Fatigue Detection For Online Classes Based on Adaboost

Qingyu Mo

Department of Intelligent Science and technology, Chongqing University of Post and telecommunication, Chongqing 400065, China
Corresponding author’s e-mail: 794225502@qq.com

Abstract. Online Classes has been widely popularized since corona virus doesn’t allow students to have face-to-face classes. However, due to comfortable environment that students have, students may get sleepy in class and cannot help with their fatigue. So students need a supervised environment to alarm them not to fall asleep. So in this article, a real-time method to detect whether the student in front of the camera is sleeping, based on the demo of students in online classes, is introduced. This article focus on detection of states of eyes of students. It uses Adaboost as the training method to detect the face of the student, uses Canny to process the picture with human face, and calculates the area of the exposed eyeball and the radius of the eye. Finally, PERCLOS is used as the criterion to determine whether the student is sleeping in class. The method in this article has high detection rate on faces and eyes, which is respectively 99.7% and 98.5% for image set that made by the author. It can be applied on online classes and shows good results in detecting the fatigue of people in practical.

1. Introduction
Online Classes has been widely popularized since corona virus doesn’t allow students to have face-to-face classes. However, due to comfortable environment, namely home, that students have, students may get sleepy in class and cannot help with their fatigue. So students need a supervised environment to alarm them not to fall asleep. So the purpose of this essay is to detect whether the student is getting sloppy in online classes. Since most platforms, like Classin, can record students’ face clearly in real time, a model for eye detection could be built to detect whether the student in a picture, which is a frame of a demo of a class, is having his eyes closed or not. Adaboost is widely used on face detection for its good performance and it can avoid overfitting. So in this essay, Adaboost is used for preliminary face detection. After the face in the picture is located, precisely locate the position of the eye and extract the contour of the eyes by Canny. Then, the area of the exposed eyeball and the radius of the eye are calculated, and the proportion of how much the eyeball is covered by eyelid is obtained. The final step is to judge if the student is sleeping based on PERCLOS.

2. Methodology

2.1. Adaboost
It’s publicized that Adaboost is widely used on face detection for its high accuracy and fast speed of processing[1], which helps with real-time process. It trains many weak learners through interactions and grant weight for each learner according to its training error. At last, a strong learner is constructed, which is the combination of the weak learners. In detail, the boosting algorithm for learning a query online could be illustrated below.
T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors.

- Given example images \((x_1, y_1), \ldots, (x_n, y_n)\) where \(y_i = 0, 1\) for negative and positive examples respectively.
- Initialize weights \(w_1, i = \frac{1}{2m}, \frac{1}{2l}\) for \(y_i = 0, 1\) respectively, where \(m\) and \(l\) are the number of negatives and positives respectively.
- For \(t = 1, \ldots, T\):
  Normalize the weights, \(w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}\) (1)
  Select the best weak classifier with respect to the weighted error
  \[ \varepsilon_t = \min_{f,p} \sum_{i=1}^{n} w_i |y_i f(x_i, p, \theta) - y_i| \] (2)

3. Define \(h_t(x) = h(x, f_t, p_t, \theta)\) where \(f_t, p_t, \theta\) are the minimizers of \(\varepsilon_t\).
4. Update the weights:
   \[ w_{t+1, i} = w_{t, i} \beta_{1 - e_i} \] (3)
   where \(e_i = 0\) if example \(x_i\) is classified correctly, \(e_i = 1\) otherwise, and \(\beta_t = \frac{1 - \varepsilon_t}{\varepsilon_t}\).
- The final strong classifier is:

\[
C(x) = \begin{cases} 
1 & \sum_{a} a \cdot h_t(x) \geq \frac{1}{T} \sum_{a} a, \\
0 & \text{otherwise}
\end{cases}
\] (4)

With Haar-like features, Classifier with Adaboost can detect human face in milliseconds.

Database of MIT is used for training, which contains 2429 images of faces and 4000 images of nonfaces. For test set, 450 images with human faces from database of California Institute of Technology is used. Classifier can detect normal face. It also shows strong robustness for faces with eyeglasses and faces with closed eyes.

Figure 1. Results of Classification.

2.2. Sliding Window

After the detector is trained, the next step is to detect the face in an image. The author uses sliding window to detect and locate the accurate area of the face. This method puts a rectangle on the image, slide it over and detect possible face in it, change its size and do it again. Two essential techniques of it are enlarging the detection window and region merging[3]. Enlarging the detection window is to enlarge the size of the rectangle by multiplying the length and width by a specific constant after one time of ergodic on the image. By now, the image has multiple rectangles containing the same human face. Region merging is to combine those rectangles into one rectangle, which fits the face best.

