Project 4 Final Report

Vision Guided Mosquito Dissection for the Production of Malaria Vaccine

EN 601.656 Computer Integrated Surgery II

Yutai Wang (ywang790@jhu.edu)

Mentors: Balazs Vagvolgyi

05/07/2023

Contents

1. Introduction

Sanaria Inc. has developed a viable vaccine for malaria that involves using the salivary glands dissected from mosquitoes. Automating current manual dissection processes could help Sanaria reach global-scale production-level targets. An efficient automated mosquito salivary gland extraction system requires robust, high-performance computer vision methods for robot control and quality control. We aim to use model-based (IP) and machine-learning-based (DL) computer vision methods to facilitate robotic mosquito dissection.

2. Background, Significance and Problems

- **Background**

Malaria is a serious and sometimes fatal disease caused by a parasite that commonly infects a certain type of mosquito that feeds on humans. People who get malaria are typically very sick with high fevers, shaking chills, and flu-like illnesses. About 2,000 cases of malaria are diagnosed in the United States each year. There are over 200 million cases of malaria every year globally which results in more than 400,000 deaths.[1] The disease is caused by a parasite that incubates inside the salivary glands of mosquitoes. Figure 1 shows the life cycle of plasmodium falciparum.[2] Extracting these sporozoites from mosquito salivary glands enables the manufacturing of one promising malaria vaccine.

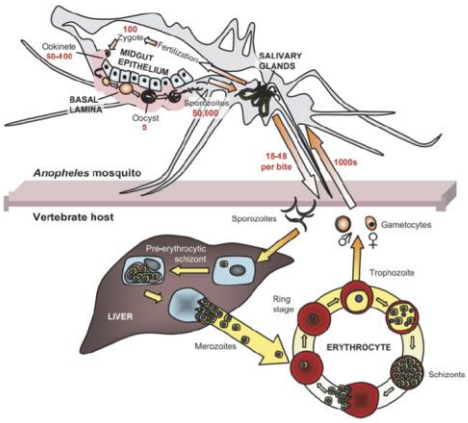

Figure 1 Life cycle of Malaria

However, the current extraction process is fully manual and requires highly trained technicians to perform delicate manual operations under a microscope. The process is time-consuming and expensive. An automated dissection process is being developed at LCSR that uses a robotic microsurgical instrument to manipulate mosquitoes.[3] The autonomy of the robotic system hinges on sophisticated computer vision methods to detect mosquitoes and their body parts and to provide quality control during the process.

- **Significance**

This project has great importance as it seeks to progress towards building a proficient robotic architecture for dissecting mosquitoes. These visual algorithms will amplify the efficacy of the system and observe its performance of the system. The computer vision methods are essential for reliable and efficient operation of the system. And these algorithms can also minimize operational expenses during implementation. This project will have much broader implications, propelling Sanaria nearer to mass-producing a malaria inoculation.

- **Problems**

Large-scale production of the malaria vaccine is hindered by the time-consuming and skill-intensive manual process of extracting sporozoites from mosquito salivary glands. Now LCSR is developing a fully automated dissection system, but this system still faces several problems as follows:

1) The existing computer vision methods cannot determine the mosquito's orientation as well as the exudate quality. If such methods are implemented, the overall performance of the automated dissection system can have significant improvement.
2) Computer vision codes from Project Gitlab do not have the proper code for the classification neural network. I need to implement it myself.
3) The automated dissection system has a large amount of image data in its database. However, many of these image data do not have the correct label. I need to check and correct the relevant label and then extract the relevant image data to generate the dataset for training and testing according to the requirements.
4) Some codes available in Project Gitlab often don't have a documentation or a wiki that is easy to understand, and I need to explore how to use these codes and refine and modify the corresponding documentation.
5) It has been a long time since this automated dissection system has had a complete experiment, this has resulted in the image data in the database not being updated for a long time. I only have a limited amount of image data available. I need to perform data augmentation on the dataset to improve the training performance.
6) Exudate quality image annotation needs Sanaria's expert help. I need to generate the exudate quality dataset as early as possible and write the relevant documentation to help Sanaria expert with the annotation work. After getting the annotations from Sanaria expert, I need to review each image and the corresponding annotation to make sure there are no errors in the communication between the two parties.

3. Prior Work

The previous research involved developing the mechanical design and software framework for a robot system capable of simultaneously positioning, decapitating, and extracting salivary glands from mosquitoes. The system also includes several vision algorithms that utilize both

image processing and deep learning. Figure 2 provides an image of the hardware and the output of one of its vision algorithms.

