University of Kentucky

## UKnowledge

2022

# USE OF IMAGE PROCESSING TECHNIQUES AND MACHINE LEARNING FOR BETTER UNDERSTANDING OF T GONDII BIOLOGY

Amer Asiri
*University of Kentucky*, aas231@uky.edu
Author ORCID Identifier:
https://orcid.org/0000-0001-6189-3920
Digital Object Identifier: https://doi.org/10.13023/etd.2022.377

Right click to open a feedback form in a new tab to let us know how this document benefits you.

USE OF IMAGE PROCESSING TECHNIQUES AND MACHINE LEARNING FOR
BETTER UNDERSTANDING OF *T GONDII* BIOLOGY

_____

THESIS
_____

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in Biomedical Engineering the
College of Engineering
at the University of Kentucky

By

Amer Asiri

Lexington, Kentucky

Director: Dr. Abhijit Patwardhan, Professor of Biomedical Engineering

Lexington, Kentucky

2022

ABSTRACT OF THESIS

USE OF IMAGE PROCESSING TECHNIQUES AND MACHINE LEARNING FOR
BETTER UNDERSTANDING OF *T. GONDII* BIOLOGY

Almost one in every three people worldwide is infected with *Toxoplasma gondii* (*T. gondii*). The biology and growth of the parasite's bradyzoite form in host tissue cysts are not well understood. *T. gondii's* metabolic state influences the morphology of its single mitochondrion, which can be visualized using fluorescence microscopy with specific dyes. Hence, fluorescence microscopy images of cysts purified from infected mouse brains carry biological information about bradyzoites, the poorly understood form of the parasite within them. With the help of fluorescence microscopy techniques, previous studies extracted images of the mitochondrion, nucleus, and the inner membrane complex (IMC) providing information on *T. gondii's* cysts paving the way for image processing techniques and machine learning to analyze the bradyzoite form of the parasite. Previously, multivariate logistic regression (MLG) was used to classify shapes of mitochondrion. In the present study, in addition to the previously used MLG model, two other machine learning models, Support Vector Machine (SVM) and K Nearest Neighbors (KNN), were used to explore the possibility of better model selection for mitochondrial classification. A minimal model error was used to optimize the classification model performance. Error in any machine learning model is driven by bias, variance, and noise. Through trial and error, the optimal hyperparameters for each model were selected to minimize error. The dataset used consisted of 1940 labeled mitochondrial objects with 22 features, and consisted of five classes Blob, Tadpole, Donut, Arc, and Other. 50% of the dataset was used for training, and the other 50% was used for testing. The overall models' accuracy of MLG, SVM, and KNN were 79.1%, 78.9%, and 80.3% respectively. Overall classification performance did not vary, but the F score for some classes like Tadpole and Donut showed improvement when using the two newer models. One of the 22 features used was an application of the Histogram of Oriented Gradients (HOG). The HOG feature was replaced with a novel feature that used linear regression of object boundary segments to extract the HOG for only the object's boundary. The model that used Boundary HOG showed some improvement over the HOG feature. Finally, a new module including a graphical user interface was developed to process and extract shape and intensity information from TgIMC3 images which facilitate further investigations of the parasite biology.

KEYWORDS: *T. gondii* Biology, Machine Learning, Model Selection, Histogram of Oriented Gradients

Amer Asiri

09/23/2022

USE OF IMAGE PROCESSING TECHNIQUES AND MACHINE LEARNING FOR
BETTER UNDERSTANDING OF *T GONDII* BIOLOGY


By
Amer Asiri


<table>
<tr><td>Abhijit Patwardhan</td></tr>
<tr><td>Director of Thesis</td></tr>
<tr><td></td></tr>
<tr><td>Sridhar Sunderam</td></tr>
<tr><td>Director of Graduate Studies</td></tr>
<tr><td></td></tr>
<tr><td>09/23/2022</td></tr>
<tr><td>Date</td></tr>
</table>

ACKNOWLEDGMENTS

TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

CHAPTER 1. INTRODUCTION

The focus of the work in this thesis is to utilize fluorescence microscopy images of cysts harvested from mice infected with *Toxoplasma gondii* (*T. gondii*), image processing techniques, and machine learning methods to further the understanding of *Toxoplasma gondii* biology, of a form that is poorly understood.

*T. gondii* is an Apicomplexan parasite discovered by Nicolle and Manceaux in 1908 [1]. Since then, a large number of researchers have investigated *T. gondii* due to its prevalence and severity, especially when the immune system is weakened [1]. Almost a third of the world's population is infected with some type of the parasite [1]. Toxoplasmosis is a disease resulting from the infection by the protozoan parasite *T. gondii*. In infected individuals, after a period of rapid growth, the parasite triggers the immune system which kills most but not all parasites. This is so because parasite avoids the immune system by entering a slow-growing stage termed a bradyzoite within structures called tissue cysts [2]. Tissue cysts form in in different body tissues (e.g., brain, heart, and muscle tissue). Fortunately, the immune system keeps the parasite under control, but does not eliminate the cyst stages. When the immune system is weakened, the previously dormant cyst form of the parasite can reactivate causing life threatening infections. For example, toxoplasmosis in the brain can lead to seizures and fatal illnesses like encephalitis [3]. There are pronounced reasons for more research on *T. gondii*, particularly with regard to the poorly understood cyst stages, because of the potential danger when the immune system is weakened, given the longevity of the parasite inside the hosts' tissue, and its ubiquity.

*T. gondii* can infect cells in all three stages of its life cycle which include an acute phase, a slow growing phase and a sexual phase that is restricted to cats. Tachyzoite is the fast-growing cycle of the parasite and is believed to be the cause of the clinical disease [2]. Tachyzoites can differentiate into bradyzoites, another phase of *T. gondii's* life cycle, and are identified as the slow-growing form [2]. Unlike tachyzoites, bradyzoites have unique properties that prevent the immune system from eliminating them [4]. Consequently, bradyzoites can live in host tissue for prolonged times. Bradyzoites' metabolic activity is potentially reflected in the shape and size of its single mitochondrion, but there is little direct evidence of how bradyzoites' mitochondrial morphology and its relationship to metabolic state are connected. Thus, a knowledge gap necessitates research on bradyzoites' cellular activity. Such information is vital to the development of new drugs as current approved treatments do not kill this stage of the parasite.

The bradyzoite form can remain within an intracellular cyst or reactivate by conversion into tachyzoites. These cysts' tendency to erupt on a regular basis is believed to be the source of the long-term immunity against Toxoplasma. In the case of immune deficit and the instance of AIDS, the bradyzoites released after a burst of a cyst are transformed into tachyzoites, which due to the absence of immunity present a life-threatening condition [3]. Due to the weakened immune system, the proliferation of tachyzoites is not regulated by the immune response, and the parasite causes serious damage to the host's brain and other tissue sites. [2,6]. Though long viewed to be metabolically dormant, earlier studies established that the slow-growing bradyzoites in tissue cysts had a remarkable amount of metabolic activity in vivo, namely mitochondrial function and the ability to multiply [7]. Notably both mitochondrial function and

replication potential were found to be heterogeneous within a cyst suggesting the cyst represents a dynamic population of bradyzoites.

Imaging techniques that quantify mitochondrial morphological information are necessary for understanding the metabolic activity of the *T. gondii* parasite. The parasite's metabolic activity can be studied using fluorescence microscopy imaging targeting the parasite's actively respiring mitochondria, as the shape of mitochondria can be linked to certain physiological states [8,9]. The electrochemical membrane potential drives the fluorescent MitoTracker dye buildup in the actively respiring mitochondria [10]. This dye is trapped in the mitochondrion and can be observed using fluorescence microscopy. MitoTracker assisted previous research in showing that the single parasite mitochondrion fragments quickly and selectively, causing an irreversibly parasitic death [8]. Mitophagy, autophagy-guided mitochondrial degradation, is the primary mechanism of programmed cell death in *T. gondii* due to the lack of confirmed apoptotic events [8].

Targeted entities, such as the respiring/active mitochondria marked with MitoTracker dye, corresponding nuclei marked with DNA dye 4',6-Diamidino-2-phenylindole DAPI, and associated parasite's inner membrane complex marked with TgIMC3 can be observed in fluorescence microscopic images through illumination in an optical slice of tissue. DAPI (4',6-diamidino-2-phenylindole) is a fluorescent blue dye that is commonly used to stain DNA in cells. In toxoplasma studies, DAPI is used to stain the DNA of the parasite so that it can be seen under a microscope, which allows researchers to visualize the nucleus of parasite.

