IC Chip: Automated Clay Target Scoring System

Project Plan

Team Number: sdmay19-08
Client: Dr. Henry Duwe
Advisor: Dr. Henry Duwe

Team Members
Eva Kuntz
Keith Snider
Cole Huinker
Michael Ruden
Steven Sleder
Philip Hand

Team Website: http://sdmay19-08.sd.ece.iastate.edu/
Team Email: sdmay19-08@iastate.edu
Revised: October 17th, 2018 / Version 2.0

Table of Contents

List of Figures

List of Tables

List of Symbols

List of Definitions

Below is a list of definitions of the terms used through this paper.

Focal Length: In the context of the project describes how zoomed in or out the view is based on the shape of the lens.
FOV (Field of View): An angle that represents the range of what can be seen or captured.
Gemineye: The official name of the mobile application.
GPU: Graphical Processing Unit.
Ground Station: see 'Houston'.
Houston: The official name of the physical device that will be placed on the skeet shooting field.
PCB: Printed circuit boards.
Physical Device: see 'Houston'.
Shooting Session: A session where users join and do a round of skeet shooting.
Shooting Squad: A set of users participating in a session.

1 Introductory Material

1.1 Acknowledgement

Client - Assistant Professor Doctor Henry Duwe

Advisor - Assistant Professor Doctor Henry Duwe

1.2 Problem Statement

While numerous aspects of clay shooting sports have been automated, specifically clay target loading and launching, there still remains one notable exception. Scoring for clay shooting sports has been a source of significant difficulty and cost. It requires and individual with good eyesight who is knowledgeable in the rules and procedures of the sport. Finding those who are qualified and willing to score at a reasonable cost has proven increasingly difficult.

The focus of IC Chip is to create a low-cost, fully-automated scoring system for clay target shooting sports, primarily skeet. The project intends to integrate machine learning and computer vision on a dedicated hardware package with consumer-grade devices such as cellphones and tablets. The system is intended to be rugged, portable, and easily deployed in order to allow for a one-time cost for the system as opposed to repeatedly hiring individual human scorers. As with many engineering problems, the solution is to remove the human element.

1.3 Operating Environment

The IC Chip system is intended to be deployed and operated on a standard skeet shooting range. As such, there exists potential that the deployed system may be subject to numerous environmental hazards. These include, but are not limited to stray target fragments, stray shot, and adverse weather conditions.

From these hazards, it is necessary to produce a system that displays a reasonable degree of ruggedness to ensure its survival in potentially damaging events. To this end, the system must be comprise of a reinforced case which also is water resistant to allow the system to continue to function in precipitation. Additionally, the system must be modular, with easily replaced components which are low cost in the event that protective measures fail. This will help to ensure a product that is up to spec with user expectations and robust towards its environment.

1.4 Intended Users and Intended Uses

The product is intended for use by an individual who is relatively knowledgeable in the layout of a standard skeet field, but has a very limited degree of technical knowledge. From this, the design must produce a product whose interfaces and instructions require only basic knowledge and understanding of the underlying system. Almost all technical aspects must be abstracted away to help to ensure a pristine user experience.

As such, IC Chip must fulfill the *plug and play* paradigm. The integration between the components must be seamless, robust, and and simple. Instructions must be easily understood and navigated, and the user interfaces readily accessible and utilized by the end user.

1.5 Assumptions and Limitations

Below are lists of assumptions and limitations with brief justifications.

Assumptions

- The system will only be used for skeet shooting: The rules that the customer wants us to implement are for skeet shooting.
- A round will take place in the afternoon or evening: Although this isn't a set rule, it is more likely for a round to take place in the afternoon or evening.
- The target will be orange and standard size (about 110 mm): These are the type of targets that the range will have on stock and what we will train on.
- Not intended to out-perform human senses and decision making: The system only needs to abide by the rules of the sport which depends on vision which can be subjective.
- The devices are to be constrained to the field: The device won't be used anywhere else outside of a skeet field.
- The ground station will not be directly shot at: We are going to consider protection for the ground station but will not expect the station to withstand a direct shot.

Limitations

- Cost must be less than $1500 in resources: Shooting ranges only have a limited budget and the client wants the least costly solution.
- The system must be portable: The system needs to be moveable and needs to run on a battery.
- Performance of video analysis: analysing video can be computationally expensive and can be processes more effectively in parallel. Such dedicated hardware can be expensive and can consume lots of power.
- Range of wifi: The mobile device can only be so far away from the ground station before a loss of connection. There also may be times when the wifi is unreliable.
- Hardware protection: The ground station and cameras need protection because they will be in areas vulnerable to clay target shards and shell casings.
- The project will need to be completed before May of 2019: The end of the second semester of senior design is the deadline and the deliverables must be submitted by then.

1.6 Expected End Product and Other Deliverables

The final product is split into two main deliverables, a hardware and a software deliverable. The first deliverable is the portable ground station with camera and a protective house, and the second deliverable is the mobile application for use on a mobile device, such as a smartphone or tablet. Although our team would like to deliver both the ground station and the mobile application at the same times throughout both the first and second semesters, it was decided that in order to produce the mobile application to our client's specifications, multiple iterations of development and testing may be required. Thus, there are multiple delivery dates (milestones) for the mobile application. A ground station (hardware) and mobile application (software) manual will also be provided with the final product at the end of the second semester.

- Physical Device: The physical device deliverable will be split into three major milestones, the prototype, the minimal viable product, and the complete product (Houston).
  - Prototype - December 7th, 2018
    - The prototype serves as a proof-of-concept and will consist of a complete set of diagrams detailing the all the hardware components that will be used in our ground station. This also includes how the ground station will be powered and the dimensions, layout, cost, and manufacturer of each hardware component. In addition, the prototype will also consist of the camera lense that will be used on the ground station, the technical specifications of the lense, and the camera's field of view. Lastly, the prototype documents will also contain diagrams documenting the placement of the ground station on a skeet shooting field for the best target classification accuracy.
  - Minimal Viable Product (MVP) - March 10th, 2019
    - The MVP will be a physical device with minimal functionality. The MVP will stand on its own and turn on and off without an external power source. In addition, design plans for a protective house for the physical device will also be provided.
  - Houston - April 26th, 2019
    - Houston will be a fully-functioning physical ground station with a protective house. The ground station will be able to connect with the mobile application (see below) and communicate clay target classification data to the connected mobile device. At this point, the ground station will have been rigorously tested to improve the object detection and classification algorithms, resulting in an accuracy of 95% or higher.
- Mobile Application: The mobile application will be split into two main deliverables: a prototype and the final version of the mobile application, Gemineye.
  - Prototype - November 30th, 2018

