IC Chip: Automated Clay Target Scoring System

Design Document

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

Team Members
Eva Kuntz - Software Architect; Project Manager
Keith Snider - Software Architect; Webmanger
Cole Huinker - Software Architect; Data Analytics; Computer Vision
Michael Ruden - Hardware Architect Lead; Prototype Manager
Steven Sleder - OpenCV and Machine Learning Lead; Data Analytics Lead
Philip Hand - Device connectivity Lead and Hardware

Team Website: http://sdmay19-08.sd.ece.iastate.edu/
Team Email: sdmay19-08@iastate.edu
Revised: November 26th, 2018 / Version 2.0

Table of Contents

List of Figures

List of Tables

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 Introduction

1.1 Acknowledgement

Client - Assistant Professor Dr. Henry Duwe

Faculty Advisor - Assistant Professor Dr. Henry Duwe

1.2 Problem and Project 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 Operational 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 Uses

This project is intended to produce a product which may be utilized by an individual reasonably familiar with the layout of a Skeet range and with limited technical ability. 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 ensure a pristine user experience.

The final product will consist of a physical device and a mobile application for use on a mobile device, such as a smartphone or tablet. The physical device will record and process video footage to determine whether a clay target was hit (classified as "dead"), or if the shooter missed the target (classified as a "loss"). Thus, he physical device will act as a second pair of eyes for the referee and the shooting squad. The mobile application will allow the referee and shooting squad to review video footage and challenge the target classification for the most recent shot.

Due to the nature of the hardware and software components, the system must fulfill the plug and play paradigm. Therefore, integration between the components must be robust and redundant, deployment instructions simple, and user interfaces easily navigated.

1.5 Assumptions and Limitations

Below is a table that lists the assumptions of the hardware and the software components of this project.

Assumption	**Justification**
The system will only be used for skeeting shooting.	The client desired a system that fulfills only the rules of skeet shooting, as this time.
A shooting round will occur in the afternoon or the evening.	Although this is not a set rule, shooting rounds are more likely to occur in the afternoon and 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 our device to recognize and track.
Users will only use Android Devices	Developing for Android does not require any developer licences whereas ios does.
Device needs to be placed in a protective, yet removable housing.	Device needs protection from flying range debris, such as shotgun shell and clay target fragments.
All video analysis will be done on the device and must be done in almost real time.	Client requirement.
WiFi will be used to transfer video recordings from the physical device	WiFi is more reliable than Bluetooth.

to the mobile device.	

Table 1: Project Assumptions and Justifications.

Below is a table that lists the limitations of the hardware and the software components of this project.

Limitations	Justification
Time of day	Light throughout the day will change.
Cost	The total cost of project, including hardware and software components, must be less than $1000 (client requirement).
Hardware device must be portable.	Client requirement.
Device will be operated with battery power.	This allows the device to be portable and versatility in set-up.
Performance of video analysis.	Analyzing 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 the WiFi signal.	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.
Camera Resolution and framerate	As the resolution and framerate increase, so will the cost which will cut into the project budget as a tradeoff to video quality.
Project Deadline	All deliverables must be completed on or before May 2019.

Table 2: Project Limitations and Justifications.

1.6 Expected End Product and 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.

To summarize, the final product will be as follows:

- **Houston:** A small, portable, battery-powered device with a camera lens able to detect clay target pieces that may be invisible to the naked eye and classify a target as dead, loss with attempt, or loss without attempt. This device will also be able to differentiate between clay target pieces, the shotgun shell and wad, and other objects the enter the camera view.

- **Gemineye:** A mobile application suitable for use on consumer-grade devices, such as a smartphone or tablet, that, once connected to Houston via WiFi, allows a user to create and track scores of members in a shooting squad during a shooting session. The mobile application will also allow users to challenge target classifications and, upon video footage review, accept or manually override the original target classification.

Both the physical device and mobile application will be accompanied by user manuals.

2 Specifications and Analysis

2.1 Design Specifications

The design specifications, or requirements, for this project, are split into two groups. The first relates to the physical device and the second group relates to the mobile application. Each group has both functional and non-functional requirements.

