UNIVERSITY of VIRGINIA

SCHOOL of ENGINEERING & APPLIED SCIENCE

Final Report

EGGS BY THE DOZEN

March 31, 2024

Section 001 - Group 9: James Baker, Aiden Ha, Dongwha Shin, Tommy Williams, and Eddie Xiao

Susan Skalak

Contents

Executive Summary **1**

1. Introduction **2**

1.1 Client Problem 2
1.2 Objective and Approach 2
1.3 Report Organization 2

2. Problem Definition and Background **3**

2.1 Client Interview Summary 3
2.2 Literature Review 5
2.3 Problem Definition 7

3. Initial Solution Ideation **9**

3.1 Development of Alternative Designs 9
3.2 Selection of Initial Design 10
3.3 Detailed description of design 11

4. Testing Results, Analysis and Design Revisions **15**

4.1 Initial fabrication and testing plan 15
4.2 Solution Development, Testing, and Revisions 19
4.2.1 Software Development 19
4.2.2 Summary of Algorithm Performance Testing 23
4.3 Informed Design Revision 26

5. Summary of final solution **28**

5.1 Description of final design 28
5.2 Fabrication workflow and schedule 29
5.3 Reflection of Final Solution fulfilling Customer Needs 31

6. Recommendations and Future Work **32**

6.1 Future Work for Eggs by the Dozen 32
6.2 Recommendation for Future Work 32

7. References **33**

List of Figures

Figure 1: *Fecal egg count sample taken under a microscope*
Figure 2: *Famacha Anemia Color Guide*
Figure 3: *Convolutional Neural Network General Process*
Figure 4: *General Feature-Based Image Analysis Algorithm Process*
Figure 5: *Workflow Chart*
Figure 6: *Low Fidelity Eclipse Identification*
Figure 7: *Output Error analysis*
Figure 8: *Second Test Sample Generated Result Metric Page*
Figure 9: *Equation for Total Error*
Figure 10: *Binary Threshold Stage Example on Medium Sample Image*
Figure 11: *Simplified Software Flow*
Figure 12: *Ellipse algorithm within EllipseDetector Module*
Figure 13: *FitEllipse_Polygon function within EllipseDetector*
Figure 14: *FitEllipse_RANSAC function within EllipseDetector*
Figure 15: *DetectParasites Method in detect.py*
Figure 16: *Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples with Constant Threshold Value of 210*
Figure 17: *Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples with a Threshold Value Calculated for Each Image Using Otsu*
Figure 18: *Error Analysis Graph*
Figure 19: *Final GetBinaryThreshold method in Threshold*
Figure 20: *Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples with Modified Otsu*
Figure 21: *Overall Flow Chart of Software Development*
Figure 22: *Updated Workflow Diagram*
Figure 23: *Updated Gantt Chart*

List of Tables

Table 1: *Comparison between McMaster/Mini-FLOTAC tests and an automated fecal egg count algorithm*
Table 2: *Comparison on Proposed Designs with Strengths and Weaknesses*
Table 3: *Bills of Materials*
Table 4: *Workflow*
Table 5: Gantt Chart
Table 6: Workflow and Gantt Chart
Table 7: *Classification of Health Condition Based on Egg Count*

List of Acronyms

CNN – Convolutional Neural Network

OpenCV – Open Computer Vision Library

ICIP - International Conference on Image Processing

UI – User Interface

MVC – Model View Controller, Common Software Architecture

API – Application Programming Interface

YOLO – You Only Look Once

EXECUTIVE SUMMARY

Client Susan Skalak faces the challenge of efficiently monitoring parasites in sheep, particularly barber pole worms, to ensure animal health. Current methods involve manual counting of parasite eggs using the Modified McMaster Test, which costs $32 per fecal sample per animal, making regular testing costly. To address this issue, three potential solutions were explored: a convolutional neural network (CNN), a feature-based computer vision algorithm, and a chitin dye design.

While the CNN appeared accurate, its reliance on a large dataset and image variability posed challenges. The chitin dye design wasn't pursued due to the lack of knowledge. Although the feature-based algorithm was less accurate than a well-functioning CNN, it was chosen due to being more practical given the available resources and time constraints.

Implementing the initial design required developing the website, the server, and the Computer Vision algorithm. The website was developed using HTML/CSS/JavaScript; the server was developed using Express.js; the Computer Vision algorithm was developed using the OpenCV library in Python. The application was containerized using Docker and deployed using Adaptable.

Testing the initial design involved checking whether the application can take valid images from the client, send it to the algorithm, receive data from the algorithm, and post the results back to the client. During this process, an error involving image file size limit was found, which was caught by implementing an error-catching middleware for Multer in Express.js and setting an explicit 1 MB file size limit when declaring an instance of the Multer middleware. In addition to testing the server, three tests were used to test the backend algorithm's performance. These tests tested for parasite count discrepancy, correct health condition identification, and location discrepancies of parasites in detected images. The software was improved based on the results of testing and a final performance assessment indicated that the algorithm met the success criteria of the client by correctly identifying the health condition of all samples tested.

Future improvements for the project include acquiring more data to enhance algorithm performance, expanding the scope to detect additional parasite egg types, and developing a mobile app for improved user accessibility and faster analysis of parasite images.

1. INTRODUCTION

1.1 Client Problem

Susan Skalak, the client at Ringadal Farm, needs an efficient way to monitor parasites in her sheep, particularly barber pole worms. Goats and other livestock often contract intestinal parasites, and identifying their eggs in fecal samples is crucial for animal health. Currently, vets use a McMaster slide, and they sit and count all the eggs under the microscope manually. The client must pay $30 per fecal sample per animal. Ideally, clients would conduct several tests a year to check for parasites. However, when tallying up the costs of each test, it becomes very expensive.

1.2 Objective and Approach

The project's objective was to create a reliable, accurate and cheap method to detect and count parasite eggs in fecal samples of goats and sheep, that specifically minimizes undetected, infected goats. More specifically, each test must cost less than 16 dollars, must provide results faster than two weeks, be as or more accurate than the current parasite detection method, and be able to run natively. Thus, the approach to the problem was to create a web application that would be able to take images taken from a microscope of the fecal matter and apply a feature extraction algorithm leveraging OpenCV and a novel parameter optimization algorithm to successfully quantify the correct number of parasites in the shortest time frame possible.

1.3 Report Organization

This report is organized into six different chapters, with Chapters 2-6 detailing the numerous steps that the engineering team took towards providing the best possible solution for the client: Susan Skalak.

- **Chapter 2** describes the problem definition, specifically Susan Skalak and her need for an efficient way to monitor parasites in her sheep. The chapter also describes the background of the problem, including different ways people have tried to tackle it in the past.
- **Chapter 3** describes the initial solution ideation including the alternative designs that were conceived, the reasoning behind why a feature detection algorithm was chosen opposed to YOLO or Convolutional Neural Network algorithms, and a detailed description of the design.
- **Chapter 4** describes the development of the software solution to the problem, numerous descriptions of the testing processes that were employed to ensure that the results provided were correct, and the revisions that were made towards the software solution.
- **Chapter 5** summarizes the final solution that was developed and reflects on how well the web application that was developed fulfills Susan Skalak's needs.
- **Chapter 6** describes the future work that must be developed for further improvement of the web application. Additionally, chapter 6 lists recommendations for future engineers to implement if they want to make meaningful contributions to the project.

2. PROBLEM DEFINITION AND BACKGROUND

2.1 Client Interview Summary

During an interview with the client on February 21, Susan Skalak introduced herself as a fiber farmer at Ringadal farm, where she raises animals like goats and sheep for their wool. One challenge she faces is the presence of parasites, which can affect the health of the livestock. Specifically, barber pole worms, a type of roundworm, have been known to cause haemonchosis, which leads to severe anemia and even death. To combat these parasites, the client has been utilizing fecal egg counts to determine the number of roundworm eggs per gram of feces. Using this method, she has been able to identify whether her livestock was sick, evaluate the effectiveness of wormers (chemicals that kill roundworms), and determine breeding methods that promote the birth of parasite-resistant livestock.

To ensure an accurate fecal egg count, the client currently utilizes the Modified McMaster Test, which involves a two-chamber McMaster slide and a 100x microscope to analyze pellets of feces. After extracting fresh or refrigerated feces (stored less than a week) from livestock, a flotation test is performed where fecal matter and a flotation solution are mixed, resulting in the less-dense eggs floating while the rest of the more-dense solid matter sinks. The mixture is then stirred 30 times and a sample is collected from the top to ensure only the eggs are extracted. Finally, the sample is placed on one chamber of the McMaster slide and the process is repeated so that another sample can be placed on the second chamber. Once the samples are ready to be examined by a microscope, the eggs in both chambers are counted and multiplied by 50 to calculate the eggs per gram of feces. Figure 1 shows a fecal sample with eggs when viewed under a microscope:

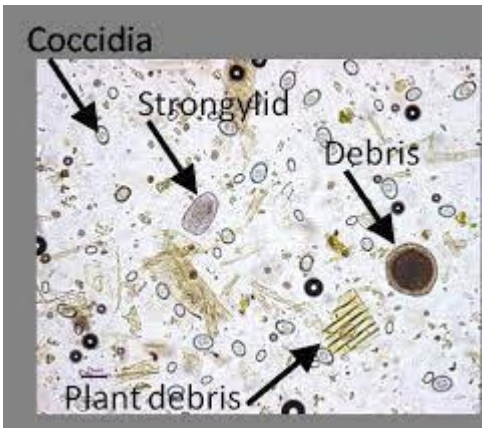


Figure 1: *Fecal egg count sample taken under a microscope*

Although quite accurate, the current McMaster Test is not ideal for the client because it is both expensive and time-consuming. Besides feces extraction, the Modified McMaster Test is mostly performed by veterinarians in the client's local laboratory, which results in up to $64 per animal for every year if wormer is included. Moreover, the lab could take up to two weeks to complete the test, which includes a two-hour drive back and forth between the client's farm and the lab. To circumvent this process and perform fecal egg count more efficiently, the client requests that the development an app that can process photos of McMaster slides with fecal solution and calculate the total fecal egg count per gram of feces. The app must not confuse parasite eggs with air bubbles, which can be differentiated by the oval shape and lighter edges of

a parasite egg compared to round and darker-edged air bubbles. Furthermore, as mentioned in Appendix A, the app must meet the following list of constraints: work with an iPhone 6, no microscope used, and perform the test at a cheaper cost and faster speeds compared to the McMaster test without sacrificing accuracy. The client also recommends that the app run without the need for internet since some fiber farmers do not have internet access.

Customer Discovery

Internal parasite infection remediation is one of the most prominent issues facing livestock farmers. In the United States, the most common parasite species observed in cattle are *Nematodirus*, *Haemonchus*, *Cooperia*, *Oesophagostomum*, and *Ostertagia* (Stromberg et al., 2015). Parasite species such as *Cooperia* and *Ostertagia* can cause subclinical issues, while *Haemonchus* is known to cause more severe issues. *Haemonchus contortus*, otherwise known as the barber's pole worm due to its distinctive curlicue pattern, is a blood-sucking nematode that elicits a unique combination of anemia and oedema characterized as haemonchosis, which can be fatal if left untreated. Additionally, *Haemonchus contortus* is known to infect the abomasal mucosa of a wide range of domesticated ruminants, such as bovine, ovine, caprine, and even cervine (Farm Health Online, n.d.).

In addressing internal parasite challenges, cattle producers resort to anthelmintic drugs. Unfortunately, the inconsistency of and ignorance to efficient anthelmintic administration practices amongst cattle farmers in previous decades has led to the rise of anthelmintic-resistant parasites in livestock (Gasbarre, 2014). In the absence of new anthelmintic classes, researchers have explored alternative strategies to curb or prevent the emergence of anthelmintic resistance in pastures. One such strategy involves maintaining refugia within the herd. According to Kaplan and Vidyashankar, this approach entails treating only those cattle with a high parasite load using an effective anthelmintic drug. By preserving refugia in the herd, the development of resistance is slowed, and a population of susceptible parasites is retained in the pastures (2012). Despite farmer utilization of refugia-maintenance techniques, each animal is still administered anthelmintic drugs an average of 1-2 times a year.

