flow Diagram

3.1 Facial Landmark Detection

The pertained facial landmark detector encompassed in the ‘Dlib’ library is used to estimate the location of 68 (x, y) coordinates that map the key facial features on a human’s face. This method starts with a training set of labeled facial landmarks on an image by stating specific (x, y) coordinates of regions surrounding each facial structure, and then estimate the probability on distance between pairs of input pixels. Based on the training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the pixel intensities. The result is a facial landmark detector that can be used to do real time facial features localization with a higher level of performance [16].

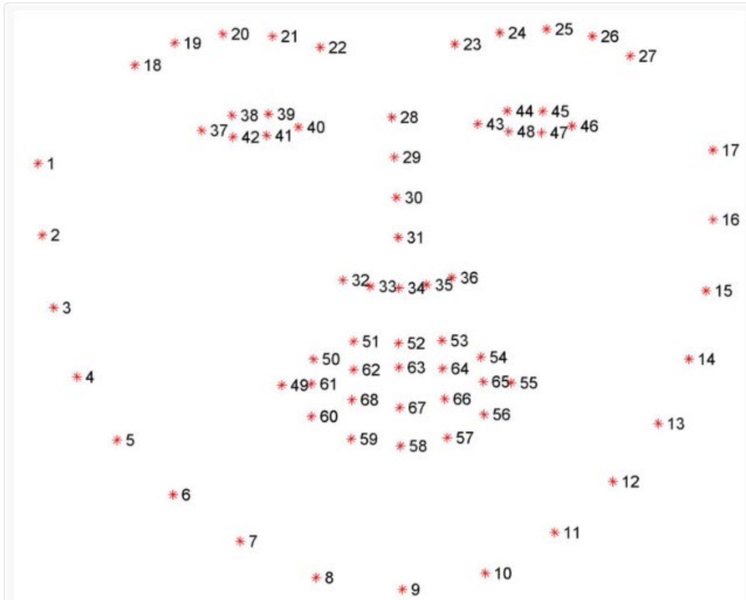

Figure 2: 68 facial landmark coordinates from iBUG 300-W dataset

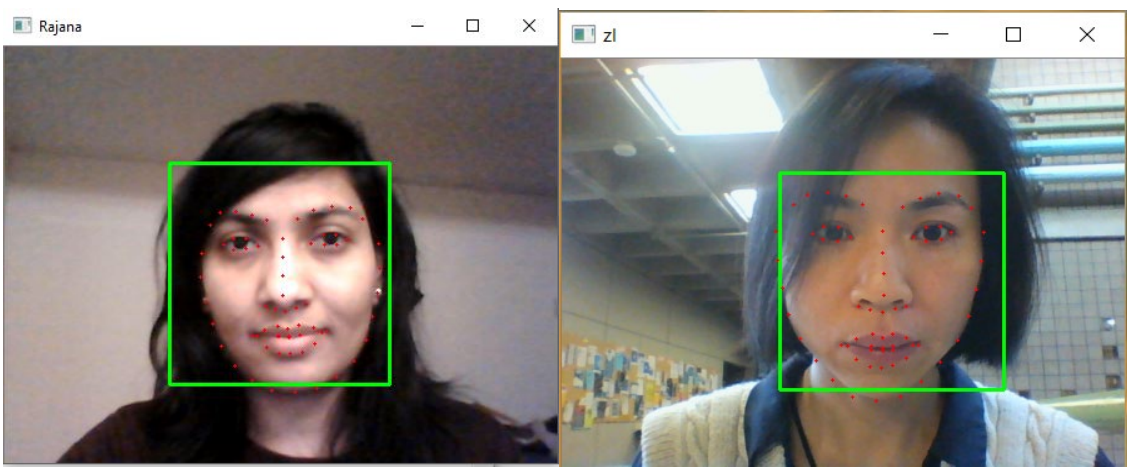

Figure 3: Facial landmark coordinates applied on human faces

3.2 Eye Contour Localization and EAR

Given the facial landmark detector included in the dlib library, 12 (x,y) coordinates are localized in each video frame and connected together to show eye contour. Each eye is represented by 6(x, y) coordinates, starting from the left-corner of the eye and going clockwise around the rest of the eye region, as shown in figure 3 [15]. As you can see, the distance between P_1 and P_4 is the width of the eye, and the height can be calculated by the vertical distance between P_2/P_3 and P_6/P_5.

