# Rapid Prediction of Bacterial Growth Inhibition using Google's Coral AI Platform

## Design Document

Team 18
Client: Meng Lu
Team Members:
Declan Costello
Shijin Huang
Wenjie Jin
Owen Schacherer
Ani Tangellapalli
Iris Top

sdmay21-18@iastate.edu
**http://sdmay21-18.sd.ece.iastate.edu**

Revised: 10/25/2020 V2

# Executive Summary

## Development Standards & Practices Used

- We will be using Google's recommended workflow & standards to develop the TensorFlow ML model

## Summary of Requirements

- Portable, Hand-held device to collect bacteria video data
    - Must be usable in a lab setting
    - Must be able to collect video of E. coli samples
- Machine Learning model to detect and classify wild & anti-microbial E. Coli bacteria

## Applicable Courses from Iowa State University Curriculum

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

- COMS 227/228
- COMS 474
- CPRE 288
- CPRE 482x

## New Skills/Knowledge acquired that was not taught in courses

- Understanding of Google Coral AI board
- Developing a new ML model
- CAD modeling using SolidWorks

# Table of Contents

## List of figures/tables/symbols/definitions

# 1 Introduction

## 1.1 Acknowledgment

We would like to thank our project advisor Dr. Meng Lu, Shirin Parvin, Rachel Shannon, and our team members.

## 1.2 Problem and Project Statement

Every year waterborne diseases cause a substantial economic burden, costing more than $2 billion in treatments in the US alone. Roughly 90 million patients fall ill per year to conditions such as Escherichia coli (E. coli). E. coli, one of the most common public health concerns, is spread through drinking water, contaminated food consumption, and contact with infected animals or people. Recently, certain strains have become immune to Penicillin, a common antibiotic. Therefore, it must be detected early to avoid any infections by the super disease.

Several E. coli detection methods exist, such as culturing samples on solid agar plates or in liquid media. The use of liquid growth media provides high sensitivity; however, it requires at least 18 hours for the final read-out. Solid agar plates are more cost-effective and more flexible but often take 24 to 48 hours to grow. It is also possible to use molecular detection methods to reduce the assay time to a few hours; however, the results lack the sensitivity of the tests mentioned previously. There is a strong need for an automated method that can achieve rapid colony detection with high sensitivity to accelerate the identification of dangerous diseases in a laboratory setting.

To provide a powerful alternative that can rapidly detect and classify resistant vs. non-resistant E. coli, we propose a system that will collect live growth data of E. coli with which it will use to classify the bacteria into the two required categories. The system will be composed of a physical device to collect the visual data and a software component to detect and classify the bacteria. The device will be capable of accumulating a video feed of E. coli samples. The video will be of sufficient length and quality to obtain the most accurate predictions. Due to restrictions in the lab, the device is small and portable. The software component is composed of a runner program and an ML model. The results from our system will accelerate the detection of resistant E. coli by many hours, which can help avoid many infections.

## 1.3 Operational Environment

During the fall semester, our ML experiments will be conducted using TensorFlow. However, factors such as environment and weather cannot be ignored. Therefore, in the final test, we will consider the growth rate of bacteria in different environments and whether the bacteria survive. For example, whether it is surface water or groundwater, rainwater, or snow water, there will be bacteria. According to the oxygen demand for bacteria, it can be divided into three categories: anaerobic bacteria, facultative anaerobes, and aerobic bacteria. Salmonella Enterica is one of the most common bacteria in water. Under normal circumstances, it can survive for 2-3 weeks, and it can survive for 3-4 months in the refrigerator. Its optimal breeding temperature is 37°C, and it can reproduce in large numbers above 20°C.

In the final test, we can study the growth rate and survival of bacteria at low temperatures. In addition, we can also compare the growth rate of different types of bacteria in different environments to determine which bacteria are the most threatening.

