International Journal of Hybrid Information Technology
Vol.9, No.3 (2016), pp. 341-346
http://dx.doi.org/10.14257/ijhit.2016.9.3.32

Real Time System for Student Fatigue Detection during Online Learning

*Kinjal V Joshi[1], Azim Kangda[2] and Sunny Patel[3]

[1]Department of Computer Engineering,
GTU, Gujarat, India
[2]Department of Computer Engineering,
GTU, Gujarat, India
[3]Department of Computer Engineering,
GTU, Gujarat, India
[1]kinjalvjoshi@gmail.com, [2]azim.u.kangda@gmail.com,
[3]iamsunnympatel@gmail.com

Abstract

At the present learning through online video is very popular. But there is no way to determine whether student is actually watching video or not. In this paper, an algorithm for real time eye state classification using simple web-cam is presented. Here one application is developed in which eyes of the person seated in-front of camera are detected using classifier. Four different eye positions: looking straight, looking left, looking upward and looking right are classified with the help of K-means clustering of the features of detected eyes. Here looking downward is not considered because it seems closed eyes and when closed eyes are detected the video will automatically pause. This approach is also used to detect constant gaze towards screen to prevent Computer Vision Syndrome. Another application of eye detection and eye state classification is to detect driver fatigue during driving. The experimental results prove the effectiveness of the presented methods.Given that two eyes are detected in a face; the system classifies the eye-states with an accuracy of 95%.

Keywords: *Eye detection, Eye tracking, Eye state classification, K-means clustering*

1. Introduction

Online video education is steadily becoming more important. Reasons are not only geographical differences between the teacher and the student, but also new teaching concepts such as “flipped classroom". In the U.S.A. already 60% of the 50 Million students use online education. Nonetheless, many concerns still exist towards this new teaching system: 1) Parents and teachers do not know whether or not the child really watches the video lectures or if it just lets the video run in the background. 2) The student himself or herself gets often tired or distracted while watching a video, which makes him or her to miss certain parts of the video.

In this paper, a new application is introduced that might help to solve the mentioned concerns. Here video will be automatically paused in these types of situations.

2. Related Work

This section analyses previous work on face and eye detection, eye tracking and eye state classification area. Viola and Jones [1] have used boosted cascade of features to detect particular object. Template matching and support vector machine based approach is

ISSN: 1738-9968 IJHIT

used to detect eye from face image [2] [3]. Eye-tracking is an area of computer vision that has been researched in the past years. The eye-tracking systems can be used for several applications. Some of them have already been implemented to conduct psychological studies about attention and interest [4]. [5] Describes the most important application of eye tracking i.e. driver drowsiness detection. In [6] floatboost learning for classification is described. In [7] concept of Clustering and regression is used for eye state classification.

3. Implementation

3.1. Face and Eye Detection

Here the application which accesses the computer's webcam in real-time is developed using C++ OpenCV. Here frames are separated from video and cvHaarDetectObjects are used to detect faces from each frame. The inbuilt cascades of face and eye are used to detect them, i.e. "haarcascade_frontalface_alt.xml" and "haarcascade_eye_tree_eyeglasses.xml" respectively. Figure 1 shows the result of face and eye tracking.

Figure 1. Real Time Face and Eye Tracking via Webcam

3.2. Eye Cropping

After successfully detection of face and eye, now we require to crop an eye from detected image to do eye state classification. OpenCV provides a function called Mat::crop() to crop image, using which we can crop eyes. Syntax of crop function: Mat:: crop(Frame, Rect(r->x,r->y,r->width,r->height)); Here arguments are:

1. Frame : It indicates the image or video frame that we want to crop.
2. Rect : It will specify the portion of image that is to be cropped.

Figure 2 represents cropped eye.

Figure 2. Cropped Eye

3.3. Eye State Classification

Here four different positions of eyes i.e. looking straight, left, upward, and right are used for K-means classification. Here "looking downward" label is omitted because human eyes naturally seem close when looking down so they are detected as closed eyes. K means clustering is not used for this case because in this application when eyes are detected close video will pause.

The features of the cropped open eye images are extracted by finding percentage of darker versus lighter pixels in each cell of a 3x3 grid of the eye. To extract these features, a given image's pixels are represented in three-dimensional (RGB) color space and the K-means clustering algorithm with K = 3 is used to filter the image into three distinct colors.