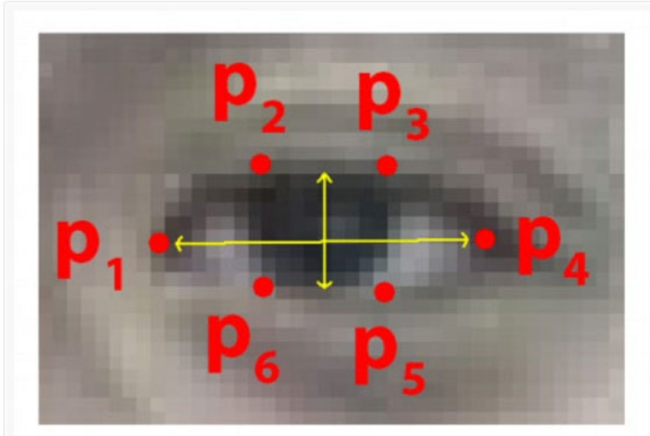

Figure 4: 6 Eye Contour landmarks

Based on the paper Real-Time Eye Blink Detection using Facial Landmarks by Tereza Soukupova and Jan Cech [17], the relation between width and height can derived by an equation called eye aspect ratio (EAR).

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 5: Eye Aspect Ratio equation

The numerator of this equation computes the distance between two pairs of vertical eye landmarks and adds them together, while the denominator computes the distance between horizontal eye landmarks and denominator is weighted by multiplying by 2, since there is only one pair of horizontal points but two pairs of vertical points. The idea behind this equation is that when eyes are open, the EAR is mostly constant but will drop dramatically close to zero when eyes are closed. Before blinks take place, the eye aspect ratio should be approximately constant and when blinks take place, the ratio will decrease dramatically and then increase again to the constant value. This can indicate that a single blink has occurred. This logic can be visualized in the figure 6. After computing EAR for both left and right eyes, we average the two eye aspect ratios to achieve a better blink estimate, based on the assumption that both eyes are blinked at the same time [15].

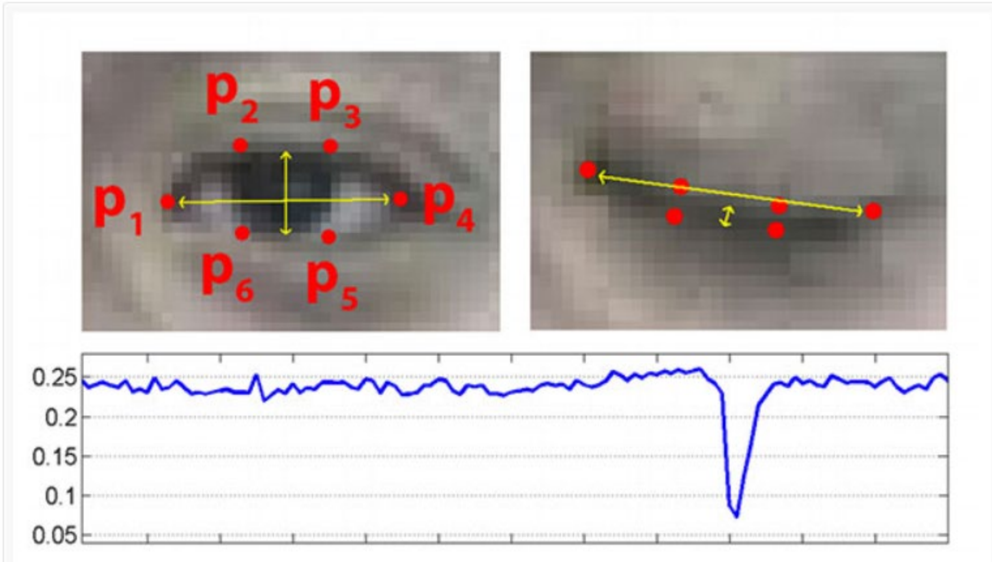

Figure 6: Visualization of eye landmarks when eye is open & closed and EAR over time

After computing the EAR, we need to determine whether a blink is happening or not. Although aspect ratio of open eye should have a small variance among individuals, each person has different pattern of blinks and different size and shape of eyes. This means a pre-defined threshold should be found for each individual: if the eye aspect ratio is below the threshold, we increase the number of conservative frames to indicate a blink has taken place [15].

