Project 04 Checkpoint Presentation:

# Vision Guided Mosquito Dissection for the Production of Malaria Vaccine


**Team Member**: Yutai Wang ywang790@jhu.edu

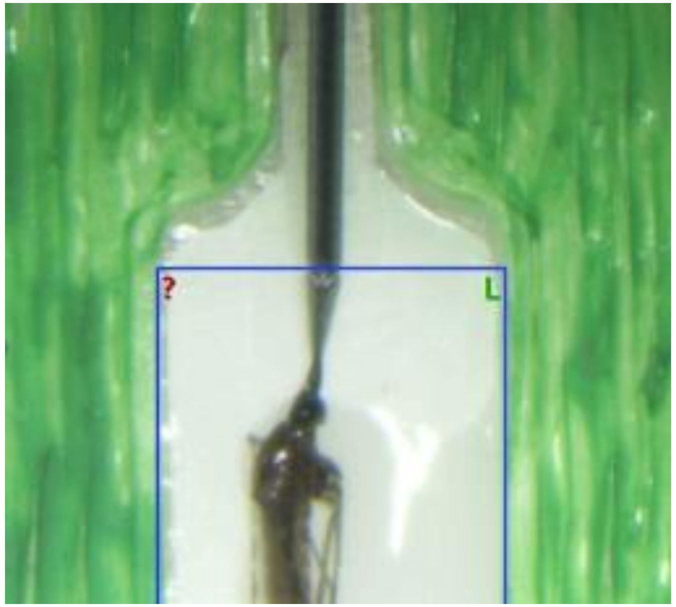
**Mentors**: Mr. Balazs Vagvolgyi balazs@jhu.edu

