PITCH PERFECT

Senior Design Fall 24’ MAY 25-49

Client: Dr. Fila

Advisers: Dr.Fila & Dr. Jones

Kolby Moorman: Lead Frontend Designer

Sam Skaar: Coordination and Documentation Lead

Cael Schreier: Bookkeeper and Code Review

Drew Kinneer: Lead System Designer

Kyle Nachiengane: Lead Testing Engineer

Contact:

Team Email: sdmay25-49@iastate.edu

Team Website: https://sdmay25-49.sd.ece.iastate.edu/

Revised: 4/16/2025

Executive Summary

Pitch Perfect is a project designed to detect illegal pitches in slow-pitch softball. In slow-pitch softball, an illegal pitch is when a ball is thrown from the pitcher's mound to home plate by a pitcher, and that pitch either doesn't reach a minimum height of six feet or goes above a maximum height of ten to twelve feet. Currently, these pitches are called by an umpire, who also calls if a pitch is a ball or a strike and if a player is safe or out at each base. The reason for our device to detect illegal pitches is that players often disagree with illegal pitch calls made by an umpire. Visually determining six feet, ten feet, or if a ball is within that zone, can be very subjective and lead to lots of arguments throughout a game. Additionally, pitches that are below six feet are dangerous to pitchers, as they are easily hit right back at the pitcher. These are the main issues that Pitch Perfect aims to solve.

Our design requires us to identify an illegal pitch within the accuracy of one softball diameter (3.81 inches). Imagine an umpire eyeballing the height of a pitch, our program should have no more variance on calls than what a human umpire would give. We also must signal an illegal pitch to the batter, pitcher, and umpire when one occurs, and this signal must be as fast or faster than when an umpire would naturally make the call. Finally, we are required not to have our device interfere with the game, so it must be placed outside the field of play.

Our design approach is to use a phone camera to view a softball game in real-time, track the softball throughout a pitch, determine the height of the softball throughout the arc, and, if the ball is detected to be illegal, output an "illegal" sound for players to hear. We are using the OpenCV library to track the softball because of its quick processing time and the amount of information available about it online. To migrate our application to mobile devices, we are using the Flutter framework on an Android application. For hardware, the phone will either be mounted to the outside of a game fence or set up on a tripod outside of the fence. Both options are valid, and the user can decide how to mount the phone. The user will connect the device to a speaker and go through a short calibration process. Once the system has been set up, the program will run for the remainder of the game, indicating pitch legality during that time. Additional features like replay last pitch, quick menu to adjust heights, and show statistics about a game, such as pitch counts and legal percentage, will be tracked as well.

Our final design has met all of our requirements except being faster than an umpire on low pitches. Some implications of our design are that the umpire/user will have to update the advanced color calibration settings when lighting conditions change. Users will also have to use a battery pack in order to keep the application running. Our implementation uses more color tracking compared to object tracking, so something close in color to the softball can be mistakenly tracked as the softball, for example, dandelions. Some next steps to take are to test more with an umpire, focus on tracking an object based on movement rather than color, and optimize the app to run more efficiently.

Learning Summary

Development Standards & Practices Used

- ISO 5725-1:2023: Accuracy (trueness and precision) of measurement methods and results
- ISO 9241-210:2019: Ergonomics of human-system interaction
- ISO/IEC 25010:2023 Systems and software engineering- systems and software quality requirements and evaluation
- ICS 17.020: Metrology and measurement in general
- ICS 17450-1:2011: Model for geometrical specification and verification
- IEEE 1448a-1996: Standard for information technology- software life cycle processes

Summary of Requirements

- Detect illegal pitches within 3.8 inches of accuracy
- Be placed outside of the field of play during a game
- Alert players of illegal pitches at least as fast as an umpire
- Not put a large additional cost on softball leagues to use

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- COMS 2270
- COMS 2280
- COMS 3090
- COMS 3190
- ENGL 3140
- SP CM 2120
- PHYS 2310

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

- OpenCV development
- Python development
- Working with clients
- Mobile App. Development
- Cross-platform development (Flutter)

Table of Contents

List of Figures/Tables/Symbols/Definitions

1. Introduction

1.1. Problem Statement

Slow Pitch Softball is primarily a recreational sport, a chance for a team to have a good time, be active, and do it in a competitive environment without the stress of professional sports. That said, there are still rules that the players must follow. One such rule plagues pitchers, batters, and umpires everywhere: illegal pitches. All pitches must reach a minimum height, usually 6 feet, and not exceed a maximum height of around 10-12 feet, depending on the league. If a pitch does not meet these requirements, it is considered "illegal," and the batter does not have to swing at it.

Human error leads to more complications than the simple parameters may let you believe. An umpire standing behind home plate is the one who must call an illegal pitch. If he calls a pitch too late, a batter may have already begun to swing and be frustrated that they didn't have to swing at the pitch. If he inaccurately calls an illegal pitch, the pitcher will be frustrated with a fine pitch, suddenly becoming a free ball for the batter. Frustration about illegal pitches causes frustration amongst players who don't need to be there in a sport like Slow Pitch Softball. This is the crux of our problem.

We were tasked with finding a better way to call these illegal pitches. We have created an external device to detect a pitch's height accurately and quickly signal if it is illegal. Cutting out the subjectiveness of an umpire in illegal calls (and allowing them to direct their focus on the rest of the game) while making a call just as fast, if not faster, than one will make illegal pitches easier to deal with for everyone involved. We use a phone camera and speaker system, pointed at the pitcher, to track and call an illegal pitch.

1.2. Intended Users

Our application has three different users. These users consist of slow-pitch pitchers, hitters, and the umpires for the slow-pitch games. The pitchers can benefit from this application as they can fine-tune their pitches to the perfect height. The batters will benefit from this application as they will be informed that it is illegal and that they do not have to swing in plenty of time. Lastly, the umpires, typically umpiring as a side job, will benefit from this application because they will not have the pressure of being perfect at calling illegal pitches based on their eyesight and will be able to rely on the application itself to do it.

The pitchers who play slow pitch need this system to ease consistent illegal pitch calls. Right now, the pitcher relies on the umpire to call it early enough in the pitch's life cycle so that the batter knows they don't need to swing. Although it may sound like only a benefit for the batter, the batter knowing they don't have to swing allows for a safer environment for the pitcher as there is a smaller chance that the ball gets hit right back at the pitcher.

The batters that play slow pitch need this system, so they know when an illegal pitch is present, and they aren't required to swing. In slow-pitch softball, there is an unwritten rule not to hit the ball right back at the pitcher. If the batter has two strikes and the last pitch they see is illegal but not called, there is more of a chance the ball gets hit right up the middle, where players from both teams may be upset. This application will also allow the batter to be more specific on pitches they are swinging at so that they can be selective on when to swing and when not to.

The umpires who work during the slow pitch games need this application so that they are required to do less and can focus on watching the ball in play and the bases. The umpires benefit from this the most because they will have less weight on their shoulders in being consistent throughout the game, knowing the exact height of a pitch. They won't have to hear complaints from the teams about calling illegal too late or not calling illegal at all, as the application will take that aspect out of their job, and no one can complain about it.

2. Requirements, Constraints, And Standards

2.1. Requirements & Constraints

Functional Requirements:

- Our device must be able to detect an illegal pitch lower than six feet (specification)
- Our device must be able to detect an illegal pitch higher than 10-12 feet (specification)
- Our device must be able to detect illegal pitches within 3.8 inches of accuracy (specification, constraint)
- Our device must alert players as fast or faster than an umpire (specification)
- An audible signal must be made upon detection of an illegal pitch (specification)
    - The signal must be loud enough for the batter, pitcher, and umpire to hear
- The device must be usable in a location where it does not interfere with the game (physical, constraint)
- The device must be usable in a location where it is not in danger from the game (physical, constraint)
- The device must be portable (physical, constraint)
- The device must have a fast enough camera to process a pitch, greater than 30fps (resource)
- The camera must have a resolution high enough to accurately process each frame, at least 1080p (resource)

User Interface Requirements:

- Our device should be simple for any softball player to set up and use (user experiential)
- Our device will have clear instructions for users to input needed measurements for calibration (user experiential)
- Our device should be cheap enough to not:
    - A) Effect league fees if leagues provide the device (no more than $10)
    - B) Be a large expense on a player or team (economic, constraint)
- Our interface should record and replay pitches (user experiential)
- Our interface should clearly indicate how a pitch was illegal (too high or too low) (user experiential)

2.2. Engineering Standards

Engineering standards are important because they give us, as engineers, a baseline for minimum requirements to ensure safety and health. These standards play a vital role in making sure we, as users or consumers, are given a quality product and not something that will just break shortly over time or not be able to perform certain actions it claims. Through these standards, engineers are given a minimum benchmark to ensure all of these standards and qualifications are met.

2.3. Engineering Standards in our Project

ICS 17.020: Metrology and measurement in general

This standard covers a variety of aspects in regards to measurements. The standard serves as a guideline for engineers to ensure consistency, accuracy, and reliability in terms of measurements.

ICS 17450-1:2011: Model for geometrical specification and verification

This standard is part of the GPS standards that cover the geometric features of products. This standard ensures that measurements in a geometrical way are consistent and accurate throughout different procedures.

IEEE 1448a-1996: Standard for information technology- software life cycle processes

This standard provides a framework for the software development and management processes, creating software processes. It establishes a common ground for definitions and processes that help software engineering teams follow best practices for large-scale software projects and development.

2.4. Engineering Standards Relevance in our Project

ICS 17.020 provides relevance to our slowpitch softball project because we are creating an application that relies solely on measurements. If the measurements are inconsistent or inaccurate, our application is useless and will give an unfair playing field in a softball game.

ICS 17450-1:2011 provides relevance to our slowpitch softball project because, as we are gathering our measurements, we will be relying on geometrical measurements to ensure the measurements of the height are consistent and accurate.

IEE 1448a-1996 provides relevance to our slowpitch softball project because our project's end deliverable is an application that users can use to track the height of a softball to determine whether or not a pitch is illegal. Throughout this process of creating our application, we will rely on the software life cycle process in order to deliver the best version of our product.

2.5. Engineering Standards that could be considered in our Project

ISO 5725-1:2023: Accuracy(trueness and precision) of measurement methods and results

ISO 9241-210:2019: Ergonomics of human-system interaction

ISO/IEC 25010:2023 Systems and software engineering- systems and software quality requirements and evaluation

2.6. Modifications to meet these standards

One modification we made is improving our device calibration and how we detect height. We weren't quite at a level of measurement accuracy that would comply with this engineering standard (ICS 1745-1:2011), so we did more work to get there. Specifically, we made sure to have multiple known variables during our testing to make sure our min and max line measurements were accurately drawn, before doing full game testing.

We also worked on modifying our user interface to provide users with many features to accurately detect illegal pitches. To comply with standard IEE 1448a-1996, our application provides clear instructions on how to operate and maintain our product. We include clarifications for the user to follow in order to make the life cycle process of our application easier to understand and adaptable for many users.

3 Project Plan

3.1 Project Management/Tracking Procedures

For Pitch Perfect, we used a hybrid style process where we adapted features from both the agile and waterfall management styles. We used waterfall elements like a fixed timeline and sequential progression. We chose to adapt these elements from waterfall because with our app, we cannot do things like track the height of a ball if we can't track the ball itself. We did, however, use Agile elements for the user interface as these tasks can be split up into a bunch of different smaller tasks and allow for a more precise workflow. Our team also had weekly client meetings, which were a lot like agile. Throughout the year, we created many tasks to delegate so that no two people were working on the same issue, and we could enhance the productivity of our group.