2.1.1 Houston Ground Station Functional 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.

2.1.2 Houston Ground Station Non-Functional 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.

2.1.3 Gemineye Mobile Application Functional 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.

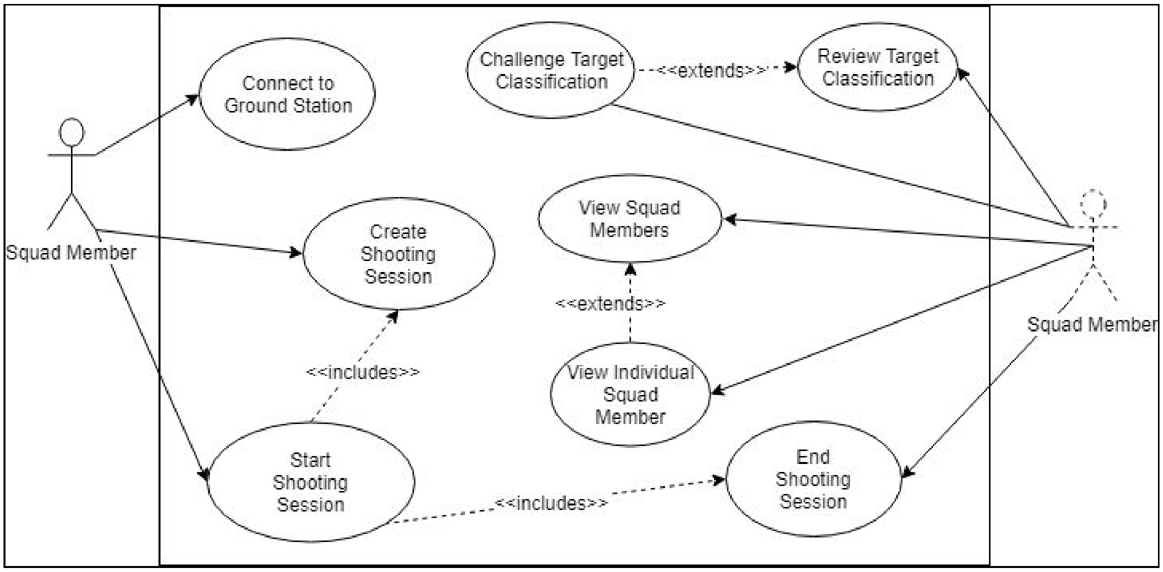

Figure 1: Use Case Diagram.

2.1.4 Gemineye Mobile Application Non-Functional 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.

2.2 Proposed Design

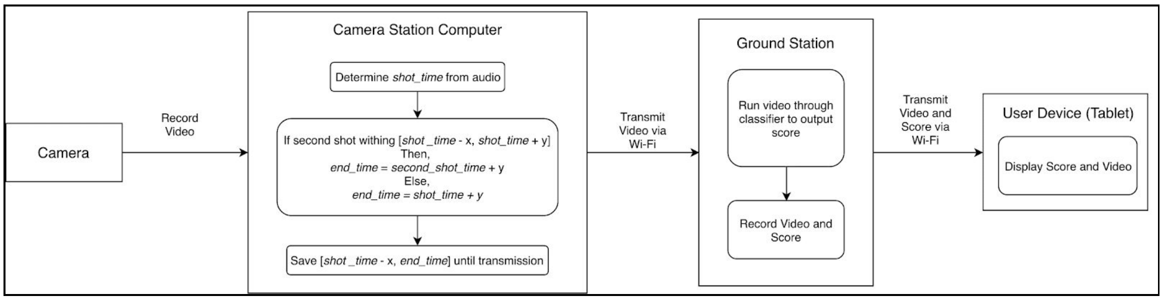

Figure 2: System Block Diagram

The current iteration of the configuration of our system is shown in the above figure 1. The system contains four primary components in a pipe-lined configuration from raw input to user presentation. The components are the camera, camera station controller, ground station, and user device.

2.2.1 Camera

The first state component is the camera, taking in raw data from the field. The primary concern for this component is as high of a resolution as possible in order to capture as many image features as we can. This is more significant in classification of targets than frame-rate as each frame is looked at for classification as an individual image and a break will persist across multiple frames. This device features a hard connection directly to the camera station controller.