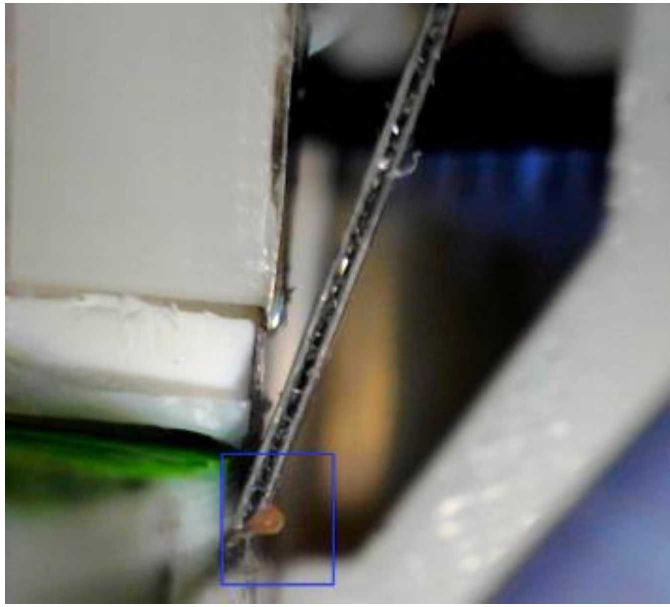

**Date:** April 4th, 2023

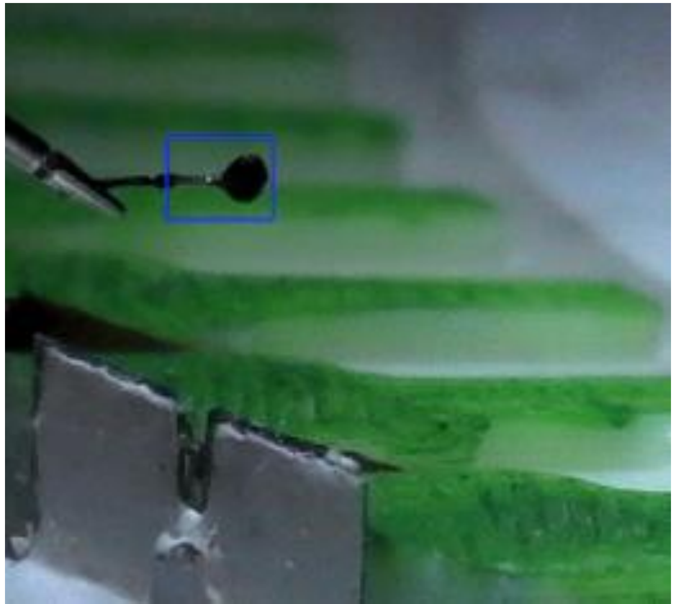
# Project Summary

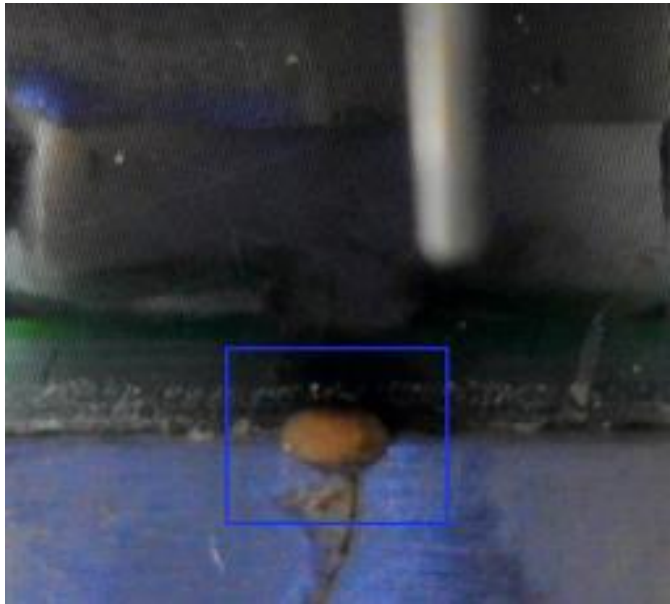
This project aims to create computer vision algorithms for the robot mosquito dissection system, an important part of continuing development. Specific aims are to develop new DL-based CV methods and integrate existing CV methods for the mosquito dissection system, which include:

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









# Deliverables: Orientation Classification (Minimum)

| Activities | Results | Status |
| --- | --- | --- |
| Review and complete the deep learning classification code. | Completed code in GitLab repository | COMPLETED |
| Generate up-to-date training data, train, and evaluate the network. Complete documentation. | Classification model on the training machine. Completed documentation in GitLab Wiki and Readme files | COMPLETED |

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

LABORATORY FOR Computational Sensing + Robotics

# Accomplished: Codes and Results

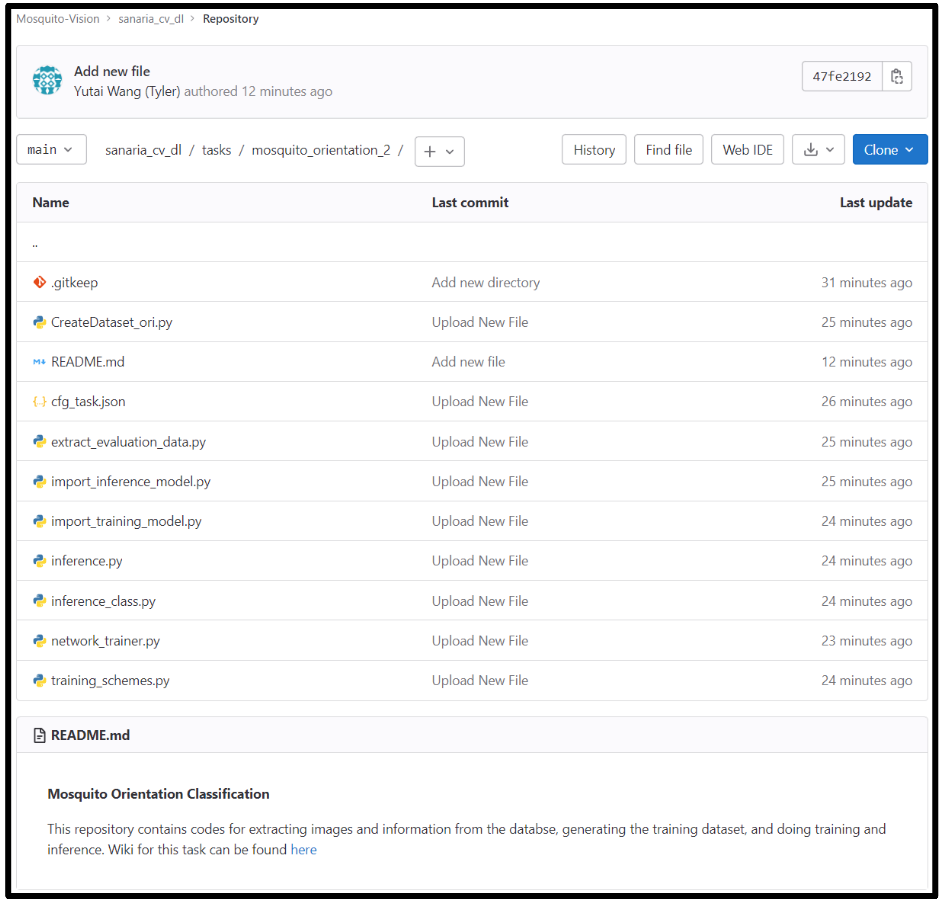
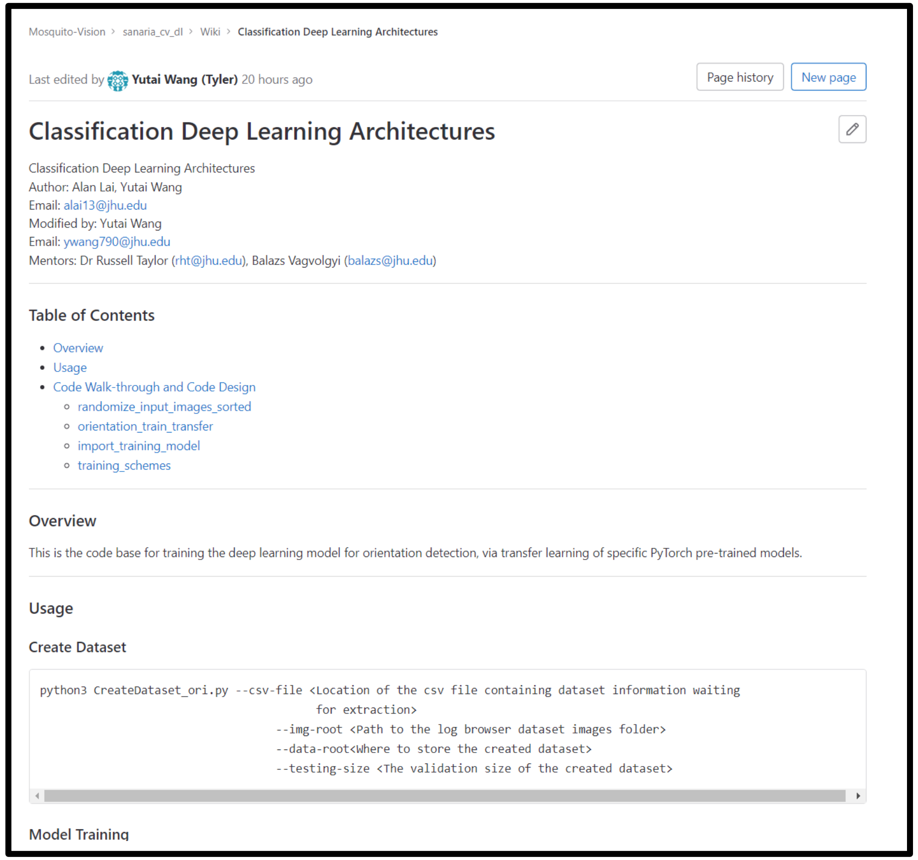



