Visual Interface Final Project: Automated Exam Proctor

Wentao Jiang (wj2223), Xiuhan Hu (xh2234)

Table of contents

I. Introduction

II. Objectives and Limitations

(1). Objectives

(2). Domain Engineering and Limitations

III. Prior Work

IV. Methods and results

(1). Face analysis steps

Face detection

Face verification

Face landmarks

(2). Gaze analysis steps

Eye extraction and resize

Gradient image generation

Pupil detection

V. Evaluation and Result

(1). Evaluation metrics

(2). Experiments and Result

VI. Teamwork

VII. References

I. Introduction

With the development of internet during recent years, the traditional face-to-face education has been extended to online education. And consequently, the exams for online courses faces the fact that we are lacking of ways of distant proctoring the process at economical cost, which may lead to plagiarism and problems with the integrity of academics. In this project, we developed an online proctoring system, which utilizes normal online webcam to detect suspicious behaviors during online exam process, and provides feedbacks to the online exam providers.

This project first identifies the problem: currently there is a lack of methods for preventing cheating in online exams with no human labor and of low costs. Then we propose a method to address this issue efficiently: to assist the protection of the academic integrity in online exams, automated proctoring software using computer vision can be implemented. It is considered to be an example of a visual interface as it gathers the visual interface elements including webcam video input, computer vision recognition of number of faces, face features and eye movements and makes judgement based on automated analysis.

In order to detect most common plagiarism behaviors of online exam takes, the major vision task can be divided into two parts:

1. Detection and recognition of faces and general face features
2. Tracking and analysis of eye movements.

During recent years, the development of deep learning and big data analytics has enabled better precision and accuracy of face recognition. Based on well trained model, the accuracy of face recognition can be higher than 99% percent[6]. Several online cloud computing resources are available for doing this job, such as Microsoft FaceSDK, FacePlusPlus. Thus building an application with face recognition functionality is not very hard.

Meanwhile, improvements in computer vision and machine learning allow fast analysis of human eye movements. Specifically, multiple methods has been developed and implemented on the tracking of pupil center and gaze. Timm F. described an approach of eye center estimation based on image gradients and maximum searching without using head device or high resolution images[4]. On the other hand, Smith B. implemented a machine learning method that extracts the pixel feature vectors from a large amount of gaze images based on which a classifier was trained[5]. In this project, we apply and implemented the techniques in both face recognition and eye tracking to complete the functions of proctoring.

II. Objectives and Limitations

(1). Objectives

We decide to design a program at least capable of:

1. Detecting if the exam taker is taking test together with someone else. The program should find more than one faces appear in the video.
2. Detecting if the exam taker has some other person take the exam for him/her. We need to recognize the face and determine if it is the right person.
3. Detecting if the test taker is gazing at the screen and not looking outside it. The program will track the direction of user's sight and alarm when it is outside the screen for a period of time.
4. Detecting if the exam taker is too far or absent from the computer .
5. The program GUI will cover the full screen of the computer and the exam taker is not allowed to view anything else than that.

Whenever the program detect an abnormal behavior of the exam taker, we will mark out that clip of video for our proctor (real person) to further inspect. The program will generate a report containing annotated video and key frames we captured during the exam.

(2). Domain Engineering and Limitations

To achieve the objectives, we set some restrictions on the problem domain and we assume:

1. The exam taker should have normal facial feature. We probably cannot track the direction of sight for a strabismus patient.
2. The exam is not in a too complicated form. The exam taker should be able to solve the problems by simply reading the questions presented on screen and respond by clicking mouse or pressing keys.
3. The video input is genuine. If the test taker use some sophisticated skills to hack the camera of the computer and change the video input into a faked one, we obviously cannot know that.

There are still some cases that a real world system probably need to deal with, but we probably will not consider for this project due to the limitation of time:

1. We cannot detect if the test taker is talking with someone outside our scope. He/She probably use a bluetooth phone and it's very subtle for a proctor to find out even in offline exam.
2. We cannot allow the test taker to do an open book exam because we suppose the person is staring at the computer. However, cheat sheet is ok since we can provide necessary information on screen.

III. Prior Work

Since online exams is relatively newly emerged, many of online proctoring services[1] simply record all data they can get from the exam taker's computer: screenshots, camera video, microphone record and so on. Then they employ many people to check those data to ensure the credibility of the exam. It is a way requires highly repeated work. Recently, ProctorTrack[2] is trying to build some automated proctoring system which is not tested widely. We believe that the topic is still an open problem worth more try.

[1] softwaresecure: http://www.softwaresecure.com/
[2] proctortrack: http://www.proctortrack.com/

IV. Methods and results

The proctor program will record the video from webcam on the exam takers laptop when the exam starts. The program will choose one frame as a sample per second. Then it starts a thread going through several steps to check whether the exam taker is cheating or not on that sampled frame. Steps are divided into face analysis steps and gaze analysis steps:

(1). Face analysis steps

There are three problems concerning face detection need to be solved for building the proctor system: Face detection, face verification and facial landmark localization. They are all problems really hard to be solved from scratch, but FacePlusPlus provides APIs to handle them. In this part, the generally methods used by FacePlusPlus are described and the logic of the proctor program is introduced.