2.2.2 Camera Station Computer

The camera station controller itself will feature a low-power ARM processor and 802.11-capable network device. In addition it will be contained in a semi-ruggedized shell as it will potentially be in harm's way due to being located closer to the target-launchers and and in range of shotguns. It will receive data from its physical connection to the camera. This video is then reduced to an individual shot based off of audio. The camera station the utilizes its 802.11-capable network device to transmit the video fragment to the ground station for classification and logging.

2.2.3 Ground Station

The ground station in this current design revision is an Nvidia Jetson TX2 due to the large amount of parallelism that the system is capable of, a necessity for machine learning and computer vision. The ground station will receive a fragment of video from the camera controller and immediately run the classifier. After running the classifier on the input video, it will be logged, a score will be determined, and it will be submit wirelessly to the user device. In order to do so it will also feature an 802.11-capable network device.

2.2.4 User Device

The user device itself is not specifically determined by the project. This is due to the fact that it is entirely determined by the user. They need only install the application which should be Android and iOS agnostic. It will receive an update to the score for each shot and provide an option to retrieve and review the video from the ground station or contest a score. It will communicate through the ground station via its 802.11-capable wireless device.

2.3 Design Analysis

Collecting a sufficient amount of training information for the classification model proves to be the first weakness of the project. Almost all other components of the project hinge on an effecting classifier. This, in turn, relies upon a sufficient training dataset. If the team manages to provide a large and diverse enough dataset to train on, this hopefully will cease to be a weakness and instead be a strength of the project. To this effect, we have been recording shots in diverse conditions from diverse orientations to prevent over-training and make the model invariant to as many changes as possible.

One of the other weaknesses of our design plan is the lack of guidelines around the hardware. Our client has been pretty lenient about what hardware components we plan to use--as long as the components are under the project budget and allow us to arrive at a solution, our client does not care what technology we use. This, however, has presented a problem for our team: At this point, we still do not have a clear cut idea of the hardware components we will use. This will be our main focus during the second part of the project.

3 Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library. In this section, we discuss the tests that our team will perform to ensure our system meets both functional and nonfunctional requirements specified by our client.

3.1 Interface Specifications

Hardware - The hardware will not have a GUI it will just have a CLI as its interaction with the user will be abstracted away by the front-end.

Software - The software will be an android application which will work as an interface for the hardware to increase user friendliness and reduce the learning curve for using the system.

A majority of our applications logic and processing will be done on our hardware. The hardware will initialize a local WiFi connection that our android application will connect to. From there our hardware will send video and the supposed outcome to the android app. The android app will then allow the user to view the most recent shot, and tell them if a hit or miss was detected.

3.2 Hardware and software

For performing hardware testing on the prototype we can use variety of equipment that is located in the Coover engineering labs for measuring and creating signals.

- Digital Multimeter - A digital multimeter is used to measure voltages and current anywhere on the prototype. This can be used to ensure connections are correct and to verify that that the correct parts of the prototype are active at the create time
- Oscilloscope - An oscilloscope can used to measure and view signals in the prototype. Viewing a signal in the prototype will allow us the verify that the signal isn't being distorted or filtered when it travels through the design.
- Signal Generator - Signal Generator creates artificial signals for the inputs of the prototype. By knowing what signal we put into the prototype and the output signal, we can determine how the prototype response to given inputs.
- Power Supply - Provides a DC signal to input of the device. Using a power supply we allow us to test the prototype without the need of batteries.

Software testing on the prototype differs in that it requires little to no specialized equipment. Instead, it will focus on unit testing, hopefully achieving 100% code coverage for the user application, communication system, and various backend helper programs that may exist on the ground station

The only software aspect that will differ is the testing of the object detection and tracking. This will need to be tested by feeding in pre-scored data that the model has not been trained on previously, and analysing the output score to determine the rate of mis-classification.

3.3 Model and Simulation

We encountered a problem when we first started working on this project of determining what camera to use and how to place it on the field. We didn't know what the camera would capture or at what the video quality would be like at a certain distance. For solving this problem without going out to the field each time we want to choose a different camera or position to see how it would look, we created a Matlab program that would help visualize the system on the field.

