CS4187 Computer Vision and Interactivity

Homework Assignment #3

WANG Yinuo 55670000

**INTRODUCTION**

**After learning about computer vision for a period of time, this assignment is to ask us to use the OpenCv Framework to carry out basic edge detection method on face and motion detection and tracking.**

**The first task is to** process the video “faceDetection.mov” and accurately detect the human face in the video.

Figure 1 Capture of faceDetection.mov

The second task is to draw interface using Color Tracking.

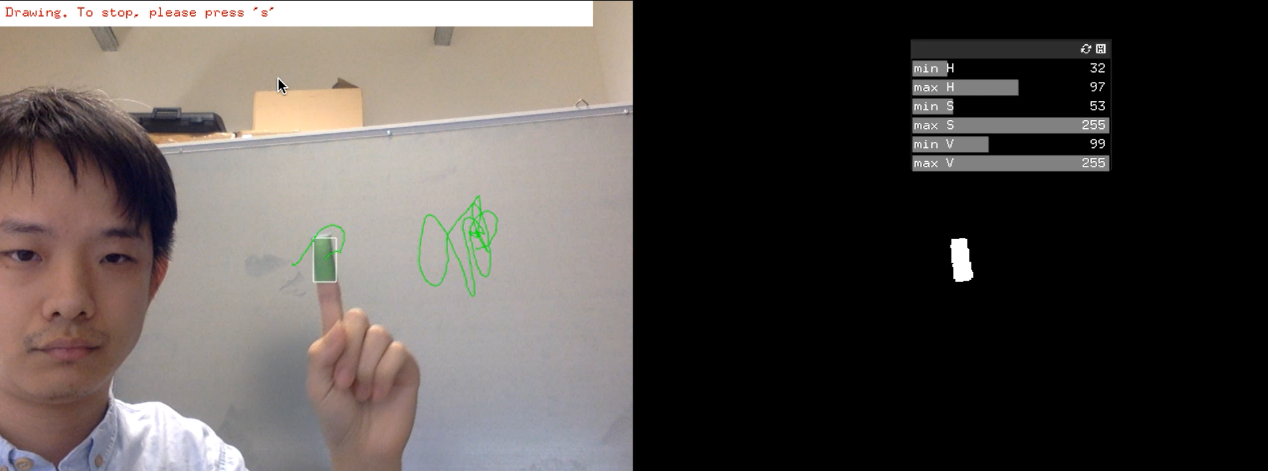
 Figure 2 example of task 2

Figure 3 Eye

Figure 2 glasses

The last task is to get the real-time images from the webcam, and detect and track the keypoints within the face area using the method of optical flow.

**RELATED WORK**

1. OpenCV Framework and c++ learning

2. Basic knowledge of face detection

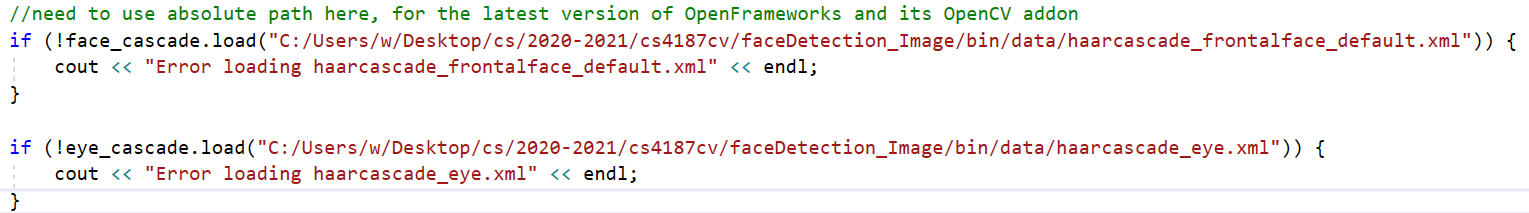
3. Basic knowledge of motion/color detection and tracking

