Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

AI low-cost Camera for counting and classification of microbes in nature water

Team Members: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey

Product Owner: Dr. Antao Chen, Beckman Coulter

Instructor: Masoud Sadjadi

Abstract

This project's aim was to develop an AI counting and classification model on a low-cost, portable chip with an integrated camera. Using the ESP32-CAM development board, we were able to gather an image dataset from algae at Florida International University's (FIU) Modesto Maidique Campus (MMC), build a convolutional neural network (CNN) model, and run inferencing directly from the chip.

Table of Contents

INTRODUCTION

Toxic algae blooms impact Florida's ecosystem and can be triggered by fertilizer runoff, particularly from Lake Okeechobee, as well as sewer drainage and from our agriculture industry. Hurricanes and heavy storms can also contribute to nutrient seepage into the ocean, leading to further algae blooms (1).

Traditionally, monitoring and classifying waterborne microbes and pathogens have required expensive instruments and lengthy testing processes. However, with the widespread adoption of face recognition AI technology for public control and national security applications, Chinese-made cameras with AI face recognition capabilities have become affordable and ubiquitous (2). Cameras integrated with microcontrollers capable of face recognition, like the ESP32CAM, are now available for less than $10. This project aims to leverage these cameras to create a device and develop code for counting and classifying microbes and harmful bacteria in open water, thereby contributing to environmental protection.

Current System

To view high-resolution algae slides, a light microscope with a total magnification of 100-400x is employed. However, traditional light microscopes are heavy, bulky, and lack portability. Digital field microscopes, which offer portability, can be prohibitively expensive, costing around $1800 (3). On the other hand, low-cost light microscopes have limited capabilities, such as poor resolution and focus. Furthermore, quantification and classification processes are manual, making them lengthy and tedious.

Purpose of New System

The ESP32-CAM development board with an OV2640 camera supports WiFi (4). Dr. Chen provided a Nikon 100-400x resolution light microscope and a 3D printed mount for the ESP32-CAM. We gathered 2,197 images and retrained a pretrained multi-layer convolutional neural network (CNN) MobileNetV1 for 4 algae classes and 1 non-algae class. Using Edge

Impulse and Espressif libraries, we flashed the model onto the chip for inferencing. Due to the board's limited internal 1.3 MB storage and 327KB RAM, our program had to be lightweight. The achieved 59.8% accuracy serves as a promising proof of concept.
User Stories

The following section provides the detailed user stories that were implemented in this iteration of the …. project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- User story 1:
  - As a developer, I want to buy a camera that can be used to capture images or live slides and processed through python preferably.
- User story 2:
  - As a developer, I want to familiarize myself with Python/Arduino Libraries related to AI/ML.
- User story 3:
  - As a developer, I want to do hardware research to determine how our camera can integrate with our development environment.
- User story 4:
  - As a developer, I need to provide the dimensions of the camera so the PO can 3D print a tube mount for the microscope.
- User story 5:
  - As a developer, I want to build a neural network model on Keras using preliminary Kaggle data.
- User story 6:
  - As a developer, I want to practice exporting a neural network onto the ESP32 chip.
- User story 7:
  - As a developer, I need to practice using Keras using google colab to prepare building our neural network.
- User story 8:

- As a developer, I want to find/create an algae dataset that contains 3-4 general algae. The images should be through a 100-400x resolution microscope.
- User story 9:
    - As a developer, I want to label our algae dataset. We want to have a minimum of 3 algae labels, including a non-algae label.
- User story 10:
    - As a developer, I want to generate additional images for our dataset using image transformation techniques.
- User story 11:
    - As a developer, I want the ESP32 UI to show what % algae class is in the field of view.
- User story 12:
    - As a developer, I want to export the model onto our chip.

Pending User Stories

- User story 13:
    - As a developer, I want to achieve 90% accuracy on our test data.
- User story 14:
    - As a developer, we want to integrate object detection to our model so that we can discriminate different algae from a single field.
- User story 15:
    - As a developer, I want to enumerate the algae in the field of view.

Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware

- Nikon SKT Optical Microscope with 100-400x resolution lenses: provided by the PO
- Custom 3D-printed mount for ESP32-CAM: provided by the PO
- Aideepen ESP32-CAM board (https://www.amazon.com/dp/B08P2578LV)
- USB-A to Micro-USB cable: to connect the ESP32-CAM to our laptops

Software

- Trello: where we managed our user stories
- Microsoft Excel: where we developed a Gantt chart
- Arduino IDE 2.1.1 (www.Arduino.cc): where our development was done
- Expressif Systems ESP32 library package (https://github.com/espressif/arduino-esp32): how we connected the ESP32-CAM board to the Arduino IDE
- Edge Impulse Library (www.EdgeImpulse.com): where we built our final inference model
- Python 3.10.6: The programming we used to build our initial CNN model
- Tensorflow 2.12.0: the Python package we used to build our initial CNN model
- Google Colab: the collaborative environment we worked on our python code
- Kaggle: where we found our initial labeling data
- Github: where we shared edge impulse zip files and arduino code
- C++: language used to write in Arduino IDE

Sprints Plan

Sprint 1

During Sprint 1, we:

- Searched for an algae image dataset for classification purposes.
- Explored available cameras suitable for our hardware requirements.
- Investigated camera setup procedures.
- Obtained camera dimensions for 3D printing the camera mount, which we provided to our product owner (PO).
- Studied methods for building AI/ML models.
- Identified suitable library packages and programming languages for our project.

