UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

NOUVEAU SYSTÈME INTÉGRÉ AUTOMATISÉ DE SURVEILLANCE DE PROGRAMMES DE PATRONS MAJEURS ET D'ÉVALUATION DE LA TAILLE DE LA COUVÉE CHEZ *C. ELEGANS*

NOVEL INTEGRATED SYSTEM FOR AUTOMATED C. ELEGANS SURVEILLANCE OF MAJOR MOTOR PROGRAMS AND BROOD SIZE MEASUREMENTS

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN BIOLOGIE CELLULAIRE ET MOLÉCULAIRE

PAR
MATTHIEU VALET

DÉCEMBRE 2023

"If I had asked the public what they wanted,

they would have said a faster horse."

— *Henry Ford*

TABLE OF CONTENTS

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, Dr. Patrick Narbonne, pour sa grande confiance, ses idées et son temps. Nombreuses furent les visites tard le soir pour machiner quelque pièce manquante à mon robot. Il m'a non seulement offert l'espace mais aussi les moyens d'aller au bout de mes idées, aussi curieuses soient elles. Sans lui, rien de ce travail n'aurait été possible.

Je remercie mon père, François Valet, qui m'a encouragé lorsque je doutais et m'a entraîné à pousser encore plus loin les plans que je m'apprêtais à abandonner. Sa vision d'ensemble de mon projet, surtout dans sa dimension mécatronique, m'a permis plusieurs échanges fructueux afin d'atteindre des échéances que je m'étais fixées.

Je remercie Jonathan St-Arnaud de m'avoir expliqué certains fondements des systèmes de positionnement XYZ et Simon Delisle de m'avoir si gentiment accueilli au laboratoire de génie électrique et de m'avoir montré comment souder.

Je remercie les membres du laboratoire du Dr. Narbonne pour leur intérêt et leurs questions autant pendant qu'en dehors de nos réunions d'équipe hebdomadaires.

Je remercie Félix-Antoine Aubé-LeChasseur pour son écoute et ses fréquentes visites qui ont ponctué ma solitude tout au long de ma maîtrise. Je remercie également

Pierre qui, dans son travail de l’ombre (i.e. entretien ménager des couloirs du pavillon) et à des heures tardives, m’a fait me sentir moins seul lorsque je passais la nuit au laboratoire.

Le résumé substantiel sera en français mais, étant donné que ce mémoire fait partie d’une publication en cours d’écriture, la suite sera en langue anglaise.

RÉSUMÉ SUBSTANTIEL

C. elegans, avec son corps transparent, son cycle de vie court et son hermaphrodisme, est un puissant organisme modèle permettant d’étudier le comportement de ponte des œufs. Avec la mise en œuvre efficace de CRISPR/Cas9 en laboratoires, les mesures de taille de portée sont essentielles pour caractériser le nombre croissant de nouveaux mutants de *C. elegans*. La méthode actuelle pour effectuer de telles mesures implique de compter manuellement le nombre d'œufs et de vers sur chaque plaque d'échantillon plusieurs fois au cours d'une semaine, ce qui introduit non seulement des erreurs humaines lors de milliers d'événements de comptage, mais peut également perturber l'animal en l'exposant à la lumière, aux variations de température et aux vibrations. Les trois objectifs de cette maîtrise sont : 1) la conception et la construction d’un robot capable de surveiller un nombre important d’animaux en une expérience ; 2) la conception et l’écriture de l’architecture informatique et du code permettant le fonctionnement du robot ; 3) la validation du robot. Nous présentons ici un nouveau robot automatisé, le Robot Automatisé de Surveillance de *C. elegans* (CeSAR), qui effectue ces mesures à l'aide d'un réseau neuronal entraîné à reconnaître les vers et les œufs, tout en analysant le phénotype des parents. Le robot a été testé par rapport à des expérimentateurs humains pour mesurer la taille de la portée : ces expérimentateurs avaient différents niveaux d'expertise tandis que le réseau neuronal était formé sur plus de 5 000 images. Ce robot contrôlable à distance peut s'intégrer dans un incubateur et effectuer des analyses en parallèle de 48 expériences distinctes, tout en contrôlant l'exposition à la lumière, la température et en surveillant les conditions environnementales. La seule action requise de la part de l'expérimentateur est de transférer les vers parentaux dans leur puits respectif avant de placer les plaques dans la machine, ce qui peut économiser jusqu'à une semaine de travail intensif. Cette méthodologie nouvelle et plus avancée sera non seulement plus efficace et précise que celle relevant des humains, mais pourra également fournir des informations clés sur plusieurs mutations affectant la prolifération des cellules souches germinales, la maturation des ovocytes et établir d'éventuelles corrélations entre les traits phénotypiques et les anomalies du comportement de ponte des œufs.

Mots-clés : *C. elegans*, taille de la couvée, automatisation, intelligence artificielle, apprentissage automatique, robot, détection, vers, ponte, traitement d’images, mécatronique.

LIST OF FIGURES

LIST OF ABREVIATIONS AND ACRONYMS

ABS: acrylonitrile-butadiene-styrene

ANOVA: analysis of variance

AP: Average Precision

ASCII: American Standard Code for Information Interchange

AWG: American Wire Gauge

BoM: Bill of Materials

CAD: Computer-Aided Design

CFA: Color Filter Array

CGC: *Caenorhabditis* Genetics Center

CMOS: Complementary Metal-Oxide Semiconductor

CNC: Computerized Numerical Control

CNN: Convolutional Neural Network

CRISPR: Clustered Regularly Interspaced Short Palindromic Repeats

CeSAR: *C. elegans* Surveillance Automated Robot

DC: Direct Current

DNA: DeoxyriboNucleic Acid

FP: False Positive

FPS: Frames Per Second

GPIO: General Purpose Input/Output

GUI: Graphical User Interface

HDD: Hard-Disk Drive

HDR: High Dynamic Range

IGF: Insulin-like Growth Factor

IoU: Intersection over Union

LCD: Liquid Crystal Display

LED: Light-Emitting Diode

mAP: mean Average Precision

ML: Machine Learning

MWT: Multi-Worm Tracker

NC: Normally Closed

NGM: Nematode Growth Medium

NO: Normally Open

OpenCV: Open Source Computer Vision Library

PLA: PolyLactic Acid

PSU: Power Supply Unit

PVC: PolyVinyl Chloride

R&D: Research and Development

R-CNN: region-based convolutional neural network

RGB: Red Green Blue

RPI: Raspberry Pi

SNR: Signal-to-Noise Ration

TP: True Positive

TPU: Tensor Processing Unit

TTL: Transcription-Translation feedback Loops

YOLO: You Only Look Once

CHAPTER I: LITERATURE REVIEW

1.1 Introduction

Laboratories around the world, whether studying fruit flies, mice, or nematodes, have greatly benefited from recent advances in technology, from whole genome sequencing analyses to better performing microscopes and computers. In particular, a massive expansion can be witnessed in the automation of time-consuming and difficult biological manipulations or analyses, which is allowing to greatly increase the speed at which research can be conducted, all while getting reliable data with independent and large assessments. Time-consuming experiments that could not even have been envisaged a few years ago now become readily feasible with a little help from informatics and robotics. For instance, how would one monitor and analyze the social behavior of courtship and aggression of a pair of *Drosophila melanogaster* in 8 different arenas, all simultaneously? Dankert et al. in 2009[1] realized this feat in no time using their clever home-made device and were able to build complete ethograms for both aggression and courtship in this model. Similarly, how would one non-invasively and continuously monitor a mouse's weight, sleep patterns, and feeding behavior? Ahloy-Dallaire et al. achieved this in 2018[2] using another purpose-build robot, thereby revealing "pronounced circadian rhythms in both feed intake and body weight." By allowing us to gather data a human being couldn't possibly or reliably on their own, automated systems clearly have the potential to uncover exciting discoveries.

Research using the nematode *C. elegans* is no exception: though the first publication on this organism appeared in 1899 by Maupas[3], the first surveillance system only appeared 86 years later[4], in 1985. In this landmark paper, Dusenbery was able to track 25 animals simultaneously but warned that “the subjects must show up as high contrast images” to be accurately detected. Another 14 years later, in 1999, Dhawan et al. refined this system to track more animals[5]. From then on, more surveillance systems were designed and improved, all with their respective specificity, applicability, and availability. For instance, how serotonin coordinates egg-laying and movement was studied by Hardaker et al. in 2001[6]. However, their system came along with costs upwards of $10,000 USD and could monitor worms only for a few hours. Similarly, locomotion and morphology were studied and theorized by Cronin et al. in 2005[7], but their low-resolution system (320px × 240px images) did not allow to score egg-laying. Some, like Li et al. in 2015, have used microfluidics coupled with automated systems to sort animals[8]. Others, like Buckingham and Sattelle in 2008, have preferred to surveil *C. elegans* in liquid environments, namely to study lateral swimming (i.e. thrashing) behavior[9]. Of the worm trackers listed by Husson et al., between 2005-2018, the only 3 published systems with costs below $3,500 USD did not have X-Y stage control[10]. Yet, the true power of computer science and robotics is to analyze spatially separated animals in parallel, requiring multiple arenas and a robust X-Y stage control. We had yet to see a fully automated system capable of performing brood size measurements for a high sample size using artificial intelligence.

With the recent keen interest in aging, stem cell research, and intergenerational behavioral inheritance through epigenetics, it is evermore clear that past and current automated systems need to be unified into one single multi-purpose device to link these different fields. Yet, there seems to be no clear attempts at doing so, which is why reviewing literature on the subject is critical to better understand the challenges and shared processes amongst these systems.

1.2 Objective

In this literature review, we aim to present and synthesize the current state of automated surveillance systems in mainstream model organisms. The mechanical, computer science, artificial intelligence, and biological axes will therefore be discussed to better understand the possibilities, challenges, and potential outcomes of such devices. Although the focus will be on automated systems designed for *C. elegans* research, a broader view is made to include significant advances developed for other model organisms.

1.3 Methods

With a rapidly evolving technology, the following inclusion criteria were applied in the selection of the reviewed papers:

- at most 10% published between 2005 – 2010,
- at least 66.7% published in 2015 or later,
- showing significant advances in mechanical engineering, artificial intelligence, or image processing for data analysis, and
- indexed on PubMed.

We further chose to focus on, but not restrict ourselves to, the *C. elegans* model and the surveillance of the animal's progeny.

This review covers 43 publications that meet the inclusion criteria. In the following subsections, reports that used a particular model organism (e.g. *Arabidopsis thaliana*, *Mus musculus*) will be presented together.

1.4 Review

All automated surveillance systems rely on mechanical engineering to function properly; they all gather some form of data from the biological sample itself or from the arena the sample is placed in. This will generally take the form of a light source and a camera, a scale, or sensors to measure environmental conditions (e.g. temperature, humidity, pressure). Data is recorded on hardware and sometimes automatically processed. Whether the latter is done through clever algorithmic design or artificial intelligence, the resulting data can then be used to find trends from which educated hypotheses can be formulated.

In the following subsections, we will see how, around the world and across models, laboratories have designed and built these systems. We will start by looking at research papers that monitor parts or remains of living models, then move on to whole organisms with regards to scale and medium: first unicellular models, unspecific systems, then swimming, flying, vegetative, walking, and finally crawling models. We finally synthesize the key strategies that made these automated systems successful.

1.4.1 Skin lesions and medical conditions

Before diving into living organism monitoring systems, we will survey different surveillance systems made for human skin, as those illustrate how inanimate objects can be monitored. Esteva, Kuprel et al. (2017)[11] have used deep neural networks to classify skin cancer with a dermatologist-level of accuracy. They trained a deep convolutional neural network (CNN) with a dataset of 129,450 clinical images, spanning 757 disease classes, for an average of 1,710 training images per disease. Their goal was image classification and not object detection. This assumes that each image is of one skin lesion only and that the entire image could be labeled correspondingly. The exact location of the lesion on the image was of no real importance assuming it was roughly in the center. The CNN, once trained, must then correctly identify new images of skin lesions. This illustrates how systems must display both specificity and sensitivity. If very specific but not sensitive, almost no images will be recognized as the correct class (i.e. most corresponding images will be false negatives, and the rest will be true negatives). Conversely, if very sensitive but not specific, almost all images will be recognized as the evaluated disease (i.e. a plethora of false positives and most of the corresponding true images as true positives). This dual requirement is well-known in the field of biomedical diagnostic tests and in this work, they show how their system improves both specificity and sensitivity by increasing the number of clinical images as input training data (Figure 1).

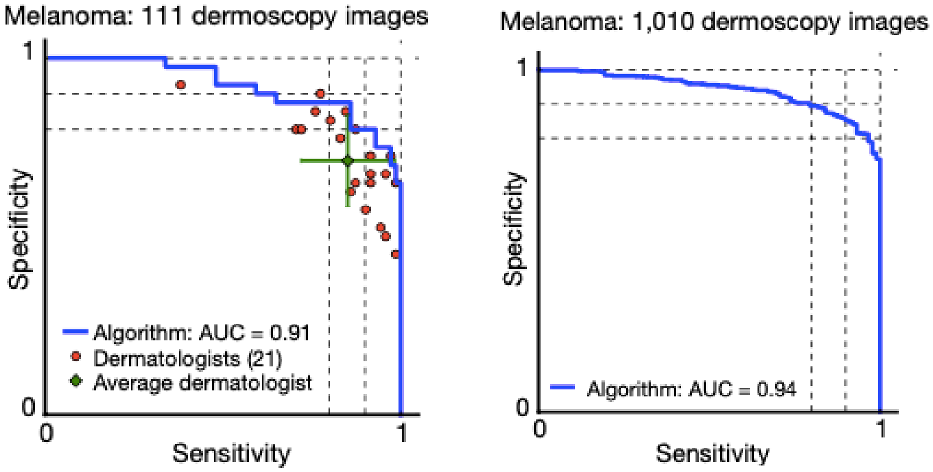


Figure 1: increase of both specificity and sensitivity of the CNN to recognize melanomas using almost ten times more input images as training data (Esteva, Kuprel et al., 2017[11]).

It is also important to note that a lesion is considered immobile on a patient. This explains why the researchers can create a handheld device that can simply be put over the same lesion the next time they evaluate the patient.

Also considering skin conditions, Bolton et al. (2018)[12] designed a low-cost multispectral imaging system that could detect an "increase in blood content of the superficial plexus layer," also known clinically as hyperemia. To capture the images at varying illumination wavelengths, they used a light emitting diode (LED) panel where they placed specific-colored LEDs covering UV (λ = 395nm) up to infrared (IR, λ = 940nm). Although somewhat bulky, their system allows multiple images to be taken of the same sample under 13 different wavelengths in 30s, which provides the researchers with data they couldn't have otherwise gathered. We will now move on to reviewing systems that were developed to monitor eukaryotic and prokaryotic cells and see what changes in design are required to adapt to both the scale and the dynamism of the studied subject.

1.4.2 Eukaryotic and prokaryotic cells

Although significantly larger than prokaryotic cells, eukaryotic cells' 10µm to 100µm remain on the microscopic scale. Tristan-Landin et al.'s (2019) system is a multicolor fluorescence microscope made using 3D-printed parts that can detect single THP-1 monocytes[13]. They use a 3D-printed frame in conjunction with Raspberry Pi 3, with a Raspberry Pi camera module V2 (8 Mpx), and Python 2.7 with OpenCV (image processing library) for their graphical user interface (GUI) (Figure 2).

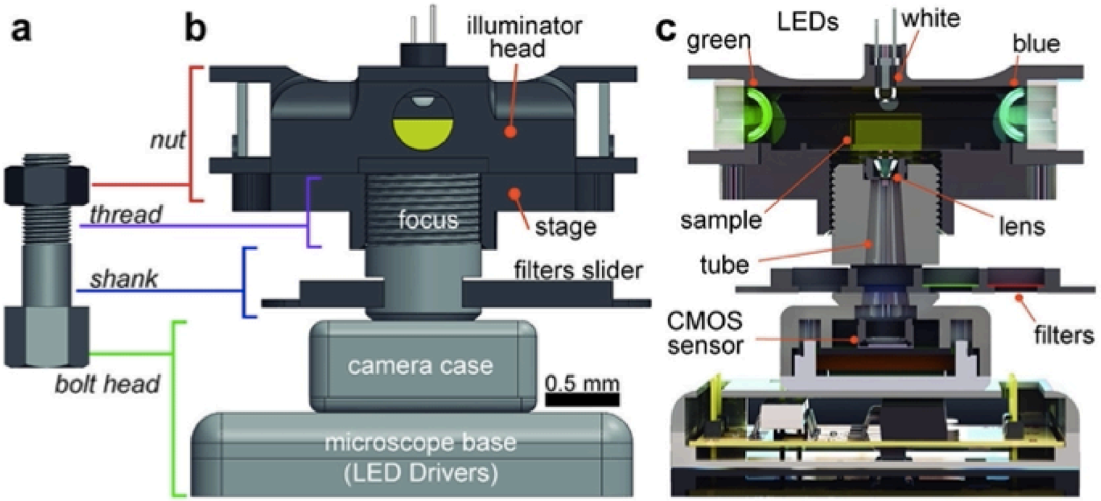


Figure 2: Multicolor fluorescence microscope from Tristan-Landin et al. (2019)[13]. *(a) Bolt and nut mechanism on which the design is based. (b) Front view of the microscope. (c) Frontal cut of the microscope.*

In the µm scale, it is vital to have proper focus design to obtain clear microscopic images (Figure 3).

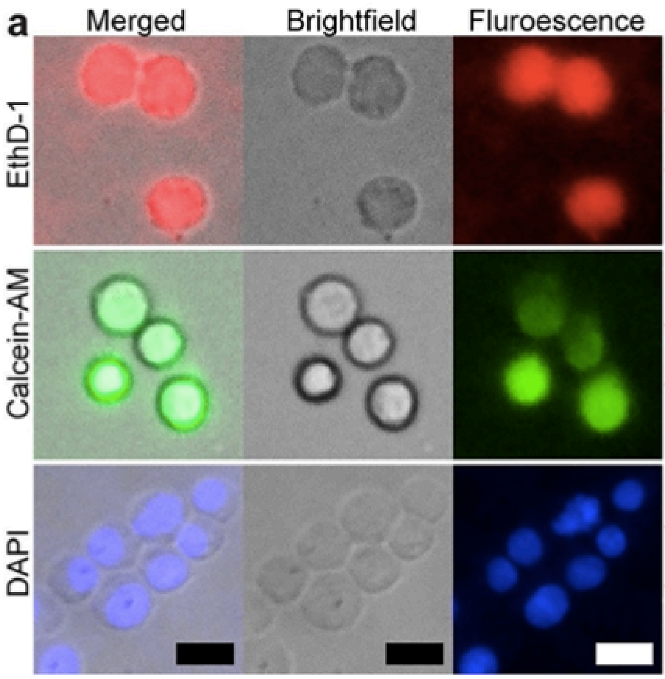


Figure 3: Fluorescence and bright-field (BF) images of THP-1 monocytes taken by the 3D-printed microscope from Tristan-Landin et al. (2019)[13].

To address this, they chose to rotate the head stage containing the LEDs to change the distance between the lens and the complementary metal oxide semi-conductor (CMOS) sensor. However, since both the threading on the nut and bolt were made of polylactic acid (PLA), it could potentially wear out after many utilizations, but is inexpensively replaceable (i.e. $~1 USD). Such design choice however forced them to rotate the sample once the focusing was done. Stewart and Giannini (2016)[14], on the other hand, have decided to 3D-print a frame based on the existing design of research-grade microscopes to observe *Tetrahymena thermophila* cells (Figure 4).

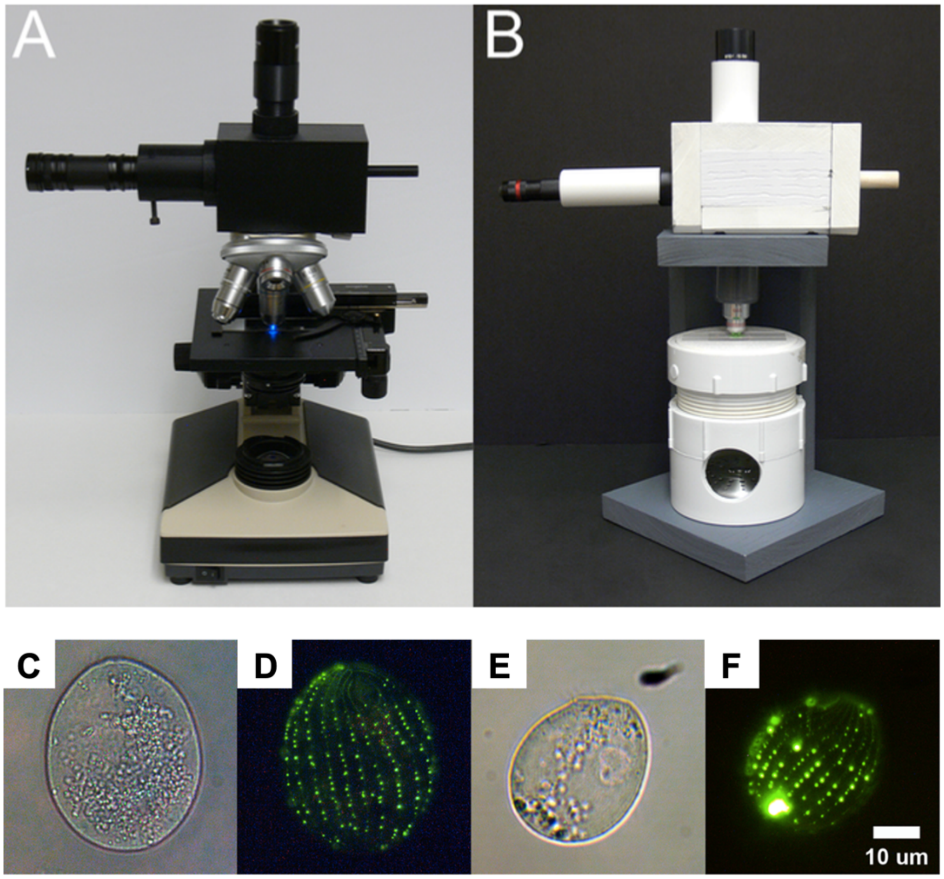


***Figure 4: 3D-printed fluorescence microscope from Stewart and Giannini (2016)[14]** and resulting images of T. thermophila cells. (A) Olympus CH-2 microscope with an adapted 3D-printed version mounted onto it. (B) Complete polyvinyl chloride (PVC) version of the microscope. (C-F) T. thermophila cells from the pICY_AA strain obtained with (A) 3D-printed mount on the left and with (B) PVC version on the right. Their cortex is tagged with a yellow fluorescent protein.*

Still at a microscopic scale, for prokaryotic cells, namely *M. tuberculosis*, Miller et al. (2010)[15] used a similar machine as Stewart and Giannini's with the exception of the lighting for which they used a bright white flashlight with an emission filter mounted onto it. They were able to detect fluorescence from *M. tuberculosis* after dying the slides with auramine orange. However, these systems so far lack mechanical automation, which hinders the creativity and extent of the experimental design.

More recently, Needs et al. (2019)[16] used a compromise where magnification was greatly diminished to observe at a macroscopic scale, in order to build a cellular observation system with a robust automated X-Y positioning system (Figure 5).

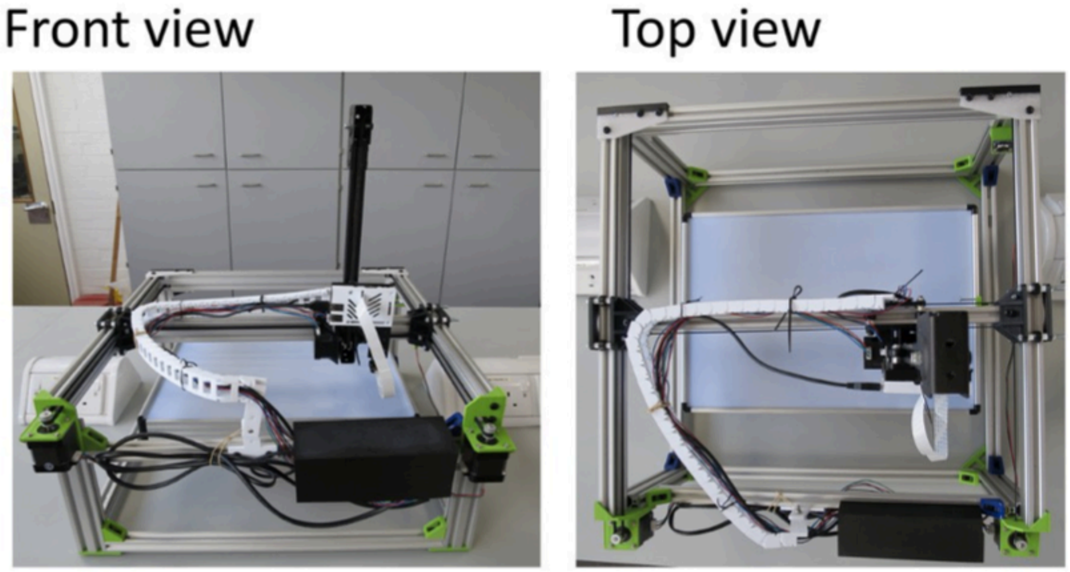


Figure 5: time-lapse imaging for analytical microbiology setup by Needs et al. (2019)[16].

They used 3D printer architecture to build their robot, most notably the OctoPi running an OctoPrint server. The latter is often used in 3D printers to monitor the printing process. They used a Raspberry Pi to send G-code to an Arduino mega board to control the X, Y, and Z stepper motors. Their X-Y stage design is based on CoreXY which is a Cartesian arrangement. It offers great pullback decrease because of continuous tension applied to both sides of the center object and is especially easy to move in straight lines along 90° angles. They used Python scripts to automate the camera settings as well as the path for image acquisitions. This allows them to conduct, in a time-lapse fashion, multiple experiments in parallel, thereby significantly increasing their sample size.

1.4.3 Broad scope systems

Some systems focus on multiple model organisms at once. Although they lose some specificity in design, they can be extremely useful in broad applications. This is the case of Chagas et al. (2017) who built a microscope powered by a Raspberry Pi computer to perform fluorescence microscopy, optogenetics and behavioral assays in zebrafish, *C. elegans*, and *Drosophila melanogaster*[17]. The general design of the microscope is shown in Figure 6.

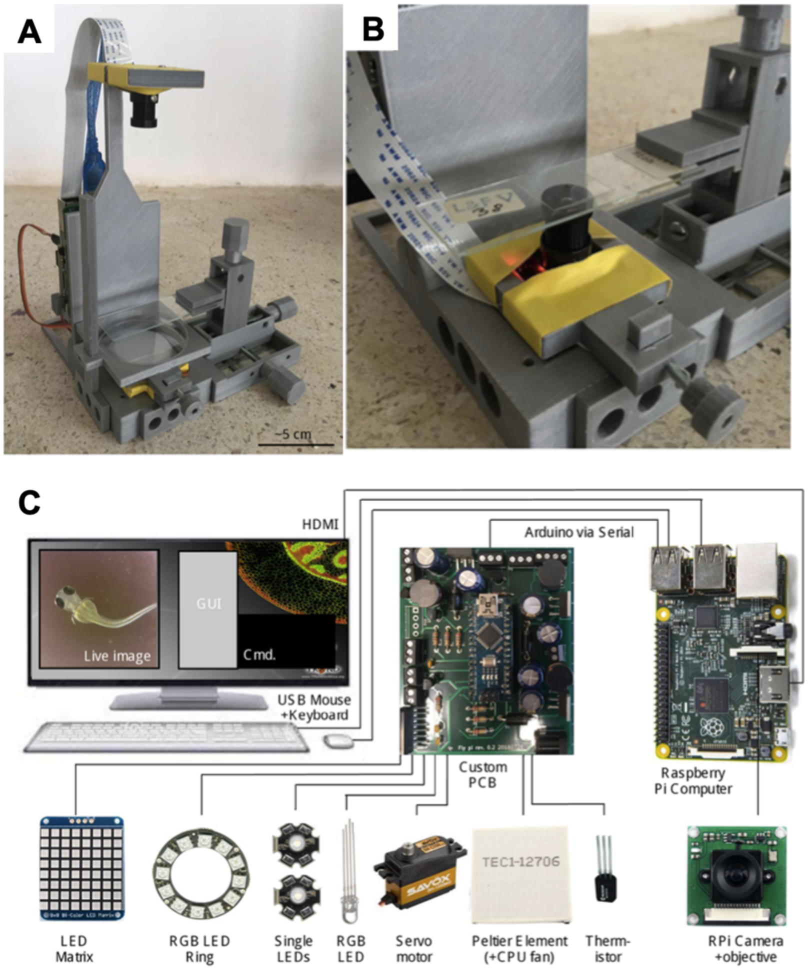


Figure 6: Microscope design from Chagas et al. (2017)*[17]*. *(A) Assembled microscope with a single LED, diffusor, and Petri dish mount. The Raspberry Pi is placed directly behind the microscope.*

(B) Inverted microscope configuration. (C) Electrical wiring between the Raspberry pi, the Arduino, the electrical components, and the screen.

Simply regarding the design, it is noteworthy that the Raspberry Pi 3 computer serves as the master whereas the Arduino serves as a slave to control all electrical components such as the lights and the servo motor to control the focus. This segregation of tasks allows for clean debugging, were a failure or error to arise. The fact that the entire design is 3D-printed and open-source makes it very accessible and inexpensive. The red-green-blue (RGB) LEDs can be used programmatically from the Raspberry Pi, which offers a lot of flexibility regarding sample illumination. On a more technical basis, the algorithmic surveillance is equally simple yet effective: automatic behavioral tracking, as shown in Figure 7.

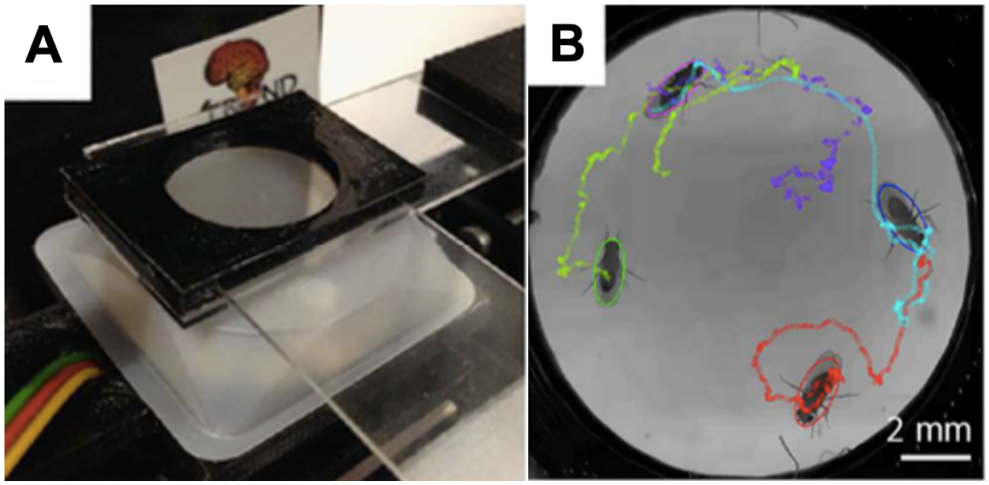


Figure 7: Behavioral tracking from Chagas et al. (2017)[17]. *(A) Chamber made up of two miscroscope slides held by a 3D-printed frame. (B) Adult Drosophila tracked using Ctrax[18].*

The system is also able to perform calcium imaging in larval *Drosophila* muscles. This, along with other microscopic images taken in this project confirm the potential of such system. One might argue, however, that this is limited: the amount of data that can be gathered is arguably diverse due to the many observable model organisms, but it is not substantial. Parallelism of experiments is impossible and there is no X-Y control. Another attempt at a versatile and customizable microscopic imaging system is by Diederich et al. (2020)[19]. Although they share a similar objective, this system is modular, as it is made up of building blocks, and thus allows for more flexibility around experimental design (Figure 8).