Figure 2 The hardware (left) and the output of one of its vision algorithms (right)

4. Goals and Deliverables

- **Goals**

This project aims to create computer vision algorithms for the robot mosquito dissection system, which is an important part of continuing development. Specific aims are to develop new DL-based CV methods and integrate existing CV methods for the mosquito dissection system, which include:

1) Mosquito Orientation Classification
2) Exudate Quality Evaluation
3) Prediction of Dissection Success
4) Exudate Volume Estimation

- **Deliverables**

The key activities and deliverables can be found in Table 1:

Table 1 Activities and Corresponding Deliverables

	Activity	Deliverable
Minimum	Review mosquito orientation detection code, complete (if necessary), generate up-to-date training data, train, and evaluate network, make sure method is properly integrated, complete documentation.	Completed mosquito orientation classification code in Gitlab repository/ Documentation in Gitlab Wiki and Readme files.
	Complete exudate quality evaluation. Establish contact with Sanaria expert, review the database, gather exudate images for training from database. Have images classified by Sanaria expert. Train classifier using classified dataset. Evaluate results on training dataset. Possibility: Evaluate exudate quality by Sanaria expert visiting LCSR.	Completed exudate quality evaluation code in Gitlab repository/ Documentation in Gitlab Wiki and Readme files.

Expected	Complete prediction of dissection success. Train classifier using exudate quality classification data, and mosquito images taken at the early stages of processing. Evaluate results on training dataset. Interpret results. Run laboratory experiments to confirm findings.	Completed prediction of dissection success code in Gitlab repository/ Documentation in Gitlab Wiki and Readme files.
Maximum	If prediction results are confirmed to be valid, investigate methods to locate specific regions on mosquito images that contribute strongest to variability in exudate quality. Document investigation and results. Present investigation results to hardware design team.	Relevant code in Gitlab repository/ Documentation in Gitlab Wiki and Readme files.
	Develop exudate volume estimation using deep learning techniques.	Completed exudate volume estimation code in Gitlab repository/ Documentation in Gitlab Wiki and Readme files.

5. Technical Approach

Here is a flow chart of the technical approach:

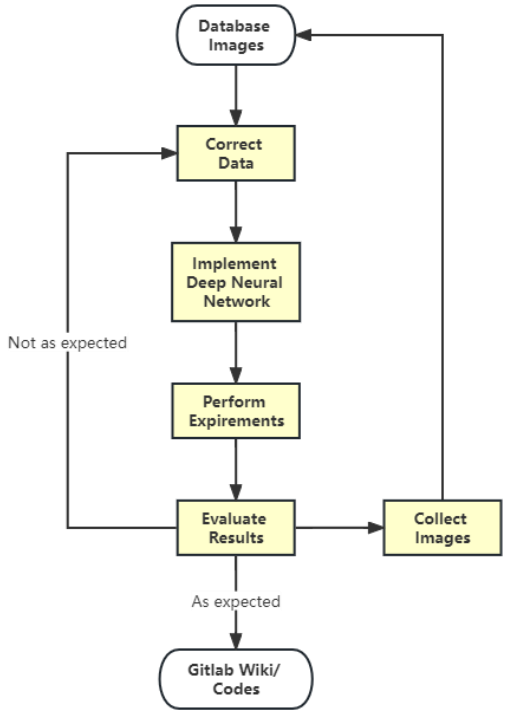

Figure 3 Technical approach flow chart

- **Images and Annotations**

A large amount of image data is needed to train the classification model. The automated robotic system has several cameras with different angles in different positions. Many images have been stored in the database since the system was built. However, not every image has the correct label. Need to correct the label for each image in this database. Extract images and their corresponding annotation information from the database. Perform data type conversion and make training sets.

- **Deep Neural Network**

Enough data has now been collected for this project. So, it is the best choice to use deep neural network to complete the above classification task. For classification tasks, deep neural network has many advantages. Deep neural networks offer a powerful and flexible approach to classification tasks, with many advantages over traditional machine learning algorithms. Deep neural networks can learn to automatically extract relevant features from raw data, without the need for hand-crafted feature engineering. This can save a lot of time and effort, as it eliminates the need for domain-specific knowledge or expert input in the feature extraction process. Besides, it can be scaled up to handle very large datasets with many input features and output classes. This makes the deep neural networks suitable for a wide range of classification tasks. Moreover, pre-trained deep neural networks can be used as a starting point for new classification tasks, by fine-tuning the network on a smaller dataset or a new set of classes. This can significantly reduce the amount of data and training time required for new tasks and can lead to better performance than training a new model from scratch. We mainly use three deep neural networks: ResNet, VGG, and DenseNet. Those three networks could be imported by Pytorch. PyTorch is used for this project because it provides a Python-based interface for building and training machine learning models, which can make the training process much easier. The following are the advantages of each deep neural networks and why this network should be chosen.