Figure 1: Completed codes repository on GitLab


Figure 2: Results obtained from the trained model on test dataset

# Accomplished: Documentation



Mosquito-Vision › sanaria_cv_dl › Wiki › Classification Deep Learning Architectures

Last edited by Yutai Wang (Tyler) 20 hours ago

Page history | New page

## Classification Deep Learning Architectures

Classification Deep Learning Architectures
Author: Alan Lai, Yutai Wang
Email: alai13@jhu.edu
Modified by: Yutai Wang
Email: ywang790@jhu.edu
Mentors: Dr Russell Taylor (rht@jhu.edu), Balazs Vagvolgyi (balazs@jhu.edu)

### Table of Contents



### Overview

This is the code base for training the deep learning model for orientation detection, via transfer learning of specific PyTorch pre-trained models.

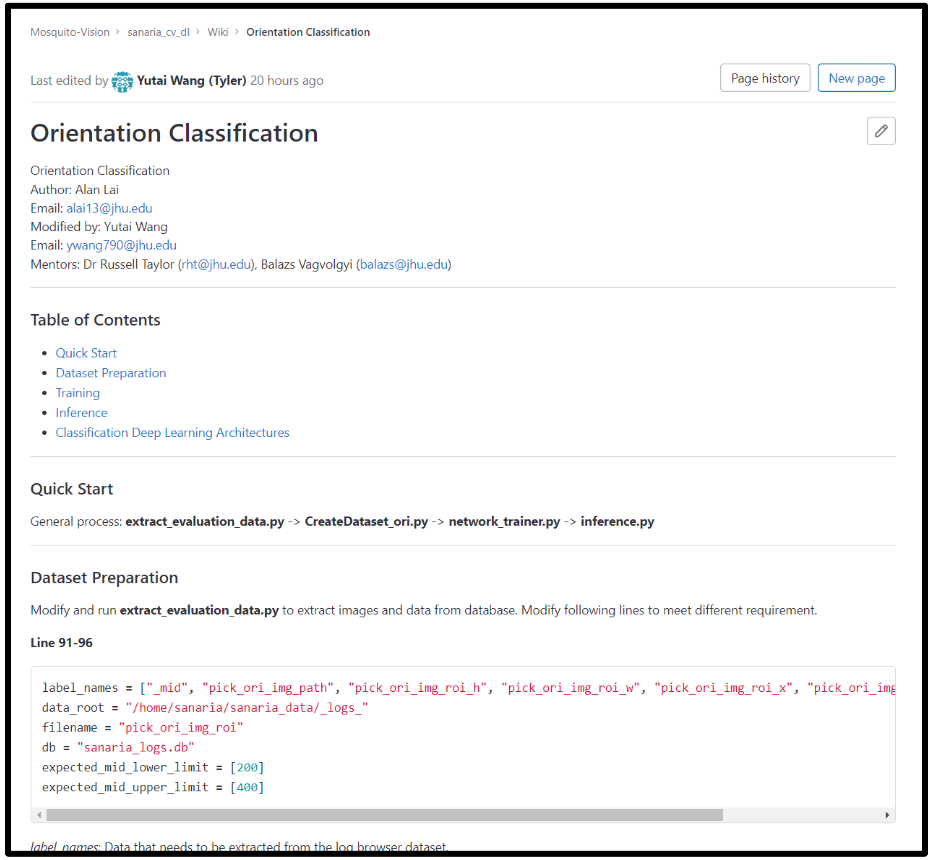
### Usage

#### Create Dataset

```
python3 CreateDataset_ori.py --csv-file <Location of the csv file containing dataset information waiting
                                         for extraction>
                                --img-root <Path to the log browser dataset images folder>
                                --data-root<Where to store the created dataset>
                                --testing-size <The validation size of the created dataset>
```

