

# Mosquito Vision Final Presentation



EN.601.456 Computer Integrated Surgery II  
Team 5: Akash Chaurasia, Alan Lai, Parth Vora  
Mentors: Balazs Vagvolgyi, Dr. Russ Taylor  
May 5, 2020

*Confidential*

# Background on Malaria

**>219 million**

Clinical cases<sup>1</sup>

**435,000 deaths**

In 2017<sup>1</sup>

**\$12 billion**

Loss annually in Africa<sup>1</sup>

*Confidential*

<sup>1</sup>“Sanaria’s Mission.” Sanaria, 2020, [sanaria.com/about/](https://sanaria.com/about/).

# Project Summary: Goals

Goal: To create a ROS-integrated computer vision system for mosquito detection and keypoint detection to guide an automated mosquito dissection robotic system for live malaria vaccine production

- Sanaria has developed a live malaria vaccine shown to be 100% effective in clinical trials
- Extraction of salivary glands vital for vaccine production
- Currently done manually; but automatic mosquito microdissection system (MMS) proposed

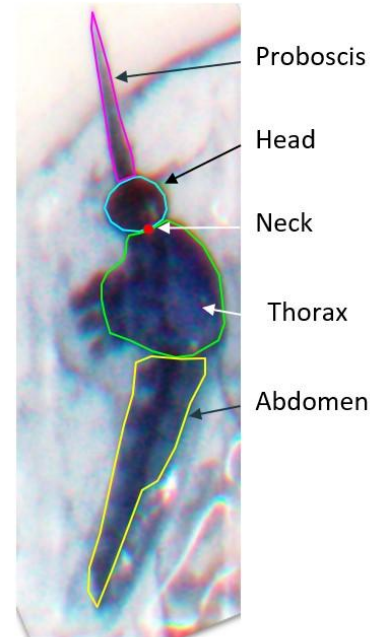


Figure 1. Parts of a mosquito

*Confidential*

# Project Summary: Process Flow

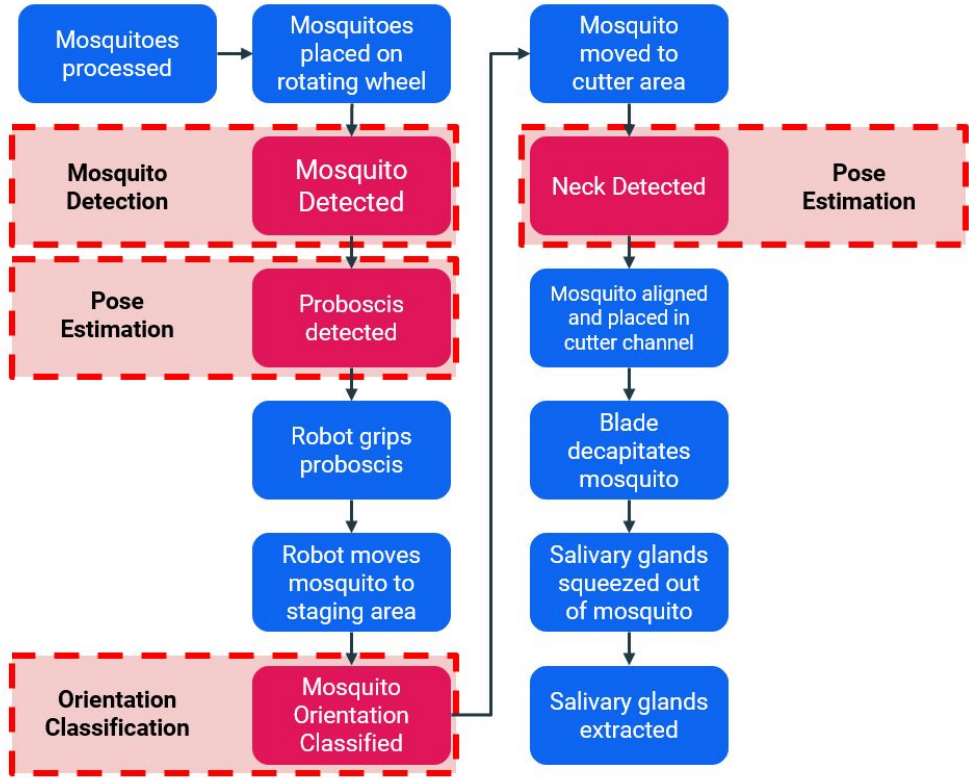


Figure 2. Process flow diagram for mosquito keypoint localization

# Deliverables Achieved

Minimum

Expected

Maximum

- Mosquito finder algorithm
- Neck finder algorithm
- Proboscis finder algorithm
- Algorithm + design documentation

In addition to minimum...