This programs takes in the three different camera properties: the field of view, the depth of field near limit, and the depth of field far limit. These properties of the camera can be calculated using focal length and f-stop value which could be found in the cameras datasheet. Using these values found in the cameras datasheet we can plug them into the equations to get the values that we need for simulating the system. The program would show the slice of the field that would be captured and within the slice it would show what would be within the depth of field. The next set of inputs would be where on the field the camera would be placed. The program has its origin located at the target crossing. If the user wants to place a camera to left it would have a negative x distance, to the right it would be positive distance. An example of an output of this program is shown below.

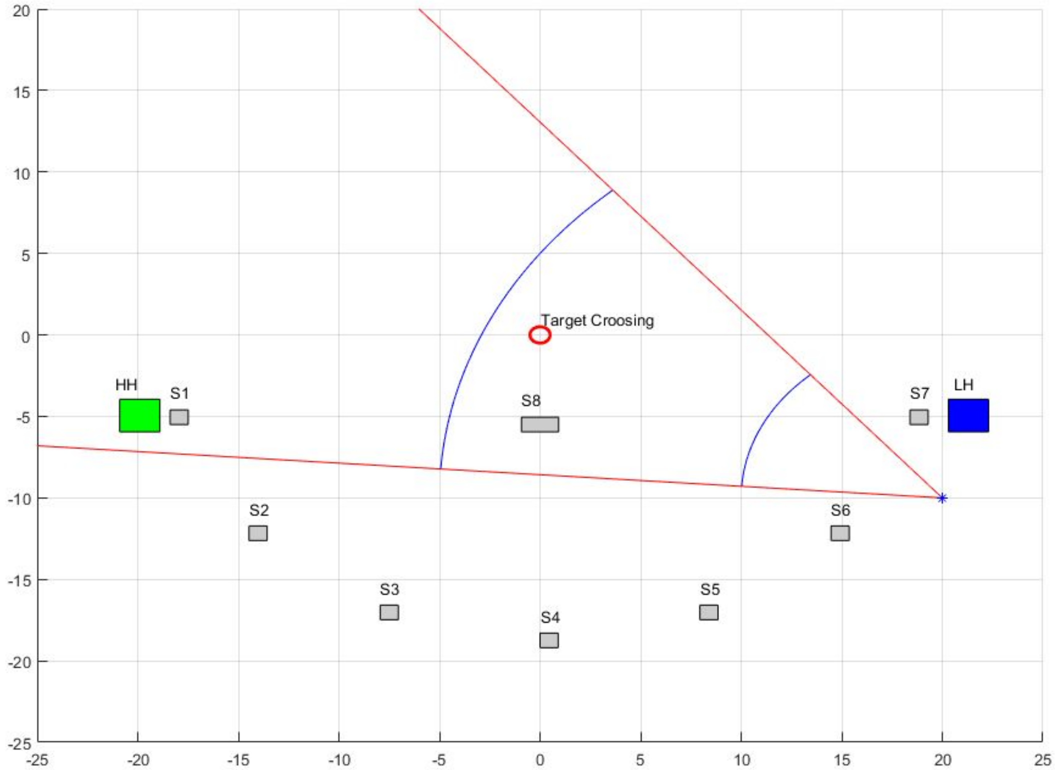

Figure 3: Camera Simulation.

The example above shows a camera place between station 7 and 8 with a field of view of 45° looking at the camera crossing with a field of depth near of 10 meters and the field of depth of 25 meters.

3.4 Test Plan

In this section, we detail the types of tests we will use to validate our system, in addition to the current status of the tests.

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 "F" and "Non-Functional Requirement" is denoted "NF."

For this section, please note that "P#" refers to a physical device test case, while "M#" refers to a mobile application test case.

3.4.1 Houston Ground Station Test Plan

The table [Table 3] 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.

In addition to the tests detailed below, our team will also observe our client using the physical device without any instructions beforehand. This way, our team will have a good understanding of how easy and intuitive our device is to use. Then, our team can create an appropriate and descriptive user manual and quick start guide to accompany our system.

