Distributed Wild Bird Surveillance and Recognition System

Project Plan

Team Number: SDMAY19-10
Client: Dr. Joseph Zambreno
Advisor: Craig Rupp

Ben Simon - Meeting Facilitator and User Interaction Lead
Claudia Athens - Team Communications Lead and Systems Integration Lead
Francisco Arreola - Meeting Scribe and Infrastructure Lead
Pierce Adajar - Repository Wrangler and Machine Learning Lead

Team Email: sdmay19-10@iastate.edu
Team Website: http://sdmay19-10.sd.ece.iastate.edu

Revised: December 2, 2018 / Version 3

Table of Contents

I. Introduction **4**
1. Acknowledgements 4

II. Project Statement **5**
1. Problem Statement 5
2. Operational Environment 6
3. Intended Users and Uses 6
4. Assumptions and Limitations 6

III. Project Deliverables and Specifications **8**
1. Deliverables and Functional Requirements 8
   1.1 Hardware 8
   1.2 Detection and Classification 8
   1.3 Data Streaming and Storage 8
   1.4 User Notifications and Frontend 8
2. Non-Functional Requirements 9

IV. Previous Work / Literature Review **10**
1. Existing Solutions 10
2. Relevant Literature 10

V. Proposed System / System / Solutions **11**
1. Hardware 12
2. Detection and Classification 12
3. Data Streaming and Storage 12
4. User Notifications and Frontend 13

VI. Assessment of Proposed Solution **14**
1. Hardware 14
   1.1 Strengths 14
   1.2 Weaknesses 14
2. Detection and Classification 14
   2.1 Strengths 14
   2.2 Weaknesses 14
3. Data Streaming and Storage 14
   3.1 Strengths 14
   3.2 Weaknesses 14

3.3 Trade-Offs 15
4. User Notifications and Frontend 15
4.1 Strengths 15
4.2 Weaknesses 15

VII. Validation and Acceptance Test 16

VIII. Project Timeline 17
1. First Semester Goals 17
2. Second Semester Goals 18

IX. Challenges: Risks / Feasibility 20
1. Feasibility and Challenges 20
2. Risks 20

X. Standards 22

XI. Test Plan 23
1. Interface Specifications 23
2. Hardware and Software Testing Tools 24
2.1 Hardware 24
2.2 Software 24
3. Functional Testing 25
3.1 Unit Testing 25
3.2. Machine Model Testing 25
3.3 Hardware Enclosure Testing 25
3.4 Interface Unit Testing 25
3.5 Integration Testing 26
3.6 System Testing 26
3.7 Acceptance Testing 27
4. Non-Functional Testing 27
5. Process 28
6. Simulation and Modeling 28
7. Results and Challenges 28

XII. Conclusions 30

XIII. Figures and Tables 31

XIV. References 32

I. Introduction

1. Acknowledgements

SDMAY19-10 would like to thank the Department of Electrical and Computer Engineering for their continued support of student's professional and technical growth through the administration of this course. SMAY19-10 would also like to thank our client, Dr. Joseph Zambreno, for the financial resources provided and his continued support of this project. We would also like to thank Craig Rupp, our advisor, for his guidance as we develop this project.

II. Project Statement

1. Problem Statement

Bird identification is a more complicated topic than one would think. As an example, can you identify the species of the four following birds?

Fig. 1: Four North American Birds[1]

In the clockwise order from the top left, the birds are a male cardinal, female cardinal, another female cardinal, and a juvenile cardinal. Four seemingly different birds, but they all happen to be the same species. Now imagine these differences happening for all the birds across North America. Because of this, even for a hobbyist, bird identification can be challenging.

The task at hand to provide a solution to this problem is to create a self-contained system that can detect and classify bird species in realtime in our client's backyard. The current consumer products available are limited to nature cameras that are constrained by basic video streaming and photo capture capabilities that are not able to classify birds.

To do this, we will use an embedded GPU platform with a convolutional neural network for real time object detection and classification. Our model will be trained and tested with two already available North American bird datasets: NABirds V1 and Caltech-UCSD Birds 200. As a result of this project, we will have delivered a well-documented HW/SW product to our client that will successfully identify North American birds in real time.

2. Operational Environment

The primary operational environment will be in our client's backyard in Iowa. Because of this, it will have to withstand Iowa's normal weather year-round. It will be exposed to dusty, windy, and wet conditions with a large temperature range. Our design will have to include some type of enclosure to prevent the damage of our project.

3. Intended Users and Uses

The primary user of this product will be our client, Dr. Joseph Zambreno. The intended end use of this product will mostly be between the client and our web interface which will allow the client to view the video stream, the captured photos, and the birds classified. Little to no interaction between the client and the actual product will be necessary unless he would like to change the position of the product within his backyard.

4. Assumptions and Limitations

Assumptions:

- The operating environment will not be in extremely hazardous weather or conditions.
    - Normal weather conditions include temperatures ranging from 40°-80° F, and precipitation not exceeding 10 millimeters an hour.
- The end product will only be used in the client's backyard while stationary.
- The client will have access to the Internet in order for the product to access the web interface.
- The system will have access to the client's Wi-Fi and power.

- Only one user will view the video steam at a time.
- The system will only need to classify birds commonly found in North America.
- The system will only need to detect and classify birds in well-lit conditions.

Limitations:

- The datasets we will be training our model on are professional-level image quality.
- The system is limited to the computing power of the NVIDIA Jetson TX1.
- The case of the system must be small enough to be attached on a deck railing.
- The system is constrained by the client’s bandwidth.
- The project will have a budget of $1500.