3.2 Task Decomposition

We have broken down each functional component of our project into a separate task that must be solved, as well as a flow of how each of these components will flow together in our final design:

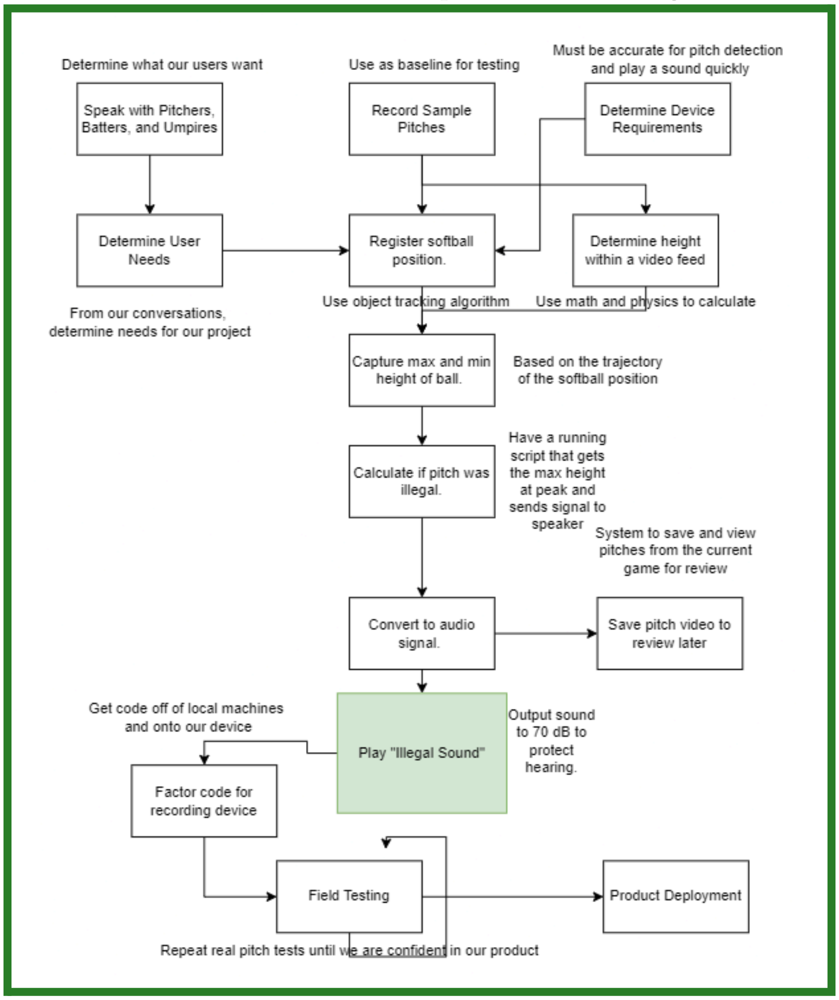

Figure 3.2.1

Many of our early tasks were completed non sequentially, in a sprint style, such as our player interviews, pitch recordings, and client requirements. As we moved forward with our project, many of our tasks fell into a waterfall style, as we cannot determine the height of a ball until we register its position, and we cannot determine an illegal pitch until we have registered its height, and so on and so forth. During our early development, we broke our team up to solve each sprint-like task, but as we got to the point where each task relied solely on an earlier task, we would break off our team to solve external tasks, such as our application development:

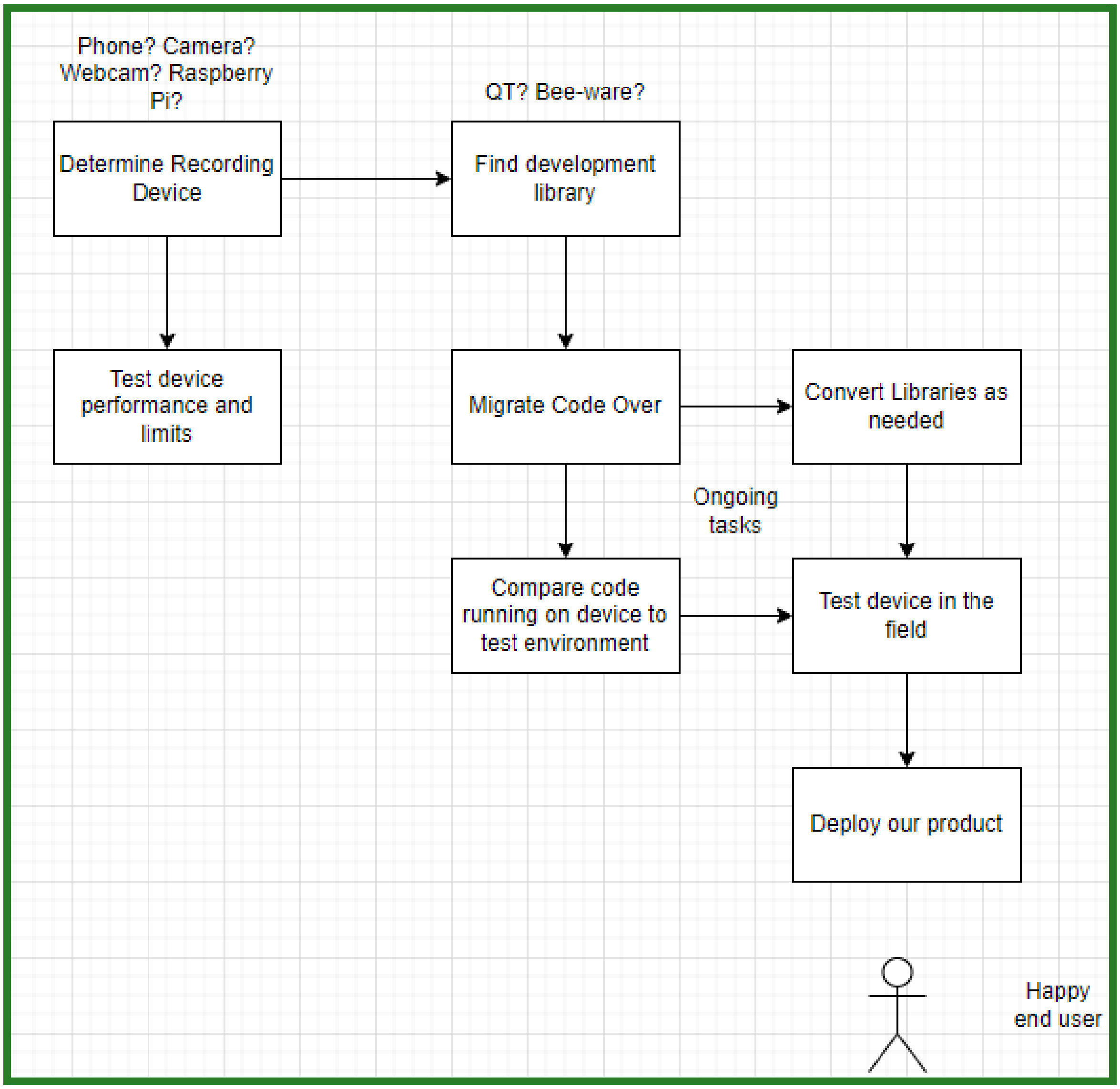

Figure 3.2.2

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Many of our milestones in our early development and prototyping process are defined by working features of our device, object detection, and tracking, tracking a pitch through its arc, tracking the height of a ball, and so on and so forth. Thus, we have broken down our developmental milestones, followed by our product accuracy milestones:

Development and Prototyping Milestones:

The device can identify and track a softball in ideal conditions.

The device can follow the arc of a softball throughout its pitch.

The device can track the height of a softball with an accuracy of 1 ft.

The device can play a sound within 0.5 seconds of detecting an illegal pitch.

The device can work with all unknown elements inputted by a user.

A working prototype to detect and call an illegal pitch in a live setting.

The device can accurately run for 10 consecutive pitches.

Product Accuracy Milestones:

The device can identify and track a softball in any lighting condition.

The device ignores all elements of the frame except for the arc of the pitch.

The device can accurately track the height of a softball throughout its arc within 4 in.

The device works with only the necessary elements inputted by a user (camera height).

The device can run for an entire softball game, for two hours.

Our device is complete.

3.4 Project Timeline/Schedule

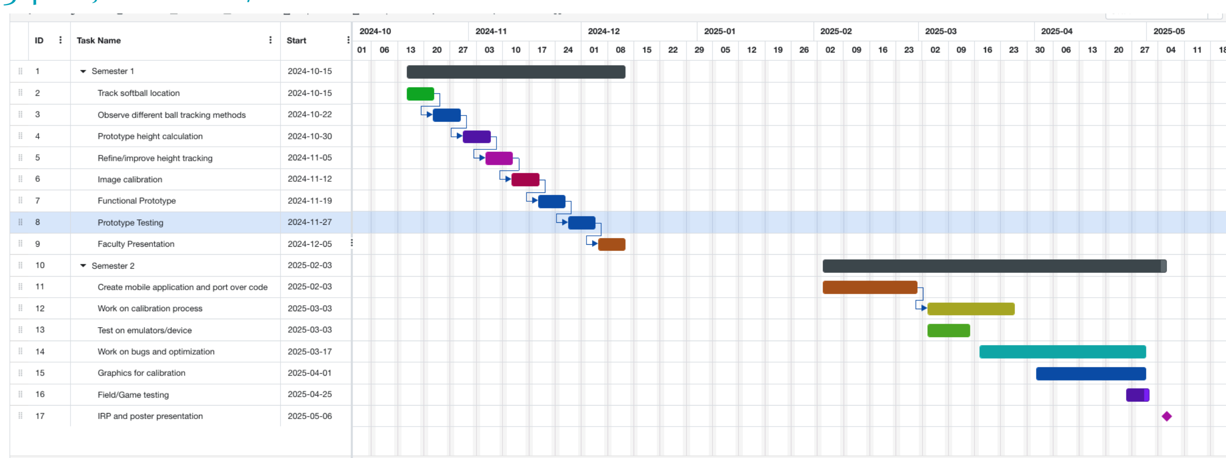

Figure 3.4

As a group, we broke our project down into various tasks and assigned these tasks throughout the months that we were working on Pitch Perfect. These tasks included brainstorming, initial designs, height and pitch tracking, device calibration, screen flow, and many more. This decomposition and scheduling of tasks helped us stay on track and work towards a final product one step at a time. Adjustments were made throughout the process as things progressed faster or slower than we originally expected.

3.5 Risks and Risk Management/Mitigation

A potential risk that comes from any live recording device is performance. How can you capture enough data in real-time and process it quickly enough to satisfy users, while still being accurate enough to satisfy our client's requirements? The more resolution our camera takes information in, the more data we will have to process, and the more our framerate could suffer, and vice versa.

We mitigated this risk by using a device with a natively powerful camera (with an Android camera) to give us options on our recording resolution, so we can scale to whatever level of performance is needed. We also simplified our algorithms and use threading on our processes to keep our processing time low.

Another risk to the integrity of our device is the accuracy of our measurements. We wanted our device to be able to measure to the accuracy of one softball diameter (4 inches) when detecting illegal pitches. If something causes that accuracy not to be met, our device will not be seen as reliable and risks not being used by our users or satisfying our clients.

We mitigated this risk by testing using as many known measurements as possible to make sure our device is meeting its accuracy requirements. Having recordings where a ball height is known, and seeing if our device can measure to that ball height, will allow us to avoid the guesswork of our accuracy during our development process. We also mitigated this risk by introducing more user-inputted fields to help us deal with unknown parameters. Additionally, we implemented a replay feature so users can go back and view the legality of the previous pitch.

Another risk comes from the nature of having a device in a sports setting: keeping the device safe. We do not want our camera to be hit with a flying softball and have our project break, but we also want it to be close enough that it can capture all the needed elements of the game.

We mitigated the risk of our device being damaged by placing it outside the fence of a given softball field. This will protect it from any foul balls, while also still allowing it to record through the links of a fence.

3.6 Personnel Effort Requirements

Task	Estimated Prep Hours	Estimated Work (Code) Hours	Explanation
Prototype Height Calculation	5	8	Doing the math to determine the height of a ball given the most known parameters + having the program compute that math.
Further Develop Height Tracking	0	10	Tweaking the code to have the most accurate height. This includes detection for pitches thrown off-center.
Image Calibration	10	20	Determining the process of setting up the camera and making the application do most of that work. This includes

			making the user experience as light as possible.
Functional Prototype	20	20	Prep time includes moving the application to a Phone development platform. Work will include changing any threading or weird Phone things, so basically a continuation of the code base transfer.
Prototype Testing	0	30	Make any changes needed to make the product as accurate as possible, even if it's on known heights and distances from the mound and plate.
Faculty Presentation	The semester	10 (each)	The semester is the preparation that we get for the presentation, but each individual can expect to work a significant number of hours to make the presentation as presentable as possible.