Face detection

Face detection is used to check if there are exactly one person taking the exam. Haar Feature-based Cascade Classifiers can be used to detect face from a picture.

If there is no face detected in the picture [Fig 1], the program will raise an ABSENCE error.

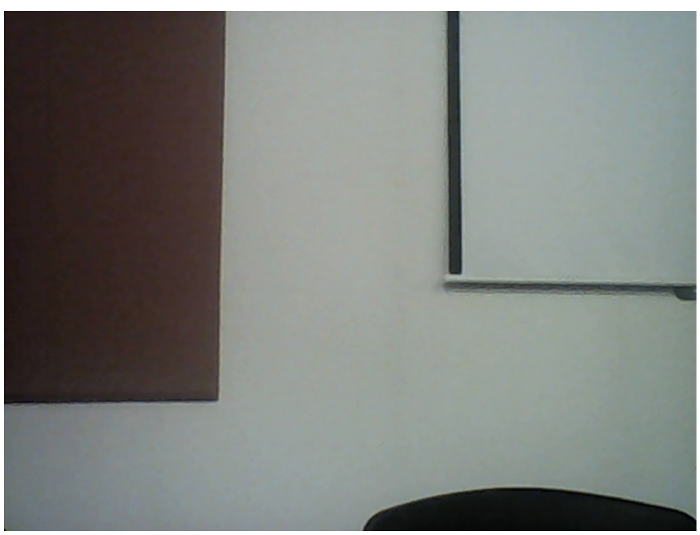


Fig 1. No face detected in the image

If there are more than one faces detected in the picture [Fig 2], the program concludes that multiple people are taking the exam.

Fig 2. Multiple faces are detected

If there only one face detected in the picture [Fig 3], the program will calculate the size of the face to determine if the user is too far from the computer:

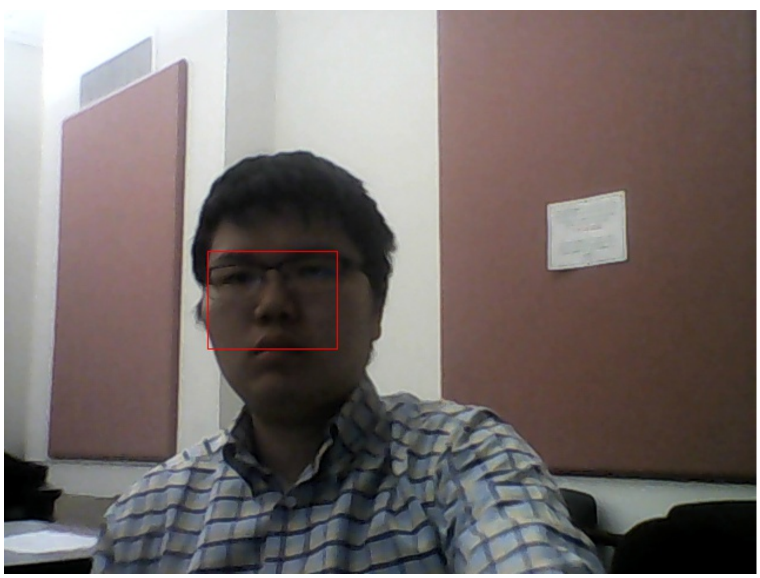


Fig 3. Face too far from the screen

After going through above checks, the proctor will proceed to next phase.

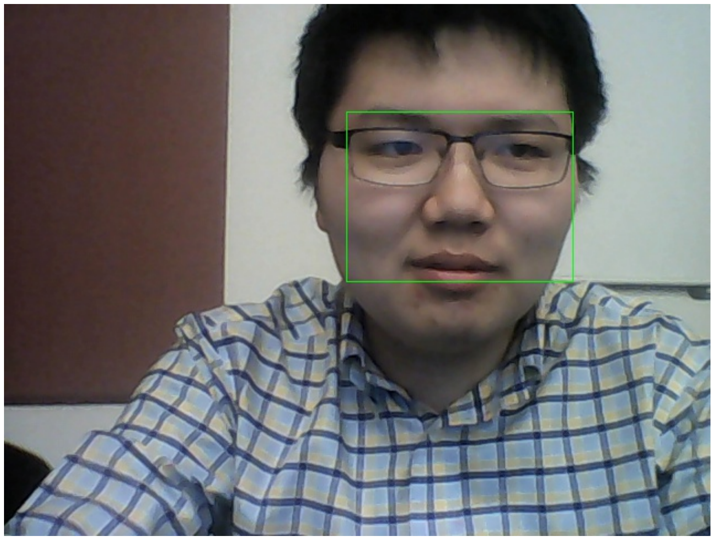


Fig 4. No face detected in the image

Face verification

Face verification is used to verify the identity of the person in front of the screen and check if someone else is taking the exam for the exam taker.