III. Project Deliverables and Specifications

The four core deliverables for the end product are the hardware system, the detection and classification system, the data streaming/storage system, and the frontend and user notification system.

1. Deliverables and Functional Requirements

1.1 Hardware

The hardware will facilitate data streaming and bird detection in real time. This system will consist of a Jetson board, 4k camera, and internet connection. The 4k camera will stream real time video from a bird feeder located in the clients yard. The system will be robust enough to withstand Iowa weather and provide a stable camera mount for a clear video stream.

1.2 Detection and Classification

The detection and classification systems will be running on the sourced hardware. This system will utilize neural network architectures such as YOLO[2] to locate and identify birds in the field of view. The detection and classification will run in real time and be able to identify all of iowa's native bird species.

1.3 Data Streaming and Storage

Once the detection and classification system has identified a bird, the hardware system will store a video of that bird. The storage system will encompass cloud backup storage as well as local storage. These stored videos will be able to be streamed to a web client on request. In addition, the system will be able to stream live video from the bird feeder. The video playback and live streaming will be at least 1080p quality or higher.

1.4 User Notifications and Frontend

The final product will deliver a frontend web client and notification system. The frontend client will allow functionality to view previously identified birds as images and short video clips, receive notification upon new identification, provide a live video stream to the feeder, allow filtering for which birds are identified, and show statistics about bird identifications.

2. Non-Functional Requirements

The non-functional requirements revolve around the ease of use of the system for the client. The client should have no issues setting up the camera and moving the system's enclosure. The client should be able to seamlessly use the web interface provided to watch the streamed video, browse the captured photos and videos, and configure the notification system. In addition, it should also be under the project's budget of $1500.

IV. Previous Work / Literature Review

1. Existing Solutions

Currently, there are no commercial solution that accomplish our task. There are a variety of wildlife cameras available on the market but, most existing solutions only take pictures when it detects motion in front of the camera. They do not attempt to detect actual objects in the frame, so they will be prone to false positives. Additionally, they do not attempt to perform any sort of classification for the objects in the picture.

We were able to find a similar hobbyist project by Kirk Kaiser[3]. On his blog, Kaiser describes a bird detection and classification solution based around an Nvidia Jetson TX1 board. Kaiser runs machine learning models for object detection on the TX1, and whenever the models detect a bird, the camera saves 240 frames from the camera to an attached SSD. Where Kaiser's solution lacks is classification - the models Kaiser is using are very broad, and will only apply the label "bird" to any birds in the image. We are looking for a solution to not only detect birds in an image, but also apply a label based on its species. Additionally, we aim to move storage requirements to a more accessible medium such as cloud based storage, rather than an onboard SSD.

2. Relevant Literature

There have been other attempts to use deep neural network architectures to identify animals in images from wildlife cameras. We found one instance where Norouzzadeh et al.[4] used a multistage pipeline to detect and classify 48 species in the snapshot Serengeti dataset. The researchers were able to achieve a very high detection and classification accuracy, above 99% top 5, but it should be noted they used a very deep architecture consisting of 152 layers. We believe this architecture to be far too deep for our embedded application, but can use other ideas from them. We can use their idea of a two stage pipeline, one stage for detection and another for classification. This will enable us to use a more expensive classification model, and just have it run less often.

The most relevant literature to our project is based around the classification solutions. Recently, there have been many breakthroughs in the field of image processing and classification. We are researching a variety of architectures such as SqueezeNet[5] and AlexNet[6].

V. Proposed System / System / Solutions

We propose using a self-contained embedded platform that is equipped with several key features: online connectivity, internal storage for images and videos, high quality optics, and an onboard recognition system. A sample block diagram of the system can be seen below.