**APPROACH**

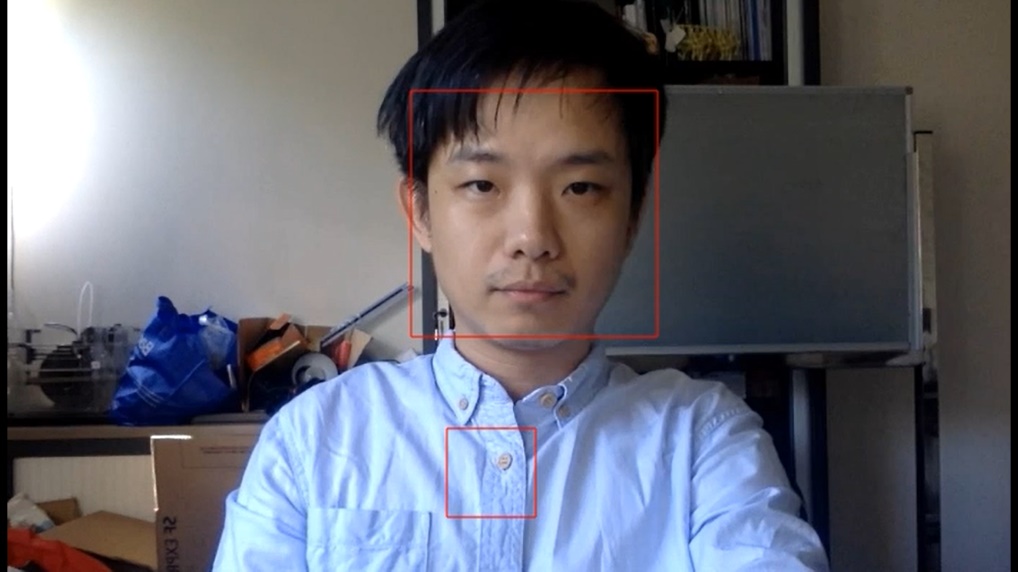
1. **Face Detection in Video**

In this task, we use the pretrained xml file to test the face in the video.

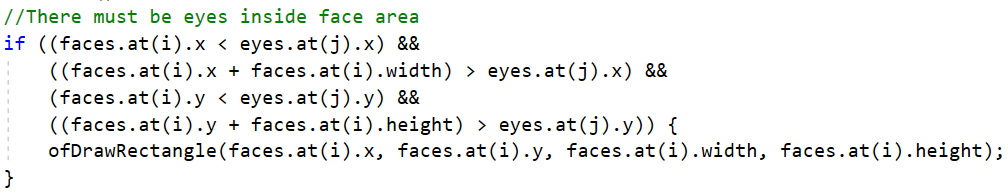
The main code is as follows, where haarcascade\_frontalface\_default.xml and haarcascade\_eye.xml are the pretrained face and eyes models respectively.

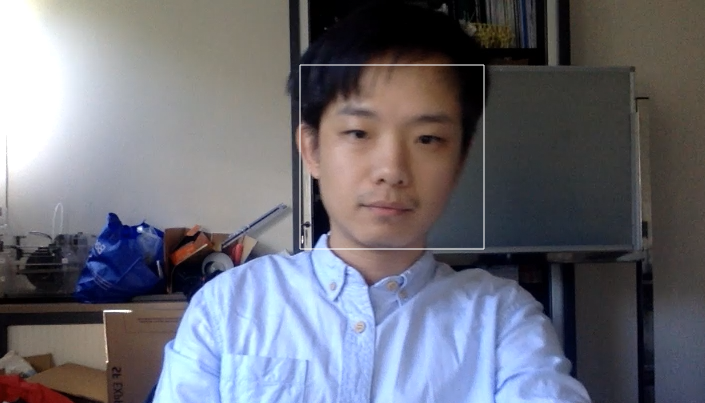


These two pretrained files should be loaded with the absolute paths to avoid error loading.

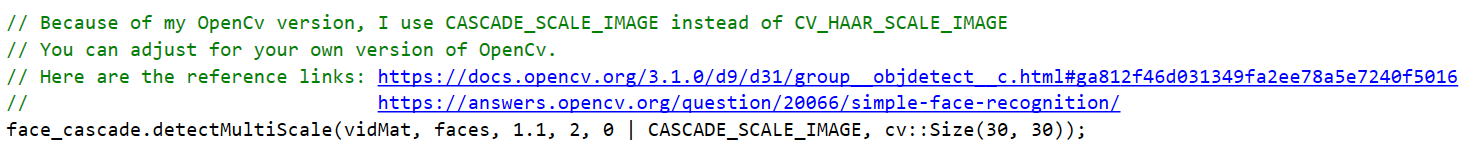
If we only use haarcascade\_frontalface\_default.xml model to detect face, there will be false positive occurred, as shown below.

To eliminate false positive, we should tell whether there are eyes inside the detected face, and then draw a rectangle to indicate the human face.



The captured result is shown below.

Besides, in this program, due to my OpenCv version is less than 3.0, I use the flag CASCADE\_SCALE\_IMAGE instead of CV\_HAAR\_SCALE\_IMAGE in the detectMultiScale() method.



You can adjust for your own version of OpenCv.