Table 1: Estimated work hours

Task	Prep Hours	Work (Code) Hours	Explanation
Prototype Height Calculation	5	8	This was done by the end of last semester, so no updates were made on work time.
Further Develop Height Tracking	5	15	The YOLO framework was less usable in an Android environment, so a pivot had to be made to a manual height tracking process. This required more

			research and implementaiton time.
Image Calibration	10	20	Getting the softball color calibrated correcly was a very in depth process, but took a similar amount of time to what we anticipated.
Functional Prototype	20	40	Moving our code over to Android and getting everything to work as we wanted to much, much more time than we anticipated as there were many more roadblocks and technical bugs to fix.
Prototype Testing	5	15	Time was reduced here because we didn't hae as much time as we anticipated to complete all of our testing.
Faculty Presentation	The semester	10 (each)	This was similar to what we expected.

Table 2: Actual Work Hours

3.7 Other Resource Requirements

To complete this project, we needed to acquire multiple resources. One is a phone fence mount to mount our device to the outside of a fence to keep it out of the field of play while being able to accurately track a softball. We also needed softballs to test our detection and height algorithms. And finally, we made a contraption with strings at 6ft and 10ft to help with calibration and testing. Some other resources that we could benefit from are a tripod and a speaker, as these give our users options for how they mount their phone and output the "illegal" sound.

4 Design

4.1.1 Broader Context

We designed this product for slow-pitch softball recreation leagues. They are all across the country and usually consist of working adults. Any league that decides to implement our product would benefit from it, but the most affected players would be pitchers, batters, and umpires, as they are the three main actors during a pitch. The main need addressed by our project is the desire for a reduction in arguments over illegal pitches, as they frustrate umpires and players. Having an external device to make these calls can reduce frustration with umpires.

Area	**Description**	**Examples**
Public health, safety, and welfare	When a slow-pitch softball league uses our product, conflicts about illegal pitches will be reduced and leading to a smoother game experience. It will also help keep pitchers safe from dangerous low pitches.	Arguments with umpires are reduced, low illegal pitches are called consistently, and users must be outdoors to use the application, increasing activity.
Global, cultural, and social	Slow-pitch softball leagues can use this product in any league or location, which is important because slow-pitch softball is a nationwide recreation activity.	A league in Iowa vs a league in California would get the same experience when using our application.
Environmental	To use our project effectively, we recommend that some sort of mount device be used. This would require extra purchases, which could lead to extra plastic being used for a tripod, but we aim to keep it minimal.	An increase in tripod and fence mount purchases could lead to an increase in waste from our users if thrown away.
Economic	We plan to keep our product as cheap as possible. Our application will be free for mobile users to use, and the external equipment to mount a phone can be found at or under $20. The largest expense may come to a league if they decide to use a league-purchased phone to run our product or a high-end Bluetooth speaker.	Our application will be free, the external equipment will remain cheap to users or leagues, and a league phone would be the largest expense, though it is not a requirement to use the project.

Table 3

4.1.2 Prior Work/Solutions

Throughout research for products similar to our slowpitch application, we found a very widely known app that is able to be purchased for tennis and pickleball. Although they are not tracking the height of the ball, their main focus is on tracking the exact location of where a ball lands in order to tell whether it is within the boundaries or not of the given court. Although there is no way to access this code, it did allow us to gather some ideas for a user interface as well as let us know that this project is able to be accomplished [1].

Another similar product that we found was Flightscope. Flightscope is an application that can be used to track pitching and hitting data like horizontal launch, spin rate, and pitch velocity, among others. This application is meant for both hitters and pitchers and allows them to review the analysis of their performance in different metrics. [2].

As far as object tracking, there are various applications that are used today like tracking people. As a team, we did a few different methods after doing some thorough research on object tracking. This ranged from MOSSE to KCF tracker. MOSSE was one method that we strongly looked at after reading that it is OpenCV's fastest object-tracking method [3].

4.1.3 Technical Complexity

Our project contains multiple components and considerations, as well as challenging requirements within those components. Within our application, we must take in video input, and process the video in real-time while:

- Identifying the softball within a pitch
- Determining the position and height of that softball
- Determining if the motion of that softball is a pitch
- Determining if that softball pitch is legal or illegal

Factors like background video elements, game times, lighting conditions, camera perspective, camera location, and processing time cause these seemingly simple tasks to become much more difficult. Edge cases are everywhere, and techniques that work for certain situations can fail in others. Finding a way to consistently identify a ball in any lighting conditions and accurately track it. Determining height and positions requires math and physics to position accurately. With all of this, we also want a simple user interface and calibration process for our users, so we want to be able to accomplish all of this with only one camera. All of this has caused significant challenges to overcome within our project.

4.2 Design Exploration

4.2.1 Design Decisions

For our project, we designed three essential components for a successful product. These include an accurate system that can track the height of a softball, quick processing time that allows for an audio output the second it realizes the pitch is illegal, and an application that can be deployed from a phone that is seamless and easy for the user experience.

Accurate System - An accurate system is the most critical design decision we have to make in order to have success with our project. If the system is inaccurate, there will be no use for any of our users, as the human eye would be better and more accurate. The entire purpose of our system is to track the height of a softball, so without focusing on this detail, our project would be nonfunctional.

Quick processing time - Quick processing time is another important aspect of our project. If the processing time is too slow, our project is also useless because, as a user of our project, they need to be able to hear that the pitch is illegal the second it becomes illegal, or they will lack the need for the system.

Application for phone - An application is also an important aspect of our project to allow users to actually operate the system. The application will allow users an easy setup to get the system up and running, and allow for the other design aspects (Accurate System and Quick processing time) to work seamlessly together. This adds to the accessibility of our project.

4.2.2 Ideation

We came up with an application idea after talking to potential users of our project. The application would make an easy and robust way for users to use our system, and not require much time to set up. This also makes our pitch detection system available to anyone who wants to try it out whenever and wherever they are located without having to go and buy much special equipment.

A few of our other ideas included:

- Multiple camera systems set up around the outside of the ballpark. This would allow for easier integration and maybe even more accuracy to the exact millimeter of measurement.
- A sensor in the ground would alert the umpire that the pitch was illegal and allow for the umpire to be the one to call it "Illegal".
- A player-focused detection where they had a clip-on camera and a gyro wristband in order to track whether the pitch was illegal and alert the users.
- An in-play component where there was a chip inside of the ball that would alert the umpire that the pitch was illegal.
- One singular camera would then alert the umpire that the pitch was illegal via vibration or light, so he could make the illegal call.

4.2.3 Decision-Making and Trade-Off

We chose to do a round-table discussion that evaluated each proposed option for our pitch detection system. During this discussion, we identified the pros and cons of each idea, considering factors such as ease of user setup, cost, accuracy, durability, scalability, and overall feasibility. We also heavily favored our clients' wishes, along with the initial interview we conducted with players.

While we considered using tools such as weighted decision matrices, we ultimately opted not to rely on formal tools. Instead, we continued our conversation and maintained an open dialogue until a clear direction emerged. This iterative discussion allowed for deeper consideration of practical trade-offs and ensured that all perspectives were heard, leading us to a well-rounded decision that best meets the needs of our intended users.

Moving forward toward the deployment of our application, we may choose to incorporate more structured tools, such as a weighted decision matrix, to help us further evaluate and weigh our options. This would provide a more systematic approach to refining our solution, ensuring that we continue to prioritize factors such as user needs, feasibility, cost, and scalability as we move from concept to implementation. By doing so, we can maintain alignment with our goals while making data-driven decisions that optimize the effectiveness and usability of our pitch detection system.

Pros		Cons	
Argument	Weight	Argument	Weight
Is precise to within one ball diameter	3	Is not precise to < a ball diameter	1
It is easy to set up hardware (hand the phone up)	4	Is not easy to calibrate	3
Addresses the user needs by calling pitches relatively cl	3	Could be affected by Sun	3
Other solutions do not exist	3	Could be too technically complex to process quickly	3
More consistent than an umpire	3	More expensive than an umpire	3

Figure 4.2.3

4.3 Final Design

4.3.1 Overview

Our design uses a camera to watch a slow-pitch softball pitch as it happens. As it records the pitch, it determines the height of the pitch and determines if the max height of the pitch is legal (between 6-12 feet). If a pitch is determined to be "illegal" or outside of these bounds, it will play a sound to alert the users that an illegal pitch was thrown. This will then be repeated for the softball game.

Our first component is the camera system, which is from a phone setup along the fence of a softball diamond. The phone will be out of play, on the outside of the fence, but still able to record each pitch. A setup process of identifying the home plate, the pitcher's mound, and showing a softball to allow for accuracy in our device is required.

The next component is our internal processing. This is the bulk of our "code" in this project, which determines the location, height, and legality of a pitch. We use libraries like "OpenCV" to help with this process, and then send our decision to our output.

Our final component is our output device. This could either be on our phone or an external Bluetooth speaker. This simply takes our processing decision and either plays or doesn't play a sound. The full design is outlined in Figure 4.3.1 on the next page.

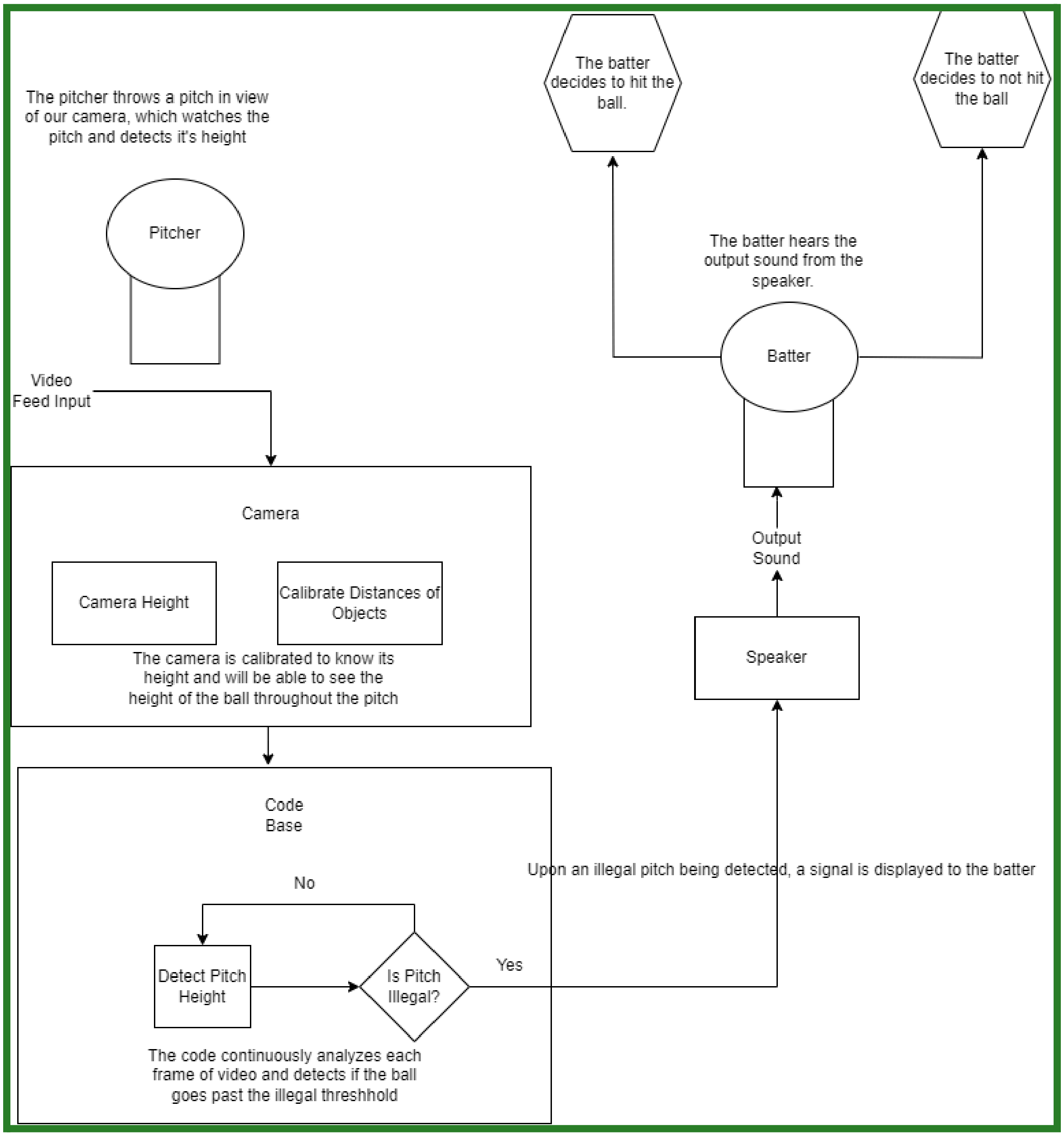

Figure 4.3.1

4.3.2 Detailed Design and Visual(s)

Our design includes a phone mounted to the outer fence of a softball field. The phone will have our Perfect Pitch app installed on it. Our app is made with the Flutter framework, integrating OpenCV to perform real-time image processing for height determination, basic design pictured in figure 4.3.2.2. The camera must then be calibrated by camera height, distance from the camera to the plate, and mount to ensure precise measurements and connected to a wireless speaker wire. With the phone set up, the application will run a script that plays audio when the ball goes over or under a specified height range. The phone will be set up in one of the green boxes in Figure 4.3.2.1.