Traditionally, farmers would closely monitor their livestock's' mannerisms and physicality in order to determine if and how severely an animal is suffering due to parasite infection. A common method to estimating the health of an infected animal, especially by *Haemonchus contortus*, is to compare the color of their lower eyelid to the colors on a Famacha anemia guide [Figure 2]. This advises the farmer when to dose the animal with anthelmintics based on an approximation of red blood cells count within the conjunctiva. After which, the farmer must collect fecal samples for testing at a ruminant testing facility. At such labs, veterinarians utilize the FLOTAC and McMaster slide methods to hand count and calculate the density of parasite eggs per gram of fecal matter. Individual sample testing can be a time-intensive and costly procedure depending on one's proximity to a certified lab.

Throughout 2021, there was a significant increase in sheep numbers internationally, culminating in an all-time new high at 1.266 billion heads. At the very same time the international lamb meat market gathered substantial financial energy, likely to grow to $8.9 billion in 2023. Projections show a substantial development course with assumptions of the marketplace getting to $12.8 billion by 2030. The wool market, which was valued at $37.5 billion in 2021, is also set for expansion, with an anticipated rise to $53.4 billion by 2030, reflecting a compound annual growth rate (CAGR) of 4.9% from 2023 to 2030 (International Wool Textile Organisation, 2021). Correspondingly, the global beef market boasted a valuation of $395.22 billion in 2021 and a head count of 942.6 million in 2023 (USDA Foreign Agricultural Service,

2024). Considering the cost of anthelmintic drugs and laboratory testing as well as the size of the meat and wool production market, it stands to reason that an alternative, more accessible, method for the quantification of parasite eggs in domesticated ruminant feces would undoubtedly attract a large portion of the markets' producers.

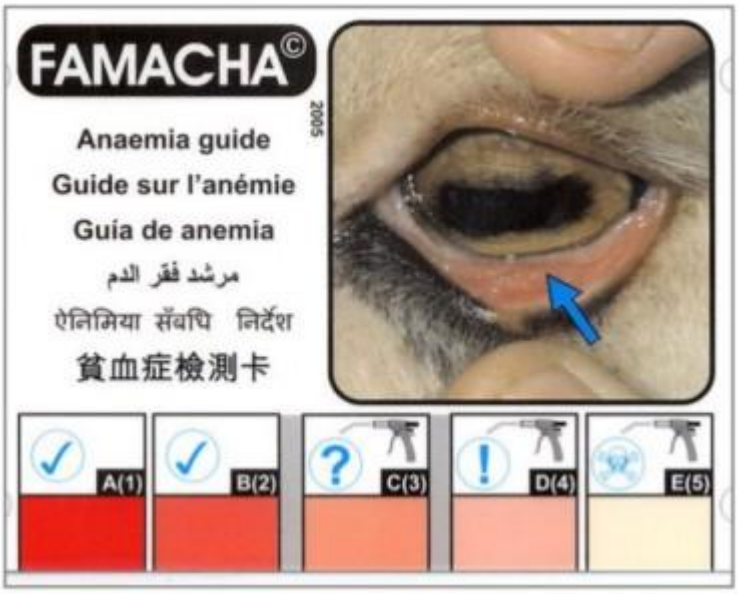


Figure 2: *Famacha Anemia Color Guide*
Note: A(1) is healthy, E(5) is fatal. Recommended dosing begins at C(3).

2.2 Literature Review

Background in Fecal Egg Counts Within Veterinary Parasitology

Intestinal parasites pose significant challenges in both veterinary and human medicine globally, especially in developing regions. Traditional identification methods through microscopic examination of feces are not only time-intensive and require specialized skill but also are not feasible in field conditions. Consequently, standardized protocols for parasite monitoring are not commonly implemented, leading to reliance on routine, mass treatments with anti-parasitic drugs, which exacerbate the issue of drug resistance. Hence, there is a pressing demand for an easy-to-use, efficient egg-counting technique that can enhance clinical practices without the drawbacks of current methods (Kaplan and Vidyashankar, 2012).

There are two predominant methods of fecal egg counting currently utilized by clinical researchers and veterinarians, namely the FLOTAC method and the McMaster Slide method. Cringoli and colleagues developed the FLOTAC technique in the early 21st century. Its development was part of an effort to improve the diagnostic accuracy for parasitic infections, particularly in veterinary medicine. The method was first introduced and described in detail in a publication around 2004 by Giuseppe Cringoli, aiming to address the limitations of existing fecal examination techniques by offering higher sensitivity and specificity for detecting a wide range of parasites (Cringoli, G., 2004).

The McMaster method, on the other hand, has a longer history, being developed in the 1930s. It was designed as a simple, quantitative technique for estimating the number of parasite eggs in fecal samples, primarily for veterinary applications. The method's simplicity and practicality for routine use have made it a longstanding staple in parasitology (Gordon and Whitlock, 1939).

Studies comparing these two approaches demonstrated that the FLOTAC method outperforms its predecessor significantly, especially when detecting samples with low egg counts. The FLOTAC method has also had a superior diagnostic performance in detecting helminth eggs in goats and sheep, making it a more reliable method for these animals (G.G. Alowanou et al., 2021; Bosco et al., 2014).

That said, the McMaster method is simpler and requires less specialized equipment, making it more accessible for routine use in many settings. FLOTAC, while offering higher diagnostic accuracy, involves more complex preparation and analysis procedures, which may limit its widespread adoption without adequate training or resources.

Contemporary Advances in Automated Fecal Egg Count Techniques

No matter the methodology, there are always limitations inherent in manual counting. Firstly, the prospect of a reliable automatic solution can reduce manual labor, save time and resources, and eliminate human error. Furthermore, it has been shown that there is significant egg-counting variability among analysts based on their level of training. This metric was quantified in a study in 2021 that demonstrated the lack of empirical standardization (Nielsen, M. K. et al., 2021). The inception of automation in the McMaster method was driven by the necessity to mitigate these limitations.

Initial endeavors in automation capitalized on image analysis technologies, employing digital microscopy and tailored image processing software to distinguish parasite eggs based on shape, size, and color characteristics. Despite the promise of these early systems to streamline the counting process, they encountered significant challenges, notably in differentiating eggs from fecal particles and artifacts due to the complex nature of fecal samples and the morphological diversity of parasite eggs (Cringoli, 2004).

The study by Slusarewicz et al. (2016) titled, "Automated parasite faecal egg counting using fluorescence labelling, smartphone image capture and computational image analysis" represents a significant advancement in the field of veterinary parasitology, particularly for the diagnosis and management of intestinal parasites. This study introduced an innovative method combining fluorescence labeling with smartphone technology and computational image analysis to automate the counting of parasite eggs in fecal samples. This approach aimed to address the limitations associated with traditional microscopic methods, which are labor-intensive, time-consuming, and require specialized skills. By automating the process, the study sought to improve accuracy, efficiency, and accessibility of fecal egg counting (FEC) for better parasite surveillance and control.

A follow-up study that examined and compared the results of the Slusarewicz et al. (2016) study is "Evaluation of accuracy and precision of a smartphone-based automated parasite egg counting system in comparison to the McMaster and Mini-FLOTAC methods" by Scare et al. (2017). This study evaluated the diagnostic performance of the automated system introduced by Slusarewicz et al. against two established manual counting methods: the McMaster technique and the Mini-FLOTAC method. The comparison focused on accuracy, precision, and the potential for widespread adoption in veterinary practices. The findings from Scare et al. indicated that the

smartphone-based automated system could achieve comparable accuracy and precision to traditional methods, highlighting its potential as a practical tool for fecal egg counting in various settings.

Although most of the modern methods have utilized computer vision techniques such as image analysis, shape manipulation, and feature extraction, there is certainly the potential for a deep learning-based solution if an adequate database is compiled. A dataset created by Nantheera Anantrasirichai as a submission for the ICIP 2022 Challenge on Parasitic Egg Detection and Classification in Microscopic Images contains images of 11 parasite types found in human stool samples (Anantrasirichai et al., 2022).

There is not much overlap between the types of parasites that prove to be etymologically dangerous among human's versus among goats and sheep, however, the dataset has been cited by nine machine-learning research papers in the past two years and has yielded great results. The dataset prudently annotated a diverse array of images taken on both microscopes and smartphones. The paper mentions that there is no prior dataset that includes images taken without a microscopic level of zoom. Because of this, a recent publication created smartphone compatible software that leveraged transfer learning and achieved a high level of accuracy in identifying human-prevalent parasites. (Suwannaphong, T. et al., 2024).

According to these sources, so long as high-quality data can be compiled, a deep learning solution theoretically should perform very well. Furthermore, more chemical preprocessing leveraging compounds such as Chitin could feasibly result in further improvements. Both methods together with a smart phone algorithm are key to a cost-efficient at-home solution for farmers. Ideally, a marketable chemical solution kit alongside an App-Store application that is available, accessible, and accurate will be created. Table 1 below summarizes the pros and cons of each test.

Table 1: *Comparison between McMaster/Mini-FLOTAC tests and an automated fecal egg count algorithm*

	McMaster & Mini-FLOTAC	**Automated FEC**
Pros	Accurate and Precise	Similar accuracy as McMaster, Cheaper and faster than McMaster/Mini-FLOTAC
Cons	Expensive ($16 per test), Time-consuming (two weeks for results)	Large Dataset Required for Reliable Algorithm

2.3 Problem Definition

Susan Skalak, the owner of Ringadal Farm, faces the challenge of effectively monitoring her numerous sheep for parasites, particularly barber pole worms, which pose a significant health risk to her flock. Currently, she relies on sending fecal samples to veterinarians in Madison County and Louisa County for testing, incurring high costs and experiencing delays of up to two weeks for results. These delays can lead to unchecked parasite infestations, resulting in potential harm to the sheep and an increased financial burden for Susan. And Susan introduces that these

problems are not unique to her but also to her fellow farmers who also need to periodically check the presence of parasites in their livestock.

To address this issue, the primary objective is to develop a cost-effective and efficient solution for parasite monitoring in Susan's sheep, which could be achieved by reducing reliance on expensive veterinarian services and minimizing testing delays. Specifically, the solution should enable Susan to conduct parasite detection in less than two weeks and under $16 per test, which typically goes into sending samples to vets and getting back the results. Moreover, the solution must have an accuracy that is equal to or superior to that of the current Modified McMaster Test, which has a 74.6% accuracy according to Das, et al.

In addition to the main objectives, the solution is bound by material constraints: the solution must be designed such that pictures taken from an iPhone 6 are sufficient in counting fecal eggs in each sample. Moreover, the application should be able to run natively in case of no internet access that some farmers face. By addressing these objectives and constraints, the proposed solution aims to empower Susan Skalak and Ringadal Farm with a comprehensive and sustainable approach to parasite monitoring, ensuring the welfare and productivity of her livestock while optimizing resource utilization.

3. INITIAL SOLUTION IDEATION

A successful solution to the problem requires both the accuracy of the modified McMaster test and sufficient simplicity for the client to use. Three potential designs have been proposed, including a convolutional neural network trained with a mass fecal egg image dataset, a feature-based computer vision algorithm that identifies fecal eggs based on shapes and colors, and a chitin dye design that makes fecal eggs easier to identify by making them glow. In the following sections, the selection process for the initial design and future progress that must be implemented to enhance the design will be discussed.