#### Model Training

Figure 3: DL-based classification architectures Wiki on GitLab



Mosquito-Vision › sanaria_cv_dl › Wiki › Orientation Classification

Last edited by Yutai Wang (Tyler) 20 hours ago

Page history | New page

## Orientation Classification

Orientation Classification
Author: Alan Lai
Email: alai13@jhu.edu
Modified by: Yutai Wang
Email: ywang790@jhu.edu
Mentors: Dr Russell Taylor (rht@jhu.edu), Balazs Vagvolgyi (balazs@jhu.edu)

### Table of Contents



### Quick Start

General process: **extract_evaluation_data.py** -> **CreateDataset_ori.py** -> **network_trainer.py** -> **inference.py**

### Dataset Preparation

Modify and run **extract_evaluation_data.py** to extract images and data from database. Modify following lines to meet different requirement.

**Line 91-96**

```
label_names = ["_mid", "pick_ori_img_path", "pick_ori_img_roi_h", "pick_ori_img_roi_w", "pick_ori_img_roi_x", "pick_ori_img
data_root = "/home/sanaria/sanaria_data/_logs_"
filename = "pick_ori_img_roi"
db = "sanaria_logs.db"
expected_mid_lower_limit = [200]
expected_mid_upper_limit = [400]
```

*label_names*: Data that needs to be extracted from the log browser dataset

Figure 4: Orientation Classification Wiki on GitLab

# Deliverables: Exudate Quality Evaluation (Minimum)

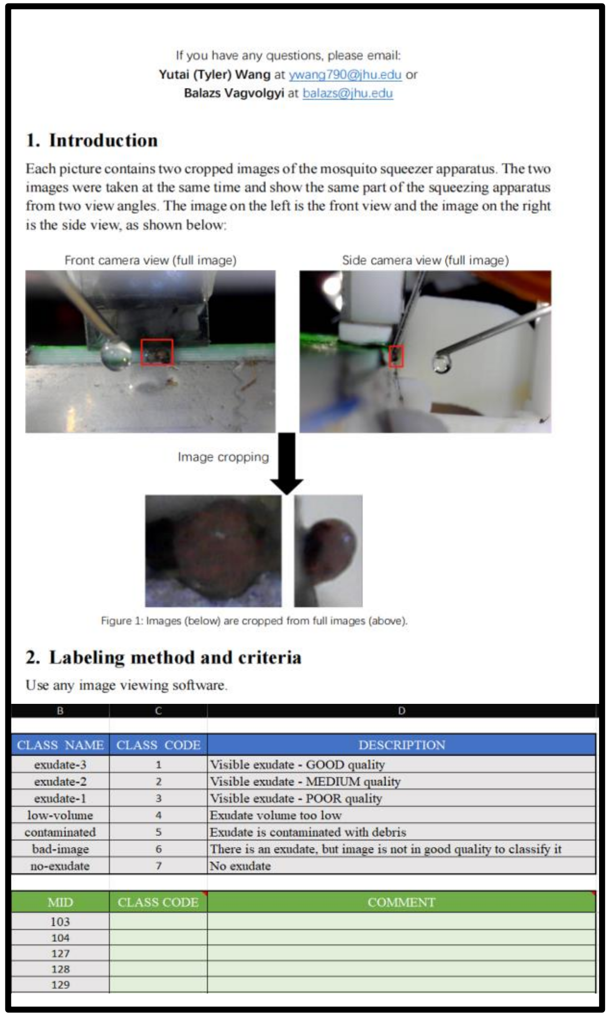
| Activities | Results | Status |
|---|---|---|
| Establish contact with Sanaria experts, review the database, and gather exudate images for training from the database. | The data set for the label was created and provided to Sanaria | COMPLETED |
| The data set for the label was created and provided to Sanaria. | Correct labels from Sanaria | Pending |
| Create training set images for deep learning training and set up the training process. | Dataset used for training | COMPLETED |
| Complete documentation for Exudate Quality Evaluation task. | Completed documentation in GitLab Wiki and Readme files | COMPLETED |
| Train, and evaluate the network. | Classification model on the training machine | Not Started |

