

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

FINAL REPORT

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: 4/25/2021

Table of Contents

1 Introduction	3
Acknowledgments	3
The Team	3
Problem and Project Statement	3
Operating Environment	3
Functional and Nonfunctional Requirements	4
Intended Users	4
Engineering Standards	4
2 Project Design	4
Previous Literature	4
Proposed Design	5
Machine Learning Model	5
Embedded Software	7
Microscope	8
Functional Decomposition	10
Project Timeline	11
Evolution of Design	11
Engineering Constraints	12
Security Concerns	12
3 Implementation Details	14
Embedded Software	14
Hardware	14
4 Testing Process and Results	17
Testing Environment	17
Testing Strategy	17
Results	18
5 Operational Manual	20
Introduction (Vocabulary)	20
Features	20
First Time Setup	20
Walk Through	21
Start Up:	21
Collecting the Data:	21
Data Analysis:	22
Device Settings	22
Maintenance	22
Hazards	23
6 References	24
7 Appendices	25

1 Introduction

1.1 ACKNOWLEDGMENTS

We would like to thank our project advisor Dr. Meng Lu, his teaching assistants Shirin Parvin and Qinming Zhang, and our teaching assistant Rachel Shannon. This project would not have been possible without their help.

1.2 THE TEAM

Our team consists of computer, electrical, and software engineers. Skills from each major were required to complete this project. Our team members consisted of: Declan Costello (SE), Iris Top (CprE), Shijin Huang (EE), Wenjie Jin (EE), Owen Schacherer (SE), and Anirudh Tangellapalli (CprE).

1.3 PROBLEM AND PROJECT STATEMENT

Macrophage are important cells of the immune system that are formed in response to an infection, after accumulating damage, or dead cells. Macrophages are large, specialized cells that recognize, engulf, and destroy target cells. Due to the importance of the macrophage, it is imperative for doctors to identify macrophage among other cells. The goal for our project is to build an effective machine learning (ML) model that is able to analyze a short video and distinguish all macrophage cells from other objects and to create a device that is capable of collecting this data.

1.4 OPERATING ENVIRONMENT

The intended location for operation of our design is inside a biology lab. We assume the user will have access to Wifi and an outlet. The microscope is small but as a result it can be broken easily with improper use. While in use, it is expected that the device will remain in the same location.

1.5 FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS

When designing the device, we had several functional requirements we were expected to meet. Our final project was expected to be able to take and store video feed of the macrophage sample. The machine learning model must be able to process and analyze at least 10 minutes of this video feed. The machine learning model must be able to effectively recognize and categorize the macrophage cells with 90% accuracy.

For the nonfunctional requirements perspective, the product should be developed under a \$500 budget including the components we were provided. Team members should follow all safety

precautions while working in the lab. All lab substances must be used and disposed of correctly. Lastly, everyone should take care of, and be responsible for our lab equipment.

1.6 INTENDED USERS

One of the goals of this project is to make the process of identifying macrophage cells possible for anyone. By having a ML model trained to detect macrophage, the user of this device does not need to know what the cells look like. However, due to the hazards of working with human cells, the device is only intended to be operated by those who are trained to work with biohazardous material.

1.7 ENGINEERING STANDARDS

We have applied the ISO/IEC/IEEE 24765:2017, ISO 8037:1986, and Google's machine learning workflow standards while we were building this project.

2 Project Design

2.1 PREVIOUS 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

2.2 PROPOSED DESIGN

Our proposed design is composed of two major components: the microscope and the embedded software which contains the machine learning model and GUI. Last semester, we designed a custom neural network based off of the model in the original paper and we modeled our microscope based on the OpenFlexure microscope. We also designed our user interface and decided to use JavaFX to implement it. However, over the course of this semester we have had to make several changes to our entire design due to change of focus from monitoring E. Coli to macrophages as well as some other smaller issues.

MACHINE LEARNING MODEL

Last semester, we trained several neural networks 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 were also working to set up equipment for taking videos of E. coli which will be used for training our time-dependent model.

Our previous machine learning model was going to be different from other common image classification models because we required it to not only detect the bacteria but also to monitor the growth rate. A typical object detection neural network can not achieve this result because it simply does not recognize the input image as part of a larger set, i.e. it ignores the context necessary to make a prediction on the future growth. Therefore, the model needed to include an additional input dimension for time to get an added total of 4 input dimensions: RGB values, and time. To account for the added dimension, we were planning to use custom conv3D layers as opposed to typical conv2D layers. 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. However, since our target changed from monitoring E. Coli growth to macrophage identification, we are able to use a simpler object detection model to detect and track the cells.