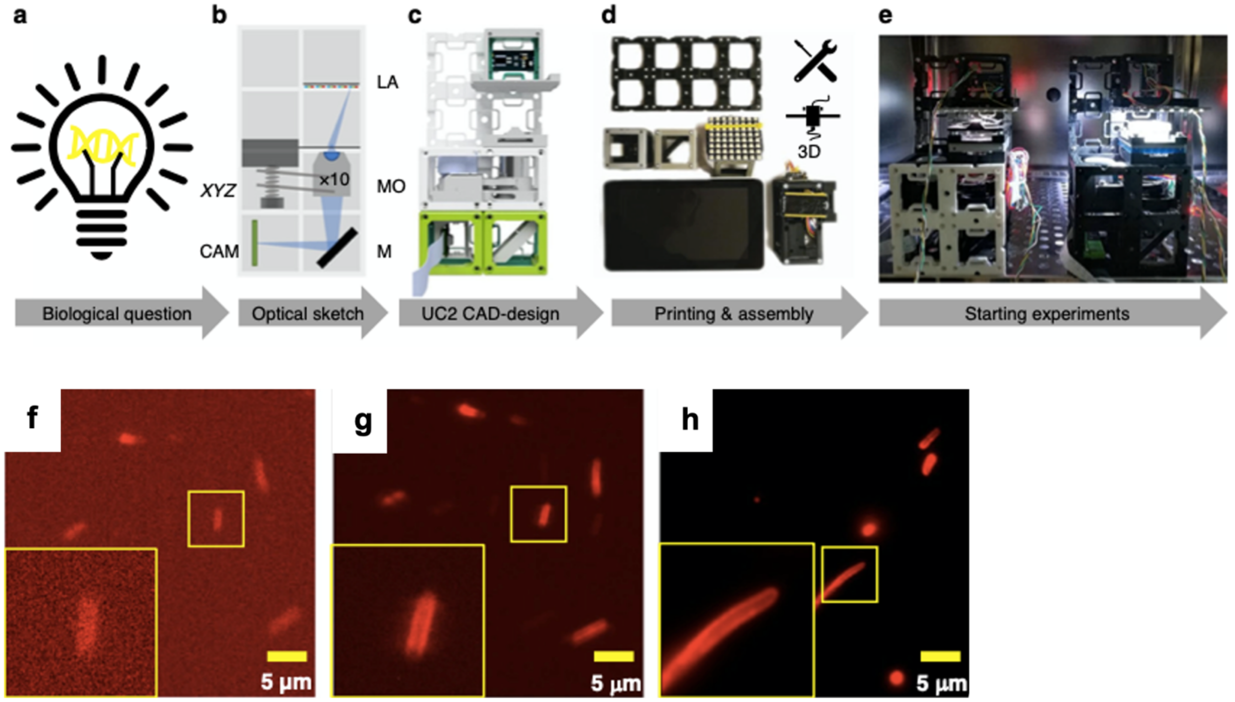


Figure 8: Microscope design and fluorescence result from Diederich et al. (2020)[19]. *(a) The biological question that requires imaging to be answered. (b) The sketch of all components needed to be built. (c) Print and assembly. (d) Placement of the system in its working environment. (e) Start of the experiment. (f,g,h) Images of the same sample mCLING-ATTO 647N labeled fixed E.coli) using respectively a Raspberry Pi RGB camera, cellphone monochrome camera (P20 Pro, Huawei), and a research-grade microscope.*

As the underlying principle of fluorescence microscopy is to establish the greatest signal-to-noise ratio (SNR), one can note the exceptional performance of the cellphone monochrome camera. Monochrome cameras lack a color filter array (CFA) which separates photons before they hit the sensor (Figure 9). For the same number of sensor pixels, a monochrome camera will not have color but will more sensitive to light. This is especially useful when color is not of particular interest, but contrast is.

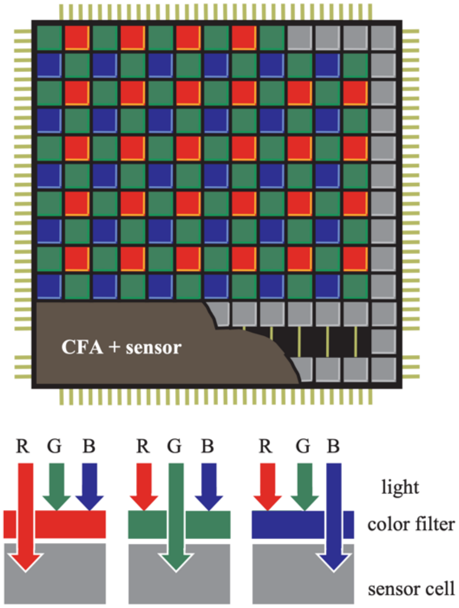


Figure 9: Representation of a camera sensor with a Bayer CFA on top of a pixel array from Lukas et al. (2006)[21].

These models still lack X-Y control, but they show how effective and flexible small monitoring systems can be designed. The monochromatic camera's sharpness relative to light intensity is striking. Similarly, the implementation of tracking seems quite accurate even though the setup is suboptimal.

1.4.4 Aqueous medium (*Xenopus tropicalis* and zebrafish)

Let us now present systems that monitor organisms grown in liquid medium. *Xenopus tropicalis*, also called the western clawed frog, is found in African countries and measures 4 to 6cm in body length. Its embryos and eggs constitute an especially good model because it has a relatively short generation time (< 5 months) and it is evolutionarily closer to humans. As a tadpole, *X. tropicalis* measures around 12mm in length and allows great visibility to its organs, on which Eckelt et al. (2014)[22] centered their research to build an automated machine that measures not only the animal's body movement, but also its cardiac activity. They built the monitoring machine using a camera mounted on a tripod looking directly down at a 24-wells plate, with each well containing a tadpole in 1.2mL of water (Figure 10).

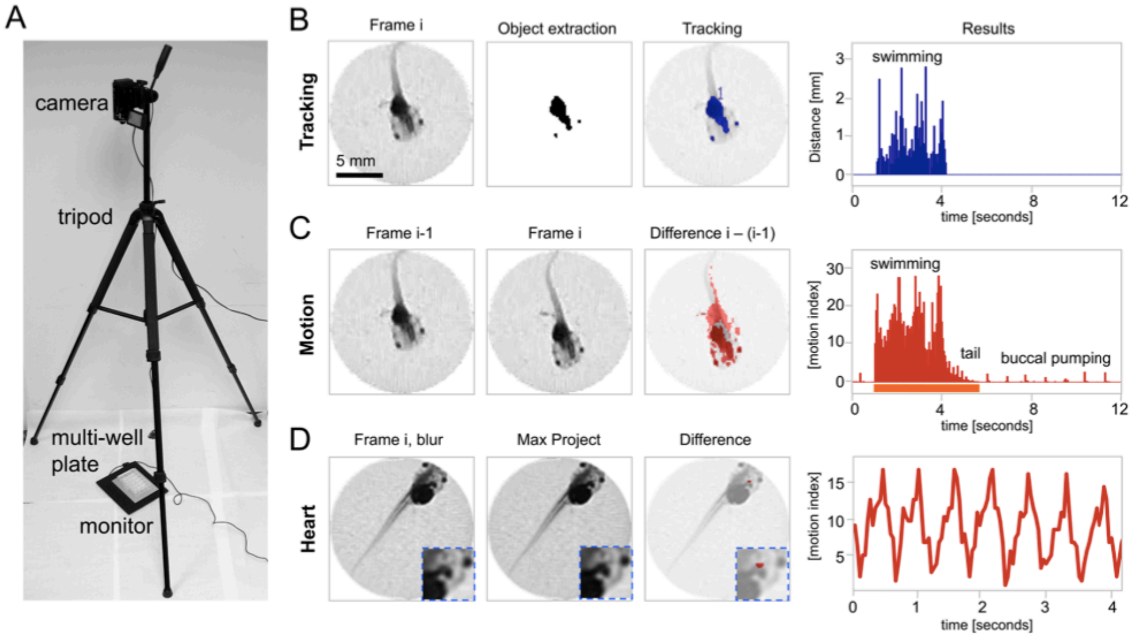


Figure 10: Overview of the video recording and subsequent analysis of X. tropicalis tadpoles designed by Eckelt et al. (2014)[22]. *(A) The recording system using a camera mounted on a tripod pointing directly down on 24-wells plate back-illuminated by a monitor. (B) The animal tracking based on extracting and recording defining features of the tadpole. (C) The establishment of*

motion based on the difference between two successive images. (D) The establishment of cardiac activity based on the difference between two successive images, where the subject is immobile.

The use of relative differences in their image processing allowed, all other things being equal, to detect buccal pumping. This achievement is therefore certainly replicable with model organisms of comparable size and transparency. For image analysis, they used macros in ImageJ on images taken by a Sony Handycam high dynamic range (HDR)-CX210 of only 2 Mpx at 30 frames per second (FPS). This allowed automatic and accurate assessment of the effect of propranolol and atropine on the animal's cardiac activity. We witness here high-quality results from a simple and inexpensive design choice.

Liquid media can also prove challenging with other species like *Danio rerio* (zebrafish). This organism has regenerative abilities and is frequently used for drug development. Indeed, Ishaq et al. (2014)[23] focused their efforts on the fact that zebrafish embryos, when exposed to certain chemicals that inhibit deoxyribonucleic acid (DNA) repair, will develop spinal deformation. At this macroscopic scale they designed a system to automatically determine, after drug administration, which animals within a given population had developed a spinal deformation (Figure 11).

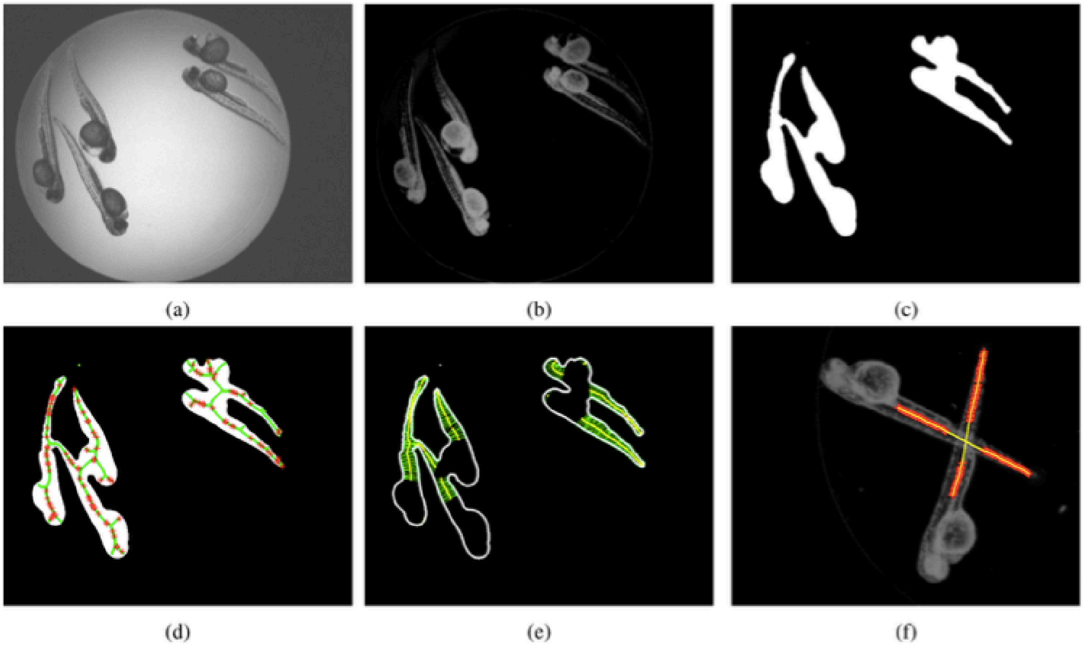


Figure 11: Steps sequence to extract zebrafish curvature by Ishaq et al. (2014)[23]. *(a) Input image. (b) Illumination correction. (c) Binary thresholding. (d) Skeletonization using medial axis. (e) Refined medial axes. (f) Fusion of the medial axes to get the complete tail in the case of animals crossing over one another.*

This is the first example we see of the skeletonization of an organism used to differentiate animals based on a specific feature. Regarding the liquid medium we have presented, surveillance systems with the studied zebrafish having an arguably moderate degree of freedom. Let us now look at an automated surveillance system that allows for a far greater degree of freedom. Barreiros et al. (2021)[24] built a large tank in which multiple animals could freely swim simultaneously (Figure 12).

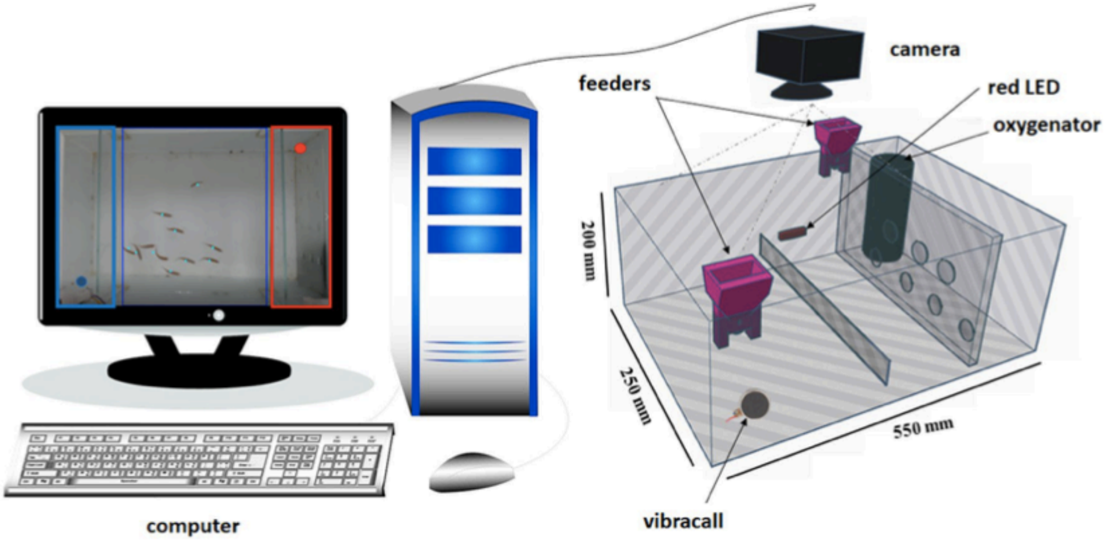


***Figure 12: Surveillance system design from Oliveira Barreiros et al. (2021)**[24] to monitor a group of zebrafish swimming in a large glass tank.*

The resulting paths and heatmaps of the animals tracked were obtained using YOLOv2 convolutional network trained on 365 images of zebrafish in which the head was marked manually (Figure 13).

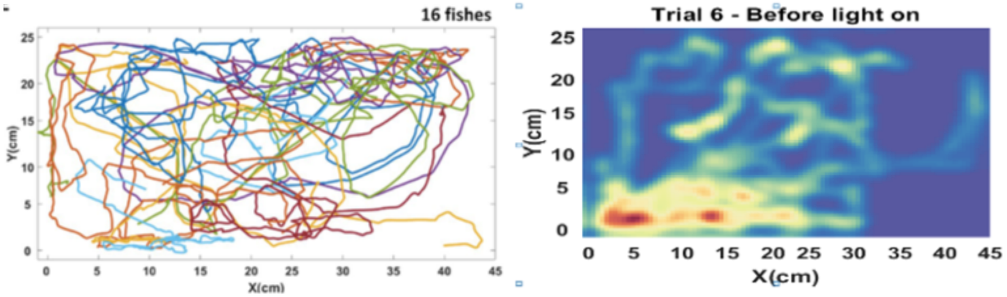


Figure 13: Tracked paths and position heatmap of zebrafish in two distinct experiments from Oliveira Barreiros et al. (2021)**[24]**.

Occlusions by other fish was remedied by using a Kalman filter which estimates states of a dynamic system based on noisy or incomplete measurements. One can notice also how a heatmap quickly gives a clear indication of where the animals were most concentrated. Over a period, such measurements could easily establish potentially interesting trends or correlations. Now that we have seen automated monitoring of animals in liquid medium where fish have room to swim, we will now review Early et al.'s[25] high-resolution of myelinating oligodendrocytes *in vivo* in immobilized zebrafish. To automate such imaging, the animal is placed in a 600µm glass capillary that can move linearly as well as around that same axis (Figure 14).

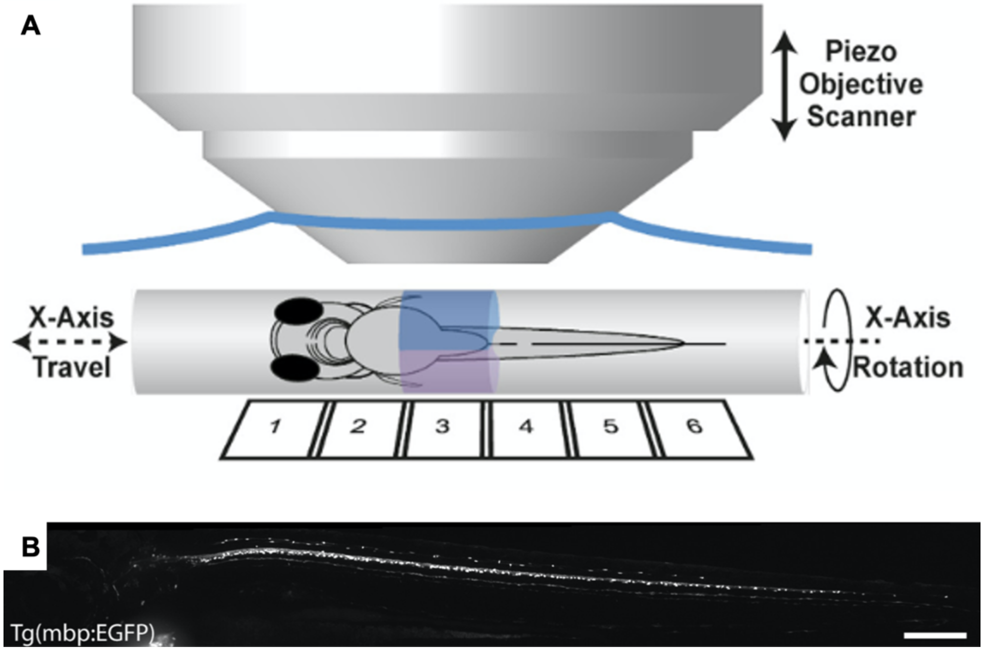


Figure 14: Microscope objective focusing on an adult zebrafish inside a capillary (A) with the resulting fluorescence of 6 stacked images with a scale bar of 200µm (B) from Early et al. (2018)[25].

Instinctively, and in practice, the compromise of clarity for freedom of movement is now evident. It would obviously be very challenging to observe and follow a freely moving zebrafish in water at a microscopic scale. We will now compare how tridimensionality affects surveillance system when the medium is air rather than water.

1.4.5 Air medium − flying sample (*Drosophila melanogaster*)

Drosophila melanogaster, also known as the fruit fly, is a great genetic model and is about 2.5mm in length. While Nouhaud et al. (2018)[26] focused on developing an ImageJ Java plug-in to count their eggs and effectively estimate female fecundity, Schumann and Triphan (2020)[27] focused on the 10 days post-embryonic development of the animal (Figure 15). They did so by developing their monitoring "PEDtracker," which took images of a 24-well plate that later were processed and analyzed using ImageJ and R (a programming language for statistical analysis and graphics).

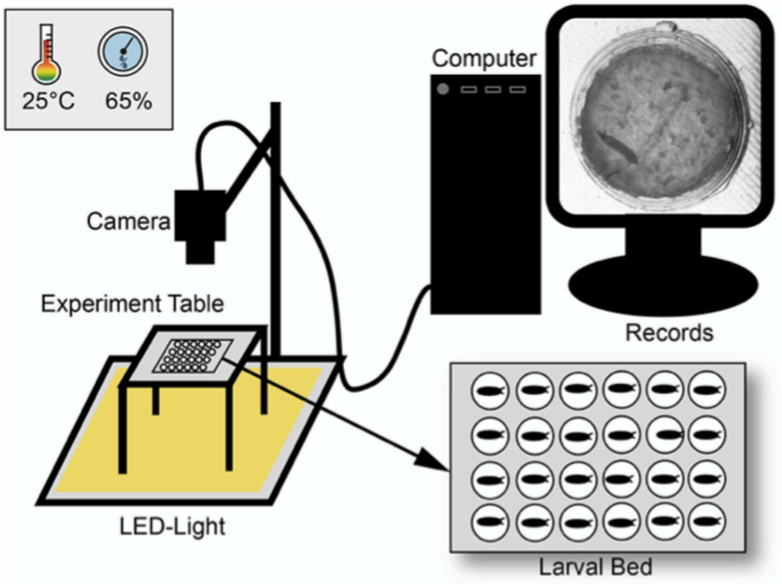


Figure 15: Automated system design by Schumann and Triphan (2020)[27] to monitor D. melanogaster larvae.

Their image processing for tracking uses blur, project median, background subtraction and a final binary threshold. With enough accuracy, they can detect molts and essentially, based on animal's lengths, aggregate them into the three distinct larval stages before they reach pupation.

The next three papers on *D. melanogaster* all deal with adults. Dankert et al. (2009)[1] designed an automated monitoring system that measures aggression and courtship (Figure 16).

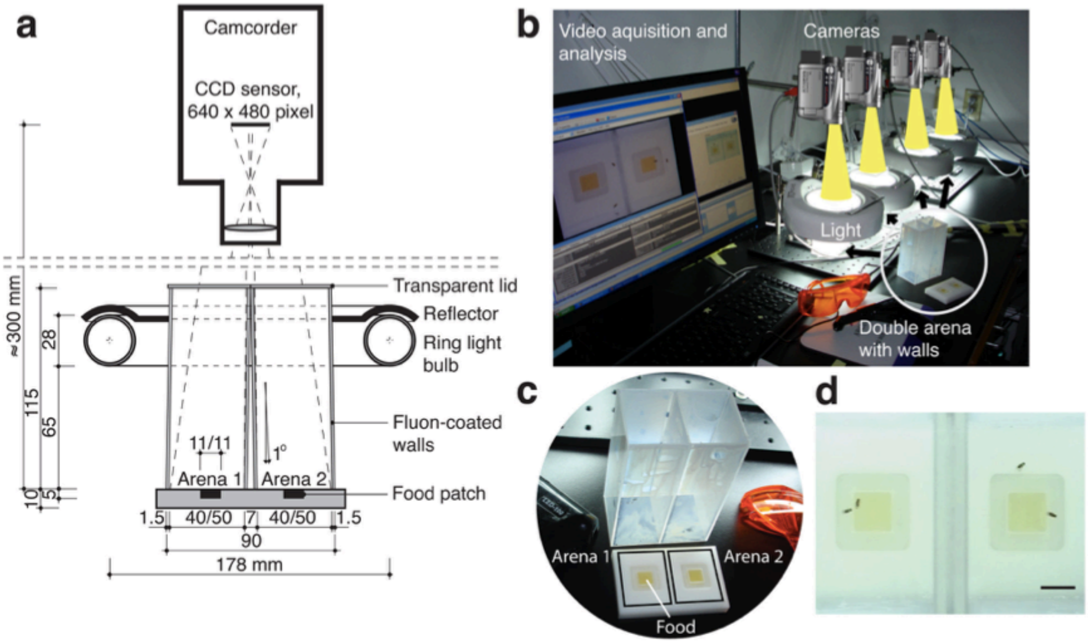


Figure 16: Imaging setup to measure courtship and aggression in D. melanogaster by Dankert et al. (2009)[1]. *(a) Frontal cut of the double arena. (b) Example screening assay where four double arenas are being monitored. (c) Double arena with the walls removed. (d) View from the camera of the double chamber.*

The lighting itself is noteworthy because the arena floor being white, reflects the light coming from above and creates a high contrast. This contrast, as for all imaging, makes post-processing that much easier. With their setup they can phenotypically define lunging, tussling, wing threat, copulation, chasing, and wing extension and circling. By analyzing their footage, they were able to make complete ethograms of courtship and aggression for this organism. With a comparable lighting setup, Scaplen et al. (2019)[28] automated the real-time quantification of locomotor activity in *D. melanogaster* (Figure 17).

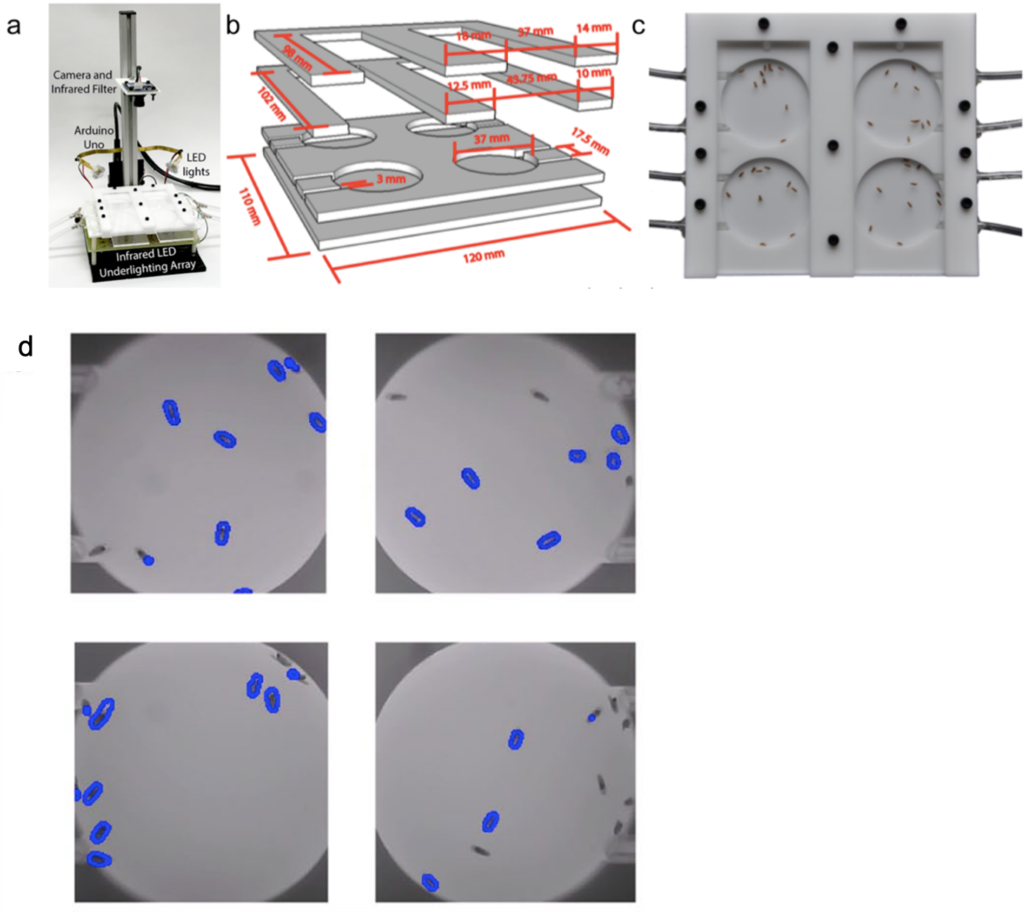


***Figure 17: Design of D. melanogaster locomotion monitoring from Scaplen et al. (2019)[28].** (a) Fully assembled machine consisting of the behavioral chamber, an underlighting LED array, two LEDs (λ = 680nm) used for optogenetics and an Arduino Uno. (b) Computer assisted design (CAD) of the behavioral chamber. (c) The assembled and populated behavioral chamber with tubes ensuring proper airflow. (d) Video frames where blue marked animals have been detected as active.*

This research group chose to make their Python based system called "flyGrAM" freely available on github, following a true open-source philosophy. To evaluate their model, they successfully characterize locomotion induced by a dose-dependent induction of ethanol. This shows how clear results can be obtained regardless of the simplicity of design of the monitoring device. Finally, Itskov et al. (2014)[29] studied the feeding behaviour in *Drosophila*, showing rhythmic extension of the animal's proboscis, and ultimately raised similarities in food intake regulation between insects, rodents and humans. They achieved this using the principle of capacitance in a custom arena (Figure 18). As an animal interacts with food with its proboscis, the capacitance between the two electrodes changes and food intake is inferred.

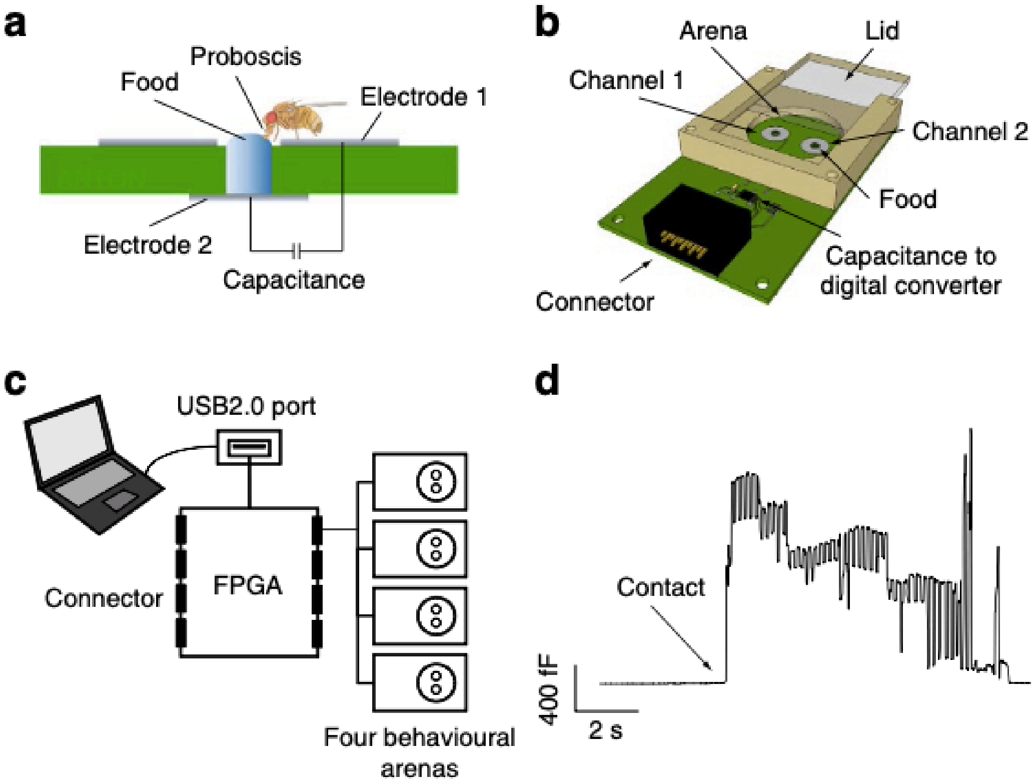


Figure 18: The "flyPAD" design by Itskov et al. (2014)*[29]*. *(a-b) Diagrams showing the flyPAD mechanism and components. (c) Diagram of the connection from the flyPADs to the computer. (d) Digital output converted from the flyPAD's capacitance values.*

1.4.6 Air medium – stationary (*Arabidopsis thaliana*)

We will now present research on automated surveillance of another 3D model that is immobile in the X-Y plane and grows upward. With its rapid life cycle and self-pollination, *Arabidosis thaliana* is a popular genetic model in plant research. The plant itself can usually grow up to 20 to 25 cm tall while its leaves can grow from 1.5 to 5mm long and 2 to 10mm broad. These dimensions, as we will soon see, constitute, as per other models, size specific requirements for the automatic surveillance of the plant. Indeed, due to the relatively contained variability of the growth of its leaves and as well as the orthogonality of its growth, this model seems to have an established and predictable pattern for the way it grows and is monitored. Both Arend et al. (2016)[30] and Tovar and Hoyer et al. (2017)[31] both share similar angles and image processing: a top and side views followed by some level of feature extraction. On the one hand, Tovar and Hoyer et al. (2017)[31] were able to construct a low-cost machine powered by a Raspberry Pi to extract important phenotypical traits from *A. thaliana* such as shape, area, height and color (Figure 19).