Here are the reference links: <https://docs.opencv.org/3.1.0/d9/d31/group__objdetect__c.html#ga812f46d031349fa2ee78a5e7240f5016>

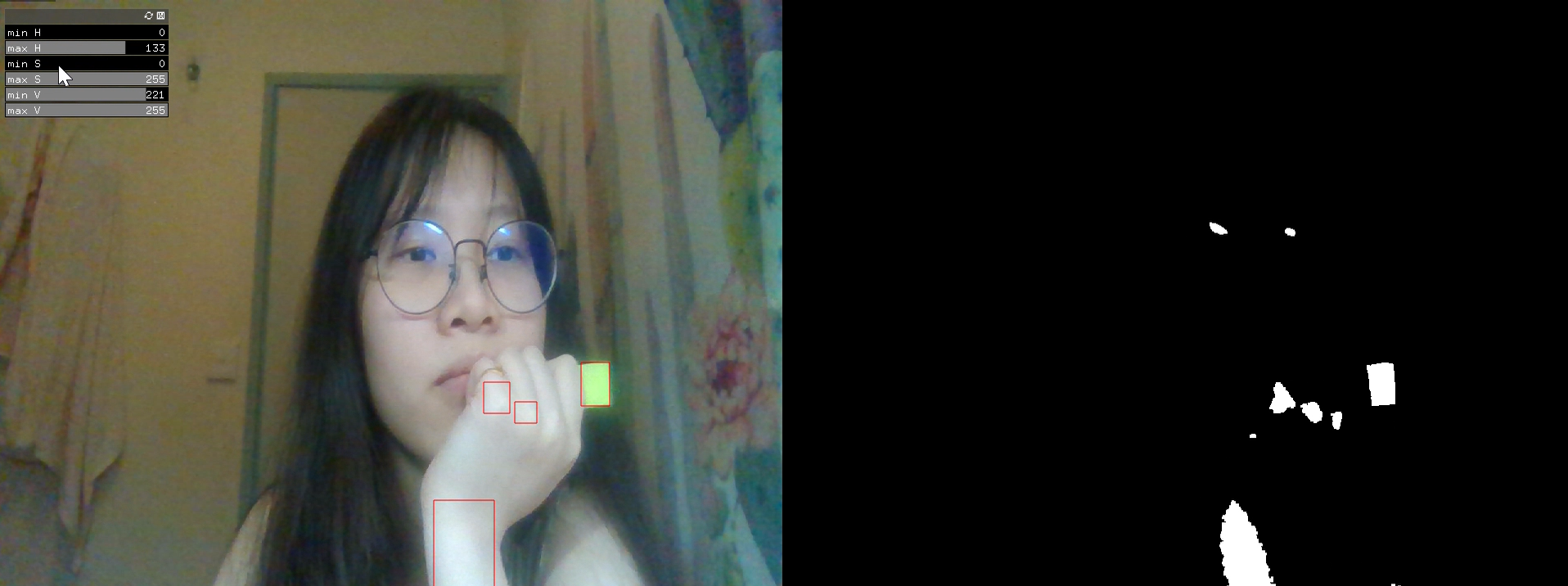
<https://answers.opencv.org/question/20066/simple-face-recognition/>

2. **Drawing Interface using Color Tracking**

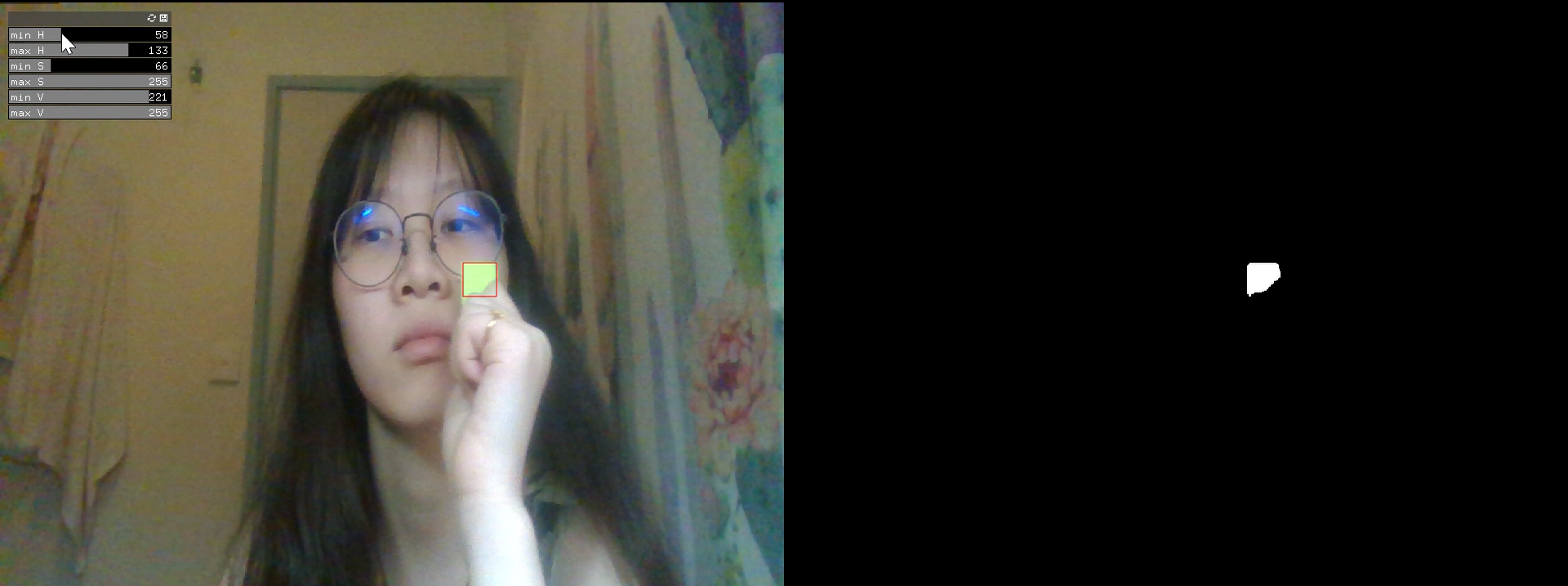
In task 2, we should use the method of color-based blob tracking, to fulfill the following interaction steps.