TgIMC3 (*T. gondii* inner membrane complex protein 3) was essential in studying the asexual reproductive activity of the parasite as it is present in higher levels in

developing parasites and those that are recently "born" [7]. In contrast, parasites that have not divided for an extended period of time contain detectably less TgIMC3 allowing for the discrimination of parasites based on when they were "born". The inner membrane complex is the name given to the cortical membrane complex, which is made up of flattened alveolar sacs supported by microtubules and intermediate filament network [11]. The aggregation of this cortical cytoskeleton is responsible for nonsexual phases of cell division [11].

Proteins that resemble intermediate filaments and have an alveolin repeat made up of valine- and proline-rich regions are referred to as IMC proteins [12]. Toxoplasma produces a family of 14 IMC proteins with an alveolin domain that are sequentially put together to form the cytoskeleton of budding daughter tachyzoites. These proteins serve as markers for various developmental stages during the budding process, and TgIMC3 is highly prevalent in daughter buds during asexual division [13].

Bradyzoites have not been a subject of research from the angle of mitochondrial morphology dynamics. As most active infections in immunocompromised individuals are caused by the reactivation of a prior chronic infection, studying bradyzoite mitochondrial function in depth can assist in reducing the current knowledge gap on the parasite's clinically important life cycles that can last in host's tissue for decades [14]. The importance of this research is further reinforced by the fact that the only medications thought to be effective in treating this difficult-to-cure phase are thought to target the parasite's mitochondrion [15, 16].

Automated classification of mitochondrial morphologies can be made possible using machine learning algorithms that learn from experts' labeling of the known

morphologies. The labeled data is used to improve the algorithm's classification accuracy by a step called as training. The use of labeled data for training models is termed as supervised machine learning. A performance comparison between different morphology classification techniques is essential for determining which machine learning algorithm is best suited for a particular application. In this thesis, a performance comparison between three different machine learning models; logistic regression, support vector machine, and K-nearest neighbor algorithm was conducted.

To calculate the number of parasites within cysts during the bradyzoite form of toxoplasmosis infection our lab previously developed image processing techniques to quantify nuclei using DAPI stained microscopy images [7]. The research revealed that bradyzoites within cysts exhibited a considerable degree of varied activity through time, opposing the notion that bradyzoites are mostly inactive [7]. Prior to this study, the number and size of tissue cysts has been adopted as the defining parameters in all functional investigations of the parasite, including those evaluating pharmacological efficacy [7]. Another conclusion of the study was that equally sized cysts showed up to two folds difference in packing density of bradyzoite counts [14].

This research revealed that the level of metabolic activity of bradyzoites in vivo was far higher than previously assumed. This included the ability to multiply within the tissue cyst, underscoring the need to reassess long-held assumptions about bradyzoite biology [7]. The hundreds of bradyzoites in each cyst are potentially great for studying and acquiring a variety of useful information about bradyzoites biology. One of these pieces of information is mitochondrial function, which is informed by its morphology, i.e. shape. However, it's tedious to manually explore each mitochondrial object to determine its shape and there is a need for an automated process.

Machine learning is an important tool to process large data related to *T. gondii*. Many recent studies related to bradyzoites and *T. gondii* data used machine learning algorithms to enhance the analysis of the parasite. ApiPredictor UniQE V2.0 is a software that uses the support vector machine learning model to predict adhesion proteins from amino acid sequences [17]. The program was used to identify adhesins in *T. gondii's*

membrane [18]. AlphaFold2 is another machine learning software used to analyze protein structures that was utilized in researching *T. gondii* proteins [18]. As an example, this software was used to predict the folding of a *T. gondii* carbohydrate-binding protein, that is more understood in other organisms, to help develop therapeutic inhibitors [19]. The use of both machine learning and fluorescence images has proved advantageous for understanding *T. gondii*. HRMAn is an example of an image analysis platform that uses machine learning to automatically identify host-pathogen interactions from immunofluorescent images of *T. gondii* parasite [20].

Using a classification criterion for Mitotracker and DAPI stained images, our lab has used image processing techniques that utilizes experts' knowledge to capture and count the images of both mitochondria and nuclei [21, 22]. Then, supervised machine learning algorithm was used to characterize the morphologies of mitochondrial objects [21, 22].

The aim of this study was to explore ways to potentially improve the classification performance of mitochondrial objects in the future through applying model selection comparisons on the current machine learning model with two other candidate models, replacing a feature with a version that is more related to shape of the outline of an object, and developing an analysis approach including a graphical user interface that facilitates extraction of parasite's relative age from images that are stained for TgIMC3 protein coregistered with those stained for nuclei.

CHAPTER 3. METHODS

Processed Images

The chronic stage of infection was assessed using *T. gondii* tissue cysts from chronically infected CBA/J mice (Jackson Laboratories) with Type II ME49 strain (AIDS Resource Center) because CBA/J mice proved to have a significant number of cysts in their brains [23]. Following infection, a Percoll gradient was used to separate the cysts from the CNS and purified, as described by Watts et al [24]. DAPI (Invitrogen), MitoTracker (Invitrogen), and TgIMC3 (detected using an antibody against this protein at 1:1,000 dilution) were used to mark the samples for the nucleus (based on DNA), actively respiring mitochondria, and inner membrane complexes.

The cysts were set in 3 percent paraformaldehyde, put on glass slides, and cover slipped before imaging [23]. A grayscale AxioCam MRM digital camera was utilized with a Zeiss Axioplan microscope that has a 100x/1.4 numerical aperture [23]. From each cyst, the single z-plain image with the largest diameter was chosen because it has the potential to reflect the greatest number of bradyzoites among all of the cyst's z-plain images. There was a possibility of overlapping objects since each z-plane image depicts a slice of the three-dimensional region of the cyst. An iterative technique was used to deconvolve the pictures (AxioVision Deconvolution Suite; Zeiss). Image processing and machine learning were performed using the deconvolved frames.

The scaled-up images had a 15.5 pixel per micron scale. The generated results were 8-bit grayscale image files in TIF format, one designated for DAPI, another for Mito, and finally TgIMC3. Figure 1 illustrates an image of DAPI, Mito, and TgIMC3 triad acquired from the same cyst.
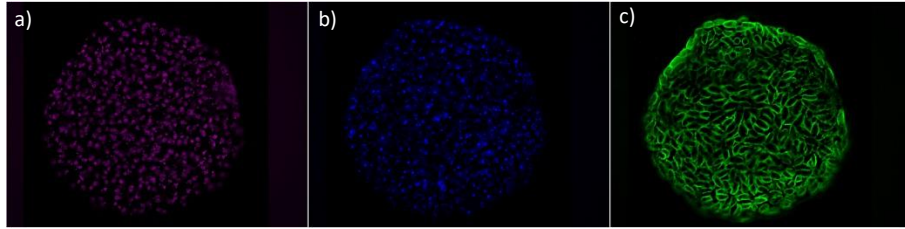
*Figure 1.  DAPI (a), Mito (b), and TgIMC3 (c) images of the same cyst.*

Since TgIMC3 stains the parasite's membranes, the imaged results will show high overlap between different parasite bodies. This issue was not notable when using images of stained *T. gondii's* nucleus and mitochondria, with DAPI and Mitotracker. Since the nucleus and mitochondrion of a parasite is positioned inside and relatively low in size, the overlap problem is not as significant as observed in images of stained membranes with TgIMC3. Therefore, an additional processing step that addresses the overlap between membranes was important for obtaining acceptable results.

Objects

The mitochondrial objects used in this study were categorized by experts into five classes. Below is a description for the distribution of objects used in comparing models' performance, after expert classification of 1940 mitochondrial objects from 10 cysts [21].

Blob
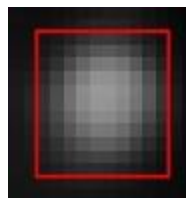


Collapsed in a round shape, with a variation in brightness and a reduced size with a diameter ranging between 0.5 and 1 micron. This mitochondrial shape is potentially caused by autophagy fragmentation. Blob count was 1249 objects, and it represented 64% of the dataset.

Tadpole
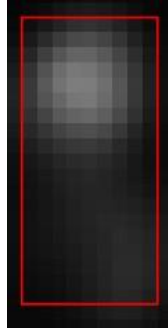


Sperm-like shape, with a bulging light end and a tail or linear string-like protrusion that is often narrower and dimmer. Tadpole count was 226 objects, and it represents 12% of the dataset.

