

EE/CprE/SE 491 WEEKLY REPORT 09

10/07/24 – 11/14/24

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 primarily focused on the deployment methods of our C++ project into the Flutter application and installing the libraries needed for object and height detection. This process was mostly done through research and trial and error. The camera setup process is also being built in C++, though we are weary to continue heavily without finding a secure version for OpenCV to run on both iOS and Android.

- **Past week accomplishments**

- **Andrew Vick:**

- This week I continued working on getting our C++ code for object detection and tracking to run on an iPhone. This, however, proved to be more challenging than we anticipated. Last week we were running into problems with the OpenCV iOS framework not having the modules we needed and being unable to build our own from source. I was eventually able to get a framework built that had the required modules and targeted the correct iOS however, that created new errors.
- After adding the new framework to the iOS project, Xcode would throw errors because it didn't like the data structures in OpenCV because Apple redefined them. I have slowly been going through our C++ code and explicitly referencing OpenCV for the different structures that are throwing errors.
- TLDR:
 - Got a framework compiled for OpenCV that will work with the current iOS and has the modules needed.

- **Sullivan Fair:**

- This week, I began attempting to integrate Andrew's C++ code into our Flutter app on iOS. However, I could not get the code to run successfully because of some issues with OpenCV and other iOS difficulties.
- Firstly, the tracking modules used in the C++ code are not present in the regular iOS framework for OpenCV and must be included via OpenCV-Contrib. However, because there is not an existing iOS framework, I needed to create a framework that included OpenCV and OpenCV Contrib.
- Second, even with a working framework, there are issues in the code resulting from different requirements needed for this like accessing an iPhone camera which is done differently than on my laptop.
- TLDR
 - Tried to integrate the OpenCV/C++ code into our app
 - Needed to create a framework to include the required modules
 - Found that the code will need to be altered to work properly on iOS

- **Casey Gehling:**

- Experimented with C++ integrations on our mobile device, primarily with iOS. Succeeded in building a minified OpenCV framework instance for iOS as other team members have, doing research on the proper and most efficient way to integrate our video streaming solution with Dart. May involve writing minimal swift code as a middleman between our C++ and Dart, but I still have more research to do.
- Continued screen developments and general UI design (past pitches screen, settings screen)
- Did some repository cleanup – we had extraneous OS related files littered throughout the repository causing conflicts in some merge requests, so took time to remove those.
- **Ethan Gruening**
 - This week, I focused on creating CMake dependency downloaders that use installation tools dependent on the platform the user is running.
 - Additionally, I started a C++ guide for setting up the camera in the ‘optimal’ setting to remove the variability in the conversion of vertical pixel distance to feet. There has been some issues with setting up the views for fullscreen or overlaid effects such as the max/min lines and text.
- **Josh Hyde**
 - This week I spent a lot of time into the actual app development side specifically for android. I did a lot of debugging and fixing issues I had with the current design of it and made sure that it was able to at least run. I also developed through android studio a rudimentary google pixel phone that can be ran as an emulator to allow the use of testing the functionality of the android code.
 - I also did a limited amount of research into other potential height detection models that are used already in other designs. I found some other alternatives and additional add-ons that could be used within our design such as the use of MiDaS (Monocular Depth Estimation) which can maybe detect the height of the objects a lot better and more consistently, and potentially the use of the SORT (Simple Online and Realtime Tracking) tracking algorithm to better track the object throughout multiple frames.
- **Cameron Mesman**
 - I continued researching ways to run c++ on iOS. I ended up running into the same OpenCV errors that the rest of the team is running into, but once those get cleared up, we will be full go on the iOS front.

- I also did some looking into the Android implementation but ran into issues with the same building error with OpenCV.

- **Pending issues**

Our major issues this week involved the dependencies of OpenCV and installation methods for Android and iOS. Currently we are working on different CMake, brew, and vcpkg techniques to install libraries depending on the platform, and working through error logs. Currently this issue is blocking our development process and usage of OpenCV in our Flutter application and tracking models in C++ programs.

- **Individual contributions**

<u>NAME</u>	<u>Individual Contributions</u> <i>(Quick list of contributions. This should be short.)</i>	<u>Hours this week</u>	<u>HOURS cumulative</u>
Andrew Vick	OpenCV integration into iOS	6	54
Casey Gehling	Screen development, C++ -> Dart design practice research, repository hygiene	6	51
Sullivan Fair	C++/Flutter integration, creating OpenCV framework	6	52
Josh Hyde	Android research for app, Height detection research	7	50
Ethan Gruening	Data collection, Python to C++ translation, multithreading research.	5	59
Cameron Mesman	Researching C++ implementation on iOS	6	40

- **Plans for the upcoming week**

- Andrew Vick
 - Get our C++ code running on a phone
 - Optimize our object detection and tracking code
- Casey Gehling
 - Screen development

- Continue to work on C++/Dart pipeline
- Ethan Gruening
 - Continue setup guide in C++
 - Continue CMake Installation
- Josh hyde
 - Get a somewhat or fully working android app that can run what is already on the IOS side.
 - Develop a more working solution to the current app
- Sullivan Fair
 - Continue to try and get the C++/OpenCV code to run in our Flutter app
 - Develop the front end more if the code can be successfully integrated
- Cameron Mesman
 - Continue trying to find ways to move forward with app implementation even with OpenCV bug
 - Join in with developing flutter screens