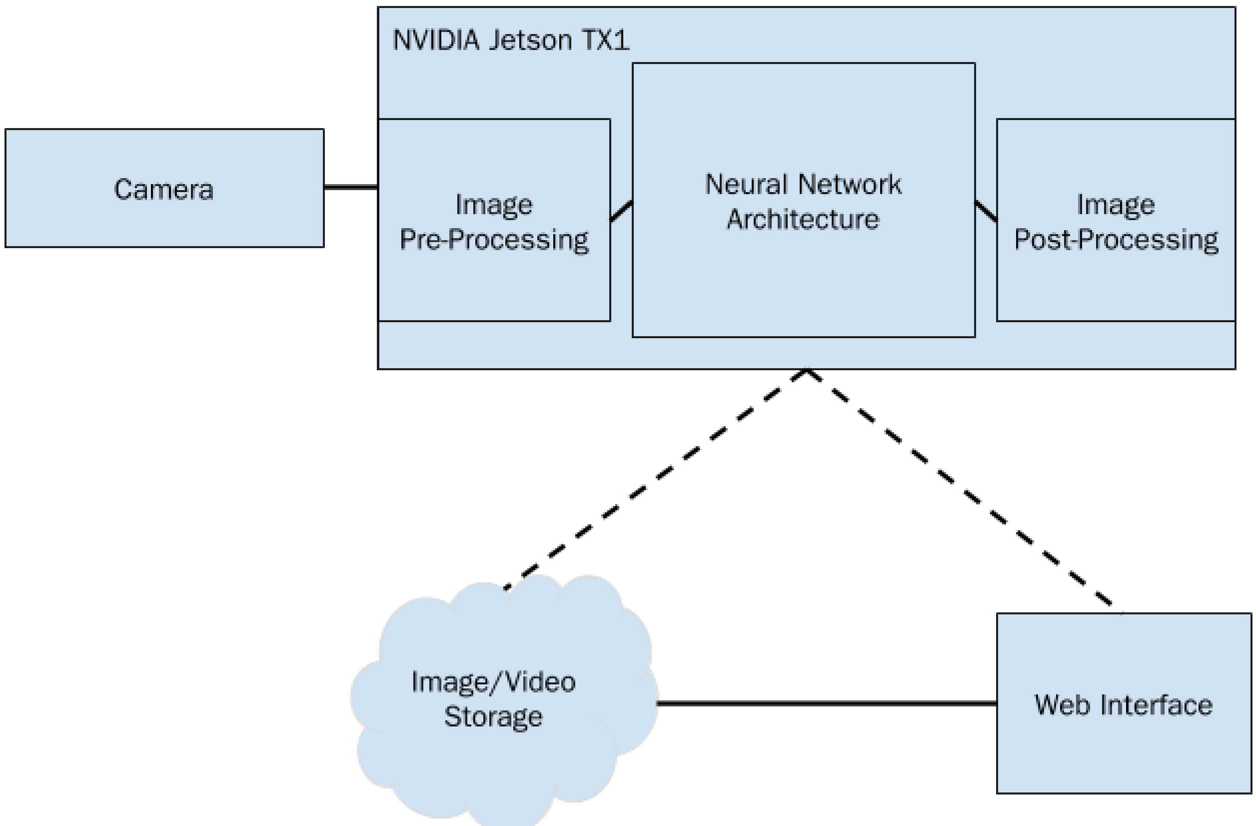

Fig. 2: System Block Diagram

The system begins with a 4K image capture or stream from the e-CAM131_CUTX2 from E-Con Systems. The camera interfaces with the NVIDIA Platform through the Camera Serial Interface. Once the frame is on the board, we will apply some image pre-processing to reduce the image size for more efficient computing. Then, the image will be fed through our neural network architecture. This may be one or more neural networks for detection and classification. After, we will apply a image post-processing algorithm. From the board, these frames will be sent to the cloud for storage over the client's home network.

1. Hardware

The hardware system begins with the NVIDIA Jetson TX1. We will leverage the board's high performance embedded GPU to run our neural network.

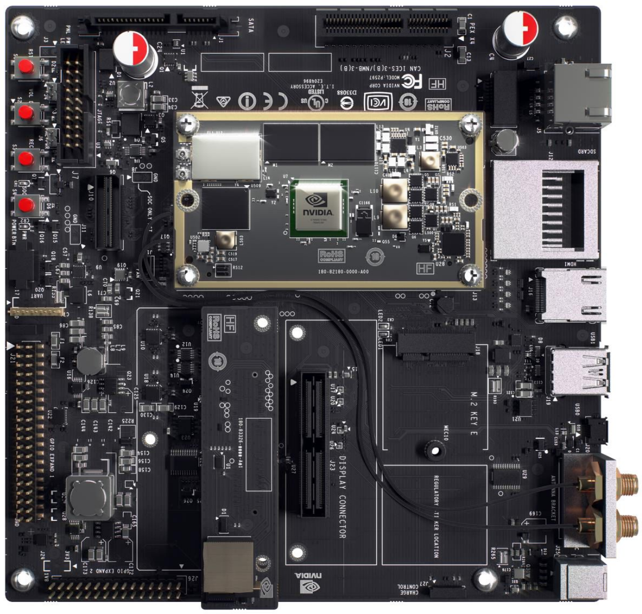

Fig. 3: NVIDIA Jetson TX1 Developer Board[7]

For the camera, E-Con Systems provides a Linux camera driver (V4L2) for the 13.0 MP MIPI CSI-2 camera. This will allow us to interface the board with the camera. On top of that, the board is also compatible with Gstreamer-1.0 for video recording and network streaming. We are also having a solid state drive connected to the board over USB to have easy access to the captured images from local storage.

2. Detection and Classification

We are considering using a cascading-style neural network architecture. The first neural network will be solely for bird detection. The second will be a convolutional neural network for the bird classification. We are splitting these two up because it is much faster to detect a bird than to classify one.

3. Data Streaming and Storage

Out storage solution will be based in the cloud. The Jetson board will connect to an endpoint for a cloud application and stream picture data as well as classification/detection data to the cloud service. The cloud service will backup the data, and make it available to the frontend.

For video streaming, the cloud service will establish a connection between the board and front end. Establishing a direct connection between the board and the user will help reduce overheard from cloud processing, and reduce cloud costs. The user can then choose if they would like to save certain clips from the stream in our cloud storage solution.

4. User Notifications and Frontend

The frontend solution will be implemented using Google Cloud Provider (GCP). The website will allow users to view previously identified birds, a live stream of the bird feeder, and change notification settings. Each previously-identified bird will have an associated video or image, and statistical information about the detection/classification. The user can select which birds to be notified for and modify the threshold at which to notify. The notification system will using the Web Notification API. Clients will use their web client of choice to subscribe to notification from the bird watching application.

VI. Assessment of Proposed Solution

1. Hardware

1.1 Strengths

- E-Con Systems cameras are supported by the Nvidia Jetson and have high frame rates at 4k video resolution.
- The Nvidia Jetson board is widely used in embedded solutions for image recognition, and is a good form factor for our outdoor application.