# Deliverables: Prediction of Dissection Success (Expected)

| Activities | Results | Status |
|---|---|---|
| Get results from the Exudate Quality Evaluation task. | Classification results from Exudate Quality Evaluation | Pending |
| Create training set images for deep learning training and set up the training process. | Dataset used for training | COMPLETED |
| Complete documentation for Prediction of Dissection Success task. | Completed documentation in GitLab Wiki and Readme files | COMPLETED |
| Train, and evaluate the network. | Classification model on the training machine | Not Started |
| Investigate methods to locate specific regions on mosquito images that contribute strongest to variability in exudate quality. | Relevant code in Gitlab repository/ Documentation in Gitlab Wiki | In progress |

# Updates: Contact with Sanaria

Figure 5: Concatenated images for Sanaria to label



If you have any questions, please email:
**Yutai (Tyler) Wang** at ywang790@jhu.edu or
**Balazs Vagvolgyi** at balazs@jhu.edu

## 1. Introduction

Each picture contains two cropped images of the mosquito squeezer apparatus. The two images were taken at the same time and show the same part of the squeezing apparatus from two view angles. The image on the left is the front view and the image on the right is the side view, as shown below:

Front camera view (full image)

Side camera view (full image)

Image cropping

Figure 1: Images (below) are cropped from full images (above).

## 2. Labeling method and criteria

Use any image viewing software.

| CLASS NAME | CLASS CODE | DESCRIPTION |
|---|---|---|
| exudate-3 | 1 | Visible exudate - GOOD quality |
| exudate-2 | 2 | Visible exudate - MEDIUM quality |
| exudate-1 | 3 | Visible exudate - POOR quality |
| low-volume | 4 | Exudate volume too low |
| contaminated | 5 | Exudate is contaminated with debris |
| bad-image | 6 | There is an exudate, but image is not in good quality to classify it |
| no-exudate | 7 | No exudate |

| MID | CLASS CODE | COMMENT |
|---|---|---|
| 103 | | |
| 104 | | |
| 127 | | |
| 128 | | |
| 129 | | |

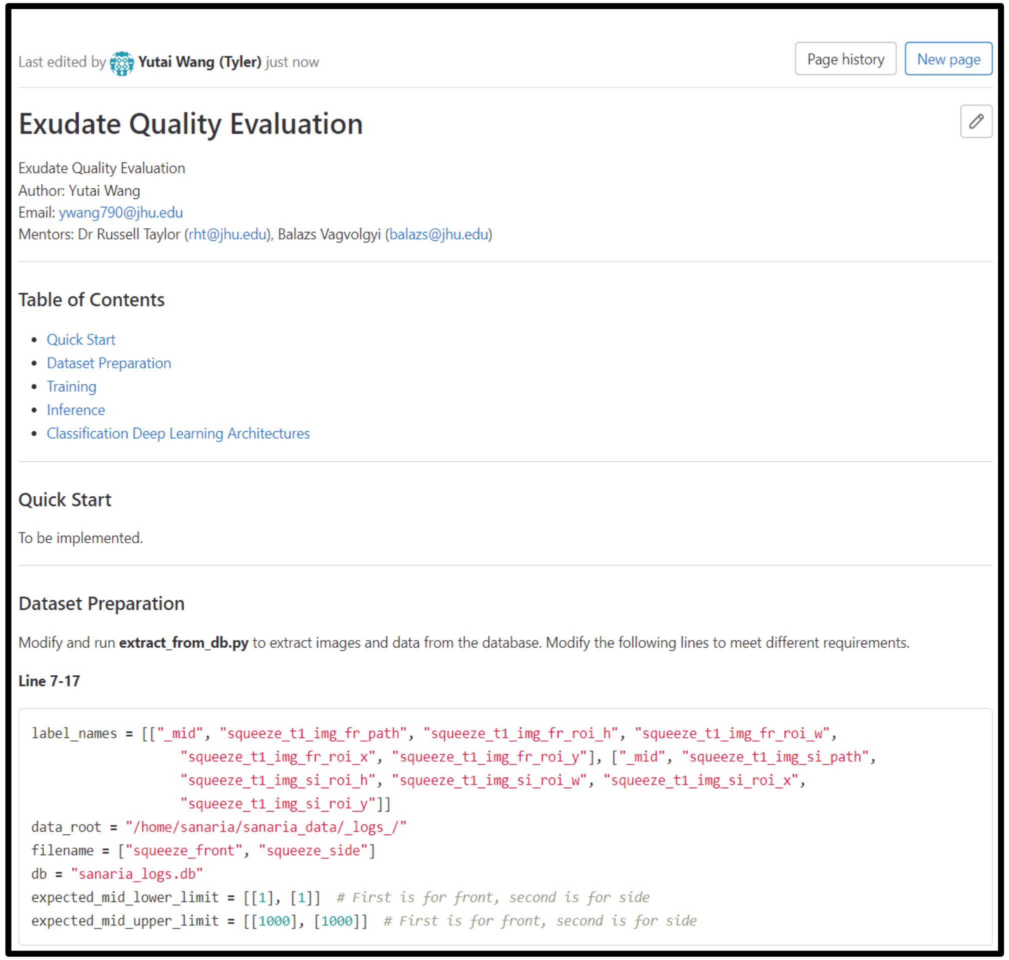
Figure 6: Documentation for the labeling provided to Sanaria