3.1 Development of Alternative Designs

During initial team ideation, there was an abundance of articles from parasitological, biochemical, and technological journals discovered, sharing their methodologies and results through the implementation of various techniques aimed to diminish the time to detect parasite eggs within fecal samples. Such techniques consisted of manipulating egg DNA using chitin-binding proteins, dyeing using methylene blue-glycerol and Lugol's iodine, alternative McMaster faecal egg floatation methods, convolutional neural networks, and computational image analysis (Slusarewicz, 2016; Khanna, 2014; Bosco e al., 2014). Although each solution employed offered its own advantages, due to resource availability, time constraints, and/or budgeting, certain techniques had to be ruled out from further ideation. Particularly, chitinase protein DNA manipulation, Lugol's-based chitinase dyes, and photodegradable hydrogel biosensors.

Considering the client's original suggestion of a smartphone app that utilizes the device's native camera, three main potential solutions were selected for their potential integration to a web application or smartphone. The most accurate, when given ample data, is the convolutional neural network, composed of a series of convolution, pooling, and dense layers. CNNs are powerful tools for visual recognition tasks because they can automatically learn hierarchical representations of features directly from raw pixel data, without the need for manual feature extraction [Figure 3]. Offering developmental simplicity at the cost of accuracy is the feature-based image analysis algorithm. This type of computer vision algorithm uses identified key features within the image, such as corners, edges, or blobs. Following this detection, descriptors are computed to encapsulate the local characteristics surrounding each key point, encoding information about gradient orientation, intensity, or color distribution. After, features identified in one image are matched with those in another [Figure 4]. The final solution, if the others proved insufficient or impossible, was the implementation of a chitin dye during the McMaster parasite egg flotation.

The greatest influences on the evolution of the alternative designs were the largely restrictive constraints and a lack of infected faecal samples and McMaster slide images. Unfortunately, as pixel data constitutes the bedrock for the convolutional neural network, the network's capacity to discern intricate patterns and correlations within images would be compromised. Moreover, pixel data affords the network exposure to a diverse spectrum of visual contexts and environmental nuances, pivotal for cultivating resilience against variability. It is for this reason that, although less accurate and resilient, a feature-based algorithm using available McMaster imagery was selected as the solution going forward.

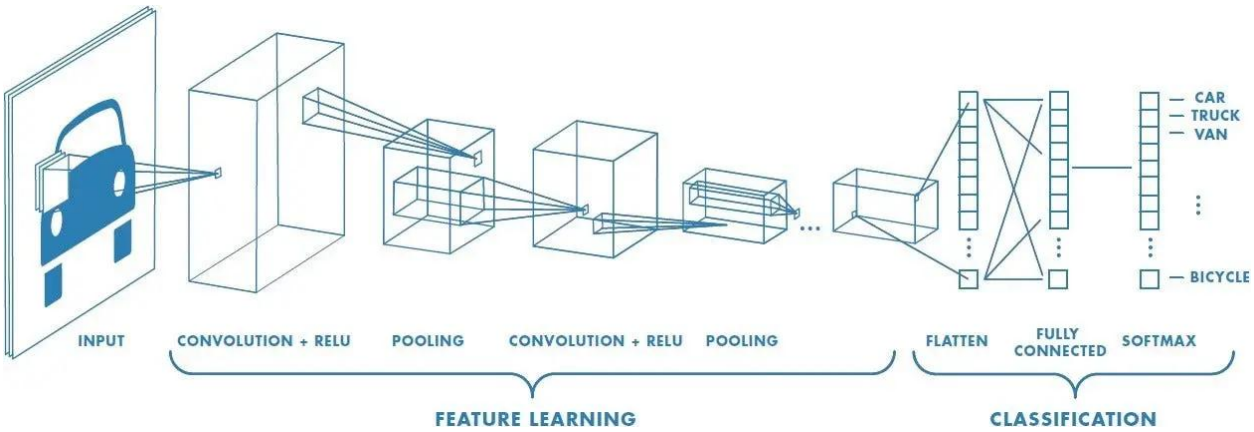


Figure 3:
Convolutional Neural Network General Process

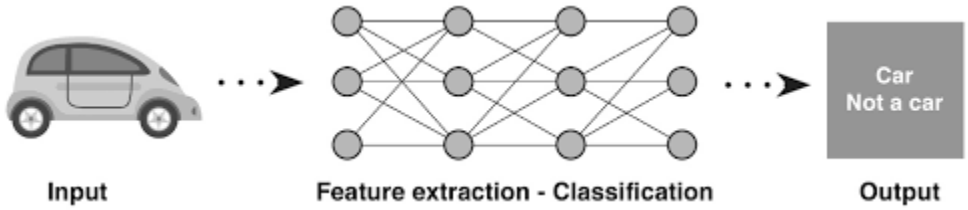


Figure 4:
General Feature-Based Image Analysis Algorithm Process

3.2 Selection of Initial Design

The biggest challenge in selecting the initial design was finding one that guaranteed accuracy comparable to a modified McMaster technique in the laboratory, a major criterion provided by the client, while keeping the design practical enough for implementation. This was especially difficult since each of the solutions was able to capture either accuracy or practicality, but not both, as shown in Table 1.

The first choice, the convolutional neural network, initially seemed to be the most accurate out of the three due to its human-like ability to recognize patterns of fecal eggs, given a large dataset of images similar to what the client would take under the microscope. However, without sample images provided by the client and tasked to find such images online, it was quickly realized that this model would suffer greatly from the small dataset available and inconsistency between its trained data and images provided by the client.

The second choice, the feature-based computer vision algorithm, avoided this problem by using pre-defined features of fecal eggs instead of learning patterns from images. However, a loosely defined threshold for such features would greatly limit accuracy, as they could also be present in debris or other compounds in ruminant feces. A similar problem persisted in the final choice, the chitin dye design, where uncertainty arose regarding whether the fluorescent dye would successfully highlight only the parasite eggs or other particles inside the feces as well.

With the inability to acquire a large dataset of reliable fecal egg images, the team began to lean more toward the feature-based computer vision algorithm. Although not as accurate as a well-trained CNN, this design was found to be practical for implementation within the given timeframe and accurate compared to the other solutions, given the project's constraints. Moreover, although the chitin dye seemed like a potential supplement to the feature-based algorithm by making the

parasite eggs more visible for identification, lack of evidence or specific research on its effectiveness on fecal eggs prevented its inclusion in the initial design.

With the initial design finalized as the feature-based algorithm, the team finished the selection process by planning ways to enhance the model's accuracy, mainly through defining specific thresholds for features unique to fecal eggs. To achieve this, an approach similar to training a CNN was taken, using various fecal egg images to map out the shape and colors of the eggs and find one that best separates the fecal eggs from the rest of the feces. Unlike CNN, however, since only limited features were sought from these images, the algorithm could be developed with a small image dataset, making this approach both accurate and practical.

Table 2: *Comparison on Proposed Designs with Strengths and Weaknesses*

	Convolutional Neural Network	**Feature-Based Algorithm**	**Chitin Dye**
Strengths	Accurate & Precise Results with Reliable Image Dataset	No large image dataset required so easier to develop algorithm	Easier for camera to detect parasite eggs, potentially increasing accuracy
Weaknesses	Finding a large reliable dataset is difficult. Inability to provide large dataset makes model reduces accuracy	Not as accurate as a well-trained convolutional neural network with large, reliable image dataset	Uncertainty on whether design would successfully highlight just parasites eggs

3.3 Detailed description of design

The low-fidelity prototype is a feature-based algorithm, leveraging the OpenCV library and written in C++. The algorithm maps contours detected in microscopic fecal images to features derived from a model of a Haemonchus contortus parasite egg. Due to its high fatality rate among ruminants, Haemonchus contortus, also known as the barber worm, was indicated as the highest priority parasite in the project and was therefore prioritized in the algorithm's design.

In the development of an OpenCV-based algorithm for the detection and quantification of barber worm eggs in microscope fecal samples of goats, a feature-based approach is adopted to analyze contours characteristic of parasite eggs. The algorithm initiates typical preprocessing of microscopic images, converting each image to grayscale to simplify analysis. This step is followed by the application of Gaussian blur to reduce image noise and improve the detection reliability of subsequent operations. Edge detection techniques, specifically Canny edge detection, are then applied to highlight the boundaries of potential egg shapes within the fecal matter. The resultant binary image serves as the foundation for contour detection, an essential step in identifying candidate blobs that may represent barber worm eggs.

The core of the algorithm revolves around the analysis of detected contours to distinguish those corresponding to barber worm eggs from irrelevant shapes. Each contour is examined for geometric properties that align with the expected dimensions, aspect ratios, and solidity of barber worm eggs, employing criteria developed through empirical analysis of known egg samples. To enhance the specificity of detection, a color filtering step is incorporated, targeting the characteristic color range of the eggs. This is achieved by converting the image from the RGB to the HSV color space, which is more effective for color segmentation. Contours that meet both the geometric and color criteria are then subjected to further analysis to confirm their identification as

barber worm eggs. This includes fitting ellipses to the selected contours and calculating their orientation and eccentricity, parameters indicative of the eggs' oval shape.

For each identified egg, the algorithm increments a count, then, following the McMaster Slide standard, multiplies the count by 100 to provide an estimate of the parasite egg load per gram in the sample. Additionally, the algorithm generates an annotated output image where detected eggs are highlighted, facilitating visual verification of the detection results.

After the algorithm is developed and incorporated into the web application, they will need to ensure they find a free domain where the application can be hosted. Finding a free domain will help their client have access to the website beyond their Engineering class without putting any financial burden on their end, helping them maintain the website in the long term.

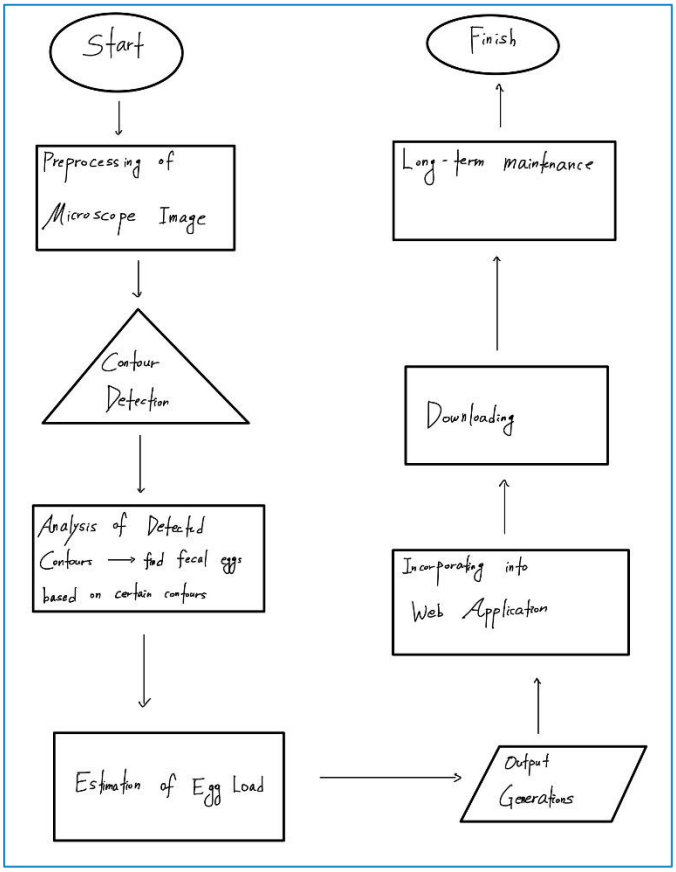


Figure 5:
Workflow Chart

Bill of Materials

The Bill of Materials includes 4-month subscriptions to Google Gemini and ChatGPT-4 to aid in providing insights during the coding process, generating code snippets, and documenting the process. Additionally, it would allow potential integration with the ChatGPT API, enabling the incorporation of OpenAI and other advanced functionalities into the web application.