- Library creation
- Library documentation
- Integration with ROS

In addition to expected...

- Validation of CV algorithms
- Validation methods + results documentation

- Orientation algorithm
- Framework for training keypoint detection
- Documentation of design choices, algorithms, and code usage

In addition to minimum...

- Implementation of DL orientation algorithm in a ready-to-be-integrated class-based model

In addition to expected...

- Creation and validation of library to streamline process of training mosquito keypoint detection

*Confidential*  
5

# Image Processing Algorithms

- **Mosquito Finder**
  - Identify connected components in image and filter by size and orientation to find mosquitos
- **Head Finder**
  - Use template matching to find the head of the mosquito
- **Proboscis Finder**
  - Find mosquito head, identify angle of proboscis in polar coordinates, return tip and base of proboscis
- **Neck Finder**
  - Find mosquito head, neck is set number of pixels below head center

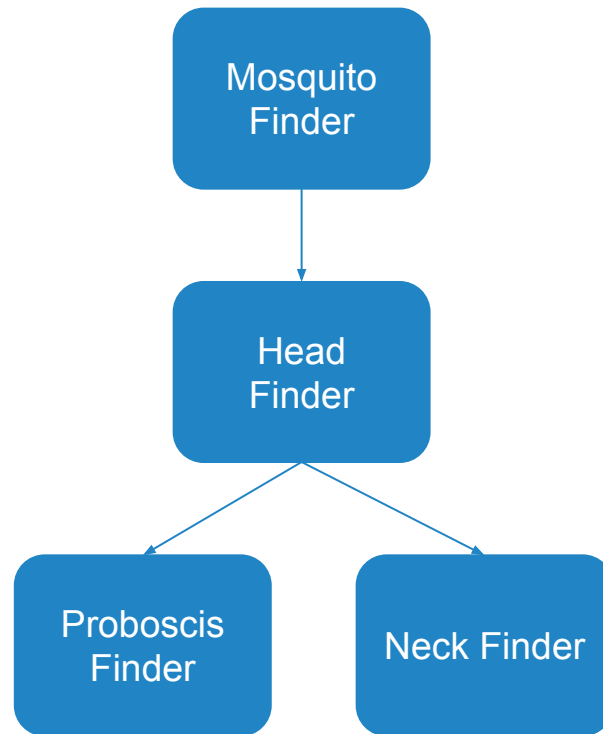


Figure 3. Flow Diagram of Image Processing Methods

# Image Processing: Results Visualized

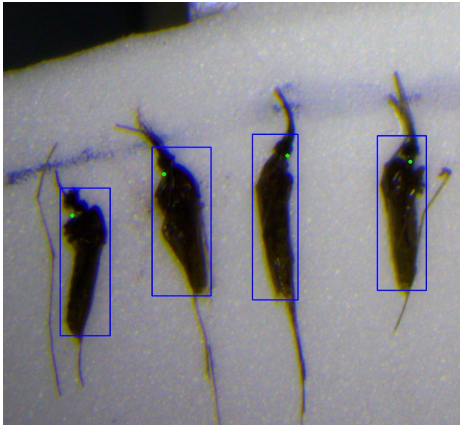