        - The prototype will serve as a functioning mobile application with all of the major requirements satisfied. This version of the mobile application may not directly correspond to the screen mockups, but the functionality will be present. In addition, this mobile application will have gone through at least one round of testing using a “fake” set of test target classification data.
    - Gemineye - April 26th, 2019
        - The final version of the mobile application will be a fully-functioning Android application that meets all the requirements specified by the client. At this point, the Gemineye application will have completed multiple rounds of rigorous validation and user acceptance testing and will be ready to use by the client.
- User Manuals
    - Hardware Manual - April 26th, 2019
        - The hardware manual will detail how to setup and turn on the ground station. This manual will also document how to place the protective cover on the ground station to prevent any damage from clay target pieces and other debris created during a shooting session. The manual will also have a troubleshooting section and instructions for proper disassembly, device cleaning and care, and storage of the physical device.
    - Software Manual - April 26th, 2019
        - The software manual will detail how to download and install Gemineye on a mobile device, in addition to providing the user with a “Quick Start Guide” for connecting the user’s mobile device to the ground station, in addition to a connection troubleshooting section. The manual will also have detailed instructions on mobile application uses and how a user can maximize the application’s functionality.

As noted earlier, there are multiple delivery dates for the ground station and the mobile application. That said, our team will deliver the final versions of both the ground station and mobile application, in addition to user manuals, at the end of the second semester.

2 Proposed Approach and Statement of Work

2.1 Objective of the Task

The goal of this task is to create a low-cost ground station with camera and protective house that can be physically taken and placed onto a skeet shooting range. The ground station will consist of multiple components, including a camera lense for video recording and a hardware component whose function is to perform computations on the video recorded and classify a clay pigeon target as dead, lost with attempt, or lost without attempt. In addition to the physical device, this task also includes the development of a mobile application that connects to the physical device. The mobile application will allow users to challenge target classifications, review video footage, and follow along with shooting group scores.

2.2 Functional Requirements

The functional requirements are split into two groups. The first group relates to the ground station and the second group of requirements relates to the mobile application.

2.2.1 Houston Ground Station Requirements

Below is a list of function requirements the ground station must adhere to.

1. The physical device hardware, including camera lense and internal power source, should be contained as a single physical device.
2. The physical device should be portable.
3. The physical device should connect to a mobile device for use with the corresponding mobile application using WiFi.
4. The physical device should produce its own WiFi signal.
5. The physical device should able to capture and process video in real time.
6. The physical device should determine and classify whether a clay pigeon target is dead, lost with attempt, or lost without attempt.
7. The physical device should be accompanied by a protective, yet removable, “house.”
8. The physical device should only be powered by an internal battery and no external power sources.

2.2.2 Gemineye Mobile Application Requirements

Below is a list of functional requirements for the mobile application our team will develop. The expected use case of the mobile application is for the user to start a recording session and to monitor and challenge the target classifications if desired. If the user wishes to challenge the target classification, they will request to review that shot’s video footage. After footage review, user can then either confirm the software’s classification, or manually change the classification.

1. The mobile application should know the rules of skeet shooting.
2. The mobile application should track the order of shooters in a squad, and make changes to the order based on skeet shooting rules.
3. The mobile application’s display screen should turn off automatically after 30 seconds of no interaction.
4. The mobile application should connect to the physical device via a wifi signal.
5. The mobile application should not store videos.
6. The mobile application should allow users to challenge a target classification.

7. The mobile application should display target classification on the screen once the shot classification has been determined.
8. The mobile application should allow users to view the individual members of a shooting squad for the current shooting session.
9. The mobile application should keep track of the scores of every shooter in the shooting squad.
10. The mobile application should keep track of and display the length (total time) of the shooting session.
11. The mobile application should not save a shooting session's scores.

Below is a use case diagram depicting the main functionalities of Gemineye based off the client requirements.

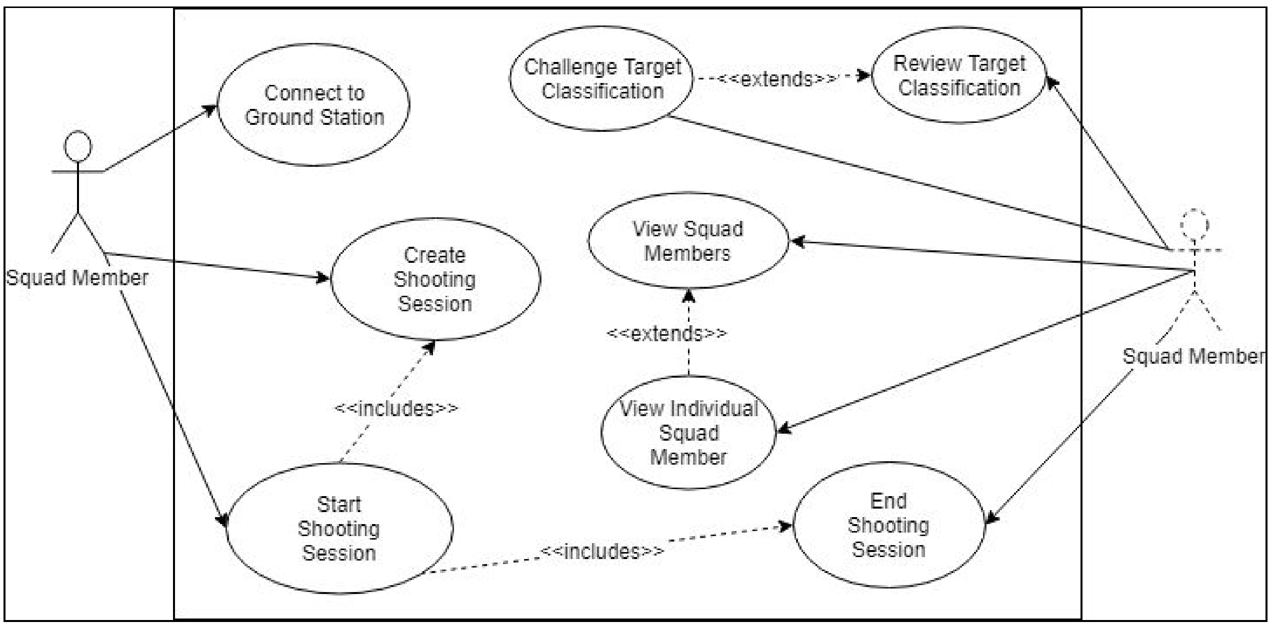

Figure 1: Use Case Diagram

2.3 Non-Functional Requirements

The non-functional requirements below are related to the performance, reliability, availability, security, and usability of our device and mobile application. Again, the non-functional requirements are split into two groups, the first in regards to the ground station, and the second in regards to the mobile application.

