IC Chip: Automated Clay Target Scoring System

Final Report

28 April 2019

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

Team Members
Eva Kuntz - Software Architect; Project Manager
Keith Snider - Software Architect; Web Manager
Cole Huinker - Software Architect; Data Analytics; Computer Vision
Steven Sleder - Machine Learning Lead; Data Analytics Lead
Michael Ruden - Hardware Development; Data Analytics
Philip Hand – Power/Hardware; Data Analytics

Team Website: http://sdmay19-08.sd.ece.iastate.edu/
Team Email: sdmay19-08@iastate.edu

Table of Contents

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 system that will be placed on the skeet shooting field. This part of the system consists of
PCB: Printed circuit boards.
Physical Device: see 'Houston'.
Shooting Session: A period of time devoted to playing a round of skeet shooting.
Shooting Squad: A group of people actively participating in a shooting session.

List of Figures

1 Introduction

1.1 Problem and Need 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 an 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, to aid in the classification of clay targets. 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.

Although the goal of many engineering projects is to remove the human element, this project will not completely remove the human element. Instead, IC Chip will act as a supplement for the cases where target classification is too difficult to determine with the naked eye.

1.2 Solution

The IC Chip solution consists of two parts: a small, portable device (Houston) that a user can physically bring onto a skeet shooting range that classifies clay targets, and a mobile application (Gemineye) that allows a user to keep track of scores during an active skeet shooting game and challenge target classifications as they see fit.

Houston is a stand-alone physical device capable of producing its own WiFi signal for a mobile device to connect to. The physical device does not use an outside power source and is equipped with a camera lens to capture a clay target on video. Our team has integrated machine learning and computer vision into the hardware package so clay targets can be tracked and classified as dead or lost.

Gemineye is an Android application for use on a consumer-grade device, such as a tablet or smartphone. The mobile application, once the mobile device has connected to the ground station, allows users to create a shooting squad and start and track a shooting session. The mobile application allows users to track their scores during a skeet shooting game, in addition to reviewing the target classification of the ground station. Users can control when Houston starts recording video from the mobile application. Once recording video, Houston will track and classify the clay target, and, upon ending a recording, will send the target classification to Gemineye for user confirmation. If a user decides to challenge a classification, they can review the video footage before manually overriding or accepting the ground station’s classification.

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 comprised of a reinforced case that is also water resistant to allow the system to continue to function in precipitation. Additionally, the system must be modular, with easily replaced, low-cost components 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

IC Chip is intended for use by individuals who are relatively knowledgeable in the layout of a standard skeet field and the scoring mechanisms but have a limited degree of technical knowledge. From this, the resulting system is intuitive to navigate and all interfaces and instructions require only basic knowledge of the underlying system. Almost all technical aspects have been abstracted away to help ensure a pristine user experience.

1.5 Assumptions and Limitations

Below is a list of assumptions and limitations that our team has worked with throughout the duration of this project. Each assumption and limitation also has an accompanying justification [Assumption/Limitation : Justification].

1.5.1 Assumptions

Below is the list of assumptions.

1. The system will only be used for skeet shooting: The game rules the customer wants us to implement are those of skeet shooting.
2. A skeet shooting round will take place in the afternoon or evening: Although this is not a set rule, it is more likely that a round will take place in the afternoon or evening.
3. The clay pigeon will be standard size (110mm) and orange: The standard size orange clay targets are the type of targets that the skeet shooting range will have in stock. In addition, these are the type of targets the machine learning model will train on.
4. The system is not intended to outperform human sense and decision making: The system only needs to abide by the rules of the sport, which depends on human vision, which can be subjective.
5. The system is constrained to the skeet shooting field: The device will not be used anywhere besides a skeet shooting field.

6. The system (ground station) will not be shot directly at: The ground station will be protected from clay target fragments and precipitation by a protective house, but the covering is not expected to withstand a direct hit from a shotgun.

1.5.2 Limitations

Below is the list of limitations.

1. The total cost of the system must be less than $1500: Skeet shooting ranges have a limited budget and the client requested a low-cost solution.
2. The system must be portable: The system needs to be easy to move and battery powered.
3. The performance of the video analysis: Analysing video can be computationally expensive and can be processed more effectively in parallel. Such dedicated hardware can be expensive and can consume lots of power.
4. The 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.
5. The ground station must be protected: The ground station and its' camera need protection, as the system will be placed in an environment with live target fragments and shell casings.

2 Design Specifications and Analysis

This section details the IC Chip design specifications, including project requirements set by our client, and an analysis of the design.

2.1 Design Specifications

2.1.1 Functional Requirements

The functional requirements of the system are listed in this section. Please note that the functional requirements are split into two groups: The requirements of the mobile application and the requirements of the ground station.

2.1.1.1 Functional Requirements: Gemineye Mobile Application

Below is a list of the mobile application functional requirements. The expected use case of the mobile application is for the user to create a shooting squad and then start a shooting session. The user will then monitor and challenge the target classifications as they see fit. If the user wishes to challenge the target classification, they will request to review that particular shot's video footage. After reviewing the video footage, the user can then confirm the software's classification, or manually enter the classification.

The mobile application should...

1. Connect to the ground station wirelessly.
2. Not use mobile data.
3. Allow users to create a shooting session with multiple shooters.
4. Allow users to add a new shooter to an active shooting session.

5. Allow users to select the number of rounds per shooting session.
6. Allow users to track their scores during an active shooting session via buttons labeled *DEAD*, *LOST*, etc..., in addition to whether the clay pigeon came from the high house or low house.
7. Keep track of the score for every shooter in an active shooting session.
8. Display a target classification on the screen when received from the ground station.
9. Allow users to challenge the target classification and override the determined classification with a manual classification input.
10. Allow users to challenge the target classification and accept the determined classification.
11. Allow users to review the video of a target classification they wish to challenge.
12. Allow users to press a "Start Recording" button which will trigger the ground station to start recording video.
13. Allow users to press a "Stop Recording" button which will trigger the ground station to stop recording video, classific the video, and notify the mobile application with that target classification.
14. Display the shooting session members and scores on a scorecard that resembles the scorecards used in a Skeet Shooting game.
15. Update the scorecard as soon as it receives a target classification.
16. Know the rules of Skeet Shooting.
17. Allow users to exit out of the main page of the mobile application without losing the current session's session data.

2.1.1.2 Functional Requirements: Houston Ground Station

Below is a list of the ground station functional requirements. The expected use case of the ground station is for the user to place the ground station on the skeet shooting field and turn on the system. The ground station will act as a WiFi "hotspot" for the user's mobile device with a mobile application to connect to. Once the mobile device has connected to the ground station, the user will create a new shooting session and use the buttons within the mobile application to trigger when the ground station will start and stop recording video footage. When the ground station stops recording, the hardware will analyze and classify any targets in the video footage and send the resulting classification to the mobile application for display.

The ground station should:

1. Be portable.
2. Be a single unit with all hardware components, including camera lens and the computational computer.
3. Be accompanied with an easily removable protective "house."
4. Be able to capture and process video in real time.
5. Start recording video when it receives a "start" trigger message from the mobile application.
6. Stop recording video when it receives a "stop" trigger message from the mobile application.
7. Analyze and classify the target in the video upon stopping video recording.
8. Label a classified video with a unique ID and store it.
9. Notify the mobile application of a target classification upon analyzing and classifying a video.
10. Act as a "wireless hotspot" that the mobile device with the mobile application can connect to.

11. Be able to determine (i.e. classify) whether a clay pigeon target is dead or lost.

2.1.1.3 Select Use Cases

This section contains a few select use cases for the mobile application, based off the use case diagram below. The use case diagram depicts the main functionalities of Gemineye based on the client's needs.

