Holcombe Department of Electrical and Computer Engineering
Clemson University

Final Report:
Automation of a *Drosophila melanogaster* Sorter, Loader, and Climbing Assay

Spring, 2026

Team 8 – Fly Sorter, Loader, and Climbing Assay

Dylan Britch, Ainara Garcia, Rivers Henderson, Camren Khoury, Will McGlone, Megan Mcnuer, Avi Patel

Executive Summary

When researchers at the Institute of Human Genetics are handling flies, they are initially in vials with unsorted (male and female) flies, potentially dead flies, and food scraps. These vials are not the ideal conditions for performing research, therefore the viable flies must be sedated, sorted, and placed into new clean vials. This process, often performed tens of vials at a time, is often time consuming, very precise, but mundane. After the flies have been hand sorted into vials of 10 flies each, they are tapped on a surface (so they all fall to the bottom of the vial), and recorded climbing the walls of the vial. This process is inefficient, time consuming, and presents infinite human error variables, tough to control, making research even more difficult to reproduce.

We propose a single-axis mechanical robot to reduce the redundancy of these tasks, minimize human error, and allow researchers to spend their time performing much more meaningful tasks. In this approach, the user is to dump a vial of sedated flies into the channel, and wait for the report generated to arrive to their inbox. In the meantime the robot (1) captures images of the flies in the channel and identifies their coordinate point (2) using motor controls the vacuum pump moves to the coordinate point of a fly and vacuums it up (3) places the fly into the sorting chamber (4) sorts the fly male or female (or unidentified) (5) places it into the appropriate vial (6) vibration motor shakes the flies to the bottom of the vials (7) records climbing assay (8) and share the report generated with the user.

The Senior Design 2 Team 8 collaborated with the Institute for Human Genetics to solve a long-standing problem: how do we sort flies faster? It is industry standard that a scientist waste their day sorting flies to perform research. With the three-part-3-camera design, Team 8's goal is that a scientist can sedate a vial of flies, dump it into the channel of their mechanical robot, and walk away. Our design is modular, customized, cost effective, and most importantly, perfectly designed for the user. At the Institute for Human Genetics, their original fly vials contain 40-50 flies. With our mechanical robot, we seek to sort 4 flies per minute, with 99% confidence in sex, and record the climbing assay based on the user's specified timeframe. This is done with 3-d parts, 3 motors, 3 cameras, custom PCB design, and a Raspberry Pi.

Abstract

The Senior Design 2 Team 8, sponsored by the Institute for Human Genetics, created a mechanical robot that sorts a vial of 40-50 flies and performs a climbing assay. This report seeks to explore the hardware components, including Raspberry Pi and PCB design, software components, including the GUI, YOLO model, and climbing assay, and 3-D printed and mechanical parts to accomplish this task, as well as explain each component and decisions made along the way.

Introduction

The Senior Design 2 Team 8 collaborated with the Institute for Human Genetics to solve a long-standing problem: how do we sort flies faster? It is industry standard that a scientist waste their day sorting flies to perform research. With the three-part-3-camera design, Team 8's goal is that a scientist can sedate a vial of flies, dump it into the channel of their mechanical robot, and walk away. Researchers can now spend more of their time doing analysis, comparing results, and reproducing tests, rather than sedating and sorting flies for the majority of their time. By reducing the redundancy of tasks, and allowing them to have an experiment complete every 10-15 minutes, the number of experiments performed in a day goes from being one every hour, to up to 4x that amount. Additionally, as the lab grows, and interest in this project grows, having accessible parts, modular design, and a 3-D printer, the table can be recreated to have multiple experiments running at a time, further improving performance and decreasing the delay of results.

Our design is built around our previous work experience. Many members of the team have had internships, research background, and strong course curriculum to aid in the success of the project. Students in Team 8 have completed coursework in Solid Works, used for the designing and modeling of this project's parts, Circuits, Electronics, and PCB research, all valuable for the connecting, wiring, and and electrical design of the project. Students also had experience with Machine Learning, creating data bases, training models, and Linux, which were all deeply valuable in the sorting and GUI implementation. While all the individual skills added great value to the project, the Team's strongest skillsets were learned in Senior Design 1, where each member participated in a different project, took leads on hardware, software, or mechanical components, uniquely contributing to their success in the Spring 2026 Senior Design 2 Course and IHG-Sponsored Project.

Problem Definition

Genetics research involving *Drosophila melanogaster* (fruit flies) often requires repetitive manual tasks such as sedating flies, sorting flies, transferring them into vials, and conducting assays. Oftentimes, these simple tasks are time-intensive, boring, and very prone to human error. On a daily basis researchers at the Institute for Human Genetics must manually sort up to 100 vials of flies a day, sort, count and group them, only then to begin their research and assays. The manual and time-consuming nature of these tasks is a bottleneck in human genetics research, the workflow limits efficiency and increases the chances of human error (from inconsistency in sorting, sedating, counting, and timing), all of which makes this research difficult to publish and reproduce.

We aim to tackle this problem by presenting a mechanical-automated design, able to reliably handle, sort flies, and track assay measurements. Research groups in the past have tried to automate this workflow with little to no success. The hardware, design, and software background required to build the desired workflow requires a completely different set of skills than actually performing the tasks manually. Additionally, the handling of live flies introduces many constraints such as time, anesthesia, cooling, all to ensure flies stay alive and able to climb the desired vials for the assay. The defining problem at hand is to reduce the redundancy of tasks completed by researchers, increase the volume of flies handled, sorted, and used in experimentation, to allow for more assay analysis, performance and reproduction of results.

To effectively address the problem at hand, the system has to satisfy a combination of requirements, performance benchmarks, and practical constraints that reflect the needs of the Institute for Human Genetics. In terms of performance, the system must achieve a sorting throughput of approximately four flies per minute while maintaining a classification accuracy in the range of 80–99%. The mechanical movement system, including the linear gantry, must operate with positional accuracy within 5% of the desired location to ensure reliable fly transfer between stages. Functionally, the objective was to create an automated machine to identify, sort, and capture drosophila files from a random group based on characteristics such as male/female initially. The automated machine must allow for loading and unloading of vials that will hold 10 sorted flies each, then perform an assay measuring the distance traveled of each fly. The computer vision component must be trained on a dataset of at least 1000 labeled fly images to achieve reliable classification performance. During intermediate development stages, the system is expected to maintain a pick-and-place error rate below 20%, with improvements targeted through iterative refinement. In addition, the system must include a graphical user interface (GUI) that allows users to control the system and monitor its operation; more specifically: start, see progress, validate or fix errors, and generate a report tracking fly distance.

Biologically, the system must handle flies without causing physical damage and must maintain their viability for accurate assay results (no damage to legs or wings), additionally requiring careful consideration of sedation methods such as CO_2 or cooling to avoid the chances of flies dying 48 hours after the assay is complete. Mechanically, the system must be easy to use, replicatable, and small-enough to place on a lab benchtop. From an electrical and computational perspective, the system must run computer vision algorithms within the limited processing and memory resources of a Raspberry Pi. Beyond each individual subsystem, the performance of the entire system depends on the successful integration of all components into a complete operational pipeline, beginning with fly detection in the channel and ending with assay completion and report generation. Finally, the project is constrained by time and money, as the entire system must be designed, built, and demonstrated within a single academic semester, culminating in a functional prototype by April 15th, and to remain under the $600 proposed budget.