3.3 User Interface

Since each individual has various eyes blink patterns in terms of blink speed or blink duration, a testing interface has been designed to gather data to measure the accuracy of the system. Before starting video streams, we ask users to input their name and gender that will be stored in a .csv file. When frame window shows, we start looping over each frame of the video stream to detect faces and eyes and draw bounding box and eye contour. EAR value for each frame will be shown on the upper-right corner of the window, as well as the default threshold value. Based on the real time EAR value, users can press keys to adjust the threshold value: press “i” key to increase the threshold and “d” key to decrease the threshold. For testing purpose, we will record 3 to 5 different threshold values for each individual.

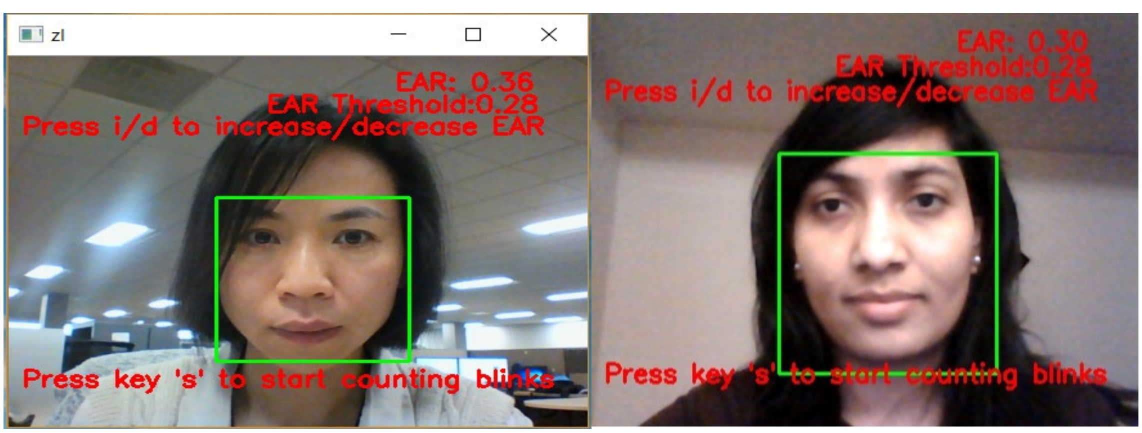

Figure 7: Interface before testing

The default value of ear threshold is defined as 0.28. To evaluate the accuracy, we will ask users to blink for 30 times and then the blink prediction calculated by the system will be divided by the actual 30 blinks to measure the accuracy percentage. The steps to get accuracy are as follows. Users will press “s” key to start the test. After “s” key is pressed, the blink count will be shown on the upper-left corner of the window. To record the actual blinks, users are asked to press “a” key when they are blinking, and the actual blink value will be shown below the blink count. A beep sound will be applied to notify users to stop blinking when 30 times of blinks has reached. The actual blink and blink count calculated by the program will also be exported to the .csv file. For analysis, the image of the users’ faces will also be captured when they press “s” key. All the data, along with the width and length of the facial bounding box, the horizontal and vertical distance of the eye and the accuracy, will be detailed in the .csv file for later analysis.