1.2 Weaknesses

- E-Con Systems cameras are not widely used in the consumer market and not many examples of implementation are around.

2. Detection and Classification

2.1 Strengths

- Convolutional neural networks are widely used today for image recognition, and are highly expandable to different image domains.

2.2 Weaknesses

- Most of the available training datasets consist of high quality images of birds and only birds, whereas the images our system will be using will likely contain more non-bird objects, leading to lower classification accuracy.
- Previous academic work with this dataset is sparse.

3. Data Streaming and Storage

3.1 Strengths

- Cloud storage will be expandable and will not have to work about hardware limitation or hardware reliability for storage and web interface.

3.2 Weaknesses

- 4k images take a lot for storage and could use a significant amount of our client’s internet bandwidth.

3.3 Trade-Offs

- We will be sacrificing some video resolution during video streaming to ensure that the stream will not be choppy.

4. User Notifications and Frontend

4.1 Strengths

- A cloud-based frontend will provide a clean interface for notifications and viewing of birds.
- Notifications can be customized based on users' interests in certain kinds of birds.

4.2 Weaknesses

- The website will be hosted in the cloud and will have recurring costs.

VII. Validation and Acceptance Test

In order to test our project design, we would need to test each individual component:

- The housing will be tested for weather-proofing, without any equipment in it. This way, we can validate the integrity of the structure without damaging any critical hardware. Once we verify that the housing is appropriately resilient, we can insert dummy hardware to check that the structure is not compromised before installing the real components.
- The camera images will have to be inspected by hand, for quality and sharpness. This will be done after encoding and uploading.
- The web interface will be evaluated by Dr. Zambreno and by other members of the team to ensure its design is user-friendly.
    - The user's experience will be measured to ensure that setup and usage can be done in a reasonable amount of time.
- The detection model will be tested against real-world conditions provided via video from the client's backyard. We are aiming for at least a 90% positive accuracy on detection in good conditions, and less than 10% false positive accuracy, both measured frame-by-frame.
- The recognition model will be tested both against sample images from our dataset, as well as against the real-world conditions described above. We are aiming for at least an 90% correct classification under ideal weather conditions.

VIII. Project Timeline

The following two Gantt charts break down our project across the two semesters.

1. First Semester Goals

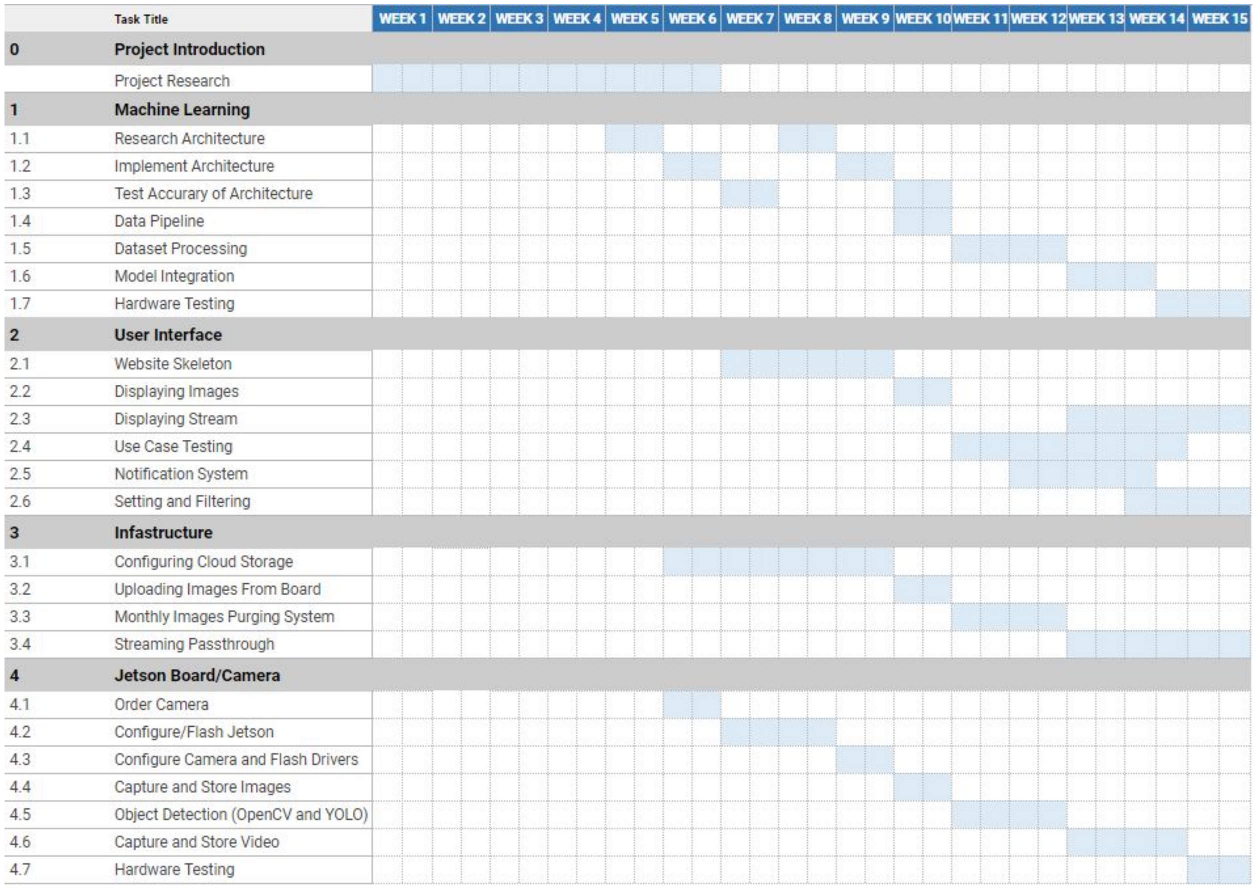

Fig. 4: First Semester Gantt Chart

After the first semester, we will have a preliminary prototype available for the client to take home over winter break. This will entail the Jetson board and the camera interfacing properly with image and video capture and storage. There will also be some basic bird detection with YOLO so that the board knows when to capture an image.