These following user stories were completed during this sprint:

- User story 1:
  - As a developer, I want to buy a camera that can be used to capture images or live slides and processed through python preferably.
- User story 2:
  - As a developer, I want to familiarize myself with Python/Arduino Libraries related to AI/ML.
- User story 3:
  - As a developer, I want to do hardware research to determine how our camera can integrate with our development environment.
- User story 4:
  - As a developer, I need to provide the dimensions of the camera so the PO can 3D print a tube mount for the microscope.

Sprint 2

During Sprint 2, we:

- Continued searching for an algae image dataset for classification.
- Attempted to collaborate with FIU marine biology researchers for dataset assistance, but faced scheduling conflicts. As an alternative, we captured over two thousand photos using live samples from FIU MMC ponds with the ESP32-CAM.
- Practiced using Keras on Google Colab to prepare for building our neural network.
- Utilized a Kaggle dataset containing 10 classes of algae to supplement our own dataset.
- Purchased the ESP32-CAM.

These following user stories were completed during this sprint:

- User story 5:
    - As a developer, I want to build a neural network model on Keras using preliminary Kaggle data.
- User story 6:
    - As a developer, I want to practice exporting a neural network onto the ESP32 chip.
- User story 7:
    - As a developer, I need to practice using Keras using google colab to prepare building our neural network.

Sprint 3

During Sprint 3, we:

- Built our algae dataset featuring 4 algae classes and 1 non-algae class.
- Started building a Tensorflow Keras CNN model using our algae dataset
- Used Keras's image generator function to generate additional images for our dataset
- Preprocessed our images using Keras

These following user stories were completed during this sprint:

- User story 8:
  - As a developer, I want to find/create an algae dataset that contains 3-4 general algae. The images should be through a 100-400x resolution microscope.
- User story 9:
  - As a developer, I want to label our algae dataset. We want to have a minimum of 3 algae labels, including a non-algae label.

Sprint 4

During Sprint 4, we:

- Attempted to export our Keras model to the ESP32 chip but encountered issues due to its size (over 50 MB).
- Explored various libraries like Tensorflow Lite, eloquentTinyML, tinymlgen, and everywhereml for exporting the CNN to the ESP32-CAM chipset. However, these packages had limited functionality or did not support CNN output, resulting in models that exceeded the onboard space on the chip. We also tried alternative classification methods but lacked familiarity with the techniques we found.
- Found Edge Impulse (6) as a potential solution to build and export our model to the ESP32-CAM board. We began constructing our model on the Edge Impulse server.
- Continued to expand our dataset by generating more images.

These following user stories were completed during this sprint:

- User story 10:
  - As a developer, I want to generate additional images for our dataset using image transformation techniques.

Sprint 5

During Sprint 5, we:

- Created a GitHub repository to share our Arduino code and zip files.
- Developed a classification model using Edge Impulse.
- Troubleshot loading the Edge Impulse library and our model into the Arduino IDE.
- Attempted to integrate the ESP32 Camera Web Server project with the Edge Impulse classification project in Arduino.
- Ordered a new ESP32-CAM board to address unresolved boot load issues with the initial camera.
- Successfully implemented inferencing on the ESP32-CAM but encountered issues with supporting a live camera feed. We suspect the problem was related to improper buffering of the camera feed during inferencing.

These following user stories were completed during this sprint:

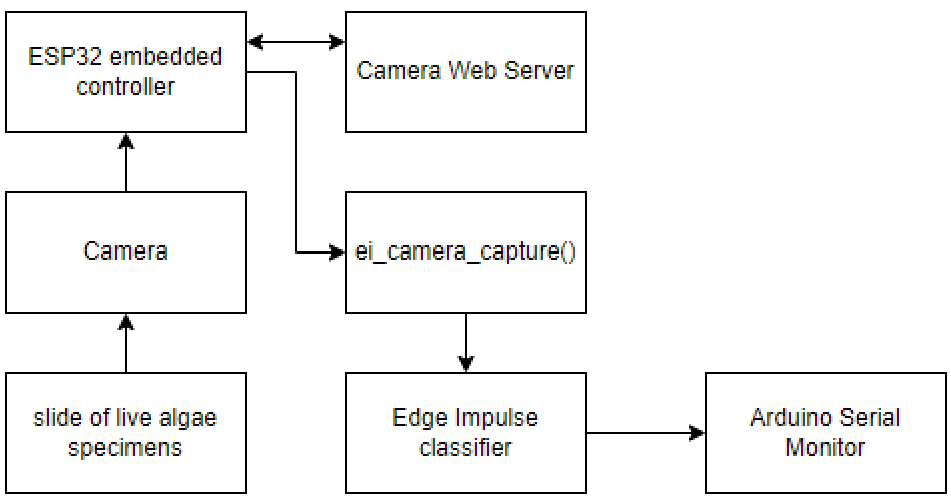
- User story 11:
  - As a developer, I want the ESP32 UI to show what % algae class is in the field of view.
- User story 12:
  - As a developer, I want to export the model onto our chip.

