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

- Unified Modeling Language
- IEEE 802.11 standard for Wireless LANs
- ISO/IEC/IEEE 24765:2017
- Google's Machine Learning Workflow

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)[5]. 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 and outbreaks.

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.

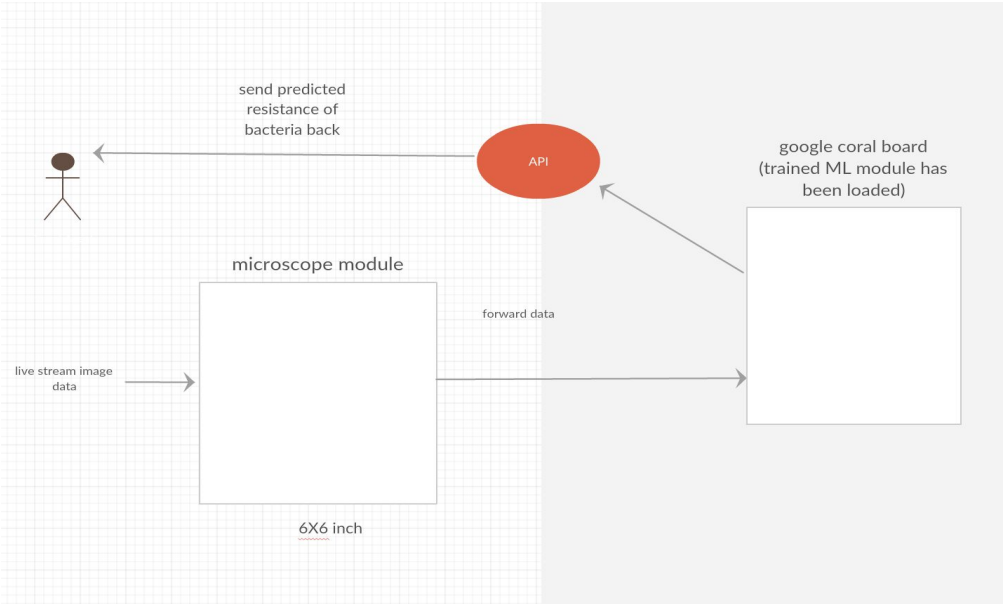


Figure 1.5.1 - Use-case diagram

1.6 Assumptions and Limitations

Assumptions:

- The end product will be used in a setting with a power outlet.
- The user will be able to provide some form of external storage.
- The user will have access to a secondary device to connect to our product.
- The user will have access to petri dishes, bacteria, and the necessary materials required to grow bacteria.
- The user will know how to properly handle the bacteria including disposal.
- The end product will not be used outside the United States.

Limitations:

- The end product will be no larger than 6”x6”x6” as specified by our client.
- We will use the google coral board, accelerator, and the coral camera in our product.
- We will need to collect our own datasets as there are no major e-coli datasets for machine learning.
- The cost to produce our design will not exceed $500 as specified by our client.
- The system must operate at 120 volts and 60 Hz inorder to be compatible with US outlets.

1.7 Expected End Product and Deliverables

Microscope (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.

Embedded Software (May 2021)

- The microscope will have an embedded software responsible for communication with an external device as well as handling the data collected from the microscope. It will also be responsible for running the image analysis using the ML model. The application will provide a user interface to interact with the various features mentioned previously.

Machine Learning Model (February 2021)

- The ML model will be created using TensorFlow 2.0. The model will be capable of running on the Google CoralAI platform using TensorFlow lite. 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