Final Design

We propose a single-axis mechanical robot to reduce the redundancy of these tasks, minimize human error, and allow researchers to spend their time performing much more meaningful tasks. In this approach, the user is to dump a vial of sedated flies into the channel, and wait for the report generated to arrive to their inbox. Figure X shows the subsystems in four parts: (1) Channel + Vacuum: Where (a.) captures images of the flies in the channel and identifies their coordinate point (b.) using motor controls the vacuum pump moves to the coordinate point of a fly and vacuums it up (c.) places the fly into the sorting chamber (2) sorts the fly male or female (or uncertain) and (3) places it into the appropriate vial (a) vibration motor shakes the flies to the bottom of the vials (4) records climbing assay (b) and share the report generated with the user.

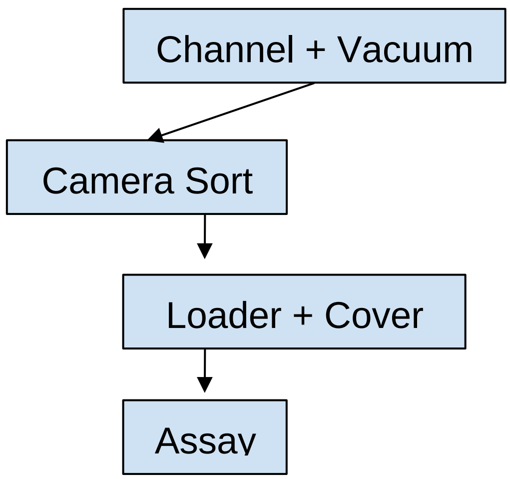

Figure 1 : Flow Chart of 4 subsystems integrated

3-D Prints

The final 3-D print design (found in Figure 2) for this project is a modular table. The table holds parts together in their necessary configurations, but since the parts are modular they can be independently redesigned and integrated into the design, which also lends itself to testing and future development. The table top is made of three major sections: the channel, sorting camera, and assay section.

ITEM NO.	PART NUMBER	ITEM NO.	PART NUMBER
1	Table 3.0	8	Pi Shelf
2	Camera Box	9	Channel Camera Stand PARTS ASSEMBLY
3	Camera Box Top	10	Assay Tube Holder 3.0 Assembly
4	Channel	11	Assay Camera Stand Total Assembly
5	CO2 Attachment	12	Assay Camera Box 2.0
6	Stepper Holder	13	LED Holder
7	PCB Stopper 3.0	14	Electric Motor Holder Assembly

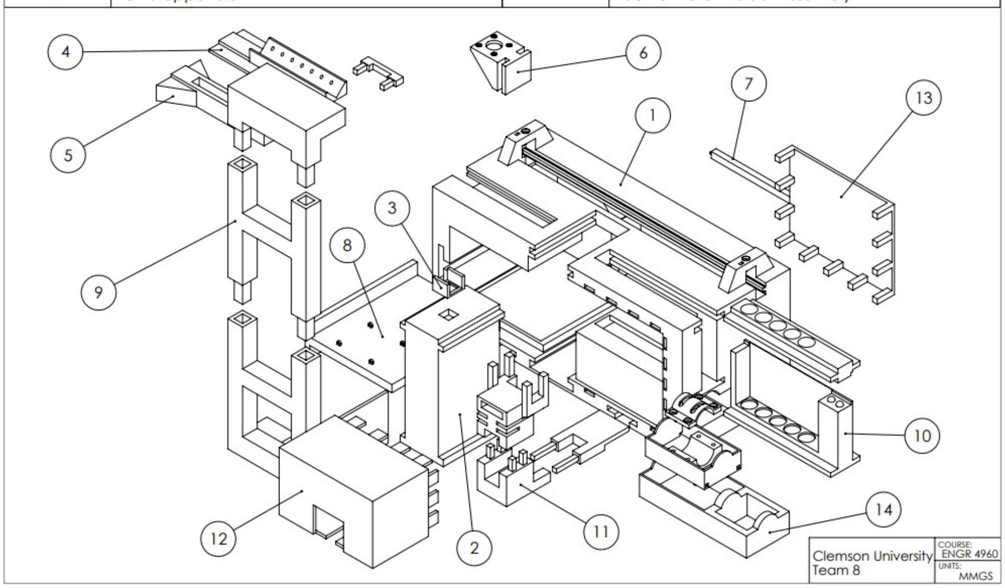

Figure 2 : Exploded View of Final Design

The channel is designed to hold the flies to be picked up by the vacuum on the gantry plate using small dimples in the bottom to facilitate isolation, and it houses an internal manifold that supplies CO_2 into the channel to keep the flies asleep via small vent holes. The channel section is made up of the channel (Figure 2 Item No 4), CO_2 box (Figure 2 Item No 5), and channel camera stand (Figure 2 Item No 9). The CO_2 box is under the channel, and it feeds CO_2 from the fitting on the outside of the table into an interface between the channel and CO_2 box. The channel camera stand holds the camera at a position that allows the whole channel to be in the camera's view and determines if there are flies left in the channel.

The fly imaging area includes the microscope camera lighting box (Figure 2 Item No 2) and the upper shade box that supports a small acrylic square (Figure 2 Item No 3). The microscope housing box consists of a solid section and a horizontal lid for internal access. The inside of the box features a negative space fitted to the Raspberry Pi high quality camera with microscope lens so that it can be pressed into place at near target focal length. It also features two side cutouts, one at the bottom for the camera ribbon cable, and a large section higher up to allow for manual focusing of the camera by rotating a section of the lens. From that opening and higher, the camera is functionally lightproofed from the outside and internally lit by an array of LEDs in the upper most section just below the table top. In essence, the camera is held facing upwards at the optimal focal length, where a fly will be dropped onto the acrylic plate housed by the shade box where it will be analyzed in highly controlled lighting conditions. The shade box also mates with the nozzle assembly that features an extended lid to the side, such that it can move out of the way once a fly was placed and effectively seal the box for complete light control. This is very important later on for the YOLO model confidence level as well as successful segmentation.

The assay section is made up of the assay tube holder and cap (Figure 2 Item No 10), the eccentric motor holder (Figure 2 Item No 14), the assay camera stand (Figure 2 Item No 11), and the assay camera box (Figure 2 Item No 12). The assay tube holder and cap holds the assay tube securely enough that they will not fall out, but loosely enough that the eccentric motor can still vibrate the tubes. The assay tube cap has a small funnel that ends in a small straw piece on the underside of the cap over each tube. The straw piece prevents flies from being able to crawl or fly out of the funnel. There are also divots in the assay tube cap so that the cap sits down onto the lip of the assay tubes, holding them in place. The eccentric motor holder holds an eccentric motor under the assay tube holder. It allows the vibrations of the eccentric motor to tamp flies to the bottom of the tubes at the start of an assay. The eccentric motor holder is made of three parts, the dampening holder, the bottom, and the clip. The bottom sits inside of the dampening holder, with a gap in between the two where dampening material is installed. This project used foam sheets as dampening material in between the dampening holder and the bottom. The eccentric motor sits in the cylindrical groove in the eccentric motor holder bottom, and the clip sits on top of the eccentric motor and is screwed into the screw holes. This allows the eccentric motor to be held securely within the table. The parts of the eccentric motor holder also have holes that allow the wires for the eccentric motor holder to be routed out toward the center of the table. The assay camera stand holds the assay camera centered on the assay tubes that hold the sorted flies (all of the tubes excluding the trash tube). The assay camera box slides into the assay camera stand and blocks environmental light around the assay camera. It has an extension piece that covers the end of the assay tubes to block the light from the sorting camera box. The assay camera box allows a more uniform lighting setup so the assay can be more consistent regardless of the system's location.