## 1.4 Requirements

Functional requirements

- The machine learning model must be able to detect resistive bacteria with an accuracy of 90%.
- The machine learning model must be able to analyze at least 10 minutes of video
- The mobile component should allow users to take and store video feed
- The whole system must be portable and be held and usable in the user's hands

Economic requirements

- The solution should be developed under a $500 budget

Environment requirements

- Keeping team members safe when working in the lab is our first priority
- Lab substances must be used and disposed of correctly
- Everyone should wear proper protective equipment and follow rules and instructions in the lab
- Everyone must take care of and be responsible for our lab equipment

## 1.5 Intended Users and Uses

Anyone whose job is related to dealing with the habitat of E. Coli could be the potential user of this project. Users could be from farmers to workers of the food industry and workers of the water purification market.

## 1.6 Assumptions and Limitations

Assumptions:

- We detect and classify all bacterias and their rate of growth
- This research only used around Ames to help farmers.
- If the data we tested are not accurate or something goes wrong, we will retest them all until correct.

Limitations:

| Limitation | Expression |
|---|---|
| Budget | No more than $500 |
| Detection/Classification | All test will be on network and lab |
| Schedule | All tese will be end beofre 11/18/2020 |
| Function | Mechine Learning model |

Figure 1.6.1 - Limitations of the project

## 1.7 Expected End Product and Deliverables

Portable device (May 2021)

- The portable device will be created using off-the-shelf components. It will contain the Google Coral AI board, which is responsible for running the ML model, and a means of collecting videos of E. coli samples. The designed device will be handheld and portable to be used in a laboratory setting. The device will be running on a battery capable of powering both the Coral AI board and the video collection unit.

Machine Learning Model (January 2021)

- The ML model will be created using TensorFlow 2.0. The model will be capable of running on the Google CoralAI platform and it is responsible for locating and identifying the bacteria in the provided data. Specifically, the model will be capable of differentiating between wild and antimicrobial-resistant E. Coli bacteria with a high accuracy. The input data format must be an image or a single frame from a video.