1) ResNet18/ ResNet152[5]

The main advantage of ResNet in image classification is its ability to train very deep neural networks effectively, which allows the network to capture more complex and subtle features in the data. ResNet18 and ResNet152 are both variants of the Residual Network (ResNet) architecture, but they differ in their depth and the number of parameters they have. Deeper ResNet architectures like ResNet152 tend to perform better on complex classification tasks, but they also require more training time and computational resources.

2) VGG16[6]

The VGG architecture's high accuracy, transfer learning capabilities, and interpretability make it a powerful tool for image classification tasks. VGG has been pre-trained on the ImageNet dataset, which contains over a million images and 1000 classes. This pre-training allows the

model to learn general features that are useful for a wide range of image classification tasks. By fine-tuning the pre-trained VGG model on a new dataset, we can achieve high accuracy with much less training data. Besides, the VGG architecture's simple and uniform design makes it easy to interpret and analyze the features learned by the model. This interpretability can be useful for understanding how the model is making predictions and identifying potential areas for improvement.

3) DenseNet121[7]

The dense connectivity of DenseNet provides an effective way to increase the flow of information and gradient signals through the network, leading to improved performance in image classification tasks. DenseNet achieves excellent performance with fewer parameters compared to other popular architectures, by reusing feature maps throughout the network. This parameter efficiency can help reduce overfitting and make the network easier to train. Moreover, dense connectivity allows for efficient feature reuse across the network, leading to better feature extraction and representation learning. This means that the network can learn more robust features with fewer layers, making it easier to train and reducing the risk of overfitting.

- **Data Augmentation**

Data augmentation refers to the process of artificially increasing the size of a training dataset by applying various transformations or modifications to the existing images. These transformations create new training samples that are variations of the original images while preserving their label or class. Data augmentation is necessary for the small number of datasets like ones I use in this project. The image data in the database are not updated for a long time. I only have a limited amount of image data available. By augmenting the existing dataset, the effective size of the training data is increased, which provides more diverse examples for the model to learn from. This helps to prevent overfitting and improves the model's ability to generalize well to unseen data. It can also introduce variations in the training samples, making the model more robust to changes in the input data. It allows the model to learn features that are invariant to transformations like rotation, scaling, or translation, which can occur in real-world scenarios. There are many kinds of data augmentation methods, such as geometric and color augmentation, such as reflecting the image, cropping, and translating the image, changing the color palette of the image, color processing, and geometrical transformations. In this project, I use image blurry to create more bad quality images, and randomly select ROI in each image to create more bad view images.

6. Results

I have completed a total of 3 tasks as follows: 1) Mosquito Orientation Classification; 2) Exudate Quality Evaluation; 3) Prediction of Dissection Success. As for the maximum task, I didn't have time to finish it. This is because one month into the project, my supervisor asked

me to assist another senior in writing computer vision algorithms for his project, and eventually published a conference paper as the second author. Here is a flow chart about the overall process for each task:

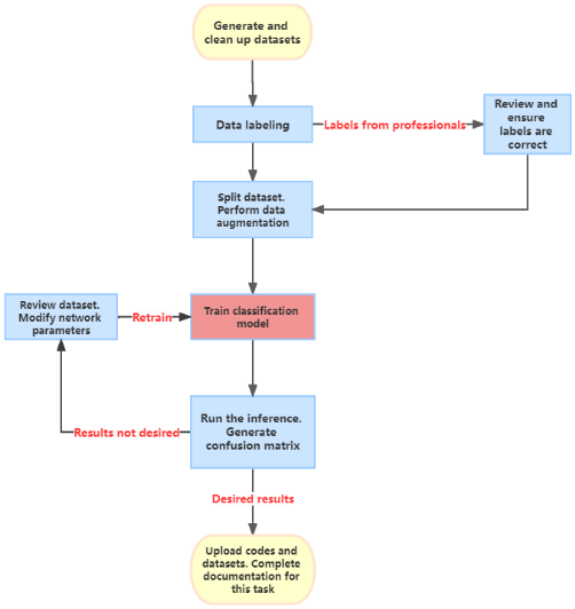

Figure 4 Overall process for each task

- **Mosquito Orientation Classification**

First, I refine the classification neural network based on the part of the code already in the Gitlab repository. I used Python and Pytorch to write and implement three different neural networks: VGG16, ResNet and DenseNet. Then, write the corresponding inference code files. The inference results on the test set are automatically saved to an Excel sheet for viewing later.