Other parts are the stepper motor holder (Figure 2 Item No 6), PCB Retainer (Figure 2 Item No 7), the Pi shelf (Figure 2 Item No 8), and the LED Holder (Figure 2 Item No 13). The stepper motor holder simply holds the stepper motor at the end of the rail to connect to the belt, which allows the gantry plate to move along the rail. The table has a slot in the left side of the table that the PCB slides into. The PCB retainer slides into the slot behind the PCB to prevent it from sliding out. The Pi slides into the left side of the table and holds the Pi securely. It has tabs on it with screw holes spaced apart so that the Pi case can be screwed into the shelf. There is also a hole in the shelf to feed the ribbon cable through and down to connect to the PCB.Finally, the LED Holder slides into the back of the table on the right side and holds the LEDs behind the assay tube holder. This allows the assay tube holder to be backlit when the assay runs, which also helps maintain consistent lighting.

Mechanical Design

After many iterations, rapid prototyping, and failure analysis, a final design was reached in the linear gantry and vacuum nozzle. A 1-D gantry is designed with four wheels mounted to a plate with a 2mm pitch belt running through slits, secured via brass clamps. Tension is maintained via a screw on the right side of the gantry. The plate houses an arm that holds the vacuum tube above the channel. This arm allows a variety of nozzles to be fit into it.

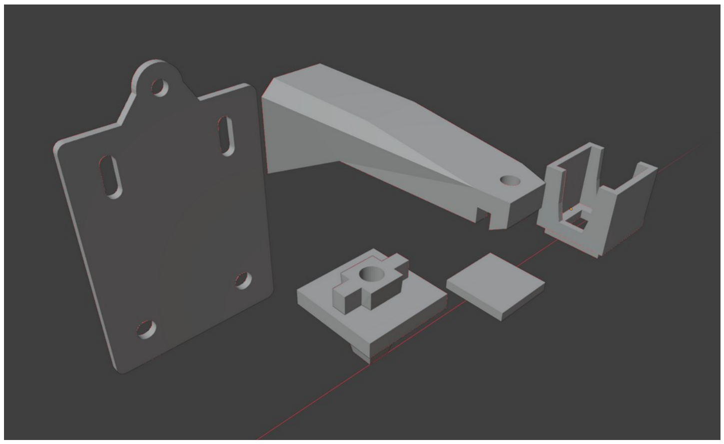

Figure 3: Image of Gantry Plate, Arm, Nozzle Connector, and Sexing Camera Slot

The final nozzle design is a 19.5mm length, with donut structure 2.0mm diameter nozzle that includes a steel screen to prevent flies from entering the system. The screen was cut to size and secured via super glue. A white plate cover was designed to surround the nozzle to provide a more consistent image background when capturing the sexing image.

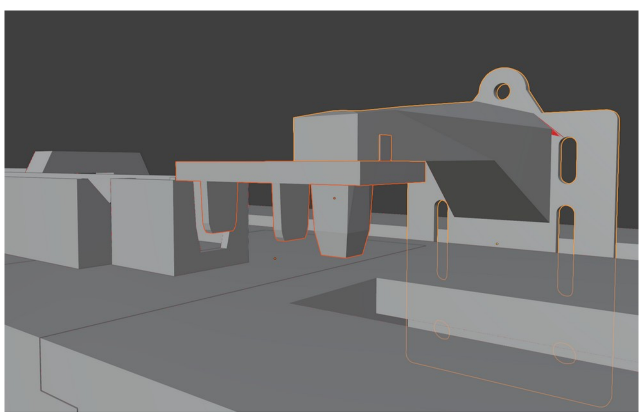

Figure 4: Image of Nozzle Integrated with Gantry

The channel features small hills and ridges and a capacity to hold ~25 flies effectively. The channel design allows for a brush or similar device to distribute flies more evenly. Additionally, holes are present on the sides of the channel to allow for carbon dioxide exposure to automatically sedate the flies.

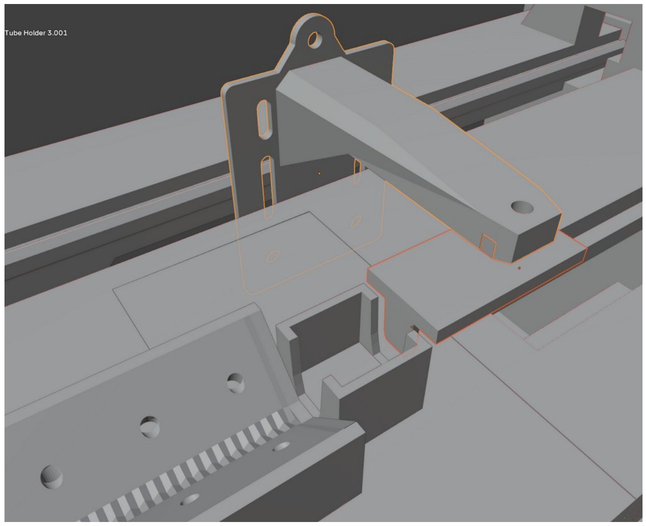

Figure 5: Image of Channel in Full Design

Most failures are experienced in carrying the fly from the channel to the sexing camera. An overall movement completion rate of 86% is achieved in this nozzle/gantry/channel combined design.

Computer Vision

The computer vision subsystem can be broken down into the interactions between a Raspberry Pi 5 and three hardware components: a Logitech Brio 100, a Raspberry Pi High Quality Camera, and an EMEET SmartCam C960. To briefly discuss the interactions of these components, the Logitech Brio 100's primary purpose was to send images of the channel portion of our design to the raspberry pi so that the relative x-position of a fly could be determined. The software portion of this detection was split into three steps. The first step was to obtain a clear background image of the channel without flies. The next step was to calibrate the channel. The left and right portions of the channel were selected so a particular region of interest could be identified. Several parameters were tuned during the testing stage of this software, including the exact width of the channel, the band width in which the inner portion of the channel would detect flies within, and the relative threshold to determine whether an object present in the channel is noise or an actual fly. Once these parameters were tuned, the third stage was determining the location itself. The code for this was fairly straightforward – the detector uses background subtraction to isolate dark fly blobs, filters out bad blobs by size and shape, finds each fly's centroid, and reports the fly count and x-position along the channel in millimeters. It also saves an annotated image, a binary mask, and a JSON result so the GUI or backend can display the result and send the next pickup location to the motion system.