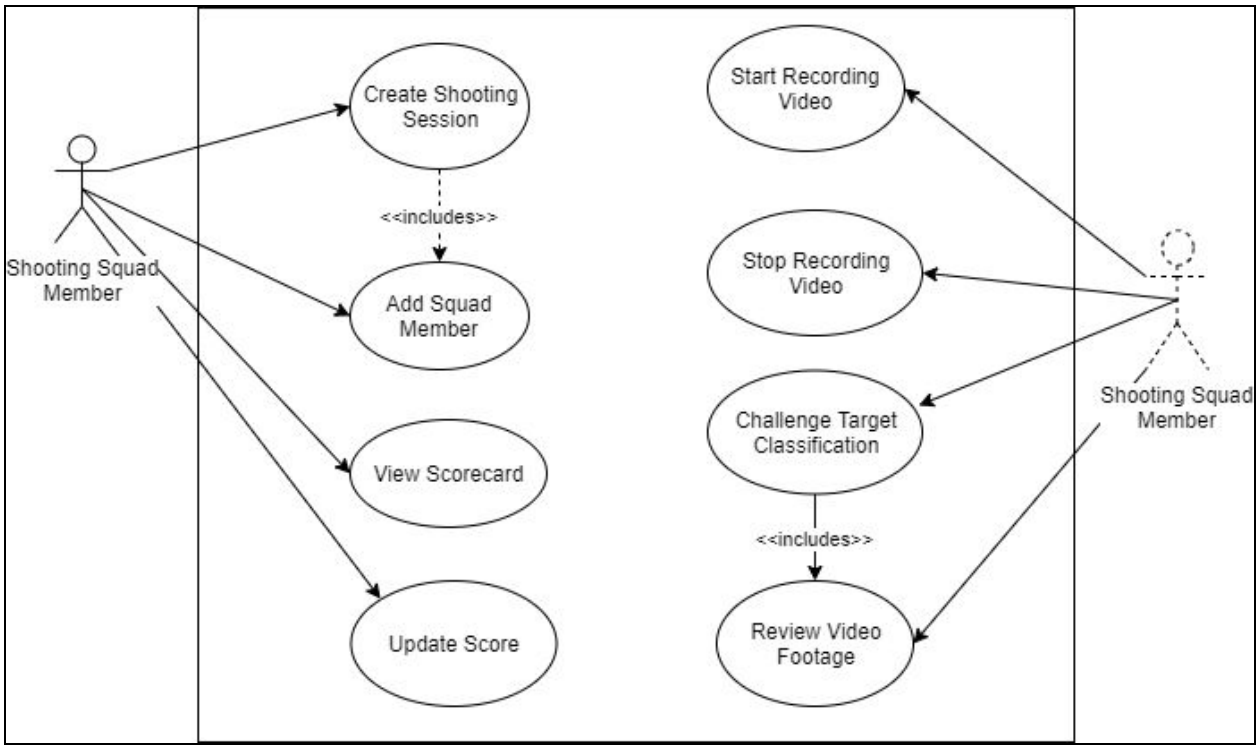

Figure 1: Gemineye Mobile Application Use Case Diagram

1 Gemineye: Create Session

Use Case: Create Session

Goal: Successfully create a shooting session with the squad members who will participate in the shooting session.

Primary Actor: Skeet Shooter [User]

Precondition: The mobile device has been successfully powered on and the mobile application has been successfully opened and connected to the ground station's WiFi hotspot.

Main Success Scenario:

1. User taps "Add Shooters" and types in the name of a squad member.
2. User taps "Done" and the name of the squad member is added to the session.
3. User repeats #1 and #2 as many times as needed to add all the squad members to the shooting session.
4. User taps "Continue" and selects the number of rounds in this particular shooting session.
5. User taps "Start Session" and is brought to the session scorecard.

Outcome: User successfully adds all squad members to the current shooting session and upon tapping "Start Session" is brought to the session scorecard, ready to begin the skeet shooting round.

2 Gemineye: Add Squad Member [To Active Session]

Use Case: Add Squad Member [To Active Session]
Goal: Successfully add a new squad member to an active shooting session.
Primary Actor: Skeet Shooter [User]
Precondition: The mobile device has been successfully powered on and the mobile application has been successfully opened and connected to the ground station's WiFi hotspot. A shooting session has been successfully created and started.
Main Success Scenario:

1. User taps "Add Additional Shooter" and types in the name of the new squad member.
2. User taps "Done" and is redirected to view the scorecard.

Outcome: User successfully adds a new squad member to the active shooting session and is redirected to view the scorecard, ready to continue with the round.

3 Gemineye: Update Score

Use Case: Update Score
Goal: Successfully update the score of an individual squad member during an active shooting session.
Primary Actor: Skeet Shooter [User]
Precondition: The mobile device has been successfully powered on and the mobile application has been successfully opened and connected to the ground station's WiFi hotspot. A shooting session has been successfully created and started. A squad member is about to shoot.
Main Success Scenario:

1. User watches a squad member take a shot at a clay target.
2. User taps the appropriate score button based on their classification determination of the shot (LOST/DEAD).
3. The scorecard is updated with the squad member's score.

Outcome: User successfully updates the score of the squad member who shot most recently, and the scorecard reflects this update.

4 Gemineye: Challenge Target Classification

Use Case: Challenge Target Classification
Goal: Successfully challenge the target classification of a squad member during an active shooting session.
Primary Actor: Skeet Shooter [User]
Precondition: The mobile device has been successfully powered on and the mobile application has been successfully opened and connected to the ground station's WiFi hotspot. A shooting session has been successfully created and started. A squad member has just shot and their score has been updated on the scorecard.
Main Success Scenario:

1. User and most recent shooter disagree on the classification of the shot.
2. User taps "Challenge Shot".
3. Video footage of the most recent shot is loaded and displayed on the screen.

4. User and shooter review the video footage.
5. User and shooter accept the original classification or manually input a new classification.
6. The scorecard is updated accordingly and the challenged shot is marked.

Outcome: User successfully reviews video footage of a challenged shot, and either manually inputs the new target classification, or accepts the original target classification. The scorecard is updated accordingly and the challenged shot is marked.

5 Gemineye: Start/Stop Video Recording

Use Case: Start/Stop Video Recording
Goal: Successfully trigger the ground station to start recording video footage.
Primary Actor: Skeet Shooter [User]
Precondition: The mobile device has been successfully powered on and the mobile application has been successfully opened and connected to the ground station's WiFi hotspot. A shooting session has been successfully created and started.
Main Success Scenario:

1. User taps "Start Recording".
2. A green light turns on on the ground station to indicate the ground station has started recording video.
3. The "Start Recording" button now displays the text "Stop Recording".
4. Once the user decides to stop the video recording, they tap "Stop Recording".
5. The green light turns off on the ground station to indicate that video footage is no longer being recorded.

Outcome: The user has successfully started and stopped video recording on the ground station.

2.1.2 Non-Functional Requirements

The non-functional requirements of the system are listed in this section. These requirements are related to the performance, reliability, availability, security, and usability of the mobile application and ground station. Again, please note that the non-functional requirements are split into two groups: The mobile application and the ground station.

2.1.2.1 Non-Functional Requirements: Gemineye Mobile Application

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 ground station stops recording video.
2. The mobile application will display the target classification within 1 second 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:

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

Data Integrity:

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

Usability:

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

2.1.2.2 Non-Functional Requirements: Houston Ground Station

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 video recording stops.
2. The physical device will notify the mobile application of the shot classification within 1 second after the device has classified the shot.
3. The physical device will send the most recent shot's video footage to the 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.

2.2 Proposed Design

This section details the IC Chip system design, including our proposed system design, conceptual sketch and architectural diagrams of the system, all system constraints, and a description of the modules that comprise the system and their functionality.

2.2.1 System Constraints

Below is a list of system constraints that our team had to keep in mind throughout the design process.

