2021 Spring EN.620.801 MSE Robotics Research Plan

Automated Mosquito Dissection - Vision

Alan Lai, alan\_lai@jhu.edu

Mentors:

* Dr Russell Taylor, rht@jhu.edu
* Balazs Vagvolgyi, balazs@jhu.edu

Objective/Goal

The main objective of this research project is to provide computer vision guidance to an automated mosquito dissection robotic system for live malaria vaccine production. The main tasks of this research include:

1. Deep learning-based prediction of mosquito decapitation/salivary extraction success or failure based on camera images of the mosquito acquired in the robotic processing pipeline prior to decapitation,
	1. Or to determine the feasibility of such.
2. Deep learning-based validation of success or failure of mosquito decapitation.
3. Integration of tasks (if successful) into the sanaria\_dl\_service package for ROS integration with the robotic mosquito dissection system.
4. Providing insight into the potential root causes of downstream decapitation and squeezing failures originating in the earlier stages of the pipeline, based on the analysis of collected image data.
5. Provide guidance for CIS II project members.

Motivation

Previous work has already been done using computer vision to support and guide the current robotic system for automatic mosquito processing. Image processing-based methods include mosquito detection, proboscis detection, and neck detection, while deep learning methods include mosquito orientation classification, mosquito detection, and pose estimation. Many of these methods are already currently in use, providing support to automate the system.

Though those methods have been successful in allowing the robotic system to operate autonomously, comparatively little amounts of effort have been put in with regards to vision-based validation and error prevention of key steps. That is, most existing methods are responsible for guiding the robot’s motion to process mosquitoes but only one method was designed to facilitate error checking and recovery. All other error checking and validation are currently done manually by the operator and resolved manually as well.

Two key steps in the processing pipeline are the decapitation step and the squeezing step. Though the current vision algorithm for proboscis and neck detection work well, the variety in mosquito shapes, orientation, flexibility, and morphology means that even if keypoints are identified well and the robot moves through the correct motions, the mosquito may not be placed in the correct location, causing issues in decapitation and/or squeezing steps. There is currently no way of resolving these issues if mosquito decapitation or squeezing is ineffective, and the corresponding mosquito would be best case discarded or worst case create problems in the processing pipeline. If an algorithm is capable of predicting, based on mosquito positioning and orientation after the robot has placed the mosquito on the decapitation blades, whether decapitation and/or squeezing will fail, this presents an opportunity for the robot to reposition the mosquito to a valid position such that such wastage would not occur.

The reason why deep learning is preferred as a method over image processing with regards to prediction of decapitation/squeezing success/failure is that there are many failure cases that could occur, be it mosquito being too far up, too much to the side, too much in front, which all depend on individual mosquito morphology. It would be impossible to come up with an empirical set of rules for image processing to enforce, and hence deep learning, which allows for greater flexibility and robustness with respect to mosquito morphology, is a more promising solution to this problem. The other advantage of deep learning over image processing is that image processing requires knowledge of the causes of the problems and a method of modeling those causes in order to recognize and resolve the issues. As we currently do not know what causes these issues in decapitation or squeezing, deep learning will be able to allow us to circumvent the need to explicitly model the system.

A secondary project that involves visual confirmation of mosquito decapitation success or failure will also be explored. The aim is to take an image of the mosquito and gripper subsequent to blade actuation, feed that image into a deep learning algorithm, which confirms whether the decapitation presents as success or failure. This is being developed in conjunction with image processing methods, to serve as a backup method for validation in case the image processing method fails to achieve desired results.

Technical Approach

*Prediction of Success/Failure of Decapitation/Extraction*

Top view and side view images will be taken after the robot has placed the mosquito onto the decapitation blades, but before the blades are actuated. If possible, mosquito images on the turntable prior to robot manipulation may also be taken and incorporated into the training. After the blades are actuated, the operator will manually determine whether the decapitation was a success. Then, the mosquitoes will be further processed and salivary glands will be attempted to be extracted via squeezing. Once the mosquitoes have been squeezed, the operator will then manually determine whether the squeezing was a success. The decapitation and/or squeezing success will then be written to a file that links these results to the original top and side view images taken of the mosquitoes prior to decapitation.

These images will then serve as training images for training our neural network. PyTorch will be used for training, and transfer learning, using PyTorch’s pretrained networks, will be used. This will simply be a classification problem, predicting classes:

1. Success for both decapitation and squeezing
2. Failure for decapitation
3. Failure for squeezing

Note that more classes may be added in the future that may further delineate differences in how successful the squeezing step was.