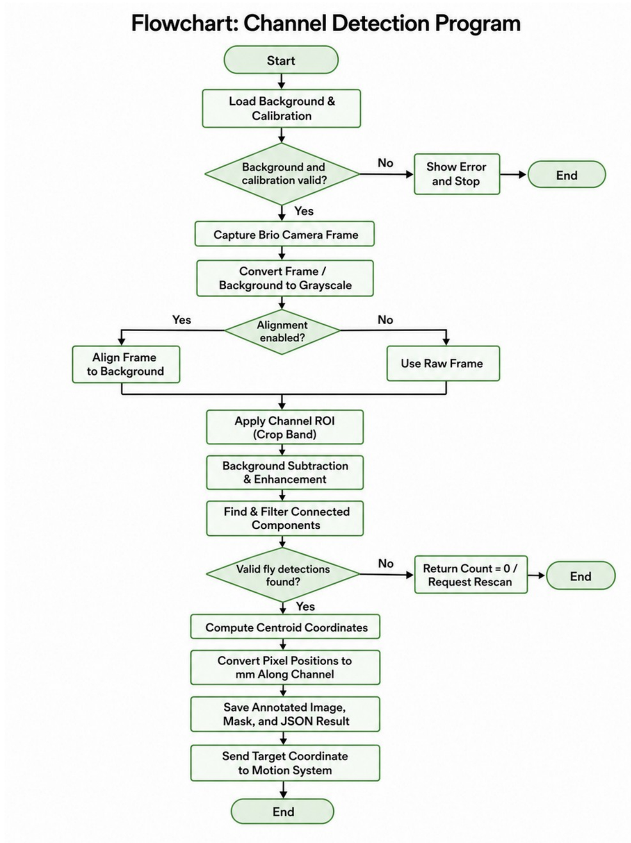

Figure 6: Channel Detection Flow

The primary purpose of the Raspberry Pi High Quality Camera was to take an image of a fruit fly in high detail so that the image can be sent to the Pi for classification. The classification was split into a hybrid model. The first step to creating this hybrid model was to have a baseline YOLO classification model, and for that, several stages of the machine learning pipeline within the hybrid approach had to be refined, most importantly pre-processing and training. Before fine-tuning, the dataset was organized and checked so the model would not accidentally learn shortcuts from repeated images, backgrounds, or camera sessions. Once a sufficiently large dataset was created, there were significant additional checks for duplicate images, class imbalance, incorrectly sexed images, etc. This was critical so that the model learned physical differences between male and female and not just backgrounds/lighting conditions. Several augmentation layers were implemented as well, including brightness changes, contrast changes, blur, noise, desaturation, JPEG compression, and crop shifting so the model would be less sensitive to small changes in lighting, focus, or fly position.

One thing that was used during the training stages was transfer learning. Instead of becoming reliant on one model, we trained multiple models, and used previously trained models as ‘checkpoints.’ Now, instead of retraining new models every iteration, the pipeline was altered to start with an existing model that we had created, and only alter things such as learning rate, batch size, optimizer, scheduler, loss functions, focal gamma, epochs, patience, weight decay, dropout, label smoothing, class balancing, and confidence thresholds across different runs. After several hundred iterations of this, the YOLO-only model was finetuned to a point where it was achieving 100% accuracy on its own validation stage, a sign that the model was fully saturated against the size of the data set. When the model was used in the physical setup, some sensitivity still existed to factors such as lighting, angle, etc. Because YOLO is a data-intensive model and typically works best with >15,000 images, it was determined that the classification accuracy could increase significantly with a coexisting segmentation model that was more resilient to these changes rather than capturing more images and having to iterate over and over again with YOLO.

The hybrid model relies on segmenting the flies as a more direct approach relying on extracting morphological features from fly body regions, complementing a lightweight finetuned YOLO model, as mentioned above. This consists of a series of masking stages that attempts to isolate the core fly body segments as seen in Figure (_a). These segments are then separately masked to extract individual features from each. Overall 28 features are extracted from an optimized segmentation pipeline to work with as many images as possible, with YOLO confidence and classification included as two more features.

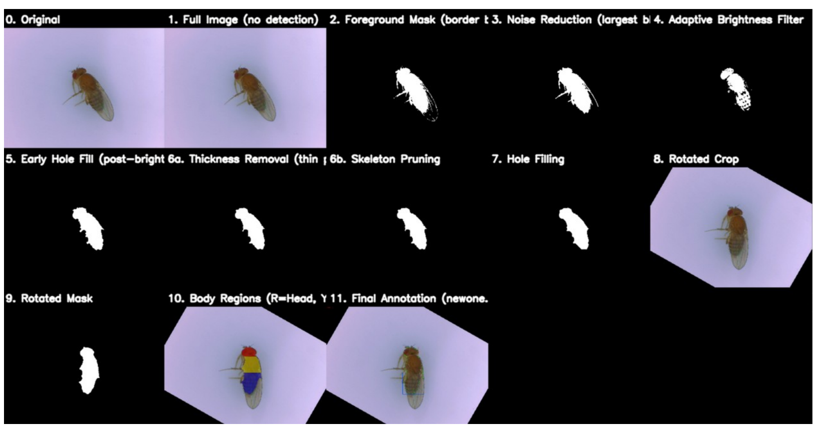

Figure 7: Fly Segmentation Pipeline

Finally, the EMEET SmartCam C960 was used for the climbing assay portion of the computer vision. The purpose of the climbing assay was to determine the amount of flies that cross a particular threshold (placed sufficiently high on a vial) and to extract different quantitative measurements of the fly movement such as velocity. The vials stay in a fixed holder, and the camera is kept in the same position so that the video is consistent from run to run, which is done by design to minimize artifacts (differences in background image and live image that don't result from flies). We used a smooth and consistent back light and a camera box around the assay section because small changes in lighting made the flies harder to separate from the background, which was done as an additional step to prevent artifacts. From the hardware side, a command was sent to record a video using the camera for a specific period of time (in this case this was mostly ten seconds).

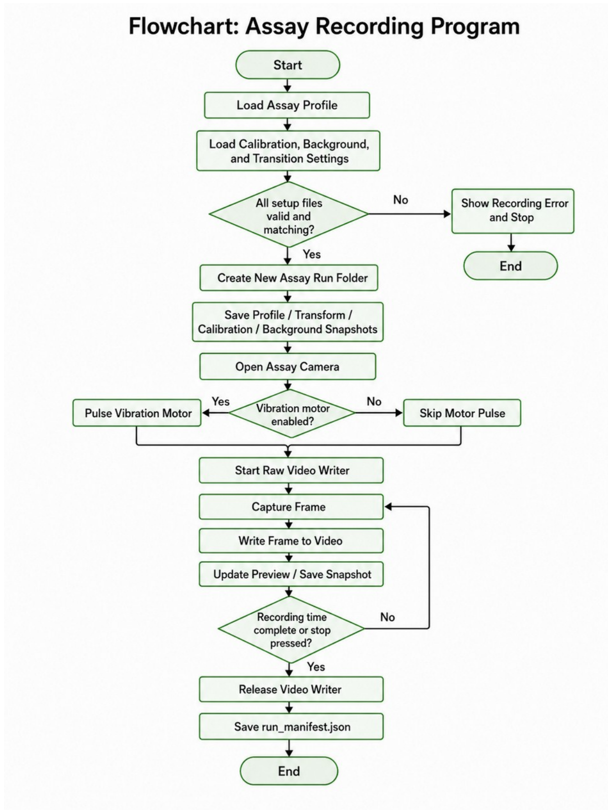

Figure 8: Assay Recording Flow

From a software perspective, this video was taken in as input and was processed frame by frame. The processing steps were mostly the same compared to the channel: a background image had to be taken and a calibration step occurred to box in regions of interest/threshold line. The code then compares each frame to the saved background from before, isolates the darker fly regions, filters out small noise or bad detections (via adjusting parameters like maximum fly size and threshold), and tracks the detected flies inside each vial region. From there, the program can estimate movement over time, including position, distance traveled, speed, and whether flies crossed a certain height threshold. It then optionally outputs this information in pdf format to a box folder.