System Design

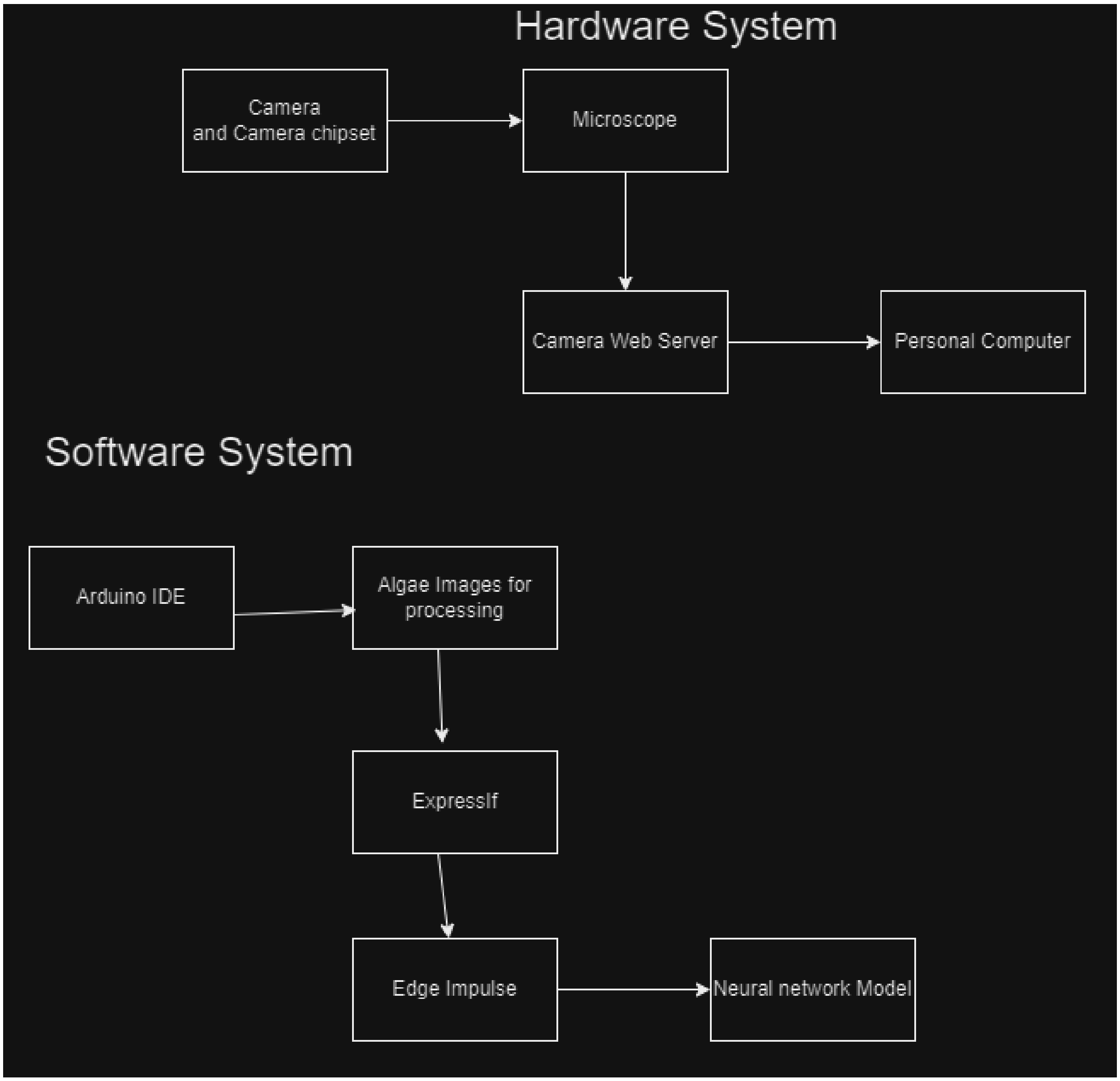
This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