2.3. Image Blurring

To meet the requirement of real time, number of haar-like features should be possibly reduced while the results are, in some extent, not that influenced. So the pixels of the testing image should be reduced. To do this, the image should be blurred. For contrast, An image with human face in the size of 896*592 pixels is processed and detected about 30 times in time longer than the same image in the size of 224*148. In test of 100 images with human face and 50 images without human face taken by
the author, which could be detected accurately before blurring, 98 percent of image with human face could be detected properly, and all the images without human face could be classified correctly, which shows that this blurring is rational.

2.4. Means of Gradient
In this Essay, Matlab function “vision.CascadeObjectDetector”, is used for initial eye location. This method is robust and applicable for closed eyes in practice. After roughly locating the eye, the next step is to detect the centre of the eye. Means of Gradient is used for its high accuracy and fast process speed [3]. This method analyzes the vector field of image gradients and derive a novel formulation of the vector field characteristics. In detail, let c be a possible centre and gi the gradient vector at position xi. Then, the normalized displacement vector di should have the same orientation (except for the sign) as the gradient gi (see Fig. 2). If the vector field of gradients is used, this vector field can be exploited by computing the dot products between the normalized displacement vectors (related to a fixed centre) and the gradient vectors gi. The optimal centre c* of a circular object in an image with pixel positions xi, i \(\in\) \{1,...,N\}, is then given by

\[
c^* = \arg\max_c \left\{ \frac{1}{N} \sum_{i=1}^{N} (d_i^T g_i)^2 \right\}
\]

Moreover, due to prior knowledge that the pupil is usually dark, a weight wc, whose value is the gray value at (cx, cy) of the image, is applied so that a dark centre could be more likely to be the eye centre. So the objective function should be

\[
c^* = \arg\max_c \left\{ \frac{1}{N} \sum_{i=1}^{N} w_i (d_i^T g_i)^2 \right\}
\]

Figure 2. Artificial example with a dark circle on a light background, similar to the iris and the sclera.

2.5. Eye Stretching
To detect the centre of the eye, the eye in the image should be clear and possibly exposed so that the accuracy and the detection rate could be promised. But in practical, students with fatigue usually have their eyes really closed and expose only little proportion of the eyeball, which lead to omission of detection and misdetection. To avoid these, the author processes images with human face, whose eyes cannot be detected, by vertically stretching the eyes. The author processes undetected images of eye by tripling the length of it. By doing this, the eyes can be extracted and become easier to be detected (Fig. 3).

Figure 3. Results of localization for a eye that is almost closed before(left) and after(right) eye stretching

2.6. Calculation of PERCLOS
PERCLOS(Percentage of Eyelid Closure over the Pupil over Time) is the superior method in fatigue detection compared to other methods like ETS, EEG recordings[5]. To judge if a man in the image is sleepy or not, P80 is used. The definition of P80 criterion is that if the percentage of eyelid closure
over the pupil is more than 80, it can be said that the man is sleepy and fatigued. To calculate the percentage, area of exposed pupil and area of the whole pupil are needed. Calculating the area of whole pupil is easy, as long as the radius of pupil is acquired. Calculating the area of exposed pupil is to calculate the pixels of pupil in the eye. The author derives a novel method to calculate the percentage of eyelid closure over the pupil. That is, first, pre-process the image of the eye into grayscale image, and then binarize the image using Otsu (Fig.4). So the black pixels in the contour of eye could be the area of the exposed eye[6]. The author then analyzed the geometrical distribution of the image and construct a geometrical model (Fig.5). The upper eyelid and lower eyelid are represented respectively by two straight line AB and CD, and the circle in the middle is the pupil. d1 and d2 respectively representing the distance between center of the eye O and upper eyelid, lower eyelid. Since the position of eye centre is located using means of gradient, the radius can be determined as the horizontal distance between the centre and the contour of the pupil, which can be calculated from figure 3. However, sometimes the centre is not that accurate due to environmental interference factors, and the reflected light in the eye could interfere with the calculation. So to be more robust, the horizontal distance between the contour of pupil and the pixel on coordinate (xc, yc ± 5) is calculated, and the longest distance is taken as the radius r. For the calculation of d1 and d2, calculate the vertical distance between the contour of upper, lower eyelid and the pixel on coordinate (xc ± 10, yc), and take the mean of them as d1, d2 respectively. So the area of the whole pupil S1 and the area of exposed pupil S2 are obtained by

\[ S1 = \pi r^2 \]  
\[ S2 = \arccos\left(\frac{d1}{r}\right)r^2 + \arccos\left(\frac{d2}{r}\right)r^2 - d1\sqrt{r^2-d1^2} - d2\sqrt{r^2-d2^2} \]

Therefore, percentage of eye closure P is \( \frac{S2}{S1} \times 100\% \). If P<80%, this student is not fatigued. But if P \( \geq 80\% \), it can be judged that this man is fatigued and sleepy.

