UC San Diego

Swaying Attention: A Software and Hardware Approach
03/16/2021

Yusuf Morsi, Presenter
Pranav Maddireddy, Presenter

UC San Diego

Swaying Attention: A Software and Hardware Approach to Abating The Problem of Students Dozing Off During Online Learning

Utilizing Python, OpenCV, Arduino IDE, and a Sparkfun ESP32 to help solve the issue of students dozing off while learning online

UC San Diego

The Problem

Students Dozing Off During Online Learning

Problem Description

- The COVID-19 pandemic has made it clear that students have been dozing off while learning online.
- This is because of the common one-way approach of teaching [1] and because students can easily doze off without consequences [2].
- We decided that by providing a reminder that would make it more difficult for students to doze off, we would take away their ability to easily do so.

Statistics

- In **Figure 1**, it can be inferred that out of all the students that YoungTruthSurvey.org polled, 45% claimed that they get distracted while learning virtually [3].
- The Instagram poll in **Figure 2** shows that 90% of students polled claim that they have either faced ‘Zoom fatigue,’ or dozed off while learning online.
- The main takeaway from this data is that students getting distracted and/or dozing off while learning online from home is a major issue that cannot be ignored.

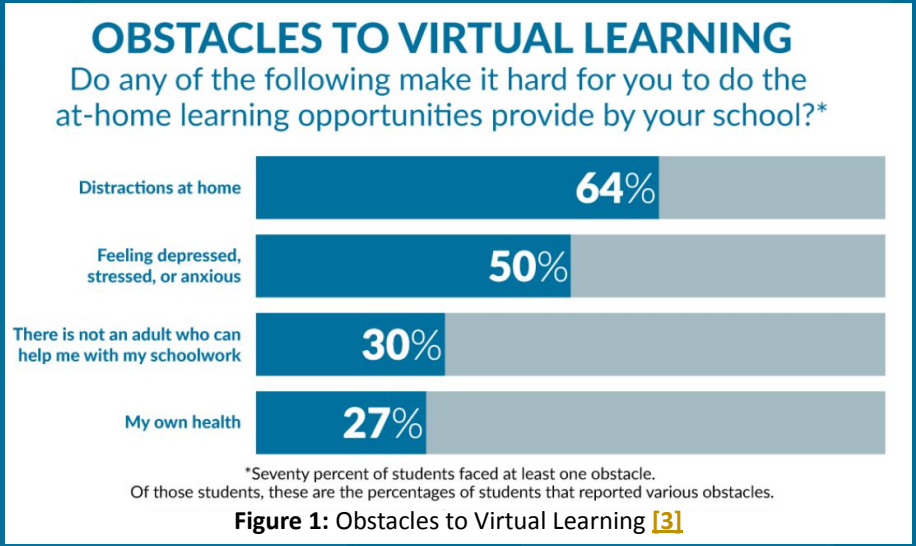

Figure 1: Obstacles to Virtual Learning [3]

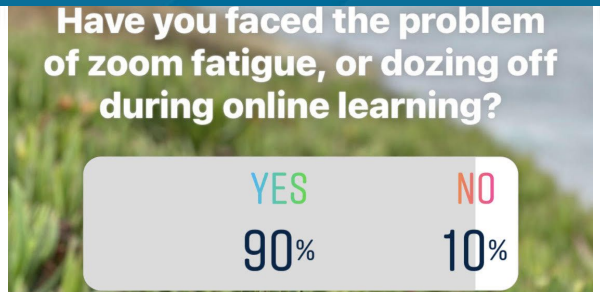

Figure 2: Poll Showing Percentage of Students That Doze Off While Learning Online

UC San Diego

The Design

Using OpenCV to Detect Students Dozing Off, and Utilizing Arduino IDE to Notify Them

Design Considerations

- Our solution to this problem involves using OpenCV to detect eye movement. However, we did have different designs in mind that still implemented OpenCV.
- Design 1: A software that uses webcams and OpenCV to track body or eye movement, and makes a loud noise with a buzzer when the user dozes off.
- Design 2: A system that notifies the teacher and that buzzes the user if they do not answer an easy-to-miss mental awareness question given at a random time during their lecture.
- Design 3: A service that helps teachers make their lectures more interactive and interesting to students to ensure that they pay attention.

Ultimately, we chose:

Design 1: A software that uses webcams and OpenCV to track body or eye movement, and makes a loud noise with a buzzer when the user dozes off.

Implementing OpenCV / Background Research

- First, we needed to learn to detect blinks with OpenCV. Using the Eye Aspect Ratio (EAR) equation derived from a Czech Technical University paper [4], we get a number that we can use to distinguish open eyes from closed eyes.