Donut
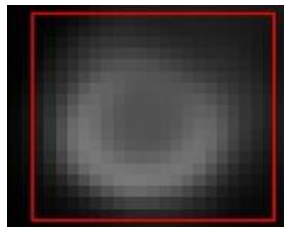


Ring or donut-shaped, somewhat larger than a blob, with a discernible decreased intensity in the middle. Considered to be the morphology of a functioning mitochondrion. Donut count was 61 objects, and it represents 3.1% of the dataset.

Arc



Typically, with arms that look like dumbbells and have a thin linkage. Not exclusively arc shaped, but also extend to linear shapes. Arc count was 349 objects, and it represents 18% of the dataset.

Other
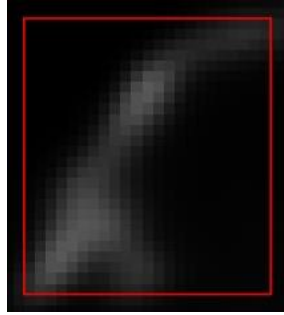


Due to proximity and/or inadequate thresholding this class represents indeterminate shapes. Usually, the shape is a combination of more than one object. The parasite's size is roughly about 7 x 3 microns, if any object's size comes closer to that size, then the object is a mixture of many mitochondria. Other count was 55 objects, and it represents 2.8% of the dataset.

Features

Since the aim of this study was to explore whether use of other machine learning approaches can improve the current model's performance, the same 22 features were used to describe each object [21, 22]. In addition, another feature, that was based on the boundary of the objects, was used to assess whether addition of this feature significantly improved the machine learning performance of the tested models. Table 3.1 describes the features used to describe each mitochondrial object. Many features, indicated with an asterisk, were obtained by using MATLAB's 'regionprops' function [25].

*Table 3.1 Description of features used to describe each object*

| 1 | Area* | Number of pixels in the object. |
|---|-------|----------------------------------|
| 2 | Boundary HOG | Number of unique gradients, in Boundary Histogram of Oriented Gradients, from eight different, equally sized, orientations that ranges from 0 to 360. The number of unique orientations present was determined to obtain a single value from these distributions. First, the eight angles of Boundary HOG were mapped to one of the eight orientations. Then, the length of the array computed by MATLAB's function 'unique' was used to obtain the number of |

| | | unique orientations. The results range from 1 to 8 with higher values denoting more circular objects. |
|---|---|---|
| 3 | Circularity* | Circularity describes how circular the object is on a scale from 0 to 1. The circularity value for a perfect circle is 1. The value of circularity is calculated as $4 \times Area \times pi / Perimeter^2$. |
| 4 | DAPI Distance | The shortest distance between any nearby DAPI object extrema* and the centroid* of the mitochondrial object. |
| 5 | DAPI Proximity | The shortest distance between any nearby DAPI object centroid* and the extrema* of the mitochondrial object. |
| 6 | Eccentricity* | Returns the eccentricity of an ellipse with the same second-moment as the object. The eccentricity of an ellipse is the ratio of the distance between its foci to the length of its major axis. The value ranges from 0 to 1. An ellipse with an eccentricity of 0 is a circle, and an ellipse with an eccentricity of 1 is a line segment. |
| 7 | Extent* | Represents the ratio of total pixels in the object to pixels in the bounding box. Calculated by dividing the area of the object over the area of the bounding box. |
| 8 | Extrema Intensity Ratio | After extracting the centroid* and orientation* of the objects the image was translated and rotated to be centered around the centroid, with the main axis aligned with the x-axis. The image is then divided in half along its major axis*, with the total of each half's intensity used to calculate the ratio of the brighter half's pixel intensities to the other half's pixel intensities. |
| 9 | HOG number of Peaks | Matlab's 'extractHOGFeatures' from the Computer Vision toolbox was used to find the histogram of oriented gradients. This method processed the image of the object like a single block and determined the magnitudes of the gradients from eight different, equally sized, orientations that ranges from 0 to 360. The number of dominant orientations present was determined to obtain a single value from these distributions. The magnitudes were first rearranged in descending order to eliminate problems caused by different object orientations. The array's point where the difference between surrounding values exceeded the standard |

| | | |
|---|---|---|
| | | deviation was then returned. Zero was returned if no value exceeded the standard deviation. This can be observed in circular objects [21]. |
| 10 | Hole Count | Extracted using Matlab's 'bwboundaries' to get the number of holes within the object. |
| 11 | Hole Size | The 'bwboundaries' function in Matlab was used to determine this feature, which is the total number of pixels that make up an object's holes or the sum of the areas of all holes discovered in the object. |
| 12 | Major Axis Length* | Calculated by finding the major axis of an ellipse with the same normalized second central moments as the object. |
| 13 | Max Intensity* | The value of the pixel with the highest intensity. |
| 14 | Mean Intensity* | Average of all the intensity values in the object. |
| 15 | Min Intensity* | The value of the pixel with the lowest intensity. |
| 16 | Minor Axis Length* | Calculated by finding the minor axis of an ellipse with the same normalized second central moments as the object. |
| 17 | Mito Proximity | The shortest distance between any nearby mitochondrial centroid's* and the extrema* of the mitochondrial object. |
| 18 | MSE Intensity | The mean squared error between the object pixel intensities and the enhanced object pixel intensities. |
| 19 | Peak Count | The number of peaks that remain after applying the watershed method (Matlab's "watershed") to suppress peaks that are higher than one standard deviation of the entire object within the bounding box (using height transform in Matlab's 'imhmin' function). |

| 20 | Perimeter* | The total number of neighboring pixels surrounding the object's border. |
|---|---|---|
| 21 | Ratio | Calculated by dividing the Major Axis Length* by the Minor Axis Length*. |
| 22 | std Intensity | The standard deviation of pixel intensity values within the object. |
| 23 | var Intensity | The variance in pixel intensity values within the object. |

Boundary HOG

Histogram of oriented gradients (HOG) is a feature descriptor algorithm commonly used in computer vision. HOG breaks down an image into a set of square cells, calculates a histogram of oriented gradients within every cell, normalizes the output with a block-wise pattern, and outputs a descriptor for each cell [26]. The typical way of computing HOG starts by extracting the horizontal and vertical gradients before calculating the HOG descriptor. Extracting gradients is readily accomplished by applying the following kernels to the image. [-1 0 1] for horizontal and [-1; 0; 1] for vertical. The orientation of each cell is computed using eq 1 [26].

$\theta = atan(\frac{Gradient\ vector\ of\ y}{Gradient\ vector\ of\ x})$ *Equation 1. Orientation calculated through arctangent of vertical gradient y over horizontal gradient x*

This allows computing the oriented gradients matrix for each block in the image, using the obtained X and Y gradient matrices. Then, normalization is done for all blocks and the results are flattened into a 1 D feature vector [26].

In this study, a more simplified approach of HOG, which only extracts angle information of boundary segments, was used. Instead of using the whole two-dimensional matrix of the object's image, only the outer boundary was extracted. Then, the locations

of the outer layer of white pixels, the boundary, was extracted from each object using Moore's Neighboring algorithm with Jacob's stopping criterion [28]. The order of boundary locations must be connected for the next step of calculating the boundary histogram of oriented gradients (Boundary HOG). After extracting the boundaries, they are segmented into 8 connected segments. Each segment has the same number of pixel locations. Then, each segment was fitted with linear regression to extract slopes.

$$Y(X) = X \cdot \beta1 + \beta0$$ *Equation 2. Linear Regression*

The slope extraction method utilizes linear regression in eq 2 where Y(X) is the predicted value for any given value of X [27]. β1 is the extracted slope, and β0 is the intercept. The following regression coefficient equation (eq 3) is used to extract the slope of each segment along the boundary.

$$\beta1 = \frac{\sum_{i=1}^{n}(Xi-x)(Yi-y)}{\sum_{i=1}^{n}(Xi-x)^2}$$ *Equation 3. Slope extraction through regression coefficient.*

*x in this equation represents mean of X values, and y here represents the mean of Y values.*

The 8 slopes were then converted into signed angles that ranges between 0 and 360 using the arctangent of the slope. The result is an array of 8 angles that describes the boundary of each object. Last, the 8 angles are represented in the histogram distribution of angles with 8 bins.

Machine Learning Models

One of the most challenging tasks in using machine learning for classification is model selection. Many researchers evaluate different models using the bias-variance trade off. Bias is a classification error that results from the difference between the model's estimated values and true values [27]. High bias is one of the main reasons of underfitting, which increases error in both training and testing. Variance is another classification error

that is caused by the increase of the average squared differences from the mean [27]. High variance is one of the main reasons of overfitting, which may cause low error in training but high-test error.