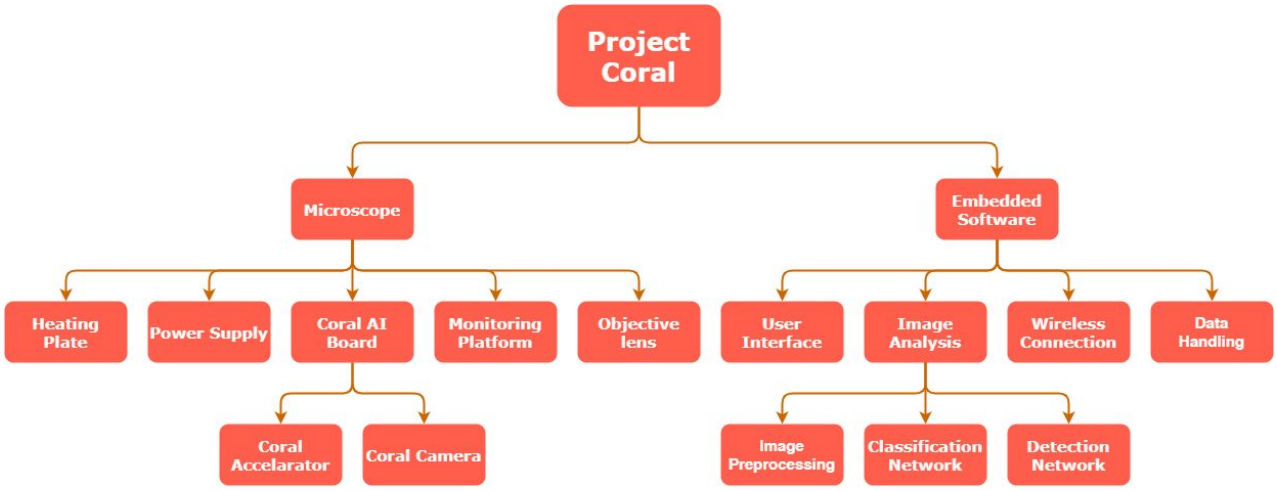


Figure 2.1.1 - Graphical representation of the tasks and their dependencies

There are two major components to complete, the microscope and the embedded software. The microscope is responsible for collecting the image data to be fed into the machine learning model. It will also be housing the Google Coral AI board which will be running the software required to manage the microscope and run the image analysis. The embedded software will also be providing a user interface through a wireless connection which users can connect to using another device capable of connecting to a wireless network like a laptop or phone. Below is a breakdown of the sub-components which are required for the core components to function.

- Microscope
    - Heating plate
        - Maintains the petri-dish at the optimal temperature for bacteria growth
    - Power Supply
        - Provides power to the components in the device such as the board
    - Google Coral AI Board - Single board computer with an Edge TPU
        - Coral Accelerator
            - Will be used by the ML model for improved performance
        - Coral Camera
            - Will be used to collect image data from the microscope
    - Monitoring Platform
        - Moves the petri-dish to allow the microscope to monitor the entire area
    - Objective Lens
        - Magnifies the view of the microscope to the required level

- Embedded Software
    - User Interface
        - Allows the user to configure and manage the device and can be accessed wirelessly through a secondary device
    - Image Analysis
        - Image Preprocessing
            - Stitches the images from the microscope to provide an image of the entire petri dish and steps such as background removal, grayscale filter, etc.
        - 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.
    - Wireless Communication
        - A wireless network will be broadcasted from the Coral AI Board
    - Data Handling
        - This portion of the software will be handling the storage of the results from the ML model and the data collected by the microscope

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 2.4.1 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. 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 2.4.2.

TASK TITLE	START DATE	DURATION
Developing ML Model		
Research time-series models	9/30/20	6
Develop Bacteria Detection Network	9/30/20	14
Develop Bacteria Classification Network	9/30/20	14
Bacteria Classifier based on growth rate	10/28/20	14
Combine the two networks	10/14/20	14
Put together and test/tweak	11/11/20	7
Monitoring/Runner Program	10/14/20	14

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 creating a CAD model. The timing for each step can be seen in figure 2.4.3 below.

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

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 already have, components we need to print, and components we need to purchase, 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 and Colaboratory to store documents and jupyter notebooks, 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 purchasing materials we do not already have for the creation of the portable system's prototype.

3 Design

3.1 Previous Work And Literature

Our project is heavily inspired by a research paper published in *Light: Science & Applications*[5]. This paper proposes the use of a lens-free holographic image capturing device to feed two DNNs to identify growing bacteria. Our project has decided to forgo their image capturing device as it does not provide the magnification needed to see bacteria on an individual level. Instead, we will be using a more traditional microscope with a lens capable of seeing individual bacteria. However, we will be incorporating their idea of using a two-stage network structure to identify growing bacteria. Our model structure will be using their research as a basis especially their addition of the time dimension to the input of their models. Additionally, their process requires colonies of bacteria to develop which requires up to 24 hours of incubation time. Our idea is to look at individual growth rates that will significantly reduce the time required to make a prediction.