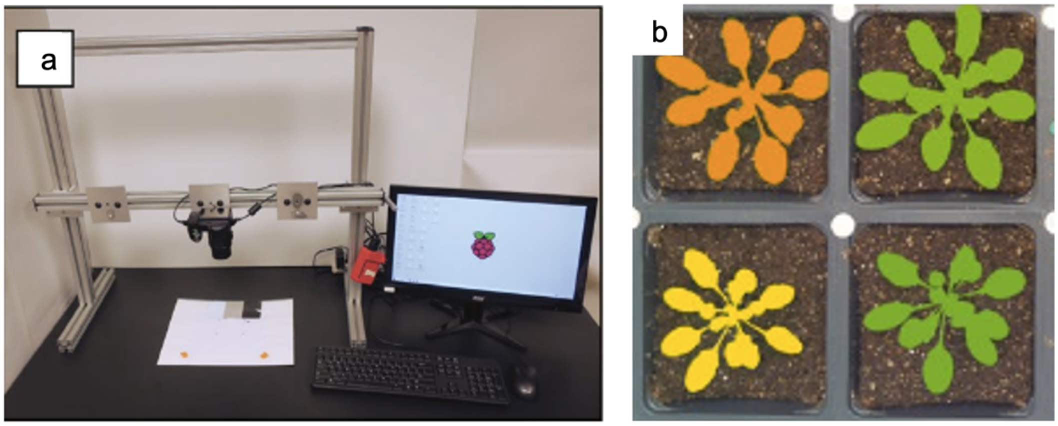


Figure 19: Plant monitoring system and resulting image from Tovar and Hoyer et al. (2017)[31].
(a) Stand with Raspberry Pi and its camera. (b) Segmented image of A. thaliana using Plant-CV (a toolbox derived from the OpenCV Python library specialized in the analysis of plant images).

Likewise, Arend et al. (2016)[30] devised a robot capable of extracting the features of the the plants. The phenotypical traits they gathered were strikingly resemble those of Hoyer et al. (2017)[31]. We can assume that the similarity between the imaging systems from both research papers is because there is only little variability in the growth of the plant.

1.4.7 Solid surface medium – walking (*Mus musculus*)

We will next review the way mobile model organisms are monitored on solid surfaces, starting with large-scale walking models. *Mus musculus*, also known as the house mouse, is undeniably one of the most studied model organisms worldwide. With a body length of 7.5 to 10 cm and a tail length of 5 to 10 cm, they are the longest model organism we have covered so far. This significant size allows for quite specific automated monitoring to take place, as we will discover with the next three research papers.

A non-invasive and automated monitoring of feeding and body weight was developed by Ahloy-Dallaire et al. (2018)[2] who designed a system that recorded body weight every time the mouse went to eat (Figure 20).

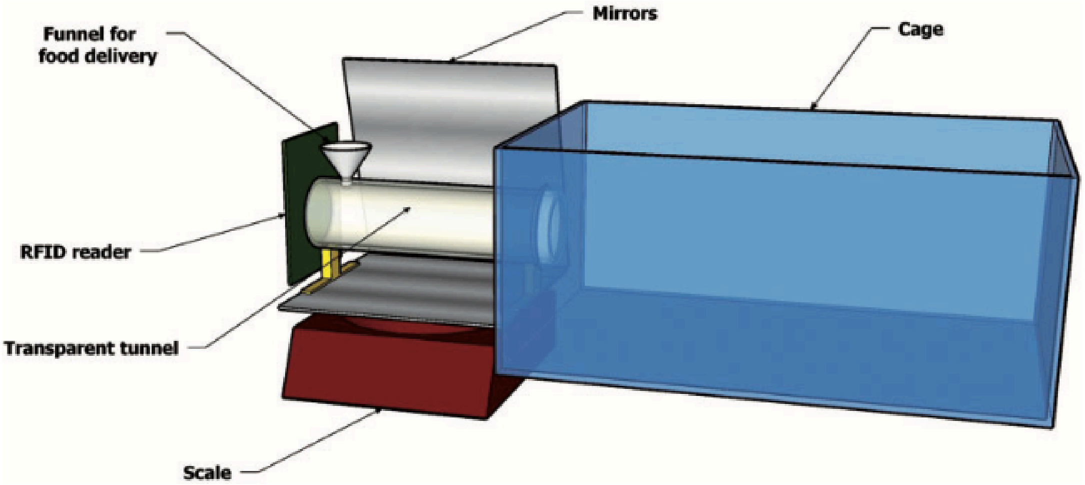


Figure 20: Health monitoring apparatus from Ahloy-Dallaire et al. (2018)[2]. *On the occasion that the animal entered the transparent funnel, a food pellet would be delivered through the funnel and a picture would be taken.*

Such level of precise health monitoring proved better at detecting sick mice after injection of a drug as compared to human-based detection. This is therefore a non-invasive and reliable way to monitor the health of mice and allow for a faster reaction to early signs of illness from animals. Another group of researchers, namely Nanjappa et al. (2015)[32] managed to create a system that is able to estimate the 3D posture of a mouse seen simultaneously from above and below (Figure 21).

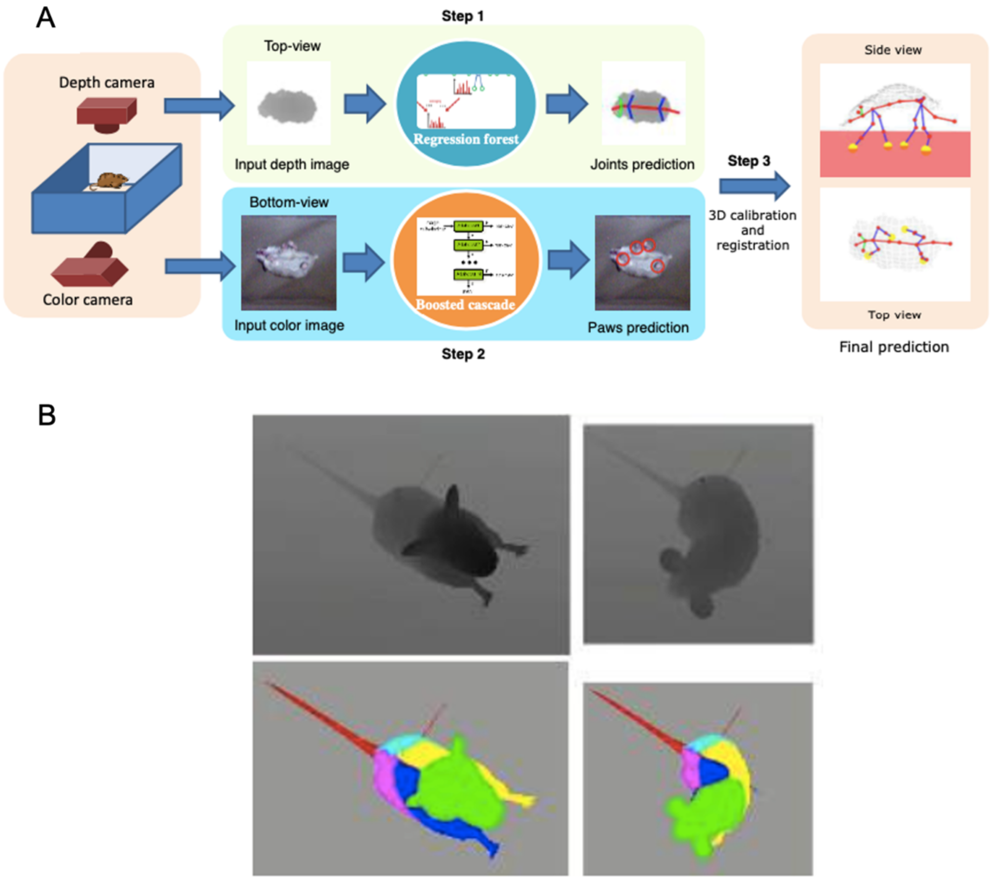


Figure 21: Experimental setup from Nanjappa et al. (2015)[32] *with an open-field cage with two cameras: one above and one below. (A) Entire process from acquisition to final prediction. (B) Mouse postures and rendered depth images.*

Their system allows for a very detailed 3D posture estimation of the animal. Finally, we will go into more detail with the research from Andresen et al. (2020)[33], where they assessed the well-being of the mouse by analyzing its facial expressions automatically via their custom deep learning network architecture (Figure 22).

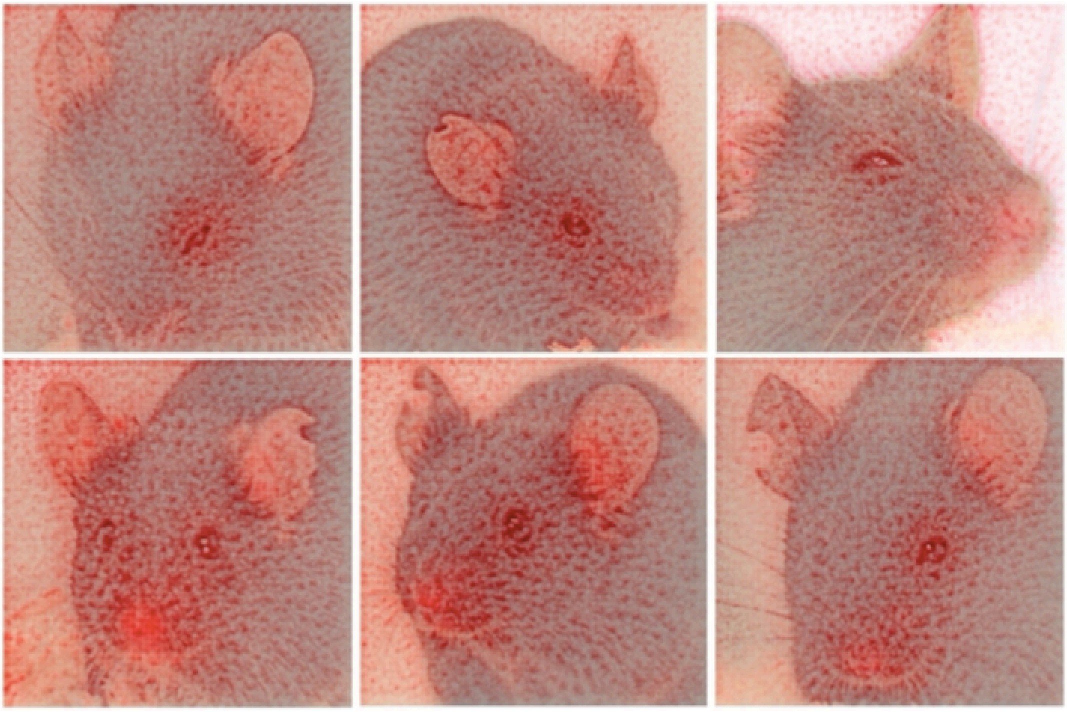


***Figure 22: Imaging of the decision process from Andresen et al. (2020)[33].** The red color density is proportional to the area's contribution to the classification. Left mice underwent castration, while the other two underwent different types of anesthesia (ketamine/xylazine and isoflurane from left to right). Mice in the top row were correctly classified as post-anesthetic effect and mice in the bottom row were correctly classified as no post-anesthetic effect.*

One can appreciate the level of complexity used to make this classification. It appears the nose, ears, and eyes are major components of *M. musculus* facial expression. They were also able to show that the same trained model was performant only on the treatment it was trained with. This further exemplifies the power of trained neural networks in subtle inferences for image processing.

1.4.8 Solid surface medium – crawling (*Heterodera glycines*, *Caenorhabditis elegans*)

We will now conclude our literature review by decreasing the scale of the studied organism to a mm while staying on a solid surface medium. *Heterodera glycines* is also known as the soybean nematode and is a devastating pest in soybean farms. This is why Akintayo et al. (2018)[34] have come up with a deep learning framework to count their eggs in order to improve pest assessment and subsequent management. They use MATLAB and CNNs to differentiate between the eggs and other contaminants that came with the sample from a crop. This allowed them to gain considerable time while maintaining an accuracy comparable to that of trained humans.

Caenorhabditis elegans is also a nematode but, contrary to *Heterodera glycines*, is amongst the most popular model organisms studied today. Its cuticle is transparent, measures 1 mm in length as an adult hermaphrodite, has a short replication cycle, and crawls when on a solid surface, or thrashes in an aqueous medium. These facts are critical in this review because they establish *C. elegans* as an excellent model for automated surveillance. Indeed, by crawling in two dimensions from larva to adult, they can be observed using an X-Y motorized stage coupled with a high or low magnification lens. This organism offers a lot of flexibility in the way it can be observed. Let us first review automation of lifespan assays, starting with Pitt et al. (2019)[35] who devised a robot sequentially recording n = 144 worms (Figure 23).

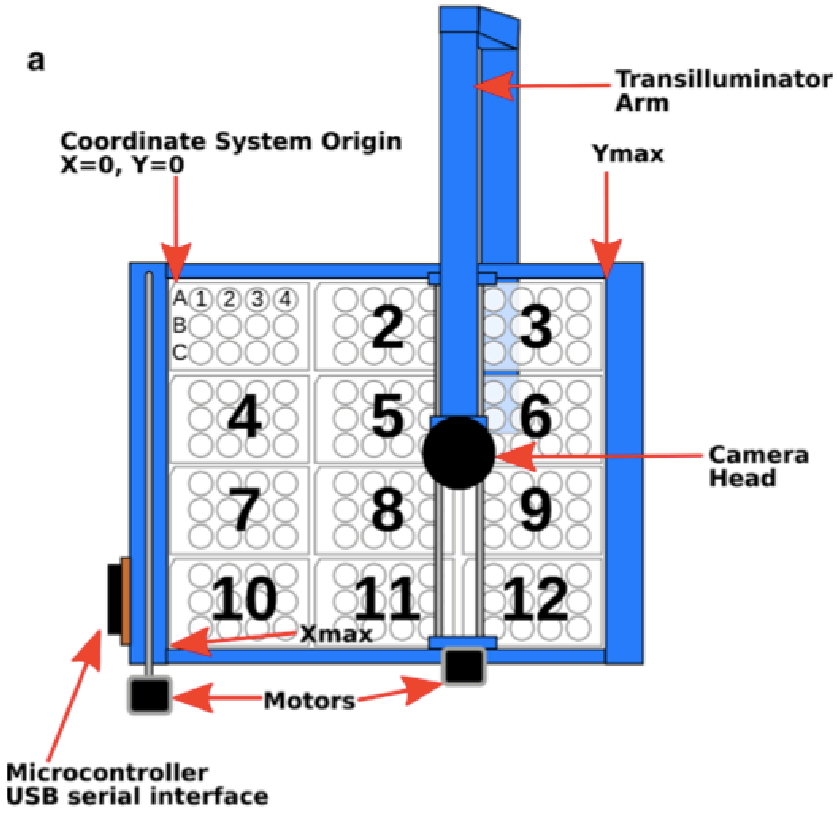

a
Transilluminator Arm
Coordinate System Origin X=0, Y=0
Ymax
A
B
C
1
2
3
4
2
3
4
5
6
7
8
9
10
11
12
Camera Head
Xmax
Motors
Microcontroller USB serial interface

Figure 23: WormBot as designed by Pitt et al. (2019)[35]. *The system is viewed from above. The plates are held by dowel pins and are on top of a clear acrylic table.*

They created a semi-automated software to measure *C. elegans* lifespan: the experimenter is required to mark the center of mass of each animal in the well for the algorithm to go chronologically backwards through the frames until the worm trespasses its final bounding box. One must, however, point out the fact that the arm they designed is heavy and requires a balancer load on top of the camera head. It is the high sample size and thus, statistical power that makes this system powerful at its core. However, where they used regular post-processing to identify the animals' position, García Garví et al. (2021)[36] used CNN (Figure 24).

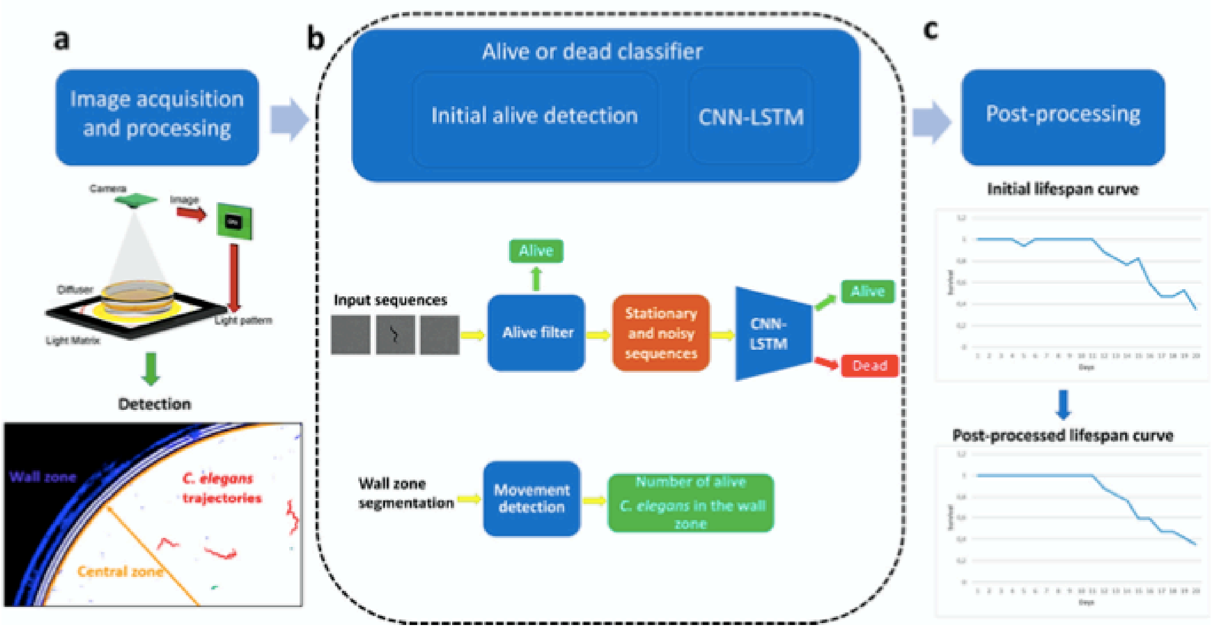


Figure 24: Overview of the method proposed by García Garví et al. (2021)[36]. *(a) Image acquisition step. (b) Classification step using CNN. (c) Data curve and post-processing.*

This, compared to Pitt et al.'s (2019)[35] implementation, is completely autonomous, so no human initial marking is required. They power their setup using a Raspberry Pi 3 computer along with a v1.3 camera (OmniVision OV5647). This choice of design gave them a lot of flexibility: they could easily change parts of their design, like the camera, without having to redesign the whole device. Overall, it achieved an error rate per plate of 3.54% ± 1.30%. Another paper to have achieved successful automation of *C. elegans* lifespan is from Puchalt et al. (2021)[37] and is based on active vision and image processing (Figure 25).

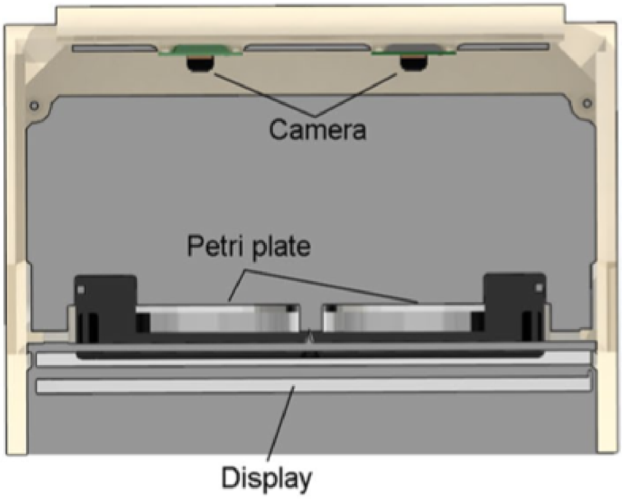


Figure 25: Arenas and camera setup for lifespan assays from Puchalt et al. (2021)^{37} *where their display is continuously adjusted to offer the best contrast between the animals and the background, based one of their previous work on backlighting control by Puchalt et al. (2019)*^{38}*.*

Using this setup in conjunction with their previous work, they were able to perform accurate lifespan assays with this model organism. The last paper on lifespan comes from Jung et al. (2014)[39], who developed QuantWorm, which is a software package especially designed for *C. elegans* and helps perform phenotypic assays (Figure 26).

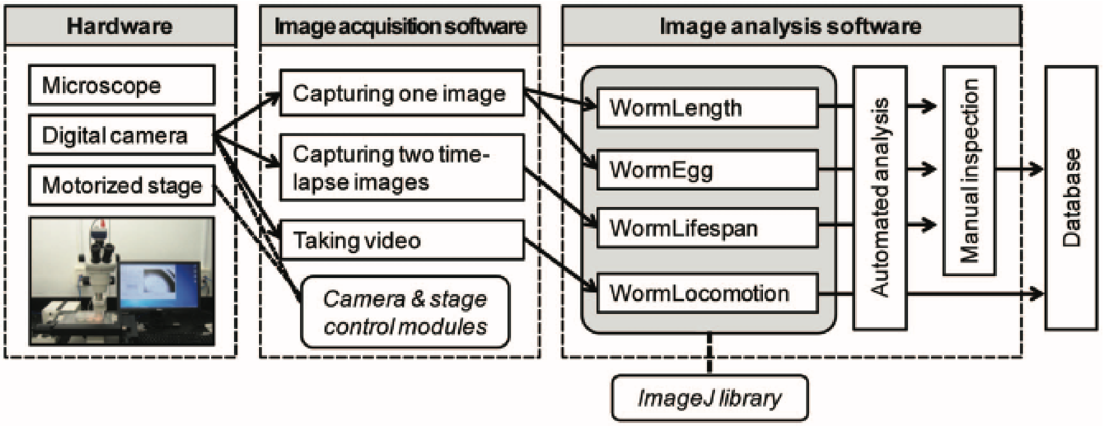


Figure 26: QuantWorm method by Jung et al. (2014)^{39} *which uses a microscope, a digital camera, and a motorized stage to assess such things as the number of eggs in the image, the length, lifespan but also the locomotion of the animal.*

Although this design implements a motorized stage, its sample size is still limited, and it has no control over the temperature. This is important because some strains require observation at 15°C. For their lifespan assay, they take images at the same place on the sample two minutes apart and then perform image subtraction. If there is a significant difference between the two, the worm will be considered moving and thus alive, otherwise it will be considered "most likely dead." Their egg counting algorithm uses no artificial intelligence but rather, a form of clustering to identify the objects with the highest chance of being eggs and remove duplicates (Figure 27).

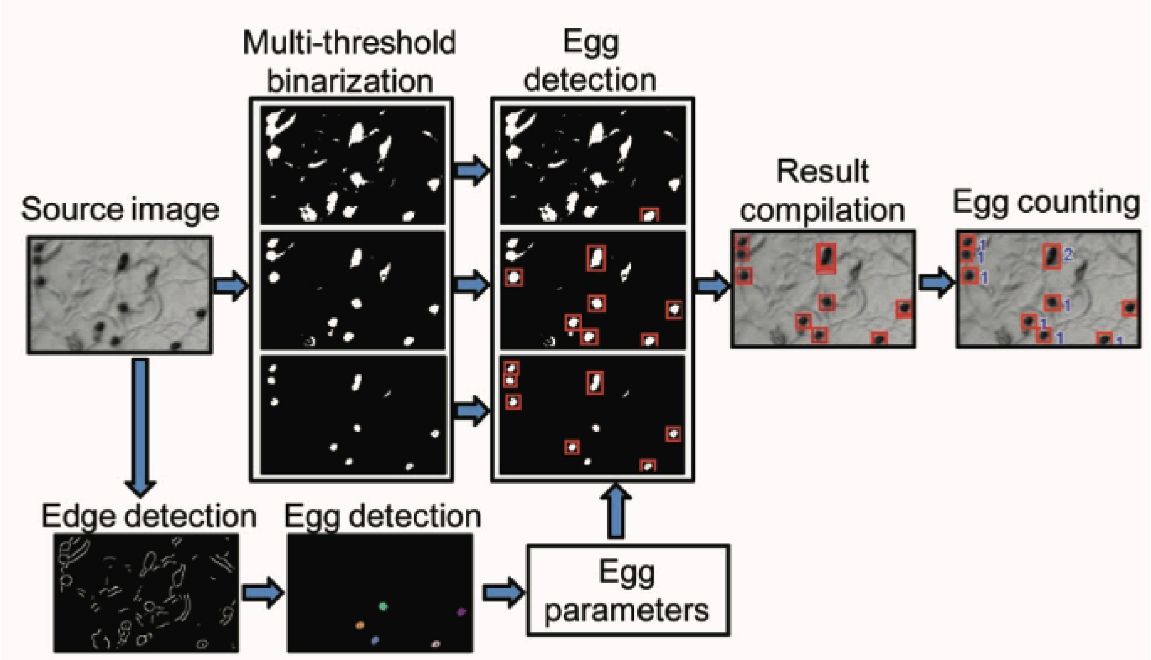


Figure 27: WormEgg algorithmic design from Jung et al. (2014)[39] to count eggs through clustering and duplicate removal.

Thanks to the past few examples, it is now clear that lighting plays a major role in the successful identification and proper analysis of *C. elegans*. Let us dive deeper into what Puchalt et al. (2019)[38] have done to improve backlighting. Using a Raspberry Pi 3, an official Pi display, and a Pi camera for feedback and observation, they were able to change the display's pixels intensity to account for such things as the condensation on the lid or changes in ambient light. In their research, they provide extremely valuable information as they try to find the best SNR, by changing the color of the display. They found that using orange (R = 255, G = 190, B = 0) gave an intensity of 48 to the background while giving an intensity value close to 0 for the animals. It is possible that with a monochrome camera, this contrast could have been further enhanced. In addition to changing the color for better contrast, they also compensated the light so that the illumination would be more uniform (Figure 28).

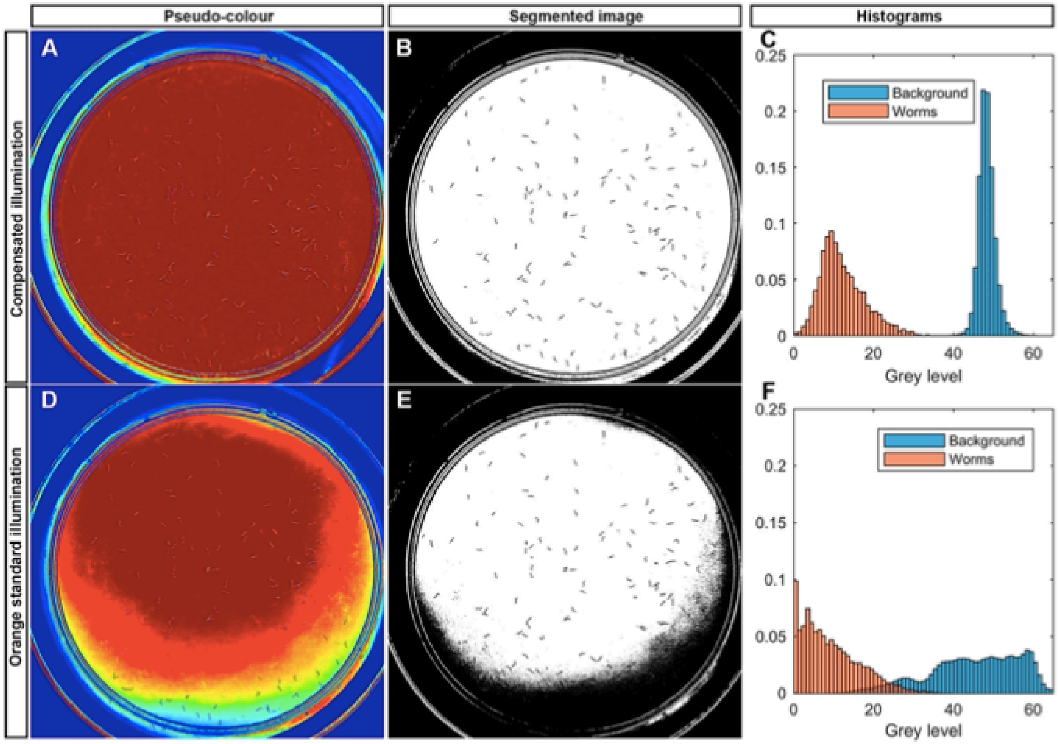


Figure 28: Lighting variability under orange light with and without compensation from Puchalt et al. (2019)[38]. *(A and B) Images taken with compensated illumination, respectively in color and after thresholding. (D and E) mages taken without compensated illumination, respectively in color and after thresholding. (C and F) Grey levels of background and the worms.*

It is quite clear, again, how much of an impact the lighting can have of the studied sample. This is even more true with behavioral analyses of *C. elegans*, where its crawling movement along the agar needs to be parametrized. A. Swierczek et al. (2011)[40] developed their own robot for high-throughput behavioral analyses. In 2011, this was one of the first more advanced robots in that field with a display used as back-illumination as previously discussed. Only two years later, the work of Yemini et al. (2013)[41] helped build a somewhat rudimentary yet efficient monitoring setup (Figure 29). In their motorized design, they chose the camera and light source to be moved rather than the sample. This design choice was probably due to ease of implementation: using the microscopy stand, it is much easier to have the motorized stage on the bottom flat surface rather than have it in the middle of the system. It limits, however, the ability to automatically focus the camera and may introduce artefacts from misalignments between the light source and the camera.