In this thesis, three models were evaluated, and comparison metrics were calculated for each model. The data were split into 50% training, 50% test. The comparison metrics used in this study were True Positive Rate (sensitivity), True Negative Rate (specificity), Precision, and F Score which is an accuracy measure that uses sensitivity and precision (eq 4-7).

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad Equation\ 4.$$

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Negative} \quad Equation\ 5.$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad Equation\ 6.$$

$$F\ Score = 2 \cdot \frac{Precision \cdot Sensitivity}{Precision + Sensitivity} \quad Equation\ 7.$$

### 3.1.1   Multivariate Logistic Regression

A logistic regression model is simply an application of the sigmoid function that is used to map linear weighted combination of feature values to probabilities between 0 and 1 using eq 9 [27].

$$g(z) = \frac{1}{1+e^{-z}} \quad Equation\ 8.\ The\ sigmoid\ function$$

$h(X) = g(Y(X))$ *Equation 9. Logistic regression is the sigmoid function applied on a linear function.*

To use regression models for classification, a certain threshold is decided which separate probabilities into two discrete classes and is called a decision boundary [27]. A

previous study used an application of logistic regression for multi-class problems called multivariate logistic regression with a one vs all methodology [21]. The multivariate logistic regression model classifies objects based on highest computed probability [27].

### 3.1.2 Support Vector Machine

Support Vector Machine (SVM) is a widely used supervised machine learning algorithm. The goal of SVM is to predict the classification of objects by finding a hyperplane that maximizes the distance between two classes in objects' hyperspace (eq.10) [27]. Feature data points of each object are mapped into objects' hyperspace and then the hyperplane with the maximum distance acts as a decision boundary [27].

$y = w \cdot f(x) + b$ *Equation 10. Support Vector Machine*

The predicted value of the SVM model is y, f(x) is the feature vector, w is the calculated feature vector weight given to f(x), and b is the model's bias. The algorithm is usually used for binary classification. However, it can work for multi-class classification by fitting one model for each pair of classes. This process is known as one vs. one [27]. Another way of using SVM binary classification for multi-class applications is by fitting one class vs the rest for all classes. This process is called one vs. rest, which is the same as one vs. all [27]. SVM uses mathematical operators, known as kernels, that transform inputs to the desired more separable form (e.g., Linear kernel) [27]. For the SVM model in this study, the chosen method was one vs. one with a linear SVM kernel computed through MATLAB [25].

Two sources of inputs are needed to train an SVM model and find the decision boundary that maximally separates the two classes with a hyperplane. With a one vs one approach, the SVM model represents multiple binary classifiers. One classifier per each

17

pair of classes. In other words, each pair of classes is separated with a hyperplane neglecting the other data points from the other three classes. Therefore, the number of classifiers used the SVM model uses are calculated through $\frac{N(N-1)}{2}$, which equals 10 binary classifiers in our situation.

Each classifier uses the hyperplane equation (eq.10) to predict any input (x0) based on the predetermined decision boundary calculated in eq 11 [27].

$$D = \frac{|w \cdot f(x0) + b|}{\|w\|_2}$$ *Equation 11. Hyperplane Distance Equation*

The Euclidian norm $\|w\|_2$ is used to calculate the length of w where $\left\|w\right\|_2 = \sqrt{w_1^2 + w_2^2 + \cdots w_n^2}$. Support vectors are the closest distance of two datapoints from a different class [27]. Using the support vectors to calculate the maximal hyperplane distance is the main principle of support vector machine classifiers. Therefore, the maximum distance between two closest points from different classes in the feature space (fig.2) is calculated through eq 12 [27].

$$w_{optimal} = \arg_w \max\left(\min(D)\right)$$ *Equation 12. Hyperplane Distance Equation*
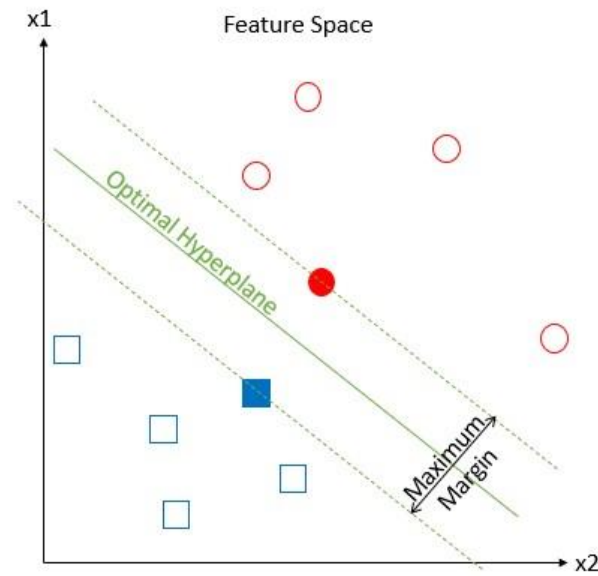
*Figure 2 An illustration of how SVM is calculated*

After defining a decision boundary, predicted values above the hyperplane will be classified as belonging to same class. Similarly, the values on the other side of the hyperplane will be classified as the other class.

### 3.1.3   K Nearest Neighbor

K Nearest Neighbor (KNN) is a supervised classification algorithm suited for multi-class classification. The algorithm classifies an object by a majority vote among its k nearest neighbors [27]. The k is a positive number that represents the number of nearest objects, from the labeled training set, in the multidimensional feature space [27]. However, the morphology data is imbalanced, and basic majority voting is biased towards more frequent classes. A way to resolve the problem of imbalanced classes is to multiply each k nearest neighbor by a weight that is proportional to the inverse distance from datapoint to be predicted. Weighted KNN favors nearby votes over farther ones.

Common methods to calculate the distance between objects in feature space are Euclidean Distance and Cosine distance [27]. In this study, Cosine Distance showed better prediction results than Euclidean Distance. Therefore, the KNN model used in the models' performance comparison is a weighted KNN model with the cosine distance method computed through MATLAB [25, 29].

IMC3 Processing Module

A program that includes a graphical user interface (GUI) that processes TgIMC3 stained images, automatically selects thresholds for binarization and segmentation, and allows users to change threshold values for refining the results. The objective of the program was to extract the intensity information of each TgIMC3-stained object as this information can be useful in determining the age of the object.
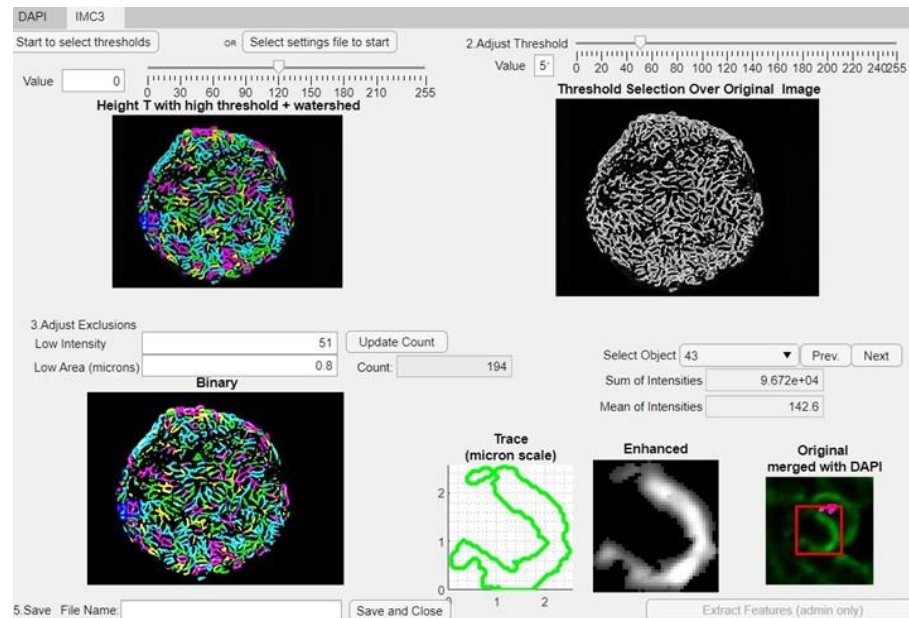


*Figure 3 The Graphical User Interface developed for the IMC3 module used to process TgIMC3 images.*

Figure 3 shows a screenshot of the GUI developed for IMC3 module. The program can be started by selecting the Mito image, or by selecting a previously saved setting file.