Requirement	**Test Case**	**Critical**	**Attempted**
F2. The physical device should be portable.	P1: Device Portability	yes	no
F4. The physical device should produce its own WiFi signal.	P2: WiFi Signal	yes	no
F5. The physical device should able to capture and process video in real time.	P3: Capture and Process Real-Time Video	yes	no
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	no
F7. The physical device should be accompanied by a protective, yet removable, "house."	P5: Protective House Offers Device Protection and is Removable	yes	no
NF1: The physical device will perform computation and classify the clay target within 2 seconds of when the shot was made.	P6: Classification Time	yes	no
NF2: The physical device will notify the mobile application of the shot classification within 2 seconds after device has classified the shot.	P7: Classification Notification Time	yes	no
NF3: The physical device will send most recent shot's video	P8: Video Footage Transfer Time	yes	no

footage to mobile application within 3 seconds of the shot being made.			
NF4: The physical device will classify targets (human-visible breaks) with upwards of a 95% accuracy rate.	P9: Classification Accuracy Rate	yes	no

Table 3: Houston Physical Device 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.

5. **Test Case-P5: Protective House Offers Device Protection and is Removable**
   a. Description: This test will validate that the physical device is protected from skeet shooting range debris and that the protective house is removable.
   b. Steps:
      1. User sets up (model) physical device with protective house.
      2. User successfully removes protective house without any physically strenuous movements and without damaging the physical device.
      3. User tosses pieces of clay target debris at the protective house and determines whether damage to the physical device has occurred.
   c. Expected Results: The protective house successfully prevents clay target and other debris from damaging the physical device and user is able to easily remove the protective house without strenuous physical effort and without damaging the physical device.
6. **Test Case-P6: Classification Time**
   a. Description: This test will validate that the physical device makes a target classification within 2 seconds of an attempt shot.
   b. Steps: (Note: This test will be performed in a monitored, testing environment.)
      1. Physical device code will be modified to start a timer when the camera detects a clay target and a shot has been made and to stop when a classification has been made.
      2. Total time (timer stop time - timer start time) for target classification will be recorded.
      3. After at least 250 shots, the total time to classify a target will be averaged and recorded.
   c. Expected Results: The physical device successfully makes a target classification within 2 seconds of an attempt shot.
7. **Test Case-P7: Classification Notification Time**
   a. Description: This test will validate that the physical device sends the target classification to the mobile device so it is received within 2 seconds of being sent.
   b. Steps: [See M9]
   c. Expected Results: [See M9]
8. **Test Case-P8: Video Footage Transfer Time**

a. Description: This test validates that the physical device sends the video footage associated with a challenged shot to the mobile device for display within a total time frame of 3 seconds.
b. Steps: [See M11]
c. Expected Results: [See M11]

9. **Test Case-P9: Classification Accuracy Rate**
   a. Description: This test validates that the algorithms on the physical device are able to classify clay targets with a 95% accuracy rate.
   b. Steps:
      1. User successfully connects to the physical device's WiFi network and creates and starts a shooting session.
      2. User keeps track of the device's classifications and the human referee's classification.
      3. Accuracy rate is determined at the end of the 250 shots.
   c. Expected Results: 95% of clay targets are classified accurately, as validated by the human referee.

3.4.2 Gemineye Mobile Application Test Plan

The table [Table 4] 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.

Similar to the physical device testing, our team will first observe our client using the mobile application without prior instruction. Again, this is so we understand how easy and intuitive our system is to use. Once we have made observations, we can then create a more effective user manual and quick start guide.

Requirement	Test Case	Critical	Attempted
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	no
F3. The mobile application's display screen should turn off automatically after 30 seconds of no interaction.	M2: Display Screen	no	no
F4. The mobile application should connect to the physical device via a wifi signal.	M3: Mobile Application Connects to Physical Device's WiFi Network	yes	no
F5. The mobile application should not store videos.	M4: Application Should Not Store Videos	yes	no

F6. The mobile application should allow users to challenge a target classification.	M5: Challenge Target Classification	yes	no
F7. The mobile application should display target classification on the screen once the shot classification has been determined.	M6: Display Target Classification Results	yes	no
F8. The mobile application should allow users to view the individual members of a shooting squad for the current shooting session.	M7: View Individual Shooting Squad Members	no	no
F11. The mobile application should not save a shooting session's scores.	M8: Shooting Session Scores Deleted After Session	yes	no
NF1: The mobile application will receive the target classification from the physical device within 2 seconds after the shot is made.	M9: Mobile Application Time to Receive Target Classification	yes	no
NF2: The mobile application will display the target classification within 1 seconds after the classification is received from the physical device.	M10: Mobile Application Time to Display Target Classification	yes	no
NF3: The mobile application will display the recording associated with a challenged shot within 3 seconds of the user challenging the target classification.	M11: Mobile Application Time to Display Video Footage After Challenge Target Classification	yes	no
NF5: If physical device and mobile application connection breaks, the mobile application will save the current shooting session's statistics until the session is terminated.	M12: Mobile Application Save Session State Upon Lost WiFi Connection	yes	no
NF6: The mobile application will not rely on internet, outside of the ground station's WiFi signal, to perform all functionalities.	M13: Mobile Application No-Other-Resource Use	yes	no