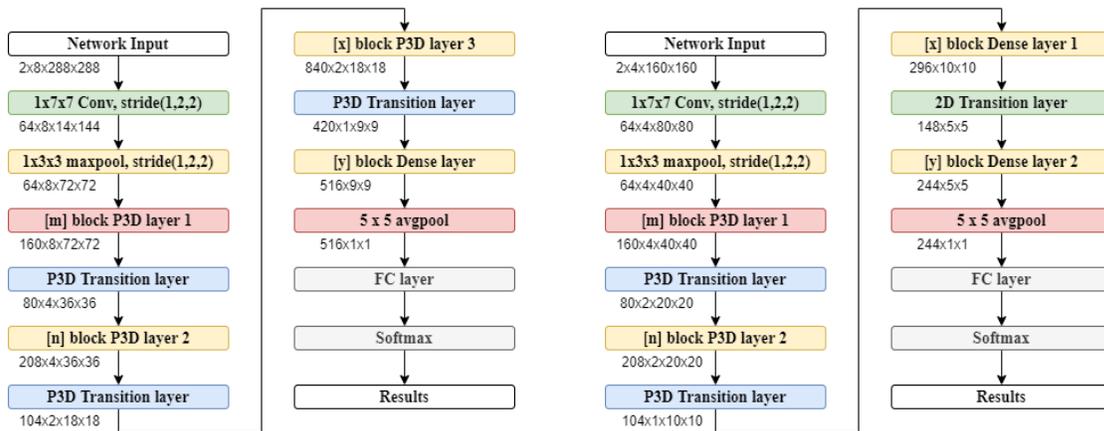


Figure 2.2.1. Stages of first network. The first stage (left) was responsible for locating the bacteria in the image. The second stage (right) then classifies the bacteria based on the growth rate.

At first, we decided to use an existing convolutional neural network, Faster-RCNN. Faster-RCNN is a widely used object detection network known for its speed and reliability which already has several implementations, even one provided by TensorFlow’s model zoo. We fine-tuned the Faster-RCNN model provided TensorFlow using the m-zero dataset obtained from the lab. However, after training the model, we were unable to convert the model into a form usable by the Google Coral AI board. The board only accepts TensorFlow-Lite format models and Faster-RCNN uses custom layers and other properties which cannot be ported to TF-Lite format. Therefore, we decided to change our model architecture once again to Single Shot Detector (SSD).

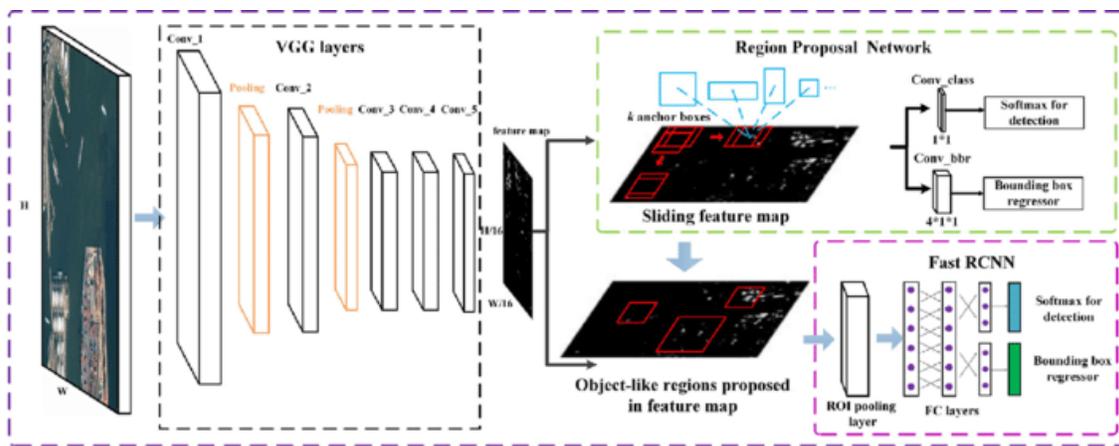


Figure 2.2.2. Diagram of Faster-RCNN.

The SSD model is fortunately also provided by TensorFlow and is well-known for its use in mobile devices. It is the most popular object detection network for mobile inference because of its speed without impacting its accuracy significantly. It is also one of the few models which can be easily ported to the TF-Lite format. We fine-tuned the SSD model using our initial m-zero dataset and

with our newly acquired m-zero and m-one mixed dataset. This model is capable of identifying both m-zero and m-one type macrophage.