Images will be split 80-20 for training and validation sets, and parameters such as optimizer, learning rate, batch size, and other potential data augmentation schemes will be attempted in order to both (1) determine the feasibility of prediction of mosquito processing success (2) create a predictor that attempts to predict the result of the decapitation and/or squeezing steps. After training occurs, an inference pipeline will be created, and then used for testing with a new test set. Subsequent to testing, a ROS test client and server will be created in preparation of integration with the larger system.

*Validation of Mosquito Decapitation*

Top view and side view images will be taken after the blades have been actuated for the mosquito decapitation step. The operator will manually determine whether decapitation was a success, and this label will be saved along with the corresponding images to serve as our dataset.

The images will then serve as training images for training our neural network. PyTorch will be used for training, and transfer learning, using PyTorch’s pretrained networks, will be used. This will again be a simple classification problem, predicting classes:

1. Successful decapitation
2. Failed decapitation

Images will be split 80-20 for training and validation sets, and parameters such as optimizer, learning rate, batch size, and other potential data augmentation schemes will be attempted in order to create a validation algorithm to confirm success of the decapitation step. After training occurs, an inference pipeline will be created, and then used for testing with a new test set. Subsequent to testing, a ROS test client and server will be created in preparation of integration with the larger system.

Testing Plan

* Task 1: A testing set of 100 image pairs of mosquitoes after placement on the decapitation blades will be obtained and fed into the success/failure predictor. 50 images shall be of failure cases, and 50 images shall be of success cases. We desire that at least 70% of the mosquitoes will be properly predicted in terms of salivary gland extraction success/failure further down in the pipeline.
* Task 2: A testing set of 100 image pairs of mosquitoes after blade actuation will be obtained and fed into the decapitation validator. We desire that at least 90% of these images will be successfully classified as either decapitation success or failures.

Deliverables

* Working algorithm on mosquito decapitation/salivary gland extraction success prediction
* Working algorithm on decapitation validation
* ROS integration test client/server code for prediction and validation algorithms
* Thorough documentation on all algorithms and usage

Timeline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | February | March | April | May |
|  |  | 02/15 | 02/22 | 03/01 | 03/08 | 03/15 | 03/22 | 03/29 | 04/05 | 04/12 | 04/19 | 04/26 | 05/03 |
| Dependencies | Camera Setup |  |  |  |  |  |  |  |  |  |  |  |  |
| Image Collection |  |  |  |  |  |  |  |  |  |  |  |  |
| Success Prediction | Environment Setup |  |  |  |  |  |  |  |  |  |  |  |  |
| Train Network |  |  |  |  |  |  |  |  |  |  |  |  |
| Setup ROS integration |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  |
| Success Validation | Environment Setup |  |  |  |  |  |  |  |  |  |  |  |  |
| Train Network |  |  |  |  |  |  |  |  |  |  |  |  |
| Setup ROS integration |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  |

Milestones

* 2021/02/28 - Deep Learning Environment Setup
* 2021/02/28 - Repo, Wiki Setup for Prediction task
* 2021/03/15 - Feasibility of Prediction Algorithm Determined
* 2021/03/21 - Success rate >70% for Prediction Algorithm
* 2021/03/28 - ROS integration for Prediction Algorithm
* 2021/04/04 - Repo, Wiki Setup for Validation task
* 2021/04/25 - Success rate >90% for Prediction Algorithm
* 2021/05/02 - ROS integration for Validation Algorithm
* 2021/05/09 - Documentation Complete

Dependencies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Dependency** | **Solution** | **Alternative Plan** | **Progress &****Deadline** |
| **1** | DL Camera Setup for Data Collection | Communicate and work with Balazs | Use temporary camera setup used by Wanze | Pending2021/02/19 |
| **2** | Mosquito images for DL (~400) | Wanze will take images parallel to his experiments | - | Pending2021/02/26 |
| **3** | Computational resources  | Lab computer (NVIDIA Titan Xp, 12GB) | CS Ugrad Servers, Personal Computer | Solved |
| **4** | Libraries (OpenCV, Pytorch, DL Models)  | Free online access/open source | N/A | Solved |
| **5** | Integration with robotic system (ROS) | Communicate and work with Balazs | N/A | Solved |
| **6** | Feedback from Mentors | Lab meetings | Communication via email | Solved |