# 2 Project Plan

## 2.1 Task Decomposition

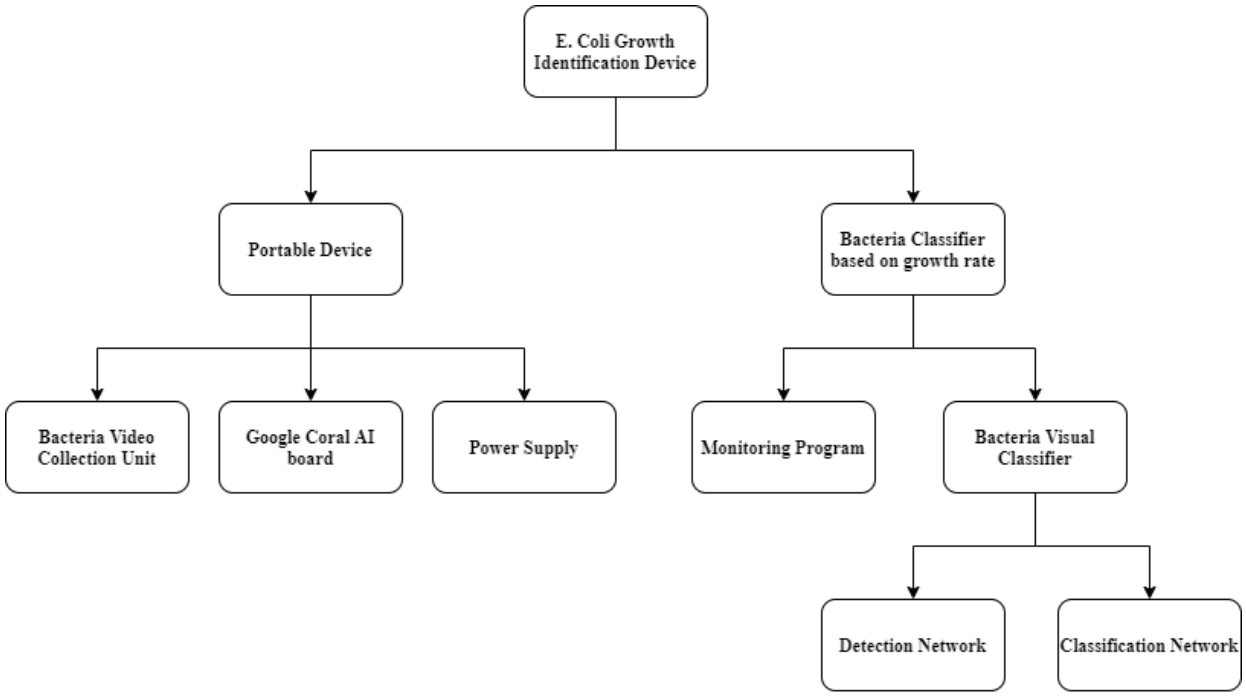


Figure 2.1.1 - Graphical representation of the tasks and their dependencies

- Portable Device
  - Houses the Google Coral AI board, the video collection unit and the power supply
- Bacteria Video Collection Unit
  - Capable of collecting video of E. coli samples
- Google Coral AI Board
  - Similar to a Raspberry PI; Responsible for running the ML model
- Power Supply
  - Responsible for powering the components on the portable device
- Bacteria Classifier
  - The full application including ML model and the runner program
- Monitoring Program
  - The runner program for the ML model and displaying results
- Bacteria Visual Classifier
  - The ML model capable of identifying resistant vs. non-resistant bacteria
- Detection Network
  - ½ of the ML model which locates any object in the given image
- Classification Network
  - ½ of the ML model which identifies the objects from the detection network into bacteria, dirt, etc.

### 2.2 Risks And Risk Management/Mitigation

Risks for our project include scope, hardware, and COVID. When gathering training data, uncontrolled changes and continuous growth of the scope of our project can occur. As we collect training data, we can sample out valid data at the cost of time. Another risk for our project is the hardware and software malfunctioning. Malfunctions can be mitigated by investing in better equipment as well as trying other variations of equipment. Another risk is COVID in general. COVID can make it hard to keep up with the current restrictions put on campus to go and physically collect our sample data. This can be mitigated by overcommunicating with our supervisors when talking about the current precautions. COVID can also harm our group's availability to meet. This can be mitigated by using better software to meet and communicate.

### 2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Some key milestones in our proposed project include mastering TensorFlow, choosing a machine learning algorithm, and choosing the correct metrics to measure our project. More milestones include collecting up to about 12,000 valid training data sets and revisiting and optimizing past tasks. These soft goals will help us to reach our hard goals of raising our machine learning models to 80% accuracy. This agile project will grow with iterations as we go back and optimize different past tasks and collect more sample data.

### 2.4 Project Timeline/Schedule

A Gantt chart has been created in google sheets for the team to use as a project timeline tracker. An overview of this gantt chart can be seen in figure X-1 and the whole gantt chart can be seen here. Our goal this semester is to design the physical components, create the needed embedded software to run the physical component, and train the bacteria detection models in TensorFlow. The goal next semester is to build the prototype and combine all the components into one and test.