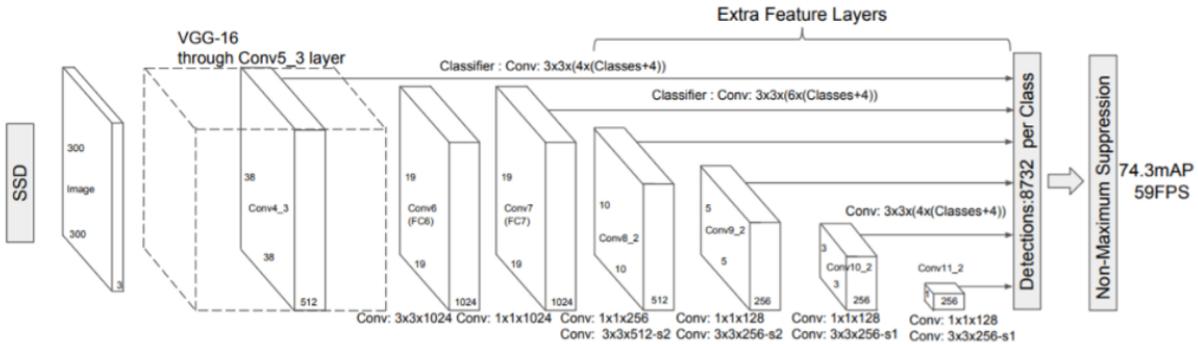


Figure 2.2.3. Diagram of SSD model.

After the model has been trained, the TensorFlow graph is converted to a TensorFlow-Lite graph. The TF-Lite graph is then exported to a TF-Lite model which is ready for inference on the Google Coral AI board. An important note to keep in mind is the adaptability of the model to learn new targets. Our model, if trained with new data, should be able to recognize other types of cells or bacteria.

EMBEDDED SOFTWARE

The user must be able to easily access and control the microscope. To accomplish this, we have designed and implemented a GUI using ReactJS. The board must be connected to a wireless network, which will be available in the lab. The React app will then be hosted on the board using a lightweight Apache web server. The board does not need to be accessed from outside of the local network, therefore, it is sufficient to simply run the web server and access the website using the board's IP address. The website does not require any additional network configuration.

The website is composed of five screens/pages which provide an easy workflow for the user to run an analysis using our microscope. The user will start on the Home page to access the Library or Device Settings pages, or they can begin an analysis by going to the Configure Run page. They can configure various settings for the analysis such as the duration and where to save the files, if different from the default settings. While the analysis is in progress, the user will be on the Run in Progress screen. This simple workflow should be easily accessible by a wide range of users.

When the user decides to run an analysis, the configuration from the Configure Run page is used to start running a separate process. This process is responsible for controlling the microscope and photographing the cell samples. The images will be fed directly into the Machine Learning model described in the previous section and the images will be saved in the location specified in the configuration.

MICROSCOPE

The microscope we are creating for this project will be based on the OpenFlexure microscope [2]. While researching cost effective microscopes we found the OpenFlexure microscope project. This will be a cost effective way to mount the provided endoscope and have control over the x, y, and z (fine focus) axis of the sample. 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, LED, endoscope, etc inside the 3D printed microscope compartments.

Mods for the microscope will include an endoscope mount, a base that allows the endoscope to go through it, legs to raise the height of the microscope, and a more powerful LED. Below is an exploded view of the OpenFlexure microscope as well as designs for the feet and the mount.