Step 1: Wear a green finger cap on the index finger.

Step 2: Adjust the sliding bars to track the green finger only in the webcam image. (1 points)

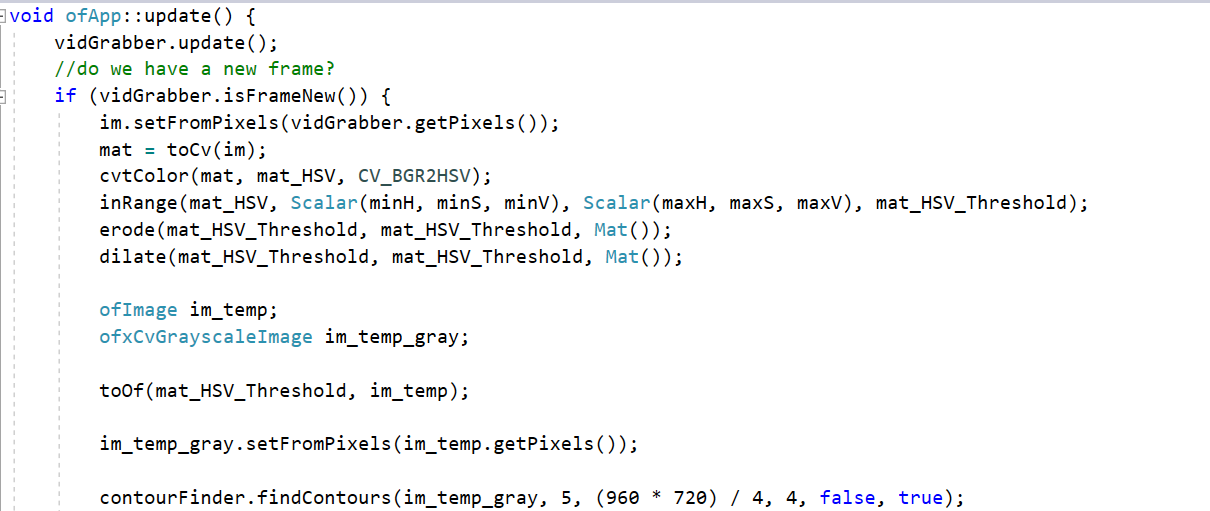


In this case, I used a green sticker instead of a finger cap, and then adjusted the slider to track my green sticker only in the webcam image.



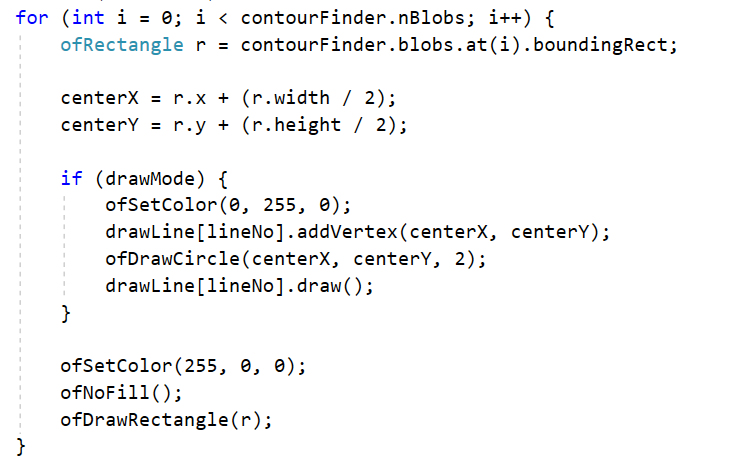
Here is the main code.