2.3.1 Houston Ground Station Requirements

Below is a list of non-functional requirements for the ground station.

Performance:

1. The physical device will perform computation and classify the clay target within 2 seconds of when the shot was made.
2. The physical device will notify the mobile application of the shot classification within 1 second after device has classified the shot.
3. The physical device will send most recent shot's video footage to mobile application within 3 seconds of the shot being made.
4. The physical device will classify targets (human-visible breaks) with upwards of a 95% accuracy rate.

2.3.2 Gemineye Mobile Application Requirements

Below is a list of non-functional requirements for the mobile application.

Performance:

1. The mobile application will receive the target classification from the physical device within 2 seconds after the shot is made.
2. The mobile application will display the target classification within 1 seconds after the classification is received from the physical device.
3. The mobile application will display the recording associated with a challenged shot within 3 seconds of the user challenging the target classification.
4. The mobile application will delete a video from memory within 1 second of a user accepting the target classification.
5. The mobile device with the mobile application should be within 5 feet of the ground station at all times.

Reliability & Availability:

5. If physical device and mobile application connection breaks, the mobile application will save the current shooting session's statistics until the session is terminated.
6. The mobile application will not rely on internet, outside of the ground station's WiFi signal, to perform all functionalities.

Data Integrity:

7. The mobile application will not store personal data.
8. The mobile application will not require user login information upon startup.

Usability:

9. The mobile application will be available to all users who have an android tablet or mobile phone.

2.4 Constraints Considerations and Standards

This section includes a list of constraint considerations and information about standards that our team will comply with.

2.4.1 Constraints

Below is a list of constraints regarding both the physical device and the mobile application.

- The physical device must be small and portable, as users may want to move the device to different locations on the skeet shooting field.
- The physical device must be accompanied with a protective covering, or “house,” to prevent damage from clay pigeon chips and other materials on the shooting range.
- The physical device must be a low-cost device (i.e. < $1,000).
- The physical device must not rely on an internet connection to perform its’ computations.
- They physical device must be battery powered.
- The physical device must not rely on any outside/external power source.
- The mobile application must not rely on internet to display the target classifications.
- The mobile application must not store videos in the mobile device’s memory.

2.4.2 Standards

There are a few standards that our project will comply with. Most of these standards are pre-built into the technology and frameworks that we will be using, such as OpenCV and Android Studio. At this point in time, we are unsure of what hardware components specifically we will be using, so we cannot say for sure the exact protocols we will be following. This will become clear once we pick specific hardware components.

2.5 Previous Work And Literature

When we first were introduced to our project concept, our team did research to determine whether there was any similar devices out on the market. The closest device we found was something called “ShotKam,” a small recording device that is attached directly to a gun barrel (ShotKam.com). ShotKam captures footage of the target from the shooter’s point of view and stores the video footage via a mobile application for later review and feedback. This device is similar to our project in the sense that ShotKam records the shot, however, there are a few key differences.

First, ShotKam’s focus is the target the shooter sees and follows the path of the gun barrel, due to being fixed on the underside of the barrel itself. We want our device to be stationary and on the ground--not on a gun barrel--and have a wide view of the shooting range. Having a wider view of the range allows our device to have a more accurate computational picture when classifying a target as dead or lost.

Second, the mobile app that accompanies ShotKam shows a live feed of the recording and stores the video at ⅓ the speed on the user's mobile device. While we want our mobile application to have access to the video recordings on our device, we do not want the videos to be stored on the user's phone, as this may cause memory issues. It is noteworthy to think about the slow motion aspect of ShotKam's stored videos. The slow motion feature allows users to watch their shot at a slow speed and determine if they hit their intended target. This might be a feature that our team wishes to include in our design, but will require more discussion with both our team and our client.

In terms of existing research into the area of object tracking and detection, there exists a promising algorithm: You Only Look Once (YOLO). The current implementation, YOLOv3 is described in Redmon and Farhadis' paper *YOLOv3: An Incremental Improvement*. Given sufficient computational power, it is capable of performing in real time.

We intent to, at the very least, utilize Redmon and Farhadis' current implementation of the algorithm in C++ which they have named DarkNet. By building on this previous work should allow us to expedite the prototyping of the scoring software.

2.6 Proposed Design

For the machine learning model itself, there currently exist no other true alternatives. You Only Look Once (YOLO) is the current state-of-the-art model. It's benchmarks tremendously supersede traditional computer vision, even on limited hardware. It also allows for rapid prototyping as it comes in several forms of pre-packaged software. It has rapidly become the industry standard for numerous applications for object detection and tracking.

The second part of our proposed design is an interactive user interface. We will be designing a mobile application for the user to review scores and potentially modify them after replaying the video of a shot. This will be incredibly helpful for the user as some clay pigeons are barely hit and is hard to be seen by the naked eye. It is easy to distinguish whether a clay pigeon has shattered in mid-air, but one of the bigger challenges and need for an electronic system is to help capture the harder to see hits on a clay pigeon. The application will indicate whether it believes the clay pigeon was hit or not.

Finally, the true design decision arises in the number and placement of cameras on the field itself. Thus far, the project has experimented with a single camera placed behind the field. Unfortunately, this severely limits the features in the recorded video, which is expected to drastically reduce the performance of the classification model (YOLO). To this end, a decision must be made on whether to move the camera to a closer position on the field. This decision will unfortunately lead to a reduction in the overall viewing angle of the path of targets. To resolve this issue, multiple cameras must be considered, though this implementation will increase cost, risk to the hardware, and increase resource requirements for storing, processing, and transmitting necessary data.

2.7 Technology Considerations

Capturing, processing, and transmitting video of a high speed object imposes a several challenges that we had to solve and implement solutions for. For capturing the object, a single camera doesn't effectively cover the entire field in a stationary point. If the shooter hits the target early or later in its path then the camera will not see it. Using a single very high resolution camera with a very high frame rate would be exceed the total cost of the budget. We had to decide on a cost-efficient solution that involves multiple low cost cameras that would cover the entire field. This multiple camera systems will allow the device to have a constant view of the object in flight. This would also allow the system to view the object at different angles; if one camera didn't see a hit, another camera could. Another consideration regarding the cameras would be their field of view (FOV) and focal length. Having a wide FOV would be great for covering the entire field although having a FOV that is too wide can cause significant distortion that could affect image processing. Also having a wide FOV will cause the image to be zoomed out which may cause problems with detecting the clay targets. Considering all these factors for the cameras are import when it comes to video data collection.