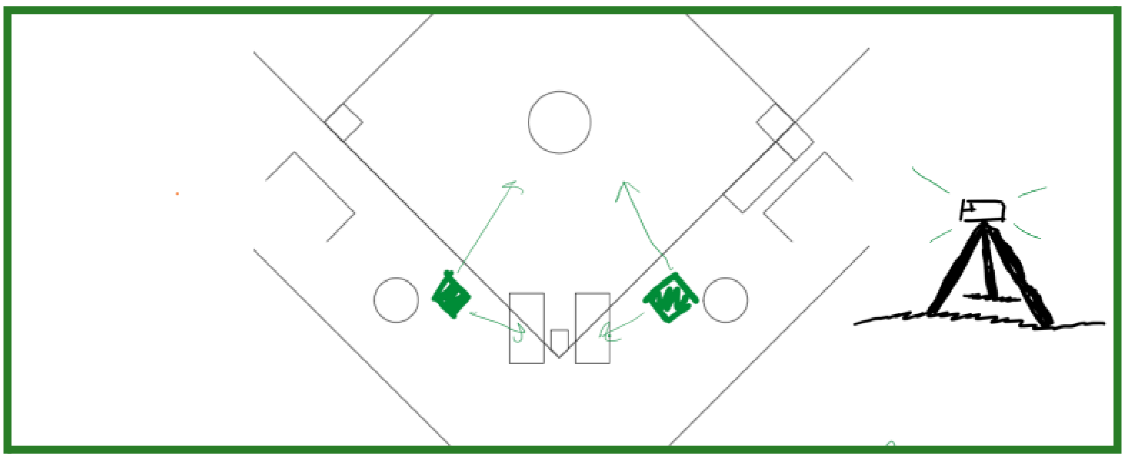

Figure 4.3.2.1

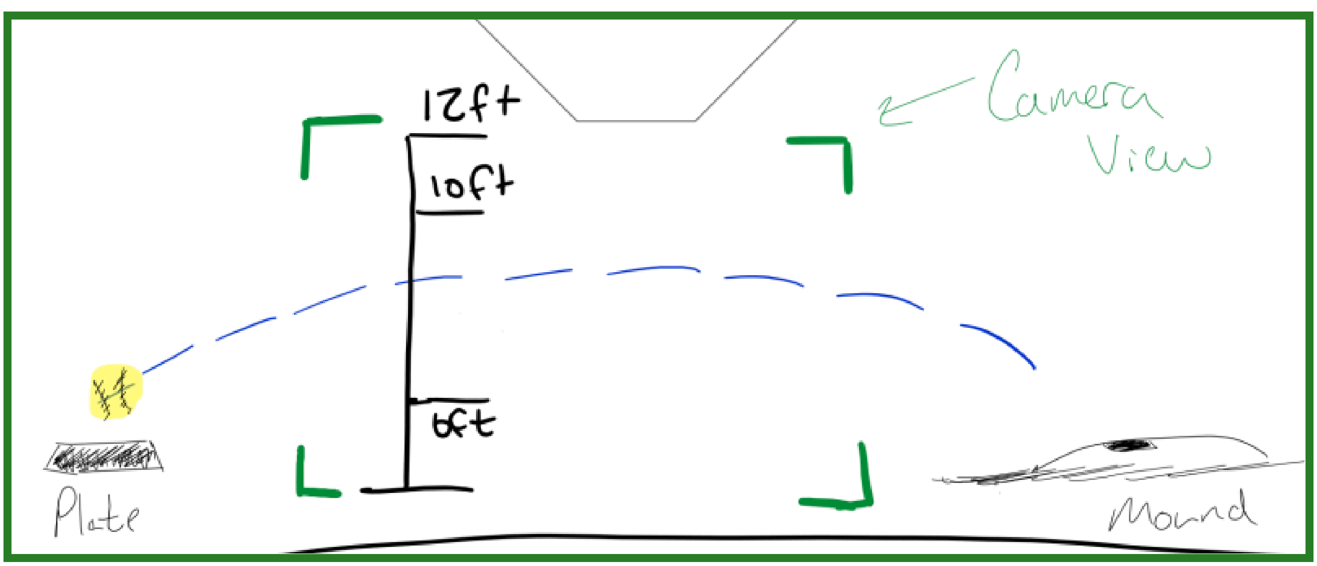

Figure 4.3.2.2

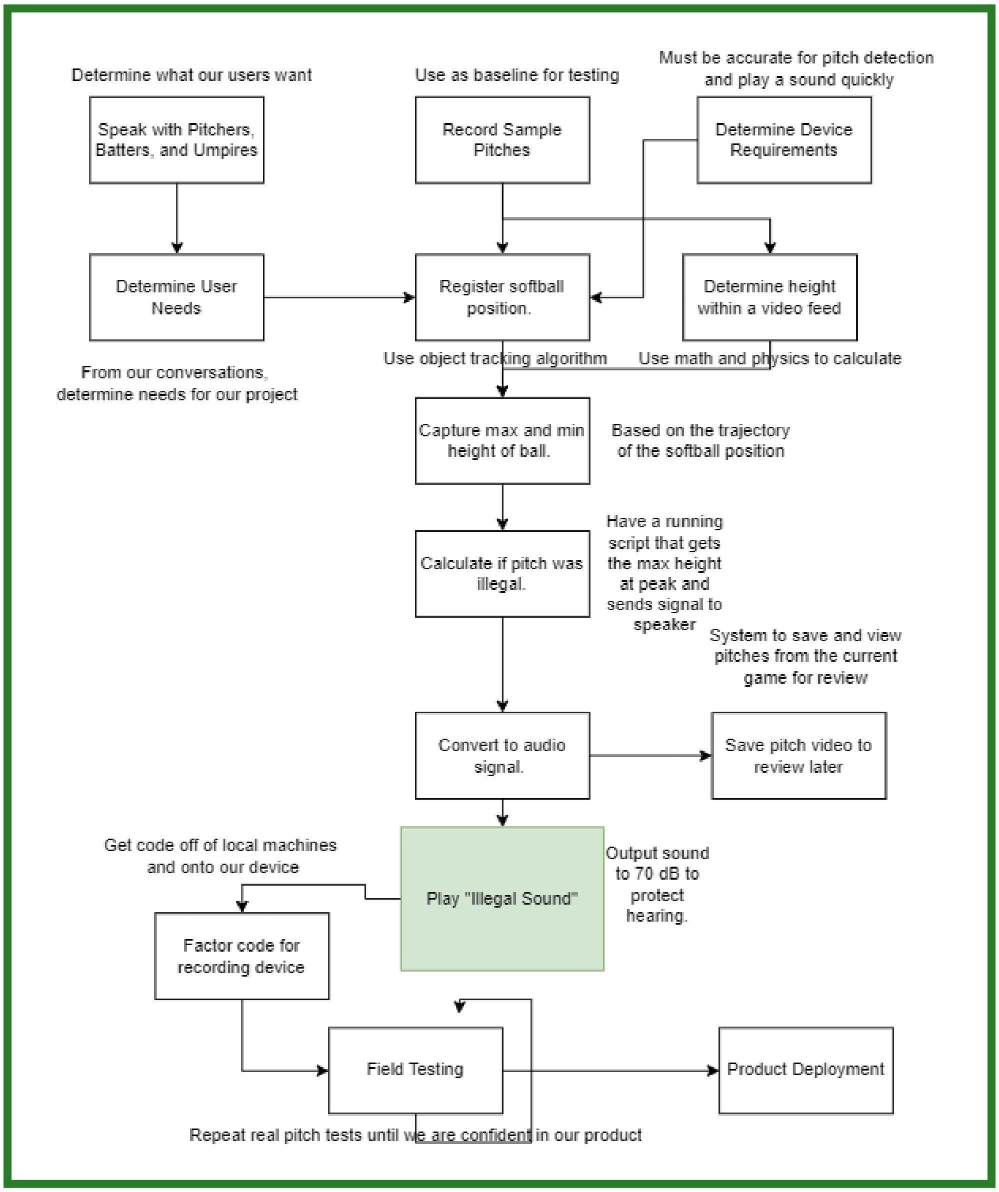

Figure 4.3.2.3

4.3.3 Functionality

The users of the game should be hands-off with the device after set-up. Pitchers will throw the ball, and given their pitches' performance, an audible sound will be played so the batter can react to an illegal pitch. Recalibration options are available to users if they have to switch which side of the fence the phone is mounted on, the height requirements of a pitch, or if the lighting conditions change and cause a need for new color calibration. Additionally, users have the ability to replay the last pitch to review if a call from our device was correct.

4.3.4 Areas of Challenge

We encountered many challenges during the development process of Pitch Perfect. Some were very simple to solve, but others took up a lot of our development time. The largest challenges we faced are outlined here:

1. Pitch detection:
    a. Tracking a softball is very difficult. Our approach focused on tracking the color, but this caused many problems, as other "neon yellow" objects would often be in frame
    b. We worked around this by writing an algorithm to detect pitches. We would first look at the pitcher's mound for a ball, then watch as it moved towards home plate in an arc, and only focus on that area of the frame at a time.
    c. This alleviated many of our issues, but caused many complications in development and took much of our second semester to complete.
2. Mobile development:
    a. We were able to get prototypes working in Python very quickly and easily in our first semester. However, once we fully transitioned to a mobile environment, we encountered many issues. Importing libraries, getting different components to work together, and balancing device performance were all major headaches for our team. We got passed them, but it took a lot more time than originally anticipated.
3. Calibration:
    a. To be able to follow a pitch, we first needed our device to know how to do so. We were able to solve our height tracking issues pretty early by using people as a reference height, but calibrating a ball's color was more difficult.
    b. Due to how long softball games run, lighting conditions change and are inconsistent. This made it hard for us to track a consistent softball color. We used a workaround, "advanced mode," to have users manually input more data about their ball to track, specifically its hue, saturation, and value (see section 9.1), but this was difficult to implement and added more complexity to our user experience.

All in all, we solved a lot of issues with this project, but not without significant time or design changes to do so. These challenges defined our second-semester experience, but led to a project that we are all proud of.

4.4 Technology Considerations

We used an Android phone for our application and camera. There are multiple strengths and weaknesses that come along with using a phone for object detection and tracking.

Strengths:

1. Allows users of all kinds the ability to use our product, as there is no needed extra hardware
2. Users won't have to worry about purchasing anything besides a mount for the system to run as expected
3. Seamless setup for the user

4. Android phones have a native quality camera

Weaknesses:

1. The quality of the video may not be as good as a high-quality camera

2. Developing for a mobile architecture could add more bloat versus, for example, a script written for a Raspberry Pi
3. Each phone is different, so we need to accommodate for that in our tracking algorithm
4. Not all users have an Android phone compared to an iPhone

5 Testing

Testing with recordings: We currently have a bunch of video recordings that we took that consist of live gameplay as well as testing where we knew all controllable variables. In terms of controllable variables, we have gathered two 10-foot PVC pipes and attached strings at 6 feet and 10 feet high so that we know the range of a legal pitch.