Figure 4. Successful Neck Finder

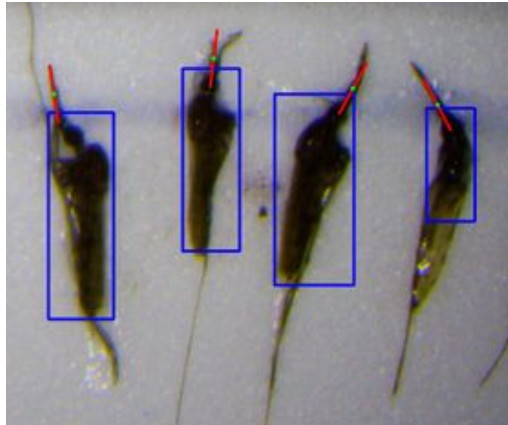
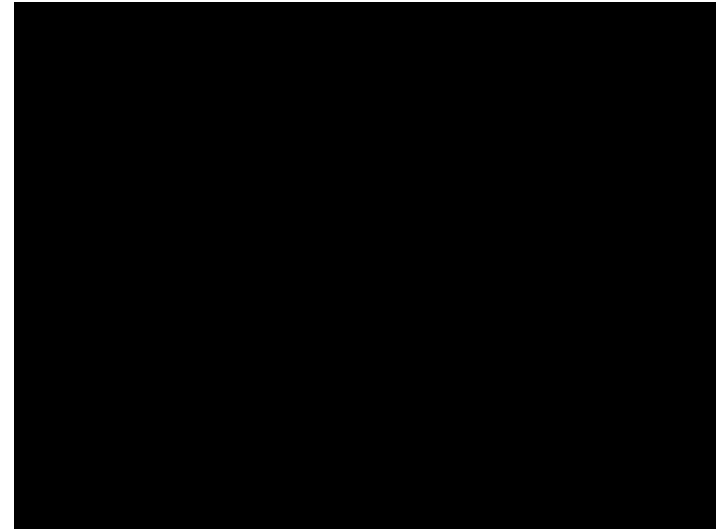


Figure 5. Successful Proboscis Finder



Successful ROS integration of proboscis finder

# Image Processing: Results Summary

Mosquito Finder		Proboscis Finder		Neck Finder	
# Mosquitos	306	# Proboscis	261	# Necks	272
# Found	291 (95%)	# Found	251 (96%)	# Found	259 (95%)
Avg. IOU	0.736	Avg Lateral Error (pixels)	3.46	Avg y-axis Error (pixels)	10.91
STDEV	0.126	STDEV	3.39	STDEV	7.62

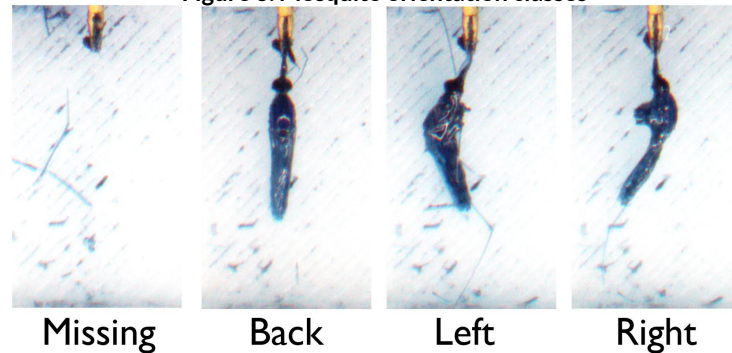


*Sample results: mosquito & proboscis finder. Blue boxes indicate detected mosquitos, red lines indicate detected proboscises*

# Deep Learning - Orientation Classification

- Transfer learning via PyTorch
  - Pretrained models on ImageNet
- Dataset of 433 images
  - Split 70-30 for training, validation

Figure 6. Mosquito orientation classes



Model	Accuracy	System	Inference (s)	Inference + Processing (s)
ResNet18	96%	CPU	0.102	0.104
		GPU	0.004	0.005
ResNet152	97%	CPU	0.588	0.590
		GPU	0.028	0.029
VGG16	94%	CPU	0.614	0.616
		GPU	0.016	0.017
DenseNet121	96%	CPU	0.297	0.299
		GPU	0.022	0.023