Figure 4. The eye before being processed(left) and after being binarized using Otsu, and the red cross is the centre of the eye

Figure 5. The geometrical model of the eye

The final step is to calculate the percentage of eye closure, over time. Set the number of all the frames of the demo of online classes in a period of time as M, and initial number of fatigued frames as N=0. For each frame, determine whether the man in this frame image is fatigued using algorithm above. If so, let N=N+1. At last, the total fatigued frame is acquired. But this is regardless of the blinking of the man because when blinking, the man could close his eyes and be detected as fatigued state. Set the number of blinking frames, which are misjudged as fatigued frames, as N1. A man would blink his eye for about 15 times per minute. So the time, in which the man is blinking his eyes but detected as fatigued state, is t=15(t3-t2) in one minute, according to figure 6. So N1 equals to \( \frac{M(N-N1)}{4} \), and the PERCLOS is \( \frac{N-N1}{M} \). If PERCLOS is bigger than a threshold, whose value is usually 70% or 80%, it can be defined that this man in this period of time is fatigued.
3. Suggestion
In this essay, a novel method, which is similar to template matching, is used for Calculation of percentage of eyelid closure. Template matching shows its good performance in common environment. But when the student is not sitting right towards and looking at the camera, like when he is taking a note with his head down, this method would not work well. This could be a common problem for methods using template matching. Therefore, it is more applicable for online classes where students watch the PPT on phone or pad.

4. Conclusion
All the codes of this article are written in Matlab. All the codes are ran on the author’s laptop. The CPU of the laptop is Intel(R) i7-8750H, and The test image set is made by the author. The Initial face detection shows good performance, with a process speed of about 0.08 s per image and detection rate of 99.7%, which means it can process about 12 frames of image per second. The eye centre detection in the next step has the detection rate of 98.5%. Moreover, for eyes that are almost closed completely, the eye detector could not detect it at all. So for images that contain a face but in which the eye detector could not find any eye, the state of the face could be defined as fatigue. To judge the performance of final calculation of PERCLOS, the author analyzes the demos of online classes, which he has taken. The author then finds 10 periods when he was sleepy in class and calculate the PERCLOS of them. The result shows that 8 periods of them are defined as fatigue according to P80 criterion. When 25 periods, during which the author is not sleepy, are analyzed, 2 of them are wrongly detected as fatigued state. The author then analyzes all the periods that are wrongly detected and find something in common, that the face of the author was not right towards the camera. In conclusion, this classification is rational for common situations, like the students is looking at the camera or looking at the slide on the phone.

Acknowledgements
First and formost, I want to show my gratitude to Professor Kumar. He has taught me a lot in Machine Learning and given me a lot of advise. This thesis would not be accomplished without his effort. Further, I want to thank Yang Yiyuan, who has helped me organize my thought and my teacher Eva Chen, who has helped revise this thesis.

References
[1] Cao Y, Miao Q G, Liu J C, et al. Advance and Prospects of AdaBoost Algorithm[J]. Acta Automatica Sinica, 2013, 39 (6): 745-758.
[2] Viola P A, Jones M. Robust real-time face detection[C]. international conference on computer vision, 2001, 57(2): 137-154.
[3] Long L M. Reseach on Face Detection Method and Eye Localization Method Based on Adaboost[D].University of Electronic Science and Technology of China,2008.
[4] Timm F , Barth E . ACCURATE EYE CENTRE LOCALISATION BY MEANS OF GRADIENTS[C]/ Visapp - Sixth International Conference on Computer Vision Theory & Applications. DBLP, 2011.
[5] Udo Trutschel, David Sommer, Martin Golz, David J Edwards.PERCLOS: An Alertness Measure of the Past.2017.
[6] Xu Z, He F, Hua X Q, Li J L. Research on Fatigue Driving Detection System Based on AdaBoost Algorithm[J]. Automobile Technology, 2019(05):17-21.

[7] Yan J, Kuo H, Lin Y, et al. Real-Time Driver Drowsiness Detection System Based on PERCLOS and Grayscale Image Processing[C]. International Symposium on Computer Consumer and Control, 2016: 243-246