

**EE/CprE/SE 491 WEEKLY REPORT 03**

**Start Date – End Date**

**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 continued developing and testing new methods of detecting and tracking a softball in flight along with determining how far the ball is to give us a reference to perform height calculations. We also brainstormed what functionlites we would like out app to have to interface with our device.

- **Past week accomplishments**

- **Andrew Vick:**

- This week, I looked into what benefits we can get by calibrating our cameras and what the margin of error is when trying to calculate the distance to an object with just one camera.
- I also looked into different solutions to get more accurate depth information so our calculations can be more accurate. If we decide to do this all on a phone we may be able to use the built in sensors but I can not currently find libraries to interface with a phones lidar, or dual camera capabilities using react native.
- Also got a script working that detected only moving objects which can come in handy when verifying if the ball is in flight.

- **Sullivan Fair:**

- This week, I created a basic softball detection script using OpenCV and Python. I could not get the script to track the height, but I could track the distance of the ball from the camera. The distance displayed was  $\pm 2$  feet off. I don't know at this point if that is consistent enough for us to simply account for it. We also discussed ideas on how we are going to approach our solution and decided to build a react app that can be used on ios and android.
- TLDR
  - Wrote a python script to detect a softball
  - Collectively decided to build a react app for our implementation

- **Casey Gehling:**

- This week, I continued work on our OpenCV implementation, researching ways to port over our Python solution to a possible C++ solution that would be compatible with both IOS and Android devices. As discussed in our weekly meeting, it may be necessary to implement our object detection/distance tracking solution within C++ for performance and compatibility reasons.
- Refreshed myself on some React Native skills. For our implementation, we've decided to use React Native to build a platform-fluid application. I have used React quite a bit in the past as well as have used Expo to build some React Native apps, however a refresher was useful to hone

in on those skills again.

- **Ethan Gruening**

- This week within Senior Design 491, I began collecting testing data at the ISU softball fields to train an object detection model using YOLO. I collected 35 softball pitches in various lighting conditions, heights, speeds, and camera positions. This will provide our YOLO configuration file to detect softballs precisely and yield the most accurate results.
- I also created the template for our Product Research lightning talk slideshow outlining the main points of our findings.
  - Included screenshots and descriptions of all similar products.
  - Included pros and cons of design decisions for the two main distribution of applications
- I have also researched React-Native and its ability to run C++ and Python files for iOS and Android operating systems. React-Native using Expo, our current model, does not support running other languages other than TypeScript. Removal of the Expo module would be needed to incorporate C++ or Python scripts in our design.

- **Josh Hyde**

- This week I finally put in the work to get a working OpenCV application that could use images as inputs and be able to detect an object using an image. It is very buggy but it can occasionally detect the object. It is only working for the most part with images and not videos but I wanted to get at least a better understanding of how to get a system that can be able to consistently identify a specific object within an image.
- Additionally, I looked into photogrammetry. I put in work in trying to find how to potentially use photogrammetry to find the height of an object within an image or a video related to our project. The main problem seemingly arising from this project is determining the height of the ball efficiently, quickly, and accurately enough. Photogrammetry seems like the main solution to this problem so trying to better understand it and applying it to our project seems like it could benefit the problem in many different ways.

- **Cameron Mesman**

- This week, I researched ways different ways to identify the height of an object. Specifically, I looked into ways to do this with just an image. From what I found, this is technically possible but can be difficult and often inaccurate. OpenCV has some tools to help with these kinds of problems but not exactly what we're looking for. Some of the tools can get the distance to an object as others in the group have looked into, but none of the tools give us the distance between two

objects (the ball and the ground in our case). The best way I found to achieve what we want would require a lot of calibration which could cause problems if the camera is not always placed in the same exact spot. Because of this, it would probably be good to continue with a different method.

- I also researched techniques used by the MLB to track pitches. Although this was interesting to study, it didn't provide much help since their budget is much larger than ours. They are able to set up a series of cameras around the stadium for maximum accuracy and coverage. This is not feasible for us.
- I refreshed myself on some react knowledge since we will likely be using that going forward, and it's been a while since I've ever done anything with it.

○ **Pending issues**

We will need to work on converting our detection and tracking logic over to C++, and find a framework to develop an Android and IOS app that can run C++ code on the phone. We are also still running into issue with getting accurate distance data and then using that to find the height of the ball.

○ **Individual contributions**

<b><u>NAME</u></b>	<b><u>Individual Contributions</u></b> <i>(Quick list of contributions. This should be short.)</i>	<b><u>Hours this week</u></b>	<b><u>HOURS cumulative</u></b>
Andrew Vick	OpenCV testing, Object detection, photogrammetry research	6	18
Casey Gehling	Python -> C++ Research, React Native refresher	6	19
Sullivan Fair	Product research, prototype scripts	6	18
Josh Hyde	OpenCV work, Photogrammetry/height work	6	18

Ethan Gruening	Sketchnotes, PocketRadar research, team contract, team coordination	7	19
Cameron Mesman	Object height/OpenCV research, product research, react refresher	6	6

○ **Plans for the upcoming week**

- Andrew Vick
  - Continue working with OpenCV
  - Test the margin of error when using one camera to calculate depth
  - Port my code from python to C++
- Casey Gehling
  - Start on app implementation – decide whether we want to follow a frontend/backend team structure as in 309, or follow some other pattern.
  - Work on porting OpenCV solution to C++
- Ethan Gruening
  - Project research in app development frameworks with C++ integration.
  - Project research in the modularity and design of the mobile application.
- Josh hyde
  - I want to get a better working OpenCV application that more consistently works
  - I want to look more into photogrammetry as a possible solution for our height issue
- Sullivan Fair
  - Continue to develop my script to try and calculate the height of the ball
  - Do some research into the app side and determine what features we may want to include
  - Look into different implementations of my script, likely using C++
- Cameron Mesman
  - Continue researching options for how we can get the softball's height.
  - Continue react refresher and possibly start on initial react app