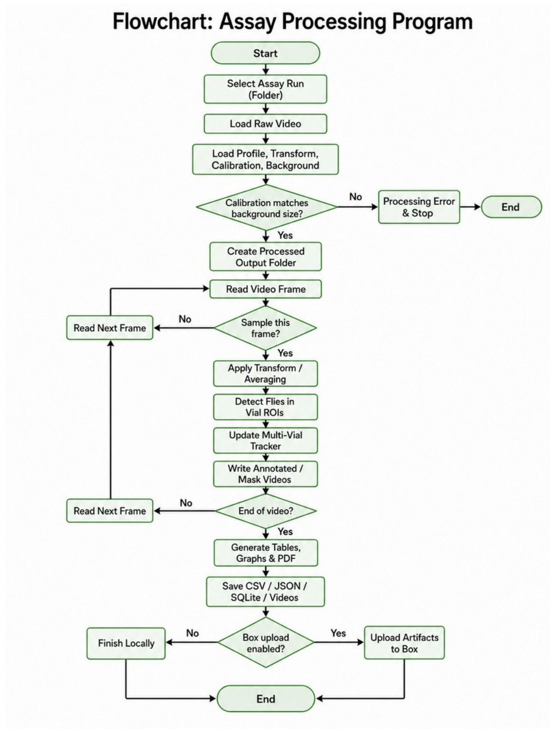

Figure 9: Assay Processing Flow

Performance Characterization and Discussion

Computer Vision Results

The quantitative performance of the vision subsystem for the classification stage can be split into three parts; YOLO Only, Segmentation Only, and Hybrid. To briefly discuss data collection, a team member used the real setup to take >1000 images of pre-labelled fruit flies for the data set. These images were uploaded to a shared box folder, sorted, and verified. The following are the results for the yolo only approach. Note that these evaluations are done using a separate dataset (test dataset), that the model is not trained on. However, the test dataset still might not be indicative of true performance in a real world environment due to factors such as data set bias. The final YOLO model chosen had the following confusion matrix for the test set.

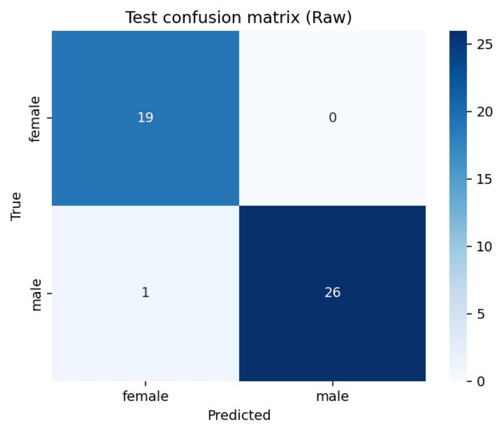

Figure 10: Test Set Confusion Matrix

These results can be interpreted as the following. All females in the test set were correctly identified to be female. One male was incorrectly identified as female, as seen below.

Figure 11 : Example Incorrect Prediction

The model had a final accuracy of ~98% within the new dataset. The image below shows how the accuracy changes as the confidence required to make a prediction increases.

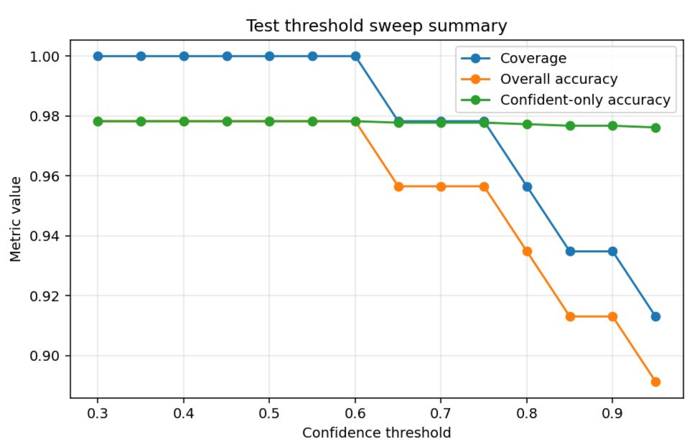

Figure 12 : Threshold Sweep

Although these tests occurred within a small test data set, it was important to test the accuracy of just this model within a real setup. The model alone initially struggled to reach accuracies seen during testing, which was estimated to be caused by potential data set bias, which was the primary motivation for the next form of classification: segmentation.

The body of 30 features for the hybrid model is used to train a RandomForest which is a decision tree style model resistant to overfitting. The final result was highly successful with an 80/20 train test split on 1000 mixed images of new and old. The validation stage here contained the largest image set so far and with the highest variation between images, and as such is the most accurate accuracy score reported up to this point. As seen in Figure (_b), the hybrid model achieved just over 94% accuracy compared to the two other individual models sitting at mid to low 80% accuracy. We may also note that the highest ranking features of the 30 in the combined model were a mix of the YOLO features, and morphological features that are commonly used to visually classify these flies, indicating a success and resilience to overfitting and balancing occasionally unpredictable decision making from a smaller YOLO model.

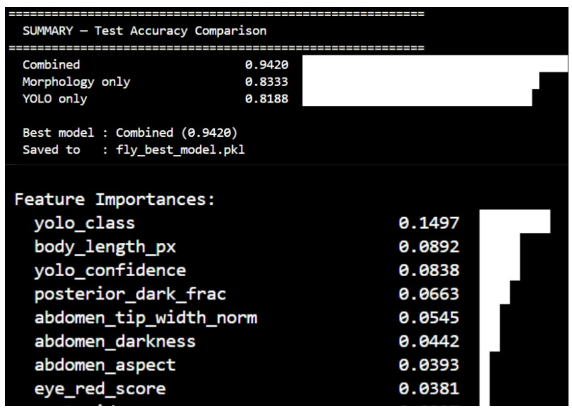

Figure 13: Composite of Test Accuracy Comparison and Force Plot

Integration Code

The overall architecture of the program includes a variety of scripts that are interacting with each other instead of a single script from start to finish. This design was chosen to ease the process of debugging as well to allow for multiple people to work on code at the same time. The code is designed to run on the raspberry pi itself, then an API layer relays the information to a GUI for user interaction or control. There is a controller on both the user side and the computer side to handle the communication between the FastAPI server. The code was designed to work for manual commands such as movements, vacuum,or vibration. However, it was also designed to handle a full automation workflow which included the combination of motion, computer vision, and the assay process.

The system was also designed with a runtime state system which keeps track of current states like position, actuator status, errors, and code execution. The GUI will poll this state and relay these real time updates to the user. This runtime state system also includes detection results, and classification outputs, and system health information. Automation scripts and boot scripts have been designed around the main operating system to relaunch the API services, and start the machine services on reboot, as well as handle crash detection. This runtime state system was integral to the core functionality of the machine but also aided greatly in the debugging and monitoring processes.