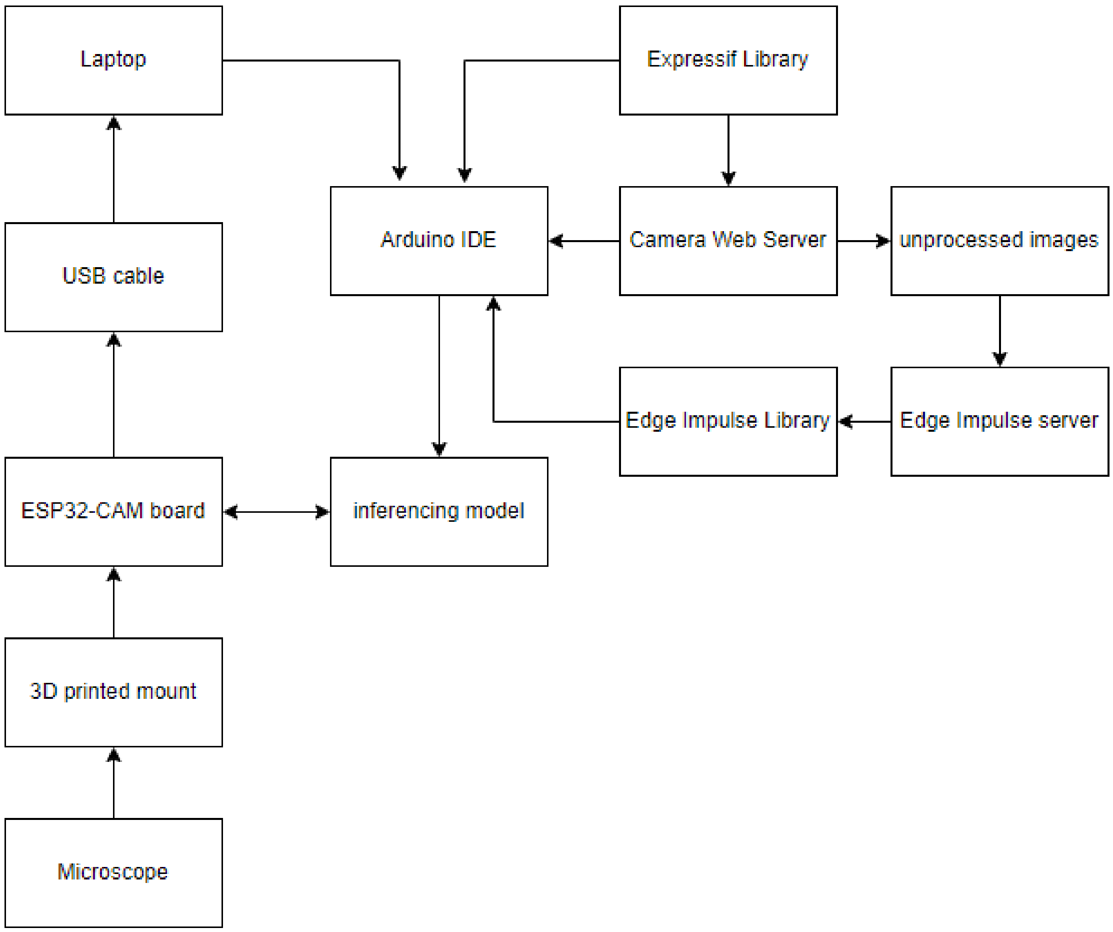
The architectural pattern we chose to use was the “Command and Control” pattern. Our ESP32 embedded controller serves as the command center or central controller of the system. It interacts with the camera sensor, captures the data, and also runs the classification algorithm.