Processing video and transmitting at almost real time would need dedicated hardware. Most single-board computers (i.g. Raspberry Pi, Arduino) lack the hardware and the necessary connections to accomplish this. We decide that creating our own single-board computer would be beneficial because it would allow us to have control on the speciations of the hardware. in the custom single-board computer, we will have a microprocessor that will be able to process video in real time, memory to store the video, and WiFi chip capable of transmitting the length of the field.

Additionally, there is the issue of the classification model. Existing literature almost guarantees that, given sufficient diversity in training data, sufficient classification of targets is entirely feasible. The issue that arises is again a hardware constraint. These models require massive amounts of parallel operations on vectors in order to perform in a reasonable amount of time. Performing in near real-time will require even more parallelism. This is usually achieved through the use of a GPU. However, new, low-power embedded systems with the capabilities to perform similarly have begun to hit the market in recent years.

As far as the user being able to interface with the ground station, a mobile application will be developed using the Xamarin framework. Xamarin is a .NET framework used to develop mobile applications natively on many platforms. Sockets will also be used to communicate with the ground station from the mobile application.

2.8 Safety Considerations

The major safety concern of our project is the safety of device users and our team members when we test the device on the shooting range. It is extremely important, on any shooting range, to be aware of one's surroundings, as shotgun shells or clay pigeon chips, for example, may come flying towards a person. Because of this, most ranges require participants to have eye and face protection (i.e. baseball cap to protect forehead). Range participants and onlookers also have to protect their hearing, as the sound level of a shotgun blast can permanently damage a person's hearing (Stewart).

When our team first collected video data at the Boone County Sportsmen's Club with Dr. Henry Duwe, we wore eye protection, baseball caps, and ear protection. It was made clear to us that whenever we are on or near the shooting range, hearing, eye, and face protection are required. So, when we go on to test our physical device, we will ensure all team members have the appropriate protection.

Unfortunately, we cannot ensure that our device users will wear the proper eye and hearing protection equipment. Although most shooting ranges do require these methods of protection, users may decide not to use protection. So, we will provide safety recommendations along with our device instructions in the hopes users will take the safety precautions seriously.

2.9 Task Approach

2.9.1 Detection System and data collection

Our client has asked us to help devise a system that will help skeet competitors determine whether a clay pigeon has been struck. To solve this problem, we will need to be able to:

- Clearly identify if the object moving through the air is a clay pigeon
- Determine if the clay pigeon has been struck by the shotgun shell
- Determine the type of break on the clay pigeon
- Reveal an indicator to identify whether there was a hit or miss

To capture the movement of a clay pigeon and accurately identify whether the pigeon was struck we will be using a two or three camera system rather than a single camera. The multiple camera system would cover the the entire field, a camera placed near the high house would cover the left side of the field and a camera place at the low house would cover the left side of the field. A third camera could be placed in the middle if the right next to station 4 that would cover the middle section of the field (the camera locations related to the field are shown in the diagram below). The multiple camera systems has the added benefit of seeing the object in multiple angles. Having multiple angles helps determine if the clay pigeon has been hit or not, and which type of break the pigeon endured.

For determining what camera and position would be the best we created a MatLab program that would plot out the field with a given camera properties (field of view and field of depth) at a certain position. This program helps with visualizing the camera properties and what placements of the the camera system would result in a full range of coverage of the field.

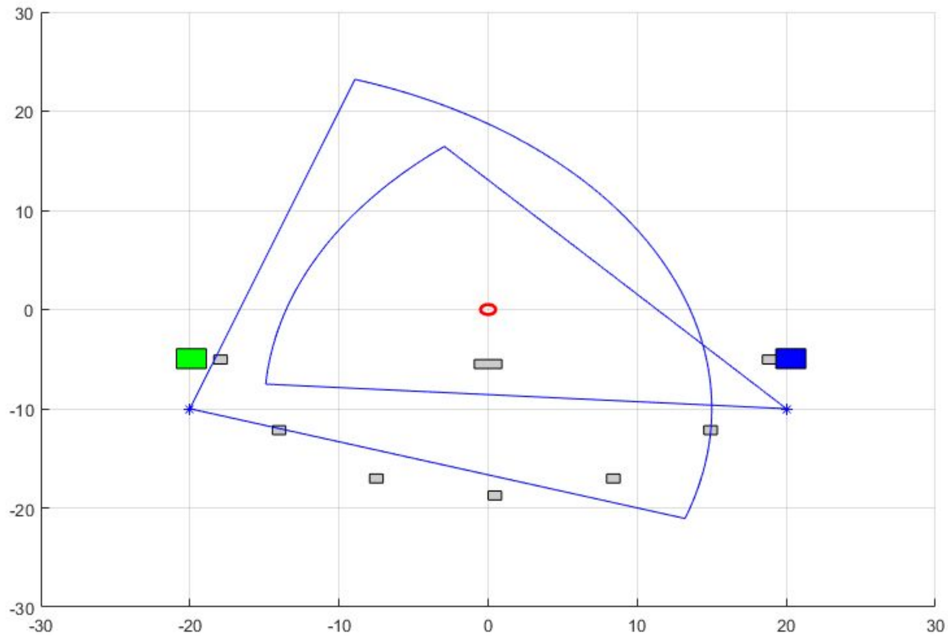

Figure 2: Camera Position on Field Program

2.9.2 Software Task Approach

The software development for this project will include the development of the mobile application and the development of the target detection model. The design process that our software development will follow is Test Driven Development (TDD). Using TDD as a design process makes since for us since we are in the process of writing the software and not just maintaining it. The purpose of using TDD is to make sure that during development, testing is a major priority and that it is done often.

The process of development will include short cycles with the following steps:

1. Create, Reuse, and Revise Tests: When given specific requirements, write or modify tests that will satisfy the requirements.
2. Run Tests: Run the tests (even though nothing new is implemented yet.) This step will show what needs to be worked on and what is already satisfactory. It is very likely that most (if not all) tests will fail during this step.
3. New Implementation: Based on the tests that failed, implement a solution. This step will be the bulk of the development process. Implement the best solution that works and can make sense to other developers
4. Run Tests (again): This step will show whether or not the new implementations of the code will pass the tests. If the test passes, then for the moment the

requirement will be considered complete. If the test fails, then the development for that particular requirement will continue into the next cycle.

