EE/CprE/SE 491 WEEKLY REPORT 10

11/14/24 – 11/21/24

Group number: 11

Project title: Slowpitch Softball Pitch Detector

Client &/Advisor: Nick Fila

Team Members/Role:

Andrew Vick - Machine Learning Integration

Casey Gehling - Client Interaction

Sullivan Fair - Individual Component Development

Ethan Gruening - Team Organization

Josh Hyde - Research

Cameron Mesman - Testing

- **Weekly Summary**
  - This week we began to develop specific functions within the Flutter application to create setup options, compile photos into bytes, and integrate byte transmission with an OpenCV tracking C++ backend. We were also successful in getting a proof of concept working on an iPhone which was able to detect, track and draw bounding boxes around a softball.

- **Past week accomplishments**

  - **Andrew Vick:**
    - This week, I was able to get our C++ code for detecting and tracking a softball to run on my iPhone. The app was able to utilize the YOLOv5 model we have and OpenCV's tracking modules. There were a lot of bugs with it but the app does provide a basic proof of concept with it being able to find the ball, outline it with a bounding box, and move that box as the ball moves.
    - I also discussed with the team on how we should handle implementing this C++ code into out final app since it was extremely complicated to get working even in its most basic form. We agreed upon refactoring the code to take in frames passed from fluter and then output the location of the ball and size so Flutter can handle the displaying of the bounding boxes and video feed. This will cut donw significantly on potential implementation errors since overall there'll be less C++ code for iOS to conflict with.
    - TLDR
      - Got our C++ code for detecting and tracking a softball to run on an iPhone
      - Created a plan of action for the rest of the semester and next.
  - **Sullivan Fair:**
    - This week, I utilized the framework I created to build the project on iOS with the C++ code successfully; however, the code was unable to run because it lacked an executable. I found that bundling the code into a Dynamic Library file may be required so the iOS environment can utilize the code while running. Bundling the code will likely require some CMake configuration, but based on the result of some team discussion, we plan on refactoring the backend code anyway.
    - For the team discussion, I helped devise a plan and created steps to complete for the remainder of the semester. Specifically, I plan on condensing the C++ code to exist in a single function that only returns the position of the ball. We decided the rest of the functionality should be included in the front end to ensure better cross-compatibility.
    - TLDR
      - Built the project with my created framework, but ran into issues with missing executables

        - Created a plan for the rest of the semester

- **Casey Gehling:**
    - This week I created a refined camera screen for our flutter app as well as patch pitches functionality. Spent time experimenting with different flutter camera solutions to maximize the native camera usage and decrease bloat.
    - Spent time building the Dart ffi bridge – was able to properly display some form of grayscale image to the screen proving the bridge between OpenCV and Dart, encountering some latency issues that will need to be addressed moving forward.
- **Ethan Gruening**
    - This week I created two new pages within the Flutter app for setting up the device for the field.
    - The first page is the SetupHeights screen which inputs the maximum and minimum height for the slowpitch game and records the height of the reference height that will be used to pinpoint known heights with taken pictures.
    - The second page is the SetupLines screen which prompts the user to take three pictures, a normal picture of the field, a picture with the reference height on home plate, and a picture with the reference height on pitcher’s mound. The user then has to click to identify the base line and known heights from the picture.
    - All height and position values are stored in a ChangeNotifier which can be accessed with a Provider in Flutter to both edit and get the setup values.
    - I also worked on the automatic deployment and compiling of our git main branch to an iOS IPA file to install on an iPhone. This will be useful for developing without a Mac in the future and testing the amount of time it takes to setup the application.
- **Josh Hyde**
    - This week I got the android side of the flutter app to fully run and work. I was able to get it to run on the emulator as well as my android phone. I did a lot of debugging and fixing to make sure that the dependency issues I had worked for it to actually run properly and get a working rendition of our flutter app and our code on an android phone or emulator. The first working version of this android side I got working on the older version of our camera app code for flutter. However, I implemented the new code that we added and did more fixing and debugging for it to actually run on the newer version of our code as well.

  - Additionally, I put in some work into implementing an illegal pitch code within dart to be able to potentially detect each pitch and to output whether or not each pitch is illegal or not. It is only a very rudimentary rough model right now, but I got some work done on it.
- **Cameron Mesman**
  - This week, I began working on outputting audio using Dart. Although I ran into some separate flutter issues that kept me from running it on my device, I found several helpful tutorials that walked me through the process. It's essentially as simple as creating an AudioPlayer object, setting an audio file as the source, and then playing the audio. Hopefully, next week my other issues will clear up and I can test it on a device to make sure everything works properly.

○ **Pending issues**

○ **Individual contributions**

NAME	**Individual Contributions** *(Quick list of contributions. This should be short.)*	**Hours this week**	**HOURS cumulative**
Andrew Vick	OpenCV integration into iOS	7	61
Casey Gehling	Screen development, C++ -> Dart implementation, debugging	8	59
Sullivan Fair	Created plan for the rest of the semester, finished framework testing	4	56
Josh Hyde	Android app running, illegal pitch code work	7	57
Ethan Gruening	Data collection, Python to C++ translation, multithreading research.	8	67
Cameron Mesman	Researching C++ implementation on iOS	6	46

Plans for the upcoming week

- Andrew Vick
  - Simplify C++ code to make it easier to use with Flutter
  - Optimize our object detection and tracking code
- Casey Gehling
  - Screen development
  - Improve detection latency
- Ethan Gruening
  - Make a cleaner and accessible screen for the setup of the app.
  - Test the amount of time it takes to set up the device for a field.
- Josh hyde
  - Get a finished dart code model of the illegal pitch detector
  - Help integrate everything together and make sure everything works properly on android
- Sullivan Fair
  - Condense C++ code
  - Integrate the C++ code with the front end once it is refactored
- Cameron Mesman
  - Clear up issues running flutter app on my device
  - Test audio on my device
  - Possibly link audio to height detection to play audio when an illegal pitch is detected