Firstly, set up the webcam and the slider bar for adjustment.

Then we get images from the webcam and set the image within the range and eliminate the noise by erode() and dilate() functions.

Then we use findContours() function to detect the objects in particular color.

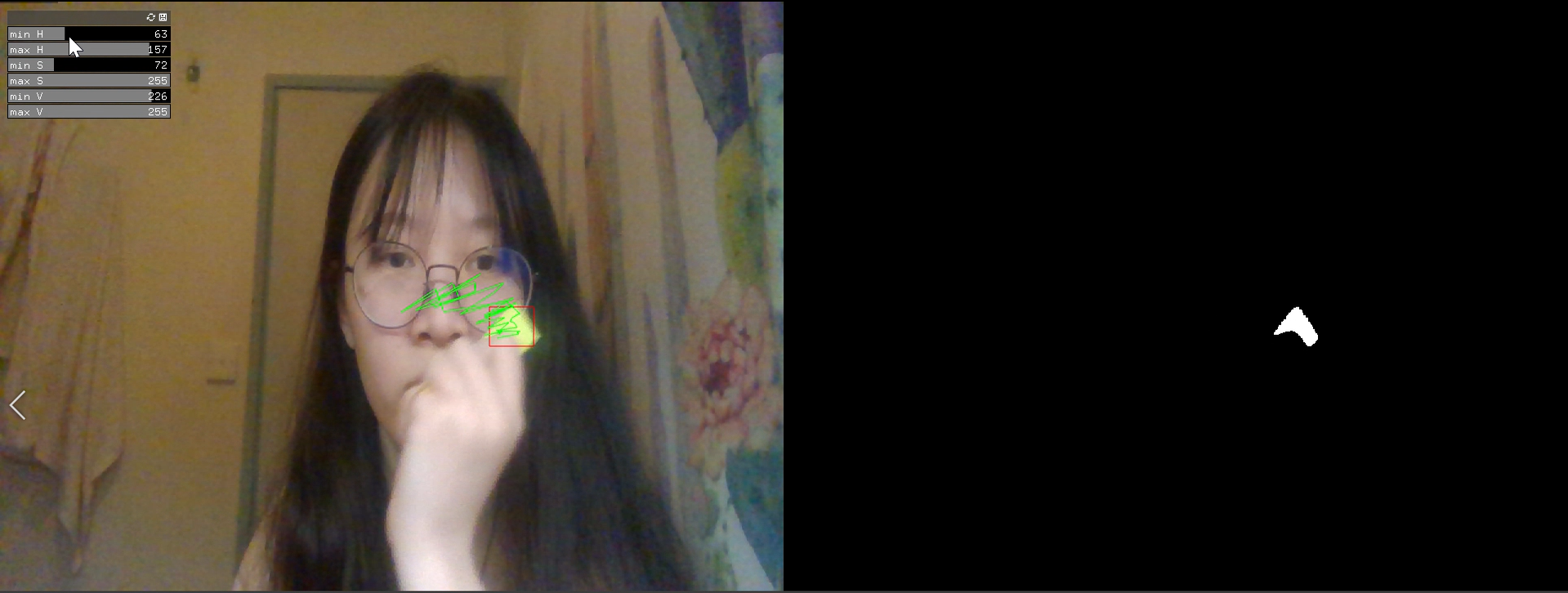
If we find something, we draw the results.