After writing the code required for training and testing, the orientation labels of the first 1000 images in the database are reviewed and the incorrect labels are corrected. For mosquito orientation classification, there are six classes:

Table 2 Mosquito orientation classes

Class Number	Mosquito Orientation
0	No mosquito
1	Left
2	Right
3	Back
4	Belly
5	Other

After correcting all orientation labels, write codes to extract the corresponding image data and generate the dataset. After training, the model is obtained and inference is performed on the test set, and the results obtained are shown in the figure 5:

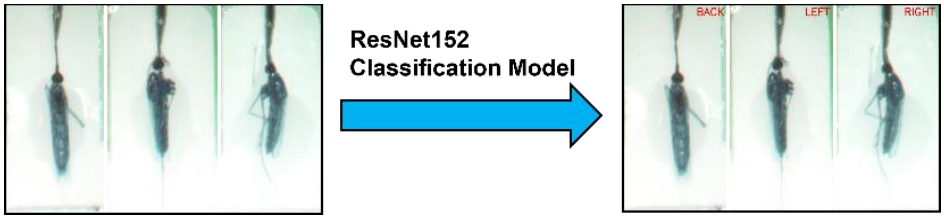

Figure 5 Mosquito orientation classification result (with classification results on the upper right)

It can be seen from the figure, the resulting model works well on the test set. This indicates that my classification neural network implement is correct. After confirming that the results are correct, write the documentation and the Wiki in Gitlab. It allows for better subsequent development.

- **Exudate Quality Evaluation**

Exudate quality labels can only be annotated by Sanaria expert. First, I filtered the image data in the database and selected only the images with the correct ROI to be used for the dataset for Sanaria expert. Then, together with my mentor, we wrote the documents and Excel tables to provide Sanaria expert with the instruction on how to label and the characteristics of each classification class. To make it easy for Sanaria expert to label, I specifically put the images of front view and side view with the same id together. All the concatenated images were scaled equally to 512x512.

Figure 6 Concatenated images for Sanaria expert to label

1) Image quality classification

After getting the Sanaria expert's label, review the entire label sheet and make modifications accordingly. For this task, a total of 2 classifiers are trained, one is image quality classifier and the other is exudate quality classifier. For image quality classifier, there are three classes:

Table 3 Image quality classes

Class Number	Image Quality
0	Good quality
1	Bad quality
2	Bad view

These are template input images for each class:

Figure 7 Good quality image

Figure 8 Bad quality image

Figure 9 Bad view image

Bad view means the ROI is not correct, the image does not have the area where the exudate is available. It also represents the situation that some obstacles, like droplet, are in front of the camera and the exudate is not visible.

In the initial attempt, I concatenated the front view and side view images together and wanted to use a single classifier to determine whether both images of the same mid were good quality, or only one view was good quality, or both views were bad quality, or both were bad views. However, the models trained in this way was very ineffective.

After discussing with the mentor, we decided to separate the front view and side view images for the image quality classification, and use the data augmentation to blurry the good quality images to generate more bad quality images. Then we randomly selected the areas other than ROI and use them as bad view images. Then, I reviewed the whole dataset, removed the image data that do not belong to the current class. The trained model works very well, and the confusion matrices are shown in the figure:

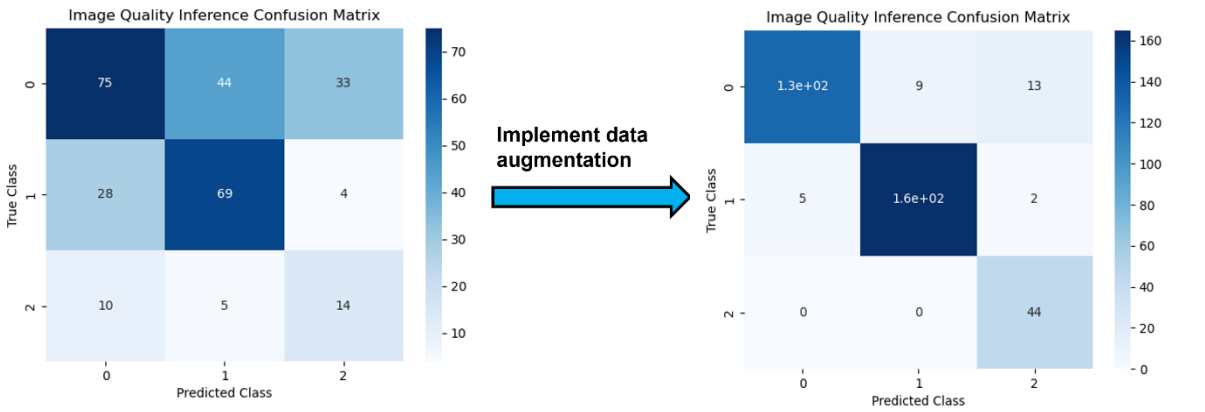

Figure 10 Image quality confusion matrices (left is before the data augmentation, right is after the data augmentation)

From the figure, the number on the diagonal of the confusion matrix increases significantly after implementing the data augmentation. This indicates that the performance of the image quality classifier is better. The image quality classifier is ready to use.

2) Exudate quality classification

For the label of this classification, mentor and I worked closely with Sanaria's expert to label all the classes of the exudate. For exudate quality classification, there are five classes in total:

Table 4 Exudate quality class

Class Number	Exudate Quality
0	No exudate
1	Low volume exudate
2	Exudate quality 1
3	Exudate quality 2
4	Exudate quality 3

For the exudate quality classification, I concatenated two views images together, and rescale the images to 512X256. Template input images are shown below:

Figure 11 No exudate

Figure 12 Low volume exudate

Figure 13 Exudate quality 1

Figure 14 Exudate quality 2

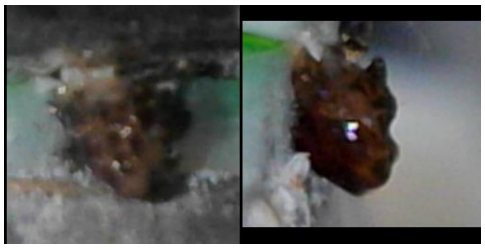

Figure 15 Exudate quality 3

For exudate quality classification, I trained two classifiers. One is to input image where both views are good quality, the other is to input image where both views are good quality and input images where only one view is good quality. The confusion matrix for each of them is shown below:

(a) both views good quality images *(b) both views good quality images/one view good quality images*

Figure 16 Exudate inference result

A large number on the diagonal of the Confusion matrix indicates that the classifier is performing well. It means the predicted class matches the true class. It can be seen from the figure that, both classifiers have good results. The classifier which only input both views are good quality images performs better. However, for class 1,3 and 4, there is limited amount of such data in the dataset. After discussing the results with my mentor, we all agree that the exudate quality classifier will perform much better if there are more data for each class in the dataset. If more exudate image data are collected and the model is continuously trained, this exudate quality classifier will perform better and better.

- ## Predication of Dissection Success

For prediction of dissection success, input good quality image. **The classes are the same as exudate quality classes above.** It has the following classes:

Table 5 Exudate quality class

Class Number	Exudate Quality
0	No exudate
1	Low volume exudate
2	Exudate quality 1
3	Exudate quality 2
4	Exudate quality 3

This task is to investigate whether the way mosquito is placed in the early stage on the turntable influences exudate quality. A template input image is shown below:

Figure 17 Template input image for the prediction of dissection success

The inference confusion matrix is shown below:

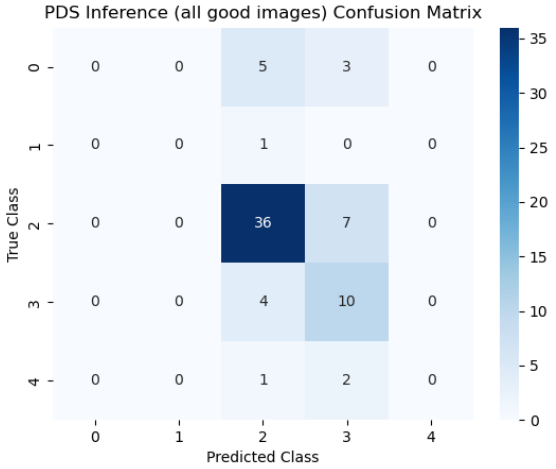

Figure 18 Prediction of dissection success inference result

Surprisingly, the numbers in the confusion matrix are concentrated along the diagonal in the confusion matrix rather than randomly distributed. This shows that we can predict the exudate quality given the placement of mosquitoes in the early stage. It shows that the classifier has some predictive power. It can predict the exudate quality based on the placement image before the dissection. It is an enlightening result. Next step can be investigating methods to locate specific regions on mosquito placement images that contribute strongest to variability in exudate quality.