Table 4: 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 Case-M3: Mobile Application Connects to Physical Device's WiFi Network**
    a. Description: This will validate that upon opening the mobile application, the mobile device can successfully find, identify, and connect to the physical device's WiFi network.
    b. Steps:
        1. User powers on the physical device.
        2. User opens the mobile application on their smartphone or mobile tablet.
        3. The mobile device's WiFi menu appears and shows the physical device's WiFi network as available to connect.
        4. User selects physical device's WiFi network and device successfully connects to that network.
    c. Expected Results: Upon opening of the mobile application, the mobile device should successfully find and connect to the physical device's wifi network.
4. **Test Cast-M4: 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).

5. **Test Cast-M5: 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.
6. **Test Cast-M6: 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.
7. **Test Cast-M7: 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.
8. **Test Cast-M8: 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.

9. **Test Case-M9: Mobile Application Time to Receive Target Classification**
    a. Description: This test validates that the mobile device, therefore, mobile application, receives the target classification from the physical device within 2 seconds of the physical device making the classification.
    b. Steps: (Note: This test will be performed in a monitored, testing environment.)
        1. Modifications to the mobile application code will be made to note the time when a classification is received from the physical device. Modifications to the physical device code will be made to note the time when a classification is made and is sent to the mobile device.
        2. User successfully connects to the physical device's WiFi network and creates and starts a a shooting session.
        3. User shoots clay targets.
        4. Physical device timestamps are sent after the classification is sent.
        5. Total time to receive target classification (mobile device receive timestamp - physical device sent timestamp) are noted.
        6. After 250 shots, average receive time will be computed.
    c. Expected Results: The averaged time to receive a target classification (from physical device to mobile device) is equal to or less than 2 seconds.
10. **Test Case-M10: Mobile Application Time to Display Target Classification**
    a. Description: This test validates that the mobile device displays the target classification within one second of receiving the classification from the physical device.
    b. Steps: (Note: This test will be performed in a monitored, testing environment.)
        1. Modifications to the mobile application code will be made to start a timer when a classification result is received from the physical device and stop the timer when the classification result is displayed.
        2. User successfully connects to the physical device's WiFi network and creates and starts a a shooting session.
        3. User shoots clay targets.

4. Total time (timer stop time - timer start time) to display classification are noted.
5. After 250 shots, the average time to display classification will be computed.

c. Expected Results: The averaged time to display target classification on mobile application upon receiving the classification from physical device should be equal to or less than 1 second.

11. **Test Case-M11: Mobile Application Time to Display Video Footage After Challenge Target Classification**
    a. Description: This test validates that the mobile application displays the video footage associated with a challenged shot within 3 seconds of the user challenging the classification.
    b. Steps: (Note: This test will be performed in a monitored, testing environment.)
        1. Modifications to the mobile application code will be made to start a timer when a challenge action occurs and to stop the timer when the video footage is received from the physical device.
        2. User challenges a target classification after successful connection to the physical device and creation and start of a shooting session.
        3. Timer (modifications from step 1) will record time (ns) it took for video to be received from physical device.
        4. Mobile application will display the time for User to see.
    c. Expected Results: The mobile device receives the video footage of the shot associated with the challenged classification within 3 seconds of the challenge action.
12. **Test Case-M12: Mobile Application Save Session State Upon Lost WiFi Connection**
    a. Description: This test validates that the mobile application saves the current shooting session state if the WiFi connection between the mobile device and the physical device is lost during an active shooting session.
    b. Steps:
        1. User successfully connects to physical device's WiFi network.
        2. User successfully creates and starts a new shooting session.
        3. At some point during the session, user turns off the mobile device's WiFi service.
        4. User waits 5 seconds and then turns the WiFi service back on.
        5. Mobile device reconnects to the physical device's WiFi network.
        6. User checks that that shooting session was saved correctly and is able to resume the session.
    c. Expected Results: The shooting session should be saved and resumed without error or lost data.
13. **Test Case-M13: Mobile Application No-Other-Resource Use**