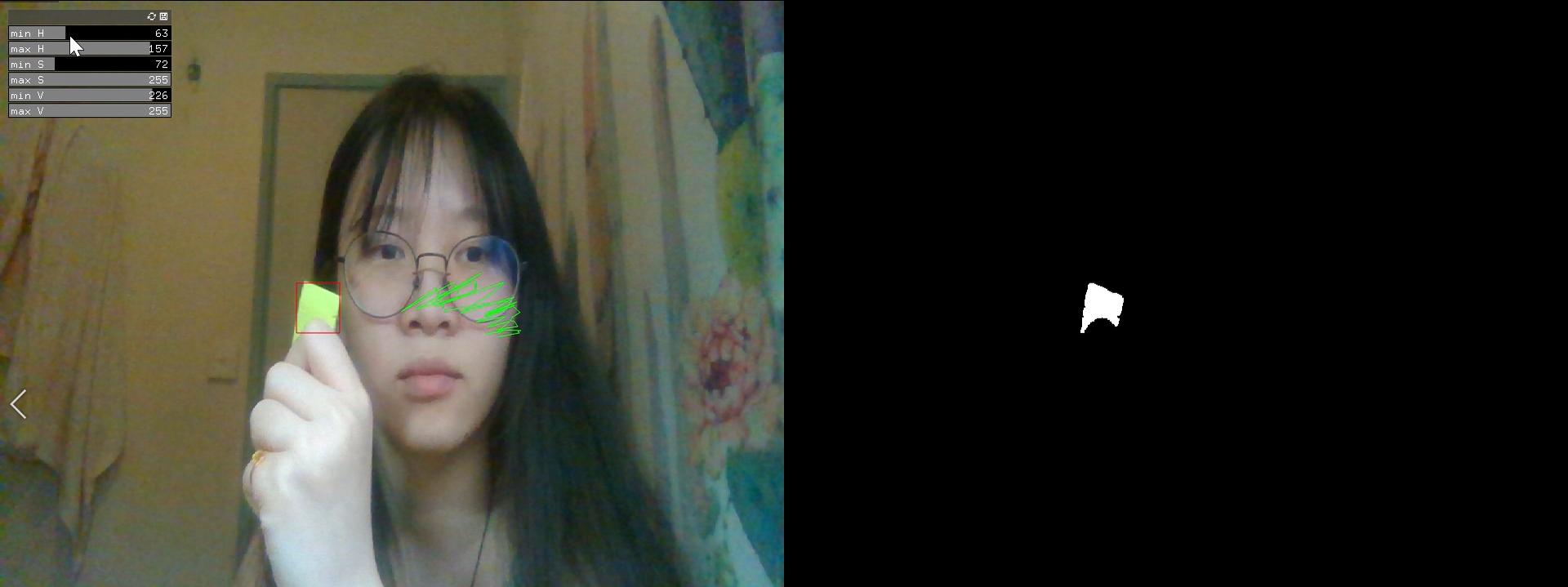
Step 3: Press key ‘s’ to start/stop the drawing mode.

Under the drawing mode, the user can move his/her finger in front of webcam and draw the line sketch with the color of the finger cap on top of the webcam image. (2 points)

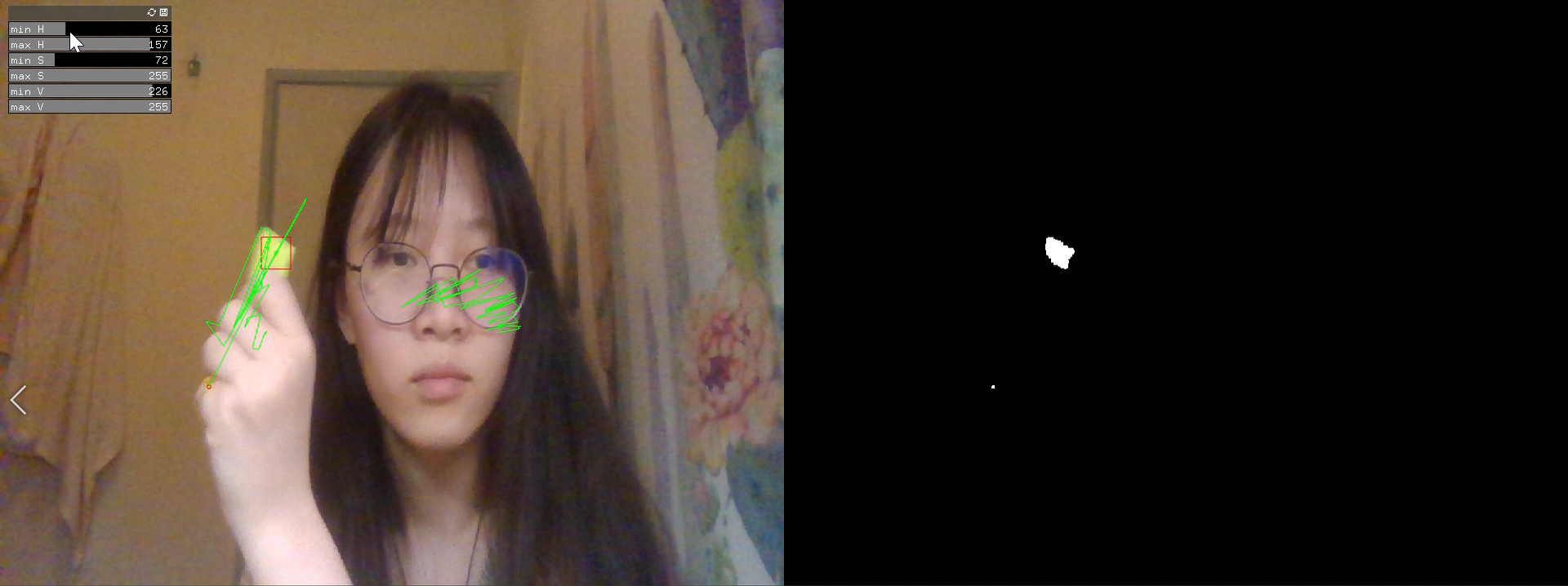
Drawing Mode:



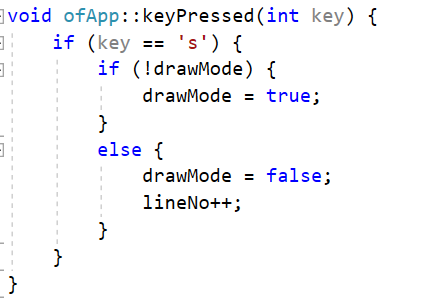
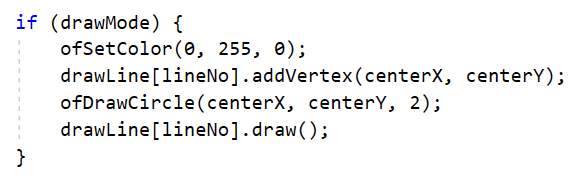
Stop Drawing:



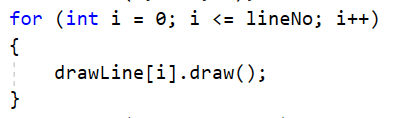
When the user stops the drawing mode, and presses ‘s’ to start again, a new line sketch will be created. (3 points)



Here is the main code.

In addition, We allow users to create different line sketches multiple times, so I created a ofPolyline variable “drawLine[10]” to represent different drafts, which means users can draw up to 10 different line sketches.



3. **Optical-Flow-based Face Tracking**

To complete task 3, we should get the real-time images from the webcam, and detect and track the keypoints within the face area using the method of optical flow.

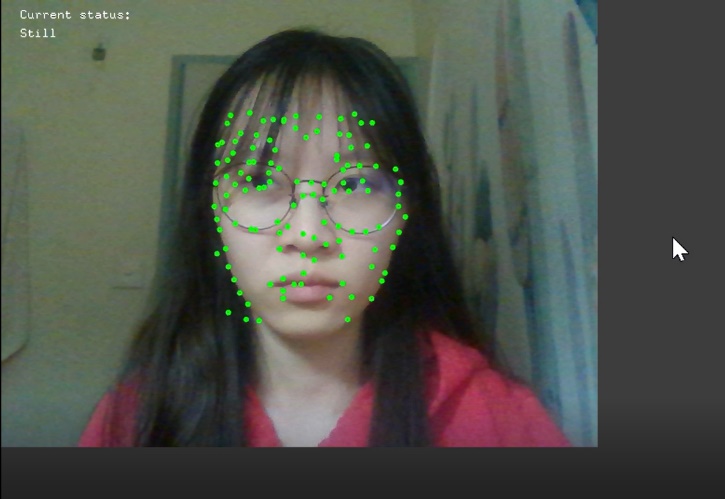
Detect the face/head movement, and display the messages accordingly as below:

- When the user shakes his/her head left and right, show the text “Shaking”

- When the user nods his/her head up and down, show the text “Nodding”

- When the user doesn’t move his/her head, show the text “Still”

Add a key-press interaction to your program, so that the program calculates the keypoints and starts tracking only when you press the key ‘r’.

Firstly, we should get the real-time images from the webcam and allow user to start tracking only when the key ‘r’ is pressed.