Figure 2.2.4 - Exploded View of the OpenFlexure Microscope

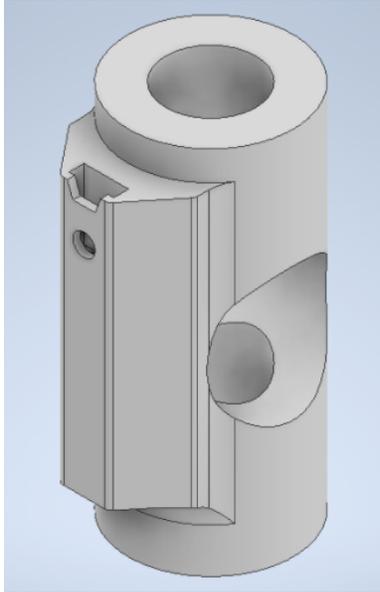


Figure 2.2.5 - CAD model of the modified endoscope mount

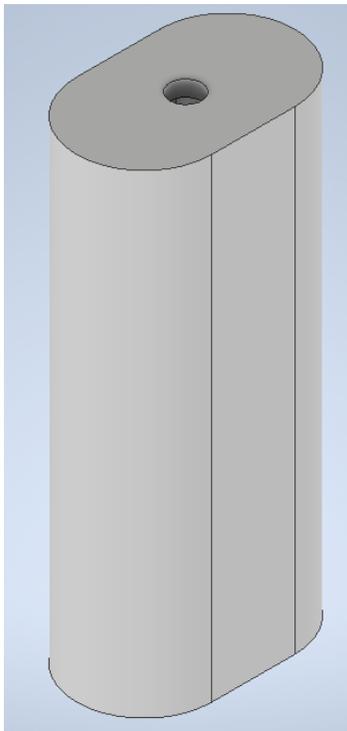


Figure 2.2.6 - CAD model of a microscope foot

2.3 FUNCTIONAL DECOMPOSITION

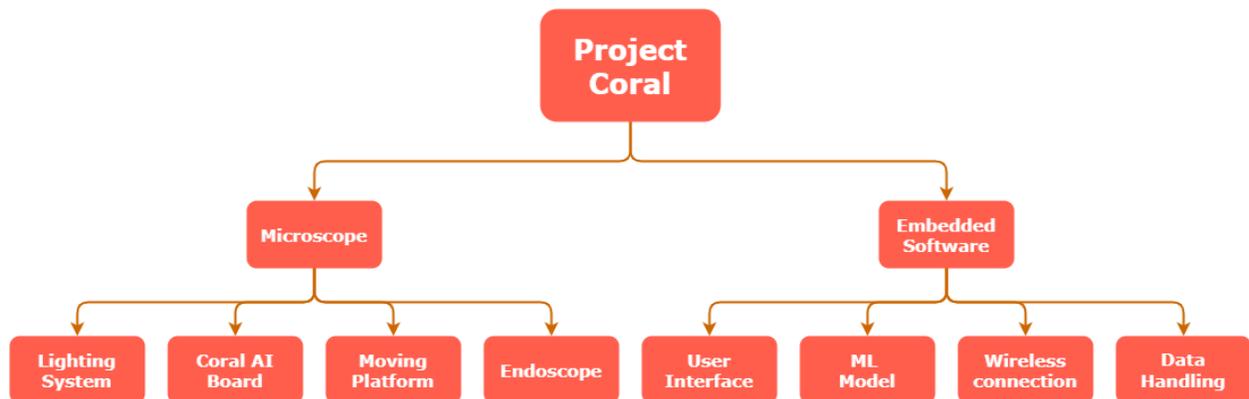


Figure 2.3.1. Functional Decomposition of the device.

There are two major components to complete, the microscope and the embedded software, which also includes the machine learning component. 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
 - LED
 - Provides top down light to allow for clear images to be taken
 - Google Coral AI Board - Single board computer with an Edge TPU
 - Runs the embedded software, specifically the machine learning model.
 - Endoscope Microscope
 - Magnifies the view of the sample to the required level and captures images
- Embedded Software
 - User Interface
 - Allows the user to configure and manage the device and can be accessed wirelessly through a secondary device
 - Machine Learning Model
 - Image Preprocessing
 - The images from the microscope must be converted to grayscale and the image will be cropped to 300x300 pixel range in the center
 - The model is capable of identifying Mo and M1 macrophage cells.
 - Wireless Communication
 - The Google Coral AI board will need to be connected to a wireless network to be controlled
 - 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

2.4 PROJECT TIMELINE

During the first semester, we worked on creating the design to fulfill our client's requirements. The second semester we had to rework our design based on our client's changing requirements and implement our design.

2.5 EVOLUTION OF DESIGN

First semester, we were given the goal of determining if E-coli bacteria was resistance to penicillin based on its initial growth rate. We aimed to have results in less than 20 minutes. To do this we had to make sure to produce the ideal growing conditions and look at the individual bacteria. This required a high magnification objective, a heating element, an automatic moving platform, a lighting system, and a much more complex neural network. Due to Covid, our advisor changed our project scope to identifying different types of macrophage. This meant that we no longer needed the heating element, the motorized platform, or as complex of a network.

We started with a custom neural network (NN) for growth analysis and classification. We changed to the Faster RCNN model for generic object detection. Finally, we pivoted from the Faster RCNN model, because it is not convertible to a TensorFlow-Lite format or tpu execution. We needed our model to be a TensorFlow-Lite model to work on the Google Coral Board. We used the Single Shot Detector MobileNet NN for our project.

After discovering the website Roboflow.com, we started using this tool instead of python LabelImage to perform image annotation. Roboflow is good for preprocessing, datastorage, team collaboration, etc. We made this change for the sake of saving time and the superior data annotation tools.

Additionally, we changed the communication method between the user and the Google Coral board. The original idea was to use an Ad Hoc network hosted on the board itself and using a detached JavaFX application. The JavaFX app would be run on a secondary device and would connect to the board using a socket connection. However, the socket connection was simply too slow to transfer any data, therefore we switched to a ReactJS website hosted on the Coral board through an Apache web server. Due to this change, our device now requires a WiFi connection to be fully functional. However, this is not a major setback because the microscope is used in a lab environment, therefore, we can assume a wireless network will be present.

2.6 ENGINEERING CONSTRAINTS

We had many constraints while working on this project. The three main types we faced were health and safety, economic, and time constraints. The Covid-19 pandemic started before our senior design team formed and has continued to this day. This has caused major changes in our project objective to resolve any health concerns. We also almost met entirely online. To make up

for any loss of communication from meeting online, we met at least once a week with our project advisor to remain on track and aware of the project's progress.

We were also given a \$500 budget. This led to the design decision of using a 3D printed microscope and parts our adviser already had in his lab. We were also constrained by the amount of time we had to work on this project. All of our members are seniors/grad students and as a result have a lot of other projects going on at the same time. We also had slightly shorter semesters than normal due to Covid.