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

Figure 3: EAR Equation [4]

- We use this number in our code to identify whether eyes are open or closed [6].
- When a face is detected, the EAR ratio is calculated, and if it's lower than usual for 20 consecutive frames [7], our buzzer will start to make a noise.
- To achieve the code for this, we used, [7] and [10], which guided us to import certain libraries to get our Python code for detecting drowsiness . For further understanding, we also used [5]. We then built upon our Arduino code from previous labs for the buzzer implementation.

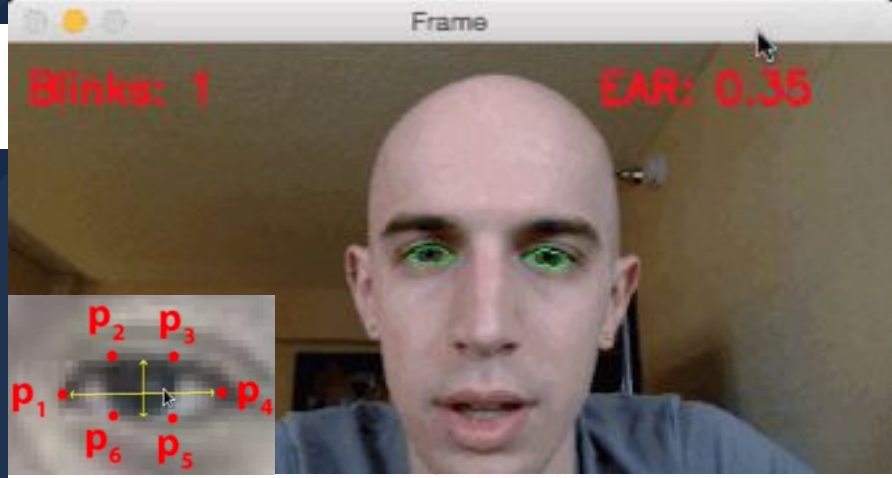

Figure 4: Blink Detection in Action **[6]**

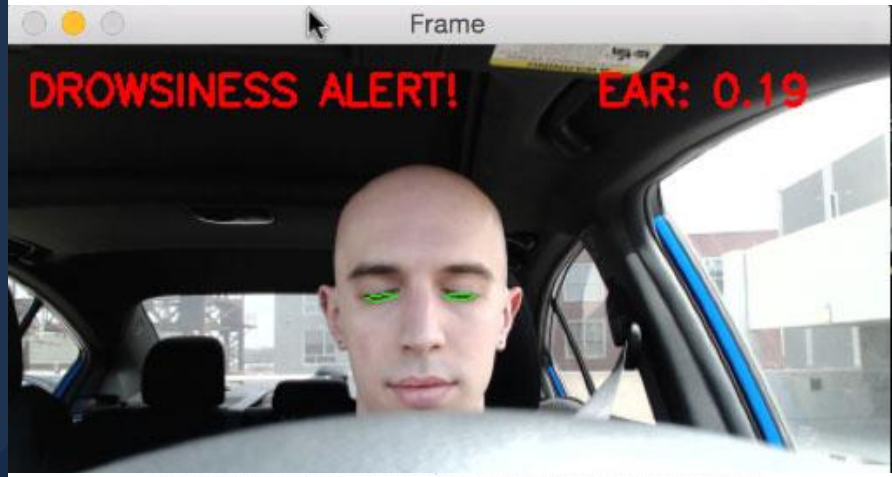

Figure 5: Drowsiness Detection **[7]**

Building The Circuit

- We based our circuit on the design that we implemented in our Lab assignments throughout the quarter.
- This included our Sparkfun ESP32, wires, a breadboard and a buzzer.
- **Figure 6** shows the circuit that we used with our code explained in the previous slide.