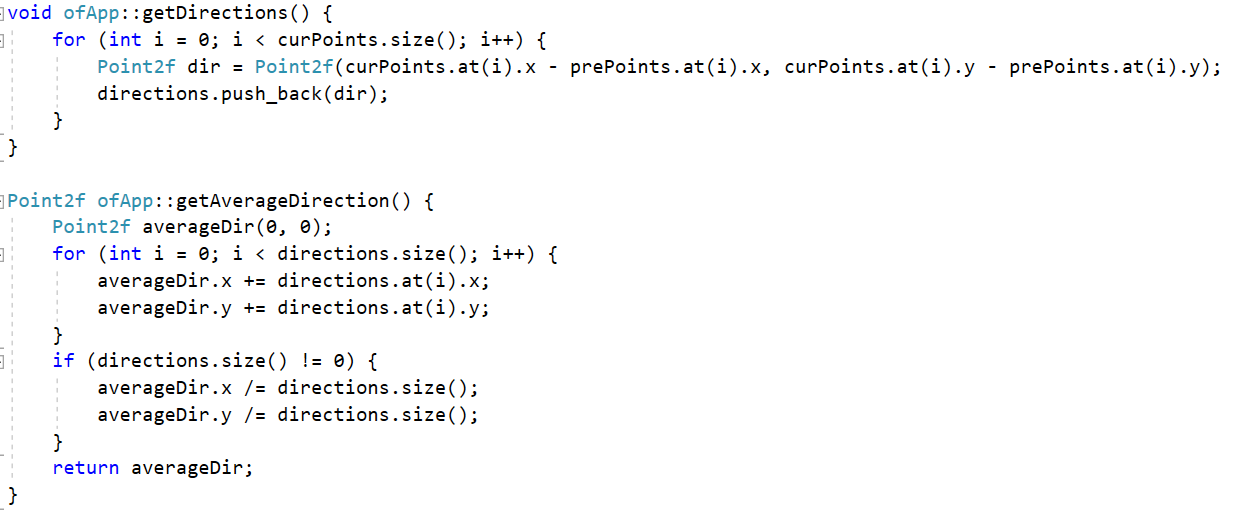
To accurately detect the face, we set the position of key points based on two models (haarcascade\_frontalface\_default.xml and haarcascade\_eye.xml).



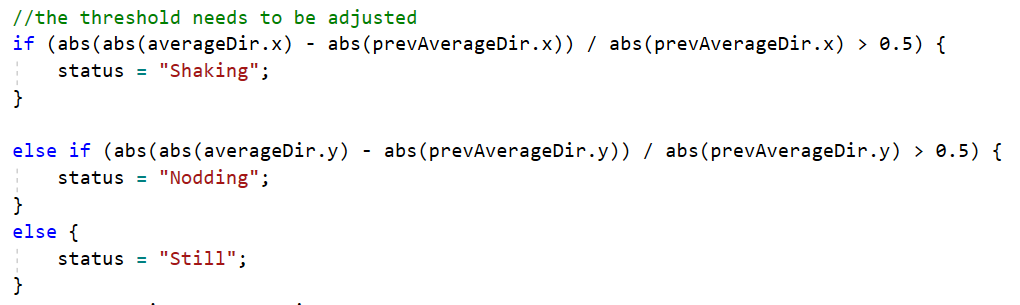
We track these key points:

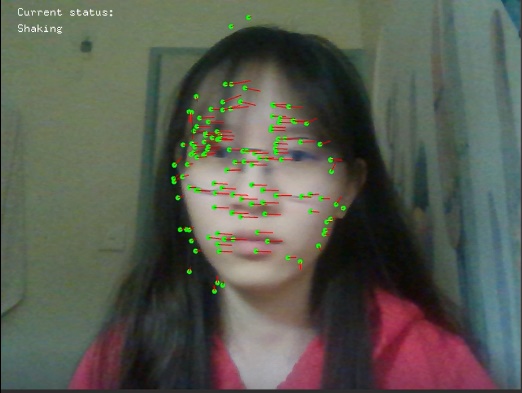
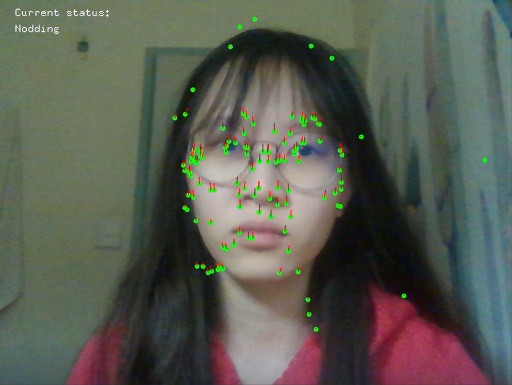
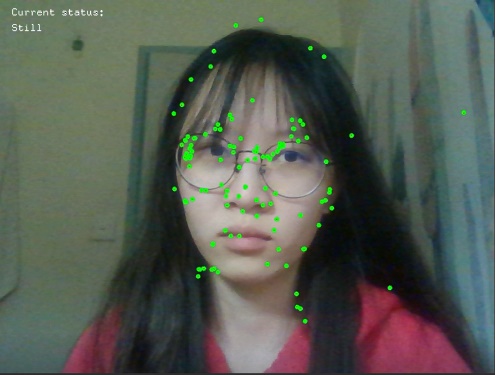


And get their directions:



Check whether the movement is larger than a threshold, which needs to be adjusted. If so, change the status.



We can trace three different states (shaking, nodding and still) and show the text using the function ofDrawBitmapString(String s, float x, float y).

Finally, if we want to clear all the key points, just press the key ‘c’, they will be all cleared.