Here functionality of K –means algorithm is described [8]. K-mean is an algorithm to categorize the objects based on attributes/features into K-number of groups. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

K-means function can be defined as follows [9]. Given a set of observations ($\mathbf{x}1$, $\mathbf{x}2$, …, $\mathbf{x}n$), where each observation is a *d*-dimensional real vector, *K*- means clustering aims to partition the *n* observations into k ($\leq n$) sets $\mathbf{S} = \{S1, S2, \ldots, Sk\}$ so as to minimize the within-cluster sum of squares (WCSS). In other words, its objective is to find:

$$\arg_{s} \min \sum_{i=1}^{k} \sum_{x \varepsilon S_i} ||x - \mu_i||^2 \quad (1)$$

where μ_i is the mean of points in Si.

Thus, the purpose of K-means clustering is to categorize the data. The user sets the desired number of clusters and then K-means rapidly finds a good location for those cluster centres, where “good” means that the cluster centres to be located in the middle of the natural clumps of data.

Given the nature of the eye, these three distinct colors approximately represent the iris, the skin around the eye, and the sclera in order of darkest to lightest. To remove the skin from consideration, the second cluster is ignored. Each filtered pixel is then placed in its original location within the image.

After dividing the image into a 3x3 grid, the percentages of cluster 1 out of cluster 1 and 3 pixels are calculated in each region to form the set of features. Figure 3 represents the result of K-means clustering. Here pixels those represent iris are shown with red color. From filtered pixels the centre of image is found and then red pixels are calculated. From the calculation of both (centre and red pixel), it is decided that in which direction person is watching and if eye states are not in proper condition(i.e. eyes are closed or person is looking outside of window) then video will be automatically paused.

Figure 3. K-means Clustering

4. Experimental Result

This work is performed over 70 videos. Faces are detected in almost all the images. Figure 4 shows results for true face and eyes detection. For the faces having spectacles or tiny eyes are not detected properly or only one eye is detected. Eyes are also not detected properly if there is no proper lighting condition. Figure 5 represents partially true detection of face and eye in which face is detected perfectly but instead of both eyes single eye is detected or any eye is not detected.

Figure 4. True Detection of Face and Eyes

Figure 5. Partial True Detection of Face and Eyes

For 80% of the frames, face and two eyes are detected correctly i.e. 56 frames. Eye state classification is done perfectly for 53 frames. So overall result for eye tracking and eye state classification is 75.71%. If eye detection and eye tracking is done more accurately then accuracy can be improved because after eye tracking, eye state classification gives correct result for almost all cases.

Figure 6 shows results of K-means clustering for four different direction of iris.

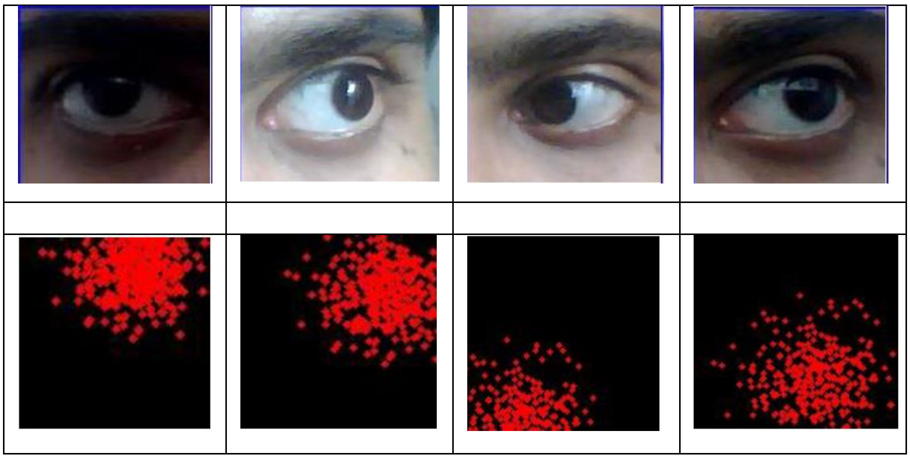

Figure 6. K-means Clustering Results

Figure 7 shows result of our GUI application that shows the user is watching in which direction. Here the message related to iris position is displayed for every frame of input video. The person seating in front of web cam changes iris position by looking in different direction and according to that message is displayed on to the screen. If iris position is not proper then video will be automatically paused.