Advantages

- Neural Network works with respect to time

Shortcomings

- Does not look at individual bacteria, looks at a colony
- Very slow due to looking at colony

Additionally, we will be using the OpenFlexure microscope design as the basis of our microscope[2]. We chose this design due to it being open-source and meeting our budget. We will be extending the original design and adding a heating plate to incubate bacteria as well as modifying the UI/software of the microscope to incorporate our machine learning model and data collection needs.

Advantages

- Open-Source design and software
- High Quality
- High magnification

Shortcomings

- No heating plate

3.2 Design Thinking

Our goal in the define phase was to narrow our project scope down and determine exactly what our end product needed to do to be considered successful. We had input from our client and completed research on previous designs and cost effective microscopes.

In the ideate phase, we spent a lot of time researching various solutions based on our definitions from the define phase. We researched various microscopes and existing machine learning models to create our own design.

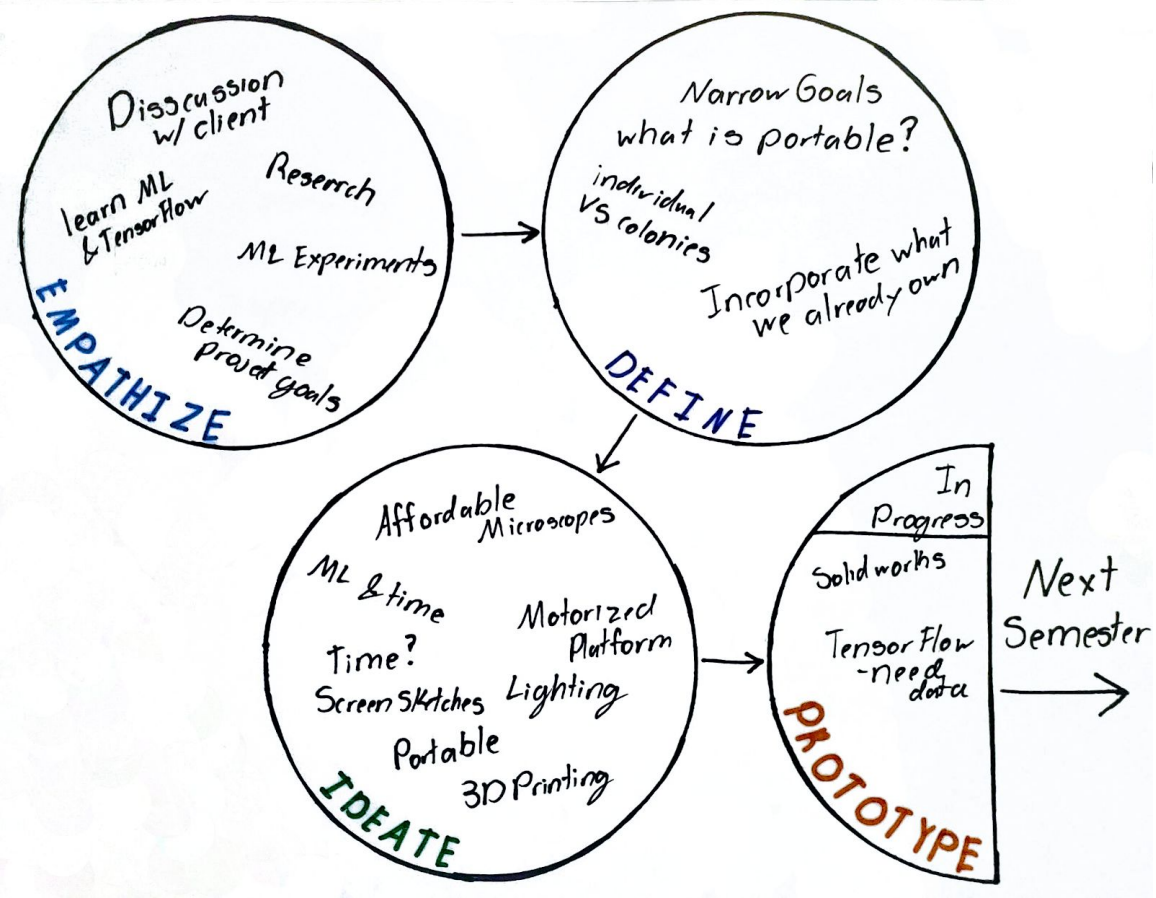


Figure 3.2.1 - Conceptual Design Process Diagram

3.3 Proposed Design

Our proposed design is composed of two major components: the microscope 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 (the materials we do not already have access to), we will be able to satisfy the budget constraint.