Testing live: This semester, after getting our height tracking down to an appropriate range, we went to the ISU Intramural softball fields and tested the operation of our application to gauge whether or not our application was quick enough for live feedback and its accuracy with PVC pipes we knew were 10ft tall.

Testing time amount: One constraint we haven't been able to cover yet is the idea of a potential memory issue or code crash, as a game could span up to a few hours. After gathering a consistent height-tracking application, we will want to attend a game and start the application, and observe any issues.

Testing withan umpire: When we were ready, we took our device to a slow-pitch recreational game to test it against an umpire in a real game environment.

5.1 Unit Testing

We aren't able to use Unit Tests at this point because inputs and outputs are not virtual and cannot be simulated in terms of exact height. To compensate for this, we ran many physical tests of clicking on various components to test their interactions, simulating "pitches" in our room to test logic and tracking, and a lot of physical tests outdoors.

5.2 Interface Testing

Our application consists of three primary user interface screens, each serving a distinct purpose within the user experience. The central **Home Screen** acts as the main hub, offering navigation to both the **About Screen** and **Stats Screen**, while also providing a prominent button to launch the calibration and tracking features. The **About Screen**, located to the left of the Home Screen, displays background information on the project, including our team, our advisors, and the origin and goals of the application. On the opposite side, the **Stats Screen** provides a summary of all past games, showcasing recorded statistics in a readable and scrollable list format.

At this stage of development, formal interface testing was limited to **User Testing**. Our team has spent considerable time using the application internally, iterating on feedback and ensuring that basic screen transitions, layout consistency, and button functionality meet expectations. Through repeated use, we have verified that the navigation between screens is stable and intuitive for users familiar with mobile applications.

Given the constraints of our timeline and the importance of perfecting the core tracking functionality, we have chosen not to allocate resources toward writing automated UI tests for screen transitions or layout validation. These elements, while important, are not the critical components of our project's value proposition. By focusing our testing efforts on the calibration and pitch-tracking pipeline — the most complex and mission-critical part of the application — we were able to maximize our impact and deliver the most meaningful results for our end users.

5.3 Integration Testing

Integration testing for our application focuses on validating that key components work together as expected across functional boundaries. Specifically, our testing efforts are centered on the communication between the UI, tracking logic, and data persistence layers. Since our application's core functionality revolves around real-time pitch tracking and post-game stat analysis, it is crucial that these subsystems interact reliably and without data loss or inconsistency.

The main areas targeted for integration testing include:

- **Calibration and Tracking Flow**: Verifying that when the user initiates tracking from the Home Screen, the calibration process runs smoothly and transitions into live pitch detection without crashing or freezing. This test ensures that the camera module, pitch height detection algorithm, and user interface are properly synchronized.

- **Stat Recording and Retrieval**: Ensuring that data collected during tracking sessions is correctly stored and then accurately displayed on the Stats Screen. This requires integration between the tracking module, local storage/database, and the stats display UI component.

- **Navigation and State Persistence**: Checking that navigating between screens during or after a session does not result in data being lost or app state being corrupted. This includes handling interruptions such as screen rotations or app minimization.

Our current approach relies heavily on **ad-hoc validation**, where developers confirm the functionality of multi-component workflows during regular usage and debugging. This method was effective for catching obvious integration issues, especially in high-traffic flows like pitch tracking and stat display.

Given our limited development timeline and the technical challenges involved in implementing robust integration tests for real-time camera and tracking functionality, we've prioritized manual validation and functional reliability over formal automation. That said, we acknowledge the importance of integration testing — especially for ensuring long-term maintainability and catching edge cases — and did all we could to substitute this testing method.

In short, our confidence in the integration of major features stems from hands-on testing and iterative development, rather than a formal suite of integration tests. A future team taking on the project may consider introducing automated integration tests in their testing strategy.

5.4 System Testing

To test our entire system, we ran live pitch tests at a softball diamond with our device outside of the fence, as if it were an actual game. We will throw both legal and illegal pitches to ensure that our device is making correct calls and that all components mentioned in 5.3 are working as expected at full scale. This testing allowed us to verify both our calibration and pitch detection logic in a real environment, while also allowing us to make live changes and updates when things didn't work as expected.

5.5 Regression Testing

To ensure that new additions did not break app functionality, we made multiple "experiment" folders for our members to work in to ensure they are able to mess around with code without it affecting the main app. We also utilized Gitlab, which allows code branching features to ensure that we can verify new features as a group before anything is pushed to our main branch.

5.6 Acceptance Testing

To ensure our device met its functional requirements, we tested it against an umpire to make sure it is at least as consistent, accurate, and fast as an umpire. We also tested that the height tracking throughout a pitch is within 3.8 inches of accuracy, as that is our goal.

5.7 Results

Our results had mixed, but mostly positive success. In terms of our core functionality and requirements, we found:

1. Our device is able to accurately measure the height of a softball within 4 inches of accuracy.
2. Our device successfully outputs a sound when an illegal pitch is detected and is able to consistently detect these pitches.
3. Our device performs as well or better than an umpire
    a. We found that umpires are more lenient than our requirements would indicate, as they often didn't call pitches "high illegal" unless they reached about 13 feet. Our device can be adjusted to call this height as well.
4. Our device quickly outputs a sound once a pitch is detected.

However, we also found some limitations of our device through testing that we were unable to completely fix within our timeline:

1. Our device is prone to crashing after long periods of use.
    a. This could be from a memory issue, an issue with our testing phone, or an issue with how we deploy our application. It is too inconsistent to find a true cause, so we were unable to identify exactly why or when this happens.
2. Our device requires strict calibration
    a. This isn't a fault of the device's functionality, more its usability. It is difficult to track the height of the softball without the user manually inputting some settings, and that limits the simplicity of our application compared to the rest of the process.

3. Color issues
    a. Since we focus mostly on using the color of a softball to track a pitch, objects in frame with a similar "neon yellow" color would consistently throw off our algorithms. This wouldn't be much of an issue if players didn't often wear jerseys with neon yellow softballs on them. (This also makes our application a wonderful dandelion finder.)

6 Implementation

Our implementation for this project is within an Android application. We use the Flutter framework to create a set of homescreens that can be run on any device (Figure 6.1), and then run a native Android application that uses the OpenCV library to track a softball's height and position.

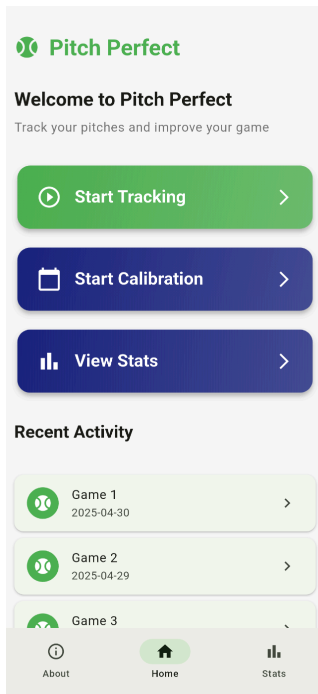

Figure 6.1

For softball detection, we use a color mask to search for the neon-yellow of a softball and track the movement of that color to determine where that ball is during a pitch. In Figure 6.2 below, you can see a ball being tracked and having its arc displayed, while also seeing the color mask used to identify the ball below it.

Figure 6.2

To track the height of our softball, we are using fixed height points at the pitcher's mound and home plate to determine a known height conversion for the start and the end of a pitch, and linearly interpolate between the two measurements to find the height of a ball throughout its arc. We use a line between the pitcher's mound and home plate as a mark of the ground for the ball. In Figure 6.3 below, we have a 10 ft pole being walked down our illegal boundary lines to ensure accuracy and consistency. Figure 6.4 shows this method running on our application.