The cloud storage will also be configured so there is somewhere to send the captured images and videos of birds. The connection between the cloud and the web interface will also be configured so that images and video taken can be viewed by the client.

Finally, we will also have to have a clear convolutional neural network architecture established. We hope to have a 90% accuracy on training images and 60% accuracy on real data from the board. We do not plan on having the neural network running on the board for winter break, but these metrics will be tested later on images captured from the board over break.

2. Second Semester Goals

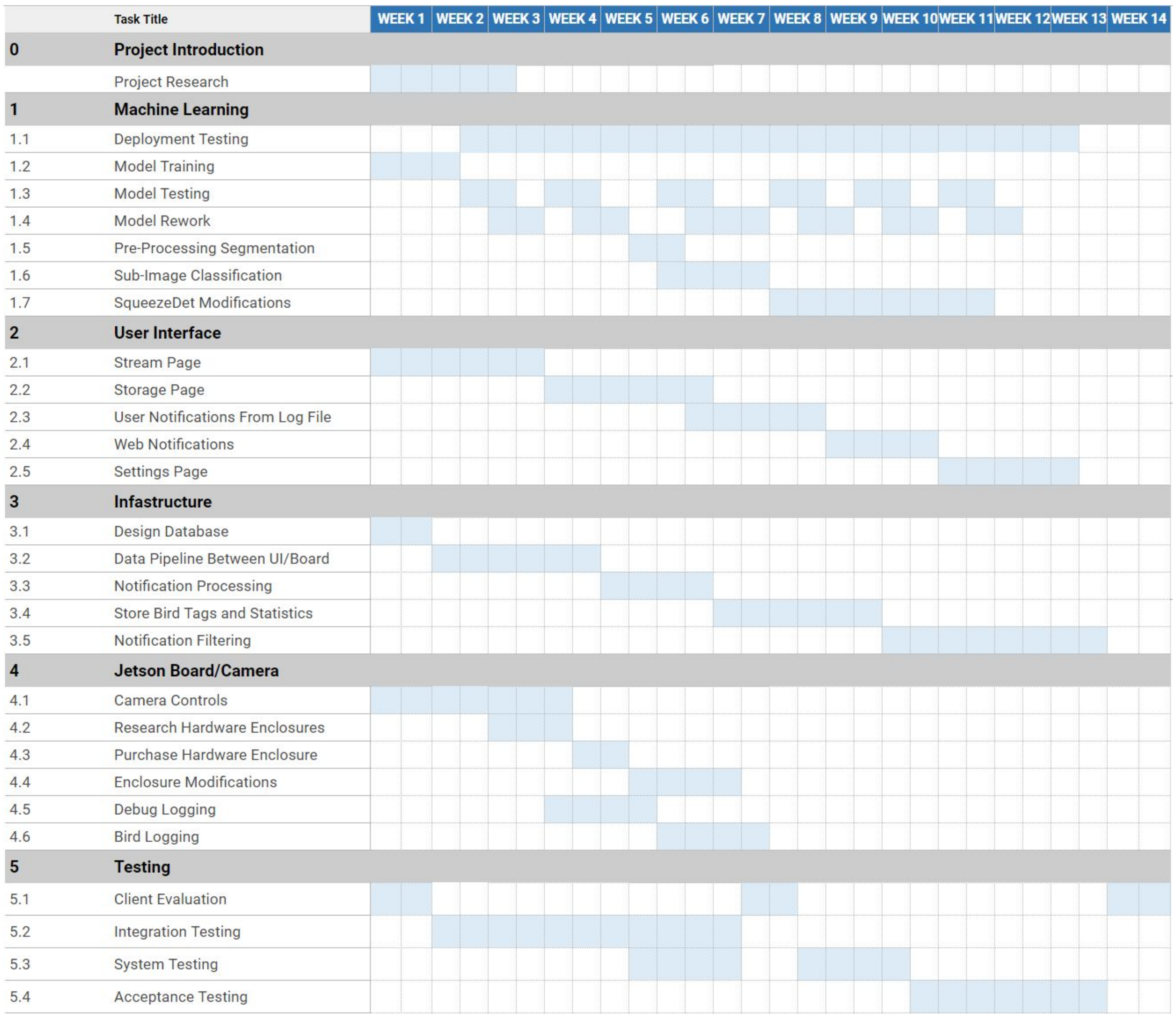

Fig. 5: Second Semester Gantt Chart

Our progress next semester will revolve around more system-level features like notifications to the user and more communication from the camera and board to the web interface to allow the user to control the brightness of the camera. In addition, the machine learning model will be more fleshed out and put on the board. Once that happens, we will do all of our testing and operation against the model. The web interface will also be more fully developed with several key pages with clear functionality with respect to video streaming, image storage, and notification settings. To bring the whole product together, a hardware enclosure will also be ordered and modified to meet our needs of a clear camera view.

