EE/CprE/SE 491 WEEKLY REPORT 08

10/31/24 – 11/07/24

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 made great progress involving converting our code into C++, running C++ code in our app, and creating prototypes for height detection. We were able to create a "Hello, World!" C++ script to run in Flutter. Also, we now have a more modular implementation of Andrew's existing script written in C++ which, with some further testing, we can use to test different models within our app. In addition, we created our first height detection prototype that has complete calibration. Finally, we wrapped up our research surrounding machine and non-machine learning along with starting research on video storage.

- **Past week accomplishments**

- **Andrew Vick:**
    - I have completed implementing our Object detection and tracking code in C++. From my testing it is considerably more reliable than previous methods we have tried. By leveraging tracking algorithms and object detection models, there is no significant impact on FPS.
    - Since I was able to get everything working in C++ I began trying to get it to run on a phone as well however, I was unable to get it to work. This is because both iOS and Android require a specific OpenCV framework, and the ones they offer for download don't include the tracking algorithms we use. To get around this, we need to build the framework from source and manually include the modules we use. When I attempted to compile the framework for iOS using the code they push out for this the build would fail. I found out from OpenCV's github page that there is a bug in latest version of OpenCV that is preventing anyone from building the framework.
    - TLDR:
        - Finished implementing our object detection and tracking code in C++
        - When trying to run our code in a flutter app I found out that we need to use a specific OpenCV framework for mobile operating systems
        - I was unable to build the framework we need for iOS from source due to a bug in the latest version of OpenCV
- **Sullivan Fair:**
    - This week, I worked on developing prototype flutter screens. I developed a navigation bar on the bottom of the initial screen with a home, camera, and settings buttons. Additionally, I tested the widgets on my phone and confirmed the buttons worked as expected.
    - In their current state, the home and settings screens just display a line of text with the name of the screen. However, I set up the camera button to trigger the C++ code I developed last week. That confirms that we should be able to run our detection code when entering a specific screen.
    - TLDR
        - Developed initial Flutter screens
        - Ran C++ code when entering the camera screen

- **Casey Gehling:**
  - This week, I worked on implementing some of our basic screen designs for our final flutter application. Developed our camera screen, which will be used to view the incoming softball pitch/configure pitch detections options, as well as made other screens accessible for both settings as well as past pitches. Tested all of this functionality on an actual iPhone to simulate what our app would feel and look like in use.
  - Did some review of our previous mockups and have begun to integrate those designs into our various screens. Focusing on beginning to integrate our C++ within our application to ensure our solution runs seamlessly.
- **Ethan Gruening**
  - This week, I worked on gathering more research data for testing the current detection protocols. The testing videos all have gridded known heights as markers to validate test runs. A Go Pro and tripod were used to film these testing videos.
  - Additionally, I worked on translating last week's Python script into a C++ project to integrate into our current main branch to run on the Flutter app.
    - I am working to continue to isolate our program in classes for each calibration technique and tracking during translation to help Flutter run specific C++ programs for specific functions, not only one program for everything.
  - Starting research and development on multithread approaches to a layered detection system using OpenCV, KCF, and YOLO to find a softball.
  - Continuing adding functionality to find the maximum height of a softball pitch.
- **Josh Hyde**
  - This week I put a little more research into the Android part of the app and started trying out potential solutions for this Android side and potentially trying to get it to run on our pre-existing Flutter app stuff that works on the IOS side. I got my Flutter stuff all fully downloaded and ready and implemented the pre-existing Flutter app stuff on my Android build as well.
  - I also put time into researching a little more about previously used methods of height detection that have been used before in other similar projects to make sure we aren't missing anything too important in regards to the height detection models we currently have and use.

- Cameron Mesman
  - This week, I continued researching how to run C++ code on iOS. From what I found, it will likely be easier than I initially thought. There have been some recent updates to Swift to make it more compatible with C++. Essentially, all you need to do is include the header files of our code in the Swift code, and then you can call the C++ functions directly. Because I don't have a device to run MacOS and write some Swift, it is difficult for me to get anything running on my iPhone. Hopefully, next week, I'll find a way to get it running on my device. Either way, if what I found is accurate, we should not have much trouble running the C++ code we need to.

- **Pending issues**

  Currently, when trying to build the OpenCV framework from source to run on a phone it is failing. We can either try to find the bug in OpenCV's code and resolve it or use an older version of OpenCV that does not suffer from this issue.

- **Individual contributions**

NAME	**Individual Contributions** *(Quick list of contributions. This should be short.)*	**Hours this week**	**HOURS cumulative**
Andrew Vick	OpenCV testing, Object detection, photogrammetry research	8	48
Casey Gehling	Flutter screen development, screen mock touchups	7	45
Sullivan Fair	Developed Flutter screen, Trigger C++ code to run	7	46
Josh Hyde	Android research for app, Height detection research	4	43
Ethan Gruening	Data collection, Python to C++ translation, multithreading research.	8	54
Cameron Mesman	Flutter overview, screen sketches	6	34

	improvement		

- **Plans for the upcoming week**
  - Andrew Vick
    - Get our C++ code running on a phone
    - Optimize our object detection and tracking code
  - Casey Gehling
    - Integrate C++ script into flutter app
    - Flutter screen implementations
      - Settings and past pitches screen
  - Ethan Gruening
    - Upload and clean up data collection
    - Have translations merged with the main branch
    - Create a design for potential multithreading object detection approach
  - Josh hyde
    - Develop a working model of our app on flutter, or at least get a version on android that can run
    - Look into some more height detection models/work
  - Sullivan Fair
    - Continue developing the Flutter screens and try to implement Casey's camera plugin
    - Work on integrating our C++ detection code into the Flutter back-end
  - Cameron Mesman
    - Get C++ code running on iPhone
    - Possibly run flutter app with C++ code on iPhone