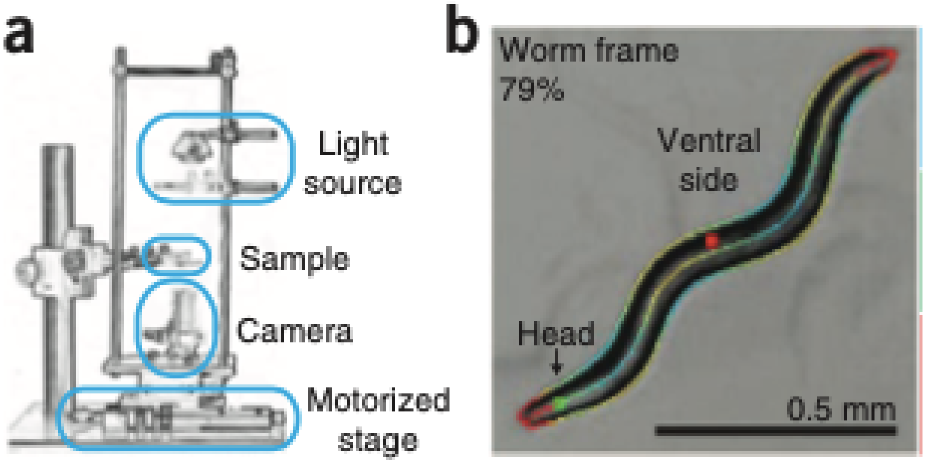


Figure 29: Schematic of the monitoring system (Worm Tracker 2.0) (a) and the resulting processed image to show the contour, head and ventral side of the animal (b) from Yemini et al. (2013)[41].

The same principle has been improved upon by Cermak et al. (2020)[42] who took microscopic images of the free-living nematode with such precision, that their behavioral assays contained data on the pumping rate, egg-laying events, defecation events, body posture as well as linear and angular velocities (Figure 30).

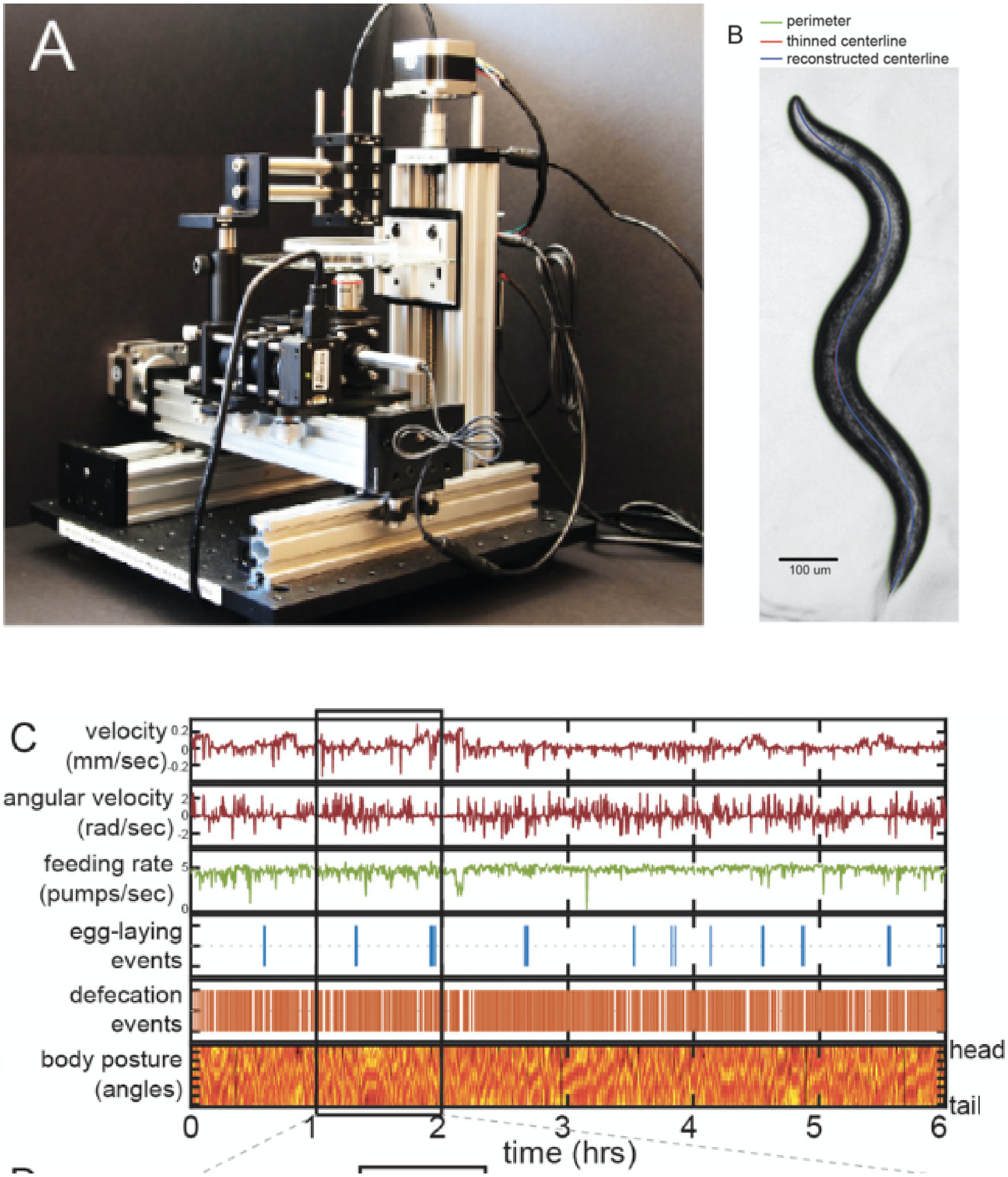


Figure 30: Measurements of several motor programs of C. elegans from Cermak et al. (2020)*[42]*. *(A) Tracking microscope system. (B) Example tracking image of the animal during a behavioral assay. (C) Example data obtained by the robot on the behavior of the animal.*

This research shows that detailed behavior can be monitored in this model to such a degree that behavioral assays could potentially help better understand the full effects of certain mutations.

To give perspective as to the range of monitoring device in terms of price and quality, we will briefly look at a low-cost attempt at a microscope for a smartphone-based mass surveillance system by Bornhorst et al. (2019)[43] (Figure 31).

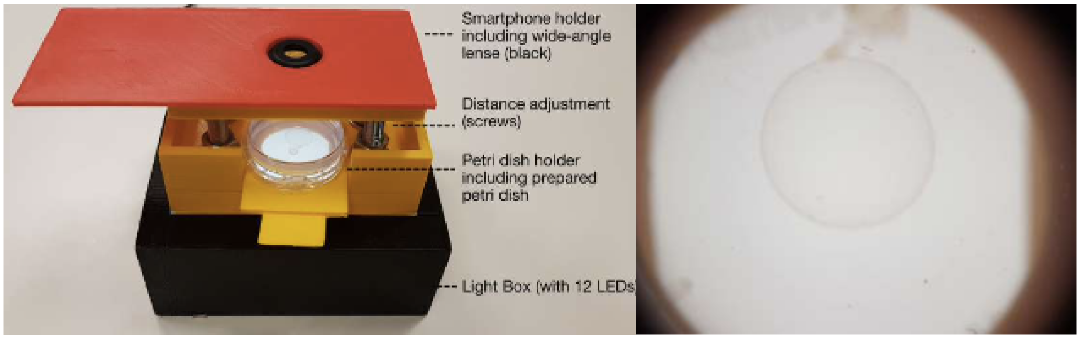

Smartphone holder including wide-angle lense (black)
Distance adjustment (screws)
Petri dish holder including prepared petri dish
Light Box (with 12 LEDs)

Figure 31: Microscope housing prototype from Bornhorst et al. (2019)[43] on the left with the resulting image at the highest magnification level on the right.

They focused on building a smartphone application to go with this architecture, but it had some issues, namely non-uniform backlighting with marker writings occluding some parts of the medium, many false negatives and false positives in animal detection and the arena, in this case a 30mm Petri dish, cannot be seen in its entirety. The latter point certainly lessens the statistical strength of any results they could obtain. They say they can increase the accuracy of their system by "applying manual correction" to the images. However, one could argue that doing so defeats the purpose of having an automated machine analyzing the animal instead of human experimenters.

Coming back to proper automation, we will now review the essential topic of skeletonization in *C. elegans* with a fundamental paper by Cronin et al. (2005)[7] which arguably set the foundations in that field. They proposed an "automated system for measuring parameters of nematode sinusoidal movement" (Figure 32).

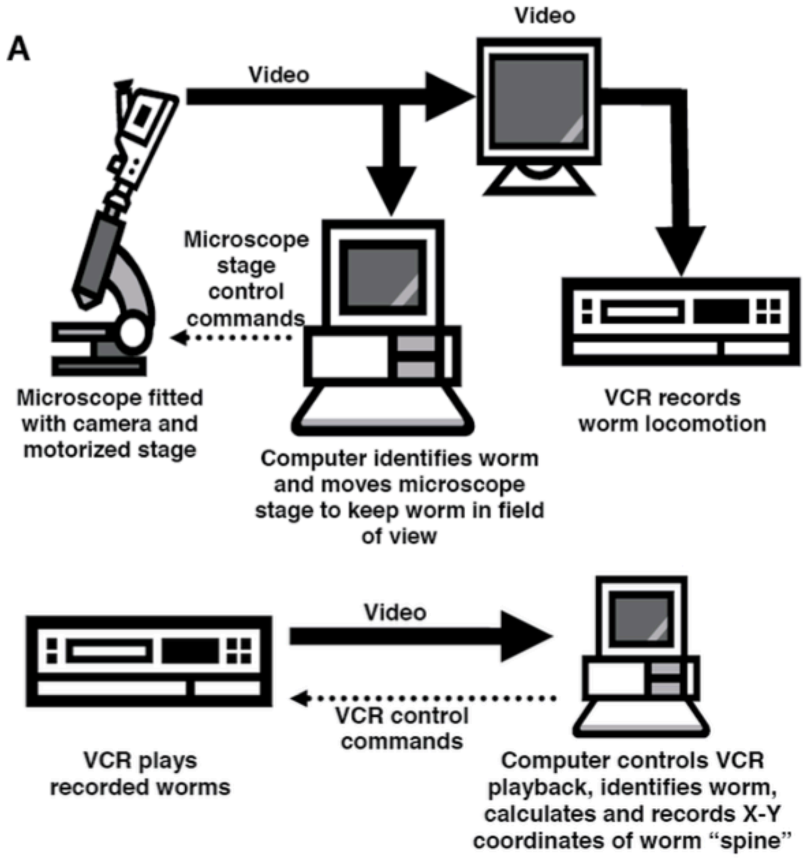
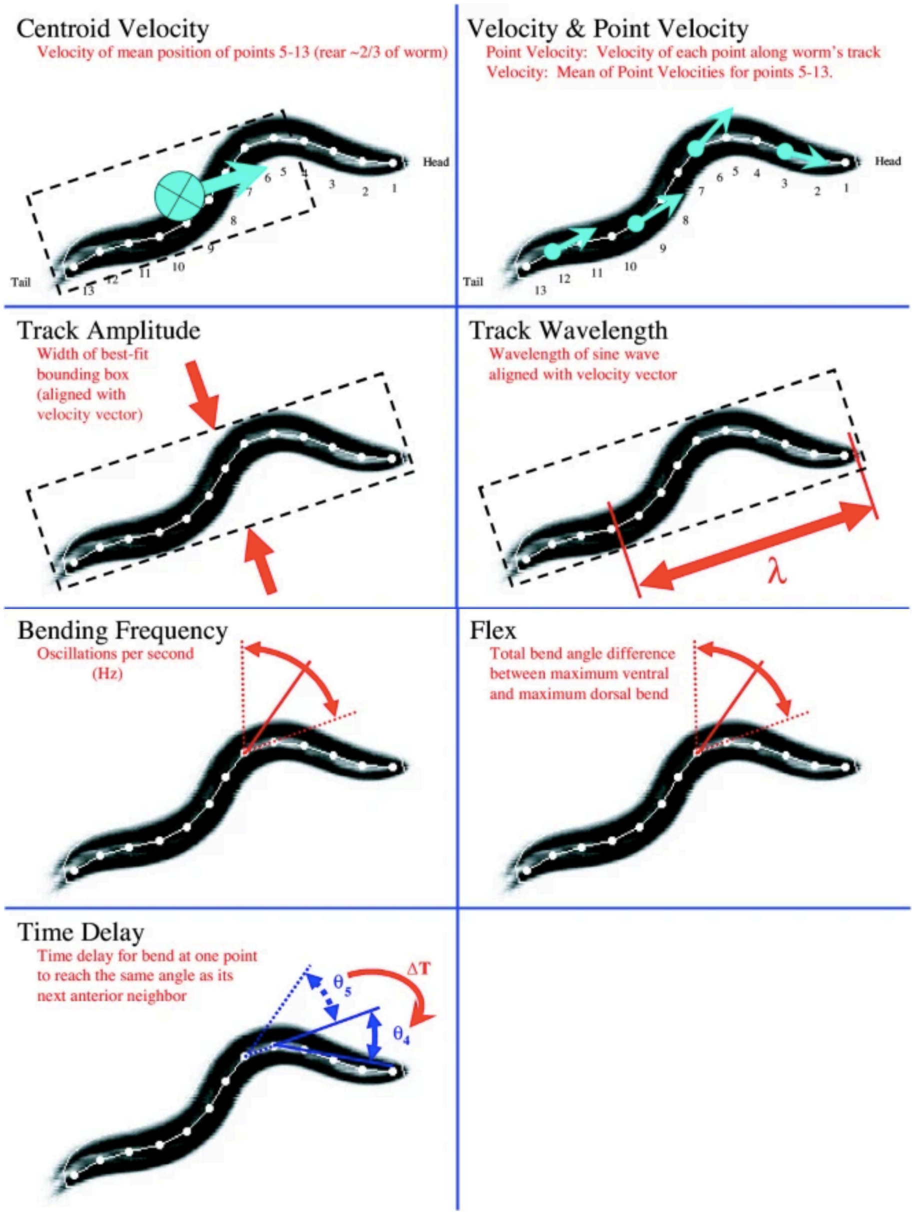

A
Video
Video
Microscope stage control commands
Microscope fitted with camera and motorized stage
Computer identifies worm and moves microscope stage to keep worm in field of view
VCR records worm locomotion
Video
VCR control commands
VCR plays recorded worms
Computer controls VCR playback, identifies worm, calculates and records X-Y coordinates of worm "spine"

Figure 32: Automated monitoring system proposed by Cronin et al. (2005)[7]. *The Petri dish with the worm is placed onto the motorized stage. Once centered, the tracking can start.*

They skeletonize the threshold image of the studied animal by taking the median line of the shape. They then arbitrarily divide the obtained skeleton into 13 points, with one at each extremity. Once this is done, they can establish centroid velocity, point velocity, track amplitude and wavelength, bending frequency as well as time delay (Figure 33).



Centroid Velocity

Velocity of mean position of points 5-13 (rear ~2/3 of worm)

Velocity & Point Velocity

Point Velocity: Velocity of each point along worm's track

Velocity: Mean of Point Velocities for points 5-13.

Track Amplitude

Width of best-fit bounding box (aligned with velocity vector)

Track Wavelength

Wavelength of sine wave aligned with velocity vector

Bending Frequency

Oscillations per second (Hz)

Flex

Total bend angle difference between maximum ventral and maximum dorsal bend

Time Delay

Time delay for bend at one point to reach the same angle as its next anterior neighbor

Figure 33: Automated features extraction proposed by Cronin et al. (2005)[7].

This was adopted by the *C. elegans* community and later improved by Buckingham and Sattelle (2008)[9] who introduced tail vs. head discrimination, as well as head movement. And while some, like S. J. Wang and Z.W. Wang (2013)[44] did not question the 13 points chosen arbitrarily to segment the skeletonization of the animal, others did. Fukunaga and Iwasaki (2017)[45] chose 50 points whereas Krajacic et al. (2012)[46] chose 100 points. Skeletonization might look simple, but several problems needed to be circumvented for it to work reliably, including self-overlapping and coiling. One purely mathematical solution was proposed by Roussel et al. (2014)[47] who designed a “shape estimation that is robust to coiling and entanglement” (Figure 34).

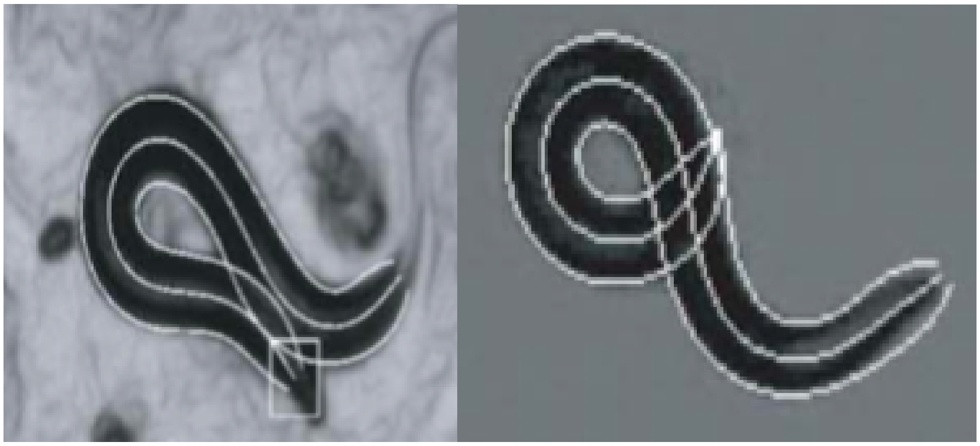


Figure 34: Shape estimation of C. elegans coiling using the mathematical model from Roussel et al. (2014)[47].

The authors pointed out, however, that such a model is very sensitive to input quality. In other words, body deformations or a lack of resolution or an incorrect magnification could have devastating effects on the performance of this tool. This is an issue that Hebert et al. (2021)[48] hoped to bridge by proposing WormPose: a convolutional network used for pose estimation in *C. elegans*. In order not to have to annotate a high number of frames by hand, they thought of annotating a few and then algorithmically create a synthetic but realistic image of the animal coiling (Figure 35).

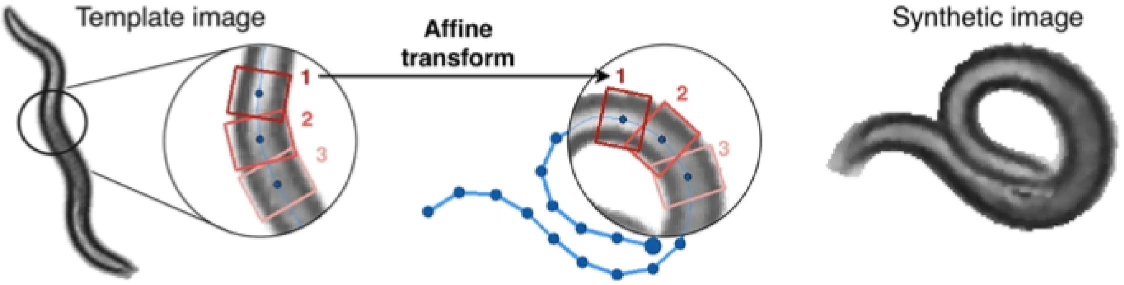


Figure 35: Synthetic image creation from a labeled one to feed to the neural network for training, from Hebert et al. (2021)[48].

They first take videos of the animal and label several frames, then train the neural network with the labeled frames. The network will generate synthetic labelled images to train on before being evaluated on the initially human-labeled images. The model is then used to estimate the pose of the animal and finally the estimation is processed chronologically with other frames to differentiate the tail from the head. These pose estimation systems, along with all monitoring devices, show that it is possible to extract various *C. elegans* features.

1.5 Discussion

There are many takeaways from these papers. We first saw the importance of having an appropriately sized apparatus relative to the studied sample. With some parts of the designs being more expensive, such as filters for fluorescence microscopy, it is crucial to not only design the system ahead of time but also perform several proofs of concept as the project progresses. That way, the risk of going down the wrong design path will be apparent much sooner. We can also gather that the flexibility of any robot generally comes at a cost. For instance, low-cost broad imaging systems will tend to be simple and inexpensive but completely unadaptable to a more specific behavioral assay with one particular model organism. In that sense, although usability across different subjects is attractive, it would make more sense to fully customize the machine around one model only. Across all systems, the common denominators that were especially important to get right were the camera and the lighting. Finally, as any robot is as good as its weakest component, it appears key to optimize every single step through their design (Table 1).

System	Main objective	Advantage	Disadvantage	Reference
Automated monitoring system	Features extraction (e.g. curvature, length, etc.)	Motorized microscopy	Only uses microscopic magnification which excludes whole arena assays.	Cronin et al. (2005)
Worm Tracker 2.0	Extract behavior and morphology features.	Ease of use / installation as the stage does not have to move.	Potential alignment and bulkiness from moving the light and camera rather than the samples.	Yemini et al. (2013)
Low-cost microscope housing prototype	Mass surveillance of a 30mm Petri dish.	Extremely low-cost, essentially requiring a phone.	Does not surveil the entire arena and may need some manual corrections.	Bornhorst et al. (2019)
Tracking microscope system	Motor programs measurements (e.g. egg-laying, defecation, etc.).	Astounding resolution and image quality.	Only uses microscopic magnification which excludes whole arena assays.	Cermak et al. (2020)
WormPose	Pose estimation using trained neural network.	Automaticity in image processing using trained neural network.	Using videos is computationally costly and they use a microscopic magnification, which excludes whole arena assays.	Hebert et al. (2021)
Robotic system (for genetic crosses)	Automated genetic manipulation and analysis of *Caenorhabditis elegans.*	Multiple camera magnification with fluorescence capabilities. Well trained neural network. Sex, phenotype, and orientation differentiation.	Expensive and bulky with no environmental control. This system, albeit extremely recent, seems very powerful.	Li et al. (2023)

Table 1: Summary of *C. elegans* surveillance systems discussed previously.

Thus far, it appears that all *C. elegans* surveillance systems have only one scale (macroscopic or microscopic), and as such, they do not allow for a view of the entire arena and of the animal, both at a relatively high resolution. Being able to have a multimodal magnification would allow to have high quality information on both the brood and its parent. Such information would be useful in determining if there are phenotypical parental traits that could be used as predicting factors for its brood size. In addition, some systems choose to leave Petri dishes facing up, which promotes condensation buildup under the lid. This yields images that are slightly blurry and cannot be used in our application due to the need to maximize image quality. Some avoided the problem by removing the lid, but this is certainly not the best solution since it makes the whole arena and sample vulnerable to contaminants. Also, very few automated systems were designed to fit within an incubator to allow for temperature control. This is essential to experimental reproducibility and further allows to characterize temperature sensitive mutants. Taking these disadvantages into account, we aim to build a system that addresses them all simultaneously.

We have been able to identify components across different surveillance systems that were key to their success and that should be considered in this project:

Illumination: although most authors chose to use LEDs as a main light source, using the Liquid Crystal Display (LCD) provides uniform background lighting. The advantage of LEDs, however, is the possibility of extending the illumination spectrum from the visible to the UV-IR.

Camera: the Raspberry Pi camera was often chosen for direct plug-and-play with the Raspberry Pi itself. It also has a very good resolution, which could be improved by removing the CFA from the sensor, essentially making it monochromatic. It must also be said that better cameras are to be expected on the market in the coming years, allowing for easy iterative improvements of image quality without compromising integration.

Diffusion: this allowed a more uniform illumination on the sample, often in the form of milky plexiglass or glass diffusers.

Computer: the Raspberry Pi seemed to be a very popular choice, likely due to its inexpensive price tag and flexibility in projects requiring code that dictates the experiment process.

Python: both OpenCV and PlantCV, which are Python libraries, were popular choices for the post-processing of images.

CNN: when dealing with more complex tasks, CNNs often seemed to be the most elegant solution.

Open-source: the papers that came from the most reputable journals tended to make all of their source code and designs freely available. This is particularly true in *C. elegans*-based research.

Interdisciplinarity: on average, there were 5.65 authors per publication, which spanned different departments at different universities. This diversity and interdisciplinarity probably constitutes a driving force towards finding creative solutions to new problems and obstacles as the project progresses.

Focus: the ability to focus is essential to get good quality images. The bolt and nut design seems interesting but begs the question of durability with a metal material instead of PVC. Indeed, screwing the lens in and out of its socket over time to precisely change the focal point may result in eventual damage to the socket itself.

X-Y motorization architecture: CoreXY seems to be popular in the custom designs that implemented a large X-Y motorization surface. It always keeps tension and is very precise.

CNN training data set: this dataset must easily surpass the 1,000 scale of images per class for satisfying results.

R: to create heatmaps or simply to automatically generate graphs of the experiment's results, this programming language has a lot of potential.

Post-processing: from the Kalman filter to fill missing data to background subtraction for determining the pumping rate, the algorithms can really improve the quality of the results.

1.6 Methodological considerations

One could argue that we might have missed excellent research or technical expertise, whether it be electrical, mechanical, or algorithmic in nature, that could have been done informally and never got published. However, without a peer-review process, it simply didn't meet our inclusion criteria and therefore would have not been discussed in this review. Also, we found it challenging to find the best way to review automatic surveillance systems across different model organisms. This is because there is such diversity across them that a narrative direction that ties them logically together seems very elusive. This is why we chose to create a narrative based on the model organisms studied, their traditional medium as well as their scale.

Similarly, we found it difficult to estimate the amount of useful information to include related to each reviewed paper. In some cases, mentioning a particular algorithmic process seemed too important not to mention, whereas in other cases, it seemed like a secondary or tertiary focus at best. Finally, we were somewhat disappointed not to have been able to find more recent and sound brood size automated measuring systems.

CHAPTER II: RESEARCH OBJECTIVES

Although manual brood size evaluations are somewhat imprecise, they have been used for the past decades because no other alternative has been proposed. This research is articulated around the following 3 objectives:

2.1 Objective 1: Robot and hardware conceptualization

We first wanted to conceptualize a computer-piloted robot able to monitor an n = 48 sample in an autonomous fashion for more than a week with a high enough resolution to allow it to detect eggs. This will have to be done in real-time while recording environmental parameters such as temperature, humidity, and vibrations. After creating a working prototype, we will upgrade it and make it fit into a small undercounter laboratory incubator.

2.2 Objective 2: Software architecture

We also wanted to conceptualize a software architecture that could not only pilot the robot but also manage and analyze data using artificial intelligence. This objective covers programming language choices but also all the communication logics between the different hardware components (e.g. the Raspberry Pi and the motors).

2.3 Objective 3: Model validation

We will ultimately validate our model and brood size data acquisition using the reference wild type *C. elegans* strain (N2). We aim to have unequaled precision in our measurements and experimental temperatures (i.e. controlled by an incubator during the entirety of the experiment). We also hope to find phenotypical traits in the animal parent (P_0) that have a predictive force on the brood size measurements.

CHAPTER III: EXPERIMENTAL APPROACH

This project, if successful, should represent a significant time gain for the experimenter: instead of spending a whole week counting worms daily, the experimenter would only have to setup the experiment and then let the robot do the rest. Namely, the experimenter is only required to transfer the animals into their respective 24-wells plate's well and then wait for the robot to acquire images for the following week. Such a robot would tick several improvement boxes: time-efficiency, high reproducibility and statistical power over traditional brood size analyses, while it would also allow experimenters to detect potential predictive phenotypical traits for brood size measurements.

Our experimental approach is one of pure research and development (R&D): finding ways to solve the unique challenges from our experimental requirements. Given the multidisciplinary nature of this project, there will evidently be many topics which will require researching and understanding. This is essentially expected to be an iterative process where one researched solution that is adapted to the current project's challenge is tried. If it fails, another one is researched and tried. If it succeeds, then it is used as a building block for other components to be implemented. Although such a trial-and-error process is to be expected, clear overall specifications need to be clarified. This robot must fit into an interior space of 64.8 × 61.0 × 47.1 cm (Width × Depth × Height), while being remotely controlled. It must include stage motorization for $n = 96$ well plates and with a X-Y moving tolerance of no more than 100 μm.

Finally, we want this system to be accessible by anyone, which is why we are making it open-source and trying to keep its cost as low as possible. The code will be available on the laboratory's GitHub page, and the system will have a corresponding bill of materials that is as transparent as possible. Given this accessibility, we hope that other research teams will be able to use it and modify it for their own purposes and respective sample analyses.

3.1 Scientific and biological contexts

The fitness of an organism is defined as its ability to propagate its genome to the largest number of surviving offspring. For oviparous animals that do not care after their progeny, such as fish, amphibians and insects amongst others, it translates into the ability to rapidly lay a large number of high-quality eggs[49]. The regulation of egg laying is therefore key in defining those species' fitness.

Yet, studying the regulation of egg laying efficiency for a species is nearly impossible in the wild due to constantly changing environmental conditions, making it impossible to control for the uniqueness of each individual's life experience. As such, laboratory models are ideal to study the identification and characterization of mutants affecting egg-laying behavior and efficiency. In *Drosophila melanogaster* for instance, when under constant light, egg-laying is rhythmic and follows circadian rhythms[50] which is a mechanism regulated by transcription-translation feedback loops (TTLs)[51]. In the case of *Drosophila*, the corresponding regulating genes are clock (Clk), period (per), timeless (tim) and cycle (cyc)[52]. Yet in dioecious species, male finding and mating efficiency is an additional confounding factor for brood size evaluations. This is because the potential incapacity to find a male and inefficient mating can abrogate reproduction or reduce brood size, respectively.

The nematode *C. elegans* is an androdioecious species consisting of hermaphrodites and rare males. This makes it a model particularly suited for the specific study of egg-laying efficiency. Its reproductive cycle is also short as it lays about 300 eggs within 3 days[53]. This is particularly useful when doing genetic screening where one needs a large population from one isolated mutant animal.

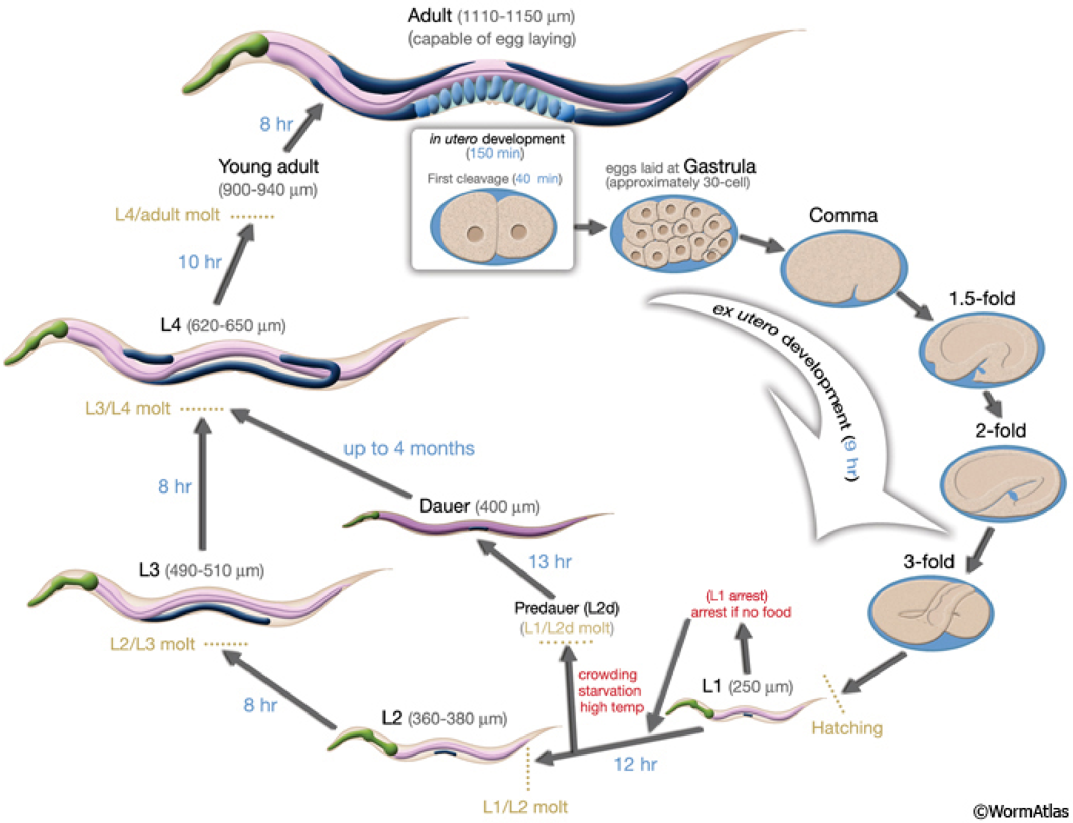


***Figure 36: Life cycle of C. elegans at 22°C[54].** The times indicated in blue represent how long the larva spends at one particular stage.*

However, even in such a clonal population, and within a highly controlled environment, stochasticity results in significant variations in fitness between individuals. Yet, before an individual is well advanced into its reproductive period, it remains difficult to predict which individual within a group of clonal siblings has the highest reproduction potential. Furthermore, the current manual methods to evaluate brood size efficiency are time-consuming, error-prone, and subject to observer biases and visual acuity. As such, to address the question of egg laying efficiency, automation appears essential.