5. Refactor: Clean-up the code. Get rid of unnecessary code such as commented out code. This would also be a good time to consider organization and different design patterns that could be used to improve the code. Any changes during this step may break the code to which it will be tested during the next cycle.
6. Repeat: Go back to step 1 to repeat the cycle.

2.10 Possible Risks And Risk Management

2.10.1 Software Related Risks

This project will be completed using Xamarin framework for the mobile application and Python/C/C++ for the software on the hardware component. Only two of our team members, Cole Huinker and Keith Snider, have experience with designing and developing a mobile application and none of the team members have had any experience with Xamarin. This poses a challenge as our software architect lead, Eva Kuntz, does not have a lot of experience with this particular technology stack. Fortunately, there are three team members working on the mobile application, so they will work together to overcome problems that arise from the lack of knowledge surrounding Xamarin. In order to mitigate some of the risks associated with this. We hold meetings specifically about our development process on a weekly basis to discuss requirements and design for the mobile app. This is a time for us to discuss what individuals have done, review code, and ask questions. This is also a time to hold workshops to get developers up to speed on the Xamarin framework.

Another risk is the limited experience with machine learning and OpenCV. Only one team member, Steven Sleder, has previous work experience with machine learning algorithms and techniques. Instead of teaching the entire team about machine learning theories, Steven will work with Cole on the OpenCV component of our project and reach out to other team members for help as needed. In order to avoid work being isolated to one person. Code regarding machine learning and OpenCV will be pushed to the repository frequently. Also any data such as videos or still frames will be shared on CyBox so that the whole team will have access to the data sets and can help label data.

Since the mobile application will be developed using Xamarin and the target classification and video analysis will be done in Python, this may pose a challenge in communication between the two systems. In order to avoid this problem, our team will do research and experiment with different things such as sockets to determine the best course of action before we commit to anything specific.

2.10.2 Hardware Related Risks

Another problem our team may run into is the hardware component assembly and connection to mobile devices. After completing research regarding connectivity, Mike Ruden concluded that installing a wifi chip on one of the hardware components would be the best way to transfer data reliably and accurately to the connected mobile device. However, Mike is the only team member who has had experience installing wifi chips, so if he runs into an issue, other team members can only provide limited help. In order to try and avoid hardware issues, the members in charge of hardware will keep the team updated on their progress. If any problems arise, we'll make sure to direct them towards a professor or someone at the ETG to help us out.

In addition to the limited knowledge of technologies, our team still is deciding which hardware components to use. This is a risk because we need to start designing our prototype soon, and in order to do so, we need to know the hardware component specifics so we can make informed design decisions. The best way to mitigate this issue to to be in contact with knowledgeable people such as professors and with the ETG.

2.11 Project Proposed Milestones and Evaluation Criteria

The milestones listed below will be used to determine the progress being made on the project.

- Project Requirements and planning: Having an extremely detailed project plan will help ensure our success to complete this project. Other planning requirements are having a design document for the mobile application, as well as design specifications and document for the hardware.
- Data collection and analysis: Collect video data out in the field. Organize and label video clips. Analyze the clips and have them ready for CV analysis.
- Clay target movement detection (moderate accuracy): Based on our video data, detecting the clay target movement should be around 90% accurate.
- Clay target object recognition (moderate accuracy): Based on our video data, the program should be able to detect whether the clay target was hit, and the shell casing fired from the shotgun. The detection of objects other than the clay pigeon or shell casing should be almost negligible.
- Mobile app prototype. An Android application that effectively communicates between multiple mobile devices.
- Camera device prototype: The camera device is implemented and tested to gather video data, communicate over WIFI, and do computer vision computations.
- Mobile app and camera device communication prototype: Mobile devices are able to connect to camera device(s), get video, and keep score.
- Client prototype test: have the client(s) test the prototype.
- Clay target movement detection (high accuracy): Based on our video data and video captured while in the field, detecting the clay target movement should be over 95% accurate.
- Clay target object recognition (high accuracy): Based on our video data and video captured in the field, the program should be able to detect the clay target when not hit, the clay target when hit, and the shell casing. Detecting other objects as

clay targets should be almost null at this stage. Detecting the objects should be over 95% accurate.

2.12 Project Tracking Procedures

Our team is using a variety of methods to track our progress. Below is a list of the methods we are using and why we decided to use the particular method.

2.12.1 Regular Meetings

Our team holds regular meetings on Mondays and Wednesdays of every week. We also have a regular client meeting with Dr. Duwe on Thursdays. These meetings allow us to check in with each other face-to-face and carve out a set time for everyone to work on our project. Since all team members are in the same physical location, the meetings allow us to determine what progress has been made since the last meeting and to discuss any concerns or issues team members may be having. If, for example, a team member has hit a roadblock on a certain task they are working on, they can brainstorm with the entire team and receive help to overcome the issue if needed.

Our regular client meetings allow us to share what progress we have made with our client. These meetings also allow us to clarify any unknowns or questions regarding projects requirements and elicit any information we need from Dr. Duwe. In addition, these meetings also hold our team accountable for those tasks we said we would accomplish.

2.12.2 Slack

Our team uses Slack as a way to communicate quickly and efficiently with other team members. If one team member has a question regarding a task they are working on, they can easily post the question in Slack and get an immediate response.

Slack provides our team with an easy way to plan meetings and clarify task assignments.

2.12.3 GitLab

Not only does our team use GitLab as a place to place our project code, but we also use GitLab as a way to document tasks and issues that need to be completed. Of all the methods used to track project progress, GitLab is the most important. Our team can follow our project progress based on the number of tasks completed and the number of commits made. GitLab will become more of a priority once we start developing the mobile application and the software that will be on the physical device.

2.12.4 GoogleDocs and Meeting Notes

In addition to the above methods for tracking our project progress, we also use GoogleDocs as a way to list “Action Items” and keep them in one place. At every team meeting, meeting notes are taken to record topics discussed, any current or pending issues and their solutions, and upcoming, or action, items that need to be done. These action items are reviewed at the next team meeting and either reassigned, troubleshooted, or closed.

2.13 Expected Results and Validation

The desired goal is to have a fully functioning, portable device with a protective house and a mobile application that meets the objectives outlined in section 2.1. To confirm our solution works at a high level, we will take the device to a skeet shooting range and track the device's clay target classifications with the classifications of a human referee, in addition to monitoring communication between the ground station and the mobile application on a mobile device. This will also help our team determine the accuracy of our solution at a high level. Our team will validate our design by completing the test plan and test cases documented in section 2.14.