The program will import Mito, DAPI and TgIMC3 images and process them accordingly. Once the image is selected, the program applied processing techniques similar to those used previously [21, 22]. Briefly: first, the greyscale TgIMC3 image was enhanced through top hat and bottom hat filters. The filters' erosion and dilation steps used a disc structuring element [21, 22]. After applying the filters and histogram equalization, the enhanced TgIMC3 image was binarized with a specific threshold that removed the background and converted it to zero intensity pixels. Then, the foreground boundaries were extracted by using MATLAB's function 'bwboundaries' [25].

Then, the GUI will show the processed image with the selected binarization threshold and a white outline for the boundaries on the top right. When the users are satisfied with the binarization results, they can shift their attention to the top left image that shows the segmentation results. If the users are not satisfied with the segmentation results, they can make the segmentation more coarse or fine. MATLAB app designer was used to design the graphical user interface and code the module [25].

Boundary HOG



(a)                                             (b)
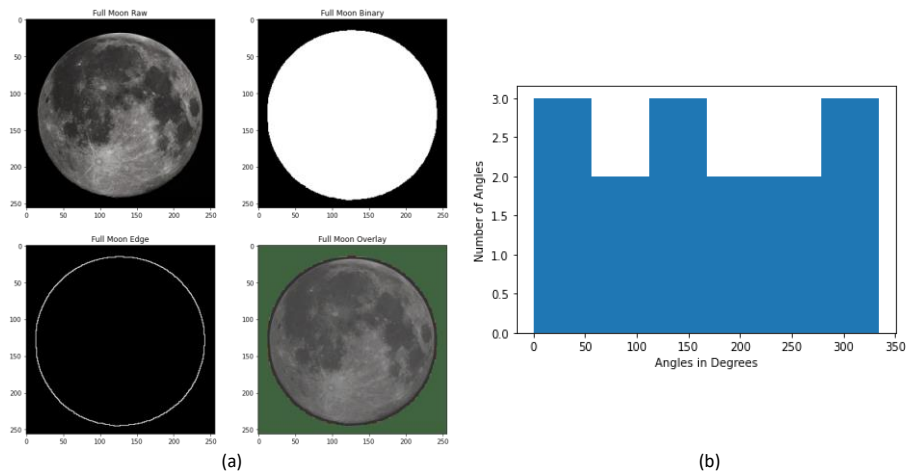
*Figure 4 Extracting the boundary of a Full Moon image by thresholding for binarization, followed by applying morphological operations to find the edge (a). Boundary Histogram of Oriented Gradients of the image (b).*



(a)                                             (b)
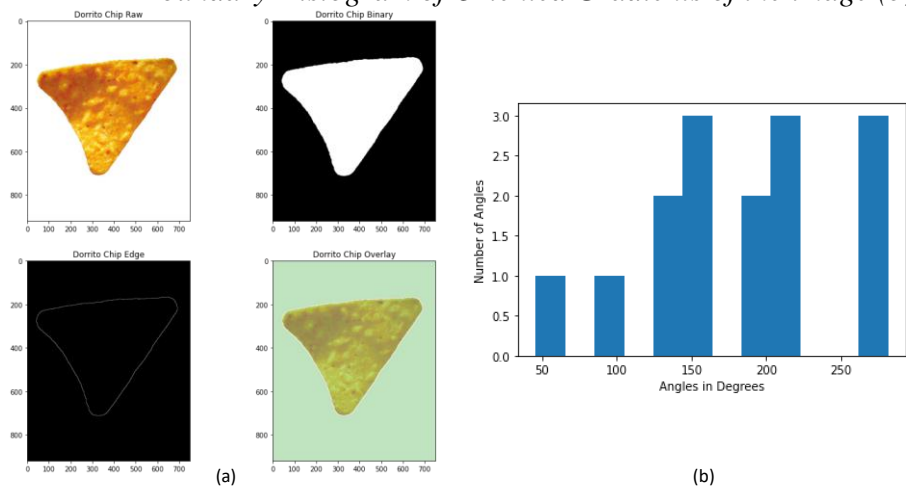
*Figure 5 Extracting the boundary of a Doritos Chip image by thresholding for binarization, followed by applying morphological operations to find the edge (a). Boundary Histogram of Oriented Gradients of the image (b).*

Before adding Boundary HOG to the mitochondrial classification model, many

iterations of tests were used to assess the feature. One iteration included using two simple

singular objects that possess different shapes. In figure 4, a full moon image was used to

assess circular shapes. In figure 5, a Doritos chip was used to assess triangular objects. In

order to extract boundary information of each image, they were converted to grayscale.

22

After extracting the grayscale of the images, an appropriate threshold was used to separate background from foreground. The result is a binarized version of the images with some dark holes in the white foreground. The holes were filled using basic morphological operations such as erosion and dilation. This resulted in a clean binary representation of the objects' shape depicted in figures 4 and 5. Morphological operations were applied again to extract the object's boundary. First, a copy of the binarized image was saved. Then, one layer of erosion was applied on that copy. The eroded copy of the image was then subtracted from the clean binarized image. The result is a binary image of the objects' boundary as shown in figures 4 and 5.

Since the boundaries were in a binary two-dimensional image format, the only steps left were to convert it into a one-dimensional array of neighboring points and to convert them to a histogram of oriented gradients. First, the white pixels' locations in the boundary images were extracted in an array. Then, the array was sorted based on location using Moore's neighboring algorithm. This step is essential for segmenting the boundary and applying linear regression. After sorting the boundary locations array, segmentation into ordered and equally sized arrays was used to convert the pixels' locations into slopes. Using linear regression, each segmented array was transformed into a slope (from multiple locations array into a numeric variable). The slopes were then converted into angles between 0 and 360 using the arctangent function described in the methods section. The resulting angles are plotted in histograms of oriented gradients depicted in figures 4 and 5.

The resulting boundary histogram of oriented gradients for a full moon shows a uniform distribution of angles. Circular objects should show uniformity in their boundary angles. On the other hand, the boundary histogram of oriented gradients for the Doritos

chip showed three main clusters of angles, and two singular angles out of these clusters. Triangular objects should only show 3 clusters of angles in their histogram of oriented gradients. However, the left side of the Dorito chip's triangle is curved. This curvature was captured by the Boundary HOG method, which attests to the precision of this approach. Therefore, Boundary HOG was tested as a feature in the models used in this study.

*Table 4.1.1 Model with HOG feature Performance.*

|  | Blob | Tadpole | Donut | Arc | Other |
|---|---|---|---|---|---|
| Sensitivity | 0.952 | 0.359 | 0.2 | 0.639 | 0.242 |
| Specificity | 0.712 | 0.964 | 0.996 | 0.902 | 0.992 |
| Precision | 0.836 | 0.605 | 0.6 | 0.616 | 0.5 |
| F Score | 0.89 | 0.451 | 0.3 | 0.627 | 0.327 |

*Table 4.1.2 Model with Boundary HOG instead of HOG feature Performance.*

|  | Blob | Tadpole | Donut | Arc | Other |
|---|---|---|---|---|---|
| Sensitivity | 0.959 | 0.305 | 0.2 | 0.644 | 0.273 |
| Specificity | 0.72 | 0.963 | 0.997 | 0.899 | 0.99 |
| Precision | 0.841 | 0.557 | 0.667 | 0.609 | 0.5 |
| F Score | 0.896 | 0.394 | 0.308 | 0.626 | 0.353 |

Table 4.1.1 shows the performance of the 22 previous features trained and tested using MLG model. Table 4.1.2 shows the result of the model when the HOG feature is replaced with Boundary HOG. Since the only difference between the two models is changing one feature out of 22 features, there was no expectation of remarkable difference in the results. The overall performance accuracy was very similar, and slight gains and losses were observed. Contrary to what's expected though, Blob and Donut classification performance did not decrease when using boundary HOG instead of HOG. Slight drops in

classification performances of Blob and Donut classes were expected because of the similarity between the two classes when we only consider the objects' boundaries. Nonetheless, the classification performance of Blob and Donut objects slightly increased.

Machine Learning Models Comparison

*Table 4.2.1 Classification results of the Support Vector Machine (SVM) Model.*

|  | Blob | Tadpole | Donut | Arc | Other |
|---|---|---|---|---|---|
| Sensitivity | 0.941 | 0.405 | 0.312 | 0.705 | 0.062 |
| Specificity | 0.784 | 0.969 | 0.987 | 0.891 | 0.996 |
| Precision | 0.813 | 0.93 | 0.959 | 0.866 | 0.942 |
| F Score | 0.872 | 0.564 | 0.471 | 0.777 | 0.116 |

According to Table 4.2.1, the classification sensitivity of the SVM model was most sensitive to classifying Blob over any other class with a true positive rate (TPR) of 94.1%. The Arc class came second with a TPR of 70.5%. The model was not very sensitive to Tadpole, Donut, and "Other" classes. However, classes with low TPR had very high specificity score. This means that if the model classified an object among the low TPR classes, the classification accuracy was greater than 93%. Blob and Arc had a specificity score of 78.4% and 89.1% respectively. Blob had the highest sensitivity and lowest specificity score. However, most Blob were classified as Blob correctly in the SVM model. The reason for the drop in specificity score and precision of Blob is that low TPR classes were misclassified as Blob. The quality of the SVM classifier was considered very good since most classes have higher than 90% precision score, and no class dropped below 80% in precision score. Therefore, the F score was higher than the quantitative performance of all classes, except for Blob which decreases slightly.