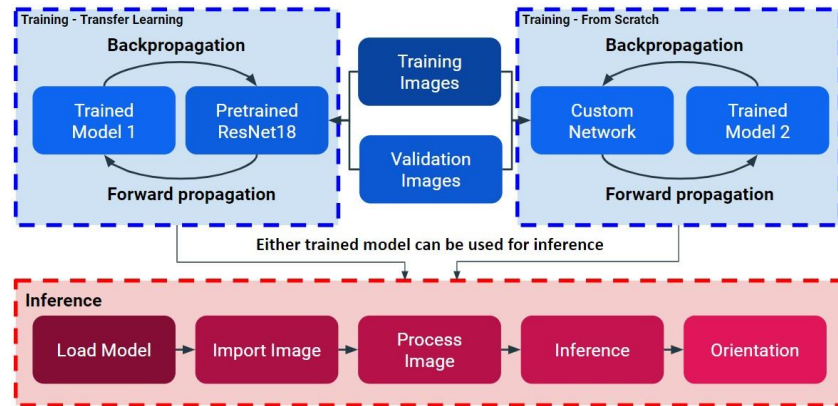


Figure 7. Flow diagram for deep learning training

# Deep Learning - Mosquito Detection

- Transfer learning via Detectron2 (Facebook AI)
  - Based on Mask RCNN
- Supports
  - Object detection
    - Bounding boxes
    - Instance segmentation masks
  - Human pose prediction
  - Semantic segmentation
  - Panoptic segmentation.

Figure 8. Detectron2 Samples



Figure 9. Training image from old setup

- Old Setup
- Training - 787
- Validation - 209

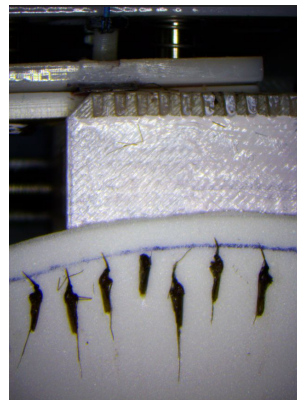


Figure 10. Testing image from current setup

- Current Setup
- Testing - 126

*Confidential*

# Deep Learning - Mosquito Detection

Validation Images

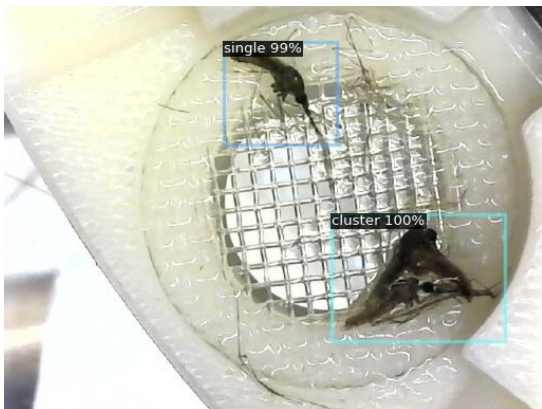


Figure 11. Mosquito detection results on old setup image

Model	mAP@IoU=0.75	mAP@IoU=0.5	Inference Time
Wu et al.	0.84	0.96	~400 ms
Mask RCNN R50 FPN	0.894	0.975	~100 ms
<b>Faster RCNN R50-C4, VOC</b>	<b>0.895</b>	<b>0.981</b>	<b>~200 ms</b>
Faster RCNN R50 FPN	0.895	0.970	~60 ms