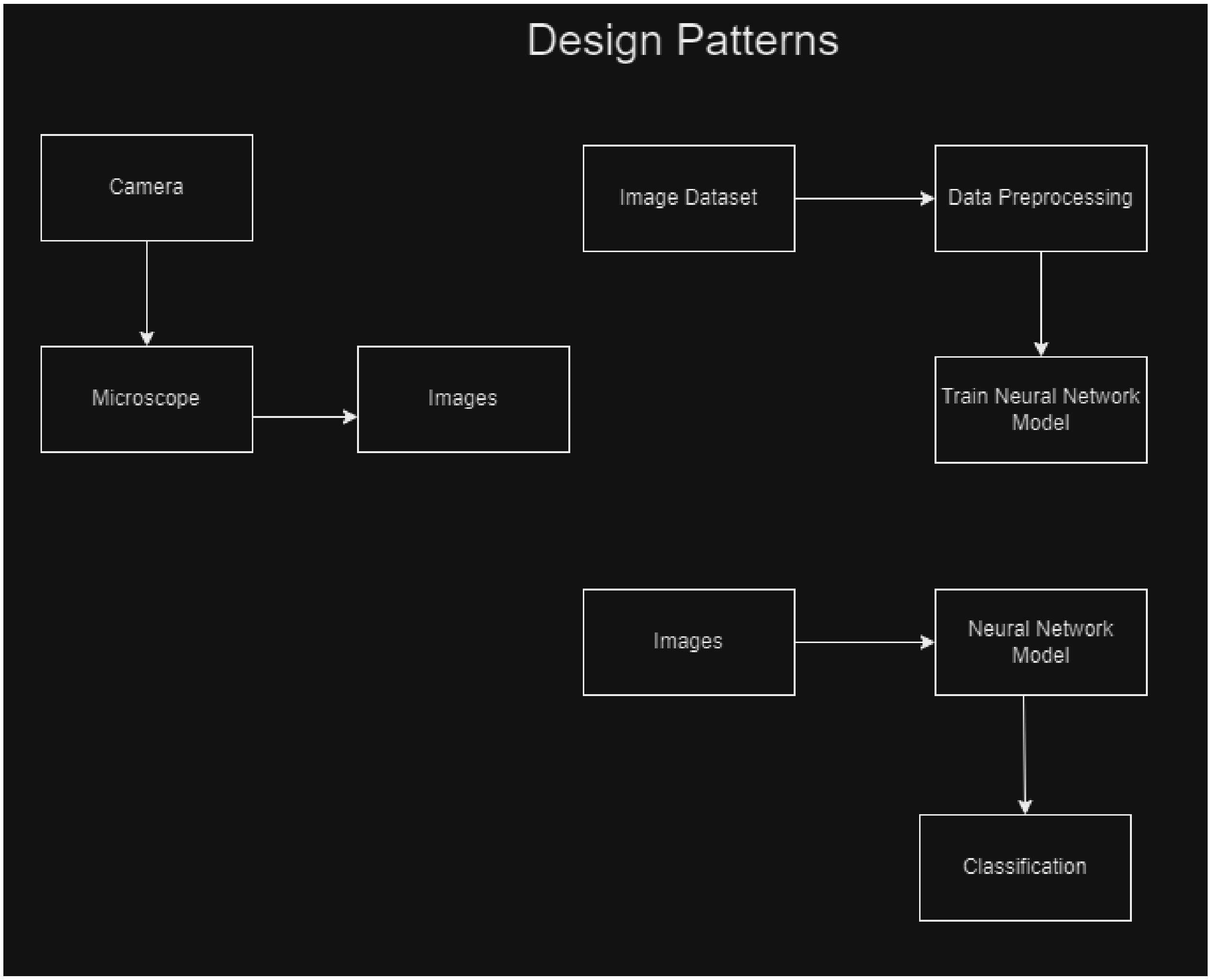
System and Subsystem Decomposition



Deployment Diagram



Design Patterns



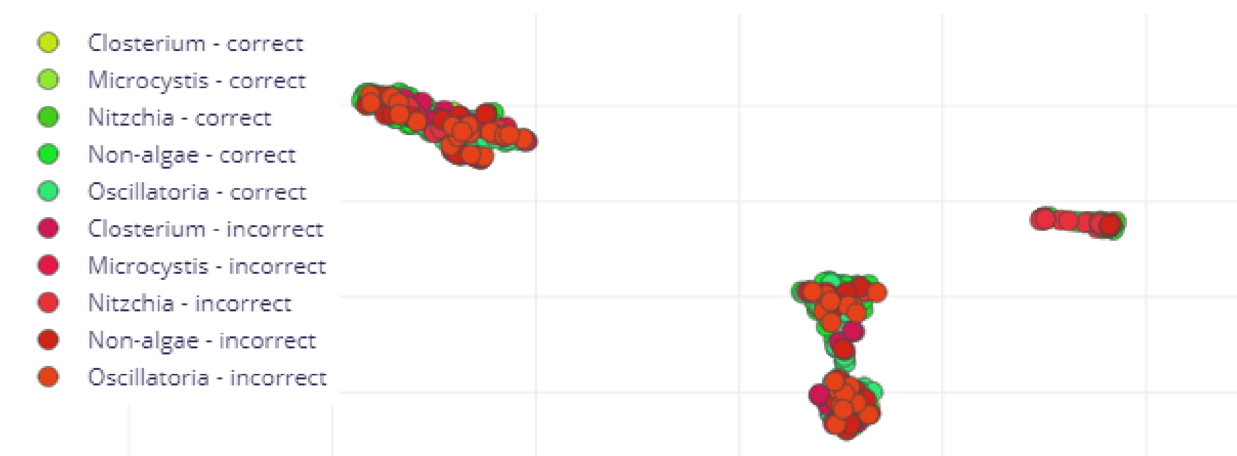
System Validation

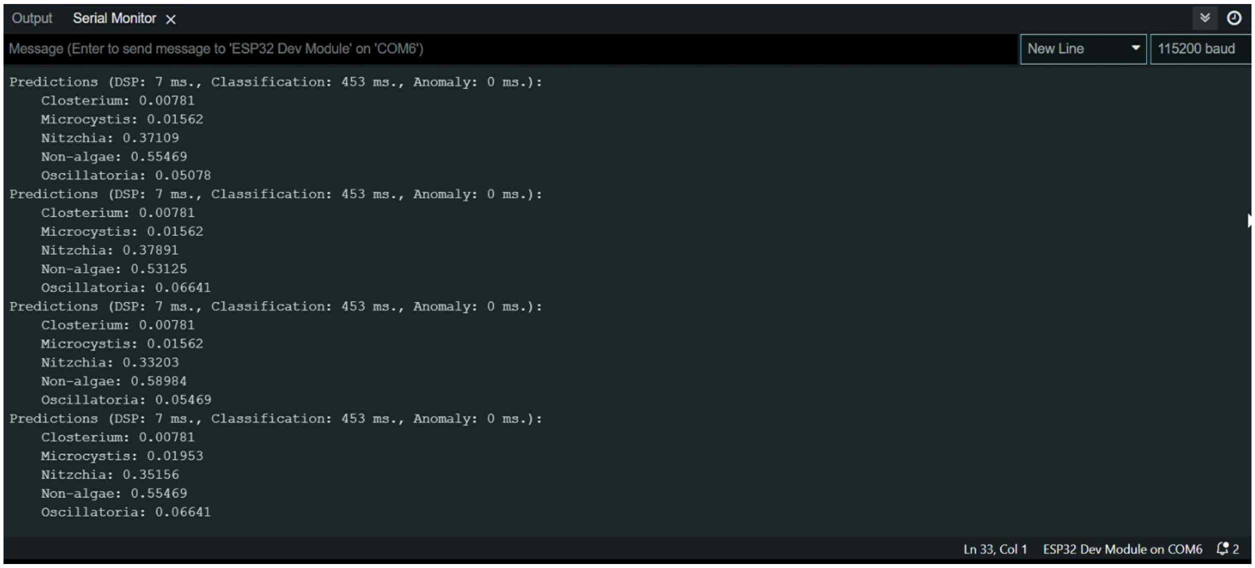
Model testing results

ACCURACY
59.82%

	CLOSTERIU	MICROCYSTI	NITZCHIA	NON-ALGAE	OSCILLATOR	UNCERTAIN
CLOSTERIUM	12.5%	0%	0%	5%	15%	67.5%
MICROCYSTIS	0%	11.1%	0%	44.4%	3.7%	40.7%
NITZCHIA	0%	0%	9.8%	31.7%	9.8%	48.8%
NON-ALGAE	1.0%	0%	0.5%	78.5%	2.1%	17.8%
OSCILLATORI	0%	0%	0%	6.7%	72.4%	20.9%
F1 SCORE	0.21	0.20	0.17	0.80	0.79	

Feature explorer ⓘ




Output
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')
New Line
115200 baud
Predictions (DSP: 7 ms., Classification: 453 ms., Anomaly: 0 ms.):
Closterium: 0.00781
Microcystis: 0.01562
Nitzchia: 0.37109
Non-algae: 0.55469
Oscillatoria: 0.05078
Predictions (DSP: 7 ms., Classification: 453 ms., Anomaly: 0 ms.):
Closterium: 0.00781
Microcystis: 0.01562
Nitzchia: 0.37891
Non-algae: 0.53125
Oscillatoria: 0.06641
Predictions (DSP: 7 ms., Classification: 453 ms., Anomaly: 0 ms.):
Closterium: 0.00781
Microcystis: 0.01562
Nitzchia: 0.33203
Non-algae: 0.58984
Oscillatoria: 0.05469
Predictions (DSP: 7 ms., Classification: 453 ms., Anomaly: 0 ms.):
Closterium: 0.00781
Microcystis: 0.01953
Nitzchia: 0.35156
Non-algae: 0.55469
Oscillatoria: 0.06641
Ln 33, Col 1
ESP32 Dev Module on COM6
2

Glossary

1. AI (Artificial Intelligence): The simulation of human intelligence in machines that can perform tasks typically requiring human intelligence, such as problem-solving, decision-making, and learning.

2. Arduino: An open-source electronics platform based on easy-to-use hardware and software, often used for creating interactive projects and prototypes.