The microscope we are creating has four modules we are constructing: a motorized platform, heating, microscope apparatus, and an interface. While researching cost effective microscopes we found the OpenFlexure microscope project. We will be using a modded microscope from this project to collect the data from our samples needed by the machine learning model. To ensure that the microscope is portable we will be keeping the coral board, accelerator, heating circuit, etc inside the 3D printed microscope compartments. The microscope we create 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 step motors to move the microscope across the petri dish (similar to a 3d printer) and stitching together the resultant images. To mod the OpenFlexture microscope we will need to change the settings in the build commands, replace the petri dish holder with the heating element, and print extra storage for the heater controller. The OpenFlexure design includes plans for a motorized platform, lighting, and storage for electronics.

Figure 3.3.1 - Exploded View of the OpenFlexure Microscope

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 plans and supplies for this will be provided by our client next semester.. The last component of the microscope 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 microscope making changes as needed following the instructions outlined in the OpenFlexture documentation [2].

Predicted Materials and Costs:

Component	Quantity	Already Own?	Cost per Unit	Where?	Total Cost
3D Printed Parts	1	N	$100	Based on various blogs online	$200
M3 Nuts (brass)	6	N	$7 for 100	Amazon	$7
30mm M3 Hexagon-head screws	3	N	$.52	Accu.co	$1.56

Washers	5	N	8 for $1.28	Lowes	$1.28
8mm M3 Screws	10	N	2 for $1.98	Lowes	$9.90
White LED, 3mm	1	N	$.18	lighthouse LEDS	$.18
40 Ohm resistor	1	N	$.15	Digi-Key	$.15
Various wiring	X	Y	X	X	X
Rubberbands	10	Y	X	X	X
Google Coral Board	1	Y	X	X	X
Google Coral Board Camera	1	Y	X	X	X
Google Coral Board Accelerator	1	Y	X	X	X
28BYJ-48 micro geared stepper motors	3	N	$12 for 5	Amazon	$12
Heating element and circuit	1	Y	X	X	X
Microscope lens objective	1	Y	X	X	X
1in petri dishes	3	Y	X	X	X
				Total	$232.07

Figure 3.3.2 - Table of materials and Approx costs

*These prices are from online vendors. As we go home over break we will scavenge our houses for appropriately sized screws, resistors, etc. We also will checkout in person stores that do not have prices online (Covid allowing) for better deals.

3.4 Technology Considerations

Technology	Strength(s)	Weakness(es)
TensorFlow	• Popular ML framework developed by Google • Lots of pre-existing models • Well-documented • Deploy to many formats	• Large learning curve since most of our group has not used it before • Low level framework therefore it has some complicated code
Google Coral A.I. Dev Board	• Optimized for ML deployment • Developed by Google and supports TensorFlow lite	• Very Expensive
Traditional Microscope	• High magnification • High accuracy • Modular	• Very very Expensive • Heavy & awkward to carry
OpenFlexure Microscope	• Open source software • Many features already designed (camera holder, motorized platform, and controlling software provided)	• Success of microscope depends heavily on the quality of the 3D print

Figure 3.4.1 - Strengths and Weaknesses of the Technologies in our project

We will be using the OpenFlexure microscope instead of a traditional microscope to meet the financial requirements of our project. We will utilize a higher quality 3D printing process to ensure the platform is frictionless enough (high accuracy flexures) to allow for smooth steps when observing samples. This will also allow us to select materials that will withstand the heat from the heat plate.

3.5 Design Analysis

So far, the design proposed in 3.3 is feasible. 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. The success of the microscope depends on the magnification of the provided microscope objective, the quality of the 3D printed components, and

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 microscope. All interfaces we use will also need to be tested which includes the final DNN and our user interface for the microscope. 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 microscope, we have not yet purchased or gathered all of the required components to construct it. However, it will also be put through the testing mentioned above. We expect that our finalized device will pass all of our tests.