2.7 SECURITY CONCERNS

No one in our group has any formal experience in cyber security. As a result, our project may be susceptible to attack. Our wireless connection is not protected and we require a static ip address in order to be able to easily connect to the device. However, this implies anyone on the local network could connect to the device. As our device is not medical grade as is, no high risk samples should be analysed using our device (no medical patient data).

Because we were working with large amounts of data (code, images, labels, etc.), data loss was a major concern in our group. To combat this, we used RoboFlow, Google Drive, Microsoft Teams, GitLab, and personal devices to store our data.

3 Implementation Details

3.1 EMBEDDED SOFTWARE

We used Python to train our neural network, split the images, and to test our model. We used Python because it is easier to use than C++ when building a Tensorflow model. It is also a very accessible language with many easy to follow tutorials online. To build the web application, we used reactJS. We choose reactJS based on previous experience and their quick start option. This allowed us to set up a basic application very quickly.

We used Tensorflow to build and retrain our single shot detector (SSD) neural network. The Google Coral Board is optimized to use Tensorflow lite. We used Numpy for data structures in the training and testing process. This made a lot of the math easier in our program. It is also a dependency for Tensorflow. We used "rafaelpadilla/Object-Detection-Metrics" on github for easy metric computation when testing our model.

We utilized several software environments while working on this project. We used Google Colab as the platform for all of our python code execution. This allowed us to train our model using external hardware. This allowed for decreased training times and for us to be able to multitask. It was also very easy to set up as libraries were easy to import and some we needed were even already

part of the environment. For dataset management we used roboflow. Roboflow is capable of storage, pre/post-processing, annotation, etc. This allowed us to easily assign groups of images to team members. This was necessary as we have around 500 labeled images. We used Gitlab to house our application for version control and team collaboration.

3.2 HARDWARE

Due to the changes from first to second semester, our implementation was much simpler than what we had designed. Most of the components from before were necessary to keep the E coli in ideal conditions for growth to speed up our process. Because our goal was now to identify the already existing types of macrophage, we did not have to worry about making it grow. We did not have to implement the moving platform (motorized), the heat plate, or the lens/objective set up. We were provided a digital endoscope from Oasis Scientific to replace the Google Coral Camera and objective.

Looking at our previous design, we decided to continue using the open source OpenFlexure microscope as our base. It fit within our budget and would be even cheaper to print because we did not have to worry about the heating element melting through the plastic. To print and build the microscope we followed the official instructions located at <https://openflexure.org/projects/microscope/build>. We printed it on an Ender 3 printer using PETG plastic. We used about $\frac{1}{3}$ of the spool of plastic with an estimated cost of \$8. This included the unused pi-camera optics parts.

Some of the non-printed parts were hard to find but after some experimentation we were able to find suitable replacements. Instead of M3x25 hexagon screws we used M3x25 button head screws and some glue, and instead of using Viton O-rings we used small rubber bands. The rubber bands are much more fragile than the O-rings so if someone is building the microscope for extreme use or is planning on using the motors they should find the correct O-rings. However, using the rubber bands works for this application. All the non-printed parts we used were found at Theisen's Home Farm & Auto.

We ignored the instructions for attaching the motors and putting together the optical element as we didn't need them. However, we needed to find a way to mount the endoscope in the microscope. This involved creating a new optics element, a few feet, and drilling a hole in the existing body. The new optics element was based on the pi-camera optics element but instead of having clips for the objective and lens, it just had a $\frac{1}{2}$ " hole for the endoscope and another $\frac{1}{2}$ " hole for the mounting arm. When modeling this part, we had to make sure to copy the mounting mechanism from the original. It attaches to the microscope by screwing in from the inside of the microscope.



Figure 3.2.1 - CAD model of a microscope foot

The design of the feet was not super important. The feet just needed to add the needed height for the endoscope. We used existing holes in the body to attach them to the bottom of the microscope. The need to drill a hole in the body could have been avoided with foresight. The correct sized hole could have been added to the design before printing.