# Updates: Prepare for the training process

Figure 7: Input images for the Exudate Quality Evaluation task

Figure 8: Input images for the Prediction of Dissection Success task

# Updates: Documentation



Last edited by Yutai Wang (Tyler) just now | Page history | New page

## Exudate Quality Evaluation

Exudate Quality Evaluation
Author: Yutai Wang
Email: ywang790@jhu.edu
Mentors: Dr Russell Taylor (rht@jhu.edu), Balazs Vagvolgyi (balazs@jhu.edu)

### Table of Contents

- Quick Start
- Dataset Preparation
- Training
- Inference
- Classification Deep Learning Architectures

### Quick Start

To be implemented.

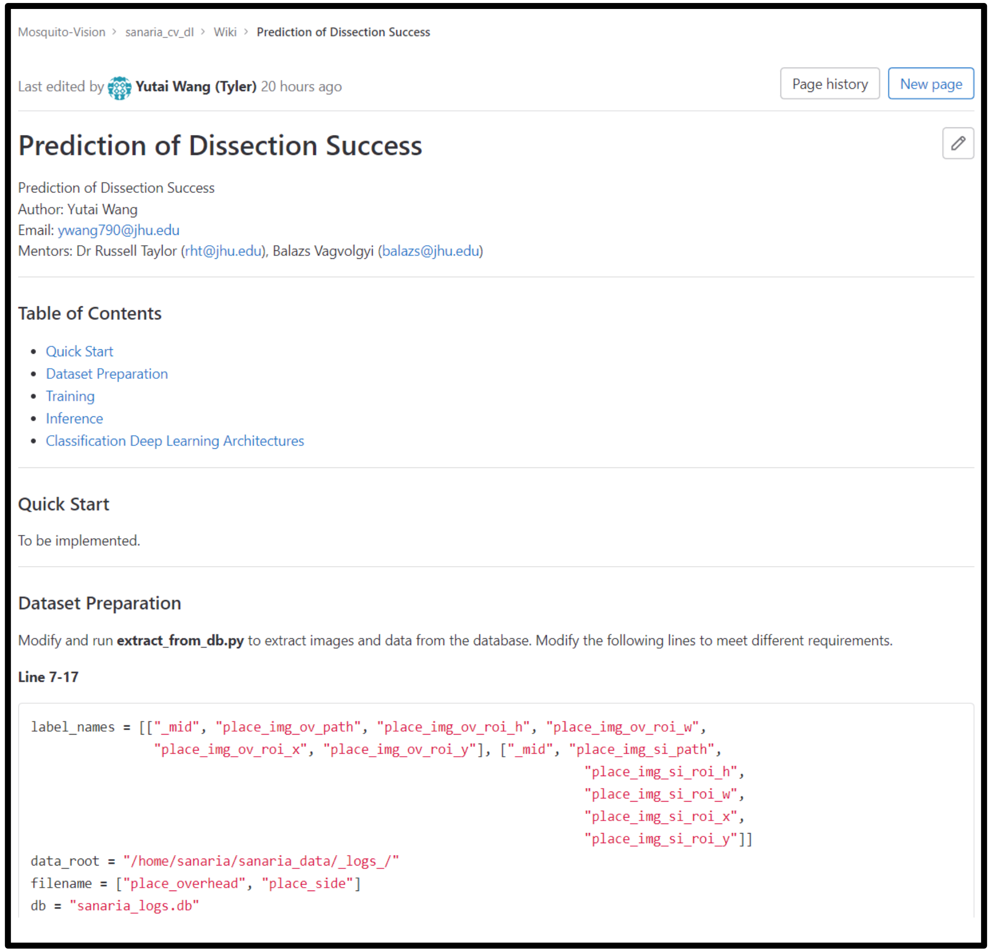
### Dataset Preparation

Modify and run **extract_from_db.py** to extract images and data from the database. Modify the following lines to meet different requirements.

**Line 7-17**

```
label_names = [["_mid", "squeeze_t1_img_fr_path", "squeeze_t1_img_fr_roi_h", "squeeze_t1_img_fr_roi_w",
                "squeeze_t1_img_fr_roi_x", "squeeze_t1_img_fr_roi_y"], ["_mid", "squeeze_t1_img_si_path",
                "squeeze_t1_img_si_roi_h", "squeeze_t1_img_si_roi_w", "squeeze_t1_img_si_roi_x",
                "squeeze_t1_img_si_roi_y"]]
data_root = "/home/sanaria/sanaria_data/_logs_/"
filename = ["squeeze_front", "squeeze_side"]
db = "sanaria_logs.db"
expected_mid_lower_limit = [[1], [1]]  # First is for front, second is for side
expected_mid_upper_limit = [[1000], [1000]]  # First is for front, second is for side
```

Figure 9: Exudate Quality Evaluation Wiki on GitLab



Mosquito-Vision > sanaria_cv_dl > Wiki > Prediction of Dissection Success

Last edited by Yutai Wang (Tyler) 20 hours ago | Page history | New page

## Prediction of Dissection Success

Prediction of Dissection Success
Author: Yutai Wang
Email: ywang790@jhu.edu
Mentors: Dr Russell Taylor (rht@jhu.edu), Balazs Vagvolgyi (balazs@jhu.edu)

### Table of Contents

- Quick Start
- Dataset Preparation
- Training
- Inference
- Classification Deep Learning Architectures

### Quick Start

To be implemented.