FacePlusPlus implements Pyramid CNN to learning the representation of faces. The network is trained layerwise in a greedy way. This method achieved 99.5% accuracy for LFW dataset.

We set the first several frames of video as training data to get a face model, then the program can get whether the face is mismatch with the the trained model.

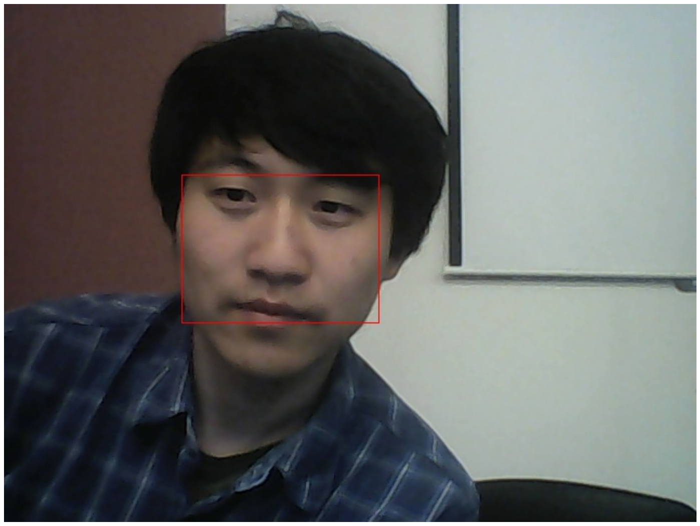


Fig 5. Wrong person detected, face mismatch.

Face landmarks

FacePlusPlus provides API for facial landmarks localization. It localize the following green dots through a Coarse-to-fine Convolutional Network Cascade. The following figure [Fig 6] shows that the face contour points are very accurate. However, the pupil center points are not very stable in many cases we tested. So in next part, we describes how to find pupils from eyes and test if the exam taker is looking off the screen.

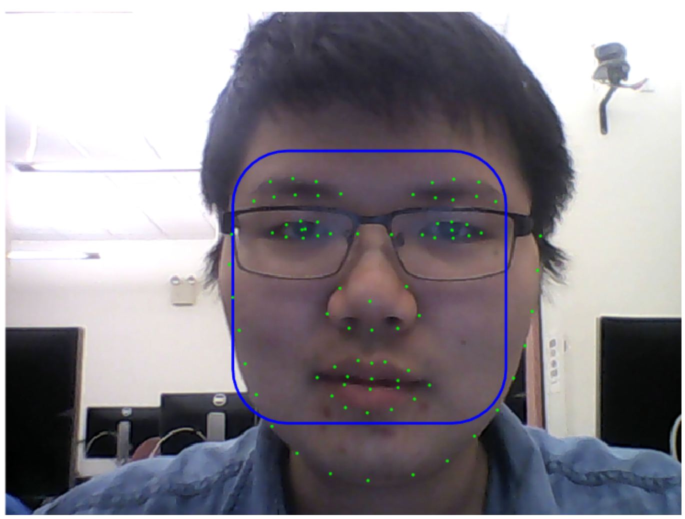


Fig 6. landmark points from face, getting from Face++ API

(2). Gaze analysis steps

The aim of gaze detection is to examine if during some certain periods the exam taker is looking outside of the screen, which is taken a potential suspicious action during the online exam.

The detection method of gaze was built partially on the features supplied by the face detection. From the first part, information including eye profile, face contour and position of nose is passed into the gaze analysis. The major steps of gaze detection are described as below. For the initial validation of the algorithm, gaze images from the following link:

http://www.cs.columbia.edu/~brian/projects/columbia_gaze.html

For the description, the following image [Fig 7] is used:

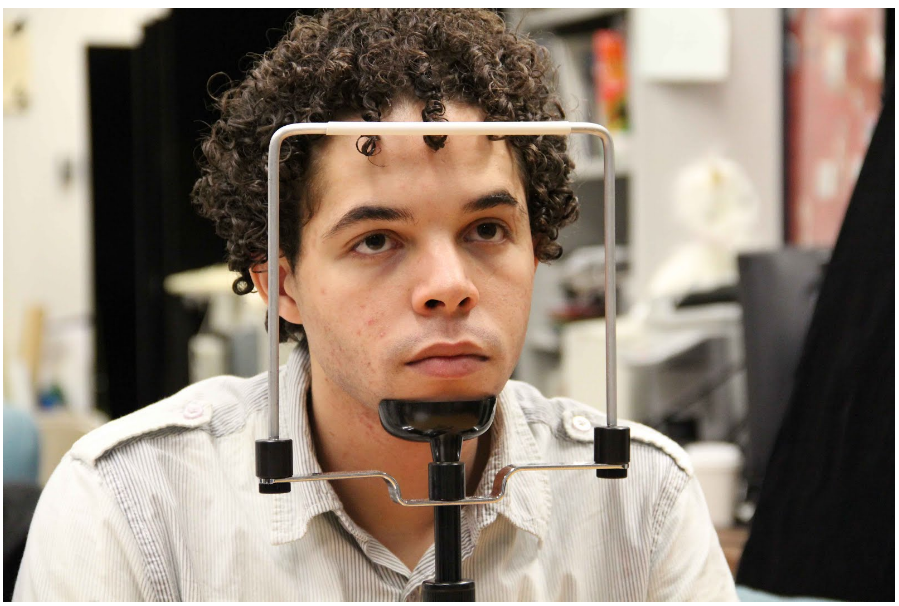


Fig 7. Gaze images used to test and validate algorithm

Eye extraction and resize

Firstly, the eye areas are extracted from face. Due to the fact that the parameters for the Hough circle search in later step is sensitive to image size, all extracted eye regions are resized to the same width (25 pixels). This is also a step used to make the system adaptive to different resolution of image and video inputs. The extracted eye areas along with resized images are shown below:

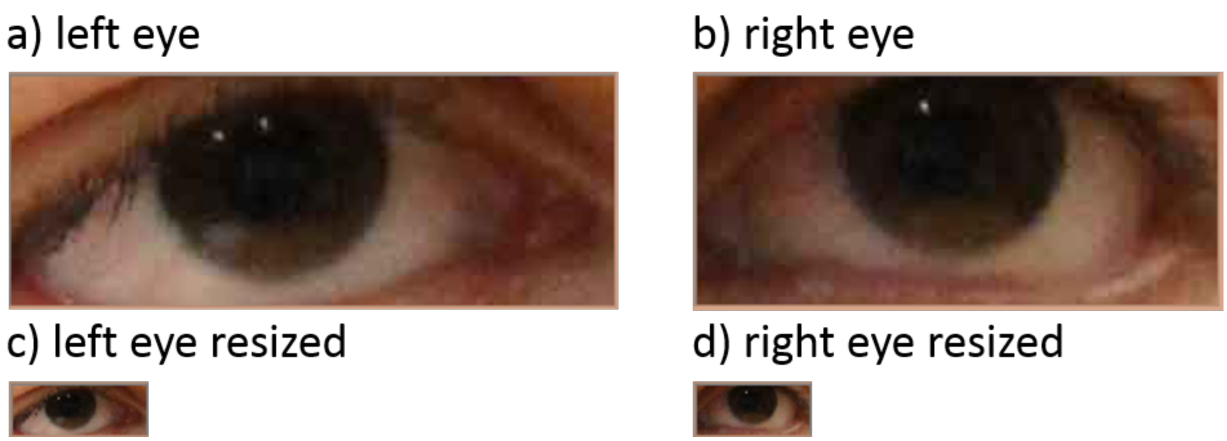


Fig 8. Eye image resized

Gradient image generation

This step is designed to generate feathers that can be used to accurately find the pupils in next step. Since the gradient from white part in eye to pupil is dramatic, it can be used as important feature for the detection of pupils.

The gradient for resized images of eyes are converted to grayscale, and then smoothed with 3x3 Gaussian filter to remove noise, after which 3x3 Sobel filter is applied to generate the gradient image. From the eye profile points generated from face recognition parts, a polygon mask is generated for each eye, and applied to the gradient image in order to remove the surrounding region around eyes which may interfere with the detection of pupil in later step. From [Fig 9] we could see that the outline of pupils has higher brightness than other region due to the fact that the gradient is relative large from white part of eyes and pupils.

Note that before we made the decision to detect the pupils with gradient image, we also tried to use color segmentation (in RGB, HSV and YCrCb modes) and Canny edge detection to distinguish the pupil from the rest part, however the results were not as good.