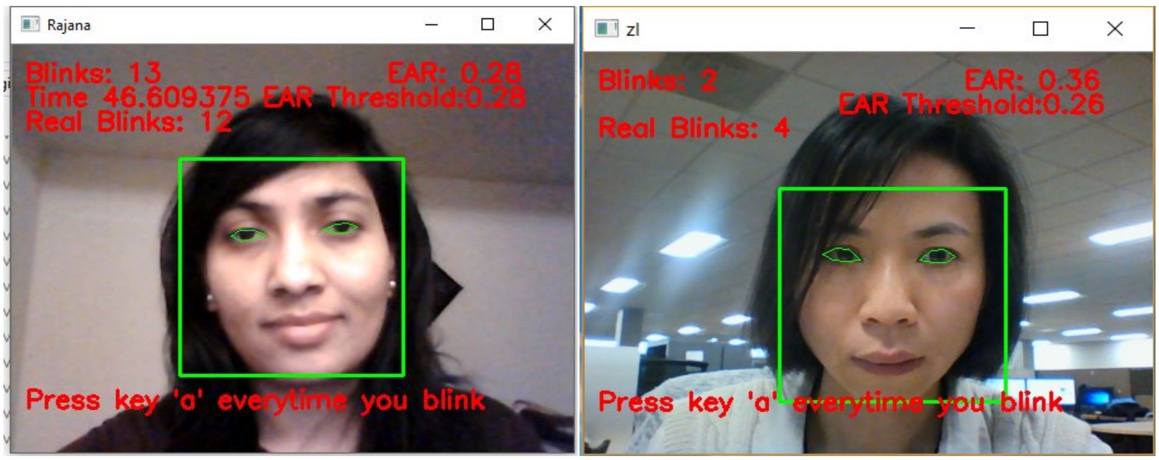

Figure 8: Interface after testing

Testing is conducted on a dataset consisting of 90 entries with different thresholds for every individual for both male and female with various facial expressions, such as glasses wearing. Each entry in the csv file has a relevant photo for the user captured by the program and saved as .jpeg file. When testing, they are asked to look straight into the camera at close distance, at a static position, not to either smile or half blink. Based the most suitable threshold for each individual, we aim to calculate the average accuracy of the system and also based on the measurements of each subject’s eye width and height as well as distance between two eyes and also the features on their faces, we want to detect driver’s conditions while they are driving in the future work.

However, during the testing, we have encountered some technical issues. We use keys to get the actual blinks; however, keys responsiveness is based on systems, which is to say when subjects press the key every time when they blink, the program cannot detect the keypress and therefore doesn’t increase the actual blink counter. In this case, we have to retest again or count manually and then change the actual_blink number in the .csv file.

4. Eye Blink Data Analysis

All the test results are exported to a .csv file. The format of the .csv file is shown as in figure 9. The dataset is a matrix of consisting of 100 rows about each test case, and 18 columns specifying the features of the test cases. The first two columns are the basic information about testing users, name and gender, followed by the EAR thresholds. For every user, there are 3 to 5 different threshold values. The following columns are the blinks counted by the system, actual blinks input by user key press, and the accuracy calculated by dividing the blinks count by actual blinks. The following columns provide the measurements about the face and eyes. These includes the face width and height, the horizontal distance between eyelids of left and right eyes, and two vertical distances between eyelids of left and right eyes, followed by four distances between left and right eye inner and outer corners, as shown in figure 10. The last four measurements will be affected by the distance between the users and computer screen. The closer the users are, the larger the measurements. Since we don't specify the exact distance at which the users are asked to sit from the computer screen, even for the same person, the distances between their left and right eye corners are not exactly the same, but somewhat close. Based on these dimensions, we can assume the approximate size of eye and face for each person and make some interesting observations.