*Table 4.2.2 Classification results of the K Nearest Neighbor (KNN) Model.*

|  | Blob | Tadpole | Donut | Arc | Other |
|---|---|---|---|---|---|
| Sensitivity | 0.936 | 0.473 | 0.625 | 0.668 | 0.25 |
| Specificity | 0.794 | 0.962 | 0.985 | 0.918 | 0.992 |
| Precision | 0.82 | 0.926 | 0.976 | 0.89 | 0.97 |
| F Score | 0.874 | 0.626 | 0.762 | 0.763 | 0.398 |

The overall accuracy of the KNN model was the highest among the three models. KNN models does not perform well in unbalanced datasets but using a weighted KNN solved that issue and this turned out to be a good model choice. Table 4.2.2 shows that the KNN model was more sensitive to Blob, Donut, and Arc classes than to Tadpole and Other. The Specificity of each class were above 90% except for Blob with a 79.4% specificity score. Since the unbalanced dataset is biased toward Blob, it was expected that more objects will be classified as blobs hence the lower specificity and precision scores for blobs. Nonetheless, the precision scores for all classes reflects the quality of the KNN model. Therefore, like SVM, the F scores of all classes in the KNN model were better than the sensitivity, except for Blob, due to increase in the high precision score.

*Table 4.2.3 Classification results of the Multivariate Logistic Regression (MLG) Model.*

|  | Blob | Tadpole | Donut | Arc | Other |
|---|---|---|---|---|---|
| Sensitivity | 0.939 | 0.398 | 0.473 | 0.684 | 0.178 |
| Specificity | 0.802 | 0.967 | 0.986 | 0.893 | 0.994 |
| Precision | 0.892 | 0.622 | 0.519 | 0.583 | 0.576 |
| F Score | 0.915 | 0.484 | 0.479 | 0.63 | 0.257 |

The MLG model results in Table 4.2.3 showed similar overall performance but was lower in precision scores compared to SVM and KNN models. The specificity score of

Blob classification is negligibly higher than the two models, and there was no notable difference in the other classes specificity scores. The precision scores for MLG were the lowest. All classes were below 90% in precision, and most were around 60%, which lowered the models' F scores.

Table 4.2.4 Overall Models' Performance %.

| SVM | KNN | MLG |
|---|---|---|
| 78.9% | 80.3% | 79.1% |

Overall models' performance was calculated through dividing the number of correctly classified objects by the number of all classified objects for each model. Table 4.2.4 shows that the three models did not vary materially in their overall performance. All models had an overall accuracy around 80%. All models showed high precision score except for MLG. While the three models did not materially differ in sensitivity, the quality of SVM and KNN models were higher than MLG. All models performed poorly in classifying Other objects. Moreover, the KNN model was better than SVM and MLG in classifying Donut objects.

IMC3 Processing Module
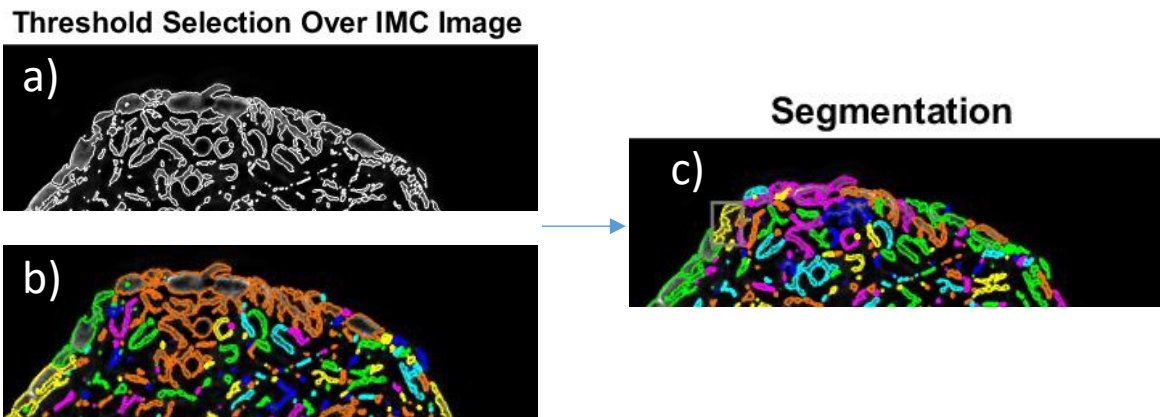
*Figure 6* Example showing *cropped images of the processed TgIMC3 image of segmentation results after threshold selection is selected (a), colored (b), and after applying Height transform for further segmentation (c).*

Using MATLAB's 'imhmin' function enabled a manual H-minima transform for segmenting the processed TgIMC3 image and yielded satisfactory segmentation results. Each colored object in Figure 6 represents an TgIMC3 segmented object. The smaller objects are processed as noise and were removed from the extracted results.