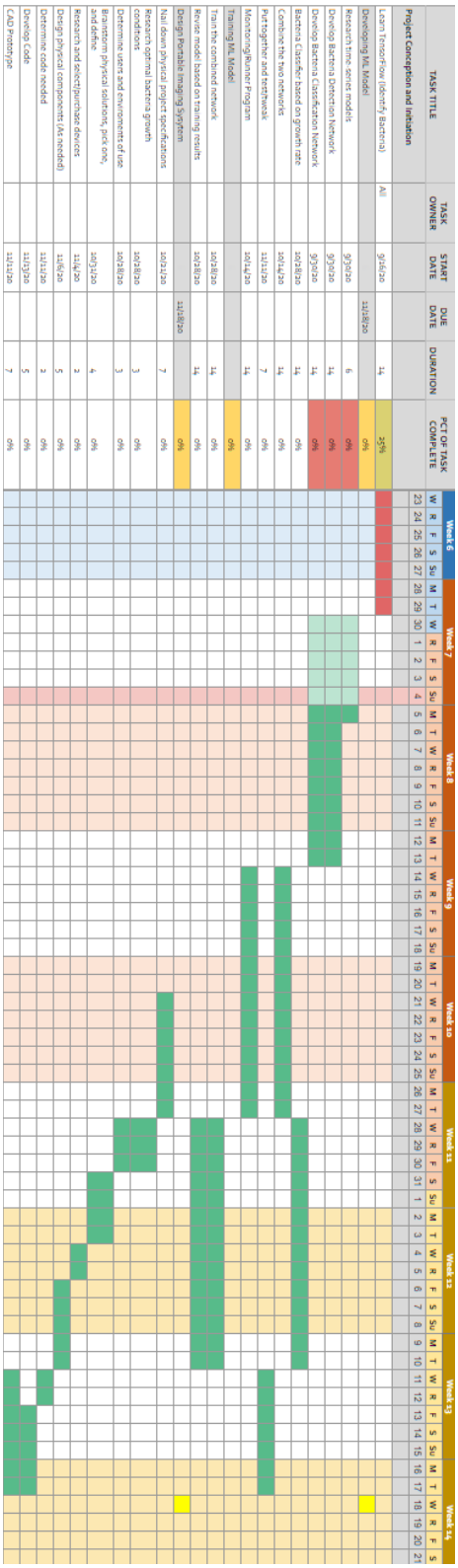


Figure 2.4.1 - Overview of the project timeline

This semester can be split into two major components both of which we plan to complete by November 18th. The first of which is a machine learning model that can accurately predict if bacteria is resistant (identification and classification) based on its initial growth. This can be broken down into 6 different sub components/tasks: bacteria detection DNN, bacteria classification DNN, bacteria classification based on growth rate, combining the networks, monitoring the DNN, training, and revisions. The timing of each of these tasks can be seen below in figure X.

| TASK TITLE | START DATE | DURATION | PCT OF TASK COMPLETE |
|---|---|---|---|
| Developing ML Model | | | 0% |
| Research time-series models | 9/30/20 | 6 | 0% |
| Develop Bacteria Detection Network | 9/30/20 | 14 | 0% |
| Develop Bacteria Classification Network | 9/30/20 | 14 | 0% |
| Bacteria Classifier based on growth rate | 10/28/20 | 14 | 0% |
| Combine the two networks | 10/14/20 | 14 | 0% |
| Put together and test/tweak | 11/11/20 | 7 | 0% |
| Monitoring/Runner Program | 10/14/20 | 14 | 0% |
| Training ML Model | | | 0% |
| Train the combined network | 10/28/20 | 14 | 0% |
| Revise model based on training results | 10/28/20 | 14 | 0% |

Figure 2.4.2 - Initial Timeline for training of the machine learning model

The second major component we will be working on this semester is the physical device and the code that will run this device. We will be following the design process to create this component to ensure we create the most viable product. It can be broken down into nailing down the project specifications, researching bacteria growth (safety, optimal conditions, sizing, etc), determining the users and the environments, brainstorming (and direction selection), selecting the store bought components (researching), design the physical components, determining the code required and writing it, and creating the final CAD prototype. The timing for each step can be seen in figure X+1 below.

| TASK TITLE | START DATE | DURATION | PCT OF TASK COMPLETE |
|---|---|---|---|
| Design Portable Imaging Sysytem | | | 0% |
| Nail down physical project specifications | 10/21/20 | 7 | 0% |
| Research optimal bacteria growth conditions | 10/28/20 | 3 | 0% |
| Determine users and enviroments of use | 10/28/20 | 3 | 0% |
| Brainstorm physical solutions, pick one, and define | 10/31/20 | 4 | 0% |
| Research and select/purchase devices | 11/4/20 | 2 | 0% |
| Design physical components (As needed) | 11/6/20 | 5 | 0% |
| Determine code needed | 11/11/20 | 2 | 0% |
| Develop Code | 11/13/20 | 5 | 0% |
| CAD Prototype | 11/11/20 | 7 | 0% |

Figure 2.4.3 - Initial timeline for the design of the physical prototype

The goal of the second semester is to take the components we have already created, combine them together, and test. This will be accomplished by uploading our trained model on to the Google Coral board and have it analyze the real time video from the physical system.