5 Implementation

Embedded Software:

There are several milestones the software team must accomplish in order to successfully complete the embedded software required for the microscope. The first of which is to successfully train and evaluate our machine learning model. First, we must implement the custom layers we will need to compile the model and then we can combine all the layers to complete our convolutional neural network.

After the model is implemented, we will begin training the network in Google Colaboratory. Some of the training scripts are already prepared and will simply need to be customized at the time of training to accommodate our new model. This training will be occurring in the background as we will be switching focus to work on the other components of the embedded application, specifically the UI.

Once the model has begun training, we will implement the UI by extending the interface code which is provided by the OpenFlexure microscope. This code will primarily be written using Vue JS as that is the framework used by the microscope's UI. We will be modifying the UI to match the screen sketches provided in the Appendix 1. Once the model is finished training and passed our evaluations, we will be writing the code to feed the data from the microscope into the model and completing the full loop.

Hardware:

We need to get access to the exact microscope objective and ensure that we print the correct adapter for the microscope. To reduce printing costs, we will use a personal printer as much as we can. However, some components will need to be printed with precision or be able to handle 37 degrees celsius without melting. For these components we will use Shapeways' printing services. While we wait, we will source and buy the other components listed in section 3.3. Once we have all the required components we will construct the microscope.

The next step is to upload the software to control the microscope to the google coral board. Once we have the microscope constructed and the software to use it, we will test it before moving on. This will ensure that our objective is strong enough to see the individual bacteria and that we are able to obtain images from the setup. The next step is to attach the step motors and make sure that the motorized platform works and that we can stitch together the images.

We then will connect the board to the interface created by the software team and upload the tensorFlow model. Once the whole project is created, we will complete our integration tests to ensure we met the goals/requirements.

6 Closing Material

6.1 Conclusion

This semester we have laid out the plan for our project. We have a TensorFlow model created and ready to train, parts ready to be 3D printed, components listed and ready to purchase, and screen sketches to be implemented. We also took the time this semester, by request from our client, to each learn and experiment with various TensorFlow models. The first part of this semester was dedicated to researching TensorFlow and previous designs of similar projects. We then narrowed our project to create goals such as time limits, size limits, magnification, etc. The second part of the semester was taking what we knew, and attempting to implement a time based machine learning model for looking at the growth rates of E-coli, and finding the necessary parts to create a device to obtain the sample data. We will be using the open source OpenFlexure microscope instead of a medical microscope to adhere to the budget, and size constraints. We also will be using tensorflow for our machine learning model as the Google Coral board is optimized to run inference for tensorflow lite. Next semester we are ready to start the implementation of our project.

Our goal for our senior design project is to create a device using the Google Coral board, accelerator, and camera to count the number of penicillin resistant bacteria in a sample. Previously, this process took up to 24 hours. Our goal is to look at the individual bacteria instead of the colonies and produce the same results in less than an hour.

6.2 References

1 DeFlorio-Barker, S., Wing, C., Jones, R. M., & Dorevitch, S. (2018). Estimate of incidence and cost of recreational waterborne illness on United States surface waters. *Environmental Health*, *17*(1). https://doi.org/10.1186/s12940-017-0347-9

2 Bowman, R., Stirling, J., Mwakajinga, G., Sanga, V., & Meng, Q. (n.d.). OpenFlexure Documentation. Retrieved November 15, 2020, from build.openflexure.org website: https://build.openflexure.org/openflexure-microscope/v6.1.4/docs/#/