Testing Images

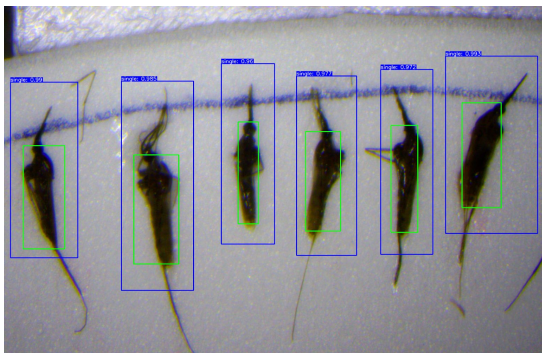


Figure 12. Mosquito detection results on current setup

Model	Total Mosquitos	# Detected	% Detected
Faster RCNN R50-C4, VOC	306	306	100%

Confidential

# Deep Learning - Pose Estimation

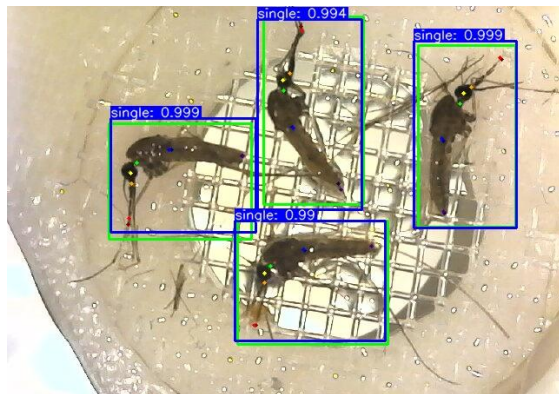


Figure 12. Pose Estimation results on old setup image

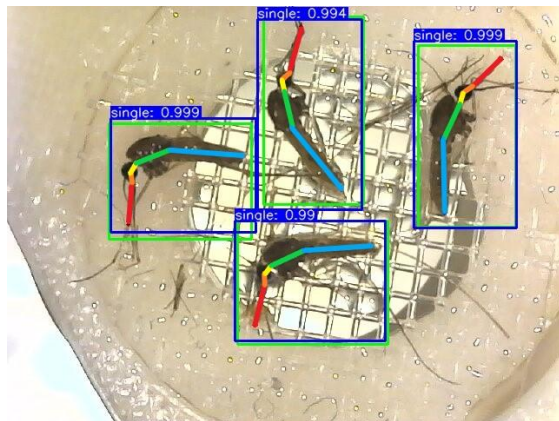


Figure 13. Pose Estimation results on old setup image with keypoints connected

Model	Wu et al.	R50_FPN_3x Detectron2	
RMSE	Validation	Validation	Testing
Proboscis Tip Error	7.1	7.17	22.15
Proboscis End Error	2.2	1.91	16.64
Head Error	1.6	1.40	
Neck Error	2.0	1.89	44.70
Thorax Error	2.6	2.71	
Abdomen Error	3.0	3.68	

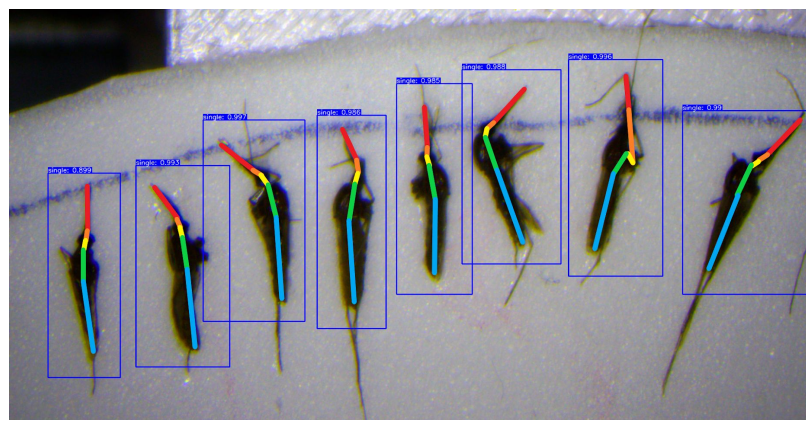


Figure 14. Pose Estimation results on new setup image with keypoints connected

# Image Processing vs Deep Learning

<b>Mosquito Detection</b>			<b>Image Processing</b>	<b>Deep Learning</b>
# Mosquitos	306	# Found	291 (95%)	306 (100%)
Runtime (ms)			~29	200
<b>Proboscis Detection (261 total)</b>			<b>Image Processing</b>	<b>Deep Learning</b>
# Found (threshold = 30 pixels)			251 (96%)	229 (88%)
Avg Lateral Error (pixels)			3.46	8.25
Runtime (ms)			~41	106
<b>Neck Detection (272 total)</b>			<b>Image Processing</b>	<b>Deep Learning</b>
# Found (threshold = 30 pixels)			259 (95%)	248 (92%)
Avg y-axis Error (pixels)			10.91	8.49
Runtime (ms)			~4.7	106

# Future Work

## Image Processing

- ▷ Further image processing for dissection pipeline validation (i.e. verification of decapitation)

## Deep Learning

- ▷ Compilation of image labels into Detectron2-compatible format
- ▷ Exploration of other pose estimation models

Thank you! Questions?