The segmented objects intensity metrics can provide valuable information about the age of the parasite as this protein's intensity is inversely related to the recency i.e., when the parasite is born. Building a graphical user interface and subsequent processing to quantify object's intensities from TgIMC3 images was the first step implemented in this module.
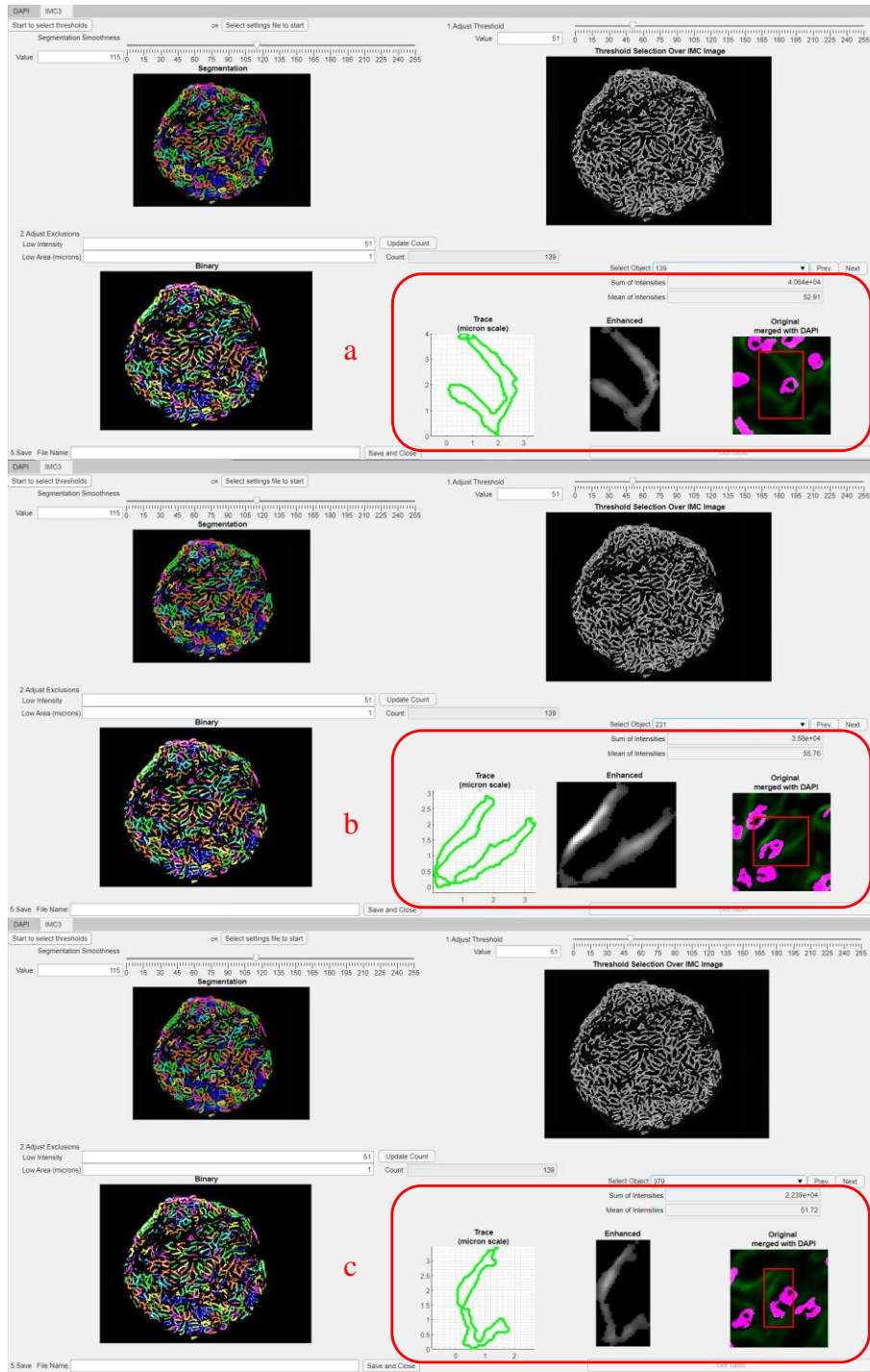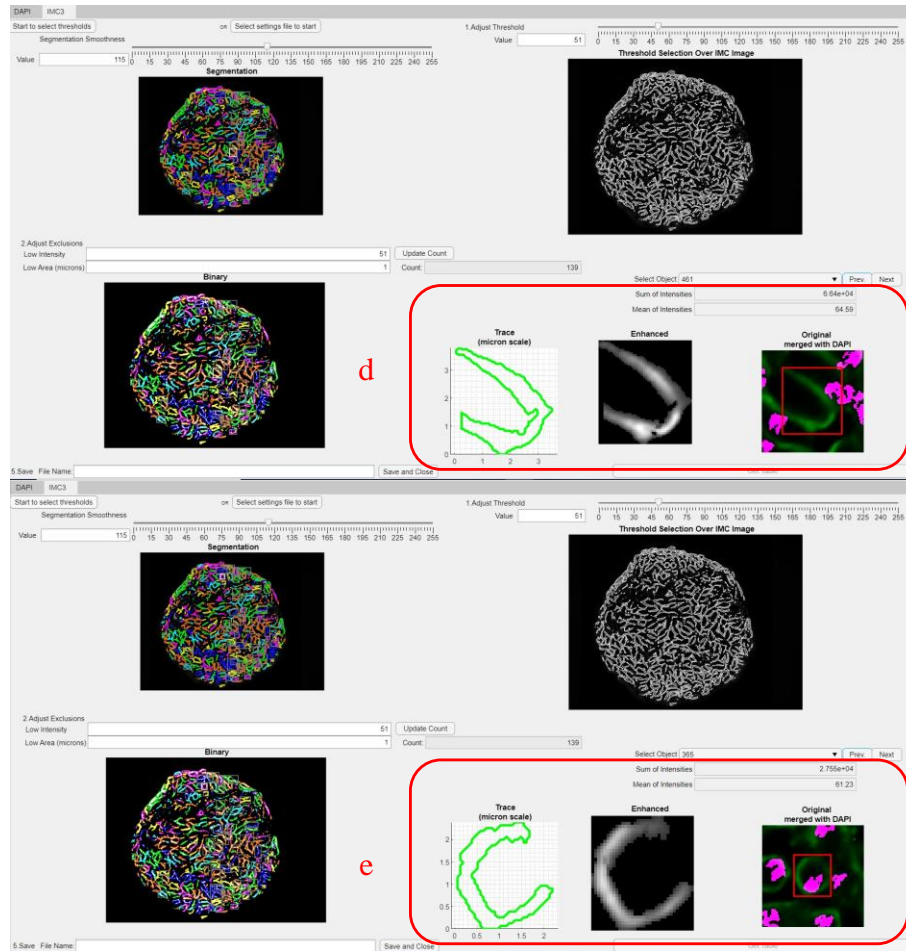
*Figure 7 a-e. Examples of isolated TgIMC3 objects in the IMC3 tab of the GUI developed to process TgIMC3 images.*

The extracted information of each TgIMC3 object can be visualized using the graphical user interface as depicted in the examples in Figure 7 a-e. From a drop-down list, the user can navigate to each TgIMC3 object in the cyst. The user can also go through images in a sequence going from left to right. Below the drop-down list is a numeric indicator that shows the sum of intensities in that object. The sum of intensities was extracted from the original TgIMC3 image using the mask (green trace).

Another numeric indicator in TgIMC3 GUI is the mean of intensities. Since the sum of intensities are heavily influenced by the TgIMC3 object mask size, the mean of

intensities can also be used to estimate the TgIMC3 object's age. Below the intensity information indicators are three images with each representing an aspect of the TgIMC3 object. The left image in the red box of figures 7a-e represents the traced boundary for the object with size indicators. In the middle, an enhanced version of the object to visualize the object. The image in the right side of the red box shows the TgIMC3 object location in the cyst overlayed with DAPI objects.

CHAPTER 5. DISCUSSION

Boundary HOG usefulness is not limited to the approach used in this study. This study focused on the number of unique angles in the histogram only. When Boundary HOG was used for classification of popular open-source datasets (results not included), the results showed notable accuracy without using other descriptive features. Even when the HOG number of peaks feature was expected to perform better, unique Boundary HOG approach was better. Boundary HOG information neglects the object's texture information but was better in classifying Blob and Donut objects, which have similar boundary information. The other 21 features, like hole or no hole and intensity metrics, must have played a role in improving the classification performance of Boundary HOG model.

The current Machine Learning Model, MLG, does provide the required performance, but the exploration of different ML models was helpful to evaluate the current model and possibly use a more robust ML algorithm in the future once there are significant improvements in performance and explainability using a different ML model. Since the results showed no notable differences between the three models, then we can conclude that the current model is sufficient for the study purposes. Furthermore, using the other two models when desired will most likely yield similar results, and hence moving forward using any ML model among the three is acceptable in similar future studies. Even though the classification models showed similar overall prediction accuracy, the prediction accuracies of each class were notably different.

It is important to note that in this batch of mitochondrial objects, only 55 objects out of 1940 objects were in the "Other" class. And since the dataset were divided in half, this must have skewed the results towards the more frequent objects. Although, the Donut

class had similar number of objects, all Donut objects shared similar features which contributed to better classification results for Donut. Based on the provided results, we can conclude that two important factors contributed to the classification accuracy of each class. The first factor is the similarity of features among the class's objects. The second factor is the frequency of objects in the training dataset.

The simplicity of KNN, by finding the 10 nearest objects in the model, might have played a role in the superior classification of Donut objects. The Donut class was only 3.14% of the dataset. Without using weights on KNN, the Donut class would have been neglected and would have shown lower model performance results. The use of weighted KNN showed better classification performance with unbalanced datasets, especially in classifying lower represented classes such as Donut and Other which made up 3.14% and 2.84%, respectively, of the dataset.

The best approach to improve the classification performance is through tackling the error problem resulting from noise, bias, and variance simultaneously. For example, reducing bias through using balanced training dataset introduce noise and variance that yields a worse model. The best noise reduction technique is choosing the right feature sets. The current model has 22 features, which means that the bias problem is being considered, but the noise problem has not been addressed. Therefore, there is a need for a study that measures which features contribute the most to the models' performance. Adding more features or pruning some of the already existing features might improve the model as well.

The TgIMC3 module was developed in a user-centered approach. In the GUI development for the module, users' input highly influenced the GUI design and functionality. For example, the original plan for segmentation was to make it fully

automated. However, some segmented results were not optimal and there was a need for better segmentation. Using the segmentation window as shown in in figures 7a-e- users can adjust, i.e. "tweak" segmentation threshold and immediately visualize the segmented results. To allow better visualization of segmentation results, the segmented objects were colored with six different colors (Cyan, Yellow, Green, Orange, Blue, and Magenta) sequentially from left to right. The color choice was evaluated with users for better visualization. The end result of the module was computation of sum and average of intensities within each segmented object. The expectation was that because the intensity of each object being reflective of the extent to which the protein TgIMC3 was present in that parasite, these intensities would allow for determination of the recency, i.e., relative age of the parasites.

In summary, our study showed that although boundary HOG did not remarkably improve the prediction accuracy of the models, it could be a promising feature especially when the feature set is pruned. The comparison among the three ML models suggests that while any of the three would yield performances that would be acceptable from a biology perspective, when the data sets grow in the future, exploration of use of models other than MLG may be worthwhile. The developed module for processing of TgIMC3 images resulted in satisfactory segmentation of objects as determined by expert users and thus permits evaluation of intensity-based measurements for the imaged parasites. Further development of the module to incorporate orientations of the detected objects would be necessary to extract additional information about spatial localization of the objects within the cysts.