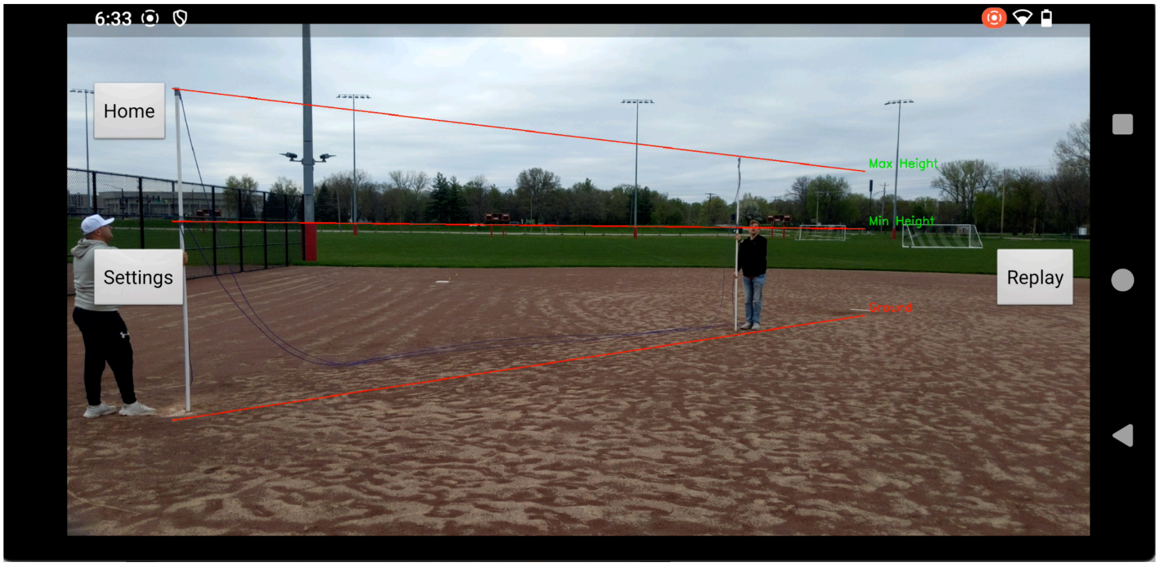

Figure 6.3

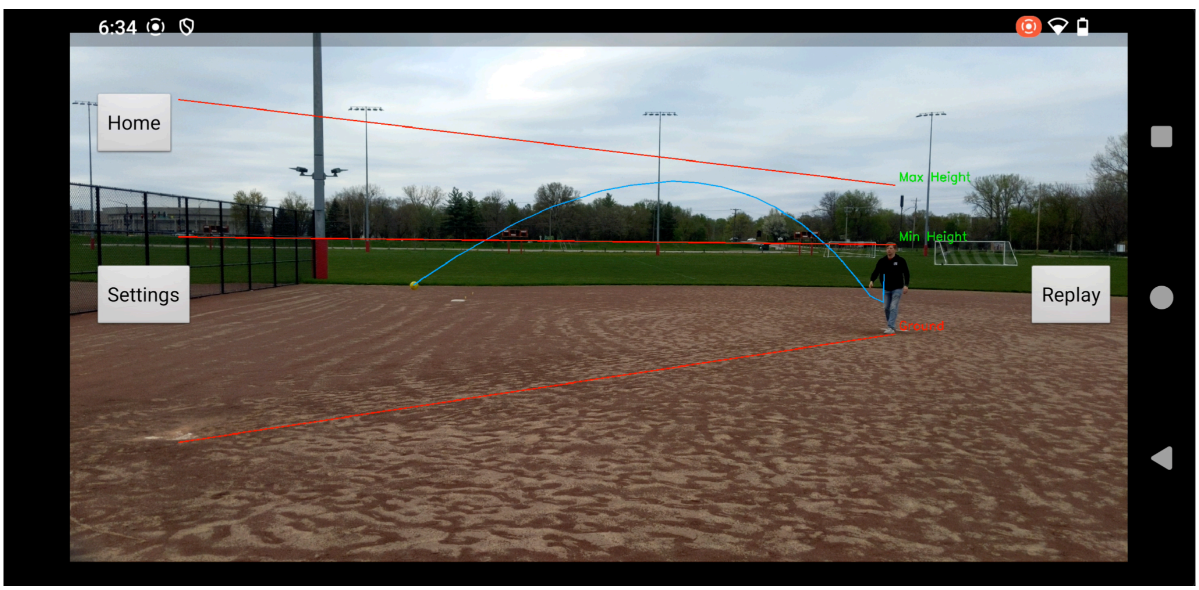

Figure 6.4

7 Ethics and Professional Responsibility

7.1 Areas of Professional Responsibility/Codes of Ethics

Area of Responsibility	**Definition**	**Relevant Items from SE Code of Ethics**	**How our team has interacted with area**
Work Competence	Ability to complete work quickly, correctly, and to a standard quality.	2.01: Providing service in their area of competence 3.01: Strive for a high quality	Our team has weekly meetings where we work on our project and expect a high level of quality from each other.
Financial Responsibility	Ensure products are given at a high value and reasonable price.	6.10: Avoid associations with conflicting businesses. 3.01: Strive for a high-quality and acceptable cost	Our team is considering the financial costs of leagues throughout our project and attempting to mitigate them.
Communication Honesty	Reporting work done honestly to clients, stakeholders, and advisors.	6.08: Take responsibility for detecting and reporting errors. 8.03: Produce well written documentation.	Our team communicates progress honestly in weekly reports and weekly client/advisor meetings.
Health, Safety, Well-Being	Minimizing health and safety risks for those involved in a system.	6.06: Obey all governing laws in work	Our project aims to reduce pitches that can pose a danger to pitchers, adding an element of safety to leagues.
Property Ownership	Respecting all personal property, ideas, and information of others.	5.09: Ensure a fair ownership of any software or process. 7.03: Credit fully the work of others.	Our team uses many external libraries and is making sure they are referenced and acknowledged in our code base.
Sustainability	Respecting the environment and natural resources.	3.15: Treat all maintenance with the same professionalism as development	Our team aims to reduce the necessary waste and carbon emissions on our project by requiring minimal external equipment.

Social Responsibility	Ensure products are a benefit to society.	4.04: Do not engage in deceptive practices	Our team is ensuring our project is accurate to allow it to be useful to slow-pitch leagues.

Table 4

Our team is performing well within the social responsibility area. We want our project to be useful to leagues and a net benefit to society, and to do that it is necessary that our project is intuitive and accurate. We are focusing greatly on these aspects during our development.

Our team could improve within the health, safety, and well-being area. During our testing process, we have been lax with our equipment treatment and could have caused damage to our equipment in the process, or to ourselves. We aim to focus more on this as we test our product in-game scenarios going forward.

7.2 Four Principles

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public Health, Safety, and Welfare	Our project relates to outdoor activities, which can help encourage physical activity among users	We aim to reduce conflict by having our device resolve illegal pitch arguments	Leagues have the choice of how much they wish to use our project	Our project is unbiased to different teams and leagues, it will be consistent throughout every game
Global, Cultural, and Social	Our project design is made to be easily accessible by many users	We hope our product can prevent disputes and conflicts among many slow-pitch games, not just in Iowa	Teams have a choice to rely on an umpire or use our project	Calls are consistent across teams and leagues
Environmental	Our app may cause extra plastic purchases for equipment	The equipment needed to run our app is minimal and does not cause much extra manufacturing.	Players have the choice of what phone our app is run on, as well as how our app is set up on a field	The purchases we recommend for our project do not need to be disposed of and can be reused many times.
Economic	Our project requires minimal external	We aim to not increase league costs for	Only one device is needed, so players/leagues	Only one team needs to provide the device for

	hardware purchases	slow-pitch softball	have the choice of who provides the device	both teams to use during a game

Table 5

We are greatly focusing on the Public Health, Safety, and Welfare/Nonmaleficence cell. A large reason that our project was proposed was to reduce arguing between players and umpires over illegal pitch calls. By having our device be unbiased and consistent, we really aim to reduce issues and disputes that show up.

An area we may be lacking in is Environmental/Beneficence. We do not plan on affecting the environment much with our project, but we do recommend the purchase of either a tripod or a fence mount for best use in our project. Because these are usually made of plastic, this could cause extra plastic purchases and extra pollution, but we aim to make this a very minimal part of our project.

These principles have remained mostly unchanged since we defined them, as our high-level design has remained mostly the same.

7.3 Virtues

Team Virtues:

- Fairness - *treating people justly, not letting your personal feelings bias your decisions about others* (viacharacter)
  - Our team believes in those getting what is earned from them. If you do something correctly, you should not be penalized for it, and someone else's bias shouldn't cause you to be.
  - If a pitcher throws a legal pitch, we believe it is fair to have it be called legal, not incorrectly called by an umpire.
  - If a pitcher throws an illegal pitch, it is fair to the batter to not have to swing at it.
  - We developed our project to be accurate and consistent, allowing for fairness in its calls.
- Determination - *A quality that makes you continue trying to do or achieve something that is difficult* (virtuesforlife)
  - Our team believes in working hard to try and accomplish a goal, and continuing to do so until that goal is met.
  - We modeled this virtue by continuing to solve problems we encountered until they were fixed, like making sure our height tracking is to our required accuracy, battling mobile environment challenges, and more.
- Flexibility - *Willing to change or to try different things* (virtuesforlife)
  - Our team understands that things don't always work and changes need to be made at times.
  - As a team, we tried different algorithms for implementing height tracking and softball detection as we find what does and doesn't work.
  - When team members have busy moments in their schedules, we understand and are flexible with their expectations during those times.

- We were flexible and understanding that our iPhone development had to be halted for the sake of a better overall product.

Individual Virtues:

- Cael Schreier
  - Demonstrated: Appreciations
    - This is important to me because I feel that when someone does something well, they deserve recognition for it.
    - I demonstrate this by making an effort to acknowledge what other members have done and thank them for the work they are putting in.
  - Grew to demonstrate: Cleanliness
    - In the first semester, I struggled a lot with keeping my code clean and documenting things.
    - While I am still far from perfect, I did work to make my code more readable, organized, and able for other team members to work off of.
- Samuel Skaar
  - Demonstrated: Inclusivity
    - This is important to me because I believe a functioning team needs all of its members to contribute. During certain stretches of a project, one member of a team may not be as effective as the others. It is still important to keep that person engaged in the project so that when the task changes to something they may be more proficient in, they're ready to go and don't feel left out or behind.
    - I demonstrated this when we were building our pole and string guides and purchasing materials at Lowe's. Some team members may maybe more effective in a coding environment than building things from the store. But bringing the team along so they see what we were working with was important when they used footage of that equipment to start our first real demonstration. It also helps with team bonding!
  - Artistry

        - I really like the creative elements of the project and making things look a step above what is required of the presentations. Typically, I would have expressed that interest in making even this document look nicer than it does. I just didn't manage my time effectively to make that happen this semester.
        - In the second semester I tried to make sure that our presentation of our product is something that could be seen by "investors." Not that we will have any watching us, but I think that type of creative feel and sales pitch is befitting the project.

- Kolby Moorman
    - Demonstrated: Open-mindedness
        - This is important to me because when working as a team, I find it very important to listen to others' ideas on top of my own as well as take their advice on different ways to go about things.
        - I demonstrated this by looking into multiple front-end frameworks that could be beneficial to our project rather than just going with one I felt most comfortable with.
    - Not yet Demonstrated: Organization
        - This is important to me because if I myself am not organized with my work others on my team may not be able to see what I am doing or how I am doing something.
        - I haven't been as organized as I would like to be as I have been going through multiple different front-end frameworks to try and find the perfect fit for our group. I have finally found one that can help me to be better organized when coming to group meetings to share the progress I have made.

- Drew Kinneer
    - Demonstrated: Adaptability
        - With the nature of this project what we are working on and how the approach we take is always changing. Adaptability is important to me so that as a team we can pivot to new potential ideas and not be narrow-minded in how we tackle this project

- I have shown adaptability by taking feedback from other members of the team and advisors and making adjustments to our overall approach for object tracking.

  - Grew to demonstrate: Thoroughness

    - During the first semester, in some aspects of motion tracking with OpenCV, more research could have been done to research more specialized techniques used to apply to our project. And I could have been more thorough in my research.

    - This semester more effort was put into research for things like deciding our app archetecture with Flutter. I was also very thorough with designing our pitch detection and accounting for multiple different scenarios that the system could expirience

- Kyle Nachiengane
  - Demonstrated: Perseverance

    - This virtue is important to me because it is crucial in this project. There are always challenges in any project, and I must stay focused and push through difficulties. I must make sure that problems are addressed and not overlooked.

    - I have shown perseverance by continuously trying to get a mobile application with OpenCV integrated, always trying different things despite many frameworks and online sources not emulating on my device.

  - Grew to demonstrate: Proactiveness

    - In the first semester, I had not been planning ahead on issues I may face and would start fresh when running into an issue.

    - I still need to be more proactive when it comes to issues in my code but I feel I did a lot better the second semester in planning for issues for things I thought would take longer and trying different approaches instead of starting over.

8 Conclusions

8.1 Summary of Progress

Our group has implemented a quality project to meet our client's specifications. We are able to track a pitch, determine its legality, and output a sound when an illegal pitch is detected. To accomplish this, we researched the current market, discussed with many softball players about what they would want in a device, prototyped in Python, developed an Android application, and tested in various softball conditions.

Along the way, we ran into many hurdles. We struggled in our prototyping phase to find consistent height and pitch tracking technologies and algorithms. Once we moved to a mobile environment, we struggled with getting all of our different libraries to integrate together. Once we got to the field, we struggled with making sure things that worked in our room worked on an actual softball field. These problems took time to solve, but our team was able to persevere through them.

As a team, we are very proud of the work we put into Pitch Perfect. We learned a lot about the software development process, and these skills will carry with us throughout our professional careers.

8.2 Value Provided

We have provided value to many slow-pitch softball players. The purpose of our project was to reduce arguments over illegal pitch calls. We provide a simple application that makes consistent, correct calls that will alleviate pressure from umpires making these calls. We were able to bring this to a softball game and show players the consistency of it to alleviate their concerns about accuracy as well.

Because it can be run on any Android phone, it is scalable to any slow-pitch league across the country. This application also has the ability to be updated in the future to better track pitches, allow easier calibration, and support other features like tracking balls and strikes of pitchers. This provides both current and future value to our users.

8.3 Next Steps

If this project were to be continued in the future, here are the main things we would recommend teams prioritize.

1. A functioning iPhone application: Many of the libraries used for this project were not supported fully by Apple's development environments, causing difficulties in development. Additionally, an Apple device, such as a MacBook, is required to do that development, which caused even more problems for our team. Despite all this, iPhones are a large portion of the mobile market, so porting our application over would greatly increase our user base.
2. Streamlining our color calibration: Advanced color calibration is a bit clunky by necessity, but a team with more time and resources could potentially find a more user-friendly way of getting these values.

3. Improving pitch edge-cases: We are confident in our application's accuracy. That said, it is difficult to test pitches that could potentially "break" our system due to an extreme curve, bad lighting conditions, or other hard-to-control factors. A team with a semester dedicated solely to these problems would likely be able to find bugs we couldn't.
4. Tracking balls and strikes: This was originally a stretch goal of our application, but it was clear we wouldn't have time to implement it. Adding a way to track balls and strikes, as well as illegal pitches, would give pitchers a lot of feedback to help improve their game.

9 References

[1] ACM Ethics, "Software engineering code," *ACM Ethics*, 2018. https://ethics.acm.org/code-of-ethics/software-engineering-code/

[2] CodeSavant, "Ball Detection Using OpenCV in Python," YouTube, https://youtu.be/RaCwLrKuS1w?si=VmYMcoLu-q4PcCe- (accessed Dec. 7, 2024).