- **Extra Work and Publication**

I did not finish the maximum task. It is because that my mentor asked me to assist another senior's project, using what I had done in 2022 Fall. Part of my project schedule was allocated to it. The extra work is called "Applications of Uncalibrated Image Based Visual Servoing in Micro- and Macroscale Robotics". It was not in my original proposal. The extra work presents a robust markerless image based visual servoing method that enables precision robot control without hand-eye and camera calibrations in 1, 3, and 5 degrees of freedom. My tasks in this extra work are implementing gripper tip detection, and screwdriver tip detection, based on YOLOv5-Pytorch. These object detection methods are essential for the hand-eye calibration method proposed in this paper. For the extra work, I collected images and trained object

detection models. Moreover, I wrapped these computer vision algorithms using ROS. Besides, I also designed the experiment platform, and helped conducting all experiments and analyzed results. Eventually, I wrote and submitted a conference paper to the 2023 IEEE International Conference on Automation Science and Engineering (CASE) as a second author. A screenshot of the paper is shown below:

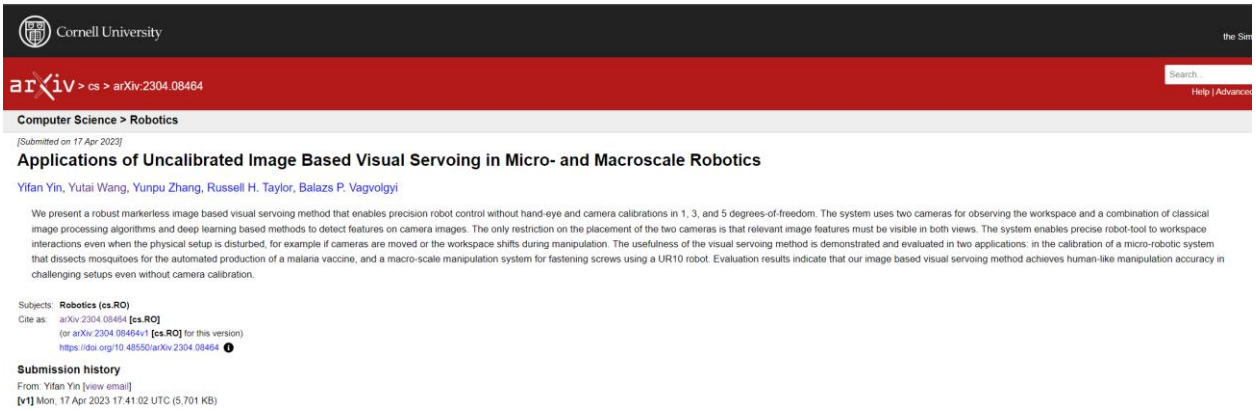

Figure 19 Screenshot of the submitted conference paper

7. Timeline and Dependencies

- **Dependencies**

The project is mainly virtual so there are no physical dependencies. The dependencies are listed in Table 6.

Table 6 Dependencies

Dependency	Need	Status	Follow-up	Contingency Plan
Computing Power	GPU for training neural network	Ready to use	Have access to Sanaria PC	N/A
Training images in log database	For annotation and training	Completed	Review available data in database, fix data labels if needed	If amount of data is inadequate, then run laboratory experiments with mosquitos to generate more data.
Sanaria expert needs to evaluate exudate images and classify them	Use correctly labeled exudate images for the exudate quality evaluation task	Completed	N/A	If evaluation results are not received on time from expert, come up with own classification. Continue other steps.