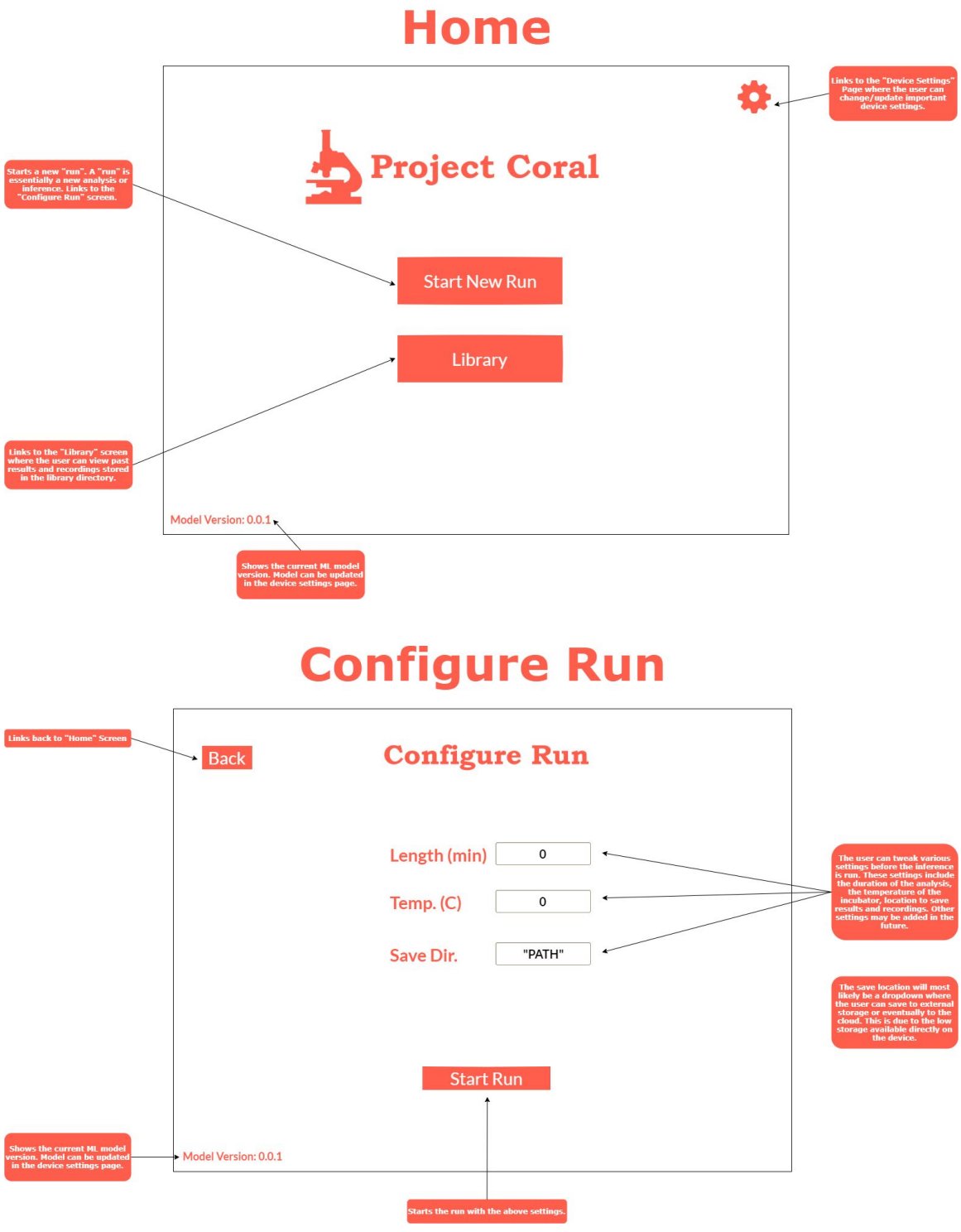
3 Dane. (2020, July 1). Google's New "Context R-CNN" Uses Past Images to Improve Detection. Retrieved November 15, 2020, from Neruohive website: https://neurohive.io/en/news/google-s-new-context-r-cnn-uses-past-images-to-improve-detection/