Table 3: *Bills of Materials*

Supplier	Item Description	Item Order	Unit Cost	Quantity	Total Cost	Link
ChatGPT	ChatGPT-4 Plus	1	$20	4 months	$80	https://chat.openai.com/#pricing
Google	Gemini	1	$20	4 months	$80	https://one.google.com/explore-plan/gemini-advanced?utm_source=gemini&utm_medium=web&utm_campaign=sidenav_evo

Proposed Workflow

These charts provide a rough idea of the tasks assigned to each member of the team and the steps needed to solve the problem. Table 4 below outlines the tasks for each team member:

Table 4: *Workflow*

Product	Responsibility	step
doucument	GROUP	Finalize the solution that we want to used based on the data we been given
	Aiden	1.1 contact local farmer and researcher to gather data and samples
		1.2 finish the Gantt chart for the assignment
		1.3 research and finish the bill of material
	Eddie Xiao	1.1 finish the flow chart for Initial Idea Solution Draft
		1.2 start working on the back end of the website with Donghwa
	Donghwa	1.1 complete selection of inital design
		1 2 working on the back end of the website with Eddie
	James	1.1 researching decent image that we can use in the algorithm
		1.2 completer the development of alternative design
	Tommy	1.1 start coding and using a vision algorithm with open CV to find ovals with certain color inside tehm
		1.2 completing description of our algorithem and our future maintenance plane
	group	2 design the function and the layout of the website
		3 code the website and test if the algorithem works
		4 ask for some picture from the client and test out if the website actually work
final Poster		
Final Report		

Table 5 depicts the team's Gantt Chart, illustrating when each task should be accomplished, and the duration required to complete it:

Table 5: *Gantt Chart*

Product	Responsibility	step
doucument	GROUP	Finalize the solution that we want to used based on the data we been given
	Aiden	1.1 contact local farmer and researcher to gather data and samples
		1.2 finish the Gantt chart for the assignment
		1.3 research and finish the bill of material
	Eddie Xiao	1.1 finish the flow chart for Initial Idea Solution Draft
		1.2 start working on the back end of the website with Donghwa
	Donghwa	1.1 complete selection of inital design
		1 2 working on the back end of the website with Eddie
	James	1.1 researching decent image that we can use in the algorithm
		1.2 completer the development of alternative design
	Tommy	1.1 start coding and using a vision algorithm with open CV to find ovals with certain color inside tehm
		1.2 completing description of our algorithem and our future maintenance plane
	group	2 design the function and the layout of the website
		3 code the website and test if the algorithem works
		4 ask for some picture from the client and test out if the website actually work
final Poster		
Final Report		

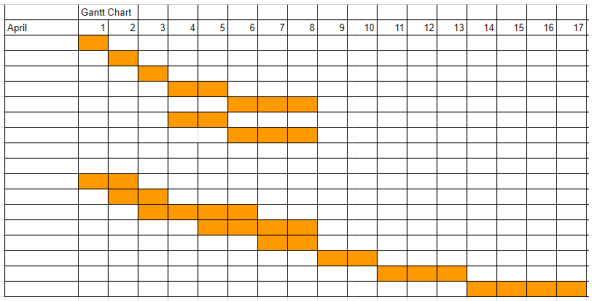
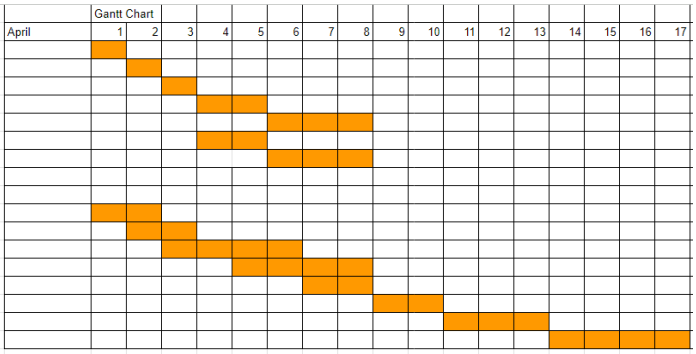


Table 6 illustrates a combined Gantt Chart, which will help highlight tasks that have already been completed and tasks that will need to be worked on in the future:

Table 6: *Workflow and Gantt Chart*

Product	Responsibility	step
doucument	GROUP	Finalize the solution that we want to used based on the data we been given
	Aiden	1.1 contact local farmer and researcher to gather data and samples
		1.2 finish the Gantt chart for the assignment
		1.3 research and finish the bill of material
	Eddie Xiao	1.1 finish the flow chart for Initial Idea Solution Draft
		1.2 start working on the back end of the website with Donghwa
	Donghwa	1.1 complete selection of inital design
		1.2 working on the back end of the website with Eddie
	James	1.1 researching decent image that we can use in the algorithm
		1.2 completer the development of alternative design
	Tommy	1.1 start coding and using a vision algorithm with open CV to find ovals with certain color inside tehm
		1.2 completing description of our algorithem and our future maintenance plane
	group	2 design the function and the layout of the website
		3 code the website and test if the algorithem works
		4 ask for some picture from the client and test out if the website actually work
final Poster		
Final Report		



Maintenace and Product lifespan

The fecal egg detection website's lifespan should be forever. There isn't much maintenance to do but, in the future, a team of experts can manage checks for data, accuracy, performance, and security. Overall, the plan is to ensure that the website remains a reliable tool for the client.

4. TESTING RESULTS, ANALYSIS AND DESIGN REVISIONS

Our web application is comprised of three components, hence three areas for testing: the frontend user interface (UI), the Computer Vision algorithm for calculating an accurate fecal egg count from a provided microscope image, and the backend server for handling data sent from the client and results from the Computer Vision algorithm. In this section, necessary tests are planned and performed to address all three of these areas to successfully build the final product.

4.1 Initial fabrication and testing plan

Testing Process Overview

Testing the initial design involved three major steps: testing whether the web application could properly supply the Computer Vision algorithm with the client's image, verifying the accuracy of the algorithm, and checking whether the results were correctly displayed back to the client. To test the first and third components, a variety of images with different file sizes were fed into the application through a form in the front-end UI. The backend program would save a file with the detected parasites and output the following in the file output stream. The frontend would extract this data and display it to the UI. Once the application displayed the results back into the UI, it was confirmed whether the correct fecal egg count, eggs per gram, and the final analyzed image were present. In addition, the uploads folder, which temporarily stores the client's images in the back end, was checked at the end to ensure the original image was only used to run the algorithm and not saved permanently.

In addition to testing the UI and server, three tests were used to test the backend algorithm's performance. The first test tested the number of correctly identified parasites. In addition to this, it yielded a visual result with an annotated test image.

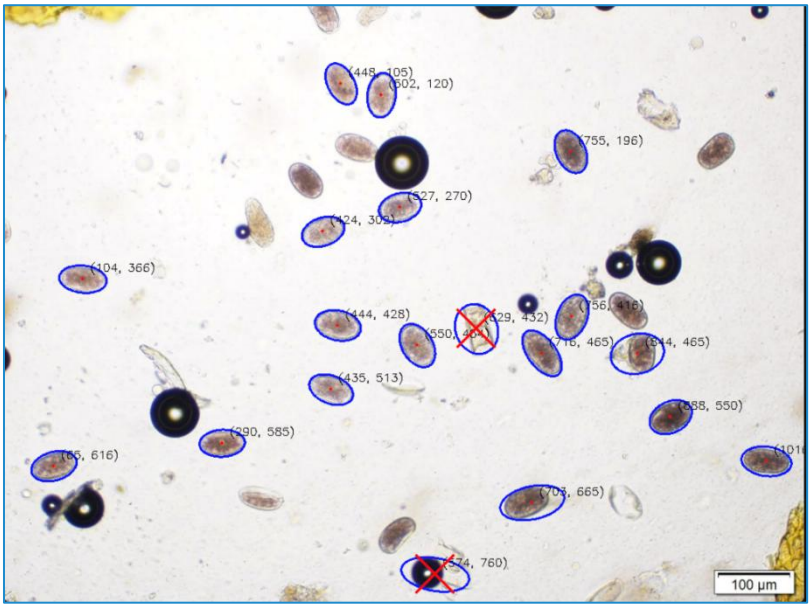


Figure 6:
Low Fidelity Eclipse Identification

The second test leveraged an error function based on the client's needs. This is discussed in more detail in the following section. This test helped prioritize certain success metrics and to analyze the success of the algorithm with more scrutiny. When running this test, a complete terminally output error analysis was implemented, which can be seen in Figure 7 below:

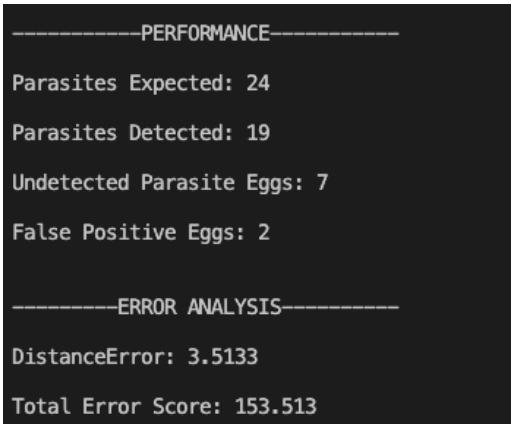


Figure 7:
Output Error Analysis

A test image file was also saved, which served to display the correct parasites bounded by a green box and the detected parasites circled in red. Alongside the test images, a graph that calculated the optimal binary threshold value was designed, a very important parameter that is discussed in a following section, for each image. This graph can be seen in the top right corner of Figure 8 and the results for each image can be seen in the other sections of the result image:

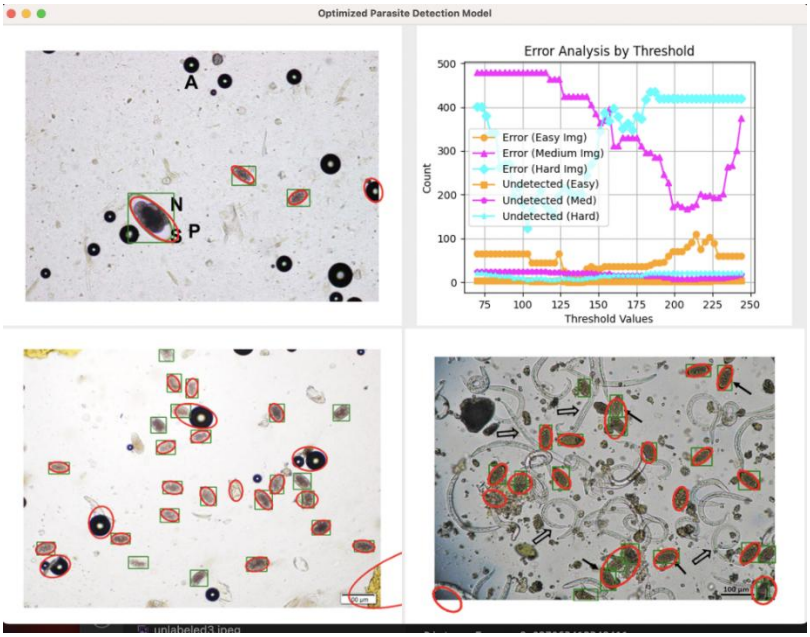


Figure 8:
Second Test Sample Generated Result Metric Page

Finally, as a third test, the observed categorical health metrics were compared, determined according to the guidelines of the American Pharmacists Association as shown in Table 7, enabling testing of the real-world results that the client will base their decisions on (SCOPS, 2022):

Table 7: *Classification of Health Condition Based on Egg Count (SCOPS, 2022)*

Worm egg count	Comment	Action
50-350 eggs per gram	Light infestation	Treatment not necessary
400-600 eggs per gram	Moderate infestation	Anthelmintic treatment may be beneficial
650-1000+ eggs per gram	Heavy infestation	Anthelmintic treatment necessary

Error Function Architecture

To test the actual algorithmic solution, the first order of business was to engineer an error function. When doing this, five test images were first selected, three of which represented heavily infected fecal samples, one represented a lightly infected fecal sample, and the last represented a completely healthy sample. After annotating the images in Roboflow and saving the results to respective files, a pipeline was created to easily compare detected parasites with the expected parasites.