Mechanical Design

The Automated Drosophila melanogaster sorter, loader, and climbing assay required specific mechanical specifications in gantry and nozzle design to reproduce high accuracy results. Over the course of design iterations, different criteria were chosen to improve and prioritize. The repeatability of gantry movement and nozzle location are deemed to be the most important, because of the goal for reproducible, high accuracy pickups at levels comparable to computer vision performance. Maintaining optimal conditions requires rigid operation of the gantry plate and proper belt tension. Additionally, nozzle design effectiveness was measured using failure modes and effects analysis (FMEA) on overall pickup completion rate with varying nozzle height, diameter, and internal structure. Results can be seen here:

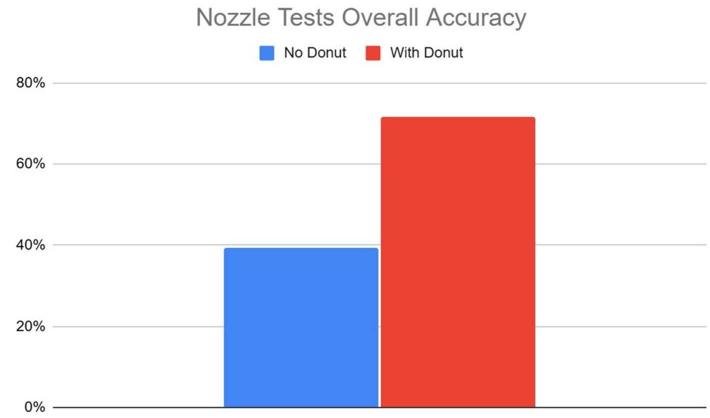

Figure 14: Internal Structure Donut Effects on Accuracy

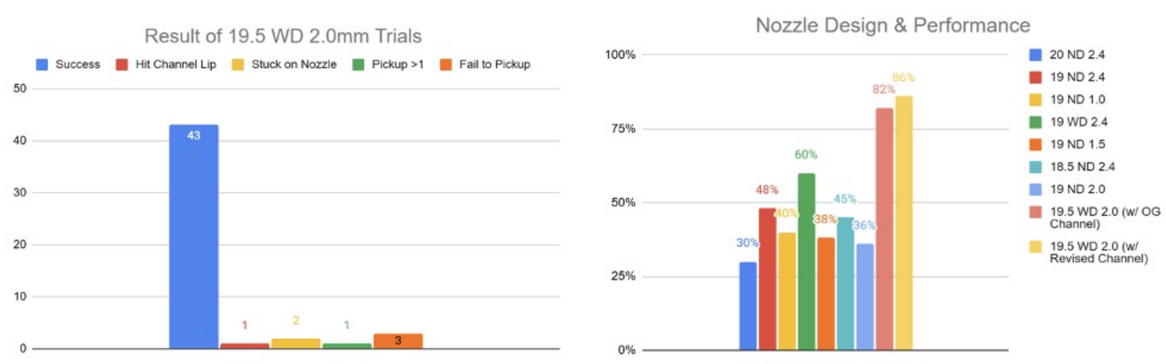

Figure 15: Results of the Nozzle testing

19.5 WD 2.0 (*Recently	Even Distribution	[redacted]	0.9	[redacted]	0.9	[redacted]	0-50mm (first half)	5tubes*10

Live Flies, Mini Ridge Channel)	Comp (X/10)	0.8		0.9		0.8	50-108mm (second half)	5tubes*10
	Failure Modes	Hit Lip (1), Stuck/No Drop (1); Out of Range (1); Pickup 2 (1); Out of Range (1); Stuck/Nodrop (1), Out of Range (1)						
Completion/ Error	Reduces Lip issues and pickups of 2, but more out of range issues, Can be used in hills instead						Overall	**86%**

Figure 16: FMEA Format and Spreadsheet

Each form of nozzle iteration and design has a naming convention that follows: Vertical Length (mm), Interior Structure (With Donut/No Donut Abbreviated), Diameter (mm). The nozzles with the donut internal structure produced much higher accuracies throughout the experiments. The optimal vertical distance from the gantry plate to the bottom of the nozzle is most likely around 19.5 mm. Though, our current setup and testings reduced the number of failure modes experienced: pickup more than one fly, fly hit channel lip and dropped, fly out of range, fly stuck on nozzle, etc. Future iterations and changes to mechanical design may yield a different combination of design attributes that are most successful.

3-D Prints

The table design worked well for our needs throughout the project. The modular nature of the table allowed us to test parts and make changes to the design while only printing the table twice. This was important because the table requires a lot of PLA. The table also performed well overall because it integrated every part of the project together in an efficient way. The overall project has a relatively small footprint because the table holds all of the pieces within it, including the PCB and the Pi. The table also requires minimal user intervention once it is assembled completely.

GUI (Graphical User Interface)

The Graphical User Interface (GUI) was developed in order to develop an integrated system where the entire system will be controlled and monitored via GUI. GUI will provide a way to control the machine via GUI and not by interacting directly with Raspberry Pi or source code. GUI has several panels for controlling and monitoring the system. System status panel will inform about current status of the system such as operating mode, connectivity status and system outputs messages. It helps user to understand what stage it is at the moment and if there are any errors.

Main Dashboard has three major sections including motion control, vision workspace and device control sections. Through the motion control section the user can control homing, moving gantry system to pre-defined positions such as channels and tubes, and position adjustment if necessary. Vision workspace shows outputs from detection, classification and assay processes in form of previews and outputs of the system. Device control section provide the user with an opportunity to control actuators like vacuum pump and vibration motor.

Remote control capabilities are achieved through communication with the backend Raspberry Pi using the API. This allows sending commands to the backend for controlling the system while obtaining system data and image data in real-time fashion. In this way, the GUI is not dependent on the hardware operations while being capable of system management from another computer.

For the climbing assay, rather than using a small window pane, a specialized workspace is opened where all processes from calibration up to exporting can take place, including preview windows for verifying the results and checking possible mistakes made by the user.

GUI is intended to be non-blocking during lengthy operations through asynchronous communication between the software components as well as performing some operations in the background. The GUI acts as an integration platform for connecting hardware operations with computer vision capabilities and, finally, the end user.

Cost Accounting

During the project there were 3 BOMs submitted, 1 PCB order, 1 PCB Components order, and a few items bought at the store during the semester. Appendix A has the full list of items purchased and used. The first BOM submitted encompassed as many items that were thought to be needed based upon initial designs. These items include the RPI-HQ-Camera, Microscope lens, rails, etc. The second BOM included items that were needed to finish the initial design and continue testing parts like the Vacuum pump. The third BOM consisted of items to finish the design and any additional parts that could be needed. The third BOM did consist of the most items not used. This was caused as it was the last submission possible but only at the halfway mark of the project. The PCB order was expedited to arrive on time and thus consisted of additional shipping charges, and the PCB Components saw 3x items ordered from miscommunication on the number of items originally submitted. The few items bought at the store included: dry ice, foam padding, nuts and screws, and the rubber feet. These were bought as last-minute additions or as needed (i.e. dry ice). In total the cost for the items is $883.38. The total reimbursed cost, the cost of items purchased minus the cost of items the ECE department provided is $551.48. The Artifact cost, the cost to reproduce the total design, is $715.53. The Artifact cost does include some replicant parts as certain items can only be found in packs of 2 or more. The original budget goal was

$600 and with a total cost of $883.38 the budget was blown by $283.38. However, the cost if replicated is $715.53 with reduced cost for multiples down the road.