1. The physical device must be small and portable, as users may want to move the device to different locations on the skeet shooting field.
2. 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.
3. The physical device must be a low-cost device (i.e. < $1,500).
4. The physical device must not rely on an internet connection to perform its' computations.
5. The physical device must be battery powered.

6. The physical device must not rely on any outside/external power source.
7. The mobile application must not rely on the internet to display the target classifications.
8. The mobile application must not store videos in the mobile device's memory.
9. The mobile application must not collect or store any personal information.

2.2.2 Proposed Design and Design Objectives

This section details the IC Chip system design, including system conceptual sketches, system block diagrams, and the architecture of our system.

We begin by presenting the overall system conceptual sketch (Figure 2), with some of the system modules labeled with their respective technologies.

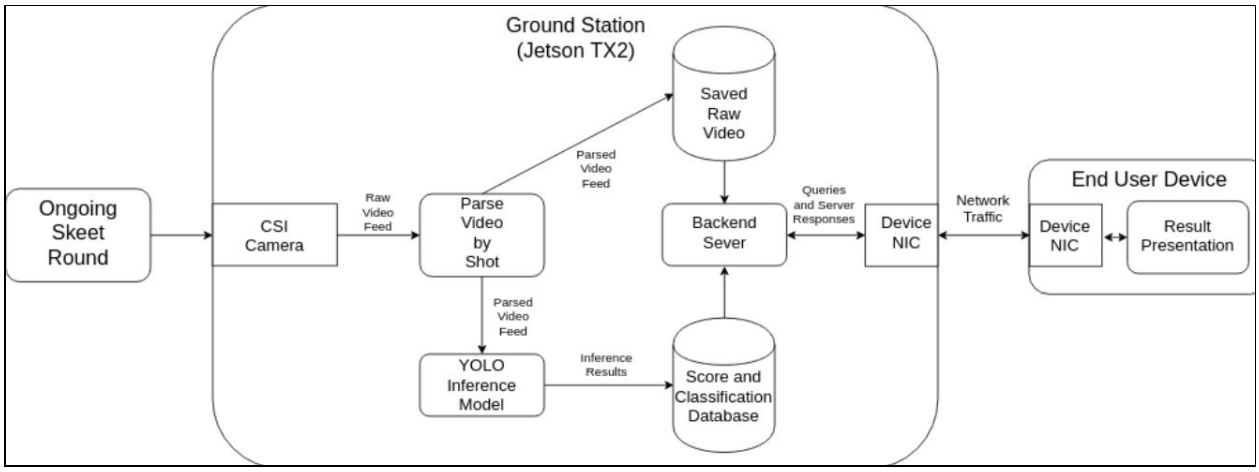

Figure 2: System Conceptual Sketch

The IC Chip design plan is as follows: First, the CSI Camera will record raw video footage of the ongoing skeet shooting round, as triggered by the mobile application. The raw video footage will be parsed for easier classification and the machine learning algorithm will analyze the video to determine the target classification. Note that the parsed raw video footage will be stored in a database. The target classification will be stored in its' own database, which the backend server will query when sending classification results to the mobile application for display to the user. The backend server will query the raw video footage database if the user challenges a target classification and requests to review the video of the shot they are challenging.

Throughout the course of this project and after multiple meetings with the client, our project scope changed. Originally, the ground station would determine when to start and stop recording video footage based off shot audio. However, it became clear to our team that this would become a "future work item," as we did not have enough time to research and implement this functionality. Our client decided it would be acceptable to have a "Start/Stop Recording" button on the mobile application which allowed a user to trigger when the camera would start and stop recording video footage.

2.2.2.1 Camera Design

We determine that the camera system solution will be based on a couple criteria: being able to capture color images at a very high resolution, able to have a wide field of view to see the entire field, having to be a standalone system rather than an off the shelf camcorder, being compatible with the Jetson TX2, and for being a cost effective.

From our initial data collection, we determined that our camera system needs to capture at higher resolution because of the problem of how small the clay target appears in the captured images. Using a higher resolution camera would increase the number of pixels that the clay target would be in the image which would help greatly with the determination of live or dead targets. We considered using a global shutter instead of a rolling shutter to help with getting a cleaner image of the clay target. A problem would arise from this change since we started training on images captured with a rolling shutter and then switching to a global shutter camera would invalidate our current model and training set. We also did some calculations that showed that the target would be moving too slowly for the shutter change to make a difference in the images. We also considered using a camera with a high speed capture figuring that would lead to cleaner looking images being captured and more images for analyzing. From the initial data collection, it showed a camera with 30FPS produced more than enough images to analyze.

For a single camera solution, it is essential that the camera lens would have a high enough FOV so that it would capture the entire field. We created a MatLab program that would help us visualize FOV of cameras to that of the field. When we collected our training data we used a camera that had a adjustable FOV, we set it to the maximum setting of around 70°. Figure 3 below is the FOV of the camera we used in the data collection at the locations where we captured (station two and station four).

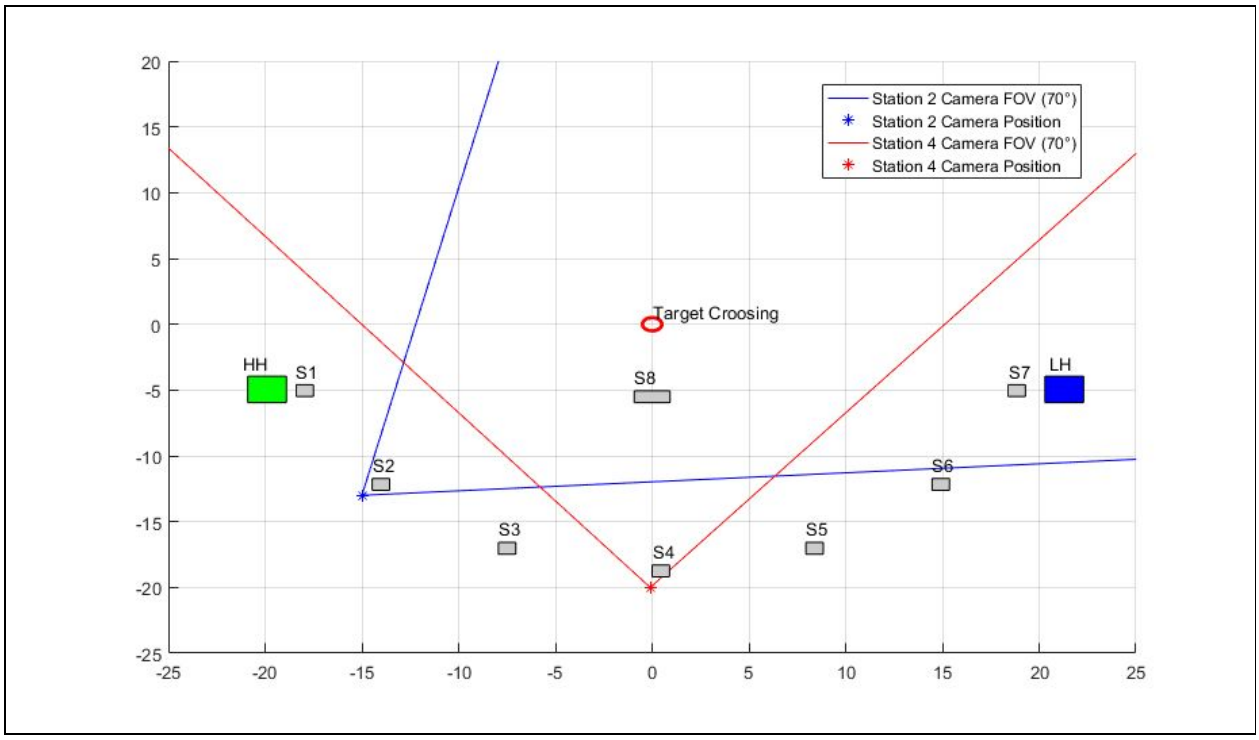

Figure 3: Camera FOV Simulation

Our initial results showed that the data collection camera had a desirable FOV, thus when we were selecting our camera systems we looked for cameras that had had equal or slightly better FOV. The problem that we could encounter if we selected a camera with a similar FOV is that the system would not capture either the high house or the low house as shown in Figure 3 above. This could cause a situation where the shooter would hit the clay target before it comes into view of the camera. In order to correct for this we could get a lens that extend the FOV but the trade off for this is that the clay target in the image would get distorted.