# Appendix: Project Documentation

- Deep learning documentation: [link](#)
- Image processing documentation: [link](#)

## Overview

This repository contains algorithms used for a robotic dissection system used to extract the salivary glands of mosquitos. Specifically, the contents of these directories contain the code for various programs used in the robotic process, such as automated mosquito finding, neck detection, and proboscis detection.

## Usage

With all the source files located in this root directory, a remote directory can be used to compile and run the code. Steps are as follows:

- Go to the desired directory for compilation
- Run: `cmake <path to these source files>`
- Run: `make`

## Mosquito Finder

```
bool FindMosquitoes(const cv::Mat& image,
                   const cv::Rect& roi,
                   std::vector<cv::Rect>& mosquito_bounding_boxes)
```

## Arguments

- **image**: whole image to be used to find mosquitos. Must be in OpenCV Mat format.
  - all return values will have coordinates relative to this original image.
- **roi**: OpenCV Rect indicating the region of the entire image in which to search for mosquitos.
- **mosquito\_bounding\_boxes**: reference to a list of cv::Rects to which all found mosquito bounding boxes will be appended.

# Appendix: Management Summary

Team Contributions	
Akash Chaurasia	Image processing algorithms (mosquito finder, proboscis finder, neck finder), data collection/labeling, algorithm validation, function documentation, ROS integration
Parth Vora	Image processing algorithms (proboscis finder, neck finder), data collection, validation scripts (mosquito finder, proboscis finder, neck finder), function documentation
Alan Lai	Deep learning algorithms (Orientation classification, mosquito detection with deep learning, pose estimation with deep learning), deep learning documentation, deep learning training framework library

- Validated the feasibility of using computer vision algorithms for guidance of robotic system for mosquito dissection
  - In terms of accuracy, speed, and ability for integration
- Established image processing as the optimal method for pose estimation
- Established deep learning as the optimal method for mosquito detection and orientation classification

# Appendix: Lessons Learnt

- Evaluation metrics need to take into account problem context
- Design specifications must also take into account problem context
- Error handling and error checking are vital for the correct operation of a integrated system

# References

- [1] Wu, H., Mu, J., Da, T., Xu, M., Taylor, R. H., Iordachita, I., & Chirikjian, G. S. (2019). Multi-mosquito object detection and 2d pose estimation for automation of PfSPZ malaria vaccine production. In 2019 IEEE 15th International Conference on Automation Science and Engineering, CASE 2019 (pp. 411-417). [8842953] (IEEE International Conference on Automation Science and Engineering; Vol. 2019-August). IEEE Computer Society. <https://doi.org/10.1109/COASE.2019.8842953>
- [2] Wu, Y., Kirillovm A., Massam F., Lo, W.Y., Girshick, R. (2019). Detectron2. <https://github.com/facebookresearch/detectron2>
- [3] H. Phalen, P. Vagdargi, M. Pozin, S. Chakravarty, G. S. Chirikjian, I. Iordachita, and R. H. Taylor, “Mosquito Pick-and-Place: Automating a Key Step in PfSPZ-based Malaria Vaccine Production”, in IEEE Conference on Automation Science and Engineering (CASE), Vancouver, BC, August 22-26, 2019. pp. 12-17.
- [4] M. Schrum, A. Canezin, S. Chakravarty, M. Laskowski, S. Comert, Y. Sevimli, G. S. Chirikjian, Stephen L. Hoffman, and R. H. Taylor, “An Efficient Production Process for Extracting Salivary Glands from Mosquitoes”, arXIV, 2019, <http://arxiv.org/abs/1903.02532>.
- [5] H. Wu, J. Mu, T. Da, M. Xu, R. H. Taylor, I. Iordachita, and G. S. Chirikjian, “Multi-mosquito object detection and 2D pose estimation for automation of PfSPZ malaria vaccine production”, in IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, August 22-26, 2019.
- [6] M. Xu, S. Lyu, Y. Xu, C. Kocabalkanli, B. K. Chirikjian, J. S. Chirikjian, J. Davis, J. S. Kim, I. Iordachita, R. H. Taylor, and G. S. Chirikjian, “Mosquito Staging Apparatus for producing PfSPZ Malaria Vaccines”, in IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, August 22-26, 2019.