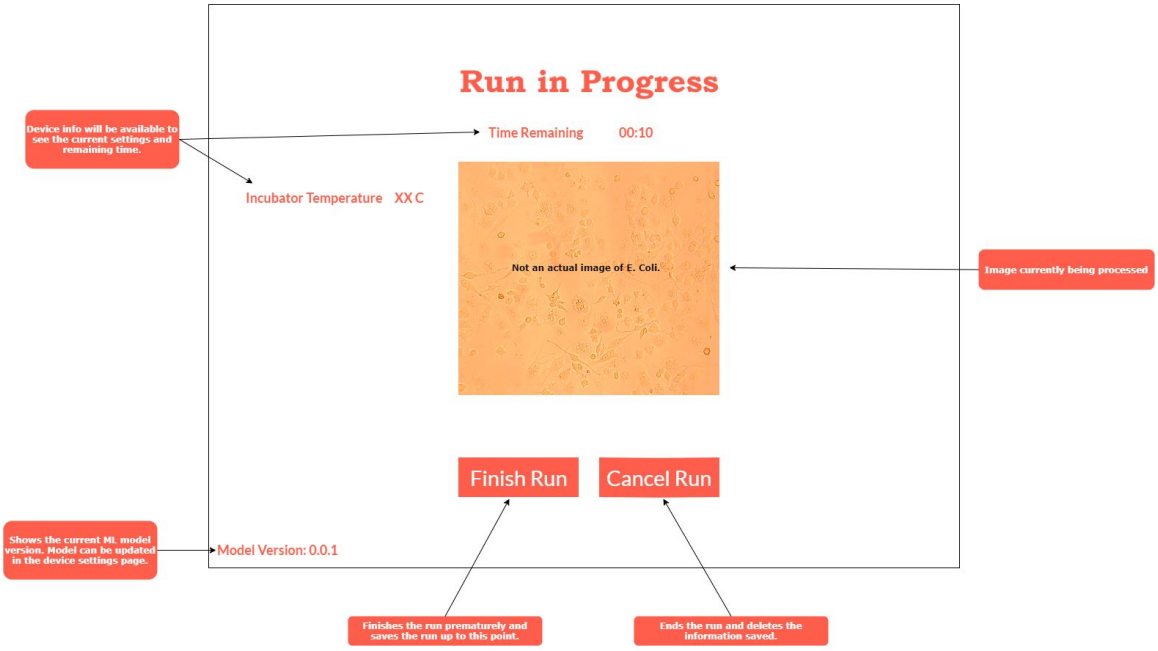
4 Wang, H., Ceylan Koydemir, H., Qiu, Y., Bai, B., Zhang, Y., Jin, Y., … Ozcan, A. (2020). Early detection and classification of live bacteria using time-lapse coherent imaging and deep learning. *Light: Science & Applications*, *9*(1). https://doi.org/10.1038/s41377-020-00358-9

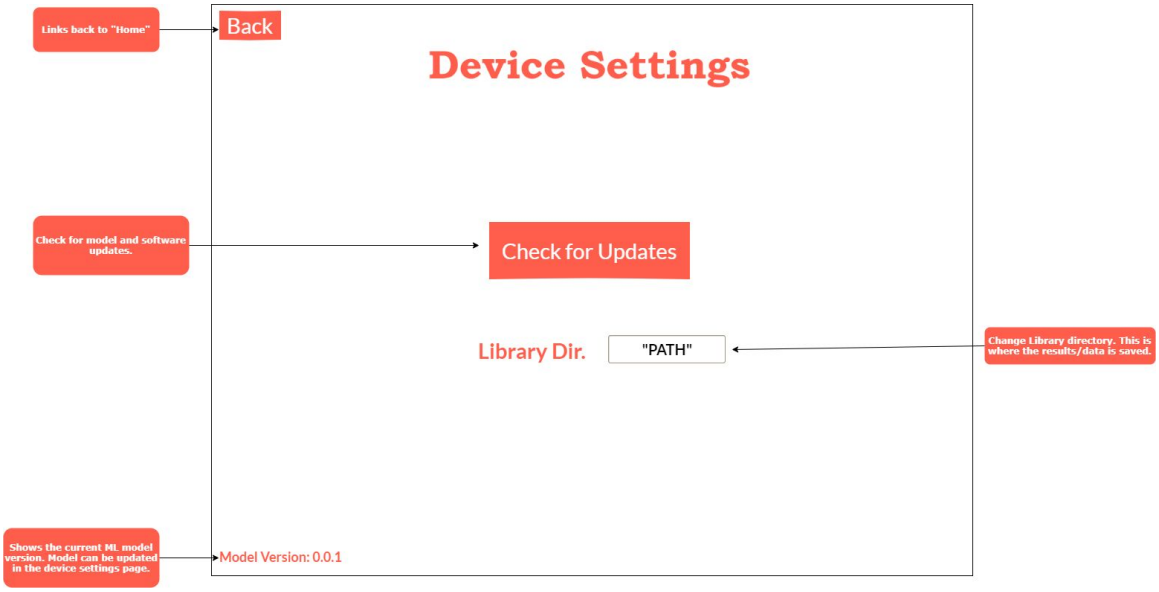
5 World Water Day | Drinking Water | Healthy Water | CDC. (2020, October 26). Retrieved from www.cdc.gov website: https://www.cdc.gov/healthywater/drinking/world-water-day.html?CDC_AA_refVal=https%3A%2F%2Fwww.cdc.gov%2Fhealthywater%2Fobservances%2Fwwd.html