REFERENCES

[1] Ferguson, D.J., 2009. Toxoplasma gondii: 1908-2008, homage to Nicolle, Manceaux and Splendore. Memorias do Instituto Oswaldo Cruz, 104, pp.133-148.

[2] Pavesio, C.E.N., Chiappino, M.L., Setzer, P.Y. and Nichols, B.A., 1992. Toxoplasma gondii: differentiation and death of bradyzoites. Parasitology research, 78(1), pp.1-9.

[3] Luft, B.J. and Remington, J.S., 1992. Toxoplasmic encephalitis in AIDS. *Clinical infectious diseases*, *15*(2), pp.211-222.

[4] Sullivan Jr, W.J. and Jeffers, V., 2012. Mechanisms of Toxoplasma gondii persistence and latency. *FEMS microbiology reviews*, *36*(3), pp.717-733.

[5] Weiss, L.M. and Kim, K., 2000. The development and biology of bradyzoites of Toxoplasma gondii. *Frontiers in bioscience: a journal and virtual library*, *5*, p.D391.

[6] Subauste, C.S. and Remington, J.S., 1993. Immunity to Toxoplasma gondii. *Current opinion in Immunology*, *5*(4), pp.532-537.

[7] Watts, E., Zhao, Y., Dhara, A., Eller, B., Patwardhan, A., and Sinai, A.P.: 'Novel Approaches Reveal that Toxoplasma gondii Bradyzoites within Tissue Cysts Are Dynamic and Replicating Entities In Vivo', mBio, 2015, 6, (5), pp. e01155-01115

[8] Ghosh, D., Walton, J.L., Roepe, P.D., and Sinai, A.P.: 'Autophagy is a cell death mechanism in Toxoplasma gondii', Cell Microbiol, 2012, 14, (4), pp. 589-607 5

[9] Ovciarikova, J., Lemgruber, L., Stilger, K.L., Sullivan, W.J., and Sheiner, L.: 'Mitochondrial behaviour throughout the lytic cycle of Toxoplasma gondii', Scientific reports, 2017, 7, (1), pp. 42746

[10] Perry, S.W., Norman, J.P., Barbieri, J., Brown, E.B., and Gelbard, H.A.: 'Mitochondrial membrane potential probes and the proton gradient: a practical usage guide', Biotechniques, 2011, 50, (2), pp. 98-115

[11] Anderson-White, B., Beck, J.R., Chen, C.T., Meissner, M., Bradley, P.J. and Gubbels, M.J., 2012. Cytoskeleton assembly in Toxoplasma gondii cell division. *International review of cell and molecular biology*, *298*, pp.1-31.

[12] Gould, S.B., Tham, W.H., Cowman, A.F., McFadden, G.I. and Waller, R.F., 2008. Alveolins, a new family of cortical proteins that define the protist infrakingdom Alveolata. *Molecular biology and evolution*, *25*(6), pp.1219-1230.

[13] Anderson-White, B.R., Ivey, F.D., Cheng, K., Szatanek, T., Lorestani, A., Beckers, C.J., Ferguson, D.J., Sahoo, N. and Gubbels, M.J., 2011. A family of intermediate filament-like proteins is sequentially assembled into the cytoskeleton of Toxoplasma gondii. *Cellular microbiology*, *13*(1), pp.18-31.

[14] Sinai, A.P., Watts, E.A., Dhara, A., Murphy, R.D., Gentry, M.S., and Patwardhan, A.: 'Reexamining Chronic Toxoplasma gondii Infection: Surprising Activity for a "Dormant" Parasite', Current Clinical Microbiology Reports, 2016, 3, (4), pp. 175- 185

[15] Doggett, J.S., Nilsen, A., Forquer, I., Wegmann, K.W., Jones-Brando, L., Yolken, R.H., Bordon, C., Charman, S.A., Katneni, K., Schultz, T., Burrows, J.N., Hinrichs, D.J., Meunier, B., Carruthers, V.B., and Riscoe, M.K.: 'Endochin-like quinolones are highly efficacious against acute and latent experimental toxoplasmosis', Proc Natl Acad Sci U S A, 2012, 109, (39), pp. 15936-15941

[16] Araujo, F.G., Lin, T., and Remington, J.S.: 'The activity of atovaquone (566C80) in murine toxoplasmosis is markedly augmented when used in combination with pyrimethamine or sulfadiazine', J Infect Dis, 1993, 167, (2), pp. 494-497

[17] Valencia-Hernández, J.D., Acosta-Dávila, J.A., Arenas-García, J.C., García-López, L.L., Molina-Lara, D.A., Arenas-Soto, A.F., Eraso-Ortiz, D.A. and Gómez-Marín, J.E., 2022. Identification of Toxoplasma gondii adhesins through a machine learning approach. *Experimental Parasitology*, *238*, p.108261.

[18] Murphy, R.D., Chen, T., Lin, J., He, R., Wu, L., Pearson, C.R., Sharma, S., Vander Kooi, C.D., Sinai, A.P., Zhang, Z.Y. and Vander Kooi, C.W., 2022. The Toxoplasma glucan phosphatase TgLaforin utilizes a distinct functional mechanism that can be exploited by therapeutic inhibitors. *Journal of Biological Chemistry*, p.102089.

[19] Fisch, D., Yakimovich, A., Clough, B., Wright, J., Bunyan, M., Howell, M., Mercer, J. and Frickel, E., 2019. Defining host–pathogen interactions employing an artificial intelligence workflow. *Elife*, *8*, p.e40560.

[20] Fisch, D., Evans, R., Clough, B., Byrne, S.K., Channell, W.M., Dockterman, J. and Frickel, E.M., 2021. HRMAn 2.0: Next-generation artificial intelligence–driven analysis for broad host–pathogen interactions. Cellular Microbiology, 23(7), p.e13349.

[21] Place, B.C., Troublefield, C., Murphy, R.D., Sinai, A.P. and Patwardhan, A., 2021, November. Computer Aided Image Processing to Facilitate Determination of Congruence in Manual Classification of Mitochondrial Morphologies in Toxoplasma gondii Tissue Cysts. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 3509-3513). IEEE.

[22] Place, B., 2021. Facilitating Analysis of Toxoplasma gondii Bradyzoite Metabolic Activity via Image Processing and Multivariate Logistic Regression for High Throughput Classification of Mitochondrial Morphologies.

[23] Watson, G.F., and Davis, P.H.: 'Systematic review and meta-analysis of variation in Toxoplasma gondii cyst burden in the murine model', Exp Parasitol, 2019, 196, pp. 55-62

[24] Watts, E.A., Dhara, A., and Sinai, A.P.: 'Purification Toxoplasma gondii Tissue Cysts Using Percoll Gradients', Curr Protoc Microbiol, 2017, 45, pp. 20C 22 21-20C 22 19

[25] MATLAB, 2021b. version 9.11.0 (R2021), Natick, Massachusetts: The MathWorks Inc.

[26] Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In 2005 *IEEE computer society conference on computer vision and pattern recognition* (CVPR'05) (Vol. 1, pp. 886-893). IEEE.

[27] Rebala, G., Ravi, A., and Churiwala, S.: 'An Introduction to Machine Learning' (Cham: Springer International Publishing AG, 2019. 2019)

[28] Reddy, P.R., Amarnadh, V. and Bhaskar, M., 2012. Evaluation of stopping criterion in contour tracing algorithms. International Journal of Computer Science and Information Technologies, 3(3), pp.3888-3894.

[29] Abu Alfeilat, H.A., Hassanat, A.B., Lasassmeh, O., Tarawneh, A.S., Alhasanat, M.B., Eyal Salman, H.S. and Prasath, V.S., 2019. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. Big data, 7(4), pp.221-248.

Amer Asiri

Education:

- University of Kentucky (Lexington, KY) 2020 – 2022
  - Master of Science in Biomedical Engineering (anticipated)
- The George Washington University (Washington, DC) 2016 – 2020
  - Bachelor of Science in Biomedical Engineering, with two minors in Electrical Engineering and Psychology

Work Experience:

- Undergraduate Volunteer Research Assistant – The George Washington University (Washington, DC), Jan 2019 – May 2020
- Research Assistant – University of Kentucky (Lexington, KY) Aug 2020 – Aug 2022

Awards

- Selected Capstone Project for Inaugural University Collaboration between George Washington University's School of Engineering and Applied Sciences, and Sweden's Chalmers University of Technology for commercialization and technical prototyping.