[3] "Virtues List - Virtues For Life," *Virtues For Life*, 2010. https://www.virtuesforlife.com/virtues-list/

[4] E. Baccellieri, "How MLB Pitch Tracking Works: Behind Baseball's Complex System," *Sports Illustrated*, Nov. 15, 2023. https://www.si.com/mlb/2023/11/15/how-it-works-mlb-pitch-tracking-baseball-statcast

[5] A. Rosebrock, "OpenCV Object Tracking," *PyImageSearch*, Jul. 30, 2018. https://pyimagesearch.com/2018/07/30/opencv-object-tracking/

[6] "Main page," *FlightScope Softball*. https://softball.flightscope.com/

10 Appendices

Appendix 1 – Operation Manual

Introduction to Pitch Perfect

Pitch Perfect is an application designed to detect illegal pitches in slow-pitch softball. An illegal pitch is a pitch that doesn't meet a required minimum height or exceeds the maximum height. Pitch Perfect is designed to use your phone camera to track these pitches and output sounds when illegal pitches are detected.

Setup

Upon opening Pitch Perfect, you are greeted with our home screens (Figure 10.1.1). Here you can start our main application, see previous game pitch stats, or view information about the developers.

To get to these pages, you can either scroll left and right or click on the icons at the bottom of the screen. It is possible to start tracking a ball directly from this screen, but to be able to accurately track pitches, you will need to click on the "Start Calibration" button (Figure 10.1.2). This will take you to the calibration screen.

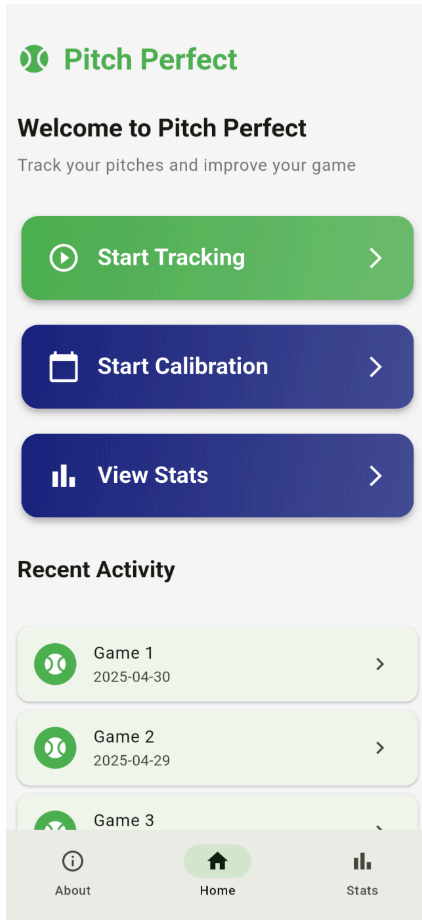

Figure 10.1.1

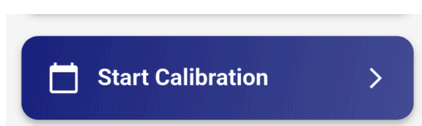

Figure 10.1.2

Upon clicking this button, you will be taken to our calibration setup. From here, you will help our system learn more about the specific softball color to track, as well as the height of your specific field. First, calibrate the color of a softball. You can do this by holding a ball in the green circle on screen (Figure 10.1.3) or by using the "advanced" mode to manually input the color of the ball using its hue, saturation, and value (Figure 10.1.4). We recommend this mode, as it makes tracking significantly more consistent. To aid with the complexity of this mode, we have provided presets for you based on various lighting conditions. Play around with these settings until only a white sphere appears where your softball is in frame (Figure 10.1.5).

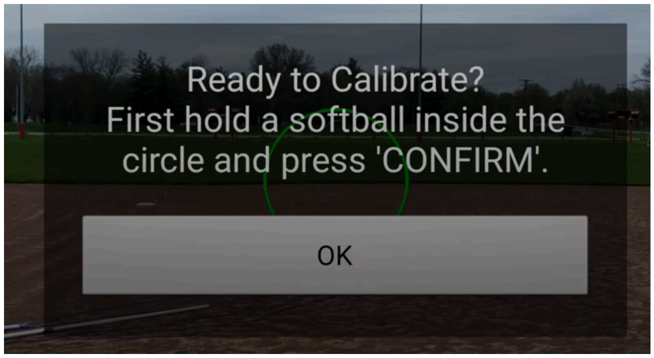

Figure 10.1.3

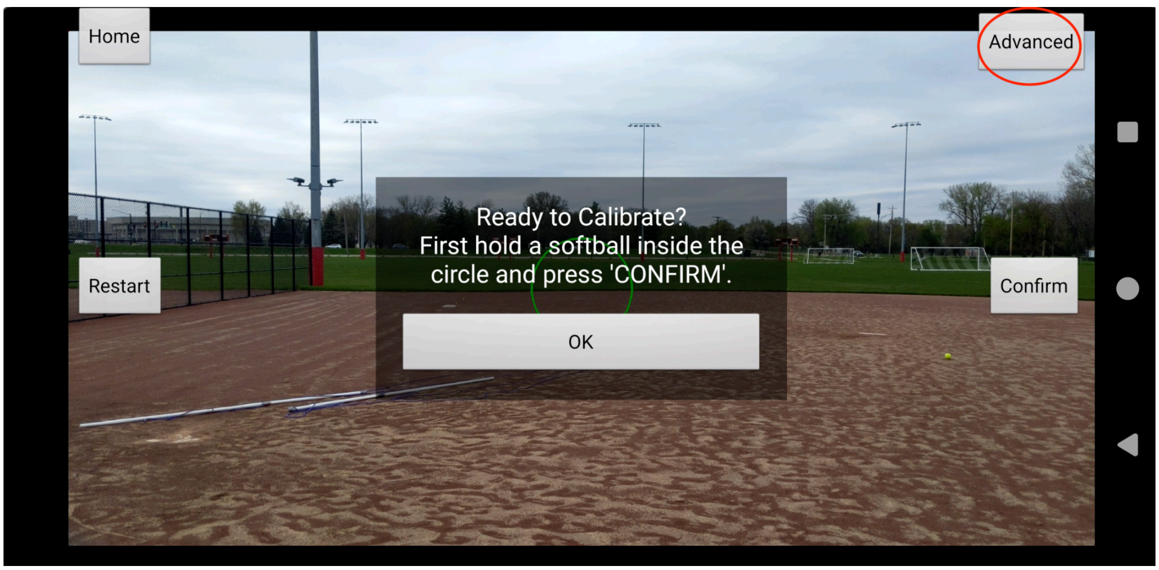

Figure 10.1.4

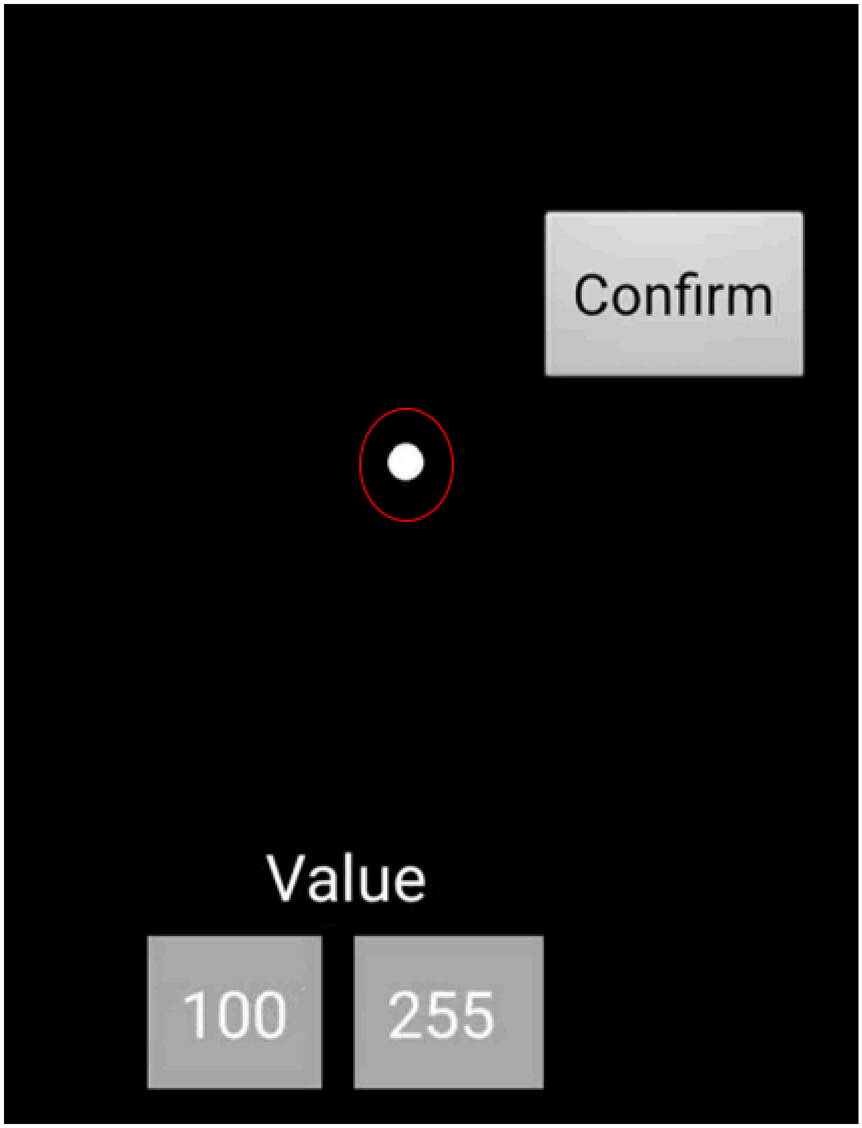

Figure 10.1.5

Next, you will start the height calibration portion. You will have someone stand at the Pitcher's Mound and Home Plate for this process, and first, you will input the height of this person (it can be you). After this, you must have your phone remain stationary. We recommend that you mount it to a fence, but a tripod will also work. During this process, it is important to make sure that the Home Plate and the Pitcher's Mound are in frame. We have provided some "target zones" to try and have these locations be within (Figure 10.1.6).

Figure 10.1.7

From here, we will use whoever you input the height of as a reference height. If this is another person, have them go stand at the Pitcher's Mound first. Once they are standing, tap the screen to drag and create a box around them. Tap on the inside of the box to adjust its location. Once it fits tightly around the user (Figure 10.1.7), press confirm, and have them go to Home Plate to repeat this process. If you are doing this by yourself, go to the Pitcher's Mound and wait for a sound. This captures a picture of you to draw a box on later. Once the first sound plays, go to Home Plate and wait for the second sound. After that you can return to the phone to draw your two boxes.

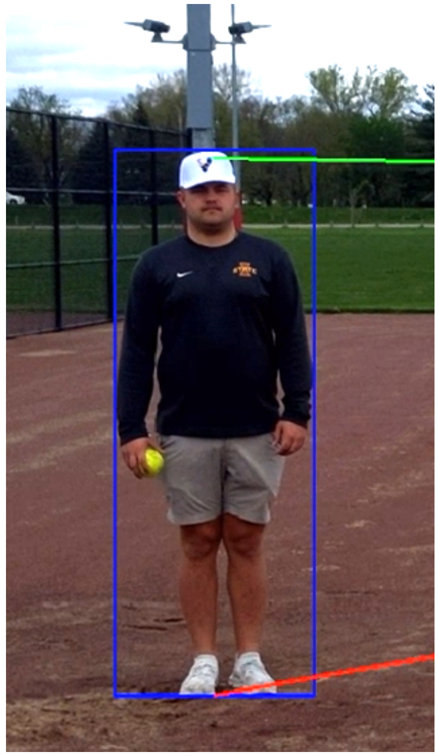

Figure 10.1.7

After the boxes are complete, the calibration process is also complete. If you want to redo any aspect of the process, click on the "restart" button (Figure 10.1.8) to restart either the entire calibration process or just a specific portion of it. Press confrim to move onto the main pitch tracking.