22	# name	gender	ear_threshold	blinks_count	actual_blinks	face_width	face_height	accuracy	LefteyeV1	LefteyeV2	LefteyeH	RighteyeV1	RighteyeV2	RighteyeH	Distance1	Distance2	Distance3	Distance4
23	gfb	f	0.28	29	30	179	179	96.66666667	9.055385138	11.18033989	32.14031736	10	11	31				
24	bjbhknk	m	0.28	9	3	149	149	33.33333333	5	5.099019514	22.09072203	5.099019514	5	22.09072203				
25	iygjhg	f	0.28	6	3	179	179	50	10.04987562	11.18033989	33.37663854	11.04536102	10	31	111.2879149			
26	fwgf	f	0.28	7	3	215	215	42.85714286	11.04536102	12.04159458	37	12.04159458	11	37.01351105	96.00520819			
27	uitywyt	f	0.28	7	6	149	149	116.6666667	7	7.071067812	24	7	7	23	59	60	36	83
28	Sivasanka	Male	0.29	22	30	104	104	73.33333333	5.099019514	5.099019514	17.11724277	5	6.08276253	19.02629759	44.10215414	42.19004622	25.07987241	61.20457499
29	Sivasanka	Male	0.3	33	30	104	104	90.90909091	5	6.08276253	16.2788206	5	6	18.02775638	44.10215414	42.29657197	26.07680962	60.29925373
30	Sivasanka	Male	0.31	30	30	104	103	100	5.099019514	5.099019514	15.29705854	5	5.099019514	19.02629759	43.18564576	39.45883931	24.18677324	58.42088668
31	padminirc	female	0.28	10	30	124	124	33.33333333	8.246211251	8.062257748	25.17935662	8.062257748	8.062257748	28.16025568	65.03076195	62.12889827	37.01351105	90.00555538
32	padminirc	female	0.3	5	30	150	149	16.66666667	9	8.062257748	20.09975124	8.246211251	9.219544457	25.70992026	59.22837158	53.60037313	33.54101966	79.24645102
33	padminirc	female	0.31	9	30	124	125	30	8	8	24	8.062257748	8.062257748	27.16615541	64.28063472	61.07372594	37.12142239	88.20430828
34	padminirc	female	0.3	9	30	150	150	30	9.055385138	9.055385138	25.07987241	9.055385138	9.055385138	28.16025568	66.12110102	63.00793601	38.01315562	91.02197537
35	padminirc	female	0.28	13	30	149	149	43.33333333	9.055385138	9.055385138	26.07680962	9.219544457	9.219544457	27.16615541	65.12296062	64.00781202	38.01315562	91.02197537
36	Yuvaraj	Male	0.26	21	30	179	180	70	8	8	29.01723626	8	8	29	71.0070419	71.02816343	42.01190308	100.019998
37	Yuvaraj	Male	0.25	4	30	149	150	13.33333333	9	9	31.06444913	9.055385138	8	31.01612484	74.00675645	74.02702209	43	105.0047618
38	Yuvaraj	Male	0.27	18	30	179	180	60	8	8.062257748	32.01562119	8	9.055385138	30	74.00675645	76.02631123	44.01136217	106.0188662
39	Yuvaraj	Male	0.24	16	30	179	180	53.33333333	8.062257748	9.055385138	32.06243908	8	8	31.01612484	75.00666637	76.02631123	44	107.0420478
40	Kalaiarasi	Female	0.28	27	30	124	125	90	4.123105626	4	22.09072203	4	4.123105626	23.19482701	55.32630477	54.23098745	32.14031736	77.41446893
41	Kalaiarasi	Female	0.27	25	30	125	124	83.33333333	7	6	21.02379604	6	7.071067812	22.09072203	53.15072906	52.08646657	31.06444913	74.16872656
42	Kalaiarasi	Female	0.26	21	30	125	124	70	3.16227766	3.16227766	21.09502311	3	2	21.02379604	51.2445119	51.35172831	30.2654919	72.33947747
43	Kalaiarasi	Female	0.25	17	30	124	124	56.66666667	3	3	19.10497317	3	3	20.09975124	50.15974482	49.16299421	30.06659276	69.26037828
44	nithya	female	0.24	20	30	124	124	66.66666667	4.123105626	4.123105626	18.11077028	4	4	18	43.0464865	43.18564576	25.07987241	61.13100686
45	nithya	female	0.26	29	30	124	124	96.66666667	8.062257748	7.071067812	17.02938637	7.071067812	7.071067812	15.13274595	35.12833614	37	20.02498439	52.03844733
46	Vimalesh	Male	0.26	43	30	124	124	69.76744186	4.123105626	3.16227766	18.24828759	3.16227766	3.16227766	19.02629759	46.17358552	45.39823785	27.16615541	64.38167441
47	Vimalesh	Male	0.24	19	30	149	150	63.33333333	7.071067812	7	21	6.08276253	6	21.09502311	51.35172831	51.15662225	30.2654919	72.24956747
48	Vimalesh	Male	0.26	1	5	124	124	20	5	6	21	5	5.099019514	21.02379604	49.04079934	49.01020302	28.01785145	70.0285656
49	Vimalesh	Male	0.27	21	30	150	150	70	5	5.099019514	20	5	5	20	47.01063709	47.01063709	27.01851217	67.00746227
50	jaishwini	female	0.25	19	-1	104	104	-5.263157895	5	4	20	4	5	18	45	47	27	65
51	Manimara	Male	0.28	79	30	124	125	37.97468354	6	6	21.02379604	6	6	22.02271555	54	53.03772242	32.01562119	75.00666637
52	Manimara	Male	0.25	19	30	124	124	63.33333333	5.099019514	5.099019514	20.09975124	5	5.099019514	20.02498439	52.08646657	52.15361924	32.06243908	72.1734023
53	Manimara	Male	0.26	2	-1	103	104	-50	4.123105626	3	18.11077028	3	4	18.11077028	45.39823785	45.39823785	27.29468813	63.50590524
54	Manimara	Male	0.26	39	30	103	104	76.92307692	5.099019514	4.123105626	17.11724277	4	4.123105626	18.11077028	44.40720662	43.41658669	26.30589288	61.52235366
55	Manimara	Male	0.27	9	30	150	149	30	7	8.062257748	27.07397274	7.071067812	8	26	66.00757532	67.06713055	40.01249805	93.04837452
56	Manimara	Male	0.26	29	30	179	179	96.66666667	8.062257748	9.055385138	30.2654919	9	9	29.01723626	68.11754546	69.35416354	39.11521443	98.32598843
57	Manimara	Male	0.26	30	30	150	149	100	7.071067812	7.071067812	26.07680962	7	7	26.01922366	65.06919394	65.12296062	39.05124838	91.13725912
58	Rajana	Female	0.28	33	30	179	179	90.90909091	4.123105626	4.123105626	29.15475947	4.123105626	4.123105626	28.01785145	70.17834424	71.34423593	42.19004622	99.32270637