We classified a prediction as "correct" if the predicted center was within 30 pixels of the correct center. When generating error, three factors were considered in the following order of influence on the total error:

1. Undetected Parasites
    a. In accordance with the interview with Ms. Shalak, minimizing the number of undetected parasites was the main priority. The worst outcome was diagnosing a treatable but infected goat as healthy.
2. Incorrect Guesses
    a. It was also prominent to avoid incorrect guesses. Minimizing incorrect guesses was less important than minimizing undetected parasites, however, because treating a healthy goat for parasites does not do them any harm and will only waste a small amount of time and money, much less significant than the life of a goat.
3. Precision of Correct Guesses
    a. Finally, but least consequentially, the precision of "correct" predictions into the error functions were factored in. It was ensured that parasite detections were spot on and not larger or skewed with respect to the actual parasites. Although trivial when extracting data about a single goat, optimizing this process will yield better results overall.

$$\text{Total Error} = \sum_{i=1}^{N} \left\{ \begin{array}{ll} \frac{1}{10} \cdot d_i & \text{if guess } i \text{ is correct} \\ 5 & \text{if guess } i \text{ is incorrect} \end{array} \right\} + 20 \cdot P$$

where:

N is the total number of "observed: parasites.
d_i is the distance from the correct center for the i-th guess.
P is the number of undetected parasites.

Figure 9:
Equation for Total Error

Explanation of Binary Thresholding

We used the error function in Figure 9 to optimize a binary threshold parameter, something that is integral to the algorithm and is hard to pinpoint for any image. This is an essential step in an algorithm that takes a gray-scale image and turns it into black and white values. It does this by selecting an intensity threshold such that pixels above that intensity turn white and below that intensity turn black. This is super useful for detecting shapes and contours in an image based on intensity. An example of a binary-threshold image is shown in Figure 10:

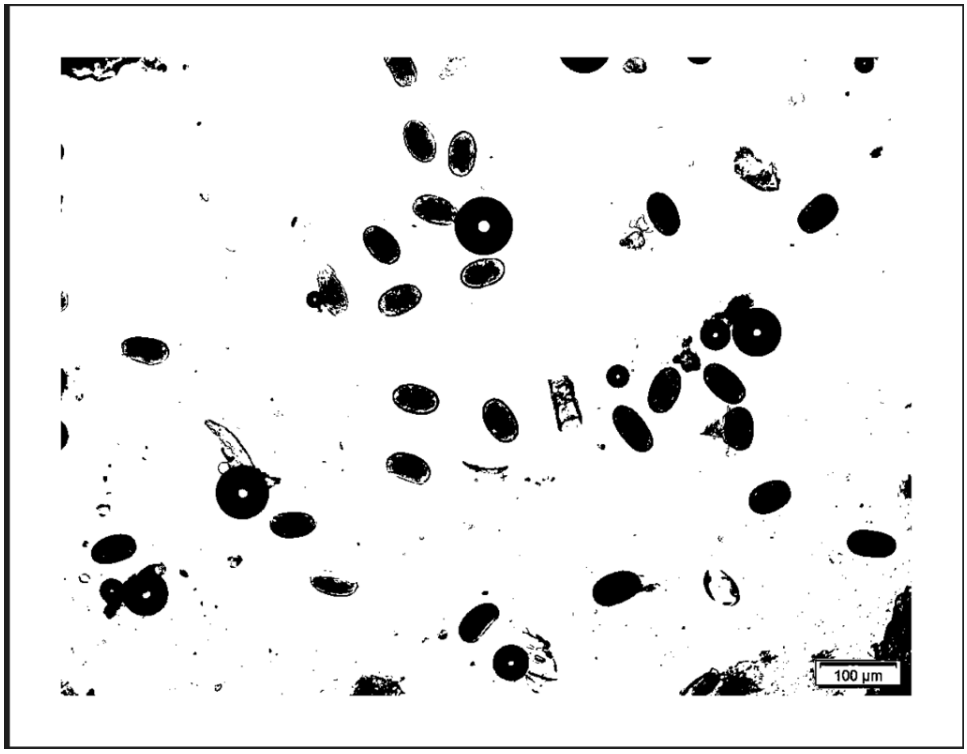


Figure 10:
Binary Threshold Stage Example on Medium Sample Image

4.2 Solution Development, Testing, and Revisions

4.2.1 Software Development

At a high level, the software follows the archetypal Model View Controller (MVC) architecture. The model tier is a backend algorithm that outputs an image showing the location of the detected parasites and information about the hypothesized health condition of the sample.

The controller tier is composed of an Express.js server which takes in an image file input from the user, sends it to the backend algorithm and runs it using a child process, receives fecal egg count results from stdout, and sends the results back to the frontend user interface (UI) of the website. The view tier is composed of the website UI which was built using HTML/CSS/JavaScript and hosted using Adaptable. After designing the UI and server, the web application was containerized using Docker, and the Docker template was hosted on a deployment service called Adaptable.

There are three important pipelines that leverage the sever. First, there is a front-end file input system enabled by a file drop. The controller sends this to the backend and the image is processed by the model tier. Secondly, there is another image-save-and-transfer protocol that allows the parasite-detected image to be sent back to the front end. Finally, there is a protocol that allows the output health data to be sent from the Model to the terminal output and subsequently to the front end.

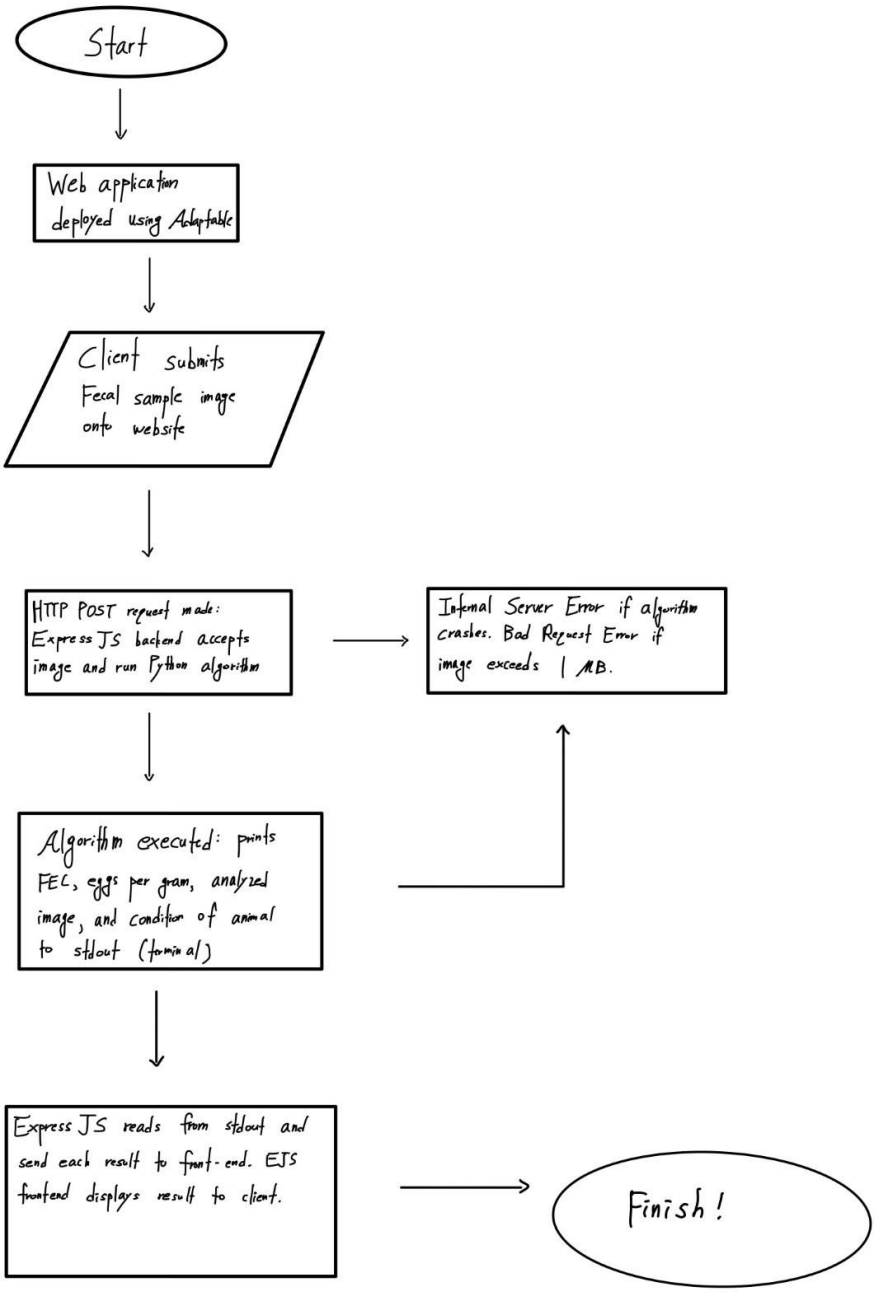


Figure 11:
Simplified Software Flow

Website Development

Testing the functionality of the deployed web application showed that, while feeding the algorithm with the client's images and sending the result back was done correctly, there was a limit to the file size that the client could upload. Specifically, the client was limited to reliably uploading a maximum of 1 MB images due to memory limitations in the free plan of the deployment service that was utilized, Adaptable. The limitation was not seen in the local host, suggesting that this could be fixed if a paid plan was used, or a database was involved.

Algorithm Development

The model tier of the software was decomposed into several models that accomplished a sequential flow. This chronological description succinctly describes the functionality of these modules that come together to form the backend algorithm.

- ImageReader – Read image file into a computable matrix
- ProcesserHelp – Converts to Grayscale Intensity Image + Applies 3x3 Gaussian Blur
- Thresholder – Applies Binary Thresholding in accordance with custom optimal threshold identification function
- ContourDetector – Returns list of all connected contours present in a binary image
- EllipseDetector – Defines the constraints of what contours are elliptical and extrapolates ellipses from a list of contours.
- ParasiteDetector - (Extends EllipseHelper) Determines locations of parasites based on a list of contours using feature matching (i.e. aspect ratio, size, color, shape etc.)

EllipseDetector

The first function within EllipseDetector, assessEllipse, determines if a detected ellipse fits the size and shape qualities of a parasite.

```
def assessEllipse(self, ellipse, size):
    OT = 0.4 #OUTER THRESHOLD
    IT = 0.8 #INNER THRESHOLD

    (center_x, center_y), (major_axis, minor_axis), angle = ellipse

    # Calculate the area of the ellipse
    area = np.pi * (major_axis/2) * (minor_axis/2)


    # Calculate the aspect ratio of the ellipse
    ar = major_axis / minor_axis

    good_ratio = (ar > OT and ar < IT) or (ar > 1/IT and ar <1/OT)

    areaMin = size[0]*1.5
    areaMax = size[0]*7

    return (area > areaMin and area < areaMax and good_ratio)
```

Figure 12:
Ellipse algorithm within EllipseDetector Module

Additionally, within EllipseDetector there are two functions that fit an ellipse to a contour. One approach, namely the RANSAC approach, took many random samples from the contour and selected the best ellipse. The other, namely the polygon approximation approach, attempted to fit an ellipse to the entire contour and discarded the ellipse if the result was unsatisfactory. Both methods could be used interchangeably.

```python
def fitEllipse_Polygon(self, Contour, size):
    area = int(cv2.contourArea(Contour))  # Cast area to int (though not necessary in Python)
    bounding_box = cv2.boundingRect(Contour)
    aspect_ratio = bounding_box[2] / bounding_box[3]  # Width divided by Height

    # Filter based on aspect ratio and minimum size
    elongation_factor = abs(1 - aspect_ratio)

    areaMin = size[0]*1.5
    areaMax = size[0]*7

    if 0.1 < elongation_factor and elongation_factor < 1 and areaMin < area and area < 8000:
        # Approximate contour to smooth shape
        epsilon = 0.01 * cv2.arcLength(Contour, True)
        approx = cv2.approxPolyDP(Contour, epsilon, True)

        if len(approx) > 4:  # Need at least 5 points to fit ellipse
            fitted_ellipse = cv2.fitEllipse(approx)
            # You can draw the ellipse on the image to visualize
            return fitted_ellipse
    return None
```

Figure 13:
FitEllipse_Polygon function within EllipseDetector

```python
def fitEllipse_RANSAC(self, Contour, size):
    if len(Contour) < 5:
        print("Contour Too Small")


    maxInliers, bestFit = 0, ((0,0),(0,0),0)

    MaxIterations = int(len(Contour)/3)
    for i in range(MaxIterations):
        sample = np.array([Contour[i][0] for i in random.sample(range(len(Contour)),5)])
        potentFit = cv2.fitEllipseDirect(sample)

        if self.assessEllipse(potentFit):
            numInliers = 0
            for pt in Contour:
                pt = tuple(pt[0])
                d = self.distPoint_Ellipse(potentFit, pt)[0]
                if d < 10:
                    numInliers += 1
            if numInliers > maxInliers:
                maxInliers = numInliers
                bestFit = potentFit

    return bestFit
```

Figure 14:
FitEllipse_RANSAC function within EllipseDetector

Although it is not discussed in detail in this paper's testing section, the holistic elliptical approach was used in the version of the software as of publication. However, the functionality still exists in the software for future exploration.