### Dataset Preparation

Modify and run **extract_from_db.py** to extract images and data from the database. Modify the following lines to meet different requirements.

**Line 7-17**

```
label_names = [["_mid", "place_img_ov_path", "place_img_ov_roi_h", "place_img_ov_roi_w",
                "place_img_ov_roi_x", "place_img_ov_roi_y"], ["_mid", "place_img_si_path",
                                                               "place_img_si_roi_h",
                                                               "place_img_si_roi_w",
                                                               "place_img_si_roi_x",
                                                               "place_img_si_roi_y"]]
data_root = "/home/sanaria/sanaria_data/_logs_/"
filename = ["place_overhead", "place_side"]
db = "sanaria_logs.db"
```

Figure 10: Prediction of Dissection Success Wiki on GitLab

# Extra Task: Uncalibrated Image Based Visual Servoing in Micro- and Macroscale Robotics

Work done:

- Collect data. Implement deep learning method using YOLO to detect tool tip in the Sanaria project and the screw driver tip.
- Build and adjust experimental scenarios for the screw driver tip alignment task.
- Conduct experiments for both robotics, and perform data processing.

## Applications of Uncalibrated Image Based Visual Servoing in Micro- and Macroscale Robotics

Yifan Yin, Yutai Wang, Yunpu Zhang,
Russell H. Taylor, *Life Fellow, IEEE* and Balazs P. Vagvolgyi, *Member, IEEE*

***Abstract*— We present 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. The system uses two cameras for observing the workspace and a combination of classical image processing algorithms and deep learning based methods to detect features on camera images. The only restriction on the placement of the two cameras is that relevant image features must be visible in both views. The system enables precise robot-tool to workspace interactions even when the physical setup is disturbed, for example if cameras are moved or the workspace shifts during manipulation. The usefulness of the visual servoing method is demonstrated and evaluated in two applications: in the calibration of a micro-robotic system that dissects mosquitoes for the automated production of a malaria vaccine, and a macro-scale manipulation system for fastening screws using a UR10 robot. Evaluation results indicate that our image based visual servoing method achieves human-like manipulation accuracy in challenging setups even without camera calibration.**

### I. Introduction

Precise unsupervised robotic manipulation requires accurate knowledge of the spatial relationship between the robot, the tools that are mounted on it and relevant objects in the workspace. In many applications the robot controller relies on known geometrical models or constraints to calculate its

methods require known point cloud geometry, they are ideal for tracking optical markers, such as AR tags [4][5] and 3D rigid bodies constructed from point-like markers (e.g. light emitting diodes, retroreflective spheres) [6][7][8].