Figure 9: Output csv file

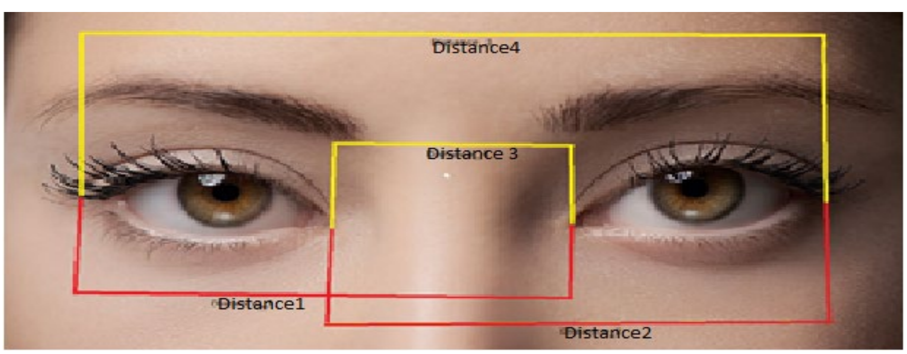

Figure 10: 4 Distances between left & right eyes

To measure the accuracy of the system, we select the test cases with highest accuracy rate for each user, add them together and divided by the total number of users. The average accuracy is 85% (84.64%). There are 6 users including us who had the maximum accuracy of 95%. Very few users had maximum accuracy as 46%. Explicitly, people with big and wide eyes are more likely to achieve a higher accuracy as high as above 90% while people with narrow and small eyes are observed to have relative lower accuracy, an average of around 50%. The EAR thresholds with which every user achieved their highest blink accuracy ranges from 0.24 to 0.31. The farther the person is from the system camera the lower the ear and the nearer the user is to the camera the higher is the ear for the same user. The standard EAR threshold for both of us was 0.27/0.28. Opposed to what we have expected, wearing eye glasses don't have much of an effect on the performance of the system. That is to say, people with eye glasses can also get high blink accuracy as long as the right EAR is defined.

5. Technical Issues

However, the algorithm used does not attain higher precision under some scenarios. For example, it cannot detect involuntary short & fast blinks or counts only once when several blinks occur, or counts several blinks when users blink only once or don't blink, or counts when users just try to open their eyes wider. This may be due to noise in a video stream, or random fluctuations of eye landmarks, or false detection for eyes with deep double-fold eyelids. To put simply, a simple threshold on the EAR could produce few false-positive detection, where a blink is reported while in reality the person doesn't blink. The key press is non responsive on certain system, is system specific and not universal. Camera Latency and system processor affect the accuracy and robustness of the algorithm too.

6. Conclusions

An efficient method for detection of eye blinks in live video frames has been presented in this paper. Using facial landmark detector, eye region features are automatically localized and the EAR, the ratio between vertical and horizontal distance between eyelids, is calculated to estimate the eyes' open and closed states, therefore indicating blinks. The performance of the algorithm is evaluated by a testing interface where threshold can be adjusted by each individual to find the best value for them. The EAR thresholds with which every user achieved their highest blink accuracy ranges from 0.24 to 0.31. The average accuracy of the interface was around 85% across 20 users. Fast core and higher quality pixel web camera can increase the accuracy and should capture each frame without much latency. Data collected under normal scenarios are used to evaluate the blink detection accuracy, and benchmark data can be used for future training. The data shows that the proposed algorithm is able to accurately track eye locations, detect normal long blinks.