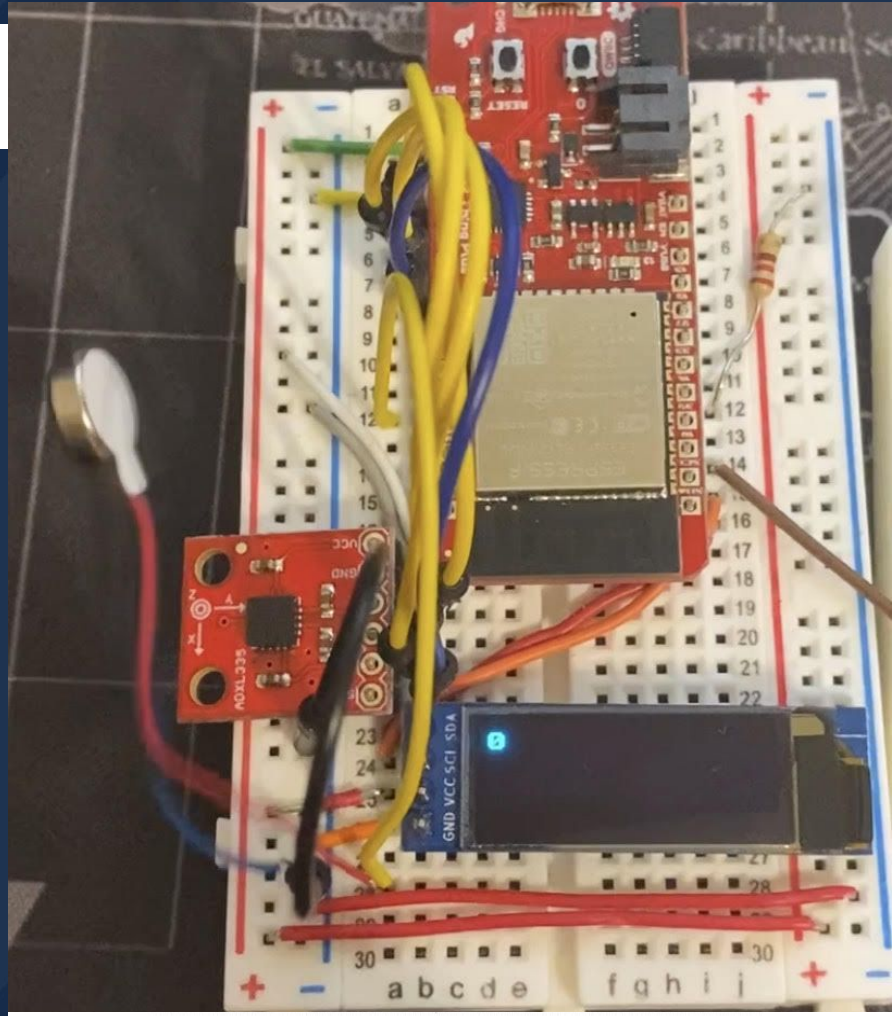

Figure 6: Circuit with Buzzer

Implementing Arduino

- Arduino code is based on Wearable Tutorial code
- Using communication, display and motor tabs
- Imported out Communication class from ECE16Lib
- 5 set commands
- Live
- Sleep, On!, Off
- Output on OLED anything else that is sent to Arduino

```
15  void loop() {
16    String command = receiveMessage();
17    if(command == "sleep") {
18      sending = false;
19      writeDisplay("Sleep", 0, true);
20    }
21    else if(command == "wearable") {
22      sending = true;
23      writeDisplay("Wearable", 0, true);
24    }
25    else if(command == "off"){
26      deactivateMotor();
27    }
28    else if(command == "on!"){
29      activateMotor(500);
30
31    }
32    else if(command != " "){
33
34      writeDisplay(command.c_str(), 0, true);
35    }
36
37  }
```

Figure 7: Excerpt from Arduino Code

UC San Diego

Testing The Design

How We Tested The Design

The Charge of the Project

- **Figure 7** is shows the mean of all the Eye Aspect Ratios from ten different plots that we created
- This is to determine the best Eye Aspect Ratio to use
- We determine that we should use a value slightly lower than the mean that we calculated, but also higher than the minimum threshold that we have from before.
- We determined that the threshold level should be 0.265

