AEROSPACE DROWSINESS DETECTOR

//TODO: TEAM NAME

Danielle Robinson	daniellerobinson@ucsb.edu	*Lead*
Victoria Sneddon	vsneddon@ucsb.edu	*Scribe*
Andrew Polk	polk@ucsb.edu	*Developer*
David Sun	davidsun@ucsb.edu	*Developer*
Brandon Tran	brandonrtran@ucsb.edu	*Developer*

INTRODUCTION

Aerospace provides technical guidance for all aspects of space systems. Current satellite launches require a team of trained professionals to be alert at all times of the day. However, full 24 hour attentiveness cannot be reasonably expected by human beings. Thus, in order to help Aerospace solve complex space-related science and engineering problems, we strive to solve the issue of drowsiness using facial recognition technology.

Aerospace's technical guidance has workers in situations where they need to be alert and respond quickly to changing circumstances. These can be high stakes situations where being drowsy or not paying enough attention could result in something going wrong. We want to provide an automated way for detecting if employees are drowsy or otherwise distracted. Our solution would innovate how detection of tiredness would create a safer work environment and lead to being more productive.

It is critical that the mission control personnel are alert during satellite launches. If they were to become drowsy, they could fail to notice problems that would've been caught had they been fully awake. By notifying the user when he/she is drowsy, potential accidents and mistakes can be avoided. Therefore, this product is important for the successful launch of satellites and safety of personnel in the aeronautical industry.

Many drowsiness detectors exist in automobile software. One open source solution uses the OpenCV framework, a real-time computer vision library. This program uses facial landmarks to determine if a car driver is drowsy. It calculates a threshold for which the driver's eyes are closed for a sufficiently long time, responding with an audible alarm to alert the driver. One automobile company has a drowsiness detection solution that utilizes an infrared camera above the steering wheel, detecting more complex signs of tiredness such as frequent blinking, deviations in steering, and distracted head movements. Most of these drowsiness detectors are used for the purpose of keeping drivers alert.

Our project is innovative in the way that we detect whether or not a face displays signs of drowsiness. We not only use facial recognition, we examine each part of the face to determine if it displays signs of sleepiness.

We plan on using Python for our coding language. The hardware we will use is a webcam as well as an ARM+GPU. The framework we will use is OpenCV. Our version control is both Git and GitHub. For project management, we will have Trello and Google Drive to communicate and stay updated on each other's progress. Finally, we will use Slack in order to communicate informally between group members.

TEAM GOALS

Our plan is to split our project into 4 sprints. During the first sprint, we plan on creating a foundation for the design/planning of the project. We will work on the proof of concept for the tools we're going to use as well as determine how we will analyze whether someone is drowsy or not. Furthermore, we will determine how we want to process images, install the framework and libraries for backend processing, and create an established workflow.

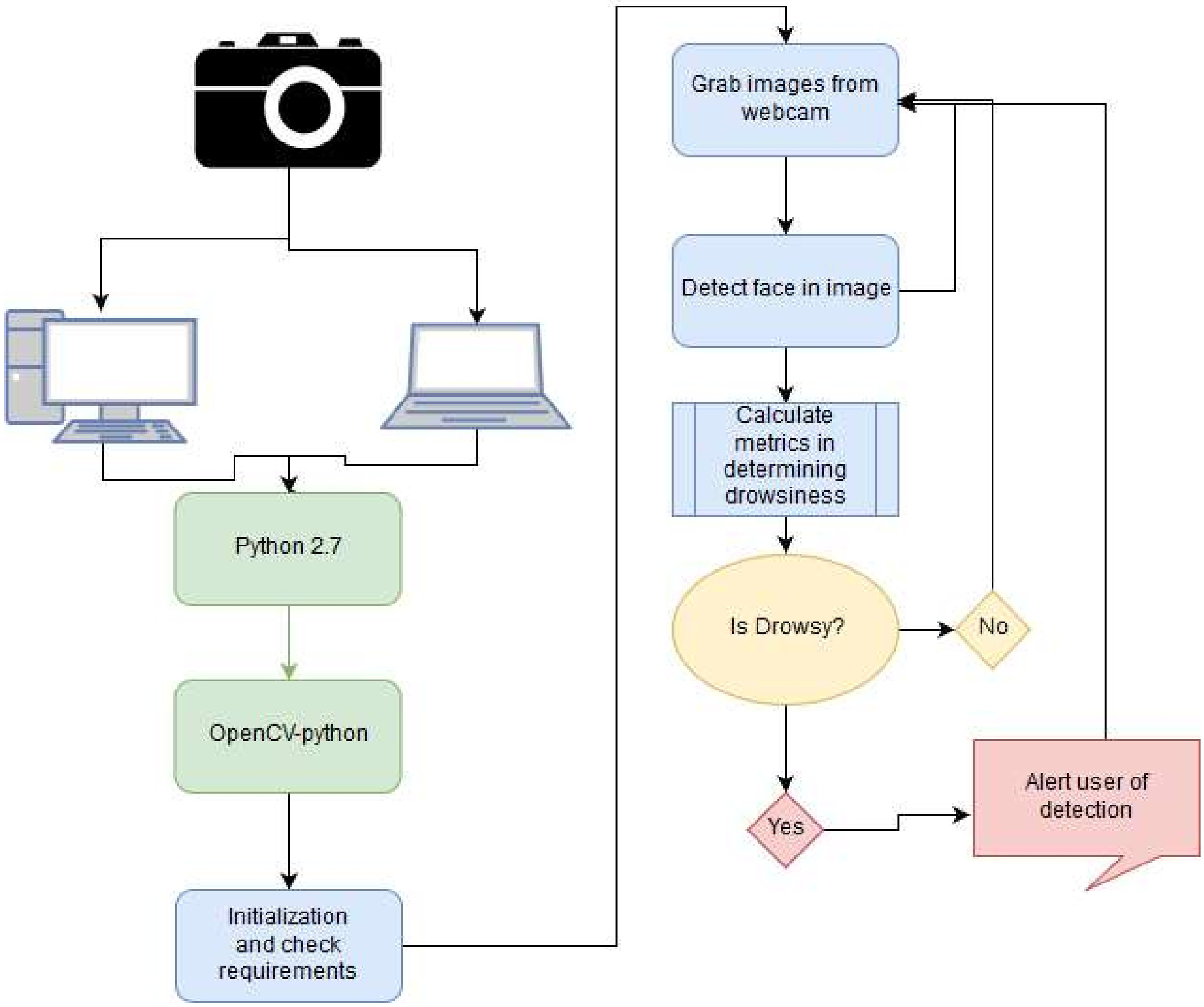
For the second sprint, we will begin by working on the basic framework of the project as well as the image processing for data inputs. We will also have continuous integration with GitHub and finally, we will continue to test different ways of determining drowsiness.