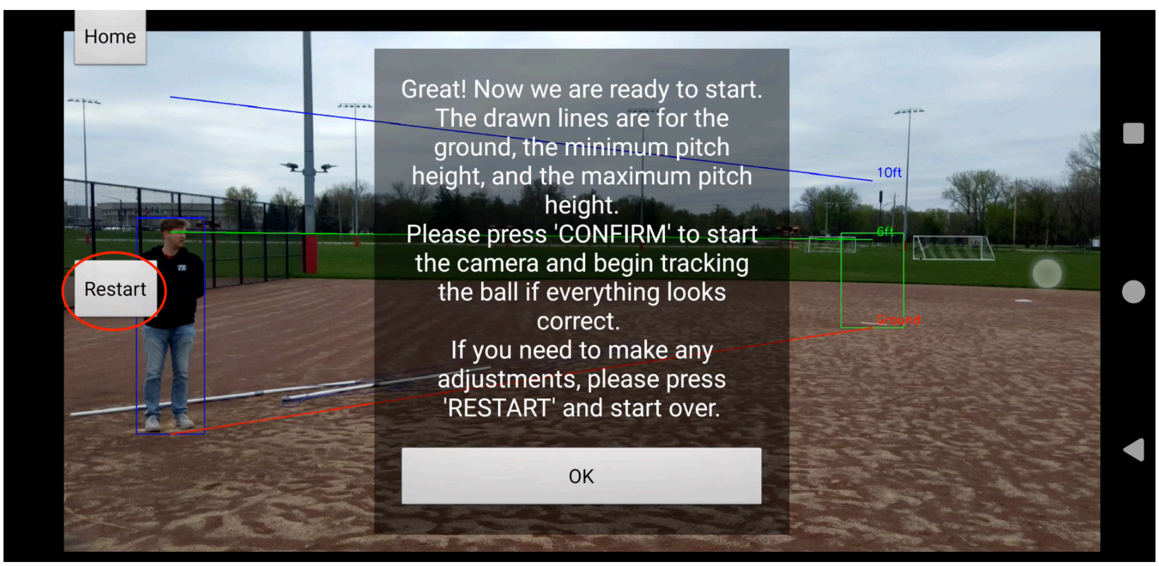

Figure 10.1.8

Tracking

Once you are within the main pitch tracking, you are ready to play some slow pitch softball! A pitch thrown from the Pitcher's Mound to Home Plate will be tracked and have a sound play if we deem its height illegal. On this screen, you are able to adjust the pitch height requirements and direction using our settings menu (Figure 10.1.9), and can replay the last pitch using the replay button (Figure 10.1.10). If something has gone wrong, you can return to the calibration menu as well.

Enjoy your game of softball with Pitch Perfect! Once you are finished, click the home button to return to our home page and have your pitch stats saved for future reference. Use this tool to help improve your slow pitch games, to practice your pitching techniques, or just to track some softballs!

Additional Notes

While you are able to directly start tracking from the home screen, we *highly* recommend going through the calibration process before each game. Without it, height will not be considered, and you will only be able to loosely follow a softball. Calibration *is necessary* for full functionality of Pitch Perfect.

We understand that the advanced color calibration can be a complicated process. To help you better understand how to use this mode, here is some more information about the fields:

Hue: The type of color (e.g. yellow, green, pink). The range is from 0 to 360, which represents degrees around the color wheel. 20 to 40 is generally the color of a neon yellow softball.

Saturation: The intensity of the color, 0 is gray, and 255 is full vibrant color

Value: The brightness of the color, 0 is black, and 255 is full white

Appendix 2 – alternative/initial version of design

One of our first considerations was using extra hardware on a pitcher's wrist to track the exit velocity and angle of a pitch to determine the ball's height (Figure 10.2.1). We decided against this pretty early due to the lack of user involvement and inconsistency from edge cases in our discussions.

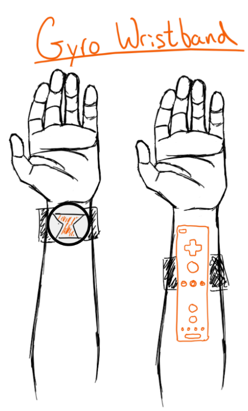

Figure 10.2.1

We also considered using a sensor on the ground between the Pitcher's Mound and Home Plate. The idea was that it would use infrared (or a similar technology) to sense the height of a ball at where we anticipate the maximum height to be (Figure 10.2.2). We decided against this for a multitude of reasons, but the main one was that we had a design requirement for our device to be outside of the field of play for its and players' safety.

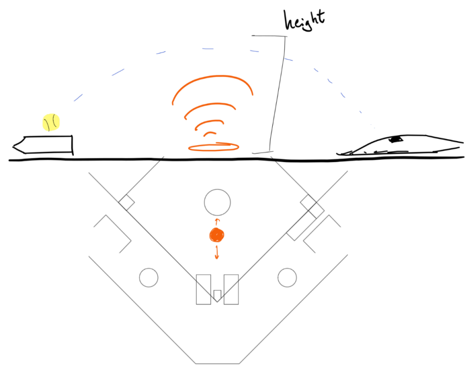

Figure 10.2.2

When we had decided on using a camera, we still considered where we should mount it. Before deciding on the side of the field, we considered using the back fence (Figure 10.2.3). We decided against this because it was harder for us to determine the height of the ball from the back angle.

Figure 10.2.3

The last major design we had to scrap was an iPhone app as well as an Android app. We originally planned on having both platforms supported, but realized that it was spreading our team too thin to be working on both applications. Thus, we decided that a great Android application was better than two "just okay" applications to have both platforms supported.

Appendix 3 – Other considerations

One thing we learned is that slow-pitch softball, a sport we originally thought was mostly a casual pastime, is taken *very* seriously by many players. Our application exists because of arguments with referees about illegal pitches, and some teams will even get in fistfights after a softball (per our advisor's experience). Thus, it was important for us to make sure our application was accurate and consistent.

To get us started at the beginning of this project, we used simple tutorials on Python Development and OpenCV ball tracking. Although most projects were immediately abandoned after answering the question or teaching the concept we were hoping to learn, one video on YouTube found its way into all of the workflows as a starting point. "Ball Detection Using OpenCV in Python" from *CodeSavant* was a great tool for us to get started initially.

[1] CodeSavant, "Ball Detection Using OpenCV in Python," YouTube, https://youtu.be/RaCwLrKuS1w?si=VmYMcoLu-q4PcCe- (accessed Dec. 7, 2024).

Appendix 4 – Code

Our code for this application can be found here: https://git.ece.iastate.edu/sd/sdmay25-49

To run, please use an Android phone or emulator of some kind. The command "flutter run" can be run from the "flutter app" folder to run the application.

Appendix 5 - Team Contract

Team Members

1) Cael Schreier

2) Drew Kinneer

3) Kolby Moorman

4) Sam Skaar

5) Kyle Nachiengane

Required Skill Sets for Your Project

a. Python and C++ Programming
b. Live image processing
c. Object detection
d. Height Analysis
e. Mobile Development
f. Team communication

Skill Sets covered by the Team

a. Python and C++ Programming: All team members
b. Live image processing: Drew, Cael, Sam
c. Object detection: Drew, Sam, Kolby
d. Height Analysis: Cael, Kyle
e. Mobile Development: All team members
f. Team communication: All Team members

Project Management Style Adopted by the team

a. Waterfall-style management and development for sequential tasks

Initial Project Management Roles

a. Cael Schreier: Bookkeeper and Code Review
b. Andrew Kinneer: Lead System Designer
c. Kyle Nachiengane: Lead Testing Engineer
d. Sam Skaar: Coordination and Documentation Lead
e. Kolby Moorman: Lead Frontend Developer

Team Contract

Team Members:

1) Cael Schreier

2) Drew Kinneer

3) Kolby Moorman

4) Sam Skaar

5) Kyle Nachiengane

Team Procedures

Day, time, and location (face-to-face or virtual) for regular team meetings:

a. Weekly Client/Advisor Meetings: Mondays, 10:00 AM - 11:00 AM, Durham 363
b. Weekly Team Meetings: Thursdays, 2:00 PM - 4:00 PM, Virtual

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

a. Discord and face-to-face for team
b. Email for advisor communication

3. Decision-making policy (e.g., consensus, majority vote):

a. Majority Vote

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

a. All Records are saved in a group Google Drive folder
b. Cael Schreier serves as the main note-taker for weekly meetings, notes are kept in Google Drive
c. All code is saved in a group Gitlab repository, team members are expected to push their code regularly.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

a. Mandatory in-person attendance for every advisor meeting
b. Hybrid attendance for group meetings, depending on what needs to be done

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

a. All team members will equally contribute to team assignments, as well as continue to make meaningful contributions toward the progress of the overall project every week, and report that progress at each client/advisor meeting.
b. A group review will be done before each submission.

3. Expected level of communication with other team members:

a. Always communicate what you are working on and what your progress is
b. If you fall behind or get ahead, let the team know so tasks can be redelegated if
c. needed
d. Use Discord and or Email to communicate meeting and work updates.

4. Expected level of commitment to team decisions and tasks:

a. Team members are expected to commit between 4-7 hours of work per week on our project.
b. Everyone voices their thoughts on tasks, even if it's brief
c. Be present at every meeting and class period

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction,

individual component design, testing, etc.):

f. Cael Schreier: Bookkeeper and Code Review
g. Andrew Kinneer: Lead System Designer
h. Kyle Nachiengane: Lead Testing Engineer
i. Sam Skaar: Coordination and Documentation Lead
j. Kolby Moorman: Lead Frontend Developer

2. Strategies for supporting and guiding the work of all team members:

a. Each member of the team will have their assigned responsibilities; however, each member does not have to only work on responsibilities assigned to their role. Collaboration and review will be important at each step of the process of this project. Each member is encouraged to reach out and ask for help if they are stuck in any step of their role. After meetings, deadlines will be set for what each member wants to accomplish for the project.

3. Strategies for recognizing the contributions of all team members:

    a. The end-of-semester documentation will highlight each team member's contributions to the project. Internally, a healthy culture of “giving props” is sufficient.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

    a. Kyle Nachiengane

        i. Experience working with sensors capturing live data.

        ii. Keeping a positive attitude within the team.

        iii. Skilled in problem-solving and finding issues in systems.

    b. Cael Schreier

        i. Skilled in debugging and analyzing code

        ii. Experience in working with clients and client communication

        iii. Embedded Systems experience

    c. Kolby Moorman

        i. Experience working in both frontend and backend.

        ii. Skilled in picking up new technologies and programming languages

    d. Sam Skaar

        i. Skilled in design documentation, team coordination, and leadership.

        ii. Have played softball and sports in general.

        iii. Embedded Systems Experience (C#)

    e. Andrew Kinneer

        i. Lots of experience with cloud computing solutions

        ii. Skilled in backend development

        iii. Experience in working in a team environment

2. Strategies for encouraging and supporting contributions and ideas from all team members:

a. During our weekly meeting, we will have a 10-15 minute standup where we talk about what we did the past week and what we plan to do the following week. This will also be a time to ask questions, communicate with other team members, and hold people accountable.
b. There are no bad ideas/questions!

3. Procedures for identifying and resolving collaboration or inclusion issues

a. Team members are encouraged to bring up any concerns directly during regular team meetings or through an anonymous feedback form, ensuring that everyone feels comfortable sharing without fear of judgment. If a team member feels the environment is obstructing their ability to contribute, they will communicate this with the designated team leader or through a group discussion, and we will work together to address the issue promptly and adjust the team dynamics if needed.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

a. Have a well-researched and tested design by January
b. Communicate promptly and clearly with team members and our clients
c. Start tasks early to allow proper testing time
d. Have fun!

2. Strategies for planning and assigning individual and teamwork:

a. First based on interest
b. Second, based on previous experience
c. Thir,d based on the urgency of tasks
d. Lastly based on the time and availability of members

3. Strategies for keeping on task:

a. "Assign" weekly tasks so that something is being done every week
b. Check-in multiple times a week with each other to make sure tasks are being completed

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

a. Schedule a meeting with all of the team members to ask why and figure out a solution so that it doesn't happen again

2. What will your team do if the infractions continue?

a. If the infractions continue, gather Dr. Fila and Dr. Jones to gather their opinions on a solution to the infractions.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) ____Cael Schreier____________________________________ DATE ______9/17/24______

2) _____Kyle Nachiengane_______________________ _______ DATE _______9/17/24_____

3) _______Kolby Moorman____________ _________________ DATE ________9/17/24____

4) _________Andrew Kinneer___________________________ DATE _________9/17/24___

5) ___________Samuel W. Skaar__________________________ DATE ___________9/17/24__