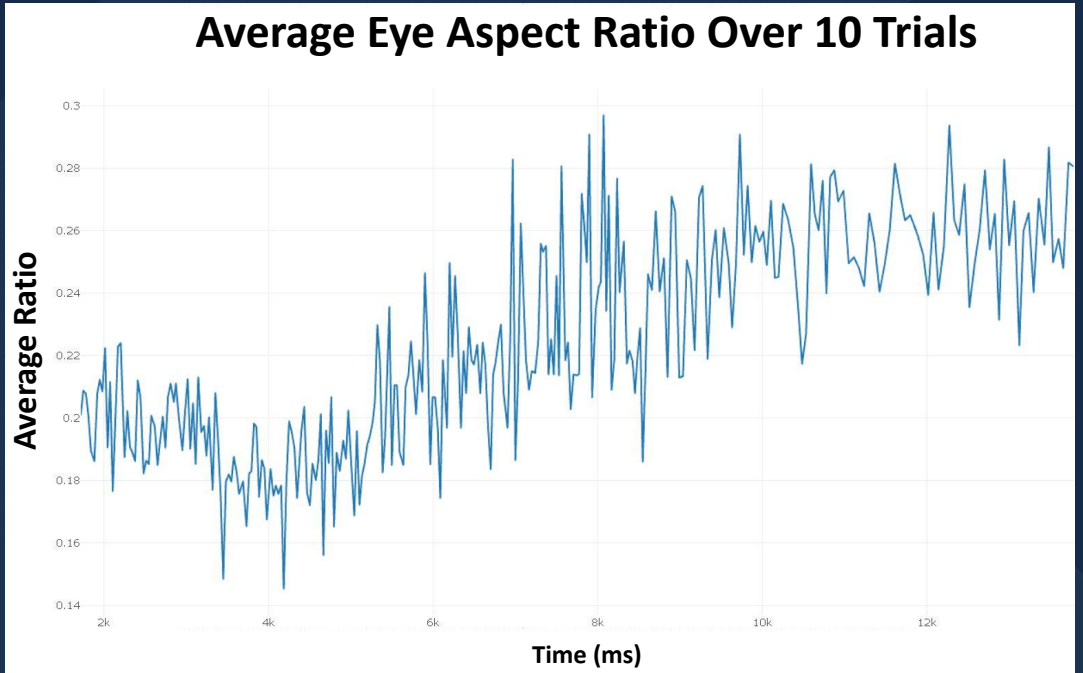

Figure 8: Demonstration of Final Product In Action

Demonstration of Final Product (Footage Taken by Pranav M.)

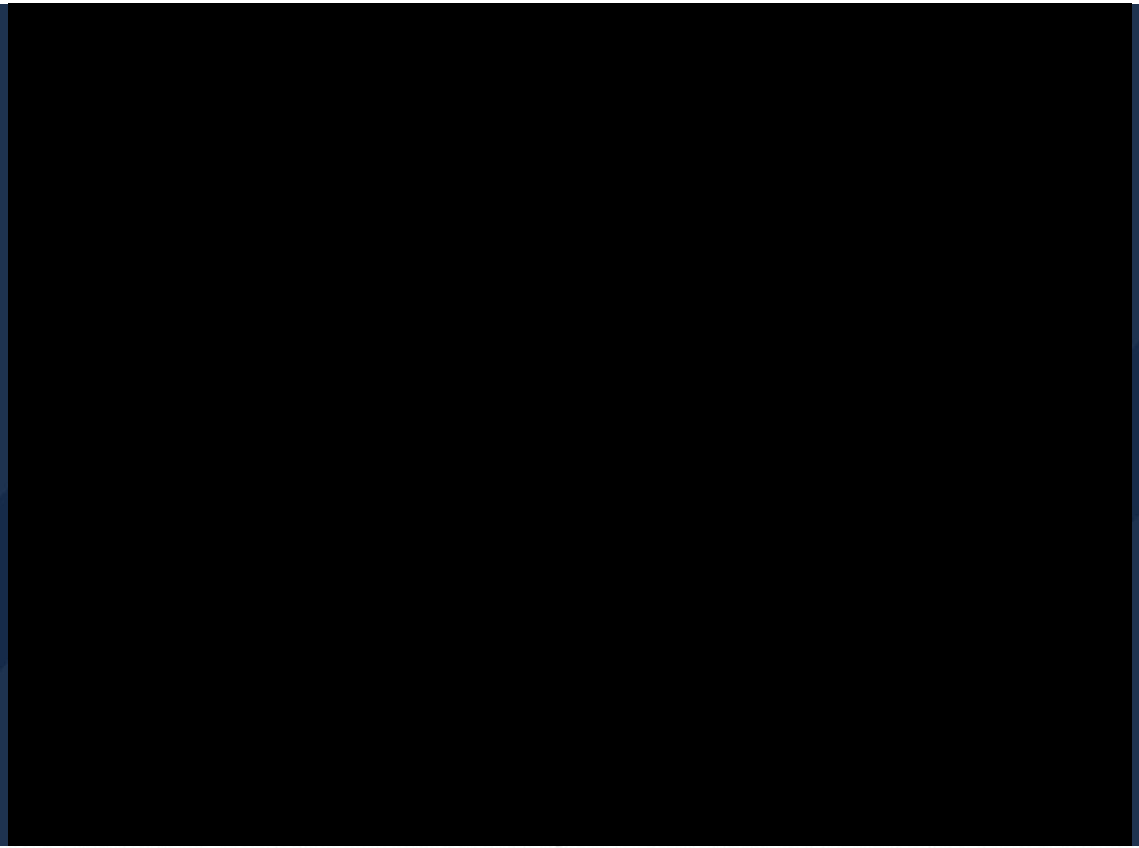

Figure 9: Demonstration of Final Product In Action

Thank You

Feel Free to Ask Any Questions!

UC San Diego

References

[1] https://www.businessinsider.com/why-students-fall-asleep-during-lectures-2014-7

[2] https://www.educationworld.com/a_curr/shore/shore005.shtml

[3] https://youthtruthsurvey.org/student-weigh-in/

[4] http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf

[5] https://pythonprogramming.net/loading-images-python-opencv-tutorial/

[6] https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/

[7] https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

[8] https://www.pyimagesearch.com/2017/03/27/how-to-install-dlib/

[9] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[10] https://github.com/akshaybahadur21/Drowsiness_Detection