a. Description: This test validates that the mobile application does not use (require) any other resources (i.e. mobile data, internet, etc…) than the WiFi signal from the physical device to function properly.
b. Steps:
    1. User turns off mobile data functionality on their mobile device.
    2. User selects and connects to the physical device's WiFi network.
    3. User ensures mobile device is not connected to any other WiFi network or mobile hotspot.
    4. User creates and starts a shooting session. User proceeds with the shooting session as normal and notes any abnormalities, if they occur.
c. Expected Results: The mobile application should function as per the client requirements (i.e. normally).

3.5 Process

Our team is following the Test Driven Design (TDD) model, especially in regards to the mobile application development. Our team believes that not only is testing important, but so is ensuring that the client gets the product that they asked for. Thus far, our project has two "processes." The first is related to gathering data and the hardware components and the second process encompasses the mobile application and mock data for testing. In regards to gathering data, our team first developed a "Data Collection Action Plan," which specified the location(s) of a camera on the skeet shooting range to capture shots. After developing the data collection guide, we physically visited a skeet shooting range and collected video data. Then, our team working on writing a script to break each video into an individual shot with clay break. Once all the video processing was finished, our team focused on drawing bounding boxes around the clay fragments to help teach our machine learning model to recognize clay targets. For the most part, data collection will be an ongoing process because more variety of data will improve the model and will avoid over-training.

As far as the mobile application goes, our team focused first on creating a detailed test plan that covered both functional and non-functional requirements. This way, the tests are what drive our application (TDD). Thus, our team will work on developing the application in short rounds, while ensuring the application passes all tests.

Below is a diagram that depicts our design process.

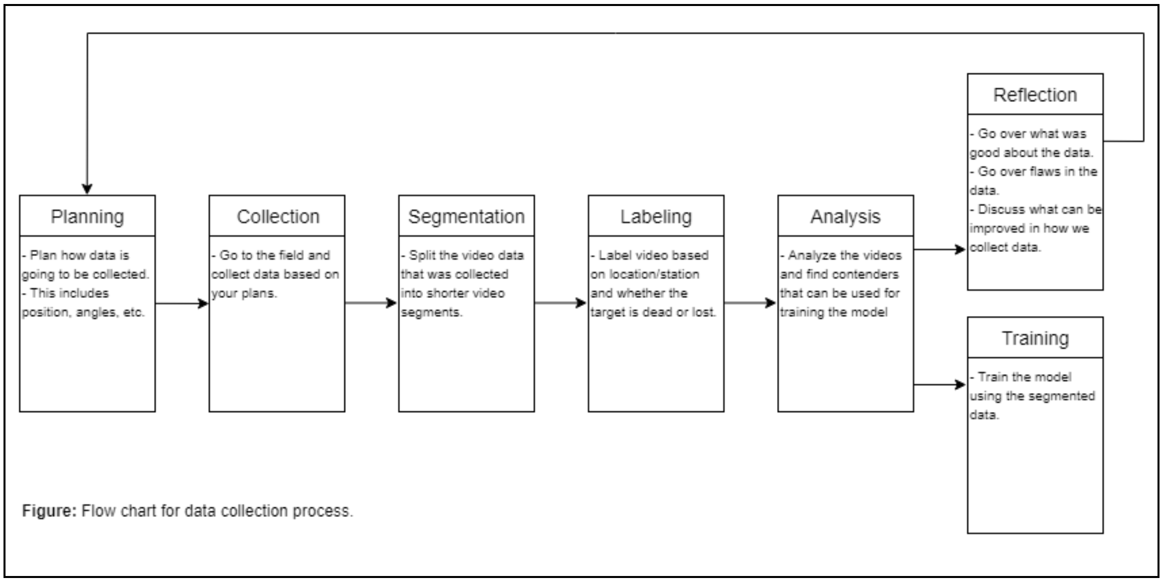

Figure: Flow chart for data collection process.