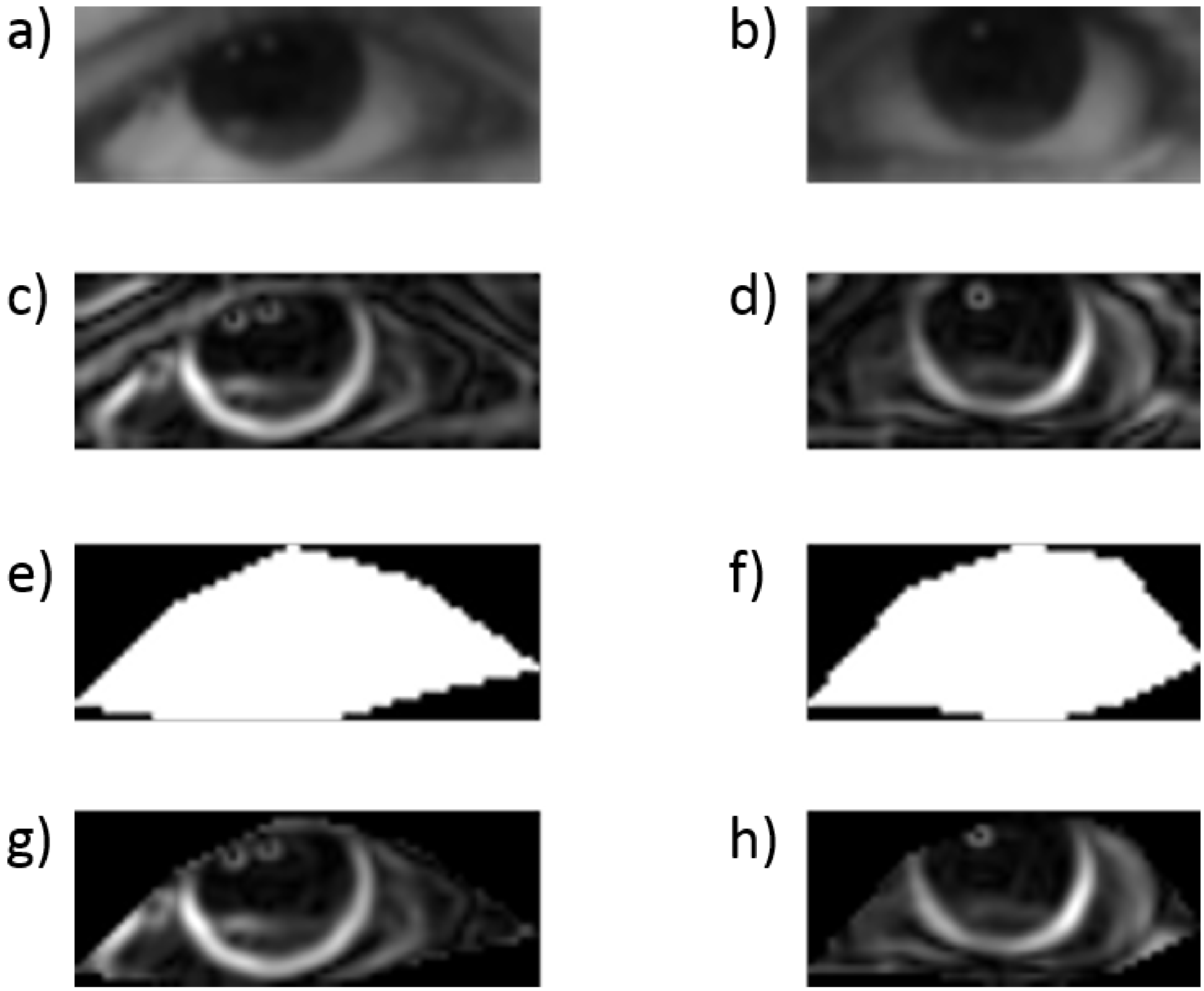


Fig. 9 left and right eyes: a) and b) in grayscale; c) and d) gradient with Sobel filter; e) and f) mask of eye profile region; g) and h) gradient image after applying mask

Pupil detection

Because the outline of pupils has been highlighted in the gradient image and are generally circular in shape, we apply Hough circle detection methods to detect the pupils and thereafter find the center of pupil.

After trying with different values, we find a set of parameters that perform relatively stable detection of the center of the pupils. The centers of the pupils in left and right eyes are calculated and transformed to coordinates in original image, and labeled as below.

Fig 10. the pupils in left and right eyes in green circles

1. Gaze angle estimation

We make estimation of the angle of gaze for both x direction (horizontal) and y direction (vertical) for the judgement that if the gaze is out of screen.

To estimate the direction of gaze relative to the webcam with considering the yaw rotation of face, two angles are calculated, the angle of pupil relative to eye center in x direction and the face yaw angle to the webcam, and the angle of gaze with respect to the webcam is the summation of these two angles. We define angles rotating to the left is positive and right negative. The larger angle the face rotates, the larger distance the center of nose is from the middle of the face contour on x axis, and according to the experiment the angle of face is generally proportional to the distance from the center of nose to the middle of the face contour. Using 106 images from the gaze dataset mentioned above, we were able to carry linear regression to fit the best coefficient in the linear relationship. In the following figure, the normalized nose deviation from the middle of face contour by dividing the face width is plotted and regressed versus the face angles labeled in the dataset with forced 0 y-intercept since the rotation of face is supposed to be symmetric. Based on the regression we set the coefficient from nose deviation to angle in degree to 100.