Thresholder

The thresholder class contained several interchangeable protocols used to generate a binary image for the sample. Extensive testing was done in pursuit of an optimal protocol within this module. Some approaches included:

- Using a universal threshold value
- Blurring the grayscale image with a 3x3 kernel or not
- Using Otsu's Thresholding Algorithm on a blurred image
- Using Otsu's Thresholding Algorithm with a translational shifting factor

The code of the final approach present in the beta version of the software is discussed later.

DetectParasites

This strategy integrated the smaller custom modules with OpenCV libraries used to implement the ImageReader, Processor, and ContourDetector modules.

```
def detectParasites(fileName, thresh, outFile = "", saveImages = True):

    im_gray = cv2.imread(fileName, cv2.IMREAD_GRAYSCALE)

    binary = getBinaryThreshold(im_gray, thresh)

    contourImage = cv2.imread(fileName)
    ellipseImage = cv2.imread(fileName)

    size = im_gray.shape

    im_gray, binary, contourImage, ellipseImage = pad_all([im_gray, binary, contourImage, ellipseImage])
    contours = cv2.findContours(binary, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)[0]

    num_eggs = 0
    ellipseHelper = ellipseDetector()
    for contour in contours:
        if len(contour)>=5:
            bestEllipse = getBestEllipse(ellipseHelper, contour, size)
            if not bestEllipse == None:
                cv2.ellipse(ellipseImage, bestEllipse, (0,0,255), 3)
                num_eggs+=1
```

Figure 15:
DetectParasites Method in detect.py

An important practical improvement that was made to DetectParasites was the inclusion of the pad_all function. This function added a 64-pixel padding to the sample images. This made the detection of parasite eggs lying padding the image by 64 pixels on all sides. This allowed for more wholistic identification of parasites near or on the edge of the slide, circumventing a large setback of the contour module.

4.2.2 Summary of Algorithm Performance Testing

Initial Success Metric Testing

Testing this software aimed at mainly improving the Thresholder module responsible for binary thresholding. The reasons for this decision are discussed in section 4.1.

Several testing scripts were deployed that quantified the success of the algorithm based on the three testing methods discussed above. To establish a baseline for software improvement, the third testing metric, the categorical health assessment of all five samples against their correct health assessment, was employed with a constant binary threshold value of 210. This value was picked based on light observational analysis of what worked well on the medium test image.

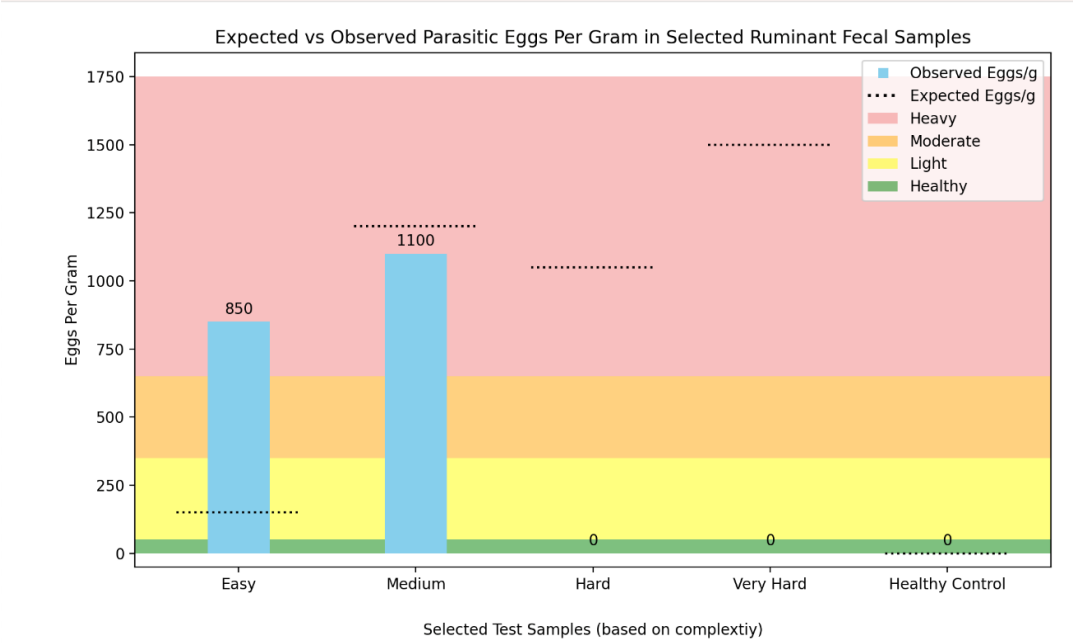


Figure 16:
Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples With Constant Threshold Value of 210

In the test 3 display, the dotted lines represent the expected egg counts for each respective test image. For this run, the algorithm only performs accurately for the medium test image, and that is a problem.

Otsu Threshold Algorithm Success Testing

As shown above, different fecal samples have different levels of contrast, which poses a problem. Some have lots of dirt in the background, almost camouflaging the eggs, while others are relatively clear. Because of that, it becomes clear that different threshold values must calculated for each image to attain satisfactory results. A very well-known algorithm that accomplishes this is Otsu's Thresholding Algorithm. The algorithm essentially picks the threshold value that minimizes the variance in pixel intensity to the left and right of that threshold value. The results of employing this algorithm to select the threshold parameter can be seen in Figure 17:

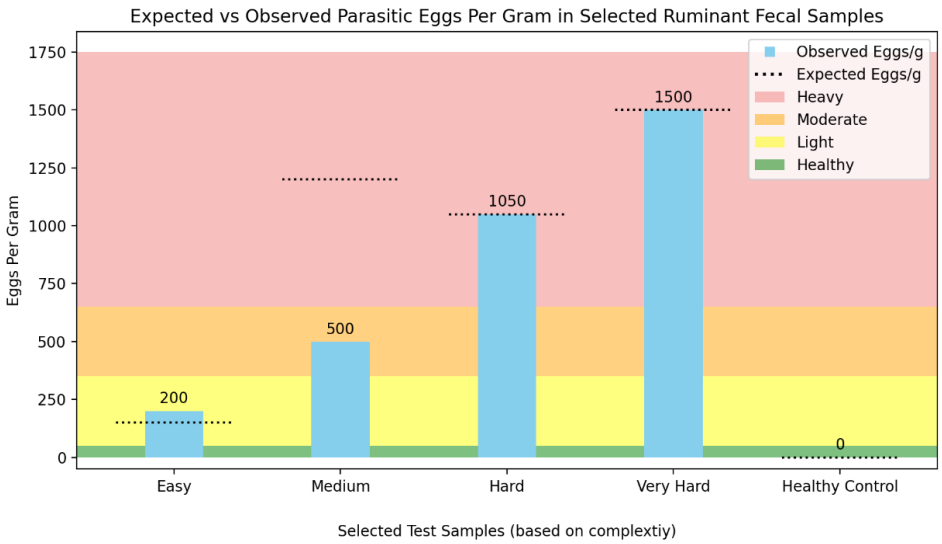


Figure 17:
Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples With a Threshold Value Calculated for Each Image Using Otsu

This is a lot better but falls short on the medium test image. To address this, more testing was employed to modify Otsu's algorithm to meet the needs.

Testing for Thresholding Algorithm Modification

To modify Otsu's algorithm to meet the criteria of the client, a script was devised to calculate the optimal threshold value for each test image according to the error function. These tests are shown in Figure 18. Green boxes highlight the expected parasite locations and red circles indicate the locations of parasites as predicted by the algorithm. The graph shows the error for different binary threshold values.

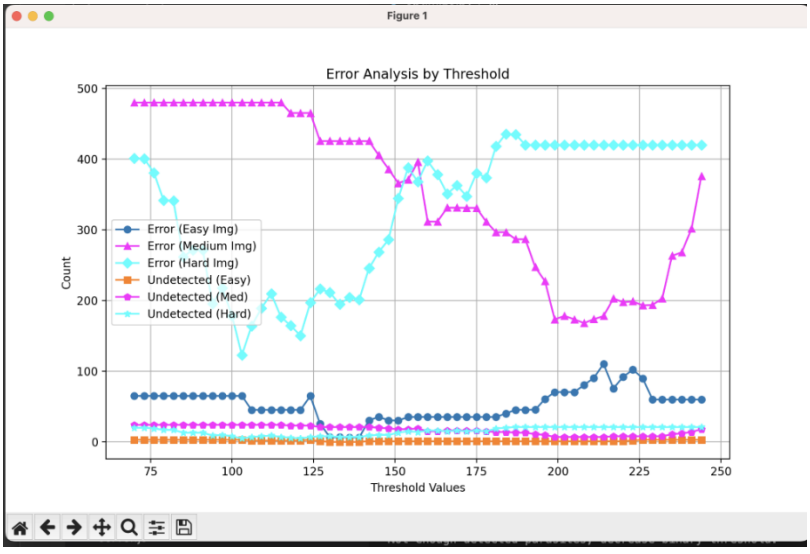


Figure 18:
Error Analysis Graph

The optimal threshold values were slightly lower than their respective Otsu calculated values, indicating that a translational shift could result in a stronger performance.

4.3 Informed Design Revision

Website Development Revisions

To address file size limitations in the deployment service, a hard limit on the client's image input was set directly in the Express back end (as shown in Appendix C). Once the client uploads an image that is too large, the web page sends a status 400 (Bad Request) error with a message informing the client to upload a smaller image. This was done to ensure the deployed service does not crash the application with a "Service Unavailable" message, which does not specify to the client what the cause of the error was and how to fix it.