3.6 Results

Most tests are unable to be performed as of the current date, we are in the midst of finalizing some of our research so that we may start some of the initial testing phases of both our hardware and software capabilities. We have begun to setup our android software application and have attempted to connect our android application to a local Wi-Fi signal initialized from our own hardware. This process is still ongoing as we have not yet been able to reliably connect our application to a local hardware Wi-Fi signal.

An initial round of data collection has taken place on the field. Although at this point it may be extremely difficult to analyze some of our data due to the small, fast moving clay pigeon targets. We have been able to successfully label and identify most of our data and determine if the clay target is dead or lost. There are some videos that have failed to provide us with useful results, as sometimes you may not see the clay pigeon at all during its flight. In the future we hope to gain access to better cameras which will make it easier to clearly see if the clay pigeon has been hit.

One challenge that we will face over the duration of our project is finding a camera that is good enough to be able to provide us with an image that we may use to accurately identify if the clay pigeon has been destroyed or only chipped. We also need to find the best location to place our camera so that we may capture most of the flight of each clay pigeon or decide if multiple cameras are needed, while also finding a way to fit this into our budget.

3.7 Implementation Issues and Challenges

3.7.1 Data Collection

The collection of data has a couples issues that we have encountered. The first being the safety of the camera system. The cameras that we used for data collection are rentals from the Iowa State Library and will not have the protective housing that the final product will have. To good about solving this we placed a protective screen of plexiglass in front if the camera before each round of shooting. This introduced an added problem of a glare that the camera, but that prove to be not an issue for the model.

3.7.2 Data Labelling

The construction of any machine learning model requires a tremendous amount of training, validation, and testing data. In order to construct this the data must be parsed into discrete and labelled units to be useful for supervised learning.

In the raw form, the model takes individual video frames as input and performs classification on them as individual still images. To train a system capable of this it must be trained on a series of labelled images.

Therefore, we must construct a labelled dataset. To begin, we utilized a program known as FFMPEG and a custom Python script to pipe in videos. This splits them into individual still .PNG format images. This format is chosen as the images are lossless and moving them between systems will not lead to any degradation in their quality from compression.

In our first dataset, this ripping of each individual frame, shot at about 60 frames per second, resulted in over 20,000 individual images. These are then taken and labelled with an open source program called OpenLabeling that is used for drawing bounding boxes on clay targets to mark them as 'Live' or 'Dead'. This is an incredibly time-consuming process.

Furthermore, the dataset must avoid any additional patterns that the model could possibly learn. The model can potentially learn any other pattern that is present in the data, therefore it must be invariant to any of these. As it is nearly impossible to determine what patterns may be learned, the best possible way to avoid this is through as diverse of a sample set as possible. An example of this is in normal and adverse weather conditions or various different lighting.

4 Closing Material

4.1 Conclusion

In summary, IC Chip intends to provide a solution to the problem of human involvement in the scoring of clay target sports. It is to be a rugged and easily deployed autonomous system utilizing current machine learning and computer vision algorithms to integrate seamlessly with readily available user devices such as tablets. The overall system is decomposed into two major parts: the physical device (Houston) and the mobile application (Gemineye). Houston will be a portable, protected device that one sets up on the physical skeet shooting field. This ground station will not only track the clay targets, but will also determine whether a shooter successfully hit the clay target or missed it with the use of machine learning and computer vision. The accompanying mobile application, Gemineye, will allow users to challenge a clay target classification and keep track of individual shooter's scores throughout a skeet shooting game. Overall, our team's goal is to create a low-cost, easy-to-use system that will make it easier to score clay target sports.

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.

[1] "Gun Camera for Shotgun, Handgun, and Rifle - Official Site," *ShotKam*. [Online]. Available: https://shotkam.com/. [Accessed: 29-Nov-2018].

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *Survival Strategies for Robot Rebellion*, 08-Apr-2018. [Online]. Available: https://pjreddie.com/. [Accessed: 29-Nov-2018].

[3] M. Stewart, "Recreational Firearm Noise Exposure," *Averican Speech-Language-Hearing Association*. [Online]. Available: http://www.asha.org/public/hearing/Recreational-Firearm-Noise-Exposure/. [Accessed: 29-Nov-2018].