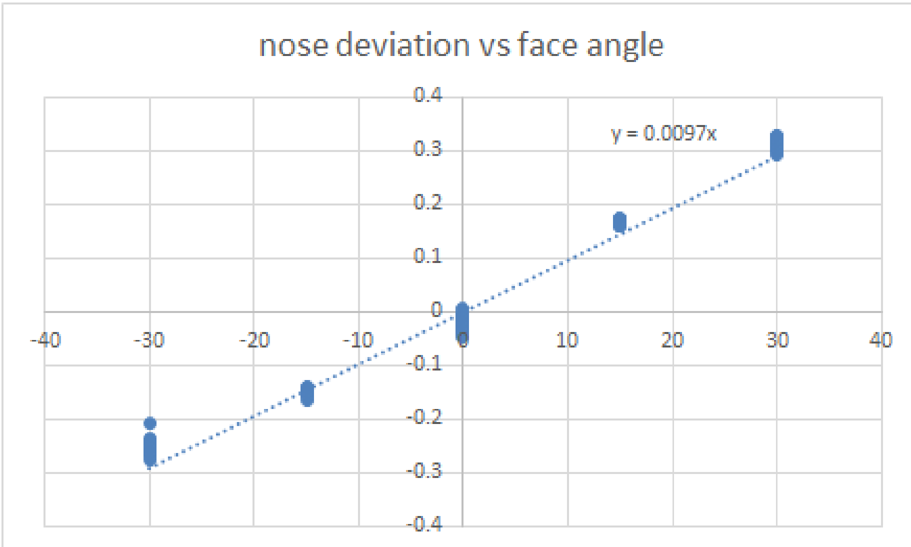


Fig 11. Nose deviation vs face angle

For the estimation of x-axis gaze direction relative to the eye, we similarly calculated the deviation of estimated center of pupil from last step to the center of eye calculated from face recognition. We also run a linear regression to get the coefficient from normalized distance to angle with the plot shown below. Note that the deviation of pupil center from the eye center is calculated as the sum of deviations for left and right eyes. And the gaze direction in respect to webcam is set to the summation of face angle and horizontal eye angle.

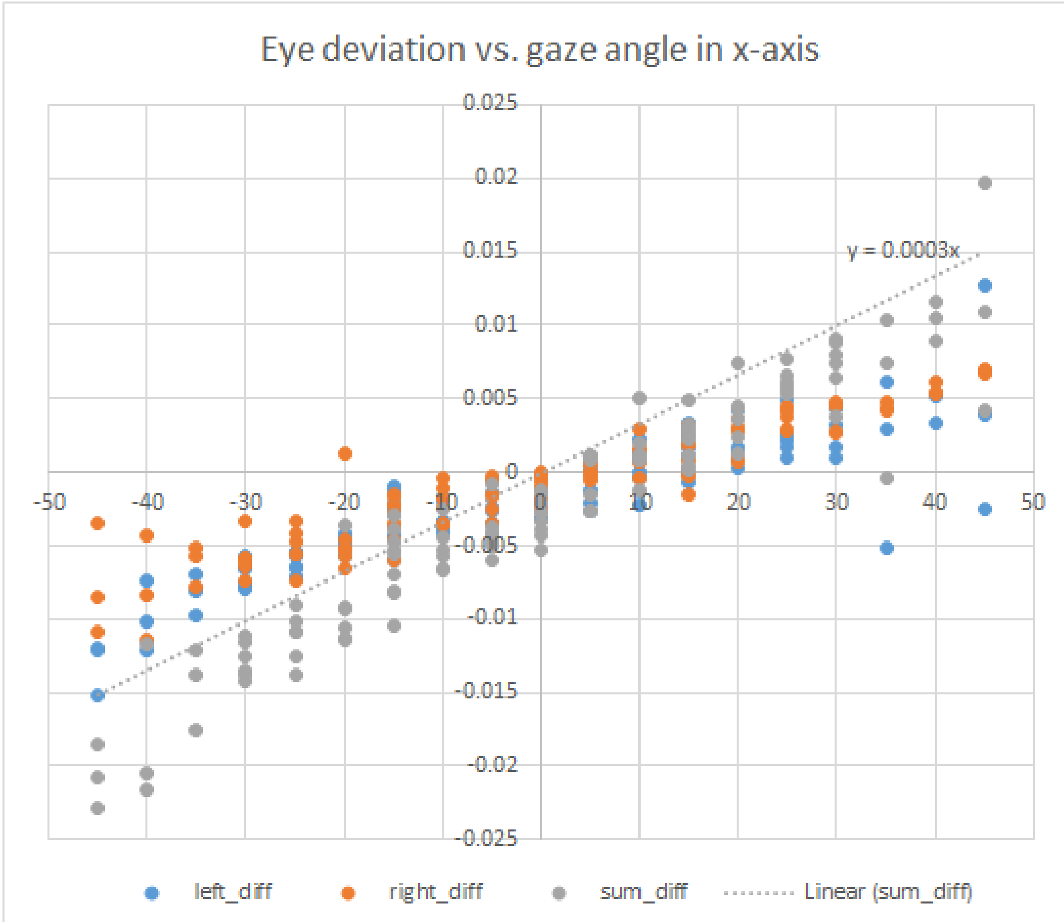


Fig 12. Eye deviation vs gaze angle in x-axis

Different from the method used for horizontal gaze angle using estimated center of pupil, the vertical direction estimation is calculated using the eye profile instead since the method used for horizontal estimation is not as reliable. We calculated the average y value of left and right eye corners, and compare it with the average y value of upper and lower eye profile points from face recognition. The smaller the difference, the higher direction the eyes are looking at, since from observation it is easily seen that when people look up, the curvature of upper eyelid will be larger comparing to that of lower eyelid, and the difference shrink when people looks downwards. Again we use linear regression to calculate the coefficient and the plot is shown below.