2.14 Test Plan

Our test plan consists of two major components: the physical device testing and the mobile application testing. The sections below describe the tests our team will perform to validate the design of both our hardware and software components.

For this section, please note that "Functional Requirement" is denoted "FR" and "Non-Functional Requirement" is denoted "NFR."

2.14.1 Houston Ground Station Test Plan

The table [Table 1] below lists the tests our team will use to validate the functionality of the ground station, specifically. The test cases are detailed later in this section.

Requirement	**Test Case**	**Critical**
F2. The physical device should be portable.	P1: Device Portability	yes
F4. The physical device should produce its own WiFi signal.	P2: WiFi Signal	yes
F5. The physical device should able to capture and process video in real time.	P3: Capture and Process Real-Time Video	yes
F6. The physical device should determine and classify whether a clay pigeon target is dead, lost with attempt, or lost without attempt.	P4: Target Classification w/o Accuracy	yes

Table 1: Houston Ground Station Test Cases

The list below details the test cases and execution steps.

1. **Test Case-P1: Device Portability**
    a. Description: This test case will validate whether the device is portable, in addition to determine if the device requires physical exertion to move.
    b. Steps:
        1. User picks up physical device.
        2. User moves physical device up to 5 feet from original device placement.
        3. User sets physical device down.
    c. Expected Results: User is able to move the physical device without showing signs of physical exertion.
2. **Test Case-P2: WiFi Signal**
    a. Description: This test will validate whether the physical device can produce its own discoverable WiFi signal.
    b. Steps:
        1. User turns on the physical device.
        2. User searches for a WiFi signal via their mobile device.
        3. User's mobile device is able to find the physical device's WiFi signal.
    c. Expected Results: User is able to search for and find the WiFi signal of the physical device.
3. **Test Case-P3: Capture and Process Real-Time Video**
    a. Description: This test will validate that the physical device can capture and process video in real-time. Note, for this test, we will modify a [the] mobile application to receive and display footage from the ground station.
    b. Steps:
        1. User will set up physical device and start recording video footage.
        2. User will watch mobile application to view the video footage the camera on ground device is picking up and processing.
    c. Expected Results: The video footage the physical device is recording and processing corresponds to the environment in real-time.
4. **Test Case-P4: Target Classification w/o Accuracy**
    a. Description: This test will validate that the physical device can classify a target. The importance is not necessarily on the accuracy of the classification.
    b. Steps:
        1. User aims and shoots at a clay target.
        2. The physical device displays a red light if the device has successfully detected and processed a target classification.
    c. Expected Results: The physical device will successfully display a red light when a target classification has been made.

2.14.2 Gemineye Mobile Application Test Plan

The table [Table 2] below lists the tests our team will use to validate the functionality of the mobile application, specifically. The test cases are detailed later in this section.

Requirement	**Test Case**	**Critical**
F2. The mobile application should track the order of shooters in a squad, and make changes to the order based on skeet shooting rules.	M1: Squad Member Order Tracking	no
F3. The mobile application's display screen should turn off automatically after 30 seconds of no interaction.	M2: Display Screen	no
F5. The mobile application should not store videos.	M3: Application Should Not Store Videos	yes
F6. The mobile application should allow users to challenge a target classification.	M4: Challenge Target Classification	yes
F7. The mobile application should display target classification on the screen once the shot classification has been determined.	M5: Display Target Classification Results	yes
F8. The mobile application should allow users to view the individual members of a shooting squad for the current shooting session.	M6: View Individual Shooting Squad Members	no
F11. The mobile application should not save a shooting session's scores.	M7: Shooting Session Scores Deleted After Session	yes

Table 2: Gemini Mobile Application Test Cases

The list below details the test cases and execution steps.

1. **Test Cast-M1: Squad Member Order Tracking**
    a. Description: This test will validate that the mobile application is able to track the order of individual shooting squad members based off their scores.
    b. Steps:
        1. User opens application on mobile device.
        2. User creates and starts a shooting session.

    3. Shooting squad purposely takes steps to change their shooting order, based on skeet shooting rules.
  c. Expected Results: Mobile application updates the order of shooting squad members accurately in accordance to skeet shooting rules, tracking the same order changes as the squad members do.

2. **Test Cast-M2: Display Screen**
  a. Description: This will validate that after a period of inactivity, the mobile device returns to the screensaver or lock screen mode.
  b. Steps:
    1. User opens application on mobile device.
    2. User does not interact with the application for at least 30 seconds.
    3. The phone screen returns to the screensaver mode.
  c. Expected Results: After 30 seconds of inactivity, the mobile device screen should go to the screensaver, or lock screen, mode.
3. **Test Cast-M3: Application Should Not Store Videos**
  a. Description: This test validates that the mobile application does not store videos on the mobile device.
  b. Steps:
    1. User opens application, creates, and starts a shooting session.
    2. Once a target classification and video are sent and received in the mobile application, User “agrees” with the classification.
    3. User then exits out of the application to check the video storage on their mobile device.
  c. Expected Results: User should not find the sent video anywhere in their mobile device video storage folder(s).
4. **Test Cast-M4: Challenge Target Classification**
  a. Description: This test will validate whether the application user is able to challenge a target classification.
  b. Steps:
    1. Mobile application receives target classification data from ground station.
    2. Mobile application displays information to User.
    3. User selects “challenge” from the “Accept Classification/Challenge Classification” options on received classification.
    4. User is able to either accept the reviewed video or manually enter their perceived classification.
  c. Expected Results: User is able to challenge a received target classification and the user’s score updates according to the results of the challenged classification.
5. **Test Cast-M5: Display Target Classification Results**
  a. Description: This test validates whether the mobile application displays the target classification results received by the ground station.
  b. Steps:
    1. User shoots at a clay target.

    2. Physical Device processes shot video footage and sends classification data to the mobile application.
    3. Mobile application receives the target classification data and displays the classification on the screen.
  c. Expected Results: A target classification will be displayed on the mobile application screen after a user/squad member shoots at a clay target.

6. **Test Cast-M6: View Individual Shooting Squad Members**
  a. Description: This test will validate that users can view the members of a shooting squad.
  b. Steps:
    1. User opens mobile application and creates and starts a shooting session.
    2. User selects "View Squad" button of mobile application.
    3. A list of all the inputted squad members is displayed.
  c. Expected Results: The list of all inputted shooting squad members, in the correct order, is displayed in the mobile application.