Several systems have been developed to automate surveillance such as the ones discussed in the literature review of this document. However, and even in simple models such as *C. elegans*, their applicability is often restricted, while their design is often non-modular[53]. For instance, the Multi-Worm Tracker (MWT) allowed for the simultaneous analysis of several worms placed in a culture dish[40]. The WorMotel improved on the sampling size of the MWT and specialized in ageing assays[55]. More recently, Pitt et al. introduced the WormBot which aimed to improve the shortcomings of the WorMotel and perform high throughput survival assays; it had larger wells with a moving light and camera system[35]. Cermak et al. (2020) then created an imaging platform to study a nematode's major group functions (*e.g.* defecation rate, pumping rate, egg-laying rate, curvature) continuously for 6 hours[42]. All these surveillance systems are impressive in their ingenuity as well as their capacity to address their own research questions. Our goal is to inspire ourselves from their successes to design a machine powerful enough to perform brood size experiments and modular enough so that other research teams could modify it to their specific needs.

Here, to predict the reproductive success of an individual *C. elegans* based on its traits, we build a robust system that can not only track and perform measurements on the brood of a high number of *C. elegans*, track the parent's phenotypical traits, but also limit the stresses induced by the environment by reducing mechanical stress, light exposure, and temperature changes.

As such, we present the *C. elegans* Surveillance Automated Robot (CeSAR), an integrated system that automatically and simultaneously measures the reproductive success (e.g. brood size, hatching rate, larval growth and survival) of 48 *C. elegans* individuals, using 24-well plates. Our system records simultaneously major motor groups and brood size measurements of the parent, which allows the experimenter to draw correlations between reproductive success with parental phenotypical features. The CeSAR follows an open-source design, with publicly available code, while it can be easily built for less than $1,500 USD using commercially available and custom 3D-printed parts. Although initially designed for the specific study of nematodes, this system could serve as a basis for generating monitoring devices adapted to a range of other species.

3.2 Materials and methods

3.2.1 *C. elegans* strains and media preparation

Animals were obtained from the *Caenorhabditis* Genetics Center (CGC). The wild-type N2 Bristol strain was used as control throughout our experiments. They were maintained using standard method[56] on Nematode Growth Medium (NGM) Petri plates seeded with *E. coli* of the strain OP50. For our assays, we filled 24-well plates treated for meniscus reduction (Eppendorf #0030 722.019 – this product is now discontinued but untreated plates can also be used) with 0.5mL of NGM. After 24h, we added 20µL of *E. coli* (OP50) at the center of each well using a custom 3D-printed template to ensure bacterial lawn uniformity. Each animal was individually picked in its attributed well at the L4 stage.

3.2.2 Robot construction

Space and materials: the robot is an entirely original design that we custom built so that it would function inside a closed undercounter incubator (VRI6P-2(89510-744), VWR, Canada) (Figure

S1) with an interior space of 64.8 × 61.0 × 47.1 cm (Width × Depth × Height). For the lower and upper stages, we used the AL-6061 T6 strong 3/16" aluminum alloy, securely fitted on 2020 T-slotted aluminum extrusion bars. To ensure axes orthogonality and uniform planar focus in our images, angle brackets and M5 bolts were used to connect every extrusion bar with adjacent ones. Linear movement was achieved by using two pairs of carbon steel rods of 1/2" and 3/8" diameter respectively, with corresponding self-lubricating bronze sleeves around each rod. The sleeves were glued using epoxy onto the stage to which they were meant to give motion. To decrease vibrations, rubber dampeners were added on the bottom of each of the four feet of the robot as well as each of the four feet of the incubator. A complete list of all parts used in this construction can be found in the bill of materials (BoM) as supplemental data.

Motors and motion system: we used four national electrical manufacturers association (NEMA) 17 stepper motors in our system. Two were used to precisely operate our examination stage in the X and Y axes while the remaining two were used to operate the focusing systems of the macroscopic and microscopic camera lenses, respectively. The motion system was based off the CoreXY kinematics (https://corexy.com/), which uses two taut Gates Tooth® (GT) 2-6mm timing belts, each controlled by a separate motor.

Light: Two identical LEDs were used in our imaging system: one for each lens. They produced a warm white (~3000K) at 500 lumens with a 40° beam angle and were encased in their respective 2.36" PVC tube with mounted 90° mirrors. The light was collimated using a plastic Fresnel lens placed at a focal length of 215.6mm from the LED and was then immediately diffused by a glass diffuser with a grain of 1500. The lights were controlled manually with dimmers and programmatically using a SainSmart 2-channel relay module.

Sensors: sensors were placed inside the incubator to monitor multiple environmental parameters simultaneously such as humidity, temperature, light intensity, and accelerations. They were connected on a breadboard that received power by and communicated with our main computer. They were also tested against a control to ensure their data output viability (e.g. for temperature, our sensor was evaluated against the incubator's own sensor as well as a laboratory thermometer placed at the level of the samples).

3D-printed supports: for pieces that were necessary to the construction of the robot but hard to craft by hand, we used a 3D-printer with a 1.75mm PLA filament. A nozzle and bed temperature of 215°C and 65°C, respectively were used for all prints. We custom designed and printed the lights holders, the plates support platform, the case of the relay switches, the fan protective cover, a tip holder to perfectly seed round bacterial mats, as well as stabilizer mounts for the lenses. All 3D objects used for this system can be downloaded as standard triangle language (.STL) or .F3D formatted files from our supplemental data. All models were designed by us unless otherwise specified.

3.2.3 Software architectural design

The software we developed is entirely open-source and available on our Github page (i.e. https://github.com/NarbonneLabCS/CeSAR). The architecture we chose for our system is a dual one: we used one Raspberry Pi 4 (RPI) computer to manage the sensors data, X-Y and macro motors and we used a second RPI to manage the micro focus. In this design, our main RPI is the overseer of the experiment's due process and signals to other major components what and when they should act.

3.2.4 Brood size measurements

Image acquisition and processing: we used two RPI High Quality (i.e. 12.3 MPx) camera sensors to acquire our images and films, respectively. Using the Python language, we first moved the stage to position the 16.2mm well in-line with the macroscopic camera. Then, we performed autofocus by turning the lens in place with its own motor and timing belt. After processing the image through our trained neural network, we used the position of the parent P_0 animal to move the stage to the coordinates of the microscopic lens with the additional $\Delta(X, Y)$ from P_0. We finally autofocused that lens and started our tracking and parental behavioral traits acquisition by filming the animal. This process was repeated in a zigzag pattern across the wells to minimize the time wasted while moving. Our system took a 120s film of the parent animal before starting hourly surveillance for ~10 days.

Graphics automation and data analysis: to ensure our data acquisition can be easily and quickly interpreted from thousands of data points, we automated the graphics generation process. The raw data is organized in maintained folders for each experiment and each sample. The graphical representations are automatically generated from that raw data using the R statistical language (r-project.org). To analyze differences between our data points from 2 strains over n=96 samples, we used 2-way analysis of variance (ANOVA).

CHAPTER IV: RESULTS

4.1 Hardware

4.1.1 Overview

The CeSAR is a surveillance robot designed to automate brood size measurements in *C. elegans* from within an incubator. It moves a 3D-printed stage and can accommodate up to 4 standard 24-well plates in a cartesian plane of 236mm (X, 188mm maximum travel distance) by 267mm (Y, 220mm maximum travel distance) (Figure 37A). It controls 2 NEMA 17 stepper motors (i.e. the left α and right β motors) to span the n = 96 wells using a coreXY design (https://corexy.com/) (Figure 37B), operating one timing belt per motor. It uses 2 additional NEMA 17 stepper motors to motorize the focus of the macroscopic and microscopic lenses respectively. These lenses are mounted on IMX477 sensors capable of 4056px × 3040px resolution (12.33 Mpx). The X-Y calibration was done by sending GBRL commands to move the stage in both directions by 100 mm. The sensitivities of both motors were adjusted until a value of δ in GBRL commands to either motor was translated to a traveled distance of δ mm in the respective axis. For the motorization of the camera, the same process was done, except that it was done for degrees (°) instead of mm.

CeSAR comprises an electric box that is located outside of the incubator. This component includes two RPI 4 computers, a power supply unit (PSU), an Arduino Uno prototyping platform with stepper drivers to control the motors, light switches, and a cooling fan. The electric box also contains effectors and sensors: in addition to sensing the interior environment of the incubator, it executes the commands coming from the electric box, namely autofocusing, and repositioning

(Figure 37C). Each camera is coupled with its own light source: a 3000K dimmable LED, with a 40° beam angle. To ensure properly collimated light, there is a Fresnel lens placed at its focal length away from each LED. It is followed by a high-grain glass diffuser for homogeneous lighting of the entire well. The lighting itself can be controlled both programmatically using code and a relay switch, and manually with dimmer switches.

Structurally, CeSAR's frame is made of 20mm × 20mm aluminum extrusion bars which are secured with one another by right-angle aluminum connectors. They hold three functional stages that were laser-cut into xx-thick or xx-gauge aluminum sheets: on the lowest one is mounted the lenses stage with their respective motor. The highest one holds the light apparatus. The one in the middle houses the middle and central sampling stages which respectively move in the Y and X directions. They are secured onto carbon steel rods using 5020 psi epoxy glue on bronze self-lubricating sleeves. The achievable precision in the X-Y plane is 500µm off from an addressed target.

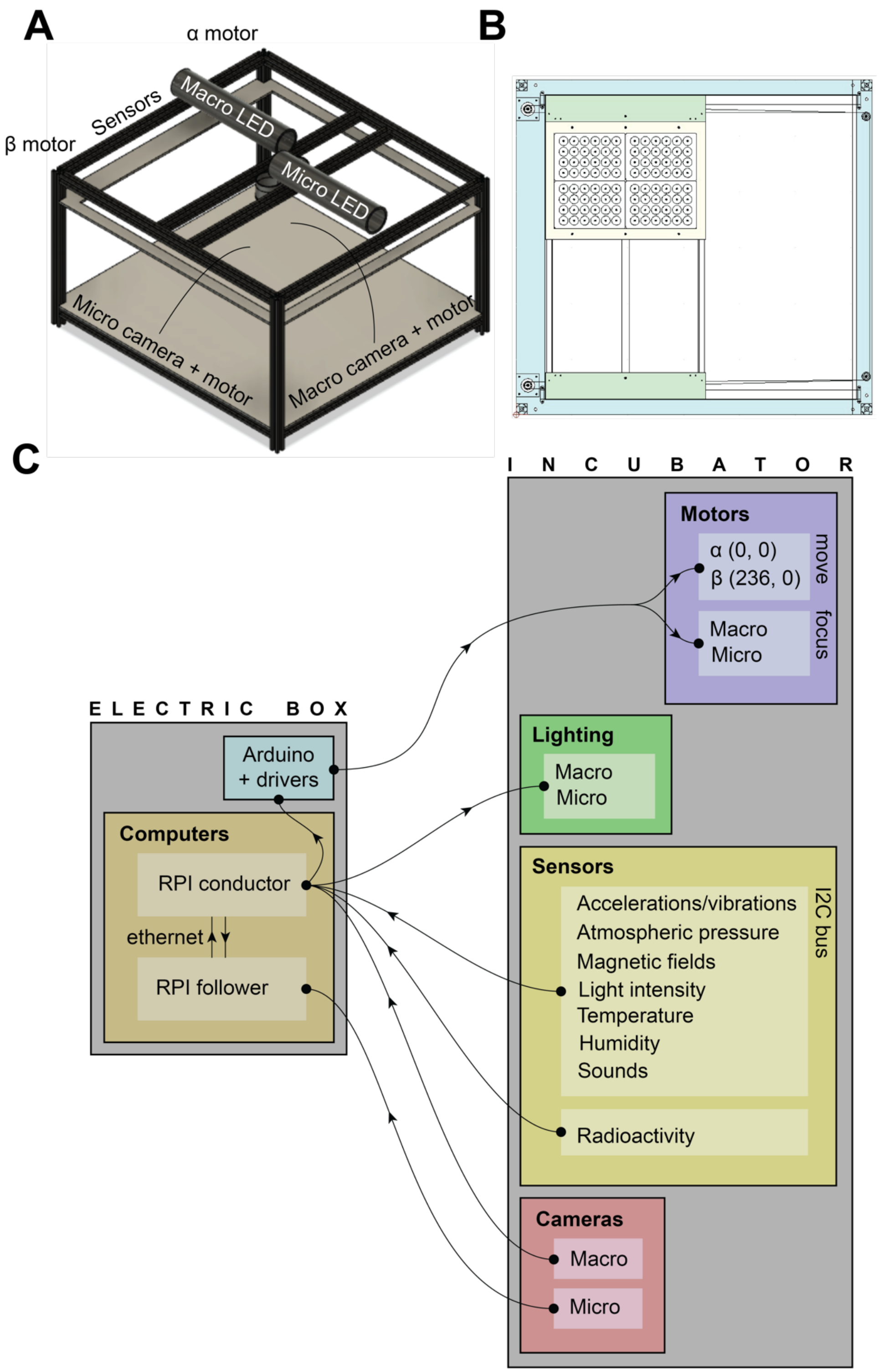


Figure 37: CeSAR construction and design architecture. *(A) 3D-model of the robot in orthogonal perspective. The upper middle level is the stage involving movement. (B) 2D schematic of the middle stage. In light blue, green, and yellow are the outer, middle, and inner stages respectively. (C) Architecture of the different robot's components and the corresponding flow of information.*

4.2 R&D process

This design came to be from an iterative process where the initial setup served solely as a proof-of-concept prototype. It only consisted of a light, two converging lenses, a glass diffuser, a 24-wells plate, and a camera (Figure 38).

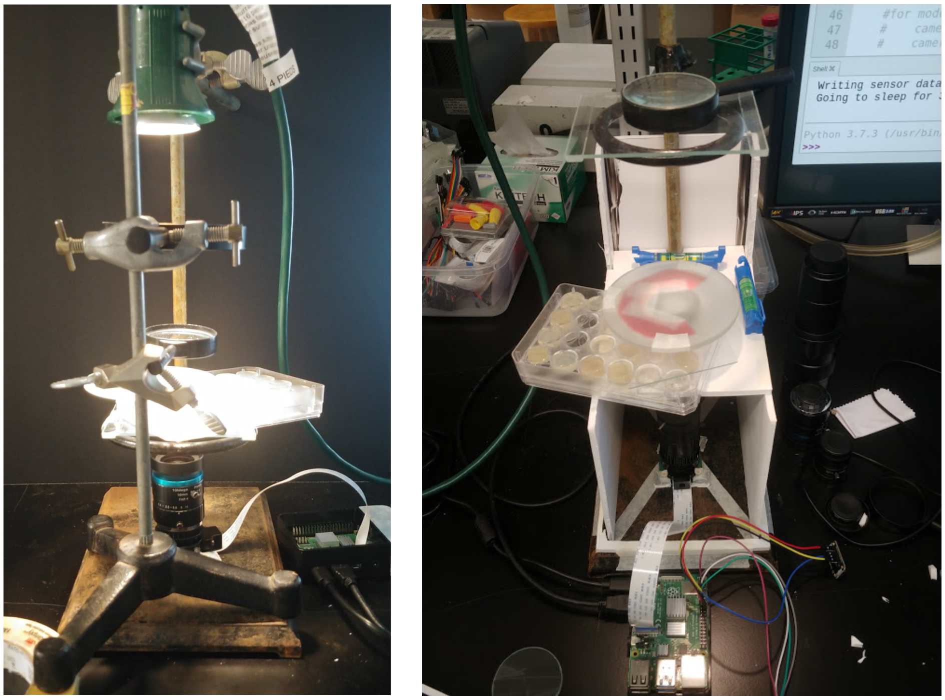


Figure 38: Initial CeSAR prototype without motorization. *First iteration on the left and second iteration on the right (with improved parallelity and controlling 4 independent motors).*

This trial-and-error initial phase showed how crucial it was for the different stages of the system to be parallel to one another. It also helped us determine the amount of NGM necessary for the plates. Having too much NGM hindered the light, which resulted in a darker, less usable image (Figure 39).

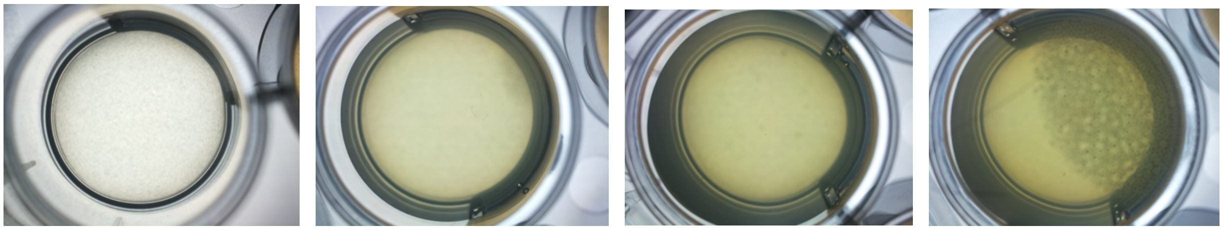
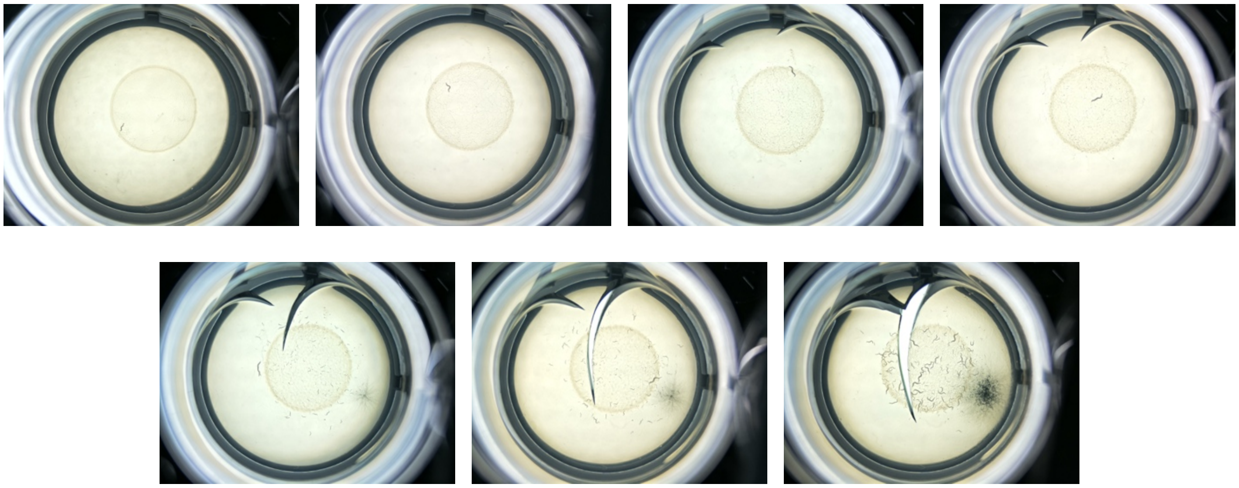


Figure 39: Visible effect of NGM volume on perceived color and light intensity. *From left to right the volumes are 0.5mL, 1.0mL, 1.5mL, and 2.0mL (with condensation forming on the inside of the cover).*

Having 0.5ml of NGM had great optical clarity but was much more prone to desiccation. This process led to the NGM cracking, especially in the corner wells of the 24-well plates. I determined this to be caused by passive air flow accelerating the normal dehydration of the NGM (Figure 40). This issue was important to fix because it meant that one could unknowingly start an experiment, only to find out that the data is unusable because of medium desiccation. I therefore applied paraffin strips to each corner, thereby reducing air flow through the closed 24-wells plate.



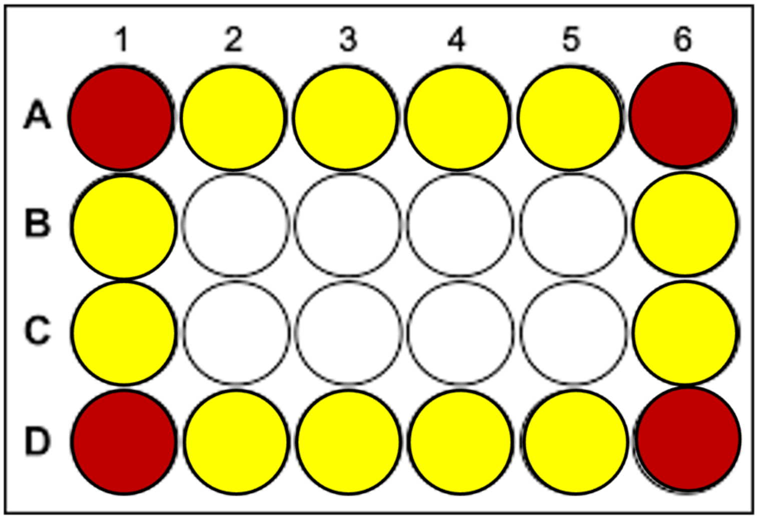


Figure 40: Process of NGM desiccation in 24-wells plate (above) and observed desiccation pattern across the plate (below) Red wells indicate those that dessicate fastest, while wells marked yellow also dessicate, but less rapidly.

This prototyping stage also allowed us to realize that, due to the formation of a meniscus on the edge of the well, a general concave curvature could be observed. This caused a lack of focus and blurry optics near the edge of the wells, which rendered eggs and worms more difficult to detect (Figure 41).

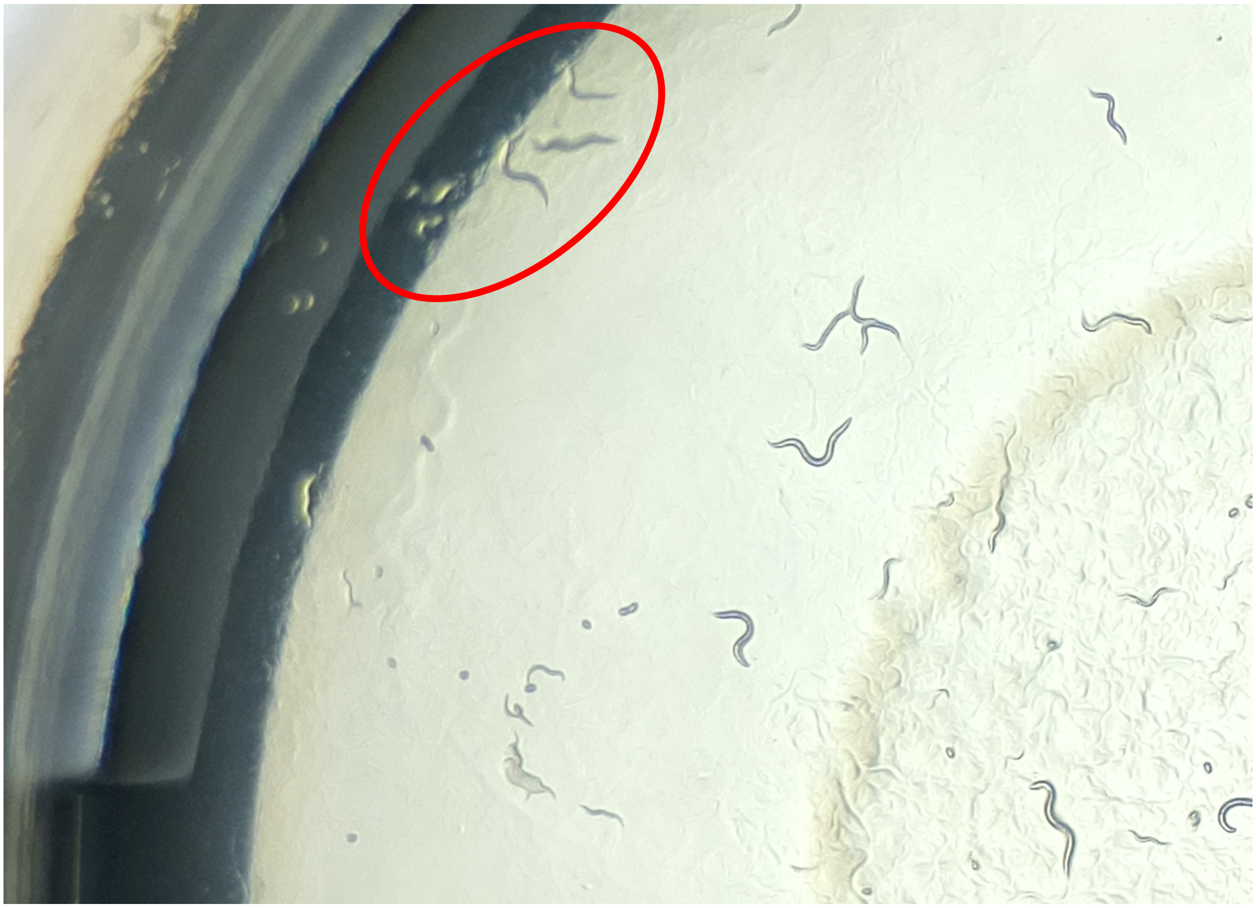


Figure 41: Meniscus formation in untreated well, causing out-of-focus edge ard optical blur. *The worms and eggs near the center of the well (i.e. on the bacterial lawn) are much more in focus than those closer to the edge, circled in red.*

In addition, it became clear that it was necessary to know the dimensions of the smallest object that our system should detect. After analyzing several images, I realized that our detection threshold was to be 10px × 5px (Figure 42). The first larval stage (i.e. L1) Although those dimensions are approximate, they represent a clear requirement for the training model: it being that it must recognize objects around that size reliably.

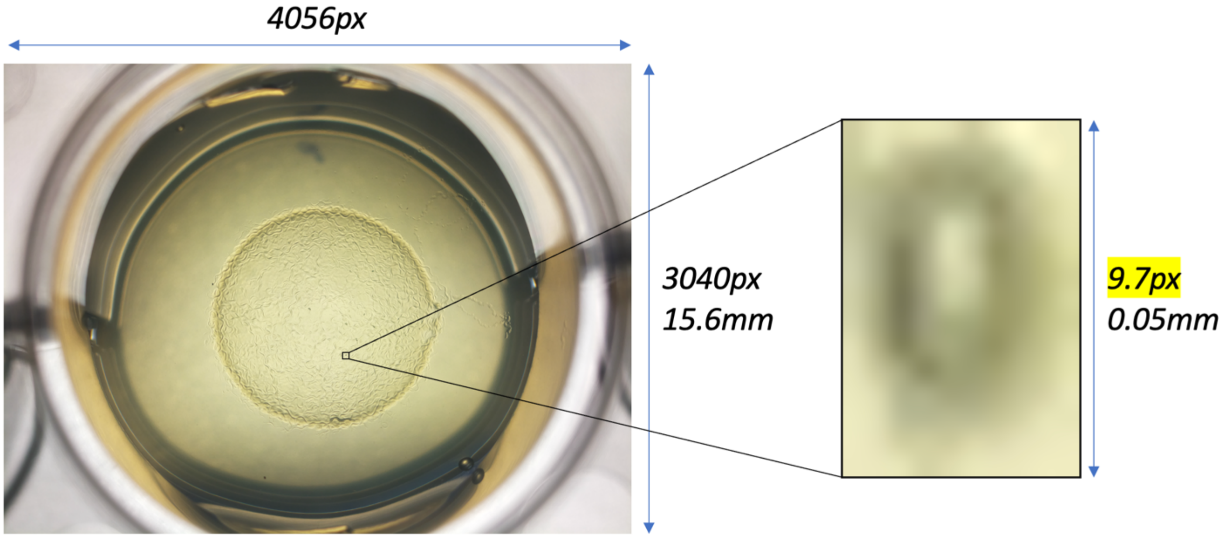
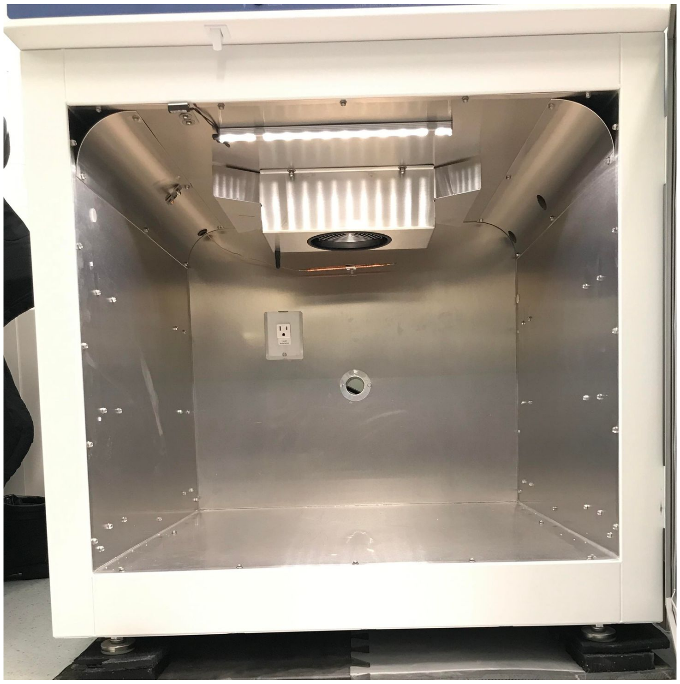
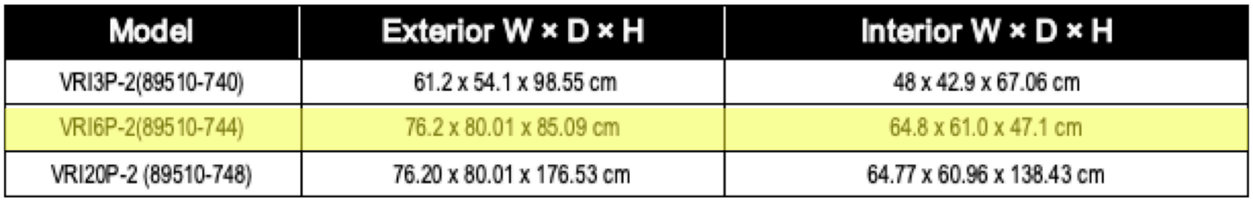


Figure 42: Scale of the smallest object (i.e. an egg) relative to the entire image.