Project Postmortem

Technical Postmortem

From a computer vision perspective, one thing that was particularly strong was the classification model. Combining segmentation and the YOLO model into a hybrid approach allowed for a significantly higher accuracy than either approach individually. Another strong feature was the ability to rapidly integrate different layers of error detection into the sexing stage of the pipeline. For example, it was incredibly easy by design to tweak the minimum fly area needed to count as one fly. This allowed for there to be accurate rejections when >1 fly was present in the sexing stage. Some aspects that were a little weaker were the assay fly tracking. Most problems arose due to fundamental limitations in both the camera and processing hardware. Some issues were resolved by changing the assay approach to be asynchronous; the recording happened first, and processing was designed as a separate stage entirely that occurred frame-by-frame to prevent the Pi from slowing down. This caused significant improvements in the maximum frame rate allowed for accurate fly tracking, yet challenges such as occasional artifacts and camera hardware issues remained. The channel detection as well was at first very slow due to hardware limitations but the time to locate the flies decreased significantly when it was discovered that slow downs were occurring during a step in the code that was transmitting live views of the channel in multiple locations. In terms of the speed of all three components – channel, sexing, and assay – the performance can be improved significantly by either splitting the workload across multiple pi’s or integrating hardware accelerators designed for the pi such as a hailo ai accelerator. Beyond this, using higher quality cameras could allow for an even greater accuracy within the channel and assay stages, although the improvements would be most noticeable within the assay tracking.

The integration code showed a clear separation between the GUI, backend, and the hardware. The modular design made debugging easier and allowed for the testing of individual subsystems independently as needed. This in combination with the remote operation showed great strengths in architecture. The remote operation allowed for the GUI to not be tied directly to the Raspberry Pi, giving the user the convenience of using their own device while also taking off the compute load of the Raspberry Pi by removing the need for a visual display output. The code and system successfully displayed integration of motion control, computer vision, classification, and the assay workflow. A large weakness of this design was the complexity of the codebase. There was over 77,000 lines over code written directly by the team, and 353,000 lines including imported libraries and directories. This made it difficult to navigate and to maintain version history. The responsiveness of the system was also variable due to the eduroam network it was designed on. Heavy tasks like vision and the API calls slowed the system, however the system was optimized to deliver the quickest turnaround time on these processes. Future improvements could include a clean-up of repetitive old files, faster vision operations, and potentially a distributed workload across multiple computational devices or accelerators.

Results of the linear gantry and nozzle development prompted reflection on future improvements and past successes. A complete overhaul of the gantry system to add more degrees of freedom could reduce the amount of failures by a large amount. Because our gantry had only one axis , the system had no way to change the vertical pickup height of a fly. With an extra degree of freedom, flies that were deemed to be out of range or unevenly distributed could be accounted for. Originally, we were limited in the length of the channel because of the table length. Therefore, a modification in channel design could prompt a lengthening of the table to allow for more fly capacity. Lastly, our nozzle was effective when paired with the 12V motor chosen, however, a vacuum pump with a higher airflow, may have allowed for a reduction in the diameter of the vacuum nozzle, and reduced the chances of picking up multiple flies. With the current design, any further reduction beyond 2mm impacted the pickup height of a fly significantly. Again, with a 2-D or 3-D gantry, this problem could have been resolved.. In conclusion, a gantry with higher degrees of freedom and higher airflow pump would allow more flexibility in the design of the channel and nozzle, possibly increasing the accuracy of the system.

The 3-D prints' strengths lie in their modularity and their integration. As stated earlier, being able to test and print pieces individually greatly helped throughout the design process and should help with future iterations of the project. The main weakness is probably the eccentric motor setup. Right now the eccentric motor shakes the tubes as intended, but it also shakes the rest of the table. Moving forward this would be a place for improvement. The primary way the system performance is limited by design choices is that the overall length of the table limits the length of the rail and the channel. Late in development, it was realized that extending the length of the channel would likely help address the issue of multiple flies being picked up at once. The project was also limited by time constraints. The prints needed to be done before most of the motor control could be finalized, which required a lot of communication to ensure that each section of the overall design had their needs met in the design before the table was printed. The table and parts also took a significant amount of time to print, which had to be accounted for in our project schedule and put some restrictions on our testing abilities. For example, the table itself takes about two and a half days to print, not including the fact that it prints in two halves so someone needs to be available to switch it out halfway through. The team’s belief is that the design works well but has potential to be advanced upon. Because the table is such an intensive print, we had minimal opportunities to make changes to the table and had to focus on modifying the parts around the restrictions of the table.

Nontechnical Postmortem

A group of seven members encounters many nontechnical challenges throughout a project such as this. Being able to keep track of assignments, progress, conflicts, and communication is a team effort but also requires a project manager. The group early on decided to have a project manager as well as a group manager for each major component. From there each subcomponent was assigned a primary and secondary. Between the project manager, group manager, primaries, and secondaries this fed accountability through each part of the project.

Holding accountability for seven members became a monumental task. Between average schoolwork days and exam weeks each member was able to be more or less active respectively. It became up to the group managers to ensure each person had worked on their task, or in lieu of the group manager the project manager stepped in. Because of this the group decided to start a discord, a group chat through our phones, and employee individual conversations to be respectful of the full group. During this it was also decided that Monday afternoons provided a time to meet as a team for updates and any questions to be passed to our mentor Dr. Poole. Wednesday evenings also became a time for the entire group to work in the lab for testing equipment, beta-ing the GUI, teaching, etc.

Meetings became very important throughout the semester. They helped clear any communication issues, know of any conflicts coming up, or allowed the group to discuss the next assignments, designs, and shortcomings. Important meetings included the initial meeting to discuss initial assignments that became a part of the PID, and the first and second milestone meetings to discuss assignments and reflect on how the milestone went. In between group meetings, there were many individual meetings: to discuss 3-D design, or the different Computer Vision approaches, or to physically wire the system.

Between group meetings, individual meetings, group texts, and the discord, communication became the milestone of how the group interacted. Without the constant communication, conflicts would have persisted. These could have been misunderstandings of the design, assignments to each member, or the expectations of actions. Communication was kept professional with lightheartened joking included. The group was able to expand their rapport throughout. In the end, communication was a nontechnical requirement in order to achieve a project of this scale.

Acquiring New Knowledge

Dylan

I have gained bulk experience in embedded computing and developing code on a shared machine that I previously was not confident in doing. This manifested largely in the computer vision sections I contributed to in not only developing the hardware related to the problem, but collaborating in developing the data set used to train a vision model, and exploring a novel classification approach. I also gained new skills in coordinating in 3-D space as well, with collaborating on many different 3-D design problems in this project. My main strategy to learn for the entire project was volunteering early on to work on project sections I was most unfamiliar with, and to push my existing skills further when I had opportunities to contribute in these ways too.

Ainara

I have learned so much about building a database, wiring and PCB Design, and modeling and 3-D printing this semester. Learning to build a database was uniquely important for the success and accuracy of the YOLO model. Initially, the photos captured were using multiple cameras, angles, heights of flies, etc.; all which changed the accuracy of the model. I learned how variable (and precise) the YOLO model needed to be for success. Camren played a large role in my understanding of wiring and PCB design. When the PCB came in, he explained the components, their logic, and importance with clarity. Lastly, modeling and 3-D printing has also been an incredibly valuable tool. As designed were discussed with the team, it was important to not only provide visuals but also 3-D objects for the team to get a better understanding of where we were going.