3. CNN (Convolutional Neural Network): A type of deep learning neural network specifically designed for image recognition and processing tasks.

4. Closterium: A genus of green algae that has a distinctive elongated cell shape.

5. Edge Impulse: A platform for building, deploying, and managing machine learning models on embedded systems and IoT devices.

6. ESP32: A series of low-cost, low-power system-on-chip microcontrollers with integrated Wi-Fi and Bluetooth capabilities.

7. Expressif: The manufacturer of the ESP32 microcontroller.

8. Inferencing: The process of using a trained machine learning model to make predictions or classifications on new data.

9. Keras: An open-source high-level neural networks API written in Python, often used with TensorFlow for building and training ML models.

10. Kaggle: A popular platform for hosting machine learning competitions, datasets, and notebooks for data analysis and modeling.

11. Microcystis: A genus of cyanobacteria known for forming harmful algal blooms in bodies of water.

12. ML (Machine Learning): A subset of AI that involves training computer systems to learn from data and make decisions or predictions without being explicitly programmed.

13. MobileNetV1: A lightweight convolutional neural network architecture commonly used for mobile and embedded systems.

14. Nitzchia: A genus of diatoms, a type of algae found in both freshwater and marine environments.

15. Oscillatoria: A genus of filamentous cyanobacteria that forms blue-green algae blooms.

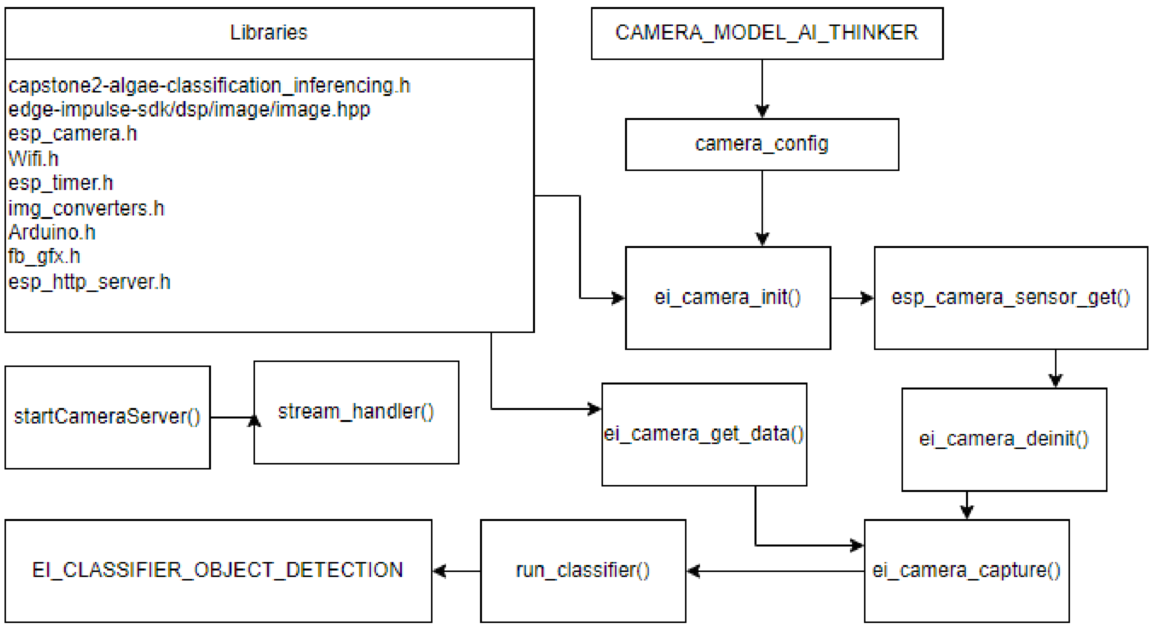
16. Python: A popular high-level programming language widely used for data analysis, machine learning, and web development.