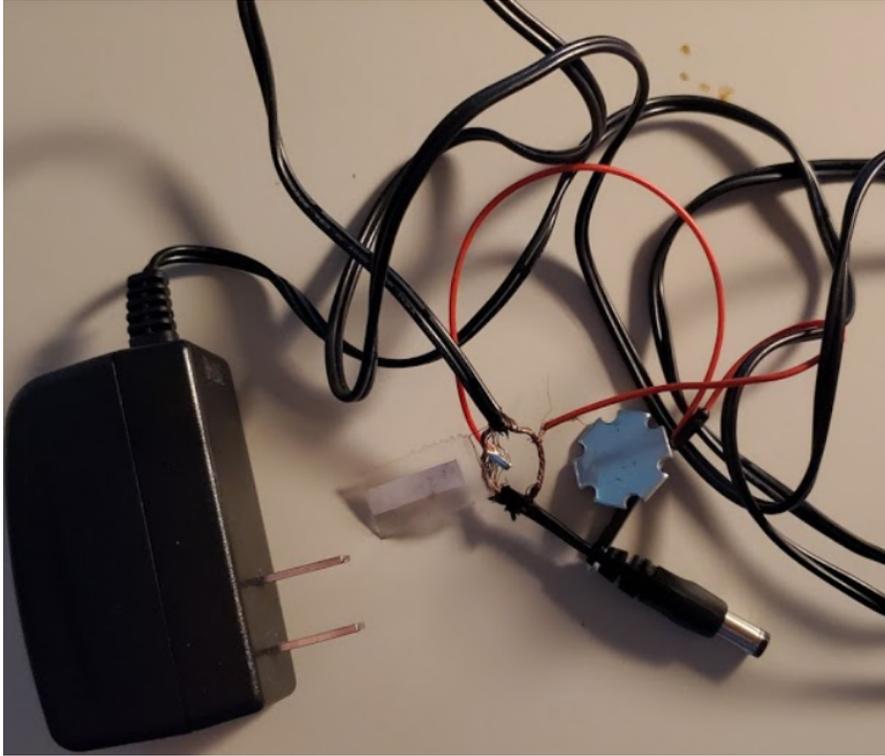


Figure 3.2.2 - CAD model of a microscope foot

The LED is powered with an old power supply with a resistor to limit the current. This was provided to us but could easily be recreated by cutting up an old charger. It is mounted with some electrical tape to the light diffuser.

Overall, we were very happy with using the openflexure microscope as the base. It was easy to build, highly modifiable, and very cheap. The endoscope was a little harder to work with as its original drivers were hard to find and do not work on modern operating systems. The device itself is not detected as anything other than something that needs power from the USB port. We worked with this by using existing screenshot tools to capture the old app's preview images (the capture image button does not work).

4 Testing Process and Results

4.1 TESTING ENVIRONMENT

The testing environment for the ML model was built in Python and utilized Google Colab. The ML model is a single shot detector neural network that was exported as a TensorFlow lite (TF lite) compatible SSD graph. We needed to use a TF lite compatible model to be compatible with the Google Coral Board. The model outputs a tensor formatted as a JSON which includes bounding box coordinates, the class of that box, and the probability it actually is that class.

To test the magnification of the microscope, we were provided with VWR micro cover glasses and a liquid from Bangs Lab containing suspended glass beads with a mean diameter of $8\mu\text{M}$. The goal was to test both the endoscope's capabilities and also the best way to provide light to create clear images.

4.2 TESTING STRATEGY

We labeled 500 images by drawing a bounding box around each cell and specifying the type of macrophage (M_0 or M_1). After labeling all the images, we split the dataset into two subsets, a training set and a testing set. There are about 350 images in our training set and 150 images in the testing set. We did this so that the ML model would not see any image we use to test it when training. After training, we used the [rafaelpadilla/Object-Detection-Metrics](#) git repository to calculate the mean average precision (mAP) of our model on the testing dataset.

Testing the microscope involved placing one drop of the liquid between two slides. We had to be very careful to not touch anything other than the sides of the glass slides to prevent fingerprints. After placing the sample on the microscope, we then had to play around with the endoscope's location and focus to get the sample in focus. Afterwards, we took sample images with the LED in different locations, one from the LED built into the endoscope (bottom up lighting) and the other from the LED at the light diffuser (top down lighting). We compared these images to determine which one matched the sample images from the lab the most.

4.3 RESULTS

After training the SSD model for 9539 epochs on google colab, it was time to test the neural network. We used tensorboard to render the statistics for the precision and recall of our model. The results are as follows:

	All	Small	Medium	Large Boxes
Mean Average Precision	0.044	0.044	0.108	0.300
Average Recall	0.119	0.118	0.183	0.300

These results are not as high as we would have liked. This means our model on average, across both classes of cells, predicts detection boxes with about 4.4% accuracy and does better as the box prediction area grows. Also, the neural network classifies each cell's type on each box, across both classes, with about 11.9% accuracy.

The results from the microscope test were less than we hoped for. The magnification of the endoscope was barely high enough to clearly see the suspended glass beads. The edges of the photos (no matter the lighting method) were blurry as a result of nearing the limit of magnification. We can use these results to determine the correct lighting conditions (top down) and how to correctly process the images before running them through the ML model (crop out edges and zoom in). Images from the test are shown below.