Rivers

I have learned SolidWorks skills and general 3-D printing knowledge (including Bamboo studio) this semester. My learning strategy has been to design new parts in collaboration with others and become somewhat involved in the printing process. My previous SolidWorks knowledge has been primarily based on recreating provided parts, not coming up with new designs. I learned a lot about creating new designs, and I also learned a lot about features of SolidWorks I had never used before. Some examples include using global variables across multiple files and using blocks. Even though I did not print the designs, I learned about the printing process and how to use Bamboo Studio. Creating the new designs with the whole team also required me to significantly develop my interpersonal skills.

Camren

Within this term, I received additional experience in implementing Raspberry Pi integration, designing the PCB, controlling the motor, remote control of the system via APIs, and overall system debugging. Regarding the first point, my strategy was to experiment directly with the hardware on Raspberry Pi, debug all possible issues connected to GPIO and camera, and implement gradual integration of each subsystem to the machine. What concerns PCB designing, I learned how to make a design of the circuit board having considered all requirements related to power supply of the motors and LED lights and wiring organization. As far as motor control is concerned, it meant testing the operation of the stepper, vacuum, and vibration motors and changing the algorithm in accordance with the errors made during working with them. Finally, software integration consisted in learning how to combine all subsystems of the system via APIs, including GUI, Raspberry Pi backend, and hardware itself. As regards debugging, my main strategy here was iterative testing: create a layer, test it, find the reason behind malfunction, and fix code and hardware accordingly.

Will

I have gained some basic and rudimentary python programming skills, specifically in accessing libraries compatible with Pi 5 GPIO and building simple programs to actuate the stepper, vacuum, and eccentric motors. Additionally, I gained experience in testing physical components for failures and presenting results to teammates with design criteria changes and recommendations for improvement. I strengthened my logistical skills by providing feedback from testing and a direction for better results. While the wiring and schematics seem inconsequential, they are tedious tasks that provide reliability and troubleshooting capabilities to the system. Software such as draw.io were required to visually represent the physical connections and components. While working in parallel with the team to provide failure modes and feedback, I learned how the YOLO and morphological models were developed, the development of the GUI, and how the changes in the 3-D designs impact the larger system.

Megan

I have learned more about PCB design, 3-D design and printing, and more about how AI models are made. My strategy to learn has been to ask questions from the "matter experts" as well as getting to use each of these in turn over the semester. I am not an expert in each item but I know more about each subject and have a starting base to continue learning. I was also able to expand my knowledge on being a project manager as I unintentionally stepped more into the role over time to keep up with communication of each member.

Avi

This project has allowed me to learn significantly about the applications of computer vision and machine learning in real-world problems, especially in a biology setting. My strategy to learn these topics has primarily been through reading through existing resources online, including browsing papers on machine learning, computer vision applications, and studying leading models for classification. Using this, I was able to expand my knowledge on being able to train, finetune, and create custom machine learning models and computer vision pipelines for applications involving sub-millimeter targets such as fruit flies. I also gained experience in rapid prototyping and testing as part of a team, due to the fact that some aspects such as the computer vision for the channel had to work closely with nozzle testing to root cause failure points in accuracy and determine the leading causes of over/undershoots.

Technical Standards

Some technical standards that were considered throughout the course of this project include:

1. IPC-2221 - Generic standard on PCB design (trace width, spacing, layout rules)
2. IEC 60364 (Low-voltage electrical installations)
3. ISO 12100 (Machine safety & risk assessment)
4. ISO/IEC 23053 (Framework for AI systems using machine learning)
5. ISO/IEC 22989 (AI concepts and terminology)
6. ISO/IEC 25010 (Software quality model)
7. IEEE 1012 (Verification and Validation)
8. ISO 261 / 262 (Metric Thread Size)
9. ISO 4762 (Hex Socket Cap Screws)
10. ISO 12090 series (Linear Motion using Rails)
11. ISO 286 (Tolerances and Fits)
12. ISO 2768 (General tolerances for machined/printed parts)

Disclaimer: There may be standards not considered during this project that could have been applied or that the team were ignorant of.

Appendix A

Item Description	Quantity	Price/ Unit	Total Cost	Payment Type	Vendor	Used in Final Project? (Y/N)
RPI-HQ-CAMERA	1	$54.39	$54.39	D	Amazon	Y
Microscope Lens	1	$21.89	$21.89	RP	Amazon	Y
Aluminum Extrusion Rail Kit (2040) 500mm	1	$26.99	$26.99	RP	Amazon	Y
Gantry Wheel Plate for 2040 rails	1	$18.99	$18.99	RP	Amazon	Y
20 AWG Connectors	1	$7.64	$7.64	RP	Amazon	Y
Rail Belt	1	$16.99	$16.99	RP	Amazon	Y
Vacuum	1	$8.88	$8.88	RP	Amazon	Y
Vacuum Tube (3mm ID)	1	$9.99	$9.99	RP	Amazon	N
Limit Switch (qty 2)	1	$8.99	$8.99	RP	Amazon	Y
Metal Filters	1	$4.99	$4.99	RP	Amazon	Y
Clear Acrylic Panel (8 piece)	1	$9.98	$9.98	RP	Amazon	Y
Logitech Brio 100	2	$38.99	$77.98	D	ECE Department	Y
Belt Tensioner	1	$7.99	$7.99	RP	Amazon	Y
Stepper Motor	1	$14.00	$14.00	D	ECE Department	Y
White LED Backlight for Assay	1	$16.99	$16.99	RP	Amazon	Y
Belt Clamp	1	$6.99	$6.99	RP	Amazon	Y
GPIO Ribbon Cable for Raspberry Pi Model	1	$2.95	$2.95	RP	Amazon	Y
Glue (for 3-D part assembly) (6 pack)	1	$18.56	$18.56	RP	Amazon	Y
Barrel Bolt (for tube holder)	1	$4.75	$4.75	RP	Amazon	N
Vibration eccentric motor	1	$19.99	$19.99	RP	Amazon	Y
Vacuum Tube (2-4mm ID tube pack)	1	$9.51	$9.51	RP	Amazon	Y
GPIO Ribbon Extender(female to female)	1	$8.99	$8.99	RP	Amazon	N
GPIO Ribbon Extender (male to female)	1	$6.99	$6.99	RP	Amazon	N
Ribbon cable for the bigger camera (15pin to 22pin)	1	$9.89	$9.89	RP	Amazon	N
Ribbon Cable Extenders	1	$6.39	$6.39	RP	Amazon	N
Channel Metal	1	$17.98	$17.98	RP	Amazon	N
Rubber matting (1/8x1x115)	1	$9.99	$9.99	RP	Amazon	N
Ice Pack	3	$9.95	$29.85	RP	Amazon	N
Rubber Feet	2	$3.88	$7.76	RP	Amazon	Y
PCB	1	$73.80	$73.80	RP	JLC PCB	Y
Nuts and Screws	1	$77.00	$77.00	RP	Ace Hardware	Y
Hose barb	2	$4.99	$9.98	RP	Ace Hardware	Y
PLA Black	2	$22.99	$45.98	D	ECE Department	Y
PLA White	4	$22.99	$91.96	D	ECE Department	Y
Dry ice (as needed) (lbs)	20	$2.99	$59.80	PP	Publix	Y
Foam Padding	1	$10.00	$10.00	PP	Michaels	Y
DC Motor driver (each)	2	$15.30	$30.60	D	ECE Department	Y
Stepper driver (5 pack)	1	$16.99	$16.99	D	ECE Department	Y