17. Serial Monitor: A tool in Arduino IDE that allows communication with an Arduino board and shows messages sent from the board during program execution.

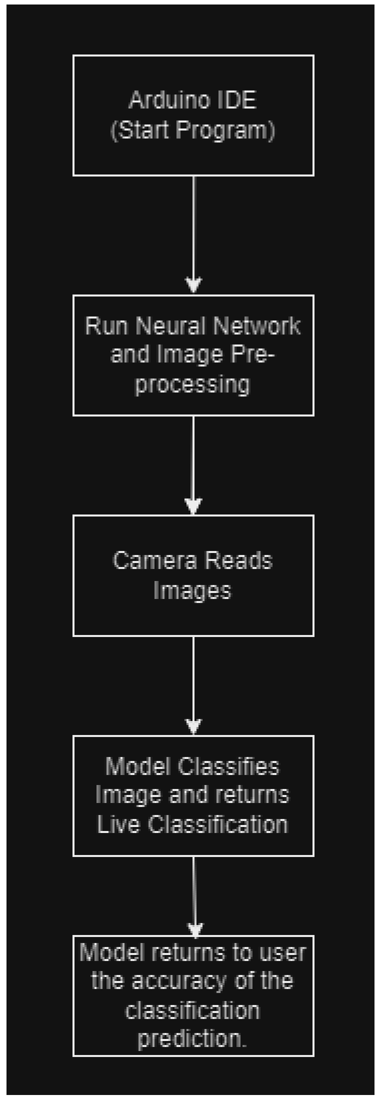
18. Tensorflow: An open-source machine learning framework developed by Google for building and training ML models.

APPENDIX

Appendix A - UML Diagram



Appendix B - User Interface Design



Appendix C - Sprint Review Reports

Sprint 1:

Sprint Review Meeting Minutes

Attendees: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey, Dr. Antao Chen
Start time: 12:00 PM
End time: 1:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

1. As a developer, I want to buy a camera that can be used to capture images or live slides and processed through python preferably.
2. As a developer, I want to familiarize myself with Python/Arduino Libraries related to AI/ML.
3. As a developer, I want to do hardware research to determine how our camera can integrate with our development environment.
4. As a developer, I need to provide the dimensions of the camera so the PO can 3D print a tube mount for the microscope.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

1. As a developer, I want to contact FIU/UM marine bio researchers who can provide me with live images of algae to build a dataset with. 4
2. as a developer, I want to find/create an algae dataset that contains 3-4 general algae. The images should be through a 100-400x resolution microscope. 5

Sprint 2:

Sprint Review Meeting Minutes

Attendees: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey, Dr. Antao Chen
Start time: 4:00 PM
End time: 4:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

1. As a developer, I want to build a neural network model on Keras using preliminary Kaggle data.
2. As a developer, I want to practice exporting a neural network onto the ESP32 chip.
3. As a developer, I need to practice using Keras using google colab to prepare building our neural network.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. As a developer, I want to find/create an algae dataset that contains 3-4 general algae. The images should be through a 100-400x resolution microscope. 5

Sprint 3:

Sprint Review Meeting Minutes

Attendees: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey, Dr. Antao Chen
Start time: 4:00 PM
End time: 4:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

1. As a developer, I want to find/create an algae dataset that contains 3-4 general algae. The images should be through a 100-400x resolution microscope.
2. As a developer, I want to build a neural network model on Keras using preliminary Kaggle data
3. As a developer, I need to practice using Keras using google colab to prepare building our neural network.
4. As a developer, I want to label our algae dataset. We want to have a minimum of 3 algae labels, including a non-algae label.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. As a developer, I want to practice exporting a neural network onto the ESP32 chip 5
2. As a developer, I want to generate additional images for our dataset using image transformation techniques 5
3. As a developer, we want to integrate object detection to our model so that we can discriminate different algae from a single field 5
4. As a developer, I want to achieve 90% accuracy on our test data 5

Sprint 4:

Sprint Review Meeting Minutes

Attendees: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey, Dr. Antao Chen
Start time: 4:00 PM
End time: 4:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