7. **Test Cast-M7: Shooting Session Scores Deleted After Session**
  a. Description: This test validates that each time a user ends or starts a new shooting session, the previous session's scores are not stored in the mobile application.
  b. Steps:
    1. User ends a current shooting session.
    2. User exits out of the application.
    3. User opens application.
    4. User attempts to create a new shooting session and should not see any old squad members or scores present when creating the new session.
  c. Expected Results: User will be able to create a new shooting session, without having to remove the old shooting session data.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Project Timeline

We believe this timeline of events is achievable because we have given ourselves ample amount of time in for each deliverable. When coming up with dates for the timeline we decided when we thought about how long it would take us to complete a milestone, and then added on a week or two. We have all experienced in past projects that our expected timelines in our head don't account for unforseen struggles or roadblocks. By doing this it becomes more likely that we will achieve our milestones before schedule than later than schedule.

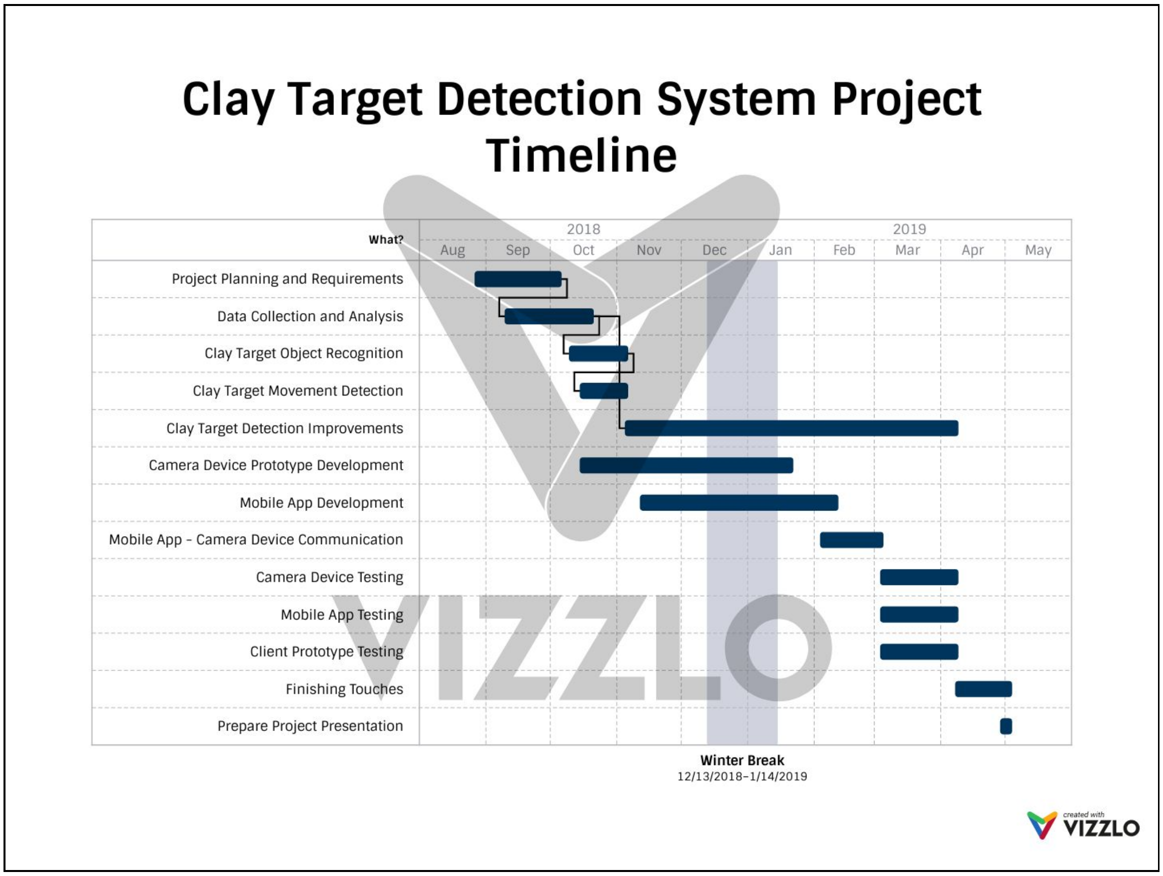

Figure 3: Project Timeline

During the first semester, our team will spend a large chunk of our time collecting video of clay target shooting and analyzing it to fit our machine learning model, we plan to have a working model that can detect a clay target by the start of November, in this time we will also be working on tracking the clay target as it moves across the frame of view in the video. Once our model is able to detect the clay target and it's movement we will start training it to improve its ability to do so. While we are training our model to detect the target, we will begin development of our camera device prototype and our mobile application.

During the second semester our team will continue improvements on our clay target detection and motion tracking. With a working prototype of the model we will begin to understand what are the computation requirements for the hardware. With the requirements known we can begin designing the necessary hardware for the ground station. The designing of the ground station would take a couple weeks for all the necessary modules. Once the modules are designed it and have been tested it would then take a couple weeks for the manufacturing of the PCBs. While the PCBs are being manufactured, we will begin designing the housing for the physical device. Assembly for the physical device would take a week. Once our application and camera device are working together we will begin testing on the prototypes. Towards the end of the second semester we will begin working on the final presentation of our project.

3.2 Feasibility Assessment

As has previously appeared in the document, the intended purpose of the system is to replace human involvement in scoring a round of skeet. This will serve to reduce cost, set-up time, and human error. As such the device must be ready for use after a brief set-up time, robust to the environmental factors it may face, and display a large degree of simplicity for the user. With these goals in mind, several potential issues are seen immediately.

The most significant issue arises in the classification of a target as a *hit* or *miss*. For all of the pre-existing literature and software that can be used to solve this problem, there still remains the issue of training a model do perform in an appropriate manner. This is due to the fundamental nature of machine learning models, which is that given a dataset to train on, they learn to extract and process that data for a classification based upon patterns which exist within it. The problem is that any bias which exists in the dataset which is used to train the model may lead to undesirable performance.

This is known as overfitting, training to a narrow set of data that does not adequately represent the real world. In order to overcome this, we must ensure that our training data is invariant to a variety of features that could change in the normal operating environment. Examples of this include things like backgrounds for ranges, lighting conditions, weather patterns, animals in the image, and many more aspects. The easiest manner to ensure this is to simple record a large variety of training data from several locations, under different conditions.

Next, there is the matter of inter-communication between the various components that make up the system. The project must utilize robust and connection-oriented systems to transmit data which are capable of multiple simultaneous connection. These also must be robust to interference from things such as competing Wi-Fi and Bluetooth signals. In this, the hardware must provide the necessary range and bandwidth, and the software the necessary error-handling to recover from unexpected events.