2.2.2.2 Powering the Ground Station

The Ground Station (Jetson TX2) is currently using a power adapter rated at:
Wattage: 60W
Voltage: 19V
Amperes: 3.16A

This power adapter is more than capable of providing enough power to the Jetson TX2 for all rated load capacities.

To optimize the portable mobility of the Ground Station, we have designed a battery powered system included with all the sufficient protection circuitry needed to protect the Jetson TX2 from the dangers a battery possesses.

Within our research we concluded that a LiPo 3s battery is the best available option to power the Jetson TX2. These batteries are not without their disadvantages. Some of these disadvantages are: 2-3 year shelf life, sensitive to high temperatures, overcharging and overheating can result in fire, and very specific charging instructions. As the LiPo 3s battery may have its downsides, there are a variety of reasons why this battery type is the best choice. These reasons are as follows: LiPo 3s batteries are lightweight, hold their charge for a long time when not in use, have a high discharge rate, and high capacities allow them to hold much more power than other battery types. If the user takes the necessary safety precautions described by the LiPo 3s Battery manufacturer, then these batteries are a safe and reliable option to power our Ground Station (Jetson TX2).

There are two important circuits used for Battery/Ground Station Protection. In Figure 4 you can see the addition of a fuse in series between the battery and the Ground Station. This fuse is used to protect the Ground Station (Jetson TX2) from an influx of current higher than the maximum rated input current of the Jetson TX2. A 6A rated blade fuse can be used in this case, if the current is greater than 6A for a short period of time the fuse will blow and protect the Jetson TX2 from OverCurrent.

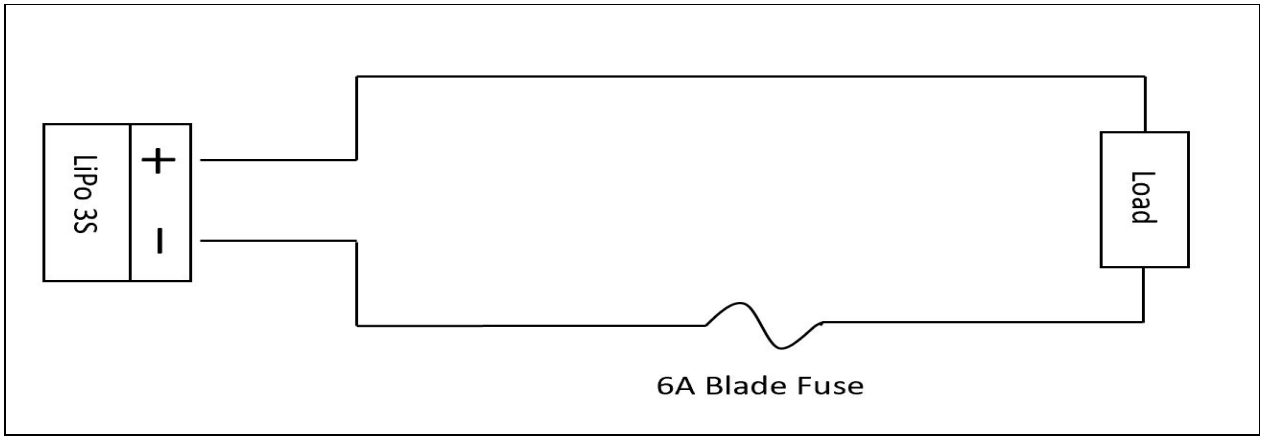

Figure 4: Fuse Protection Circuit

The next protection circuit is used to protect the battery from excessive voltage discharge. It is important to not let the voltage of each cell within the LiPo 3s battery to drop below 3.5 Volts. The circuit in Figure 5 will protect the battery from this condition by not allowing it to drop below 3.5 Volts.

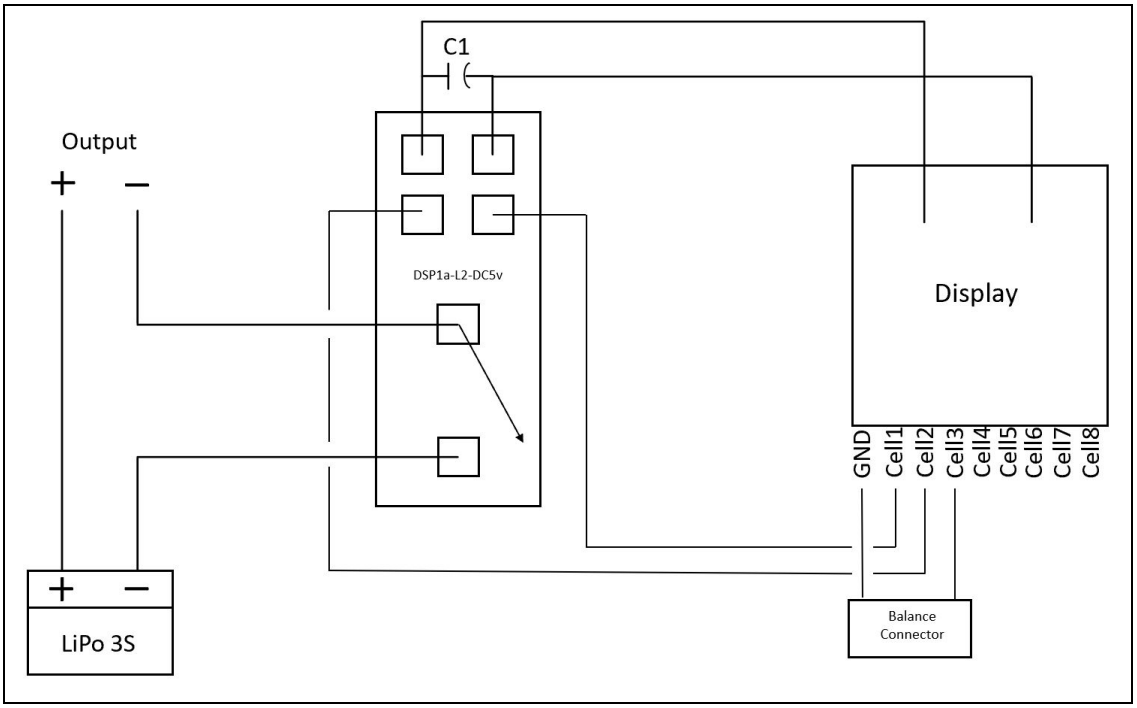

Figure 5: Voltage Discharge Protection Circuit

The addition of these protection circuits and 2x 5000mAh LiPo 3S batteries will enable the ground station (Jetson TX2) to run up to 8 hours without needing to connect to an electrical power outlet. This will eliminate the use of extension cords which could be a tripping hazard on the shooting range.

2.2.2.3 Mobile Application and Ground Station Communication

Communication between the mobile application and the ground station is done through http requests. The ground station has a web server with an API used to return data or to control the camera. The Figure
mobile device connects to the ground station via WiFi, the ground station will act as an access point for the mobile application to connect to so that both devices are on the same network. The mobile device will have controls the can be used to start and stop the camera and to retrieve data.