Evidently, the motorization was missing, and only n = 1 sample could be studied. Nonetheless, this was a steppingstone towards adding layers of complexity to the system. Once the software limiting dimensions were known, I evaluated the hardware limiting ones: the spatial limitations imposed by the laboratory's incubator (Width × Depth × Height; 648 × 610 × 471mm; Figure 43). Knowing these was necessary to prototype the final robot and anticipate potential issues caused by the limited space.

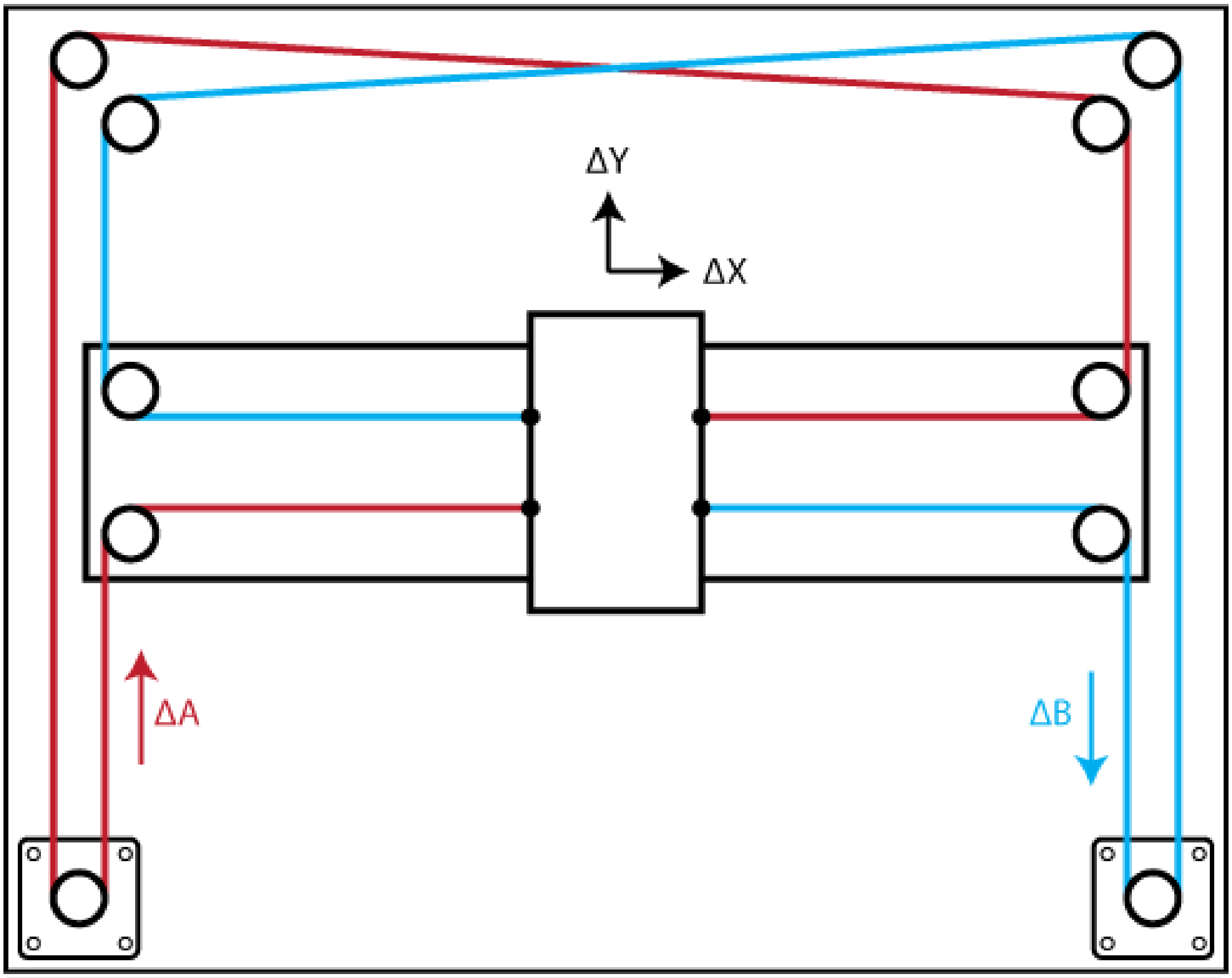




Model	Exterior W × D × H	Interior W × D × H
VRI3P-2(89510-740)	61.2 x 54.1 x 98.55 cm	48 x 42.9 x 67.06 cm
VRI6P-2(89510-744)	76.2 x 80.01 x 85.09 cm	64.8 x 61.0 x 47.1 cm
VRI20P-2 (89510-748)	76.20 x 80.01 x 176.53 cm	64.77 x 60.96 x 138.43 cm

Figure 43: Inside of the small undercounter laboratory incubator to be used (above) and its dimensions (below). Note that the incubator was stripped off from its shelfing system to maximize space.

Given these dimensions and the need to have an open optical path in the middle of the gantry system, I chose the coreXY model (Figure 44) coupled with a 3D-printed design support for four 24-wells plates (Figure 45).



$$\Delta X = \tfrac{1}{2}(\Delta A + \Delta B),\ \Delta Y = \tfrac{1}{2}(\Delta A - \Delta B)$$

$$\Delta A = \Delta X + \Delta Y,\ \Delta B = \Delta X - \Delta Y$$

Figure 44: coreXY theoretical model. *The red and blue timing belts are controlled by the α and β motors respectively.*

If both motors turn in opposite direction and do so at the same speed, then the carriage will only move up or down the Y-axis. If only one of the motors turns, then the carriage will move diagonally toward or away from it. The kinetics are not intuitive, but the model offers two main advantages:

the system does not take much space in the Z-axis, which is especially limited in this case, and the tension on the center stage is constant. This means that, when changing directions, the center plate will always move when either motor turns (i.e. reliable transmission of mechanical motion), without any freeplay. This is not the case with all systems, however, as for screws, going in the opposite direction pushes the carriage only after the space between it and the screw pitch has been moved. This results in delays and lack of precision in the kinetics of such systems.

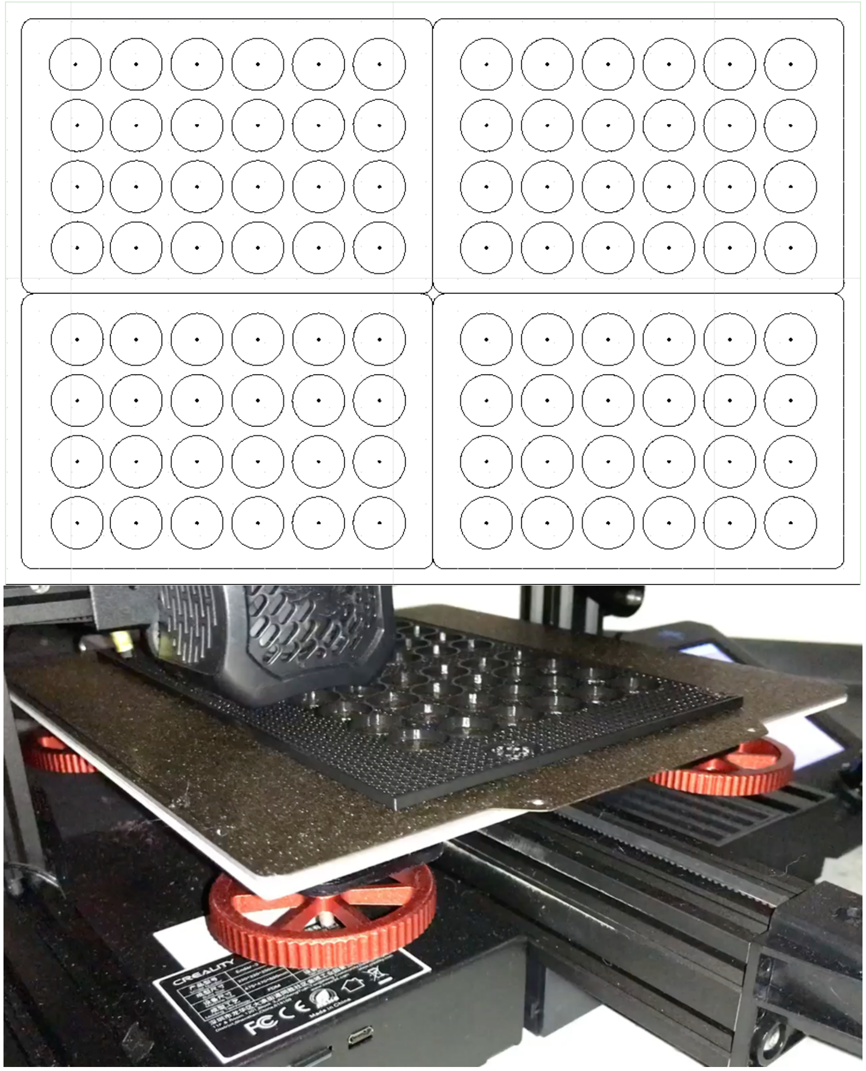


Figure 45: Model of the XY representation of the 4 24-wells plate in the center stage (above) and 3D printing of one half of that model (below).

The following was truly a trial-and-error given my inexperience with 2D computer-aided design (CAD) software: the moving stage. Following the coreXY design, it required a base stage that contained a stage moving in the Y-axis, which itself embedded an X-carriage (Figure 46). The latter holds the 24-wells plates. This trial-and-error process yielded two original designs.

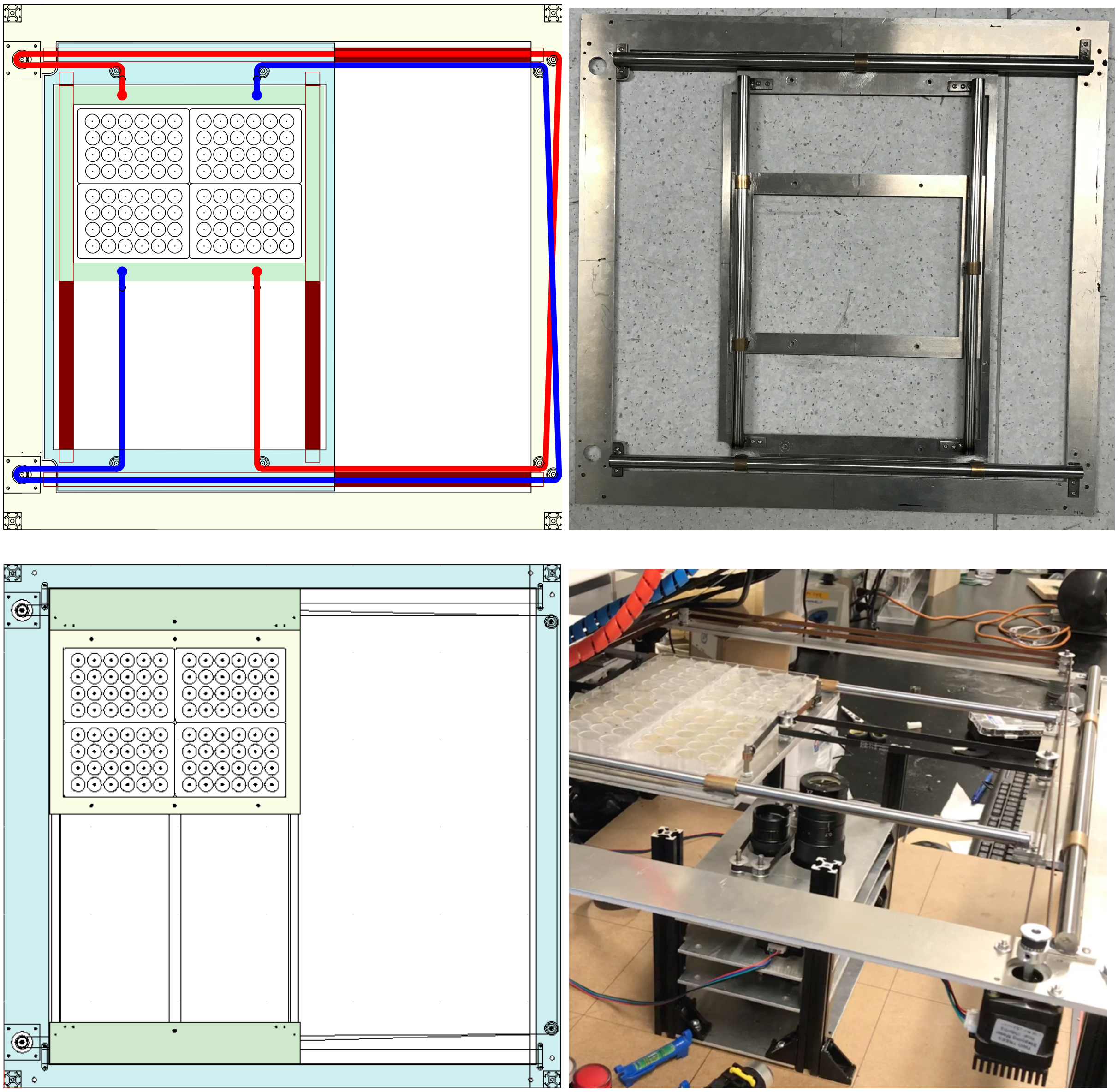


Figure 46: CAD and working prototype for the initial and welded CeSAR stage (above) and final and epoxied (below) CeSAR stage dimensions and motorization.

The first design was flawed as it didn't allow the edge wells to be properly evaluated by the camera. In addition to improving this critical flaw, the second design was much lighter, as it used less aluminum and smaller diameter carbon steel rods, which are extremely heavy (i.e. 7.85g/cm^3).

$$\textbf{Carbon steel density}: 7.85\text{g/cm}^3$$
$$\text{L} = 60{,}96\text{cm } (24") - \text{precut}$$

$$\textbf{base}_{\mathbf{3/8"}} = \pi r^2 = \pi \times (0.9525)^2 = 2.85\text{cm}^2$$
$$\textbf{V}_{\mathbf{3/8"}} = 173\text{cm}^3 \succ 1.358\text{kg}$$

$$\textbf{base}_{\mathbf{1/2"}} = \pi \times (1.27)^2 = 5.07\text{cm}^2$$
$$\textbf{V}_{\mathbf{1/2"}} = 308.9\text{cm}^3 \succ 2.417\text{kg}$$

$$\textbf{base}_{\mathbf{5/8"}} = \pi \times (1.5875)^2 = 7.92\text{cm}^2$$
$$\textbf{V}_{\mathbf{5/8"}} = 482.6\text{cm}^3 \succ 3.784\text{kg}$$

Change: $3.784 \times 4 = \underline{15.136\text{kg}}$ to $2.417 \times 2 + 1.358 \times 2 = \underline{7.55\text{kg}}$

Theoretical (precut) mass optimization: 50.1%

I also improved on the coreXY design (Figure 44) by unifying the idler pullies on the corners of the outer stage and the sides of the middle and center stages. The latter changed the applied forces from 4 points (i.e. two on the middle and center stages respectively) to 2. To reduce torquing of the center stage, which at this point is being pulled open to the side and upwards, a piece of metal was used to keep the two anchor pins solidly attached to one another. The following challenge was to motorize this design. To accomplish this, we opted to use NEMA 17 motors with 200 steps per full rotation (i.e. 1.8° per step precision), with an Arduino board and a computerized numerical control (CNC) shield, controlled by a RPI 4 model B (Figure 47). Before choosing the final

TMC2208 stepper motor drivers, which are powerful and silent, we tried the traditionally used A4988 and DRV8825 drivers (Figure 47).

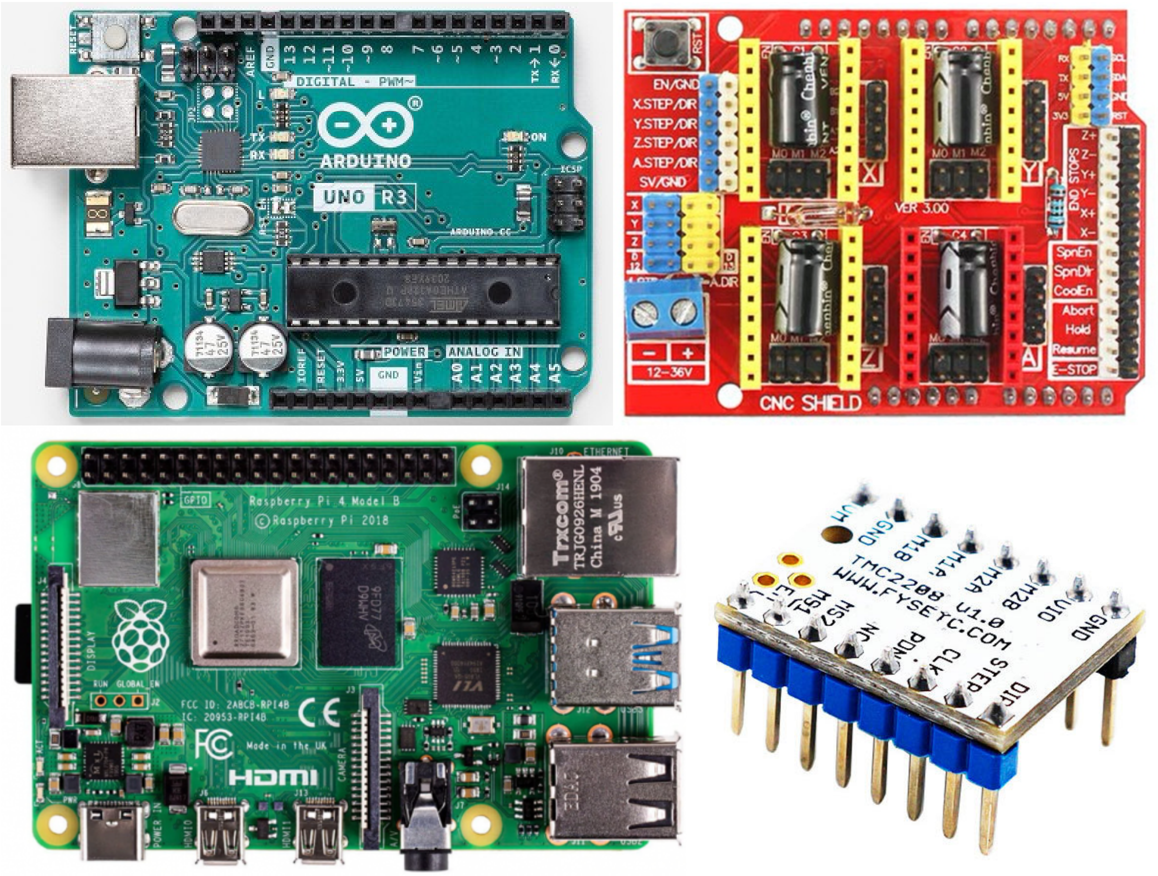


Figure 47: Computing hardware necessary for controlling CeSAR (i.e. motorization and logic of image acquisition). *From top left to bottom right are the Arduino Uno, its CNC shield, the RPI 4 Model B computer, and the TMC2208 stepper motor driver.*

Through the iterative process of choosing the correct stepper motor driver, the V_{REF} needed to be adjusted accordingly. The V_{REF} is the reference voltage at which maximum current will flow through the motors. It is calculated differently for each driver.

<u>A4988</u>:

$$V_{REF} = I_{max} \times 8 \times R_s = 1.2 \times 8 \times 0.1 = 0.96V$$

DRV8825:

$$V_{REF} = \frac{I_{max}}{2} = \frac{1.2}{2} = 0.6V$$

TMC2208:

$$V_{REF} = \frac{I_{max}}{1.41} = \frac{1.2}{1.41} = 0.85V$$

where I_{max} is the maximum current flowing through the motors and

R_s is the current sense resistor of the driver.

Allowing for higher voltage results in increase of motor temperature and higher torque. The inverse is true as, if you set the voltage too low, the motors will have limited torque which will lead to skipping steps. This will, in turn, cause imprecision when moving the center stage. This project evolved into using two RPI and two cameras: one for macro scale images (i.e. the entire well) and the other for micro scale images (i.e. the length of the animal, of maximum ~1.1mm). Even though the prototyping was promising for the micro camera, this addition was ultimately removed from the project as the lens objective amplified room vibrations over its length when placed upright. This was not the case of the macro lens due to its lightweight and compact design. One can also notice that the main RPI is responsible for sensor data and does not control the motors directly. Indeed, the latter are sent specific instruction by stepper motor drivers, which are connected to the CNC shield on top of the Arduino Uno (Figure 48).

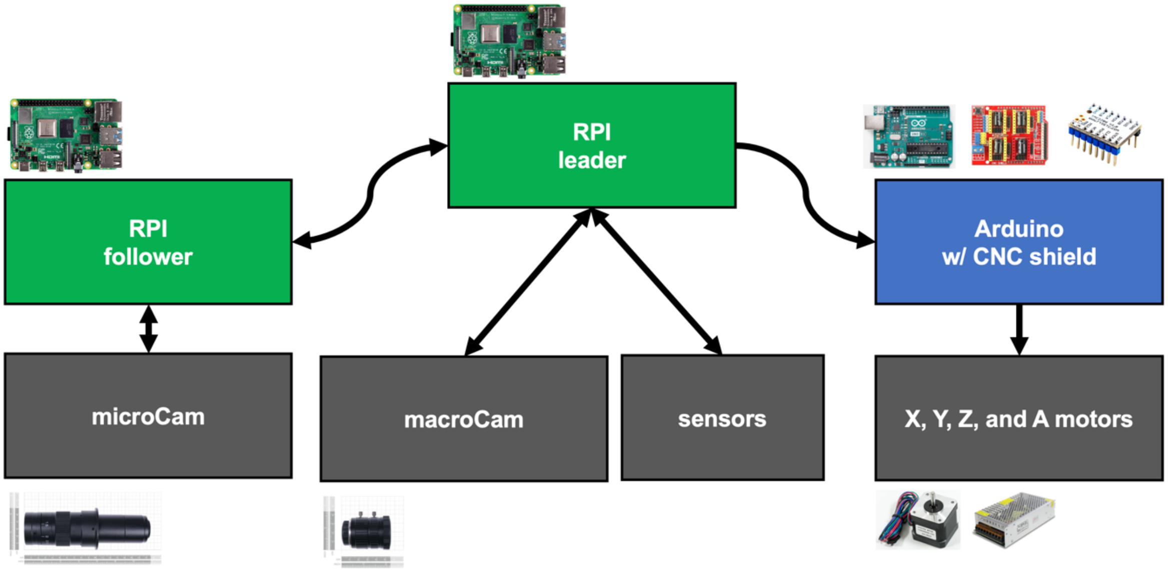


Figure 48: System hardware architecture. *This diagram shows an additional RPI and camera (left) which were not included in our final product.*

I used a LEDMO DC power supply of 240W (i.e. 24V, 10A) to power the motors, as an external electrical source through the CNC shield, and a fan that cools the system. Since the latter only takes 12V, I used a 24V to 12V step down converter. The temperature of the RPI was monitored was ~46°C after 4h of computational work, controlling all 4 motors. The black and red wires connect the microswitches to the CNC shield (i.e. signal the system's spatial limits; discussed below) and the blue-green-red-black wires connect the motors to the stepper motor drivers. The wires that capacitate the power supply (Figure 49, 3 leftmost connector wires on the power supply) were taken from a 18AWG power cord.

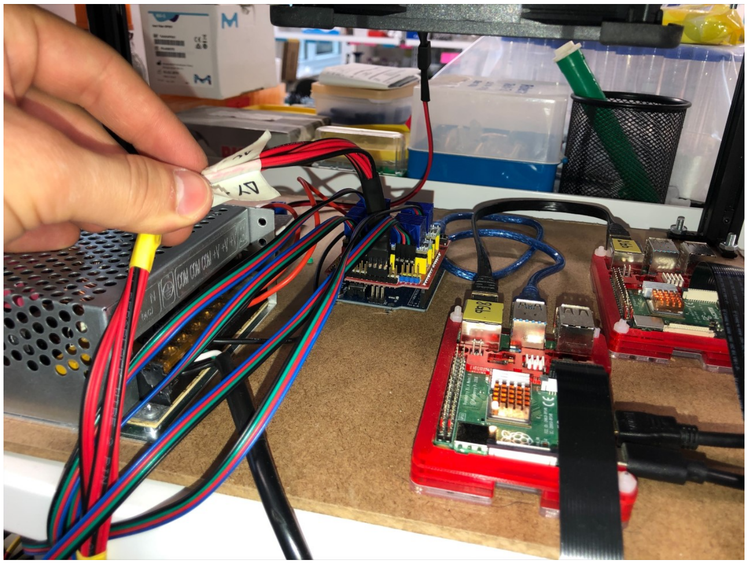


***Figure 49: Setup inside the electrical box, where every component is screwed on the base.** From left to right is the power supply, Arduino UNO with its CNC shield and 4 stepper motor drivers, the main RPI and accessory one.*

To connect to the PSU, the wires of the 18 AWG power cord were stripped and transformed into end wires ring terminal connectors (Figure 50).

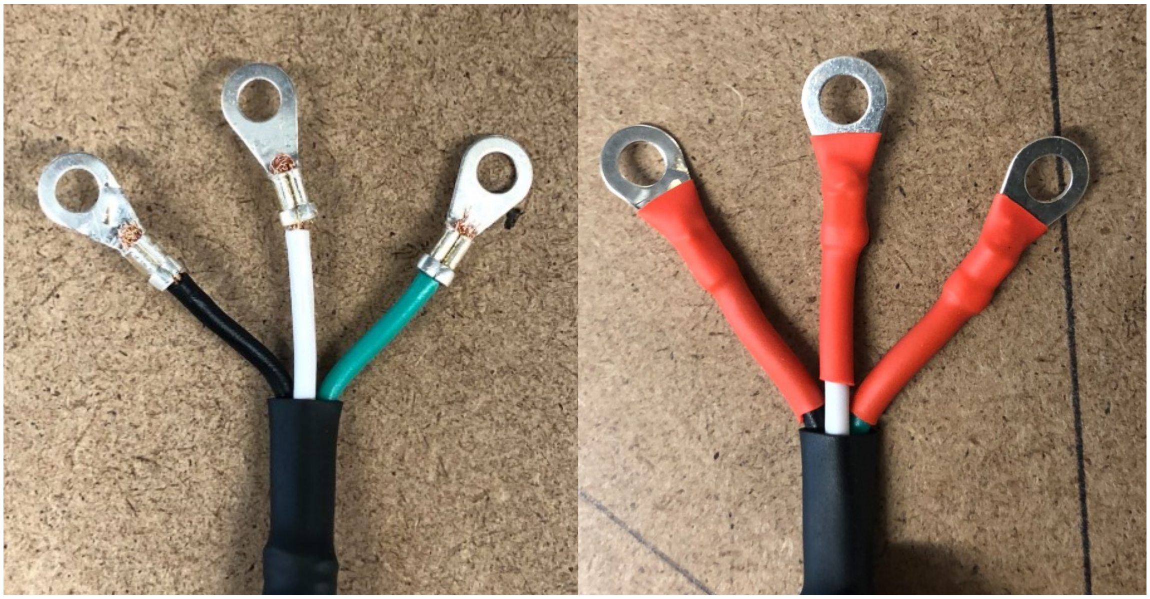


Figure 50: End wires ring terminal connectors used to power the motors and fan's power supply.

The bipolar motor wires were too short, so they were extended such that they would be able to come out the back of the incubator to the CNC shield, on top (Figure 51).

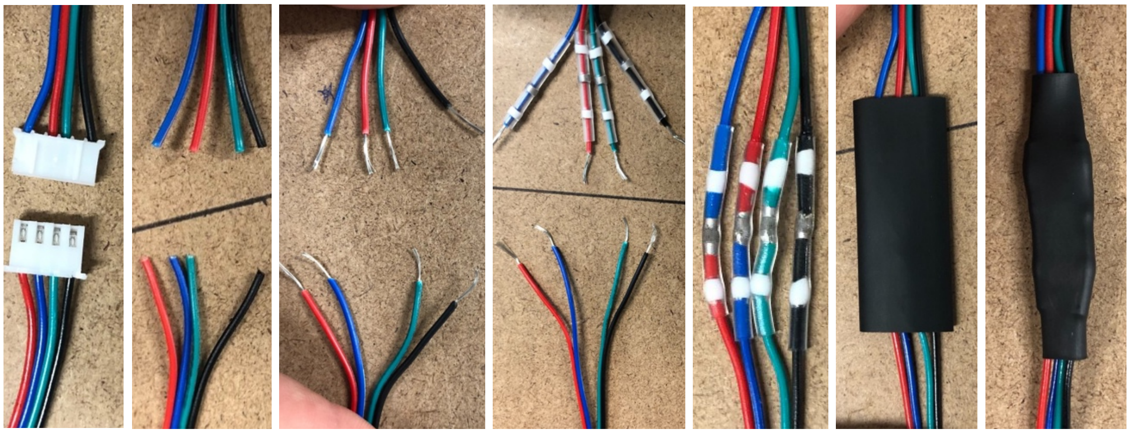


Figure 51: Process of extending the bipolar motor wires by cutting, stripping, and isolating them using thermosensitive electrical tape.

The motorization also comprises the cameras themselves since they were, at the time, the only ones available at 12Mpx, but unfortunately, they lacked autofocus capability, a feature that became available only just recently. I designed a way to control how screwed-in the camera is at any given time to precisely adjust the focal distance. A GT2 6mm timing belt is reversed and securely attached to the lens. A normal timing belt is used to either screw in or out the lens its socket (Figure 52). This essentially increased or decreased the distance between the camera sensor and lens, which moved the focal point accordingly.