Finally, there is the matter of the user experience. This project is targeted at individuals with limited experience with the various technologies that make it possible. As such it is vital that they are abstracted away. This means that any error messages must be as descriptive as possible to a lay person and reference material to solve the issue without in-depth research on the behalf of the user. Furthermore, extensive testing is necessary to ensure that such an individual is comfortable and capable when it comes to interacting with the system.

3.3 Personnel Effort Requirements

Below in table 3 is a list of personnel effort requirements. The table includes task items, a description of the task, and an effort that is an estimate of how long a task may take to complete.

Personnel Effort Requirements Table		
Task Title	Task Description	Effort
Weekly report	Every week a report needs to be submitted. Each member must record hours worked, what they worked on, and what they are going to be working on.	~2 hrs/week
Project plan and design document updates	A project plan must be created and updated throughout the project lifetime. Create design and specification documents for hardware/camera device and mobile app.	Initial draft of documents should take ~4 hours to draft. Update ~1hrs/week
Data Collection	Plan camera positions, angles,and FOV. Plan number of rounds, shots and the number of locations for the camera.. Collect video data for CV analysis and training on the field.	~3hrs per collection session ~2hrs per week of planning
Video Editing	Edit videos down only to where the shoot yells “pull”, up to a second after the target has been shot. After that, take individual frames from the cut down video and label the frames with bounding boxes.	10+ hours after collection session
CV analysis and training	Implement a solution for detecting clay target movement and object detection. Train using video data.	10+ hours per week
Hardware requirements/Design	Create design and specification documents for hardware/camera device.	Initial draft of documents should take ~2 hours to draft. Update ~0.5hrs/week

Camera device casing design.	Design the housing unit for the camera device.	3 hours
Mobile app requirements/design	Create design and specification documents for the mobile app.	Initial draft of documents should take ~4 hours to draft. Update ~0.5hrs/week
Device communication	Work on communication between the ground station and the mobile app.	7+ hours per week
Android app prototyping	Build a mobile app prototype that is able to connect to the camera devices, get data, and keep score.	8+ hours per week
Mobile app testing	Test functionality of mobile app.	3+ hours per week
Ground Station prototyping	Build camera device prototype that is able to communicate over WIFI and collect and analyse video data.	8+ hours per week
Ground Station testing	Test that the ground station can: ● Capture video ● Analyse video and detect targets ● Communicate with the mobile device over wifi	3+ hours per week
CV movement testing	Test accuracy of clay target movement accuracy.	2+ hours per week
CV object detection testing	Test accuracy of clay target movement accuracy.	2+ hours per week
Ground Station, mobile app, and communication integration and testing.	Integrate camera device and mobile app prototypes and test after integration.	20+ hours
User acceptance testing	Have the users test the prototype. Observe the user's processes and get feedback from them	6+ hours
Deployment	Deploy final version of the project.	10+ hours

Table 3: Personnel Effort Requirements

3.4 Other Resource Requirements

For the hardware and camera, we'll need access to facilities on campus in order to work with the board and cameras. For creating the housing unit for the ground station and hardware, our team will need a 3D modeling tool such as SolidWorks to design the housing unit and we would need access to a 3D printer to print the case for the unit.

For the mobile application we will need Android devices to test the app on. For the most part, I believe most of the team have Android phones that can be used, but an Android tablet may be useful for testing on different types of devices. Using a tablet may be better when viewing the view to review a shot. A small screen could make it difficult to see if any small chips came off of the clay target.

We need access to the field along with cameras in order to collect video data of shooting clay targets. We need access to figure out what the best locations for the cameras are, what angles are best suited to have the best view of the field, and what spots may be considered dangerous to the equipment.

3.5 Financial Requirements

The total budget of the project is around $1,000 - $1,300, with the physical device taking up majority of the cost of the total project. The cost of the device can be broken up into couple different parts; cost of materials, cost of assembly & manufacture, and the cost of testing.

The cost of the materials of the device will take up the majority of the budget. The design requirements needs to have multiple cameras that can capture high resolution. These cameras will be off the shelf with a cost of around $100-$200, the cost could very on the specifications needed for the camera. The computing device will either be a stock single-board computer with a GPU which is priced around $200-$300. The power supply for the device would depend upon whether it will be power by a battery or connected to an outlet. If connected to an outlet the major cost of the power supply would be a transformer. If running on a battery, the battery would be the major cost. The battery and the transformer cost would depend on the amount of power need to supply to the device. As the need for power increases the cost of the transformer/ battery would also increase.

The cost of assembly and manufacture will depend on the on either the project will require a custom single-board computer. If it is required, the cost of board will depend number of layers and the total size of the board. Since that most microprocessors have ball-grid arrays (BGA) the only way to assembly is through pick-and-place machine. The cost of using this depends on what company we get our board manufereced. The house can be made using one of the Iowa State 3D printers, which would eliminate the cost of the manufacturing cost of the house. A premade housing can also be purchased for around $40.

Testing will need to take up a small portion of the budget with going out to the field and collecting data samples. It can be broken up into the cost of transportation to getting and from the field and the cost of renting out the field. The total cost is around $21 per sample collection.

Total Budget	$1,000-$1,300
Cost of Materials	
Device Compondents	($125-$175)
Cameras	($500)
Cost of Assembly	
PCB Manufacturing	($-)
PCB Assembly	($-)
House Manufacturing	$0
Cost of Testing	
Field Fees	($18)
Transportation	($18)
Total	$300

Figure 4: Project Budget

4 Closure Materials

4.1 Conclusion

In summary, *IC Chip* intents to provide a solution to the problem of human involvement in the scoring of clay target sports. The project aims to provide a rugged *plug-and-play* scoring system utilizing current machine learning and computer vision algorithms to seamlessly integrate with readily available user devices such as tablets and smartphones. The project will achieve this goal through minimizing reliance on expensive computational equipment through own own printed circuit designs, as well as utilizing open-source software packages to create a location-invariant machine learning model for classification of targets.

4.2 References

Below is a list of references used thus far. We expect this list to grow through the rest of first semester and into second semester.

"Gun Camera for Shotgun, Handgun, and Rifle - Official Site." *ShotKam*, shotkam.com/.

Stewart, Michael. "Recreational Firearm Noise Exposure." *Averican Speech-Language-Hearing Association*, ASHA, www.asha.org/public/hearing/recreational-firearm-noise-exposure/.