Figure 4.3.2 - 8 μ M glass beads with bottom up lighting

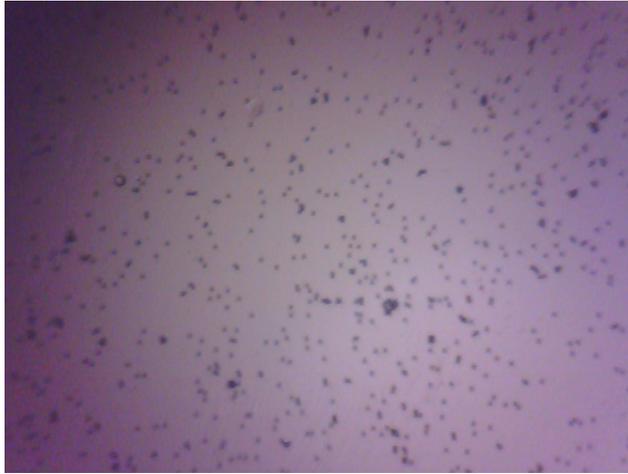


Figure 4.3.3 - $8\mu\text{M}$ glass beads with top down lighting

5 Operational Manual

5.1 INTRODUCTION (VOCABULARY)

There are a few terms we will use throughout this operational manual. It is important that the user knows and understands what each of these terms means before continuing on.

- ML - Stands for machine learning. This is what is used to identify the macrophage cells.
- Microscope - This is the physical device as a whole.
- Endoscope - This is the metal rod going through the microscope. It contains the lens and the camera for the device.
- Board - This is the Google Coral Board. It runs the ML model and hosts the user application.
- Macrophage - This is the cell the microscope is programmed to detect. It is a white blood cell that ingests anything that is not a healthy body cell.
- Device - The combination of the ML model, the Google Coral Board, and the microscope.

5.2 FEATURES

Users can use the device to observe their sample and analyze either a video stream or images for macrophage. This device is able to detect macrophage. Users are able to save the images captured from the microscope to their personal device. The user has complete control over when the sample image/video is taken through an application on their device. Processing of the sample images can run and be stopped at any time. Furthermore, users are able to configure and manage the device wirelessly through a secondary device.

5.3 FIRST TIME SETUP

There are a few steps to complete before the device can be used regularly. Please follow the steps below to set up the device. Please note that a secondary device, such as a laptop or PC, is required to control the device.

1. Plug the coral board and LED into an outlet and plug the endoscope into a usb port on your secondary device. Ensure that everything is powered on.
2. Connect the board to a wireless network. This can be done either through a wired connection. Please use the following link for reference.

- a. <https://coral.ai/docs/dev-board/get-started/#connect-internet>
3. Download the microscope application from the following link. The endoscope executable download is at the bottom of the page.
 - a. <https://www.oasisscientific.com/2omp-usb-digital-endoscope-82mm-tube-diameter.html>
4. Check access to web GUI
 - a. Once the board is powered, the board will go through the regular startup process and will start an Apache web server which will launch the website. Connect to the website by using the IP address assigned to the device in an internet browser.
 - b. If the website is unavailable, please connect to the board through a wired connection, and ensure that the Apache web server has started.
 - c. NOTE: The web server may take a couple of minutes to boot up.

As of writing this paper, the interface between the application and the microscope is not complete. Please check our website for updates as we continue to work on this project. This means that there are a couple extra steps for setting up and operating the device.

5.4 WALK THROUGH

The steps outlined below describe a typical workflow from device startup to turning off the device. Please follow the steps below to use the device correctly.

START UP:

1. Plug the coral board and LED into an outlet and plug the endoscope into a usb port on your secondary device. Ensure that everything is powered on.
2. Ensure that the board is connected to the internet and that the ReactJS website is up. Please refer to the first time setup section if the website is unavailable.

COLLECTING THE DATA:

1. Set the device on a flat surface. Ensure the endoscope is pushed all the way up. It needs to be close to the sample in order to get the proper magnification.
2. Prepare the sample slide according to the standards/ instructions of your lab. Take care not to touch the slide as any imperfections (fingerprints, dust, etc.) will show up on the data collected. A clear image is required for the ML to properly identify any macrophage.
3. Focus the endoscope on the sample. The course focus is located on the bottom of the endoscope. When using the course focus, ensure that the endoscope itself does not move.

The fine focus is the middle gear on the side of the microscope. If the sample needs to be slightly moved around, you can use the left and right gears on the microscope.

4. After you have focused the endoscope on the portion of the sample you want to analyze, run `zmp_endoscope.exe`. A couple of error windows will appear but these can be ignored.
5. Change the location the data is being saved by going to “File” -> “Set Photo Folder..”
6. Ensure the correct camera is being used by going to “Devices” and selecting the ehe microscope.
7. To collect a video, go to “Video Capture” and select “Start Capture”.
8. Collecting an image is a little harder because this software is slightly broken. You can either take a super short video with a reduced frame rate, or take a screenshot of the preview image using Snipping Tool or something similar.

DATA ANALYSIS:

1. Once the images have been collected, move the images onto the Google Coral AI board using a wired USB connection.
2. With images stored on the board, simply run a python program while specifying the location of the images.
3. The python script will save the results of the analysis in the same folder in JSON format.

As of writing this paper, the interface between the application and the microscope is not complete. Please check our website for updates as we continue to work on this project. This means that there are a couple extra steps for setting up and operating the device.