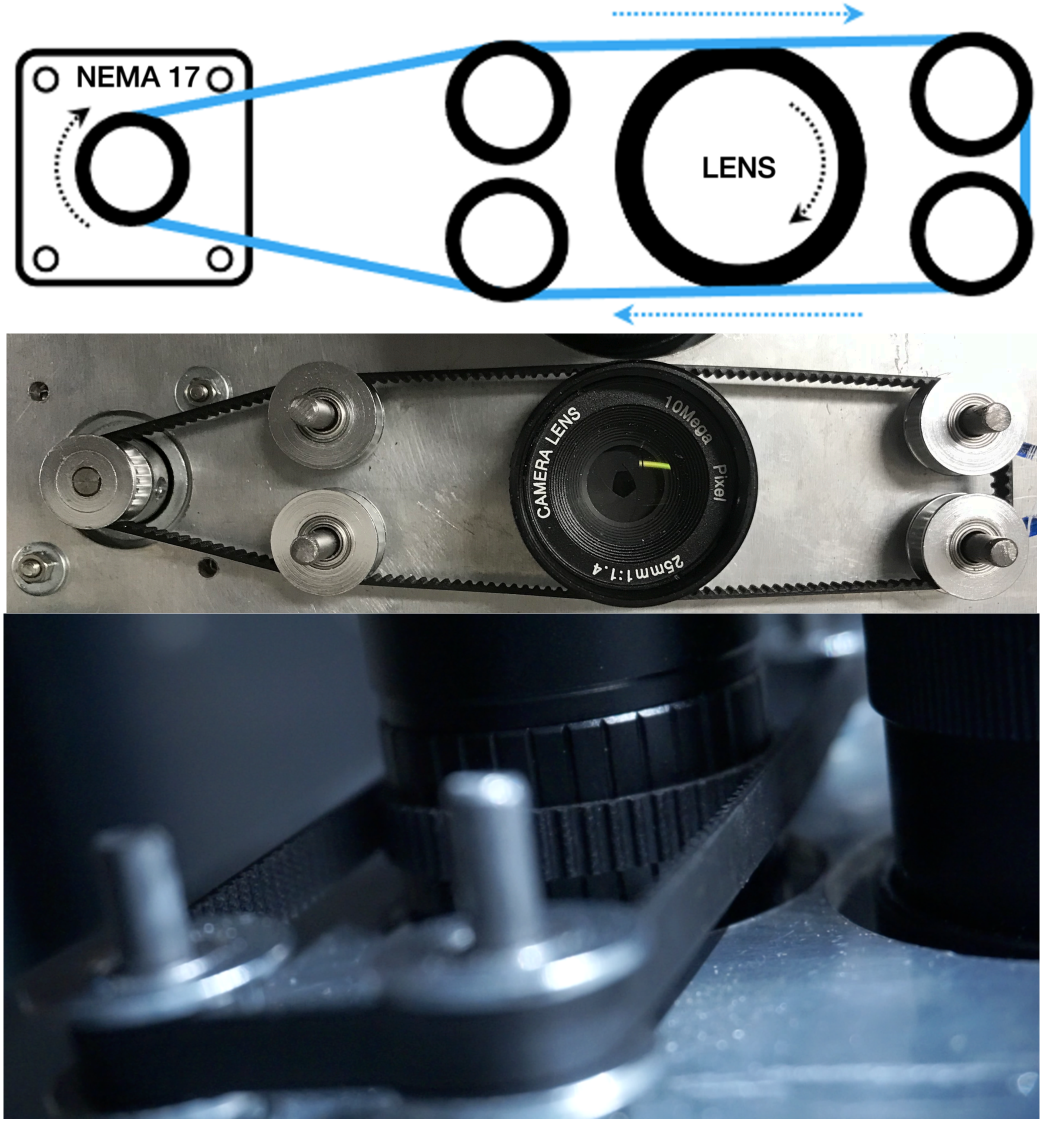


Figure 52: Lens motorization for focal distance adjustment from design (above) to implementation (middle and below).

When the motor turns, it forces the taut timing belt to move in the corresponding direction. The idle pulleys allow for proper contact between both timing belts. In addition to the cameras, motorizing the entire system was a key step in establishing this project as a validated proof-of-concept. However, as with a vast majority of gantry systems, it required a way to initialize itself, or always be able to find a "home" coordinate value. This value is understood to be the (X=0, Y=0) point of the coordinate plane. To achieve this, I carefully placed microswitches on the underside of the outer and middle stages, to respectively detect the Y and X axes travel limits (Figure 53).

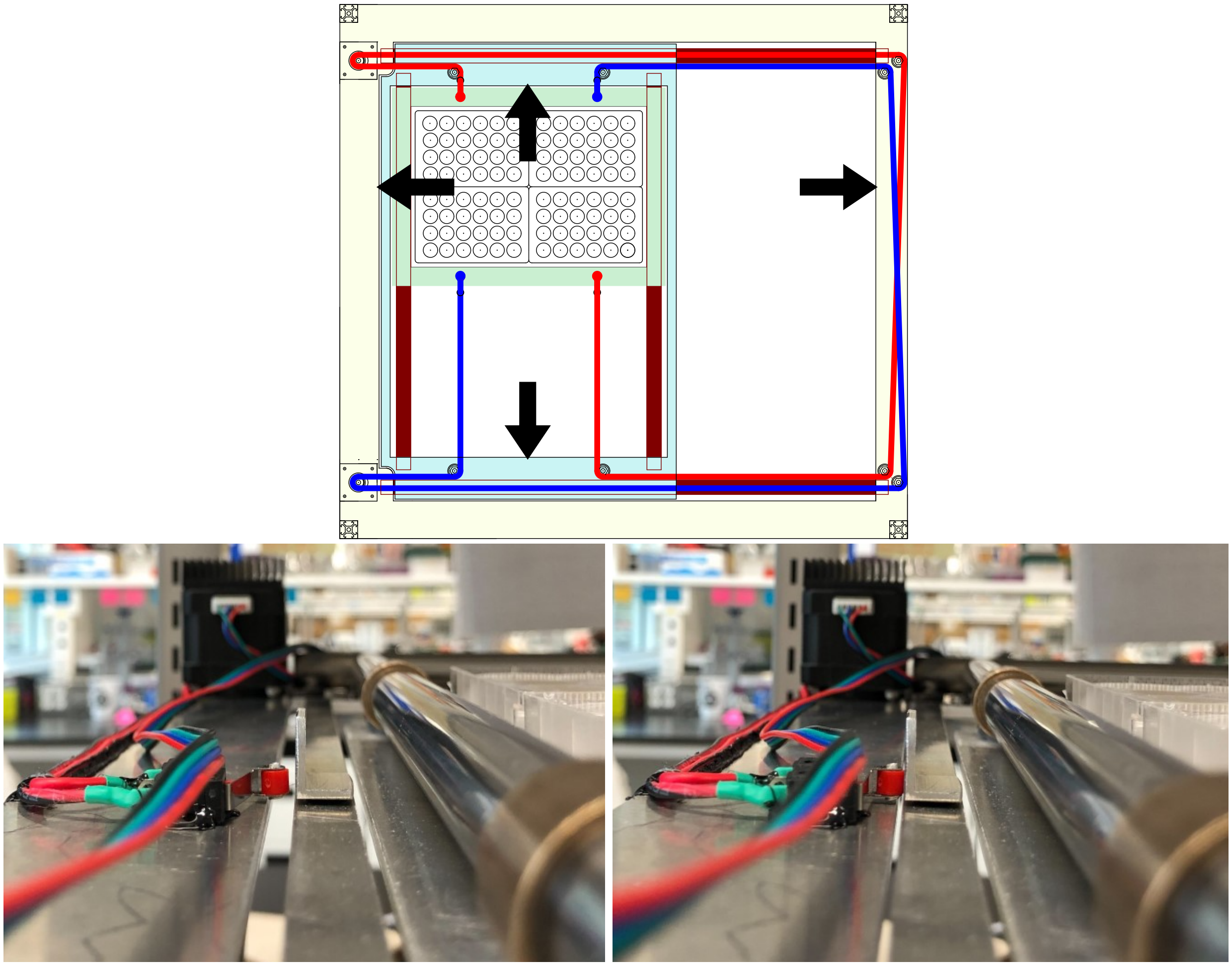


Figure 53: Microswitches placement (above with arrows) and activation (below, activated on the right).

This step is crucial to be able to get to an absolute (X=0, Y=0) or "home" coordinate from any position (Figure 54). This process becomes equivalent to moving towards X=0 until triggering the X-axis microswitch and the doing the same in the Y-axis. These microswitches can be wired to be either normally open (NO) or normally closed (NC). In our case, I designed the wiring to be NC, which means that, when a microswitch is triggered, the circuit becomes open. I used 22 AWG extension wires to connect the microswitches to the CNC shield.

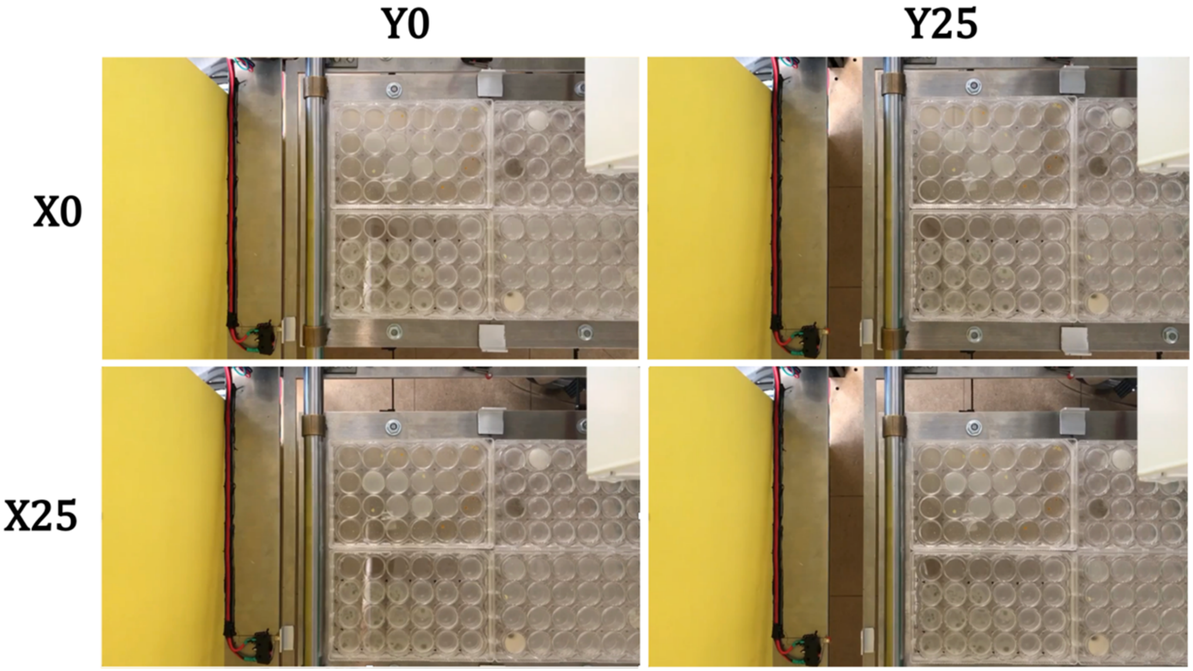


Figure 54: Different initial positions of the center stage before "homing" the system. *In each of the four cases, the center stage touches the X=0 microswitch and then the Y=0 microswitch. Once that's done it appears as the top left image.*

Having full control of both the positioning of the plates as well as the cameras, I then worked on lighting. The lighting system was designed to minimize light leakage onto wells that aren't currently being examined. To do this, I designed a light tube out of an acrylonitrile-butadiene-styrene (ABS) tube used for plumbing (Figure 55).

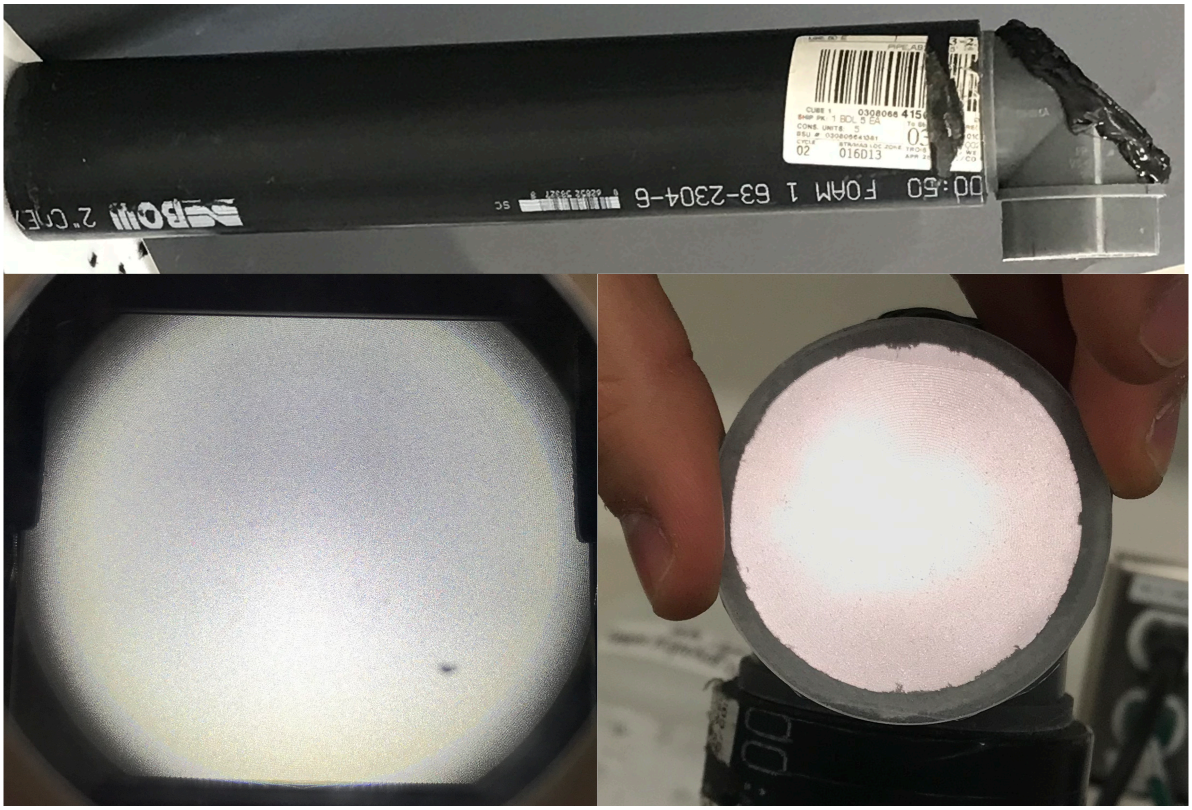


Figure 55: Lighting apparatus made from an ABS 1.5 inch plumbing tube (top). Illumination comparison between a ZEISS Stemi 305 compact stereo microscope backlight (left) and custom made CeSAR's ABS tube (right).

Two key components were then added to this apparatus: a mirror to redirect the light and an extension. The mirror was made using a diamond tip glass cutter and a bench grinder (Figure 56). A custom 3D-printed tube extension was also fitted on the end to bring the diffused light as close to the well as possible (Figure 56).

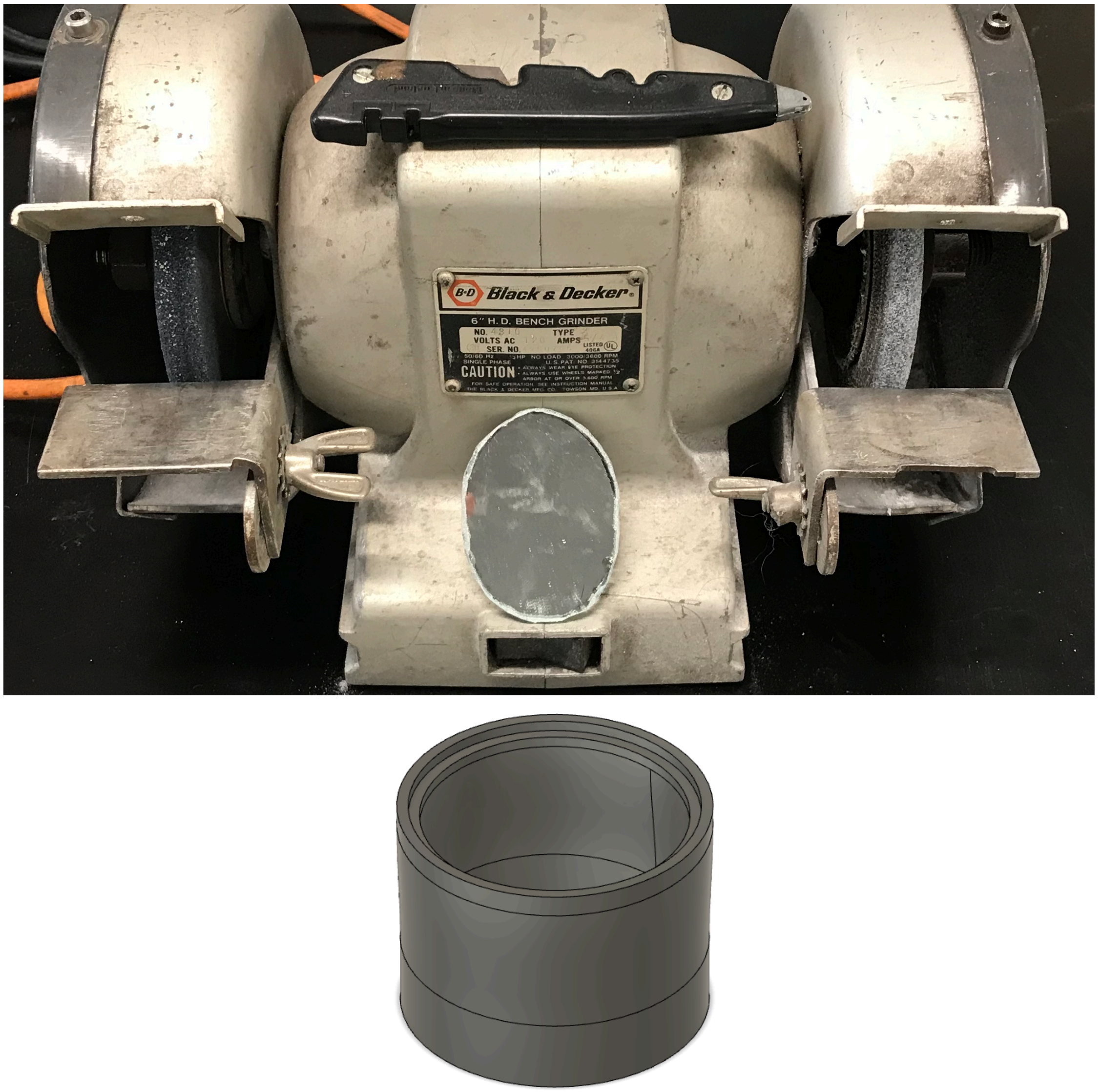


Figure 56: Tools used to shape the mirror (above) and 3D model of the light tube (below). The tip glass cutter helped getting the rough mirror shape and the grinder was used to refine it.

One problem I encountered, however, was the uneven lighting of the plates. To resolve that issue, I used a glass diffuser and collimated light (i.e. having all light rays traveling parallel to each other). Although getting a glass diffuser was fairly straightforward, collimating the LED required some calculations. Using the Thin Lens formula, I derived the following:

$$\frac{1}{p} + \frac{1}{q} = \frac{1}{f}$$

$$\frac{1}{f} = \frac{q}{pq} + \frac{p}{pq}$$

$$\frac{1}{f} = \frac{q+p}{pq}$$

$$f = \frac{pq}{p+q}$$

where f is the focal length of our Fresnel lens,

p is the object distance, and

q is the image distance.

In this case, this yields the following result:

$$f = \frac{153.5 \times \ 25.5}{153.5 + 25.5} = 21.87\ cm = 218.7\ mm$$

By placing the lens at a focal distance from the LED, the light becomes collimated. This results in a more uniform lighting. One additional hardware implementation of this robot is its ability to monitor environmental variables, such as temperature and humidity. To achieve this, I soldered and then connected sensors to the RPI using breadboard (Figure 57).

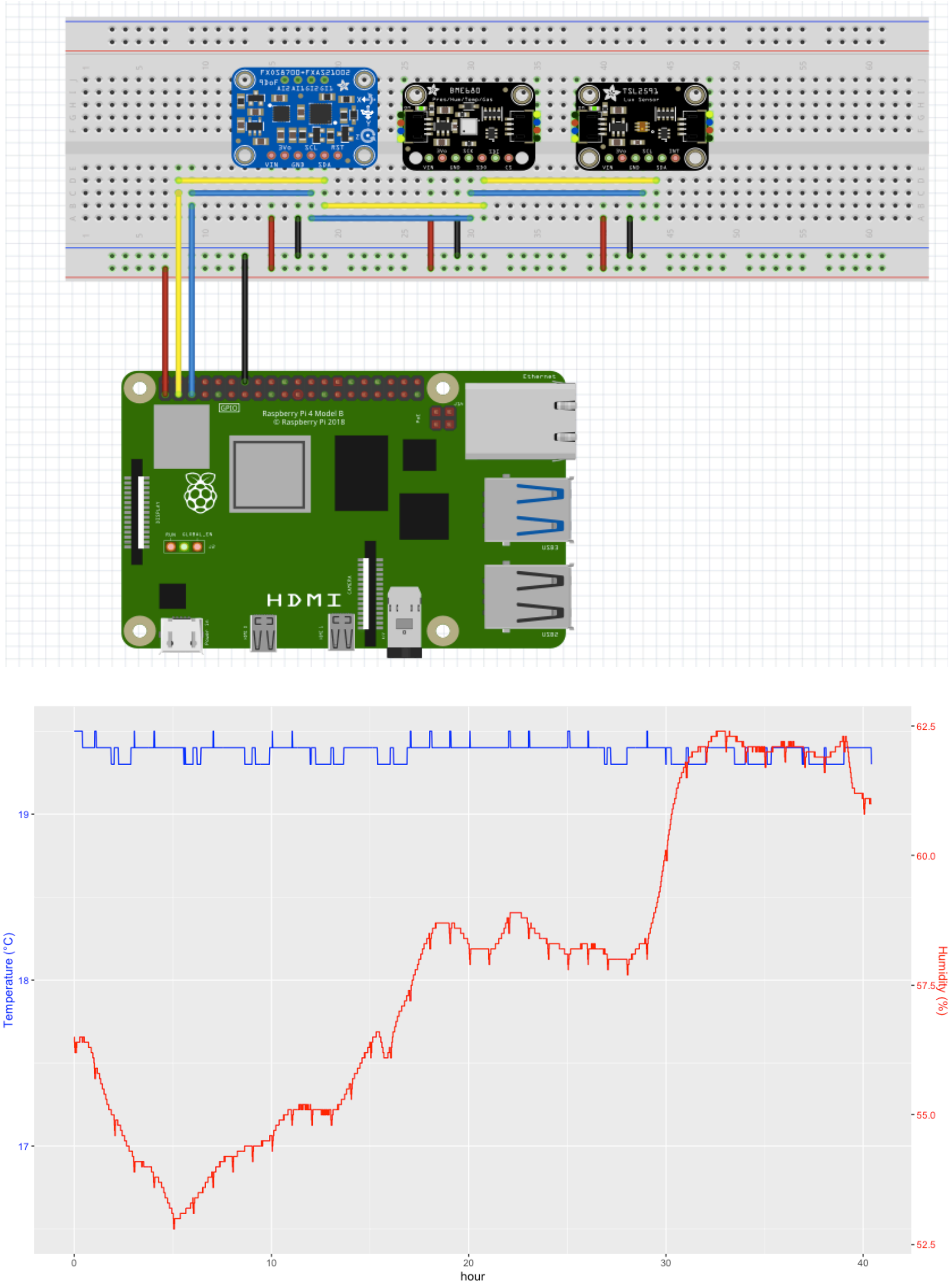


Figure 57: Design of the RPI connected to the sensors through a breadboard made using the electronic design automation software Fritzing (above) and graph of the monitoring of humidity and temperature inside the incubator over the course of 40h (below).

The point of monitoring environmental variables is to make sure that they stay constant throughout the experiments. Finally, the making the medium on which the animals are transferred is at the core of this project. The original protocol for NGM needed to be updated because the new volume was 10 times smaller and became more viscous after autoclaving. I used Eppendorf 24-wells plates (i.e. catalog ID# 0030722019) because they were treated with a proprietary surfactant that reduces meniscus formation (Figure 58). The latter represented a significant issue at the start of the project because it created an uneven focal plane. This in turn yielded animals in focus in the center and others out of focus towards the edge.

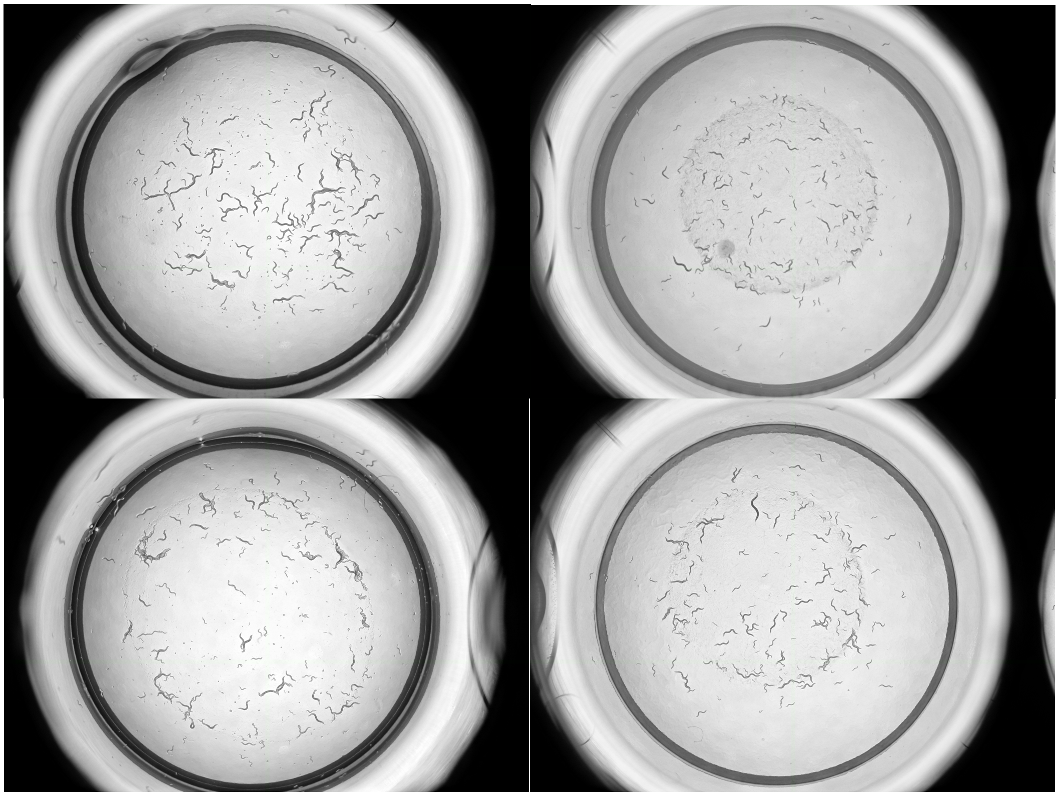


Figure 58: Images taken of two different wells that were untreated for meniscus reduction (left) and treated (right).

My setup to plate the wells uses a 2L beaker serving as a bain-marie at ~70°C for a 250mL Erlenmeyer containing the NGM. To be as sterile as possible, two Bunsen burners are placed on either side of the system and the unopened wells-plate is elevated using a plastic rack. Before using any tools, 70% isopropyl is used to wipe the working area and tools clean and as sterile as possible.

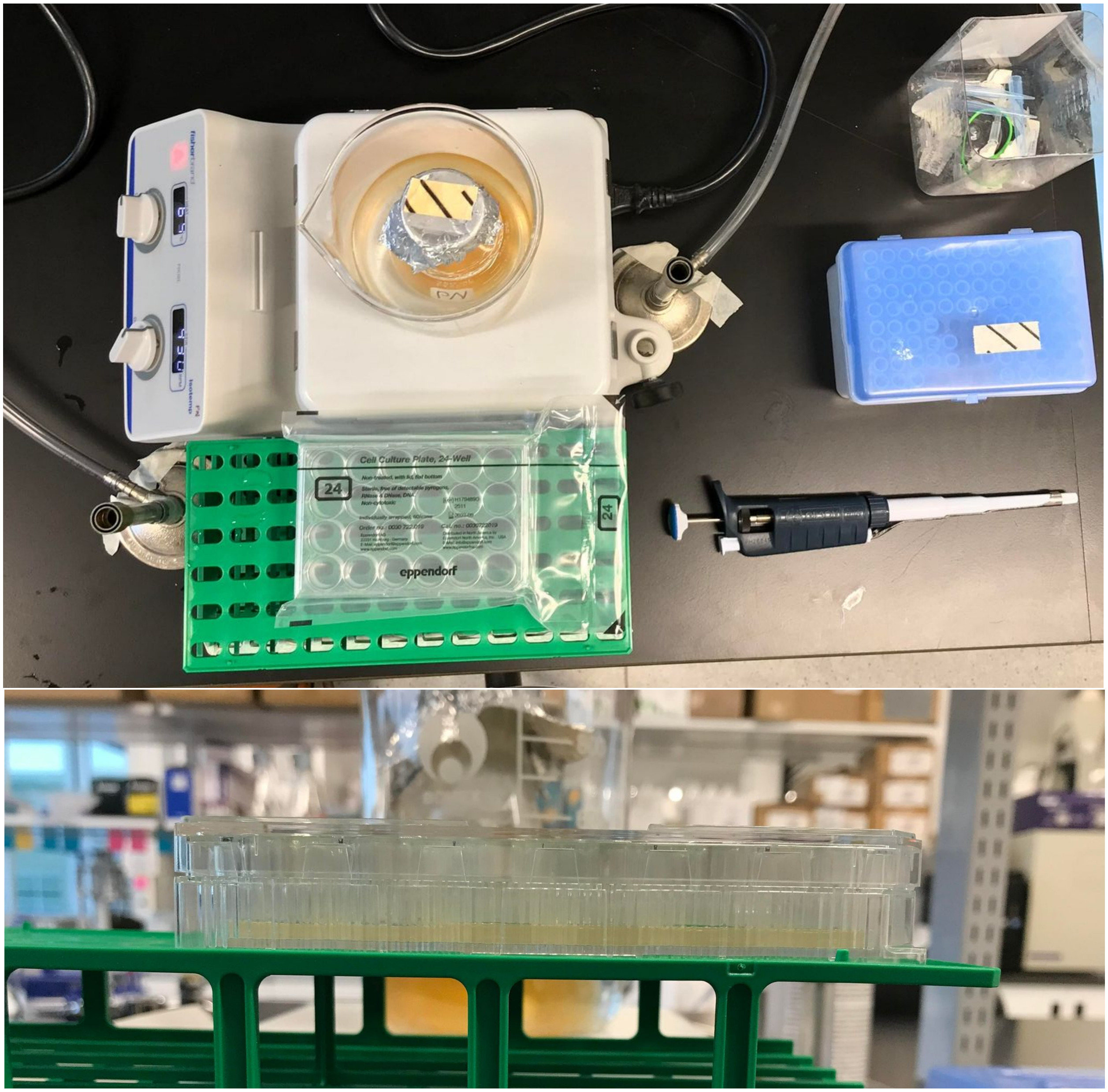


Figure 59: NGM pouring station (above) and properly filled 24-wells plate, treated for meniscus reduction (below).

Since the peristaltic pump I used had a volume error margin of ±250µL for a target of ~550µL, I opted to use a micropipette. This, however, created the issue of cooling down the medium much faster, thereby creating bubbles. To reduce the volumetric flow rate at the end of the tip, I decided the easiest and fastest way was to increase its cross-sectional area by cutting it ~1cm from the tip with flame-sterilized cable cutters. This allowed for a slower pour of medium that was devoid of bubbles. To then seed the medium, I 3D-designed an object (i.e. referred below as the Steady Seeder) that enables the user to seed bacterial lawns consistently over the center of every of the 24 wells (Figure 60).