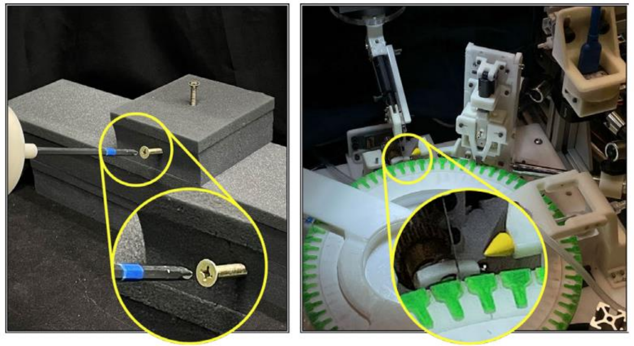
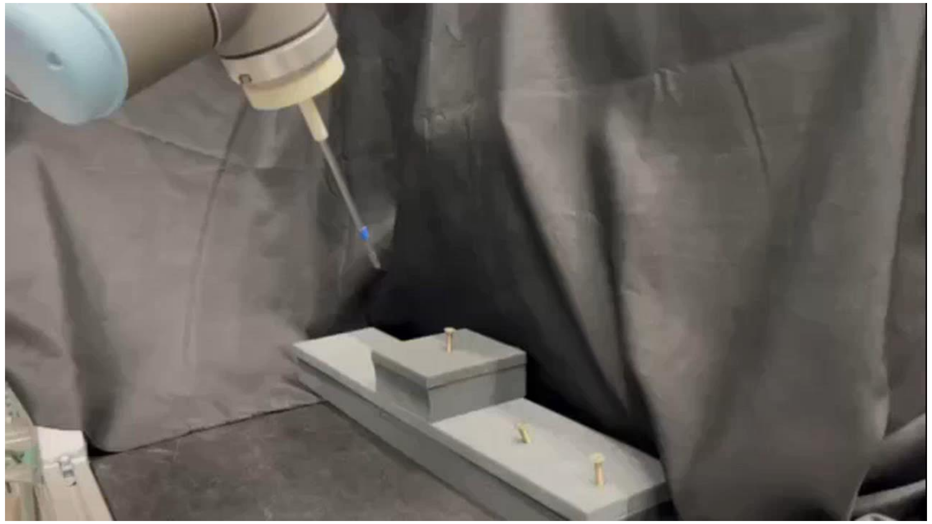
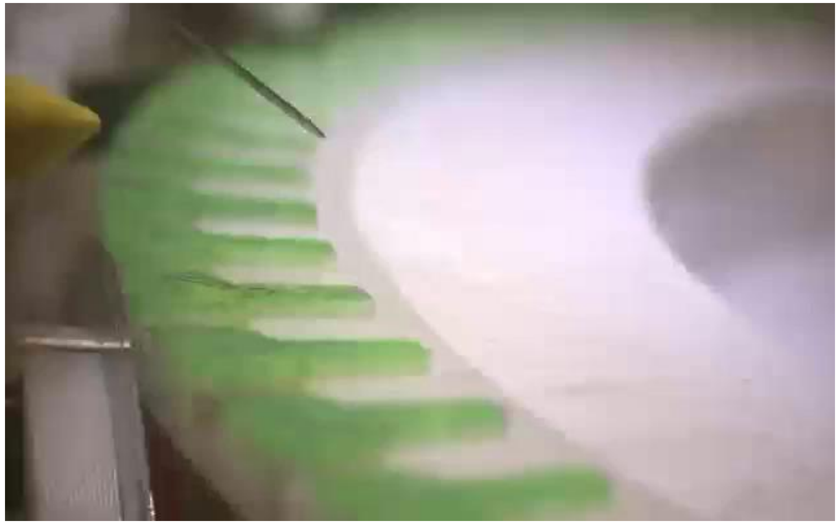


Fig. 1: (Left) Macroscale task of aligning screw driver with screw. (Right) Automated calibration of a robotic mosquito salivary gland extraction system that uses a micro-forceps for manipulating mosquitoes.

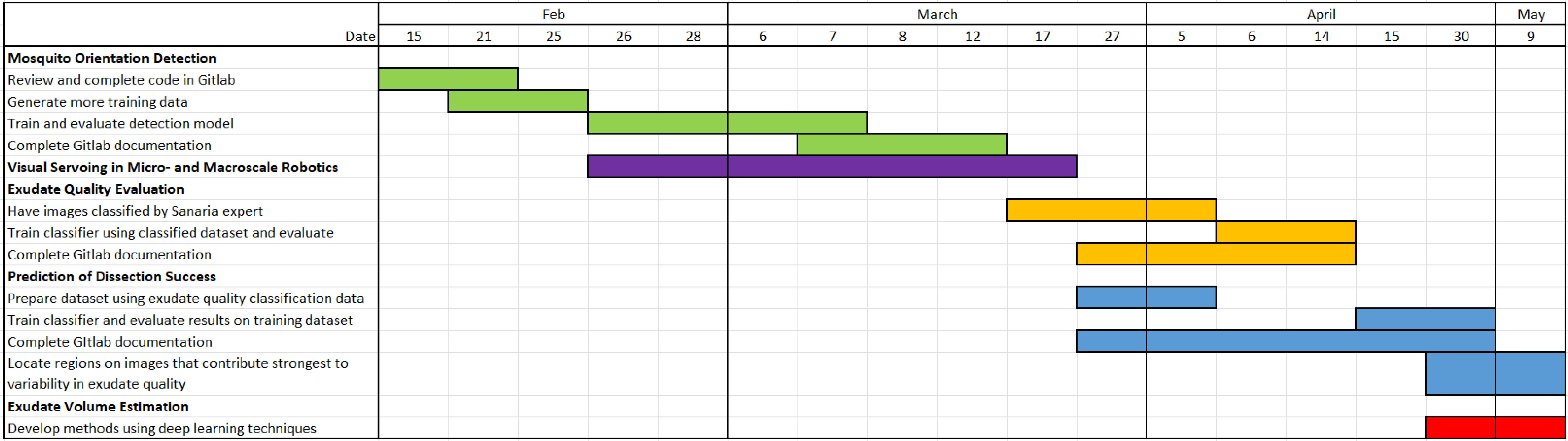
Figure 11: Submitted Paper

# Extra Task: Uncalibrated Image Based Visual Servoing in Micro- and Macroscale Robotics


Videos: Macroscale task alignment process and microscale task automated calibration process

# Updated Timeline



# 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. |

# Roles

- Yutai Wang –Team Member
- Mr. Balazs Vagvolgyi – Mentor

# Management plan

Meeting Schedule:

- Monday 9:00pm – 10:00pm: Meet with the project team over Zoom.
- Monday 11:00pm – 12:00pm: Meet with mentor in person.

Communication:
Email and text message.