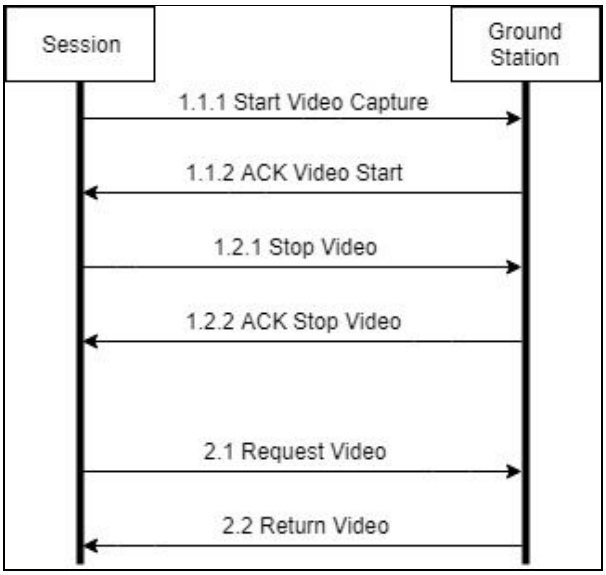

Figure 6: Communication Diagram

2.2.2.4 Mobile Application Design

The mobile application is an Android application written using the Xamarin.Android that is built using a C# .NET framework. The Xamarin.Android framework itself is structured similarly to a native Android application. Our application, Gemineye, was designed with two main activities “Main Activity”, and “Shooting Session Activity”. The activities then use fragments to keep ui design modular. The “Main Activity” was built to be a landing page. Once the application supports viewing old or previous sessions, continuing previous session, and other additional features this activity can be used as a main menu of sorts. The “Shooting Session” activity is where a majority of the application currently lays. The Shooting Session is where the user will handle everything in a shooting session. This includes tracking shot outcomes, contesting videos, adding additional shooters, checking scores, checking which shooter is up, and other future features.

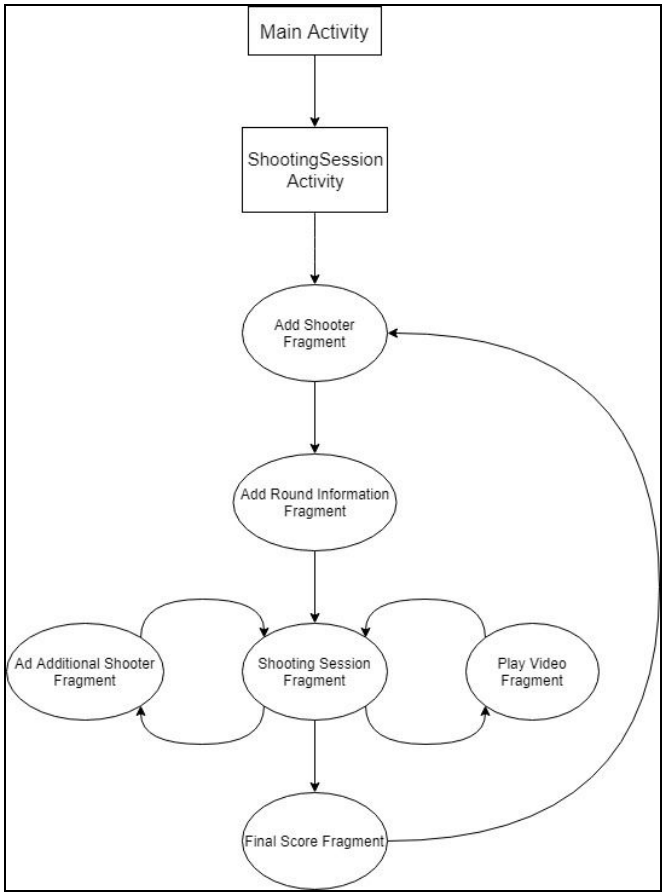

Figure 7: Application UI Flow diagram

2.3 Design Analysis

For this project the main components include the Jetson tx2 board, the 13 megapixel camera, and the mobile device. The main functions of our system where to capture high quality video, to use the video to classify data, and to provide a front-end interface for the user. The Technologies were chosen to suit the needs of the client.

The Jetson tx2 board was used as the ground station for our project. We chose the Jetson tx2 board for its graphical processing capabilities and it's low power cost. The board came with a development kit with a lot of different ports such as the CSI MIPI interface that can be used for a camera.

On the ground station, a 13-megapixel camera, the e-CAM130 CUTX1 from E-con Systems. The camera chosen was picked by our client. The camera uses the CSI-2 MIPI interface which works directly with the image service processors on the board. The board has 12 CSI-2 Lane's that each have a bandwidth of 2.5 Gbps/lane. For its category had a fairly low cost. The reason for choosing this camera was to have a low-cost camera that's able to capture high resolution video and work directly with the hardware on the Jetson board.

For the mobile device, we chose the Nexus Tab 7. We decided to use a tablet for the device to make the mobile application easier for the user to operate. We went with the Nexus Tab 7 for its availability and its size.The size was a factor in that we didn't want a tablet that was too small or too big and clunky to use and decided that a 7 inch screen tablet was what's best for its job.

2.4 Development Process

Our team used Test Driven Development (TDD), or testing as a development process for both the mobile application development and the hardware implementation. TDD involves short sprints where team members create test cases for requirements, implement the specific requirement, and then complete testing to ensure all the new tests pass. As TDD follows short sprints, the development process is done in multiple cycles, where the software is changed to pass the new test cases only.

The IC Chip project team felt that TDD offered more benefits to our project than other development processes, such as Behavior Driven or Waterfall. TDD requires that teams write the tests before development begins and only write code that passes those tests. This allows our development team to focus on one (or a few) requirement(s) at a time with detailed specification. More tests are written, with a higher code coverage, and it is easier to spot and fix any bugs.

The mobile development team followed the TDD process closely. As an example, take the mobile application functional requirement “allow users to add a new shooter to an active shooting session.” Before this requirement was implemented, the mobile application team created test cases to verify the requirement was met. The test cases included ensuring the scores on the scorecard were not changed upon the addition of a new shooter, the order of the squad members was correct, and the new squad member was placed at the end of the list of active squad members. Once the test cases were created, development on this requirement began. No other requirement was implemented during this particular sprint. Upon development completion, the application was tested with the new use cases to ensure the software improvement met the “Additional Shooter” requirement.

3 Implementation

3.1 Implementation Details and Challenges

This section discusses implementation details, including the technology stack used, any implementation challenges our team encountered, and the standards our team followed.

3.1.1 Mobile Application

This section details the technologies selected for mobile application development, in addition to implementation details.

3.1.1.1 Development Framework Selection

When our team first started research different options for Android development frameworks, Keith stumbled across the Xamarin.Android framework. This framework allows developers to build native Android applications using C# or F# in Visual Studio. Since members of the development team were familiar with C# and the .NET framework and Visual Studio, Xamarin.Android seemed like a good choice.

In addition, Xamarin.Android framework supported Fragments/Fragment Managers and Activities, which help create a modular package: Fragments are thought of as “mini-UI bundles” that can be reused throughout an application. This was an important deciding factor to our team, as we wanted to write modular code that was easy to update, maintain, and understand. Lastly, Xamarin.Android provided an easy way to display videos in a mobile application, which was one of our key client requirements. For these reasons, our mobile application team selected Xamarin.Android as the development framework.

3.1.1.2 Tablet Selection

As noted earlier, one of the key requirements of the mobile application is the ability to watch video footage of [small] clay pigeons flying across the screen. Keeping this in mind, our team decided we need a tablet with a 1080p minimum. Iowa State University has Nexus Tab 7 devices available for students to use for free, which was important to our low-cost solution requirement--especially for testing purposes. The Nexus Tab 7 has a resolution of 1900 x 1200 which passed our resolution minimum. The screen dimensions are ~8” x 4.5”, which allows a user to easily see the clay pigeon in video footage. Lastly, the Nexus Tab 7 runs on Android 6.0.1, which is the minimum version of our mobile application.

