Eye-Light PRDv1

Project Name: Eyelight
Team Name: Eye-Light

Team Lead: Louis Huang(huiyu_huang@ucsb.edu)
Scribe: Jacob Zamora(jacobazamora@ucsb.edu)
Developer: Kevin Mata(kmata@ucsb.edu)
Developer: Chistopher Lin (christopherlin@ucsb.edu)
Developer: Josh Lakin (jlakin@ucsb.edu)

Intro:

Background

Alcon is a company that specializes in eye care. They have helped surgeons and patients from multiple countries by providing them eye care devices. Alcon has suggested us to implement a software that can quickly and effectively analyze surgeries and analyze a surgeon's (or surgeons') performance. We plan to call this software "Eyelight".

What is the problem?

Surgeries take a lot of time to perform properly. In most minor cases, it can take anywhere from ten minutes to several hours. Realistically, for a surgeon who would wish to evaluate their performance, they would need to sit down and watch the entire procedure from start to finish. This would take at most a couple hours out of their day. Eyelight aims to reduce this evaluation time by cutting down a video into a manageable report that they could analyze more efficiently.

How is the problem addressed today?

Traditionally, surgeons would watch the entire video for feedback, therefore there are no video-to-report based software that we know of; therefore, Eyelight would be the first of its kind regardless of how well it performs. While there is no software like Eyelight, we could simulate the process by solving each step at a time. We would do this by combining the image and audio processing together to create a cohesive report that would highlight the video. Instead of having to go through the entire video, the software can break a video down into segments and summarize what happens in each segment. Breaking up a long video into parts and analyzing each part would be a much more manageable task than analyzing the video from start to finish, and the surgeon can skip to what part they want to focus on based on its description.

How will this be done?

Python will be used for the image and audio processing. With Python, we can import the necessary modules that will make our tasks simpler. The webapp itself will not have Python for the frontend and backend; instead, it will use other technology. Then we would need to figure

out a way to connect the frontend web app with the backend python server. A database is necessary for login information, video storage, and report storage. To accomplish this, we will use Firebase. To communicate with each other, we shall use the app GroupMe, and for the version control, we can use the universal standard of Git and Github.

Project Specifics:

1. Webpage Specifics
    a) Frontend:
        - We will implement a login page. Surgeons may need to have an account they can log in to so they will have easy access to their report
        - We will implement an interface that allows surgeons to upload videos to that webapp.
        - These tasks can be implemented with HTML and Javascript.
    b) Backend:
        - We must have our web app retrieve video data from the database so that videos can be analyzed.
        - We may use Python to implement this task.
    c) Database:
        - A database is needed to store videos, reports, and information about users/surgeons.
        - We plan to use MySQL or Firebase.
    d) User authentication:
        - Along with a login page, implementing user authentication is necessary.
        - We plan to implement this task with Python.

2. Report Generator
    a) Image Processing
        - We plan to use machine learning techniques to classify essential objects in a surgery video.
        - We will use the library OpenCV for identifying surgeons tools in a video.
    b) Audio Processing
        - The audio processing will first need to get the audio files from the video.
        - Then, it uses the Google API to convert the audio into closed captions.
        - We will then print the closed captions into a text file and store that text file, which will appear in the report.
        - Audio processing is used in conjunction with image processing to make the report.

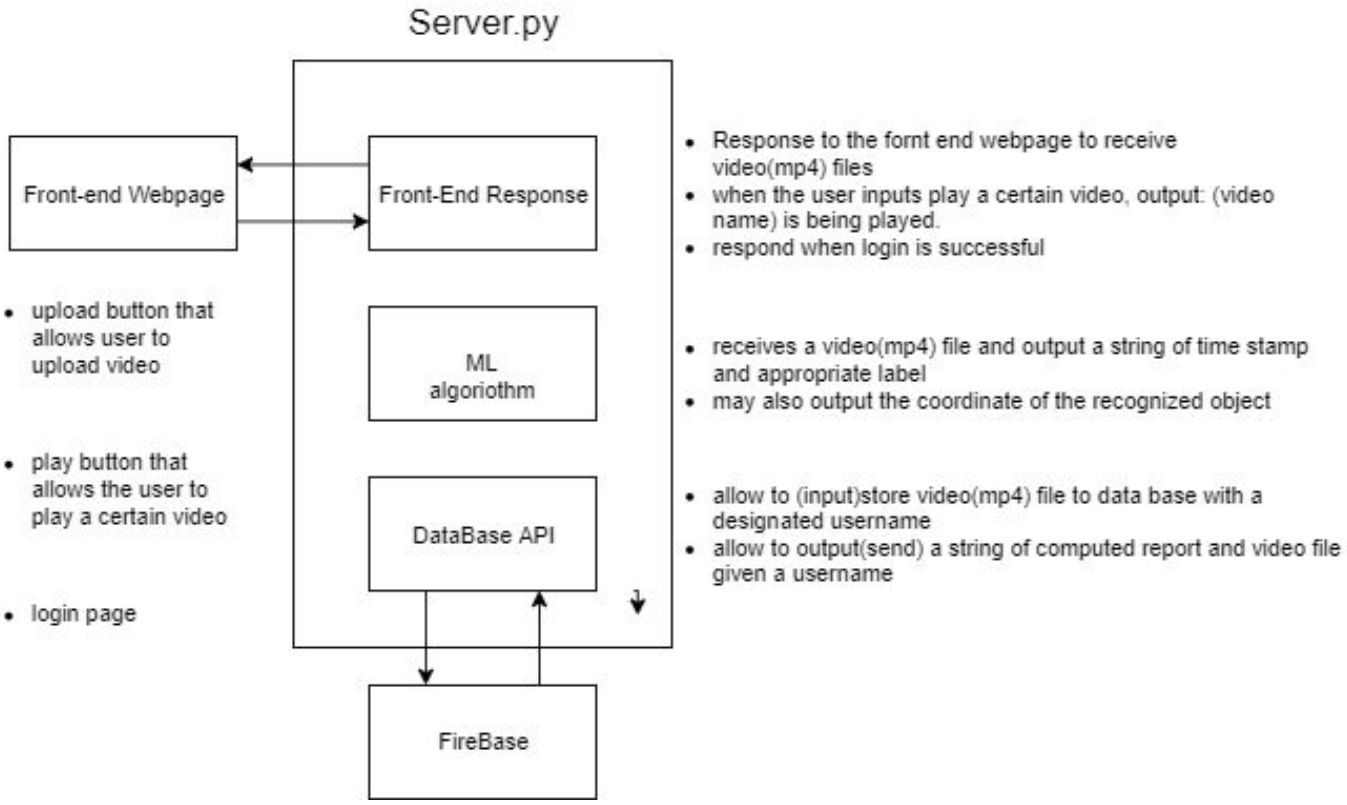
Assumptions:

While we work on this project, we have to assume that the user will know how to use the web app. We hope that most users would know how to log in to the web app, and then insert a video. It is expected that the video the user wants to upload has the MP4 format. We will make

the layout of the webapp simple to make sure the user would know the simple function of our webapp.

After a user logs in and uploads a video of their surgical operation, the report generator may take at least a couple minutes to evaluate the video. Because of this, we assume that the user has enough patience. We will also make the format of the reports simple so that a report is self-explanatory to a user. We hope that the auto-generated reports will significantly contribute to the surgeons' improvements in their operations.

System Architecture High Level Overview:



User Interaction:

Users of the service must register for either an account. Users will then be able to input a .mp4 video. The program will analyze the video and output a report. The report can also be saved into a database or released for the public.

User Stories:

1. As a user, I can sign in through the web app to view my previous surgeries.
    - Given the username and password entered is valid, the user will be directed to a webpage that will contain links of surgical footage. The user will be directed to the correct URL.

2. As a user, I can upload a video file and receive a report.
    - The webapp provides the user the option to upload a .mp4 file to the website. When this is done, a report is produced at least a couple minutes later and is visible to the user.

3. As an image, I can be classified into different tools by a machine learning algorithm.
    - The image will be classified into a valid label from the set of labels.

4. As a user, I can search through past surgeries videos by entering keywords.
    - If the keywords are valid, then it will return the past surgery videos that have those keywords.

5. As a program, I can break down the video into multiple smaller subset videos
    - Given the video is valid and of an acceptable type, the video will be broken down into smaller videos.
      Git Commit:
      https://github.com/huiyuhuang/Alcon-Surgeon-City/pull/5/commits/7c47755825e5b69077a28f9dbfd42ee1dde98de4

6. As a database, I can store the outputs of the program to keep a record of medical video highlights.
    - Given the outputs highlight report, it will be saved into the database and show up in the database.
      Git Commit:
      https://github.com/huiyuhuang/Alcon-Surgeon-City/pull/8/commits/f690effbdb939c43a5f9aade7709ccf75ea3bf16

7. As a program, I can get audio file from a video file
    - Given an acceptable video file, the program will process the video file and return an extracted audio file.
      Git Commit:
      https://github.com/huiyuhuang/Alcon-Surgeon-City/pull/5/commits/1e9ef1b79bf2611bb30eda2943ccccfd9dfbb622

8. As a program, I can make closed captions from an audio file

- Given the audio file is valid, then the program will automatically return closed captions.
  Git Commit:
  https://github.com/huiyuhuang/Alcon-Surgeon-City/pull/5/commits/1e9ef1b79bf2611bb30eda2943ccccfd9dfbb622

9. As a program, I can save closed captions into a text file.
   - Given closed captions, they will be inserted into a text file. The program returns this text file.
     Git Commit:
     https://github.com/huiyuhuang/Alcon-Surgeon-City/pull/5/commits/03b7408cca856ccc178b6393650f8cff52b077f4

10. As the closed captions, the words in me can be analyzed and compared to our predefined list to predict the highlights of the surgery
    - When there are closed captions as input, the program output the highlights.

11. As a program, given the highlights from the closed captions and images, I can generate the report.
    - Given the captions and images input are valid, its contents will be processed in the program and create a highlighted report.

Appendices (Technologies Employed):

1. Python 3
2. OpenCV
3. Firebase
4. Github
5. Trello
6. Javascript
7. Tensorflow