Project 4 Proposal

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

2/20/2023

Contents

1. **Introduction** 3
2. **Background and Significance** 3
    - **Background** 3
    - **Significance** 4
3. **Prior Work** 4
4. **Goals and Deliverables** 4
    - **Goals** 4
    - **Deliverables** 4
5. **Technical Approach** 5
    - **Images and Annotations** 6
    - **Deep Neural Network** 6
6. **Timeline and Dependencies** 8
    - **Dependencies** 8
    - **Timeline** 8
7. **Roles and Management Plan** 8
    - **Team Members and Roles** 8
    - **Meeting Schedule and Communication Platforms** 9
8. **Reading List** 9

References 9

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 and Significance

- ## 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.

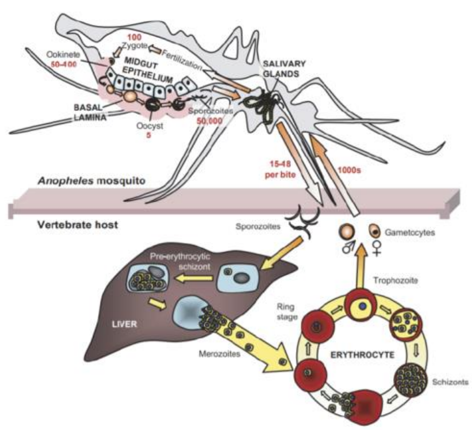

Figure 1 Life cycle of Malaria

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.

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.

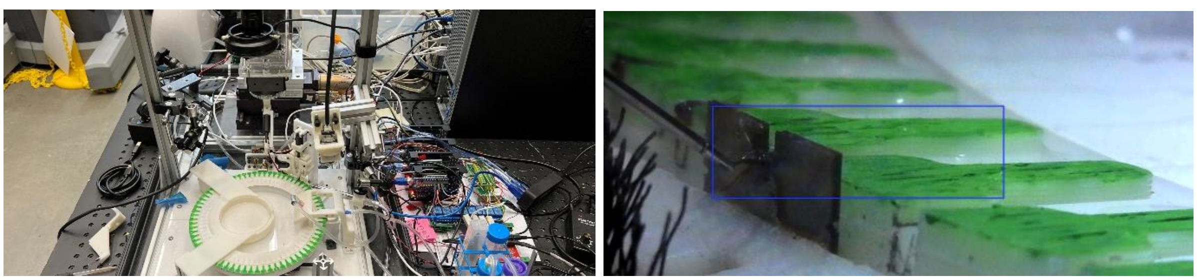

Figure 2 The hardware (left) and the output of one of its vision algorithms (right)

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 Detection
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 detection 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.
	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.
Maximum	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:

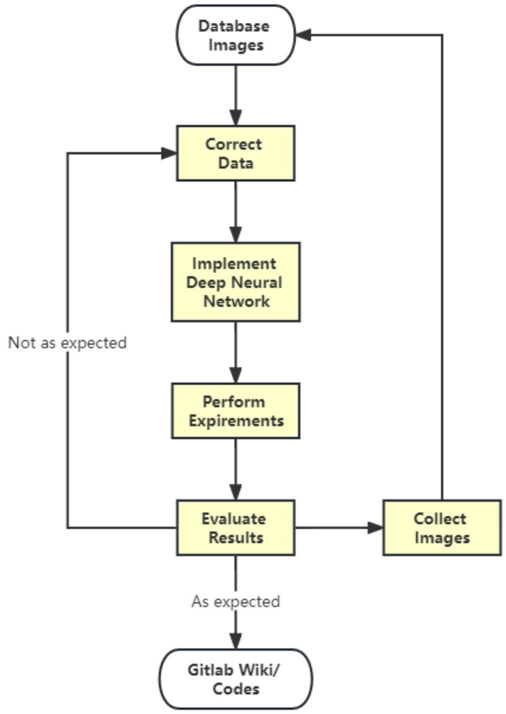

Figure 3 Technical approach flow chart

- **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.

6. Timeline and Dependencies

• Dependencies

The project is mainly virtual so there are no physical dependencies. The dependencies are listed in Table 2.

Table 2 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	In progress	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	In progress	N/A	If evaluation results are not received on time from expert, come up with own classification. Continue other steps.

• Timeline

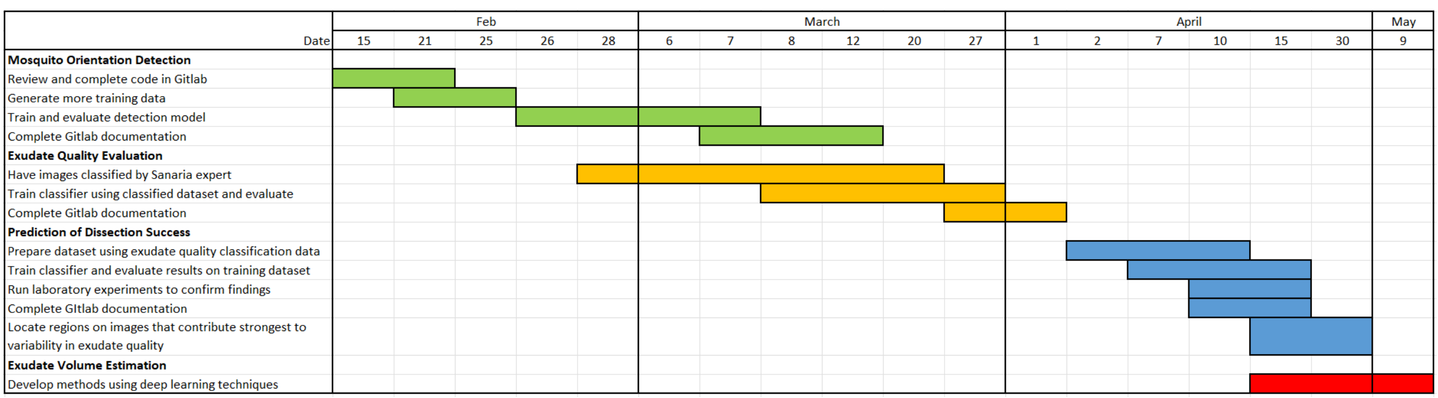

Figure 4 Project timeline

7. 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)

Mr. 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.

8. 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).