- **Timeline**

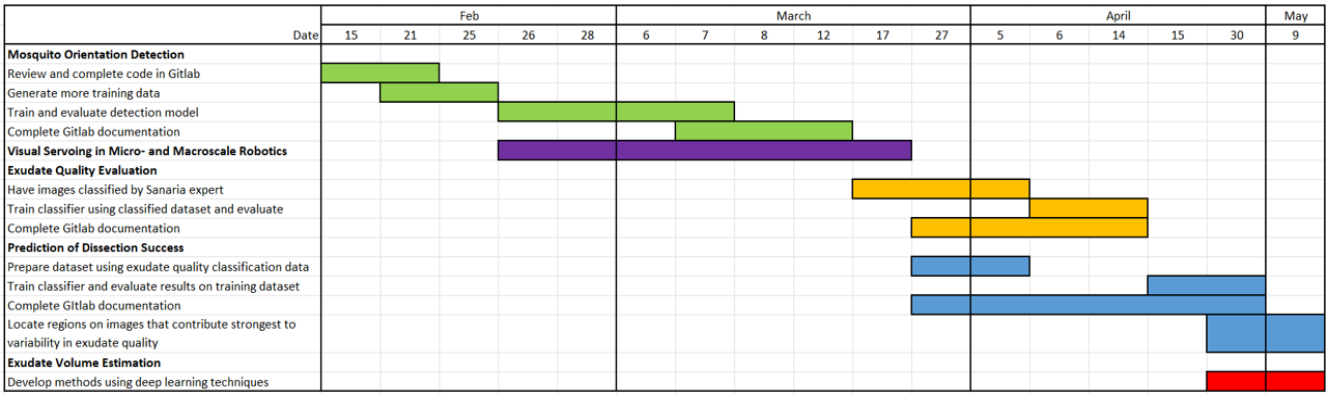

Figure 20 Project timeline

8. Roles and Management Plan

- **Team Members and Roles**

The team consists of:

Yutai Wang (ywang790@jhu.edu)

MSE Robotics, Laboratory for Computational Sensing & Robotics, first year

Sole responsibility for all tasks required for this project.

The mentor consists of:

Balazs Vagvolgyi (balazs@jhu.edu)

Associate Research Scientist (Applied Biomedical Engineering)

Balazs is very experienced in computer vision and will help provide advice on developing solutions and solving difficulties in computer vision. He will be the primary mentor for this project.

- **Meeting Schedule and Communication Platforms**

Monday 9:00pm – 10:00pm: Meet with the project team over Zoom. The purpose of this meeting is to coordinate efforts with the hardware and software teams, discuss concerns and provide suggestions.

Monday 11:00pm – 12:00pm: Meet with Mr. Balazs. The purpose of this meeting is to report the weekly progress, discuss technical difficulties, find possible solutions, and make plans for the week.

Communicate mainly by email, sometimes text message is also used.

9. Management Summary

- **Credits**

Yutai Wang (Team Leader/ Team Member): Responsible for all tasks. Developed classification neural networks. Collected image data and collaborated with labeling.

Designed the experiment platform and conducted experiments for the extra work. Wrote documentation and uploaded corresponding codes to project's Gitlab for each task.
Balazs Vagvolgyi (Mentor): Provide mentorship. Have regular weekly meeting with me to discuss results and determine what should be the next step.

- **Accomplished VS Planned**

Table 7 Accomplished VS Planned

	Planned Task	Task Status
Minimum	Develop and complete mosquito orientation detection code. Generate up-to-date training data, train, and evaluate network, make sure method is properly integrated, complete documentation.	**Completed**
	Complete exudate quality evaluation. Establish contact with Sanaria expert, review the database, gather exudate images for training from database. Have images classified by Sanaria expert. Train classifier using classified dataset. Evaluate results on training dataset.	**Completed**
Expected	Complete prediction of dissection success. Train classifier using exudate quality classification data, and mosquito images taken at the early stages of processing. Evaluate results on training dataset. Interpret results.	**Completed**
	Complete documentation in Gitlab Wiki and Readme files for each task above.	**Completed**
Extra Task	Help with the gripper tip detection and screwdriver tip detection in another project. Design the experiment platform, and conduct experiments to collect data for analyzing. Write and submit the conference paper "Applications of Uncalibrated Image Based Visual Servoing in Micro- and Macroscale Robotics" to 2023 IEEE International Conference on Automation Science and Engineering (CASE) as a second author.	**Completed**
Maximum	If prediction results are confirmed to be valid, investigate methods to locate specific regions on mosquito images that contribute strongest to variability in exudate quality. Document investigation and results.	**Future Work**
	Develop exudate volume estimation using deep learning techniques.	**Future Work**

- **Future Plan**

  1) Based on the Prediction of Dissection Success results, investigate methods to locate specific regions on mosquito images that contribute strongest to variability in exudate quality. Record relevant codes and documentation in Gitlab.
  2) Develop exudate volume estimation based on deep learning methods. Complete exudate volume estimation codes in Gitlab repository. Write documentation in Gitlab Wiki and Readme files.

