***EE/CprE/SE 491 SEMESTER 2 WEEKLY REPORT 1***

***End of 491 – 1/29/2025***

***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***

o **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. Our guided setup is more informative, visually pleasing, and takes 3 minutes to complete. OpenCV is available for Android devices and has been tested with an ImageProcessor class to use OpenCV’s functionality during a Flutter run.

o **Past week accomplishments**

* **Andrew Vick:**
  + This week, I began working on integrating the entirety of our C++ tracking code into our Flutter app. This has involved refactoring how we start the C++ code and the data being processed and handled.
* **Sullivan Fair:**
  + This week, I began tackling the full integration of our C++ code into our Flutter app. At the time of this report, I have not been able to get the code successfully integrated, but we should have the necessary components created for it to run on iOS. This includes a compiled framework used for OpenCV and OpenCv-Contrib. The next step should be to compile the code into a dynamic library which will allow Flutter to be able to use imported C++ code used for tracking.
  + I also helped the team clean up our repository and removed unnecessary branches from last semester, which should help us keep better tabs on our upcoming development.
  + TLDR
    - Started full OpenCV\_Flutter integration
    - Helped clean up the team repository
* **Casey Gehling:**
  + This week, I took time to clean up my local repository and ensure all necessary changes were reflected in our shared repository.
  + Researched potential updates we would need to make to the project due to software updates that occurred over winter break.
  + Ensured gitlab repository contained the most current of my branches and that all necessary functionality was properly merged into main.
* **Ethan Gruening**
  + This week, I began working on renovating our Flutter application and providing new features that are much more user-friendly.
    - A home screen with navigational buttons to instructions, setup, settings, and a shortcut to the camera
    - Setup guided process
      * An instruction page detailing how mathematically our program runs and the supplies needed to run the program.
      * A form page to get the max, min, and reference height.
      * An instruction page for setting up the camera mount and field calibration.
      * A page automatically collects a series of images with audio countdown cues.
      * A page to mark the field points.
  + Integrating my changes within GitHub’s main branch, I additionally helped shrink our GitHub branches. After a team discussion, I created GitHub Issues to help structure our team’s goals for next week’s integration of Flutter with OpenCV.
  + Working on Integrating OpenCV into Flutter, I was able to install OpenCV’s Android SDK and link it to the Gradle build. The CMake documents route our C++ code into the CMake build.
  + Testing the functionality for Android’s OpenCV, I created an image\_processor.dart file that interacts with test C++ code that uses OpenCV to take in an image from Flutter, turn it into a grayscale image, and then return it back to Flutter. This was fully functional.
* **Josh Hyde**
  + Since last semester, I have done quite a bit of work on making a function that can implement the illegal pitch functionality that we want for our project design. I have a function that is currently made that needs to actually fit well and work with the rest of our project. However, it is not quite fully functional/conversible with what we already have, and it still needs to be improved and integrated more effectively.
  + I also have done some work in making sure that our current app functionality that we have is able to work on my Android phone, and not be too slow or unusable. I got it working to run on my phone as an app, and it shows that what we have is able to work on a phone, which is the end goal of our project.
* **Cameron Mesman**
  + This week we met to go over what to do next for the project. I started working on implementing multithreading and asynchronous functions in our code. At my current point, I’m still working on understanding how asynchronous functions work in Flutter and how to implement them properly with some initial testing.
  + I also was a part of cleaning up the repository and planning out the next steps and roles for the next few weeks.

o **Pending issues**

o **Individual contributions**

| **NAME** | **Individual Contributions**  *(Quick list of contributions. This should be short.)* | **Hours this**  **week** | **HOURS**  **cumulative** |
| --- | --- | --- | --- |
| Andrew Vick | OpenCV integration into Flutter | 4 | 61 |
| Casey Gehling | Clean up, research | 6 | 65 |
| Sullivan Fair | Started C++/Flutter integration, cleaned up repo | 8 | 64 |
| Josh Hyde | Illegal pitch code work, android app running on my phone | 7 | 64 |
| Ethan Gruening | New Flutter Screens, OpenCV in Android | 20 | 87 |
| Cameron Mesman | Researching C++ implementation on iOS | 8 | 54 |

**Plans for the upcoming week**

* Andrew Vick
  + Full integration with flutter
  + Optimize calling of YOLO
* Casey Gehling
  + Finalize OpenCV with Flutter
  + Optimize Dart FFI
  + Documentation and testing
* Ethan Gruening
  + Integrate the image processor into the CameraScreen.
  + Work on building OpenCV for iOS through Flutter.
* Josh hyde
  + Fully integrate or at least partially integrate the illegal pitch code into what we have
  + Add or put in some work into creating a max and min height lines within our current camera app design visually.
* Sullivan Fair
  + Continue integration between Flutter and OpenCV code
  + Begin to test the tracking algorithm once integrated
* Cameron Mesman
  + Implement asynchronous functions into our flutter code
  + Test the performance of the updated code with the performance of the non-asynchronous code