## 2.5 Project Tracking Procedures

We will be using a variety of softwares to track our progress and communicate on this project. We will be using Git & Gitlab as our version control tool, Microsoft Teams to communicate, a shared Google drive to store documents, and a Google Sheets document as a Gantt chart to keep track of our progress.

## 2.6 Personnel Effort Requirements

The textual reference for this work table will be the gantt chart detailed in the above section (2.4).

A day's worth of  projected effort will be estimated 30/6/5 hours (1 hour) per person.

| Task | Owner | No. Days | Projected Person-hours |
|---|---|---|---|
| Learn TensorFlow (Identify Bacteria) | All | 14 | 84 |
| Developing ML Model | | | 83 |
| Research time-series models | Ani | 6 | 6 |
| Develop Bacteria Detection Network | | 14 | 14 |

| | | | |
|---|---|---|---|
| Develop Bacteria Classification Network | | 14 | 14 |
| Bacteria Classifier based on growth rate | | 14 | 14 |
| Combine the two networks | | 14 | 14 |
| Put together and test/tweak | | 7 | 7 |
| Monitoring/Runner Program | | 14 | 14 |
| Training ML Model | | | 28 |
| Train the combined network | | 14 | 14 |
| Revise model based on training results | | 14 | 14 |
| Design Portable Imaging System | | | 38 |
| Nail down physical project specifications | | 7 | 7 |
| Research optimal bacteria growth conditions | | 3 | 3 |
| Determine users and environments of use | | 3 | 3 |
| Brainstorm physical solutions, pick one, and define | | 4 | 4 |
| Research and select/purchase devices | | 2 | 2 |
| Design physical components (As needed) | | 5 | 5 |
| Determine code needed | | 2 | 2 |
| Develop Code | | 5 | 5 |
| CAD Prototype | | 7 | 7 |

Figure 2.6.1 - Breakdown of each task and approximate effort required

Total number of projected person-effort hours: 335

## 2.7 Other Resource Requirements

Resources we will be using throughout the semester to complete our project are listed below:

- Lab via Client
- Microsoft Teams
- Google Coral A.I. Hardware
- Google Drive
- GitLab
- TensorFlow
- Python
- Parts for physical system from vendors
- Solid Works
- Powerful computer for DNN training

## 2.8 Financial Requirements

We will be allotted a total of $500 for this project. The only financial expenses will be from the creation of the portable system's prototype.

# 3 Design

## 3.1 Previous Work And Literature

This project is based on improving an existing device described in a research paper published in Light: Science & Applications.

Wang et al. *Light: Science & Applications* (2020) 9:118

Advantages

- Neural Network works with respect to time.
- Does not count dust, bubble, speckles that could be mistaken for bacteria

Shortcomings

- Not portable
- Does not look at individual bacteria, looks at a colony
- Very slow

More research has been done developing Neural Networks (NN) that compute using time dimensions. This article pointed us to reference code:

https://neurohive.io/en/news/google-s-new-context-r-cnn-uses-past-images-to-improve-detection/

This is reference code with a NN model that uses time as a dimension:

https://paperswithcode.com/paper/long-term-temporal-context-for-per-camera#code

Advantages

- This code creates a model that uses time similarly to the way we want to.
- Will be helpful when determining the hidden layers to our A.I.

Shortcomings

- This code does not use much else that is similar to ours. It is for an entirely different application.

## 3.2 Design Thinking

**Define**

- Device must be portable
- Be able to record/take pictures of bacteria
- Predict classification of bacteria species
- Detect bacteria
- Displaying results
- $500 Design Budget

**Ideate Phase**

The ideate phase brought up other design choices. Whether we want google coral camera, microscope, powerful microscope.

## 3.3 Proposed Design

Our proposed design is composed of two major components: the incubator and a machine learning model. We started the semester with the design of our machine learning model. Before working on the complex time-dependent model we will be using in our design we worked on some smaller projects to become more familiar with TensorFlow.