3.1.1.3 Mobile Application Development

One issue we ran into while developing the mobile application was in implementing a local db framework called IridiumDB. The framework would’ve allowed us to easily implement future features by providing a local db thats schema was controlled by our Xamarin models. This would make saving previous sessions, saving ongoing sessions, user profiles, analytics possibilities, and more. The database was discovered in the middle of the second semester, but after struggling to get a stable application while using the framework we decided to postpone the implementation of IridiumDB and continue on to other features. While the time was lost on our project, we hope the partial implementation will help speed up future development.

3.1.2 Ground Station

This section details the technologies selected for ground station development, in addition to implementation details.

3.1.2.1 Ground Station Hardware

Nvidia Jetson TX-2: The Jetson was originally selected due to the low cost through an educational program by Nvidia. It allowed us to obtain a single development board with a variety of ports and modules for expansion in addition to the TX-2 module itself. The module contains 256 Pascal CUDA cores allowing for a tremendous speedup in computer vision applications. The development board measures 10”x10” and has a peak power draw of ~35 Watts allowing for the possibility of making the system battery powered.

AlexeyAB’s YOLOv3[1]: The You Only Look Once, version 3 computer vision algorithm was originally examined for the project as it is the current state-of-the-art object detection algorithm. It is capable of performing inference in real time by directly reading from a file pointer in in a **nix* operating system potentially limited hardware. Rather than choose the original fork of the algorithm, AlexeyAB’s

implementation was selected due to the more active support of the project by contributors and several useful modifications and configurations to ease deployment and testing.

Cartucho's mAP[2]: Mean Average Precision is a useful metric for comparing the overlap between the bounding boxes in the labeled testing data and the data which has been labeled by the trained model. This allows some insight into the overall performance of the model. Cartucho's mAP implements this and several other metrics to understand the overall accuracy and performance of our model.

3.1.2.2 Ground Station and Mobile Application Communication

Communication between the mobile application and the ground station is done through http requests. A web server with a RESTful API is set up on the ground station using the flask-RESTful framework. The mobile device connects to the ground station via wifi. The mobile app makes request to the API to control the camera and to retrieve data. The mobile app starts the camera recording by a click of a button and request is then made to the ground station. The ground station then accepts the request and a script gets executed to start recording with the camera. The same process happens to stop the camera from recording, a request is made and a script is executed to stop the camera. Once the camera stops recording, the video gets analysed and data is saved to files. For the mobile application get data such as bounding box info, the status of the target, and so on, the mobile application will make a request to the API and data in the form of a JSON string or another file.

3.1.2.3 Camera System Selection

The camera selected for IC Chip is the e-CAM130 CUTX1, a 13MP color camera developed by e-con Systems made specifically for the Jetson TX2/TX1. This camera system is able to capture color video with a rolling shutter at a resolution of 4192(H) x 3684(V) at 20FPS and at lower resolutions being able capture at 72FPS. The default camera system is capable of having a field of view of 74° but the camera module has a standard S-mount bracket which allows for the changing of the lens for different positions. This is a standalone camera system that attaches directly to the Jetson TX2 through the 4-lane MIPI CSI-2 interface on the development kit. The software allows complete control of the camera capture settings including exposure, digital zoom, and brightness.

3.2 Data Collection and Challenges

The largest hurdle when creating any sort of supervised machine learning model is a labelled dataset. The use for this is twofold: firstly to train the model itself and secondly to provide some performance metric of the model's inference which is indicative of its performance. To this end, the project was left with the monumental task of producing its own dataset, as nothing similar existed previously.

In an ideal case, such a dataset exhibits the target in a variety of conditions. For computer vision specifically this could be orientation, lighting, angle, or rotation. Such diversity of information is aimed at the model learning the underlying semantic representation of the object to be detected while also avoiding overtraining by relying on specific, repeated features that do not exist in the natural distribution. However in the event that these cannot be satisfied, they can be augmented in the training of the model (see: 3.3 Model Training and Challenges).

To this end, the team on two separate occasions took several hours of recordings on live skeet ranges from a variety of positions around a standard skeet-shooting field. Appropriate precautions were taken with the camera system to protect it from range debris. The videos were then parsed using the **nix* tool *FFMPEG* to create images of still frames in the lossless *.PNG* format. Individual frames were produced, post-processed, and hand-labelled over the following several weeks. This allowed the creation of 8,795 useful images featuring at least one clay target labeled *LIVE* or *DEAD*. Finally, the dataset was formed into two disjointed sets with approximately even distributions of the two classes for testing (2166) and training (6639).

3.3 Model Training and Challenges

Training the model also proved to be a challenge. As Iowa State University is tremendously behind the curve in terms of modern machine learning, we lacked the resources necessary to produce useful models in a timely fashion. As a result of this, we instead were forced to use our end-device to instead train the model. This significantly increased the amount of time to train the model due to the limited hardware and negatively impacted our ability to tune hyperparameters which had a significant impact on overall model performance. The increased training time also presented issues when the system lost power on nearly half a dozen occasions, causing us to lose work as the weights trained up to that point could not be saved. The total training time for each iteration of the model took in the range of 5-9 days.

There was also difficulty in selecting a network architecture that fit on the Nvidia Jetson TX-2 board to be used as the groundstation when running. Due to the compiled nature of AlexeyAB's YOLOv3 implementation as a C/C++ project this was an ideal fit as it could be configured to fit within just 4GB of memory. The algorithm was capable of training some initially provided weights on a custom dataset.

With our dataset complete, we began training weights. Several configurations were made to supplement the data to help train a model that was more robust do things such as camera orientation, lighting, and image resolution. These included randomly rotating select images during training batches, downsampling, and decreasing/increasing the pixel intensity of images.

Unfortunately, our second and third models displayed less-than-adequate performance, but were an improvement over the first which was simply a proof-of-concept on a few hundred images. The second model was due to it only being trained on a limited subset while the team finished the data labeling. The third's lackluster performance was eventually revealed to be due to several dozen images labeled as *live* and *dead* instead of *Live* and *Dead*. Due to their being included in the training set, but not containing any of the labeled classes, they were triggering weight updates as the fork of YOLOv3 we were utilizing could only train on positive examples (where the object is present).

The fourth and fifth models showed incremental improvement in performance. The fourth's improvement in performance appears to be attributable to the removal of mislabeled data. It was shown to be accurate in classifying ~75% of all *Live* and *Dead* targets but was still below the goal of ~90%. The fifth and most current model was simply trained for 5000 more iterations than the previous

model on the same set after it was realized that the loss function continued to decrease noticeably. The final accuracy was 91.4% on *Live* targets 77.4% on *Dead* in the training dataset. More information on the outcome of testing the final model can be found in section 4 Testing.

4 Testing

This section details the testing plan for our system, including information about our automatic testing suite for Gemineye, select test cases for both Gemineye and Houston, and the results of those tests.

4.1 Mobile Application Test Plan

Automatic Unit Testing (XUnit):

Test Name	**Activity/Fragment**	**Validates**
OpensShootingSession	MainActivity	Checks that ShootingSession is opened when 'Start Session' button is pressed
AddsShooter	AddShooterFragment	Checks that a user is added to Shooter List when entered
ContinuesWithShooters	AddShooterFragment	Checks that List of Shooters is passed to Round Info Fragment
CorrectlyGetsRound	AddRoundInformationFragment	Checks that when 'n' rounds are selected the session is set to 'n' rounds.
GoesToAddShooter	ShootingSessionFragment	Checks that when 'Add Shooter' is clicked AddAdditionalShooterFragment is opened.
GoesToContest	ShootingSessionFragment	Checks that PlayVideoFragment is opened when shot is contested.
AddsShot	ShootingSessionFragment	Checks that a shot is added to the correct user when kill/loss is pressed.
ChangesStationCorrectlyTwoShot	ShootingSessionFragment	Checks that the Session correctly manages stations with two shots per.
ChangesStationCorrectlyFourShot	ShootingSessionFragment	Checks that the Session correctly manages stations with two shots per.