Algorithm Development Revisions

Because of the observations that were made comparing the results of Otsu's thresholding algorithm with the optimally determined threshold value based on the error function, a leftward skew to the results from the algorithm was employed. This made it such that thresholding was more liberal, and it was possible to catch all relevant shapes in the image. The result was that every test image was correctly classified in accordance with its actual health condition.

```
def getBinaryThreshold(gray_image, threshold):
    blur = cv2.GaussianBlur(gray_image, (3,3), 0)
    SHIFT_CONSTANT = -20
    otsu_threshold, _ = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

    return cv2.threshold(blur, otsu_threshold + SHIFT_CONSTANT, 255, cv2.THRESH_BINARY)[1]
```

Figure 19:
Final GetBinaryThreshold method in Threshold

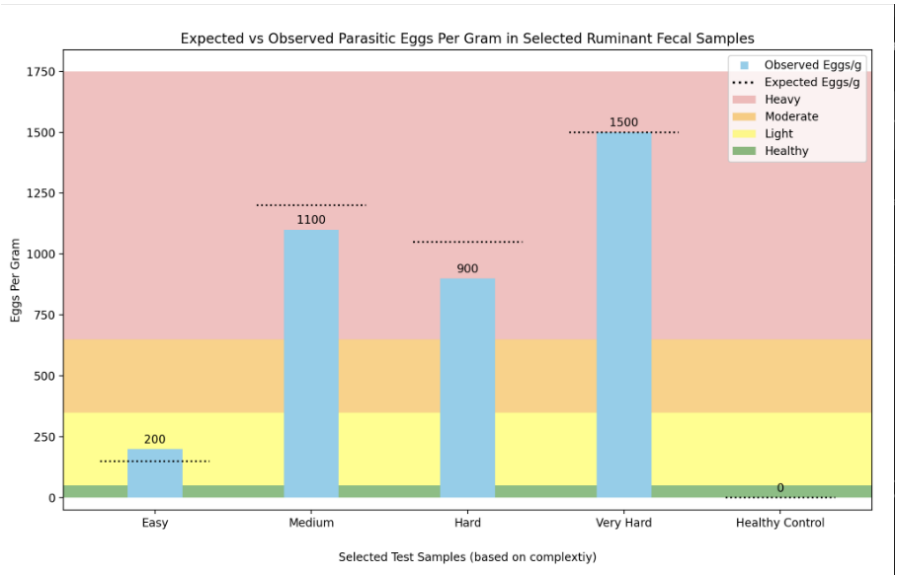


Figure 20:

Expected vs Observed Parasitic Eggs Per Gram in Selected Ruminant Fecal Samples With Modified Otsu

5. SUMMARY OF FINAL SOLUTION

5.1 Description of final design

The final solution is a feature-based algorithm, leveraging the OpenCV library and written in Python. The algorithm maps contours detected in microscopic fecal images to features derived from a model of a Haemonchus contortus parasite egg. Due to its high fatality rate among ruminants, Haemonchus contortus, also known as the barber worm, was indicated as the highest priority parasite in the project and was therefore prioritized in the algorithm's design.

In the development of an OpenCV-based algorithm for the detection and quantification of barber worm eggs in microscope fecal samples of goats, a feature-based approach is adopted to analyze contours characteristic of parasite eggs. The algorithm initiCates typical preprocessing of microscopic images, converting each image to grayscale to simplify analysis. This step is followed by the application of Gaussian blur to reduce image noise and improve the detection reliability of subsequent operations. Edge detection techniques, specifically Canny edge detection, are then applied to highlight the boundaries of potential egg shapes within the fecal matter. The resultant binary image serves as the foundation for contour detection, an essential step in identifying candidate blobs that may represent barber worm eggs.

The core of the algorithm revolves around the analysis of detected contours to distinguish those corresponding to barber worm eggs from irrelevant shapes. Each contour is examined for geometric properties that align with the expected dimensions, aspect ratios, and solidity of barber worm eggs, employing criteria developed through empirical analysis of known egg samples. To enhance the specificity of detection, a color filtering step is incorporated, targeting the characteristic color range of the eggs. This is achieved by converting the image from the RGB to the HSV color space, which is more effective for color segmentation. Contours that meet both the geometric and color criteria are then subjected to further analysis to confirm their identification as barber worm eggs. This includes fitting ellipses to the selected contours and calculating their orientation and eccentricity, parameters indicative of the eggs' oval shape.

For each identified egg, the algorithm increments a count, then, following the McMaster Slide standard, multiplies the count by 100 to provide an estimate of the parasite egg load per gram in the sample. Additionally, the algorithm generates an annotated output image where detected eggs are highlighted, facilitating visual verification of the detection results.

After the algorithm is developed and incorporated into the web application, they will need to ensure they find a free domain where the application can be hosted. Finding a free domain will help their client have access to the website beyond their Engineering class without putting any financial burden on their end, helping them maintain the website in the long term.

The main difference between the initial low fidelity prototype and the beta-version of Eggs by the Dozen is the improvement and individualization of the Threshold module. The protocol for binary threshold value computation, progressed from arbitrarily choosing a uniform value to systemically calculating a strong threshold value as a function of a sample's features, considering background, contrast, and egg overlap alongside other features. The prototype saw progress from an overgeneralized model for a single image to an algorithm capable of correctly assessing the health condition of all tested parasite samples in varying conditions.

The overall flow of our final design is illustrated in Figure 21 below:

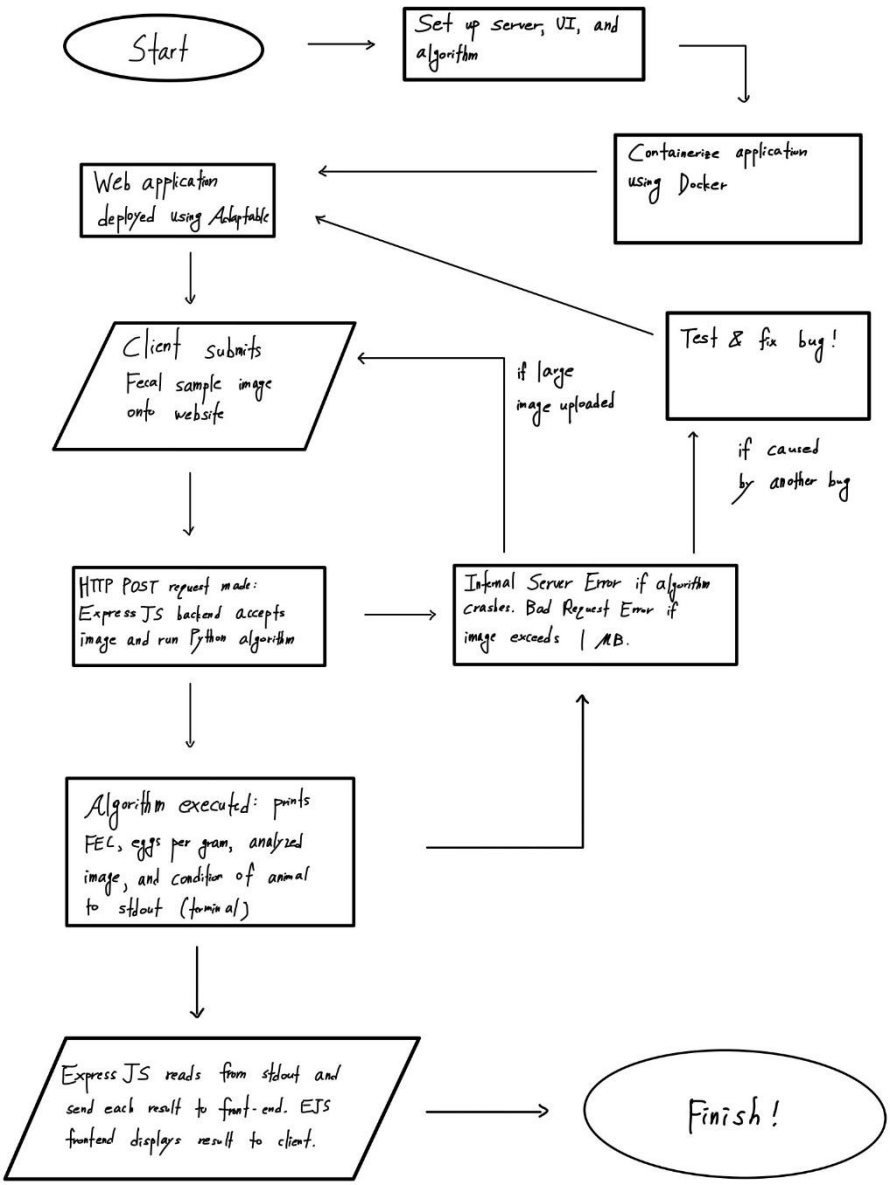


Figure 21:
Overall Flow Chart of Software Development

5.2 Fabrication workflow and schedule

Product	Responsibility	step
doucument	GROUP	Finalize the solution that we want to used based on the data we been given
	Aiden	1.1 contact local farmer and researcher to gather data and samples
		1.2 finish the Gantt chart for the assignment
		1.3 research and finish the bill of material
	Eddie Xiao	1.1 finish the flow chart for Initial Idea Solution Draft
		1.2 start working on the back end of the website with Donghwa
	Donghwa	1.1 complete selection of inital design
		1 2 working on the back end of the website with Eddie
	James	1.1 researching decent image that we can use in the algorithm
		1.2 completer the development of alternative design
	Tommy	1.1 start coding and using a vision algorithm with open CV to find ovals with certain color inside tehm
		1.2 completing description of our algorithem and our future maintenance plane
	group	2 design the function and the layout of the website
		3 code the website and test if the algorithem works
		4 ask for some picture from the client and test out if the website actually work
final Poster		
Final Report		

Figure 22:
Updated Workflow Diagram

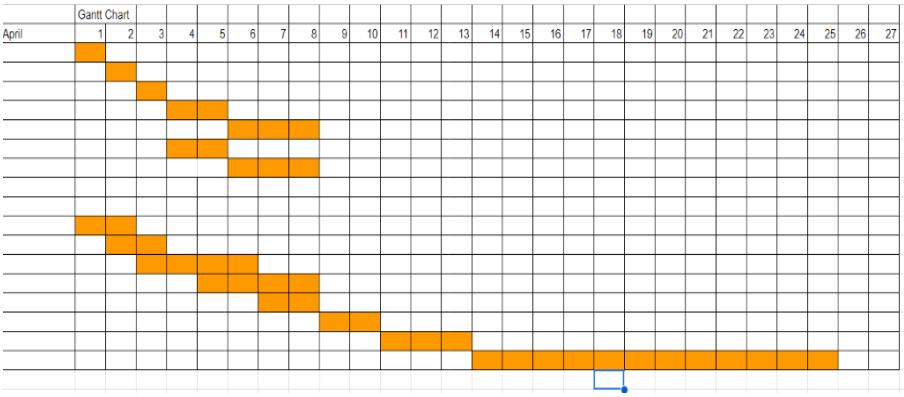


Figure 23:
Updated Gantt Chart

The difference between the Gantt Chart in Chapter 3 and the Updated Gant Chart was mainly the extension of the development of the algorithm, test cases, and the layout of the website due to the number of bugs and errors that occurred during the development of the web application and algorithms which extended the number of days that the team worked on the overall functionality of the website.

Additionally, the client and numerous different research groups was not able to send data regarding the classification of certain parasites so the algorithms that were made had to be changed to account for feature detection rather than using a Convolutional Neural Network which extended the development period.

5.3 Reflection of Final Solution fulfilling Customer Needs

The final design fulfills Susan Skalak's needs for a reliable, accurate and cheap method to detect and count parasite eggs in fecal samples of goats and sheep, that specifically minimizes undetected, infected goats.

To satisfy all the project's constraints, each test must cost less than 16 dollars, provide results faster than two weeks, be as or more accurate than the current parasite detection method, and be able to run natively. The final solution satisfied all these requirements with the web application costing 0 dollars to run per test, displaying results in less than seconds, correctly diagnosing ruminant health condition (in terms of infestation severity) for all test samples, and having web and native capabilities. Additionally, with web-hosting capabilities, the website is available not only for Susan Skalak but also for every farmer who requires it in America.

Two issues that may arise with the web application is that the server limits each uploaded image to 1 megabyte for the live hosting to remain free. While the client may get around this by utilizing image compression software, the image clarity and pixels may get compromised leading to a decrease in accuracy and results. Another issue that is outstanding is the lack of a mobile application that would make it more accessible for the client Susan Skalak to upload her images.

6. RECOMMENDATIONS AND FUTURE WORK

6.1 Future Work for Eggs by the Dozen

There are some aspects that could be improved in the future for this project. Firstly, it would be great to have more data. One of the major setbacks in this project is the lack of data. There are only a few images that can be downloaded online. However, only a few pictures are not enough to test thoroughly. With the help of more data, it becomes possible to improve the algorithm on a larger real-world scale.