We have trained machine learning models to do some simple recognition. For instance, we tested several AI models to identify hand-written numbers, cats, dogs and other small objects. While working on the smaller projects we are also working to set up equipment for taking videos of E. coli which will be used for training our time-dependent model.

The machine learning model we must use will be different from other common image classification models because we require it to monitor the growth rate of the bacteria. A common model can not achieve the same result because it will analyze each image individually instead of as a set. Therefore, our model will have an additional input dimension for time for an added total of 4 input dimensions: width, height, RGB values, and time. To account for the added dimension, we will be using custom conv3d layers as opposed to typical conv2d layers. The conv3d layers must be custom made because they are typically used for videos but in our case we will have a series of images with a long interval in between. The model will output a classification label for each object in the input image. This label will determine whether the object is a resistant E. Coli or a normal E. Coli.

The model will be a two-stage network which are the detection and classification stages. During the detection stage, the model will locate potential objects within the image. This stage of the model can be trained separately. The classification stage assigns a label to each object located by the detection stage. This stage can also be trained individually or the two stages can be trained jointly.

We will ensure this part of the design meets our function requirements by adjusting our training methods until we achieve our desired accuracy. We will attempt different ways to improve the accuracy of our machine learning model to detect resistive bacteria. For instance, adding more data. It allows the “data to tell for itself”, instead of relying on assumptions and weak correlations. Also, we may try to deal with missing and outlier values since those values in the training data often reduce the accuracy of the model or lead to a biased model. After we have tested and certified our model meets the requirements we will download our code into raspberry pi and encapsulate it to add it to our incubator.

For non-functional perspective, since all materials including the microcontroller are fairly cheap, we will be able to satisfy the budget constraint.

The incubator we are creating has four modules we are designing: power supply, heating, microscope apparatus, and an interface. To ensure that the incubator is portable we will be using a chargeable battery for power. This allows a user to move the device while it is running or to use it somewhere without outlets. The power supply we will use will power our heating, the UI/interface with the device, the microscope, and the board. The heating element we will use will heat the petri dish to the optimal temperature for bacterial growth (37 C) but be isolated from the other possibly heat sensitive components. The microscope will be powerful enough to see individual cells but will not be able to see the whole dish at the sametime. To solve this issue we will be using motors to move the microscope across the petri dish (similar to a 3d printer) and stitching together the resultant images. We will use the google coral lenses camera to capture the microscope output as it moves around the petri dish. The last component of the incubator is the interface. This will be a webpage accessed through a local network that can be loaded on a phone or computer. This will display the results, the controls, and other necessary information. Next semester we will build and test the incubator component by component making changes as needed.

## 3.4 Technology Considerations

### Machine Learning Technologies:

| **Technology** | **Strength(s)** | **Weakness(es)** |
|---|---|---|
| TensorFlow | • Popular ML framework developed by Google<br>• Lots of pre-existing models<br>• Well-documented<br>• Deploy to many formats | • Large learning curve since most of our group has not used it before<br>• Low level framework therefore it has some complicated code |
| Google Coral A.I. Dev Board | • Optimized for ML deployment<br>• Developed by Google and supports TensorFlow lite | • Very Expensive |

Figure 2.6.1 - Breakdown of each task and approximate effort required

### Hand-held Device:

We do not yet have a fully designed physical device. However, our solution involves using off-the-shelf components. Our current plan is to use a low-magnification microscope to collect the image data required for the model. The rest of our device will be the housing for the microscope and the Google Coral AI board.

## 3.5 Design Analysis

So far, the design proposed in 3.3 is feasible. Almost all of our team's current work is carried out around TensorFlow. When performing machine learning models, improving accuracy is crucial. Therefore, we have mentioned in the proposed design how to improve accuracy and reduce errors. It also describes what we will do if we encounter errors.

## 3.6 Development Process

First of all, in our senior design project, I think we are more suitable to use waterfall development as our main development process.

In waterfall development, it can be mainly divided into Requirements analysis resulting in a software requirements specification, Software design, Implementation, Testing, Integration, if there are multiple subsystems, Deployment (or Installation) and Maintenance. Such a process is closer to our senior design project.