1. As a developer, I want to practice exporting a neural network onto the ESP32 chip 4

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. As a developer, I want to achieve 90% accuracy on our test data 5
2. As a developer, I want the ESP32 UI to show what % algae class is in the field of view. 4
3. As a developer, we want to integrate object detection to our model so that we can discriminate different algae from a single field 5

Sprint 5:

Sprint Review Meeting Minutes

Attendees: Artem Andrianov, Mojeed Oladele Ashaleye, Amanda Beatriz Chacin-Livinalli, Noah C Cuevas, Max Samuel Karey, Dr. Antao Chen
Start time: 5:30 PM
End time: 6:00 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

1. As a developer, I want to export the model onto our chip 5
2. As a developer, I want the ESP32 UI to show what % algae class is in the field of view. 4

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

1. As a developer, I want to achieve 90% accuracy on our test data 5
2. As a developer, we want to integrate object detection to our model so that we can discriminate different algae from a single field 5
3. As a developer, I want to enumerate the algae in the field of view 5

Appendix D

User Manuals

To make use of this project and other products:

An Arduino IDE in which the user guide is provided here per the current version:

https://docs.arduino.cc/software/ide-v2

The ESP32-CAM is also something needed for this project. The manual/guide can be seen here:

https://myosuploads3.banggood.com/products/20210514/20210514020822ESP32CAM.pdf

A laptop or desktop with a USB port. The computer that we used during our tests were a windows computer with a 64-bit operating system, 256GB storage, 12GB of memory, and an Intel® CoreTM i5- 10210U CPU at 1.60GHz.

A mount of some kind to mount the camera on top of the light microscope. The one that was provided to us was a 3d modeled and printed mount from our product owner Dr. Chen

For the porting the edge impulse website was used.

Installation/Maintenance Document

1. The installation of our design begins with having the ESP-32 Camera mounted onto a light microscope and have it be connected to the laptop or desktop.
2. Once you have everything in its a proper connection you will open the development environment which would be the Arduino IDE. You can grab the IDE from https://www.arduino.cc/en/software if it is not already installed.
3. Go to file, preferences, and add the additional board manager url https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
4. Press ctrl+shift+b to open the board manager, search for ESP32 and install the latest package (2.0.11 at this writing).
5. Go to tools, board, esp32, and find the board labeled AI Thinker ESP-32 CAM.
6. Once the board is connected you will then navigate to the file tab on the top left of the screen.
7. Once selected you will go to examples, ESP32, Camera, CamerWebServer. Once selected be sure to use the AI Thinker model and not the ESP 32 Eye cam model. You will then need to fill out the required credentials which will be the internet connection name under ssid and password under the password
8. After entering the credentials you will upload and compile the code for the webserver. This process takes about two to five minutes but it may vary.
9. You will be provided a link in the serial monitor tab. Copy and paste this link into and search engine and you will have access to the webserver and see the user interface and edit parameters if necessary.
10. After you have established a connection to the webserver you will then build and deploy the model using the website Edge Impulse. This was done by uploading the pictures of the data to edge impulse and it building the model accordingly.
11. Once the model is built you will export the model by clicking the Sketch tab on the top left. You will then click on Add.ZIP Library which will open a file export tab. You will see a folder name libraries. Double click on the folder and the model that needs to be uploaded will be seen in there.
12. Once uploaded you can check the imported library by goingto file, example, and at the bottom the imported model will. You will then select the model, ESP32, and the esp32_camera
13. Once again make sure that you are connected to the correct board and port by going to the board manager and select esp32 dev module

14. Once everything is done and you are on the correct model of camera you will compile the code in the Arduino IDE. Once compiled you will see the how accurate the model is when it comes to classifying the different types of algae in a terminal on the bottom of the IDE

Shortcomings/Wishlist Document

During the project, we faced several challenges and limitations:

- Counting microbes in samples was not implemented. There were multiple genera of algae in some slide images. We ended up choosing to relabel the photos a few times.
- Enumeration of algae when the microscopic field panned was not achieved.
- Object detection was not implemented; the entire microscopic field (320 x 240 pixels) was processed, incorporating empty space around the algae in the inferencing model.
- The accuracy of our best model was only 60%, falling short of the target 90% accuracy.
- Time and distance constraints hindered effective utilization of algae researchers, leading to self-labeling of the dataset.
- Insufficient samples for certain algae classes (less than 100 photos) required splitting data for training and testing.
- Implementing the Camera Web Server alongside the inference model was unsuccessful, despite attempts to troubleshoot the Arduino code.
- Consideration of an LCD screen for enhanced portability was suggested for future development.
- Latency issues when running the Camera Web Server impacted using a higher resolution.

References

1. https://www.floridahealth.gov/environmental-health/aquatic-toxins/harmful-algae-blooms/index.html
2. https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/
3. https://aquaticsensors.com/product/digital-field-microscope/
4. https://www.amazon.com/Aideepen-ESP32-CAM-Bluetooth-ESP32-CAM-MB-Arduino/dp/B08P2578LV/ref=sr_1_3_pp
5. https://www.kaggle.com/datasets/mengyuy/the-algae-cell-images
6. www.edgeimpulse.com