5.5 DEVICE SETTINGS

Due to the flexibility of the data capture, you have the option to use whatever works best for you to collect the data. The web application has a couple of settings/ features that the user should know about. In the settings page the user will have the ability to set the default library location. This will be where all the data for all the runs will be saved. The update model button will check for an updated model and update the device if one exists. When starting a new run, the user has the option to create a new directory for that specific run.

5.6 MAINTENANCE

There are a few things you can do to ensure that the microscope continues to run accurately. Make sure that the area below the sample is clean and that the endoscope lens is clear. If dust settles on the lens the image quality will degrade. The image quality will also degrade if the slides used to hold the sample are dirty. Ensure that all glass slides are cleaned properly after use.

The Google Coral Board will disconnect from the wifi if the password is changed. Check regularly (before use) to ensure that the device is still connected. If it disconnects, follow the instructions linked above in section 5.3. As writing this, we are continuing to work on the software running on the Google Coral Board. Occasionally checkout our website for updated software and setup instructions.

5.7 HAZARDS

Our project, when handled with care, is not dangerous. However, please read through the following safety warnings carefully. We are not responsible for misuse or injuries that occur from the mishandling of the project.

- Glass slides are sharp. Handle with care.
- The Google Coral Board gets hot when on. Do not touch the board while the microscope is in operation. Whether or not the fan is on, the board can become very hot and may cause burns.
- The body of the microscope is very thin. If it breaks, take care when disposing of the pieces as they can be sharp.
- Ensure the cells being observed with the device are handled and disposed of properly according to the standards required in your lab environment.

6 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:

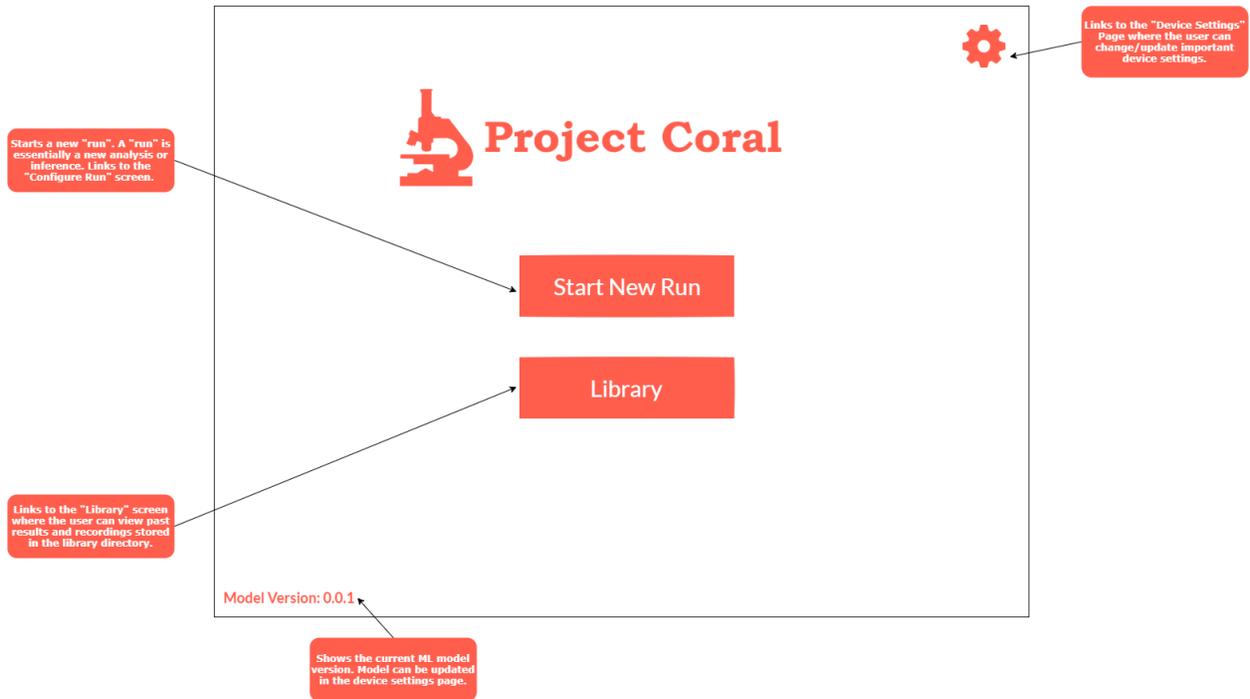
https://www.cdc.gov/healthywater/drinking/world-water-day.html?CDC_AA_refVal=https%3A%2F%2Fwww.cdc.gov%2Fhealthywater%2Fobservances%2Fwwd.html

[www.cdc.gov%2Fhealthywater%2Fobservances%2Fwwd.html](https://www.cdc.gov/healthywater/drinking/world-water-day.html?CDC_AA_refVal=https%3A%2F%2Fwww.cdc.gov%2Fhealthywater%2Fobservances%2Fwwd.html)

7 Appendices

Appendix 1. Screen Sketches.

Home



Configure Run