Secondly, whilst the focus revolves around the detection of barber worms (Haemonchus contortus), there exists an opportunity to increase the scope of the challenge. The next step would be to detect parasites in other egg types.

Finally, an imperative step toward improving user accessibility and engagement involves the development of mobile apps. Currently, the project is confined to an internet site interface, which suits the customers' demands. The system of taking pictures of the parasite, shifting them to a pc, and looking forward to analysis will cause a lot of delay. However, with the help of a mobile app, it will be easier and faster for the client to access. She can simply take a picture of the parasite, and she will be able to get the results.

6.2 Recommendation for Future Work

Continuing the work on parasite detection is an exciting opportunity for engineers looking to make meaningful contributions to this important field. To move forward, collaborating with farmers and researchers to gather more data is critical. By teaming up with these experts, engineers can access a richer dataset, which will improve the accuracy of the algorithm. These partnerships also provide valuable insights into the real-world challenges of parasite detection, helping engineers develop solutions that better meet practical needs.

Another important aspect to focus on is reducing false positives in the algorithm. Since it uses ellipse detection, there's a risk of mistakenly identifying random images as infected with parasites. Engineers working on this project should concentrate on refining the algorithm to minimize these errors. This might involve adding new features or using machine learning techniques to distinguish genuine parasite eggs from false positives. It's crucial to rigorously test and validate these improvements to ensure the algorithm remains reliable across different situations.

In summary, by collaborating with experts and refining the algorithm, engineers can contribute to more effective parasite detection methods. This work has the potential to have a significant impact on public health and agriculture, making it a worthwhile pursuit for engineers passionate about making a difference.

7. REFERENCES

Alowanou, G.G., Adenilé, A.D., et al. (2021). A comparison of Mini-FLOTAC and McMaster techniques in detecting gastrointestinal parasites in West Africa Dwarf sheep and goats and crossbreed rabbits. Journal of Applied Animal Research.

Anantrasirichai, N., Chalidabhongse, T. H., Palasuwan, D., Naruenatthanaset, K., Kobchaisawat, T., Nunthanasup, N., Boonpeng, K., Ma, X., & Achim, A. (2022, October 17). *ICIP 2022 challenge on parasitic egg detection and classification in microscopic images: Dataset, methods and results*. arXiv.org. https://arxiv.org/abs/2208.06063

Bosco, A., Rinaldi, L., Maurelli, M., Musella, V., & others. (2014). The comparison of FLOTAC, FECPAK and McMaster techniques for nematode egg counts in cattle. Acta Parasitologica.

Cringoli, G. (2004). *FLOTAC, a novel apparatus for a multivalent faecal egg count technique.* Parassitologia, 46(4), 381-384.

Gasbarre L. C. (2014). Anthelmintic resistance in cattle nematodes in the US. *Veterinary parasitology*, *204*(1-2), 3–11. https://doi.org/10.1016/j.vetpar.2014.03.017

Gordon, H.M.L., & Whitlock, H.V. (1939). A new technique for counting nematode eggs in sheep faeces. Journal of the Council for Scientific and Industrial Research, 12(1), 50-52.

International Wool Textile Organisation. (2021). World Sheep Numbers & Wool Production. https://iwto.org/wp-content/uploads/2022/04/IWTO-Market-Information-Sample-Edition-17.pdf

Kaplan, R. M., & Vidyashankar, A. N. (2012). An inconvenient truth: global worming and anthelmintic resistance. Veterinary parasitology, 186(1-2), 70–78. https://doi.org/10.1016/j.vetpar.2011.11.048

Khanna, V., Tilak, K., Rasheed, S., & Mukhopadhyay, C. (2014). Identification and preservation of intestinal parasites using methylene blue-glycerol mount: a new approach to stool microscopy. Journal of parasitology research, 2014, 672018. https://doi.org/10.1155/2014/672018

Nielsen, M. K., et al. (2021). The effect of analyst training on fecal egg counting variability. Parasitology Research.

Scare, J. A., et al. (2017). Evaluation of accuracy and precision of a smartphone based automated parasite egg counting system in comparison to the McMaster and Mini-FLOTAC methods. Veterinary Parasitology, 247, 85-92. https://doi.org/10.1016/j.vetpar.2017.10.005

SCOPS, Sustainable Control of Parasites in Sheep. (2022, September). Faecal Egg Count (FEC) monitoring. https://www.scops.org.uk/workspace/pdfs/2-7-2-faecal-egg-counts.pdf

Slusarewicz, P., et al. (2016). Automated parasite faecal egg counting using fluorescence labeling, smartphone image capture and computational image analysis. International Journal for Parasitology, 46(8). https://doi.org/10.1016/j.ijpara.2016.02.004

Slusawarewicz, P., & Hauck, E. W. (2017, February 22). Method for the Quantification of Parasite Eggs in Feces. https://worldwide.espacenet.com/patent/search?q=pn%3DEP3131655A1

Stromberg, B. E., Gasbarre, L. C., Ballweber, L. R., Dargatz, D. A., Rodriguez, J. M., Kopral, C. A., & Zarlenga, D. S. (2015). Prevalence of internal parasites in beef cows in the United States: Results of the National Animal Health Monitoring System's (NAHMS) beef

study, 2007-2008. Canadian journal of veterinary research, 79(4), 290–295. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4581673/

Suwannaphong, T., Chavana, S., Tongsom, S., Palasuwan, D., Chalidabhongse, T. H., & Anantrasirichai, N. (2023). Parasitic egg detection and classification in low-cost microscopic images using transfer learning. *SN Computer Science*, *5*(1). https://doi.org/10.1007/s42979-023-02406-8

USDA Foreign Agricultural Service. (2024, February 15). Livestock and poultry: World markets and trade. https://fas.usda.gov/data/livestock-and-poultry-world-markets-and-trade

Das, G., Klauser, S., Stehr, M., Tuchscherer, A., Metges, C. (2020). Accuracy and precision of McMaster and Mini-FLOTAC egg counting techniques using egg-spiked faeces of chickens and two different flotation fluids, *Veterinary Parasitology, Volume 283*, https://doi.org/10.1016/j.vetpar.2020.109158.

Appendix A. Client Interviews

Notes for Client Interview

Context

- Client runs fiber farming
  - Barber Pole Worms --> deadly parasites
    - Causes haemonchosis --> severe anemia and death
    - Need to use wormers to reduce population in animal
  - Other worms are ignored unless animal fails to thrive or during breeding

Test currently used

- Fecal egg count
  - Quantitative method that uses fecal material to determine number of eggs per gram of strongylid eggs including barber pole worm
- Modified McMaster Test
  - Currently: 100X Microscope (client considers buying one) --> scan individually
  - Flotation test using egg density --> eggs float to surface of counting chamber
  - Single strongylid egg shown --> parasite eggs are oval, darker in middle, lighter on edges
  - Air bubbles are round, dark on edges
  - Computer Vision to remove these dark circles
  - Count eggs in each chamber, add results, multiply result by 50 and it gives eggs per grams of feces

Our task

- Build app where client submit photo from McMaster slides
- Have app count number of eggs
- No microscope should be needed
- Only requires McMaster slide and solution

Constraints

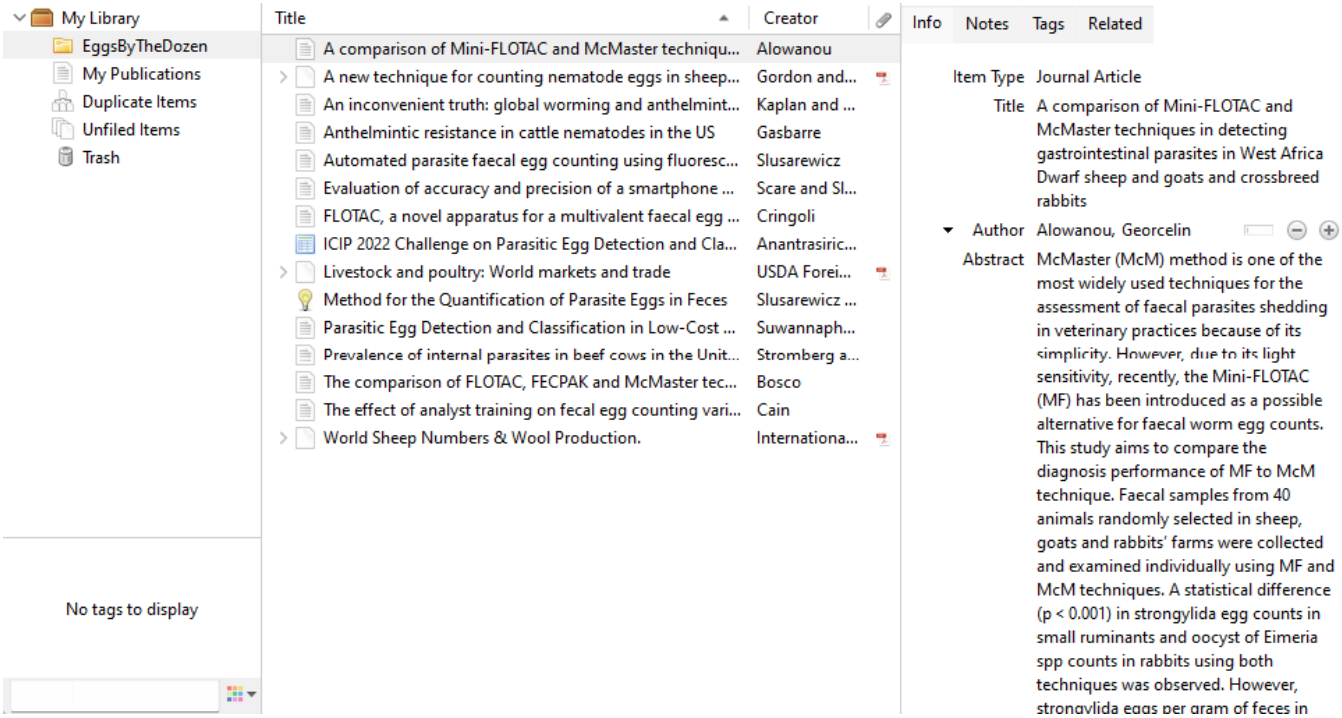
- **Time constraint** – 2 hours (how much it takes currently)
- **Budget constraint** – less than $16 per test
- **Accuracy constraint** - equal to or better than McMaster test through lab (missing 1 or 2 is fine --> but more is dangerous due to multiplication by 50)
- **Internet constraint** - best if app runs natively, instead of relying on internet
- **Client has iPhone 6, old MacBook**

Additional Information

- To get image data, image is divided into rows and columns --> go down each row in column for all columns
- Although barber pole worms are the most dangerous, fecal egg count should capture all types of round worm eggs (all look similar)
- Good to expand to more than just goats, horses, sheep, and cattle
- Even better to get data for 2 slides and average result

Appendix B. Bibliography

APPENDIX B. BIBLIOGRAPHY



Appendix C. Raw Code Reference

Figure 1: Raw Code for Image File Size Limit Middleware in ExpressJS

```
// Handles Error From Multer - POST calls next(error) if error occurs --> this
middleware is called
app.use((error, req, res, next) => {
    // Handle Image Size Exceeds Limit Error
    if (error instanceof multer.MulterError) {
        return res.status(400).send("Image Size Exceeds 1 MB Limit. Upload a
smaller image or convert to smaller image size using sites like imresizer.com");
    }

    // Some other error
    if (error) {
        return res.status(500).send("Unknown Internal Server Error");
    }

    // No error but POST called it for different reason - next() calls next
middleware
    next();
});
```

Figure 2: Raw Code for Setting 1 MB image size limit when declaring instance of Multer middleware (for parsing file-type data) in Express.js

```
/* Multer Middleware for File Uploads */
const multer = require('multer');
const upload = multer({
    limits: { fileSize: 1 * 1024 * 1024 },  // 1 MB file size limit
    dest: 'uploads/'
});     // Instance of multer
```