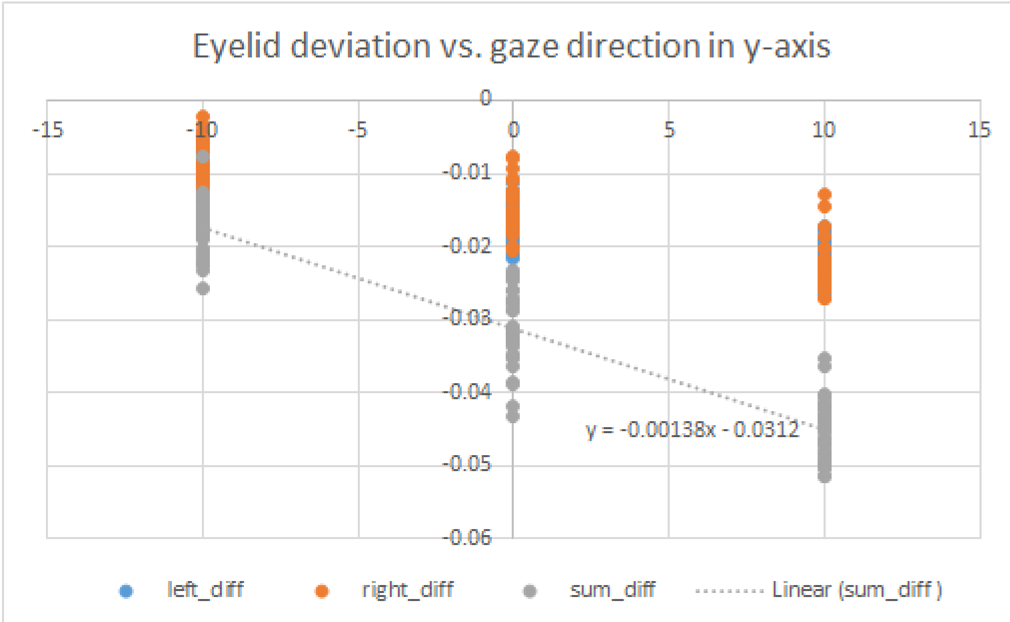


Fig 13. Eyelid deviation vs gaze angle in y-axis

We set a threshold of +-20 degrees for the test taker to decide if he is looking outside the screen horizontally, and +5 and -15 degrees to decide if he is looking outside the screen vertically. The asymmetry of degree in vertical direction is in accordance to the fact that the webcam is usually above the screen.

V. Evaluation and Result

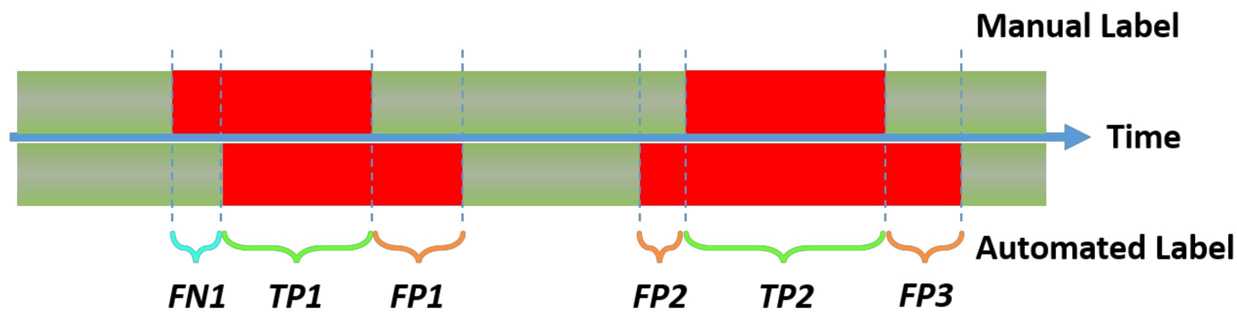
(1). Evaluation metrics

For the evaluation of the proposed system, we can use precision and recall. In general, precision and recall are defined as:

precision = TP/(TP+FP)

recall = TP/(TP+FN)

where TP, FP and FN indicates true positive, false positive and false negative respectively. Specifically in our system, the two evaluation factors will be calculated as in figure below.