IX. Challenges: Risks / Feasibility

1. Feasibility and Challenges

There are various challenges we will need to overcome throughout the course of this project. To successfully complete our project we will need to be able to detect and classify birds in real time, store images and video clips, and finally notify and make the data available to our end user. We believe that there are tools and procedures that can be used to mitigate these challenges.

First and foremost, we must find a way to efficiently classify birds. Currently, machine learning models are often run in resource rich environments. For our application, we will need our models to run in a very limited embedded environment. We believe that we can deal with this issue through a variety of machine learning techniques such as pruning our output classes, and reducing the overall size of our model. These changes may then impact the accuracy of our classification, and we will need to find a balance between accuracy and speed that meets our requirements.

Second, there exists a risk that optimizations in one subsystem will result in a loss in functionality or performance in another. We will have to take great care to ensure that we consider the health of the entire system in every change we make. To accomplish this we will develop various integration tests to test the health of the overall system. These tests should function as a means of ensuring that changes are localized, and do not have unforeseen consequences in other subsystems.

The next challenge after identification will be transmitting the images from the board over to the user. Given the high quality optics that are being used, the board will need to dedicate a significant amount of time to video encoding and data storage. This is require a balance of cpu resources dedicated to video streaming/storage and bird detection/classification. This problem is a resource optimization problems which can be realistically solved if we dedicate enough time to testing our streaming and classification systems.

2. Risks

There are two primary risks and security concerns we must consider when designing our system. They include an unauthorized individual viewing the video stream from the base station, and an attacker gaining control over the base station. The probability of either of these events is very low, but they must still be considered when designing our system. We will have to make sure that we use appropriate authentication measures and limit access to the TX1 board outside of our use case.

X. Standards

We will have to take several standards into consideration when exploring our options on case design or purchase. As an outdoor product, this project must be able to withstand some amount of dust and moisture exposure.

The IP code, for ingress protection, is an international standard that measures “Degrees of protection against the penetration of solid bodies, the penetration of water, and the access of personnel to live parts.[8]” IP codes are formatted as IP##, where the first numeral measures its protection against dust and the second numeral measures protection against moisture. It would be appropriate for the enclosure to have an IP code of IP43. This corresponds to “Protected against the penetration of solid objects having a diameter greater than or equal to 1 mm” and “Protected against rain at an angle of up to 60 degrees.[8]”

As part of this project, we will also have to adhere to different professional standards like the IEEE Code of Ethics. We will “commit ourselves to the highest ethical and professional conduct[9]” in accordance with the IEEE Code of Ethics. One principle from it in particular that is of extreme importance to our project is “to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.[9]” Throughout this project, we will have to grow our technical skills and be realistic with our ourselves, client, and advisor about what we can get done.

XI. Test Plan

To ensure our project meets our client's needs in all technical regards, we will be implementing and evaluating various types of tests. These tests will cover functional- and non-functional use-cases, as well as ensuring that the interfaces between project components meet specifications throughout the iteration process. To carry out some of these one-time and on-going tests, we will need specialized tools for both hardware and software testing.

1. Interface Specifications

The first interface is the interface between the camera and the Jetson board. We will not have to explicitly test this interface since all of the software to control the camera is provided by the camera manufacturer. That being said will still have to keep this interface in mind as a possible source of error in out other tests.

Second, we will need to test the interfacing between the Jetson board and our cloud services. This pipeline will be our only method to pull detection and classification data off of the board, so it is of the utmost importance that this interface be reliable. We will be transmitting pictures of birds along with classification data over this link. Additionally, we plan on implementing camera controls in the user interface, so there will be a backwards link to send those commands.

Our next interface consists of the various software interfaces in our cloud backend. Here we will need to make sure that pictures sent by the board are properly delivered to the various backend services. From there, data will be sent to the storage layer, where we will store pictures and metadata in our database. At the same time, our notification system should index the data and send out a notification of the event to our client. All of these services are very time-critical and we will have to test to ensure they both run efficiently, and reliably.

Penultimately, we have an interface for the board software to interact with the neural network architecture, which primarily consists of calculating detection & classification for images. This interface will need to be monitored to ensure that the neural net is operating correctly, and can be used to extract the classification data mentioned above.

Finally, we will have to test the interface between our backend services and our user interface. There will be various data streams from the backend to the user frontend. First, there will be a notification system that must quickly and efficiently receive a bird sighting event, and pass on this event to our user front end. Next, we must be able to pass along the images and metadata we have stored along to our webpage to allow the user to browse this data. Finally, we must pass along data from the live video stream coming off of the jetson board along to our user interface.

2. Hardware and Software Testing Tools

2.1 Hardware

We plan on ensuring our box adheres to the IP43 standard. These tests require the use of a table and a water jet capable of displacing 10 liters of water per minute.

2.2 Software

We plan on writing our own code to test the software components of our system. The majority of this will be accomplished using unit testing tools for the appropriate languages: PyUnit for Python, Catch for C++, and Jasmine for JavaScript. Most of these testing frameworks also provide integration testing features. Finally, we will use Selenium Webdriver for testing our user interfaces.

PyUnit (also known as unittest) is the built-in unit testing library in Python. It provides a user-friendly set of unit testing functions that will help automate the testing process.

Catch is a unit testing framework C++. Similar to PyUnit, it provides unit testing functions that will help automate the testing process. The main advantage of Catch over other C++ testing frameworks such as GoogleTest or the BOOST test library is that it is easy to set up and requires very few dependencies.