- **Lessons Learned**

  1) Give professionals the data they need to label early. In this project, the data given to Sanaria expert label took a month to complete. I think in future projects, I should keep regular communication with all parties to ensure that each task can be completed on time. In this way, the progress of the project will not be delayed.
  2) Documentation for codes is important. Having documentation to help when developing code can make subsequent development work much smoother. Well-documented code makes it easier for other developers, including myself in the future, to understand the purpose, functionality, and usage of the code. It provides clear explanations, instructions, and context, facilitating collaboration and teamwork. Documentation helps reduce ambiguity and allows developers to work together more effectively. Overall, it is helpful for this project.
  3) Expect the unexpected. By anticipating the unexpected, I can proactively identify potential risks and plan mitigation strategies. This allows me to be better prepared to handle unexpected events, minimizing their impact on project timelines, and overall success.
  4) Have regular meetings with my mentor every week is very important for the project. Mentor can offer valuable guidance and advice based on his expertise and experience. Regular meetings allow me to discuss my project's progress, challenges, and decisions, and receive feedback from my mentor with a broader perspective. His insights can help me navigate complex issues, make informed decisions, and avoid potential pitfalls.

10. Technical Appendices

- Links to each task's codes and documentation are on the project Wiki page. Project's poster, checkpoint slides, proposal, and paper review presentation are all on the project Wiki page. The project Wiki page can be accessed at: https://ciis.lcsr.jhu.edu/doku.php?id=courses:456:2023:projects:456-2023-04:project-04
- The project codes and documentation for each task can also be accessed at the project Gitlab repository (Login to the project Gitlab requires permissions. The project Gitlab is not public.):
  Mosquito Orientation Classification Task:
  Codes: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/tree/main/tasks/mosquito_orientation_2
  Documentation/Wiki: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/wikis/Orientation-Classification
  Exudate Image Quality Classification Task:

Codes: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/tree/main/tasks/exudate_image_quality_classification
Documentation/Wiki: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/wikis/Exudate-Image-Quality-Classification

Exudate Quality Classification Task:
Codes: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/tree/main/tasks/exudate_quality_classification
Documentation/Wiki: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/wikis/Exudate-Quality-Classification

Prediction of Dissection Success Task:
Codes: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/tree/main/tasks/success_prediction_2
Documentation/Wiki: https://git.lcsr.jhu.edu/mosquito-vision/sanaria_cv_dl/-/wikis/Prediction-of-Dissection-Success

11. Reading List

- Image classification. Papers With Code. (n.d.). Retrieved February 20, 2023, from https://paperswithcode.com/task/image-classification
- Deep Neural Networks. Tutorials Point. (n.d.). Retrieved February 20, 2023, from https://www.tutorialspoint.com/python_deep_learning/python_deep_learning_deep_neural_networks.htm
- Learning pytorch with examples. Learning PyTorch with Examples - PyTorch Tutorials 1.13.1+cu117 documentation. (n.d.). Retrieved February 23, 2023, from https://pytorch.org/tutorials/beginner/pytorch_with_examples.html
- T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, China, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.
- S. S. Nath, G. Mishra, J. Kar, S. Chakraborty and N. Dey, "A survey of image classification methods and techniques," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, 2014, pp. 554-557, doi: 10.1109/ICCICCT.2014.6993023.

References

[1] Centers for Disease Control and Prevention. (2022, March 22). CDC - Malaria - about malaria - faqs. Centers for Disease Control and Prevention. Retrieved February 5, 2023, from https://www.cdc.gov/malaria/about/faqs.html

[2] Schrum, M., Canezin, A., Chakravarty, S., Laskowski, M., Comert, S., Sevimli, Y., ... & Taylor, R. H. (2019). An efficient production process for extracting salivary glands from mosquitoes. arXiv preprint arXiv:1903.02532.

[3] W. Li et al., "Automated Mosquito Salivary Gland Extractor for PfSPZ-based Malaria Vaccine Production," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 866-872, doi: 10.1109/ICRA48506.2021.9560959.

[4] M. Xu et al., "Mosquito Staging Apparatus for Producing PfSPZ Malaria Vaccines," 2019 IEEE 15th International Conference on Automation Science and Engineering

(CASE), Vancouver, BC, Canada, 2019, pp. 443-449, doi: 10.1109/COASE.2019.8843147.
[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[7] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).