We first manually label the time windows of a video clip that corresponds to cheating periods, which is taken as true label, and then label the same video clip with the proctoring system. We use the following formulas to calculate precision and recall:

$$precision = \frac{\sum TP}{\sum TP + \sum FP}$$

$$recall = \frac{\sum TP}{\sum TP + \sum FN}$$

where TP is defined as the time windows which are labeled both manually and automatically to be cheating, FP is defined as the time windows which is manually not labeled to be cheating but labeled by the proctoring system to be cheating, and FN is the time windows which is manually labeled to be cheating but missed by the proctoring system. In the figure, the axis is the time of video, and the upper label is from human judgement, while the lower label is the result of the proctoring system. In this particular case we have:

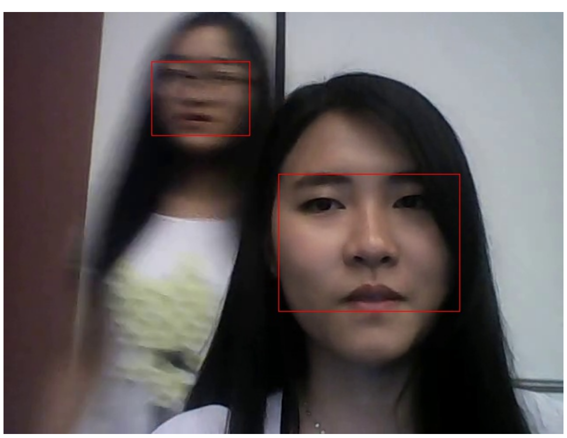
$$precision = \frac{\sum TP}{\sum TP + \sum FP} = \frac{TP1+TP2}{TP1+TP2+FP1+FP2+FP3}$$

$$recall = \frac{\sum TP}{\sum TP + \sum FN} = \frac{TP1+TP2}{TP1+TP2+FN1}$$

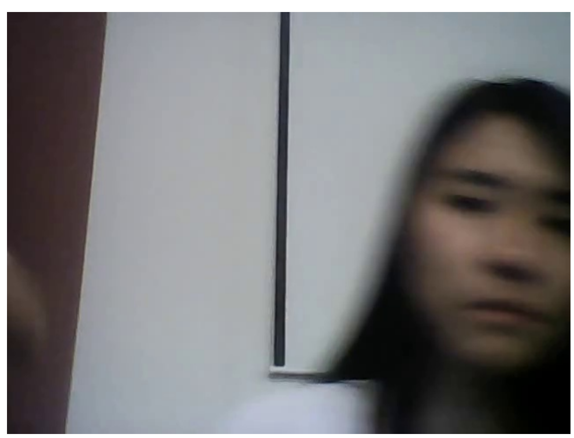
Tentatively, we would like to run the evaluation process for 5 to 10 video clips, and collect the precision and recall values. If the result is not satisfactory, we would go back to the step of facial recognition and eye movement detection algorithms and make adjustments.

(2). Experiments and Result

In order to get an reasonable and consistent result, instead of using the time periods, we use the sample frames we captured from the webcam as data for testing. We manually labeled the frames into cheating frames and non-cheating frames, and compare them with system labeled results.

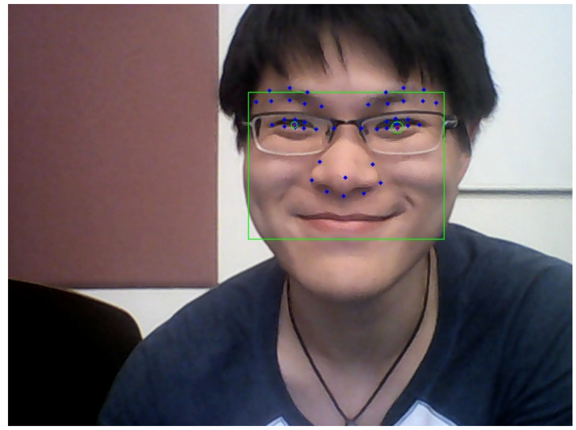


(a) True Positive



(b) False Positive:

(c) False Negative



(d) True Negative

Total test samples: 104

True Positive: 35

False Positive: 14

False Negative: 9

True Negative: 45

Precision: 71.4%

Recall: 79.5%

VI. Teamwork

We generally followed the plan we proposed while taking Professor Kender's advice into consideration.

1. Modules of program:
    a. Face analysis steps: Xiuhan
    b. Gaze analysis steps: Wentao, since we spent most of time discussing how to detect the center of pupils and Wentao implemented the algorithms, Xiuhan wrote codes for other parts.
2. Integration and Interface: Xiuhan
3. Test and experiment: Xiuhan
4. Report and document: Both

VII. References

[1] Howlett, Bernadette, and Beverly Hewett. "Securing and proctoring online tests." *online assessment and measurement: Foundations and challenges*(2006): 300-329.

[2] Bedford, D. Wayne, Janie R. Gregg, and M. Suzanne Clinton. "Preventing online cheating with technology: A pilot study of remote proctor and an update of its use." *Journal of Higher Education Theory and Practice* 11.2 (2011): 41-59.

[3] Cluskey Jr, G. R., Craig R. Ehlen, and Mitchell H. Raiborn. "Thwarting online exam cheating without proctor supervision." *Journal of Academic and Business Ethics* 4 (2011): 1-7.

[4] Timm, Fabian, and Erhardt Barth. "Accurate Eye Centre Localisation by Means of Gradients." *VISAPP*. 2011.

[5] Smith, Brian A., et al. "Gaze locking: Passive eye contact detection for human-object interaction." *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013.

[6] Megvii Inc. Face++ Research Toolkit. www.faceplusplus.com, December 2013.