Jasmine is a popular unit testing framework for JavaScript frontends. It is often paired with Angular, a common JavaScript framework, but will work with any of the popular frameworks as well. Tests are very simple to write, and will help us automate the unit testing process. Additionally, Jasmine provides very extensive Integration testing features.

Selenium WebDriver is a program for testing web pages. It is a means of automating user interactions from the perspective of a user. It will boot up a web browser, go to the specified urls, and attempt to complete the defined actions.

3. Functional Testing

3.1 Unit Testing

Unit tests will be written on a file by file basis. We will write simple functionality tests for the functions in our files. Additionally, we will write more thorough interface tests for the overall components of our project.

3.2. Machine Model Testing

When we test the ML model, our testing procedure is fairly standard:

1. Give the model an image as input, with the appropriate truth label attached
2. Compare the model's predictor confidences to the ground-truth label in (1)
3. Add this result to a table of results
4. Repeat steps 1-3 for each image in the test dataset.

For our current dataset (CalTech-UCSD Birds 200), we have 5794 test images. We can evaluate the model's performance via the formula:

$$\frac{\#\ correct\ predictions}{\#\ total\ images}$$

3.3 Hardware Enclosure Testing

We will test that the box adheres to the IP43 standard. The tests for this certification are two fold. First we must ensure there are no holes in the enclosure larger than 1mm. This should limit access into the box and prevent wildlife from tampering with the system. Second, we must test for waterproofing. These tests consist of placing the box on a table and spraying the enclosure with a spray nozzle for period of 5 minutes at various angles. After spraying the box, we will ensure the inside of the enclosure is dry.

3.4 Interface Unit Testing

Using Selenium WebDriver we will test the functionality of the web interface. Each web page will be tested to ensure that it acts according to design. In the stream webpage, we will test that the page opens with a streaming element on located in the center of the page. For the bird information page, we will ensure that the cloud links loads and can it displays at least one image. We will also test the notification system. We will also test notification testing. The unit tests will post a notification to the users. The testing framework will then validate that it has received the notification. Lastly, the navigation bar will be tested on each page. We will ensure that each link loads to ensure that no links are broken.

3.5 Integration Testing

Our integration testing consists of testing the communication links and interactions between adjacent components in our system. This set of links includes:

1. Jetson Board to Cloud backend
   a. This link is focused on sending captured images along with the output of the classifier. Additionally this link will carry the video stream from the camera in order to pass it on to the user interface.
2. Cloud to Client (Website)
   a. This link is focused on serving our webpage to the end user as well as send the large image files from our storage layer.
3. ML Model to Board
   a. We plan on working our model into our video processing pipeline. As such it will be constantly receiving video frames, and, on detection, output the classification data as well as the single frame for transmission to the storage layer.

To test these links we plan on writing unit tests in each component to test its relevant link. For example, the Jetson board will test its connection to the cloud by attempting to upload an image to the backend, then query the backend for images. If the test image is not in the query result, then the test will fail. This style of testing will be implemented on each component of our system.

3.6 System Testing

Our systems testing facilitates checking the system's performance from end to end. In our tests, we are looking for the following properties:

- Birds detected and classified in real time (faster than 3fps)
- User is notified of the birds
- Web client displays detected birds' stats accurately
- Hardware system functions unimpeded in Iowa's weather

To test these properties, we will run a set of supervised tests both inside a controlled test environment as well as at the client's deployment site. During the tests, we will be monitoring the hardware to ensure that each of the above properties are satisfied. Additionally, we will be recording the video produced from the hardware's camera to use in re-creating any scenarios where the system fails.

To test in the testing environment, we will model the real world by printing picture of birds. These pictures will be moved into the camera's field of view to simulate a bird flying into view. This will test the system's ability to detect and classify accurately. Separately, testing at the client's site will be a real world test. During which the tester will monitor the hardware and manually record any birds in the field of view. If a bird is in the field of view, the tester will record the time and check to see if the hardware detected and classified the bird. This real world test will verify the hardware system's resilience to Iowa's weather. A successful test in either environment will show the system processing at least 3 frames per second, the camera detecting the bird, notifications sent to the user, and the web interface displaying the detected birds.

3.7 Acceptance Testing

Acceptance testing will be performed in two stages. Stage one will be usage testing and demonstration of the system in a test environment. Stage two will be real-world testing at the client's site.

During stage one, the client will be invited to a testing lab. Here, we will perform a systems test on printed images of birds. The client will be able to see the performance of the entire system. We will provide a list of client's system requirements. These are the same requirements listed in the project plan. The client will be asked provide feedback on how well the system fulfilled each requirement. This feedback will be incorporated back into the design before the next stage.

Stage two is a test at the client's site. Here the system will be tested under real world conditions. The box will be exposed to outside weather and continuous use for 24 hours. After that testing, period the client will be asked to respond how well the system fulfilled each of the requirements.

4. Non-Functional Testing

Non-functional testing will be looking at empirical usability of the system, rather than measurable functionality. The non-functional requirements we will be:

- The client should have no issues setting up the camera and moving the system's enclosure.
- The client should be able to seamlessly use the web interface provided to watch the streamed video, browse the captured photos and videos, and configure the notification system.
- The total cost of the project should be under the project's budget of $1500.

To test ease of setup, the system will be installed and setup with the client. Afterwards, the client will be asked about the difficulty of the installation and if they believe they could install it themselves. If the answer is no, the client will be asked for feedback as to why and if they have any suggestion to improve the installation experience.