7. Future Work

In the future work, we intend to log the EAR values before and after the frame, when the most closed state for eye is detected and to train the output of the data to find a universal EAR, therefore increasing the accuracy of the system irrespective of the shape or size of the eyes. We will also try to test the algorithm in more varied environments to test the system's sensitivity. Based on the facial landmark features, we will try to detect and classify the conditions of the drivers while they are driving, such as drowsiness/sleepiness state, angry or excited emotional states, and evolve the system into detection and alert notification system for various drivers' conditions. The interface can be advanced into Real Time Emotion Adaptive Driving system [20] to monitor drivers' senses and change the environmental factors like increasing/decreasing temperature or changing the songs among the many other features with advanced settings for better driving experience.

Reference

[1] Adolf, F., “How-to build a cascade of boosted classifiers based on Haar-like features,” OpenCV’s Rapid Object Detection, 2003.
[2] Al-Gawwam, S., Benaissa, M. Robust Eye Blink Detection Based on Eye Landmarks and Savitzky-Golay Filtering, 2018.
[3] Al-Gawwam, S., Benaissa, M. Eye Blink Detection Using Facial Features Tracker,
[4] Ayudhaya, C., Srinark, T. A method for a real time eye blink detection and its applications. The 6th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2009.
[5] Bradski, G. R., “Computer Video Face Tracking for Use in a Perceptual User Interface,” Intel Technology J., Q. 2, 1998.
[6] Devahasdin, C., Ayudhya, N., Srinark, T. A Method for Real-Time Eye Blink Detection and Its Application, 2018.
[7] Grauman, K., Betke, M., Gips, J., Bradski, G.: Communication via Eye Blinks - Detection and Duration Analysis in Real Time. Computer Vision and Pattern Recognition, 2001.
[8] Heishman, R., Duric, Z.: Using Image Flow to Detect Eye Blinks in Color Videos. Applications of Computer Vision, 2007.
[9] Kazemi, V., & Sullivan, J. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1867-1874.
[10] Kuo, P. and Hannah, J., “An Improved Eye Feature Extraction Algorithm Based on Deformable Templates,” Proc. of the IEEE Int. Conf. on Image Processing (ICIP), Genoa, Italy, September 11-14, 2005.
[11] Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N., Laurendeau, D. Real-time Eye Blink Detection with GPU-based SIFT Tracking. Proceedings of the Fourth Canadian Conference on Computer and Robot Vision, 2007.
[12] Liting, W., Xiaoqing, D., Changsong, L., Wang, K.: Eye Blink Detection Based on Eye Contour extraction. Image Processing: Algorithms and Systems, SPIE Electronics Imaging, 2009.
[13] Morris, T., Blenkhorn, P., Zaidi, F.: Blink Detection for Real-time Eye Tracking. J. Netw. Comput, 2002.
[14] Pimplaskar, D., Nagmode, M., Borkar, A. Real Time Eye Blinking Detection and Tracking Using Opencv. 3. 1780-1787, 2013.
[15] Rosebrock, A. Eye Blink Detection with OpenCV, Python, and Dlib, 2017. https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/
[16] Rosebrock, A. Facial Landmarks with Dlib, OpenCV, and Python, 2017. https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
[17] Soukupová, T., & Cech, J. Real-Time Eye Blink Detection using Facial Landmarks, 2016.
[18] Viola, P. and Jones, M., “Rapid Object Detection using a Boosted Cascade of Simple Features,” Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR), Hawaii, USA, December 9-14, 2001, Vol. 1.
[19] Zhang, J. and Zhang, Z., “Application of a Strong Tracking Finite-Difference Extended Kalman Filter to Eye Tracking,” Proc. of the Int. Conf. on Intelligent Computing (ICIC), Kunming, China, August 16-19, 2006.
[20] https://www.autonomousvehicleinternational.com/news/adas/kia-to-unveil-future-autonomous-car-that-can-read-a-passengers-emotions.html