In our senior design project, first we need to prepare and analyze the product to understand its scope and background. Then, we will use TensorFlow as our main software to design our senior design project. Since we are contacting and using TensorFlow for the first time, we will spend a lot of time learning how to use it. In addition, we will carry out our projects according to the needs of customers. Secondly, in specific operations, we may use TensorFlow in the colaboratory, the purpose is to design how to predict the relationship between bacterial growth rate and antibiotics. Again, we will conduct multiple different tests on our design results to verify our hypothetical views. If the results in the test are significantly different from the hypothesis and experimental errors, we will re-run the third experiment. Finally, we will provide customers with the complete experimental design and results, and provide customers with a satisfactory solution.

## 3.7 Design Plan

For the hardware perspective, we will load our code into a raspberry pi which connects with the video collection unit and power supply module so that the user will be able to use this portable device to scan the E.coli and input it into the microcontroller and finally get the predicted resistance of E.coli as the output. (although we haven't dive into the hardware too deep this semester, we mainly focus on the TensorFlow part this semester.)

For the software perspective, firstly, we will have a detection network to locate any object in the given image. Secondly, we will use the machine learning model to identify the objects based on the input of the detection network. Then, we will monitor the state of bacteria over the timeline dimension  to determine the resistant and non-resistant bacteria. Finally we will use the monitoring program to display the output.

# 4 Testing

To properly test our prototype we will need to perform unit tests as we build our prototype. All sub components will need to be tested before being combined into the final product. This includes the components for the DNN and the components for the incubator. All interfaces we use will also need to be tested which includes the final DNN and our user interface for the incubator. Finally, we will need to perform acceptance testing to confirm that we meet our design requirements. Because of covid we will have to take extra precautions when performing user testing to ensure the safety of the group and our participants.

As the year progresses we will be updating this section of the document with the test specifics, the results from the tests, and changes we make to the prototype in response to the tests. In the section below we detail the specifics of these tests we will be performing on our prototype as it is constructed. By the end of the year we predict that we will have a prototype that will pass all tests.

## 4.1 Unit Testing

Unit testing is a software testing method by which individual units of source code - sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures - are tested to determine whether they are fit for use. Assuming external libraries and frameworks are working correctly, software units being tested in isolation include the User Interface as feature input such as images and timestamps. Classification and Image machine learning models will be unit tested individually and then together as a unit. We will continue to test our code that will be used the most, repeatedly changed code, and code later on that could generate many bugs in the future.

## 4.2 Interface Testing

An interface is a programming structure/syntax that allows the computer to enforce certain properties. Many interfaces will be tested in our design project. A relevant interface that will be tested in the future includes our machine learning model itself, as our overall function will be able to call on it to take advantage of the raw images and timestamp input. Testing this will involve passing in a pre labeled dataset it has ever seen before and recording its accuracy at predicting the bacteria.

## 4.3 Acceptance Testing

We will verify that our functional and other requirements are being met by own design with testing and user studies. We will test the functionality of the DNN and the physical interfaces separately. To test the accuracy of the model we will determine the accuracy using the testing dataset and again using the live data we collect with the DNN inside the incubator. If 90% of the predictions made are correct then the prototype's accuracy is acceptable. If a user is able to analyze at least 10 mins of footage, upload and save footage from the device then it meets those requirements.

To test the portability of our device we will create a poll of randomly selected people in Ames, hand

them our device, and ask them on a scale of 1 to 10 how easy our device is to carry. If a majority of users say it is acceptable (6 or above) we will have succeeded in making a handheld device. To test the ease of use as well we will ask these same people to interpret either direct readings or a copy of the readings from the device after incubating bacteria. If a majority of users are able to understand the basics of the interface we will have succeeded.  While testing we will involve our client in each step of the testing process.

## 4.4 Results

At this time we have not yet completed any tests as we are still in the process of implementing our ML model. However, the model will be put through the various acceptance tests mentioned previously. As for the physical device, we have not yet completed our design for the model. However, it will also be put through the testing mentioned above. We expect that our finalized device will pass all the tests.