Testing of the web interface ease of use will be in sequence with acceptance testing. Directly after acceptance testing, the client will be asked if they believe the system is easy to use. If the answer is no, the client will be asked to provide a reason why, and improvements will be made based on the feedback.

Cost will be determined as the semester continues. The cost of each part is logged with the Electronics and Technology Group (ETG). We keep record of the cost of each part to ensure the cost is less than $1500.

5. Process

Our testing process will follow a fairly straightforward approach. Unit testing will be run throughout the development life cycle. They will be continuously run when code is in development and we will run our tests before commiting code to the repo. This will prevent us from pushing broken code to the repo and will ensure that any future bugs arise from the code we are writing rather than from previous code.

Next, we will run Integration tests when we touch code that directly interacts with any of the interfaces we have defined above. The purpose of the interfaces is to function as a contract between two services, and whenever we touch components which interact with these contracts, we need to test the integration between the two components to ensure. This way we assure that all promises made by the contract are upheld.

Our final set of testing is system and acceptance testing. This testing will consist of full end-to-end system testing. We will manually start the process by feeding the system a picture of a bird and test the user stories we have developed. This includes testing video streaming, user notifications, and image access on the user interface.

6. Simulation and Modeling

A major component of our project is the ability to detect and classify birds. To this end, as mentioned above, we are using a CNN to determine appropriate labels from images. As part of creating a functional CNN, we will have to determine filter weights for each convolutional layer. To do so, we use loss-minimization techniques, where the loss is calculated based on processing an image whose label is predetermined, and comparing the network's predicted label to the known "ground truth" label. This process, commonly known as "training," is a form of simulation, as the input images are collected, labeled, and organized beforehand, and running them through the network is done in a controlled laboratory environment, off of the Jetson board. Once the appropriate weights have been determined, they can be loaded and run on the Jetson.

7. Results and Challenges

We have successfully connected the camera to the Jetson board, and used it to capture images and video. This required setting up the camera drivers on the Jetson. We have configured a database in Google Cloud Platform. We are able to upload images from the board to the cloud backend, and then serve them up on a mock user interface.
We are still continuing to work on our classification model. We began with a toy architecture based on two convolutional layers, two maxpool layers and an alternating configuration. This resulted in ~10% accuracy in testing (note that a random selection gives 0.5% accuracy). This is will act as a baseline for our future architectures.

XII. Conclusions

In summary, our implementation thus far has included development across the four main areas of our project: hardware, detection and classification, data streaming and storage, and user notifications and frontend. The Jetson board and E-Con Systems camera are configured properly with the JetPack SDK and drivers. Basic image and video capture are available and a third-party object-detection algorithm, YOLO, is running on the Jetson. Saved image files are successfully uploaded to our Google Cloud infrastructure. The web interface shows the images from the cloud platform. A machine learning architecture has been established and the model is in the process of training. The combination of all this progress makes a basic functional prototype of the design for our client.

Our goals for next semester primarily revolve around “end-to-end” functionality: taking high-quality imagery of birds; being able to detect & classify birds accurately; efficiently transmitting, cataloging, and storing the imagery and appropriate labels; and easily accessing that stored information via a functional web-based user interface. There will also be a significant amount of testing conducted next semester to ensure the quality of our product. The combination of all these efforts will lead to a well-documented HW/SW product to our client that meets their needs to successfully identify North American birds in real time.

XIII. Figures and Tables

The following figures were referenced and included in this document.

Fig. 1: Four North American Birds (Page 4)
Fig. 2: System Block Diagram (Page 10)
Fig. 3: NVIDIA Jetson TX1 Developer Board (Page 11)
Fig. 4: First Semester Gantt Chart (Page 16)
Fig. 5: Second Semester Gantt Chart (Page 17)

XIV. References

[1] “Northern Cardinal Identification, All About Birds, Cornell Lab of Ornithology,” *, All About Birds, Cornell Lab of Ornithology*. [Online]. Available: https://www.allaboutbirds.org/guide/Northern_Cardinal/id. [Accessed: 29-Sep-2018].

[2] J. Redmon, S. Divvala, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object
Detection.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. [v5]

[3] K. Kaiser, “Designing A Deep Learning Bird Camera Platform,” *Make Art with Python*, 14-Feb-2018. [Online]. Available: https://www.makeartwithpython.com/blog/deep-learning-bird-camera/. [Accessed: 01-Dec-2018].

[4] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” Proceedings of the National Academy of Sciences, vol. 115, no. 25, Jun. 2018.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] Iandola, Forrest N, et al. “SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size.” *ArXiv*, CoRR, 4 Nov. 2016, arxiv.org/abs/1602.07360. Accessed 27 Oct. 2018.

[7] M. Walton, “Nvidia's Jetson TX1 dev board is a ‘mobile supercomputer’ for machine learning,” *Ars Technica*, 11-Nov-2015. [Online]. Available:

[8]“Degrees of Protection Provided By Enclosures,” *Schneider Electric*. [Online]. Available: https://www.schneider-electric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/176000/FA176928/en_US/Degrees of protection IP,IK, NEMA.pdf. [Accessed: 27-Sep-2018].

[9] IEEE Code of Ethics. (2018). *IEEE Potentials*. https://arstechnica.com/gadgets/2015/11/nvidias-jetson-tx1-dev-board-is-a-mobile-supercomputer-for-machine-learning/. [Accessed: 01-Dec-2018].