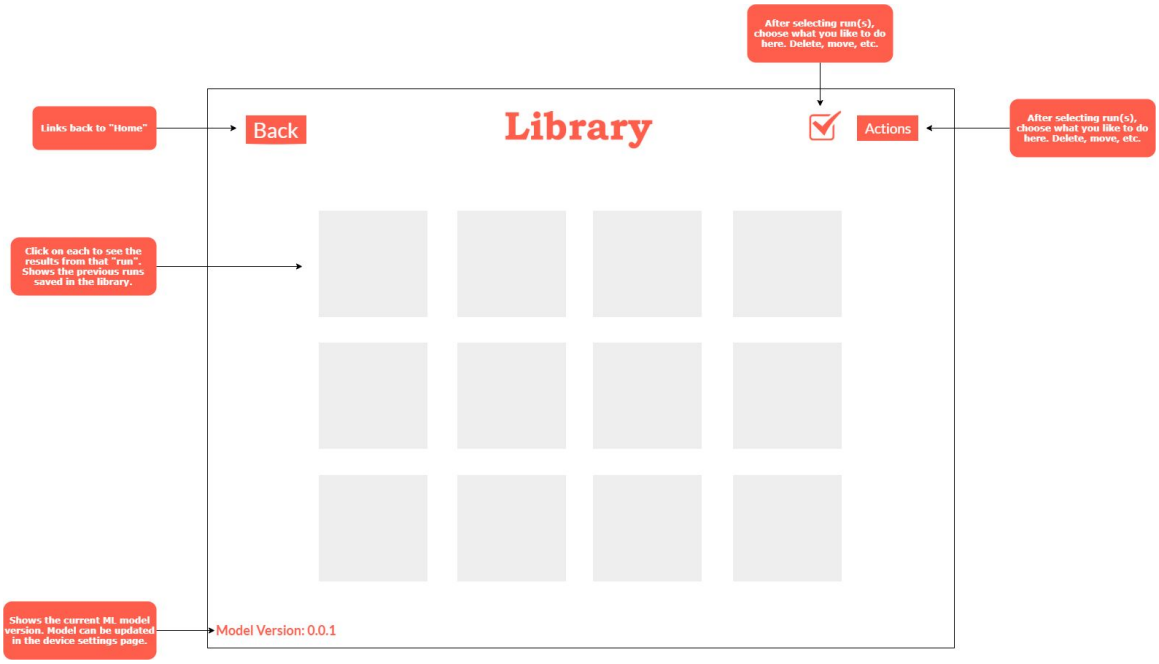
6.3 Appendices

Appendix 1. Screen Sketches of UI.




Run in Progress
Device info will be available to see the current settings and remaining time.
Time Remaining 00:10
Incubator Temperature XX C
Not an actual image of E. Coli.
Image currently being processed
Finish Run
Cancel Run
Shows the current ML model version. Model can be updated in the device settings page.
Model Version: 0.0.1
Finishes the run prematurely and saves the run up to this point.
Ends the run and deletes the information saved.


Links back to "Home"
Back
Device Settings
Check for model and software updates.
Check for Updates
Library Dir.
"PATH"
Change Library directory. This is where the results/data is saved.
Shows the current ML model version. Model can be updated in the device settings page.
Model Version: 0.0.1


After selecting run(s), choose what you like to do here. Delete, move, etc.
Links back to "Home"
Back
Library
Actions
After selecting run(s), choose what you like to do here. Delete, move, etc.
Click on each to see the results from that "run". Shows the previous runs saved in the library.
Shows the current ML model version. Model can be updated in the device settings page.
Model Version: 0.0.1