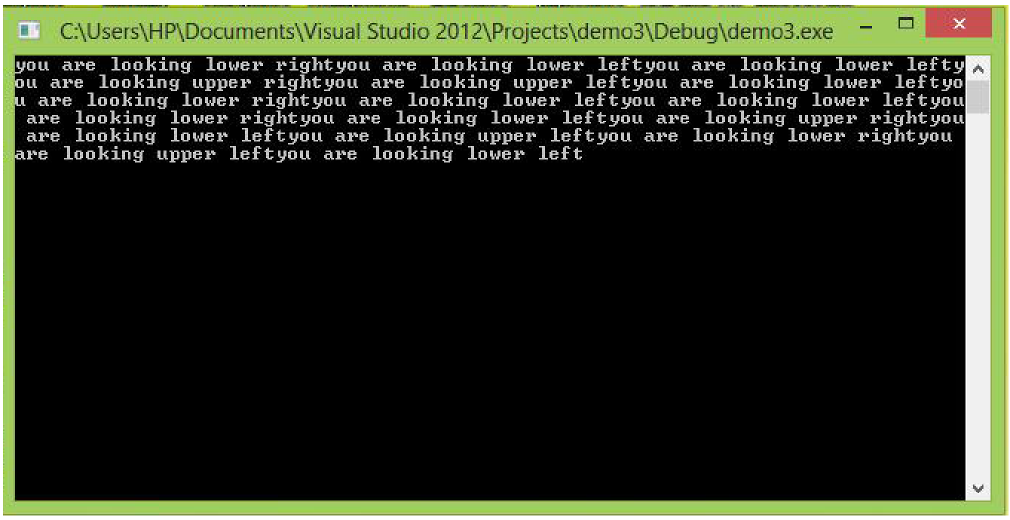

Figure 7. GUI Representation Regarding Eye State Classification

5. Conclusion

Here the developed application accesses the webcam in real-time processing. Face and eyes are detected from each frame using classifier. Features from the eyes are extracted based on color. Then the features of each eye are given as input into the K- means functions. Face & eyes tracking gives 80% true results. False detections are because of spectacles or poor lighting condition. Here overall result for eye tracking and eye state classification is 75.71%. If detection of face & eyes are done correctly then classification gives correct result for 95% case. i.e. From 70 video frames, in 56 frames eyes and face detection is done properly and from 56 frames 53 frames gives correct result for classification.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **(2001)**.
[2] Q. Wang, "Eye location in face images for driver fatigue monitoring", Proceedings of the 6th International Conference on ITS Telecommunications, **(2006)**.
[3] Kinjal Joshi, "Automatic Human Eye Localization and Its Validation for 2D Facial Image" Journal of BIOINFO Computer Engineering, vol.3 no.1, **(2014)**, pp. 60-62
[4] O. Oyekoya and F. Stentiford, "Exploring human eye behavior using a model of visual attention", Proceedings of International Conference on Pattern Recognition, **(2004)**.
[5] Y. Du, Peijun Ma, X. Su and Y. Zhang "Driver fatigue detection based on eye state analysis ", Proceedings of 11th Joint Conference on Information Sciences, **(2008)**.
[6] S. Z. Li, ZhenQiu Zhang, Heung-Yeung Shum and HongJiang Zhang, "FloatBoost Learning for Classification", Microsoft Research Asia- http://research.microsoft.com/~szli
[7] M. Picallo, Heerad Farkhoor and Thomas von der Ohe, "iDec: Real-time eye state classification via web cam", **(2012)**.
[8] M.P.S Bhatia and Deepika Khurana "Experimental study of data clustering using k- means and modified algorithms ", International Journal of Data Mining & Knowledge Management Process (IJDKP) vol.3, no.3, **(2013)**, pp 17-30
[9] A. Panjeta and A. Kumar, "Comparative performance to changeability of measuring software components", International Journal of Advanced Research in Computer Science and Software Engineering, vol.4, no.8, **(2014)**.

Authors

Kinjal V. Joshi is working as Assistant Professor in Computer Engineering department of G H Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India since 2007. Her area of interest is Image and Video Processing.

Azim Kangda and Sunny Patel are final year Computer Engineering students at G H Patel College of Engineering and Technology, Vallabh Vidyanagar, Gujarat, India.