Our third sprint will consist of us integrating individual components of the code as well as finalizing the best way to process images and determine drowsiness.

For our final fourth sprint, we will work on a presentation that displays what we've accomplished so far. We will create a demo of our current project and create a plan for what we will be adding next quarter.

SYSTEM ARCHITECTURE

High Level Diagram, User Interaction & Design:



A user will run the executable program which will then check to make sure correct version of Python and OpenCV are installed. There will be a button to press start. Once start is pressed, the program will start all functionality and detect whether the user is drowsy. The button will then say stop which can be pressed to halt the program.

For testing/display purposes, have a version that displays the screen and what OpenCV is actually detecting/tracking.

REQUIREMENTS (USER STORIES)

1. As a user, I can try to open and run the executable program. The user does not have a webcam. The program will then deliver an error because it tests for a camera upon running.

   Git Commit:
   https://github.com/polkandrew01/DrowsinessDetector/commit/b7aaacfd507a81af458333d557e896cb31f03e3f

2. As a user I try to open and run the executable program. I do not have Python 2.7 and/or OpenCV installed. Program will deliver an error saying I must download Python and/or OpenCV in order to run the program.

   Git Commit:
   https://github.com/polkandrew01/DrowsinessDetector/commit/b7aaacfd507a81af458333d557e896cb31f03e3f

3. As a user, I can start running the program. My face is not detected. The program will then test for poor lighting and alert me if that is the problem.

4. As a user, I can try to run the program. If I do not have the correct libraries installed, I can run a wrapper script which will install the necessary libraries.

5. As a user I can start running the program. My face is not detected. The program will alert me if I am not in frame and continue to try and detect a face.

6. As a user if I blink, the program will update a running count of blinks on its graphical interface.

7. As a user I begin to fall asleep with the program open. The program detects that through facial analysis and begins to warn me of my imminent slumber.

   Git Commit:
   https://github.com/polkandrew01/DrowsinessDetector/commit/a6e6a03ecc0e6a58296d1fbea8cf7b7812ecd765

8. As a user I run the program. I am not sleepy and the program continues to run without alerting me.

9. As a user, my eyes blink slowly indicating sleepiness. The program alerts me that I am drowsy.

10. As a user I can interact with the program's interface to start it. To do this, I press a start button which will then start all functionality.

    Git Commit:
    https://github.com/polkandrew01/DrowsinessDetector/commit/99d20e8e9b768100644f87df053e8dccb77e859e

11. As a user when I begin to yawn, the program detects this motion. The program begins to alert me that I am drowsy.

Git Commit:
https://github.com/polkandrew01/DrowsinessDetector/commit/4ae62a07fac82963636ffb25b1dfce72ea8f19f4

12. As a user if my eyes are closed for x number of seconds the program will alert me that i am drowsy or possibly asleep.

APPENDICES

OpenCV
Windows/Linux/Mac OS
Webcam/Camera
Python (2.7)