Hand Testing:

After each merge of code into the master branch the use cases are tested by hand on the application and bugs are logged to be fixed. This ensures that the main uses of our app are working at all times and that our main branch is stable.

4.1.2 Model Testing

Testing for the YOLOv3 model focused on utilizing 2166 labeled images that the model had not yet encountered which were selected from the full dataset with a normal distribution. These images contained approximately even ratios of *Live* and *Dead* labeled targets as the training dataset. These images were then fed into the trained model to performance inference. The output of the inference was redirected from *STDOUT* to a text file which was then parsed by Cartucho's mAP implementation to produce performance metrics.

4.2 Testing Results

This section includes a discussion about the evaluation criteria for our project, including performance metrics and test results.

4.2.1 Mobile Application Test Results

The mobile application currently has 100% of its tests passing. While the testing suite is not large, we believe it covers our applications main use cases and ensures that the application is bug free for its critical features.

4.2.2 Model Testing

The final accuracy was 91.4% on *Live* targets 77.4% on *Dead* in the training dataset which is reflected in Figure 8 The Mean Average Precision on the training set is in Figure 9 which indicates that the model produced bounding boxes which were on average slightly larger than those in the dataset, but this is to be expected as the labeled bounding boxes were drawn as precise as possible. This together with the high overall accuracy implies that the model was reasonably successful in its inference. However it should be noted that on average the bounding boxes for *Dead* targets were typically smaller when examined directly.

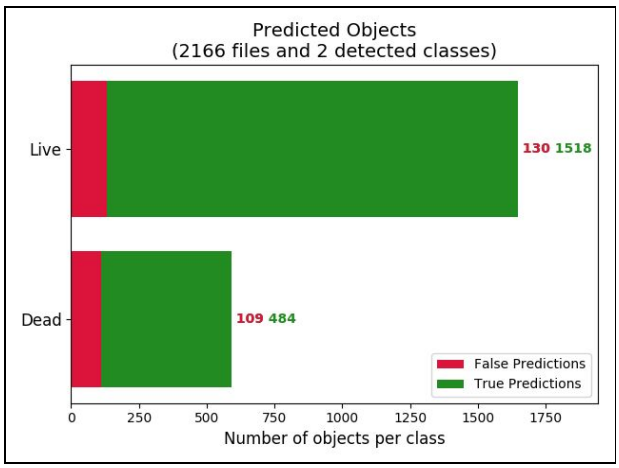

Figure 8: Training Dataset Accuracy

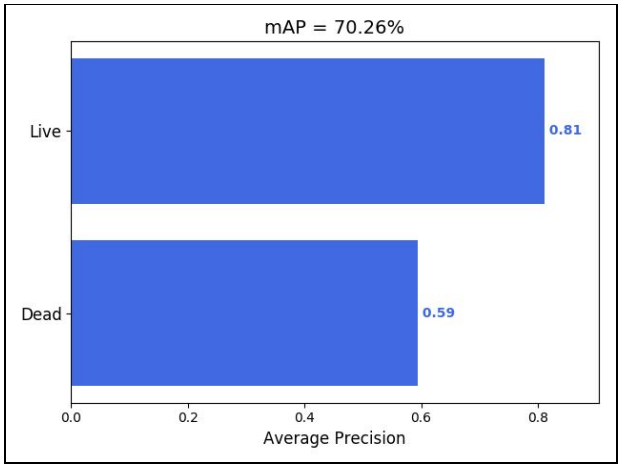

Figure 9: Mean Average Precision

5 Project Timeline and Risk Management

5.1 Project Timeline

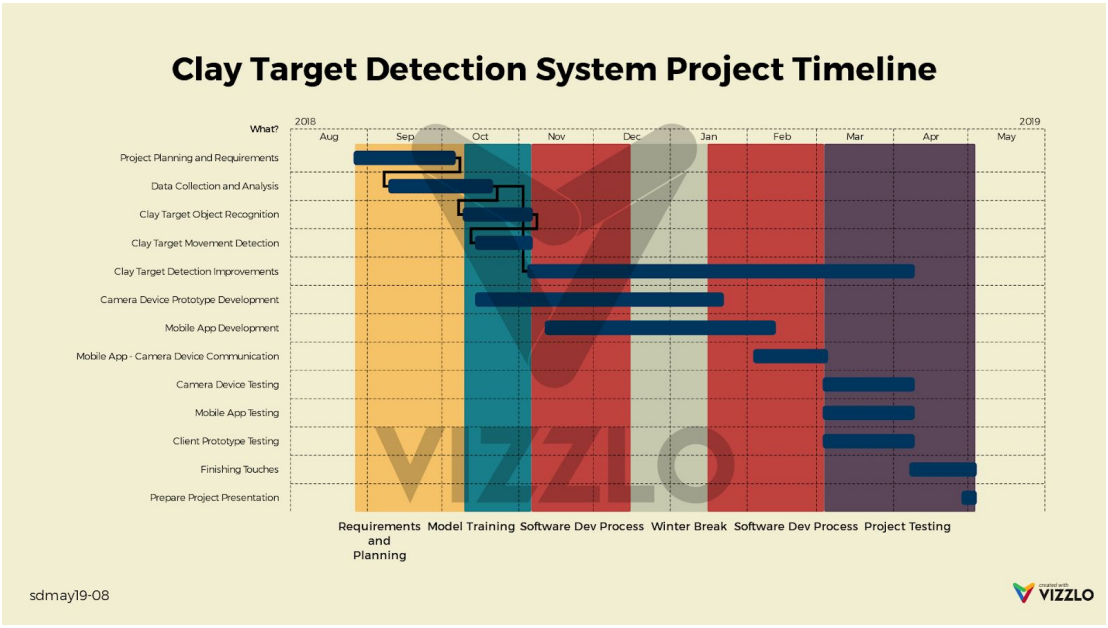

Figure 10: Proposed Timeline

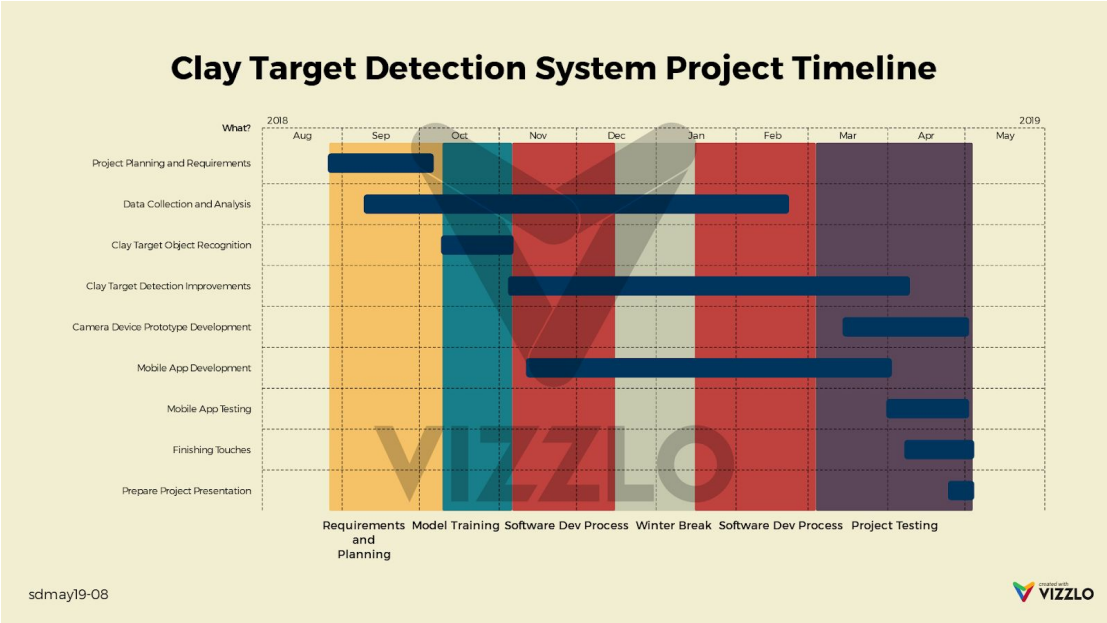

Figure 11: Actual Timeline

5.2 Project Task Decomposition & Personnel Responsibilities

Task Decomposition:

- Data Collection
    - Personnel: Cole Huinker, Steven Sleder
    - Description: Data collection was done to have data that can be used to train our classification model. This involved going out to the skeet field and recording footage of clay targets both live and dead.
- Data Labeling
    - Personnel: All team members
    - Description: This included splitting up videos from our data collection and labeling those video with their contents. The videos were then also split into into individual still frame and each frame with a target or targets has bounding boxes drawn around the targets and classified on whether the target is live or dead.
- Model Training
    - Personnel: Steven Sleder
    - Description: This involves taking labeled data and training it with darknet.
- Hardware Backend Development
    - Personnel: Cole Huinker
    - Description: This includes developing the pipeline on the backend of system which includes communication between the jetson board and the mobile device and analysing classified data that is sent to the mobile application.
- Hardware Testing/Performance
    - Personnel: Cole Huinker, Steven Sleder

    - Description: This dealt with testing the performance of the YOLO model on the board along with testing the back-end system that communicates between the camera and the mobile device.
- Hardware Specifications
    - Personnel: Mike Ruden, Steven Sleder, Philip Hand
    - Description: In charge of research and decisions made on hardware such as the Jetson board and the camera used with the board. Factors such as cost, power consumption, and performance were used for determining the hardware.
- Mobile app development
    - Personnel: Eva Kuntz, Keith Snider
    - Description: Development of the mobile application includes creating sessions for the user, keeping score for the game, and designing the UI for the end user.
- Mobile app testing
    - Personnel: Eva Kuntz, Keith Snider
    - Description: This includes testing session, challenging incorrect readings the fetched data, testing features throughout development, and etc.

Responsibilities:

- Steven Sleder:
    - Computer Vision
    - Machine Learning
- Cole Huinker:
    - Data Collection
    - Backend development
- Keith Snider:
    - Software Architecture
    - Mobile Development
    - Testing Suite
- Eva Kuntz:
    - Project Manager
    - Software Architecture
    - Mobile Application Development
- Michael Ruden:
    - Camera Specification
- Philip Hand:
    - Power Lead

5.3 Project Risks and Mitigation

Below is a table of Risks and mitigation strategies associated with our project.

Risk	Description	Mitigation Strategy
Model Underperformance	The classification model is not accurate enough to meet standards.	Acceptance, take whatever the outcomes are.
Model over training	Training the model on data that is too similar	Mitigate, try to diversify the training set.
Hardware-mobile app Communication issues	The mobile device is unable to connect or communicate with the ground station.	Mitigate, find the simplest way to communicate among devices.
Hardware Malfunction	The Jetson tx2 board or any of the hardware components doesn't work for any reason.	Avoidance, choose hardware from a trusted supplier.
Incompatible hardware	Any hardware or components that don't work with or not easily compatible with the system.	Avoidance, make sure that all components being considered are check to see if they are indeed compatible.
Miss classification	The ground station sends miss classified data to the mobile application.	Acceptance, it will happen once in a while, add a feature that challenges a classification.

5.4 Lessons Learned

This section details some of the lessons our team learned throughout the design and development of IC Chip.

First, team communication is extremely important. Although our team knew this at the beginning of the project, it became apparent just how vital it is to communicate what task(s) a team member is working on, if they are stuck, and what they plan to work on next to the entire team. Without clear communication, tasks may be duplicated and the work is completed twice. Unfortunately, this happened to our project team during data collection and labeling. Some team members had more time than others had, and were able to label more than their assigned videos. Although this appeared to be a time saver, those team members were not clear in their communication about which videos they had labeled, which resulted in people labeling the same videos twice. After this discovery, our team worked to improve communication to avoid duplicating work and ensure tasks were split evenly between team members.

Second, it is extremely important to plan for delays in design, development, and testing. As noted in section 3.2, data collection and labeling took longer than expected. Our team originally thought it would take a maximum of 2 months to collect and label all the data. Instead, it took almost an entire semester to finish labeling all the data our team collected (this may have taken a shorter time if team members did not duplicate labeling work). Unfortunately, this put a huge delay in mobile application and hardware development, as team members focused solely on labeling data. Instead of continuing to

focus on labeling data, our team should have consulted the project timeline and assigned personnel to other tasks to reduce the development and testing delays. The important takeaway to note is that project teams need to be aware of their project timelines, and, if the project is behind, the team needs to discuss how they can get back on track with their timeline.

6 Conclusion

Overall, the project was relatively successful but had several shortcomings due to unforeseen difficulties. The largest of these was by far collection, labeling, and sanitization of the dataset necessary to produce the computer vision model. Therefore, from the beginning it was impossible to know whether or not a it was currently possible to even construct a satisfactory model. This acted as a hindrance on almost every other aspect of the project. If we don't know if we even can construct the model, how do we determine our hardware requirements? Without any known hardware to work with, how will our system components interact?

Due to this, the project could not proceed immediately and a tremendous portion of the team's total hours instead was focused on the decidedly non-engineering task of drawing bounded boxes for hours on end. This directly and negatively impacted the team's ability to work on other components of the project. It was only after the construction of a final dataset and the training of several test models during the second semester of the project that development and integration took place. During this time it was decided that the end user device would also be capable of functioning purely on its own as a scoring device, without any inference or information from the ground station, but remain extensible to add this functionality in the future as an end-of-term deliverable.

To this end the project is coming to completion with a hardware package of the Nvidia Jetson TX2 with the mounted e-CAM130 CUTX1 camera module and a trained YOLOv3 model capable of near-satisfactory performance capable of directly reading frames from the camera and performing inference and storing results to be fetched by the mobile application. The second component is a user-facing Android application for creating and scoring a round of Skeet manually, with hooks in place to connect to a back-end server to fetch inference results.

What remains to be done to bring this project to completion is integration of the end-user application with the ground station by means of a back-end server that simply provides the results of the inference after fetching them from an on-board database. In addition, there is room for improvement in the model's inference through supplementing the training data with additional images, testing it for performance in the field, and devising a way to algorithmically handle and score multiple clay targets present in a single frame. Though they may sound simple, these goals are all non-trivial and vital to the project coming to full fruition. Unfortunately, they rapidly became unobtainable due to the time constraints caused by spending a positively egregious amount of time on collecting, labeling, and sanitizing a dataset to determine if the project was even possible to begin with.

References

[1] AlexeyAB, *darknet*. GitHub, https://github.com/AlexeyAB/darknet, 2013
[2] Cartucho, *mAP*. GitHub, https://github.com/Cartucho/mAP, 2018

Team Information

Below is the contact information for each team member.

Eva Kuntz, SE
Phone: (608) - 438 - 0093
E-mail: evakuntz@iastate.edu

Steven Sleder, CprE
Phone: (402) - 651 - 9953
E-mail: ssleder@iastate.edu

Keith Snider, SE
Phone: (515) - 451 - 4102
E-mail: krsnider@iastate.edu

Cole Huinker, CprE
Phone: (507) - 236 - 6920
E-mail: chuinker@iastate.edu

Michael Ruden, EE
Phone: (712) - 541 - 2169
E-mail: mjruden@iastate.edu

Philip Hand, EE
Phone: (515) - 494 - 7123
E-mail: pwhand@iastate.edu