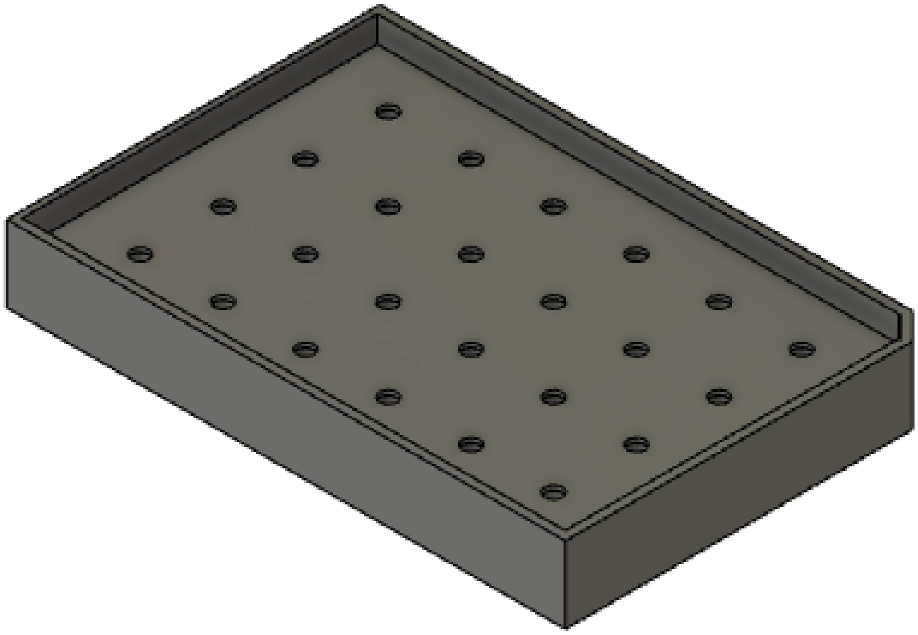


Figure 60: Steady Seeder designed to center the tip of any P200 micropipette over any well from a 24-wells plate treated for meniscus reduction.

The full protocol for preparing the NGM is as follows:

1. Maintain your strains at 15°C on OP50.
2. Prepare 200mL NGM:

   a. In a 250mL Erlenmeyer, add the following:
      i. A magnet
      ii. 0.6g NaCl
      iii. 3.4g agar
      iv. 0.5g peptone
      v. 195mL of sterile water
   b. Autoclave
   c. While the solution gets autoclaved, place 350mL water in a 2L beaker and heat it over a heated magnet mixer.
   d. Once the solution is autoclaved, submerge the Erlenmeyer in the beaker.
   e. Stir it with the magnet and wait until it is cool enough to touch.
   f. STERILE Add the following to the solution:
      i. 0.2mL 1M $CaCl_2$
      ii. 0.2mL 5mg/mL cholesterol in ethanol
      iii. 0.2mL 1M $MgSO_4$
      iv. 5mL 1M KPO_4
   g. Keep warm and agitate constantly with magnet.

3. NGM plating − in a sterile environment (exacto-knife, P1000, tips P1000, 2 bunsen burners, 2L beaker, wire cutter):
   a. Push the air out of a P1000 micropipette set to 600µL.
   b. Append a tip to it.
   c. Use sterile tweezers to cut the end of the tip so that the remaining diameter is ~3mm.

    d. Pipette just enough NGM to fill the bottom of each well by capillarity.

4. Let rest for at least 36h for the wells to dehydrate and to notice any contamination.
5. Bacteria seeding – in a sterile environment:
    a. Resuspend bacteria by agitation.
    b. Push the air out of a P200 micropipette set to 25µL.
    c. Append a tip to it.
    d. Use sterile tweezers to cut the end of the tip at the most distal ring, ~11mm away from the distal extremity.
    e. Using the Steady Seeder, gently pipette the 25µL of bacteria onto the center of each well.
    f. Wait 5mins without removing the Steady Seeder to let every bacterial mat start to dehydrate uniformly.
    g. Carefully remove the Steady Seeder and cover the plate, making sure not to shake the bacterial mats.
6. Let rest for at least 36h for the bacteria to grow sufficiently and the mat to dehydrate.
7. Transfer one animal per well.
8. Start the CESAR_master.py script from the main RPI.

The pouring and seeding both require 36hrs drying times. During the first hour of each drying process, I noticed that there was significant evaporation. Closing the lid at that time hindered the evaporation and lengthened the waiting time. I, therefore, designed and 3D-printed a plate holder which can be placed between two Bunsen burners to make that initial drying period more efficient (Figure 61).

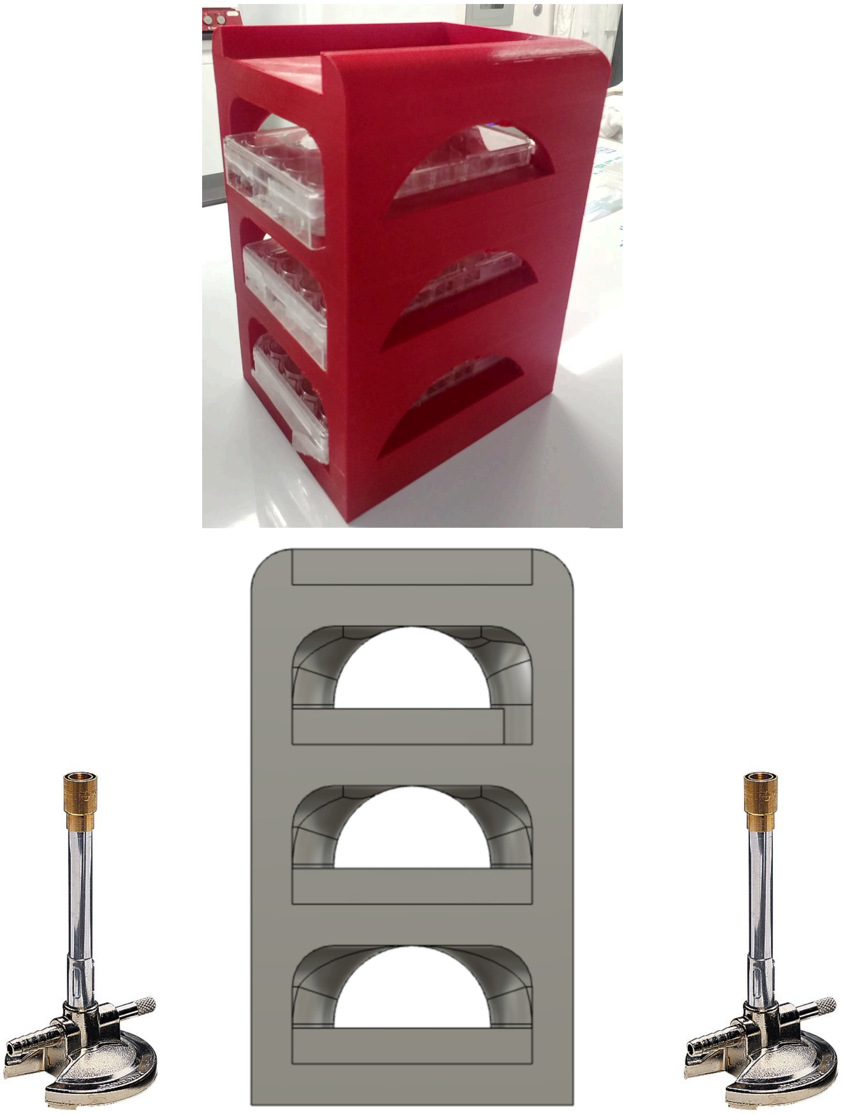


Figure 61: Drying station for both the NGM and the OP50 bacterial lawns.

Even though this hardware is sound and workable, it would be essentially useless without proper control via software, which we will now cover in more depth.

4.2 Software

4.2.1 Overview

The CeSAR operates its data analyses sequentially. It momentarily surveils every well, in order, every hour for ~10 days. It first takes a picture of the entire well before analyzing it using a trained neural network. The network was trained on 15,000 training images using the YOLOv5 architecture by Ultralytics (i.e. https://github.com/ultralytics/yolov5). When called, it can infer the position of the animals and eggs in the image (Figure 62A). Then, it uses the parent's position to move the stage by a specific Δ(X, Y) to align the microscopic lens with the worm. It then films it for ~20s and characterizes it phenotypically (Figure 62B). The phenotypical traits yielded by this method are the worm's length, change in width along its body, curvature, pumping rate, movement speed, defecation rate, and general path on the medium[7]. This entire workflow is directed by a Python script which controls movement, image processing via the OpenCV library, and calling of the neural network to analyze the images taken. Communication with the motors is done using GRBL software, which allows to transform real coordinates into G-code for both the α and β motors in coreXY configuration.

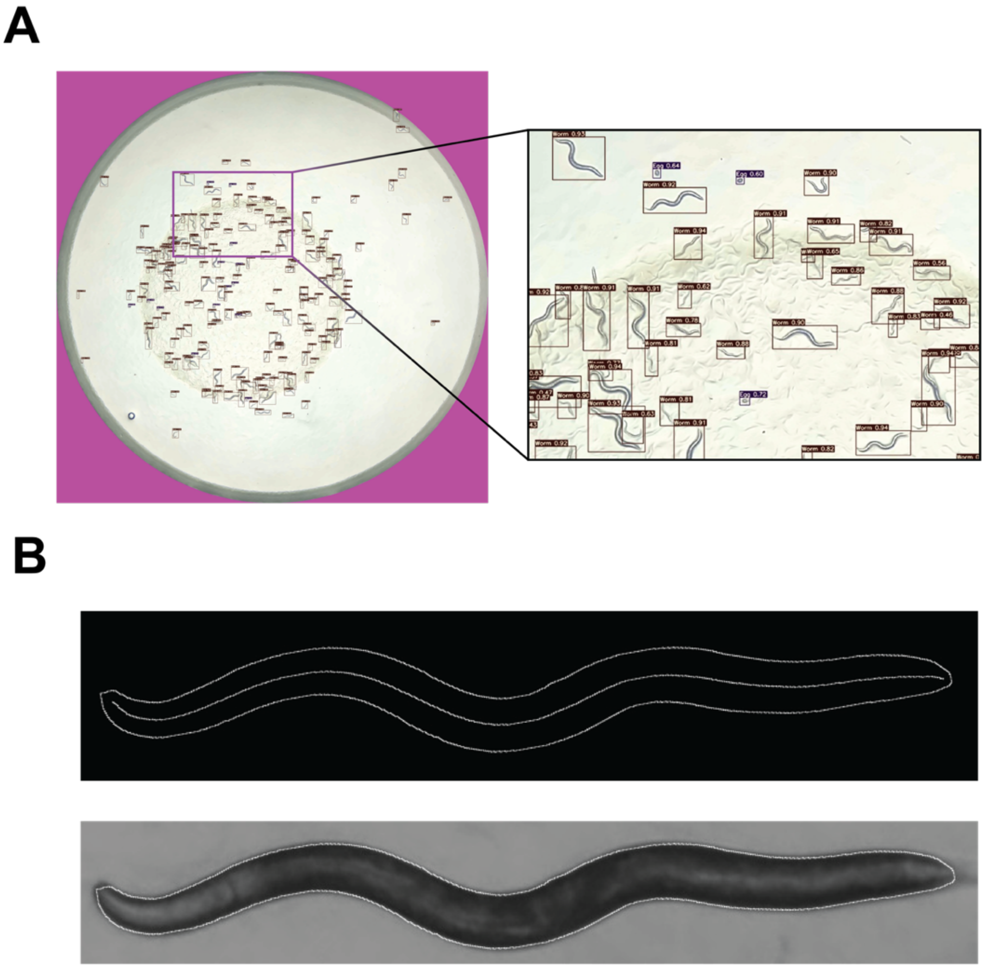


Figure 62 Images analyses. *(A) Neural network inferences of the positions of worms and eggs in a well. (B) Outline and midline determination through image processing of the adult parent P_0.*

While the process is ongoing, the robot keeps careful watch on the environment itself by measuring, every 5s, the temperature, humidity, atmospheric pressure, light intensity, noise level, and magnetic field (Table 2).

Sensor	Monitored variable
SPH0645LM4H	Microphone (audio sensor)
FXOS8700 + FXAS21002	Acceleration and magnetic sensor
TSL2561	Light intensity sensor
BME280	Temperature, humidity, atmospheric pressure, and air quality sensor

Table 2: Each sensor and its targeted measurement

To ensure constant acquisition parameters, CeSAR starts by autofocusing its macroscopic lens, by computing a Laplacian filter on the image of the current well (Figure 63A). Then, it repositions the well to be perfectly aligned with the lens (Figure 63B).

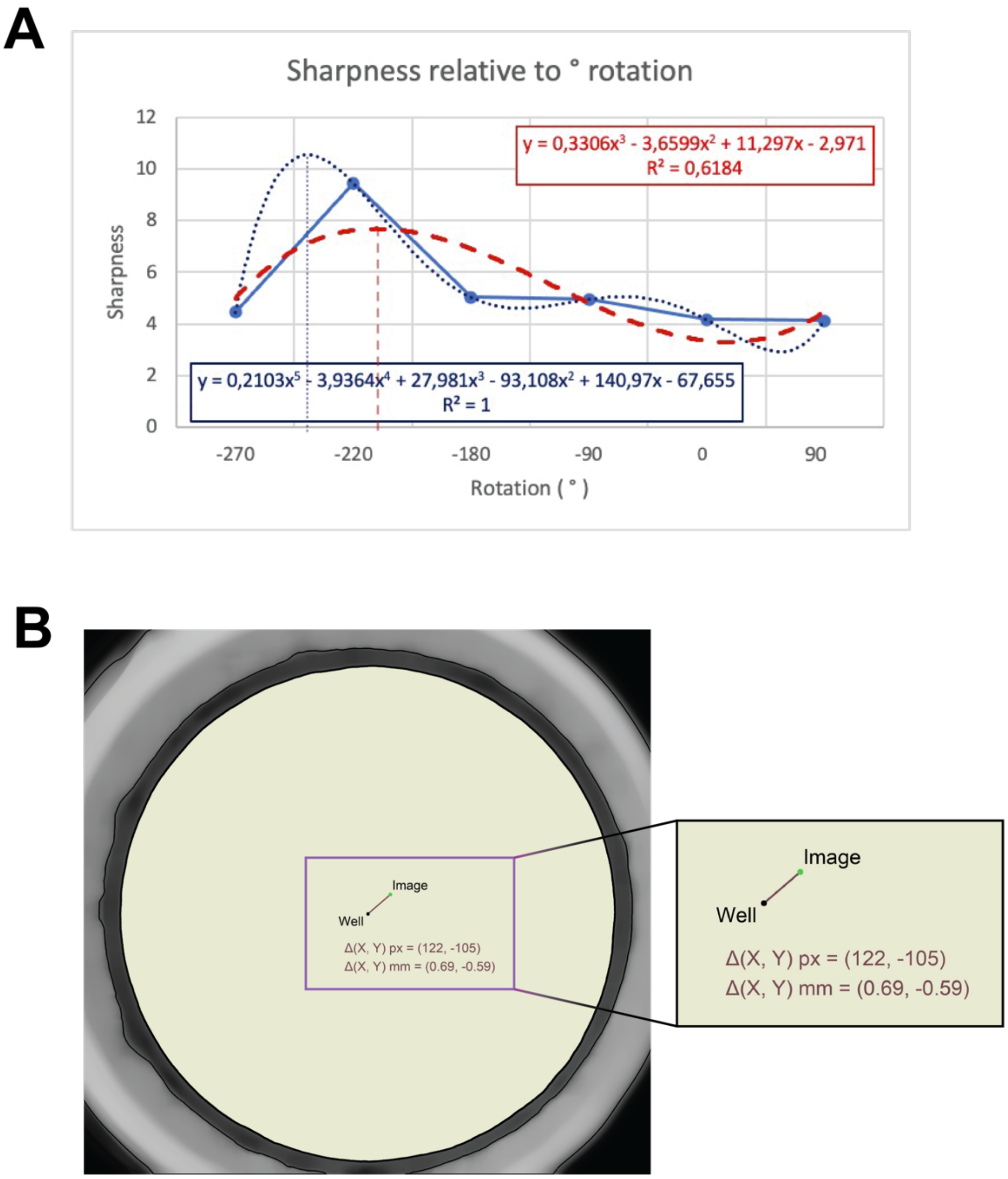


Figure 63: Automatic recalibration of focus and centering of images. *(A) Graph of the Laplacian sharpness value relative to the degree of rotation of the lens. The degree recorded is the one for which sharpness is at its peak. (B) Image processing showing the Δ(X, Y) offset of the well from the center of the image.*

4.2.2 R&D process

Before doing any training on the images to determine the location of eggs and worms, I had to make sure that they were processed correctly to have the best quality possible. I first noticed that the walls of the wells served as a reflective surface, which meant that the animal as well as its reflection would be captured in the photograph and analyzed. Since this would have duplicated some objects, I solved this issue by approximating the edges of the well using image processing (Figure 64).

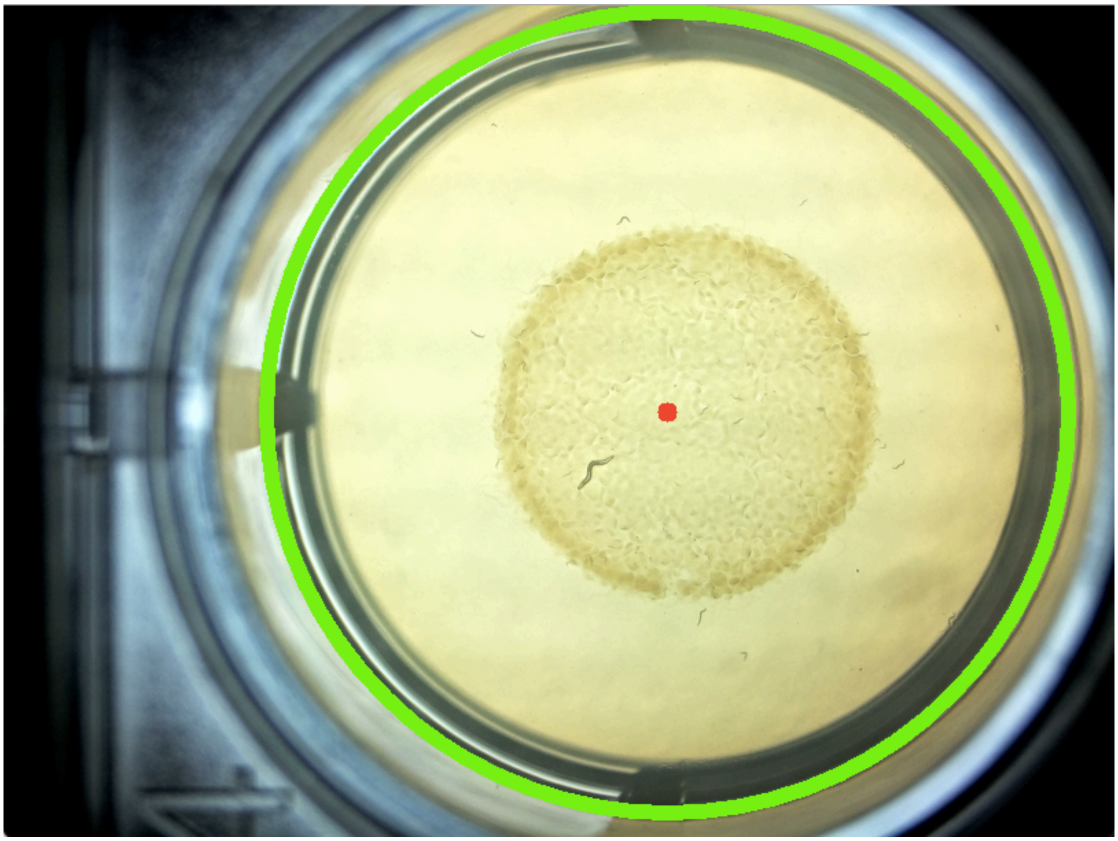


Figure 64: Approximation of a well's edges and center using the Python OpenCV library.

This allowed us to go from a 3040px × 4056px image to a ~2700px × 2700px one (Figure 65), which decreased the size of the image files, thereby making the image processing faster. This removed the reflective edge of the well, which was a source of counting error.

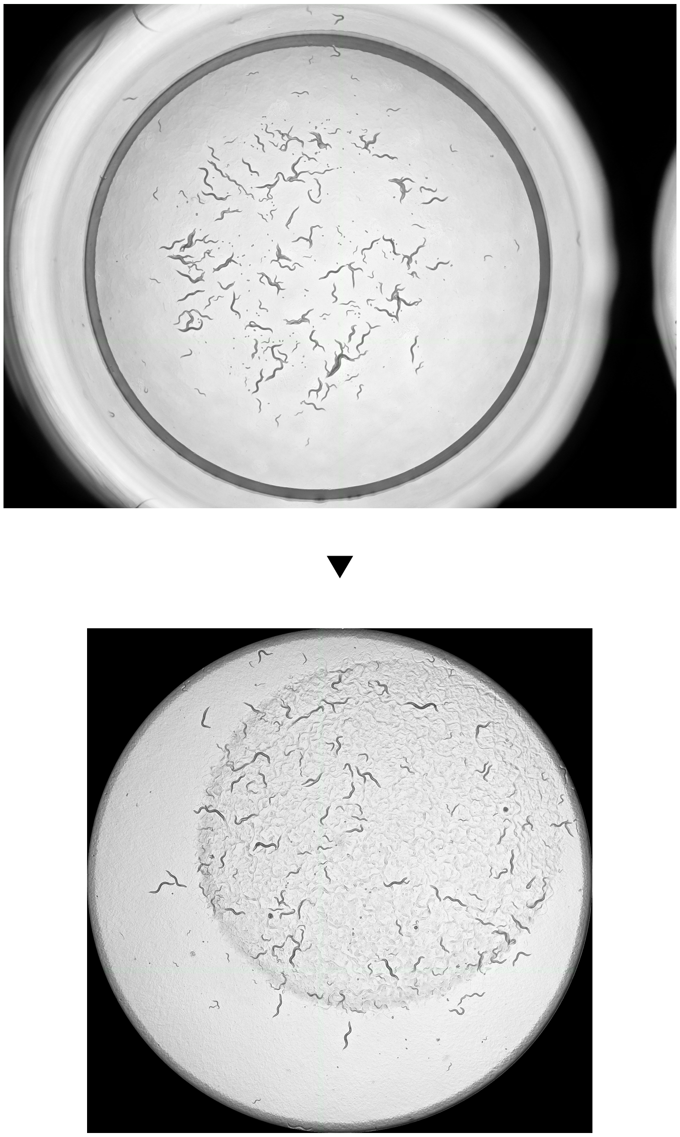
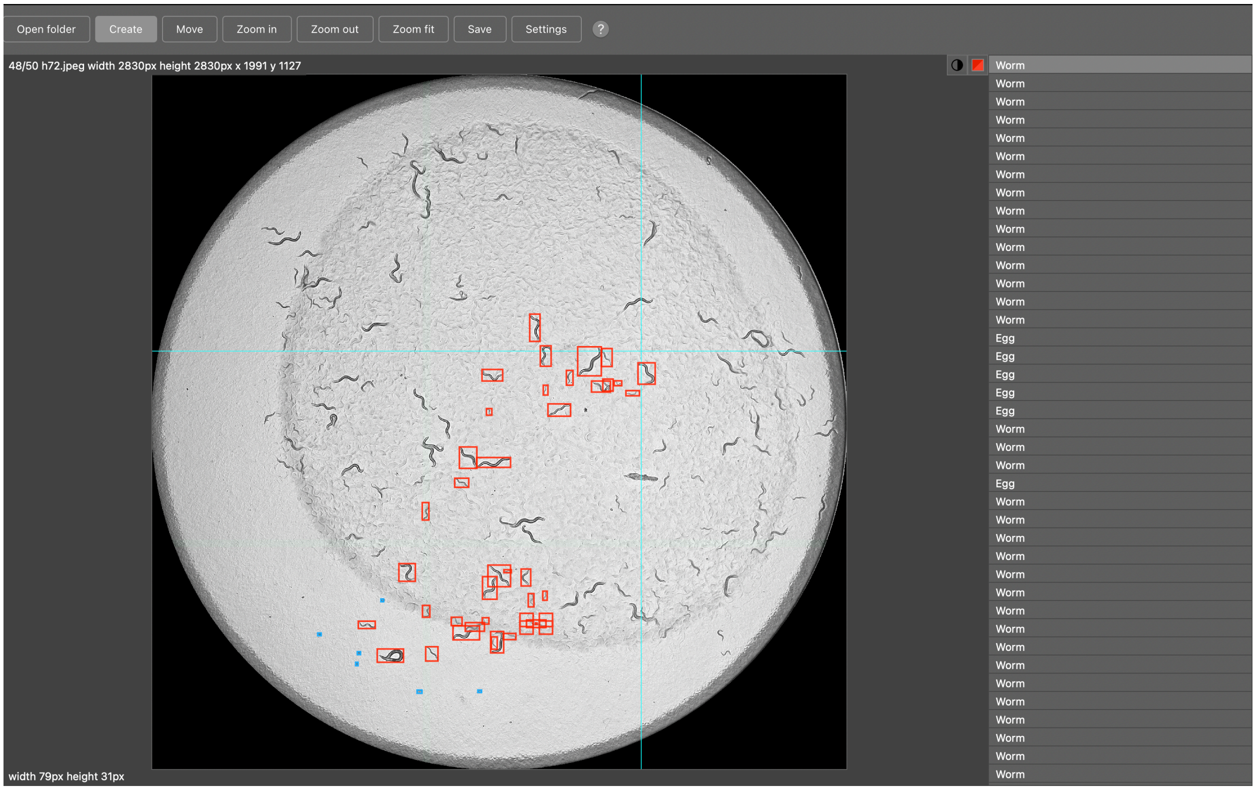


Figure 65: Image processing resulting in only the medium, without the reflective edges.

Having properly processed images, I was able to work on training a model to recognize eggs and worms in the acquired images. To have artificial intelligence requires some sort of encoding of knowledge, whether that be through code that explicitly tells the computer what to do, or by training that computer on a specific problem set. The latter is exactly what I did by identifying up to 300 objects on more than 100 images. I used the free version of RectLabel (https://rectlabel.com/) to do so, drawing ~5000 bounding boxes manually for the training dataset (i.e. ~3000 boxes for worms and ~2000 for eggs). This was done by drawing, for each object, a box closely bounded around it. For instance, drawing a bounding-box around an Egg or a Worm means selecting 4 coordinates which, if moved closer to each other, would exclude pixels of the object from the resulting rectangle.

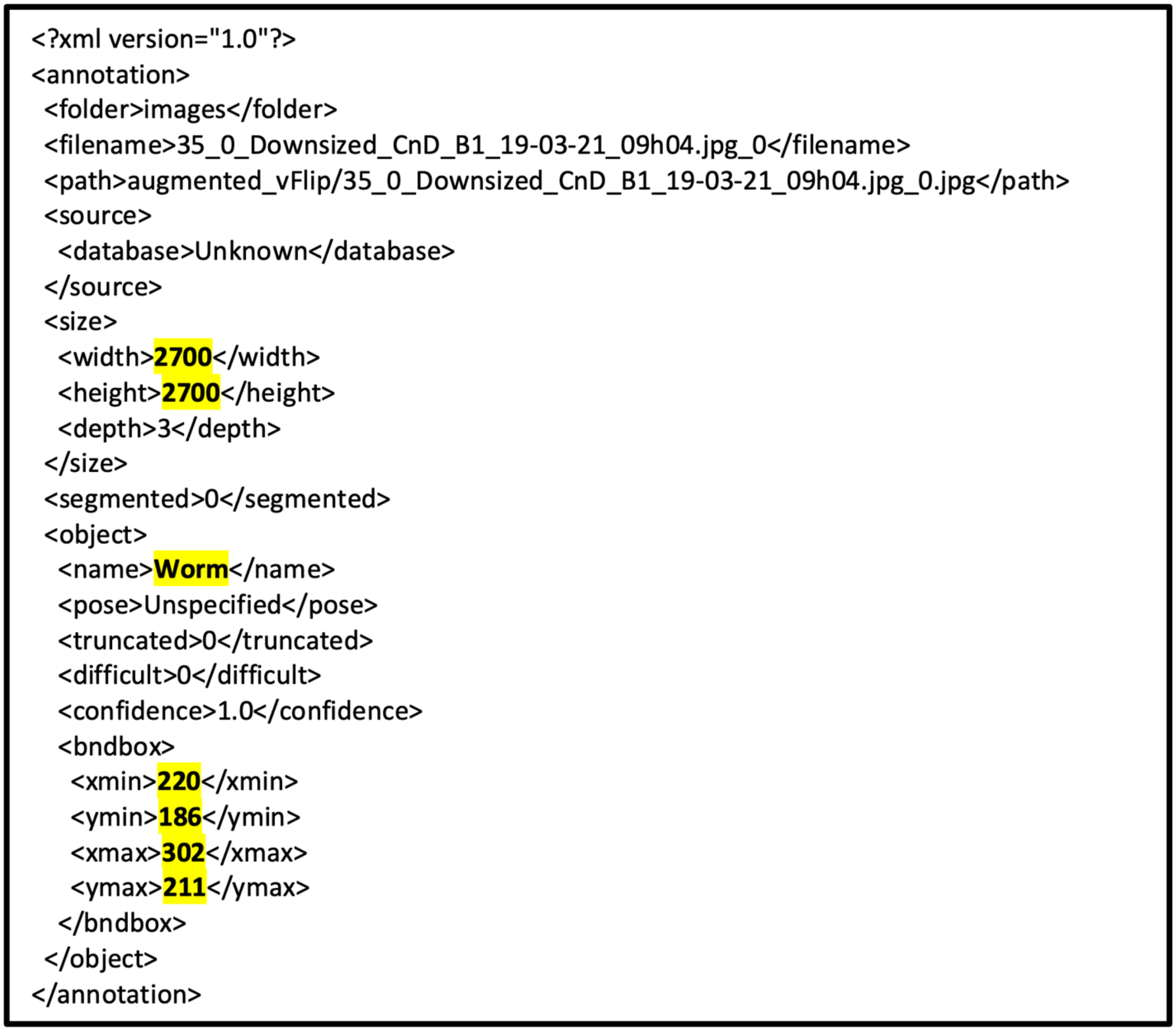


48/50 h72.jpeg width 2830px height 2830px x 1991 y 1127

width 79px height 31px

Figure 66: Graphical user interface (GUI) of the RectLabel desktop application. *In the middle is the image the user is working on. On the right are all the objects the user has already identified.*

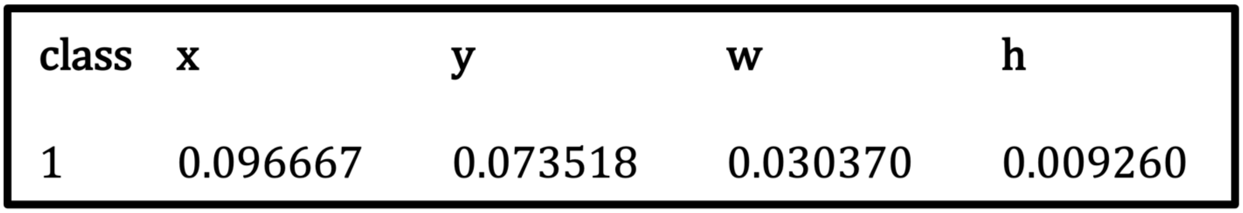
This training results in an extensible markup language (.xml) file that holds the information of every bounding box (i.e. class, minimum x and y values, and maximum x and y values).



```xml
<?xml version="1.0"?>
<annotation>
  <folder>images</folder>
  <filename>35_0_Downsized_CnD_B1_19-03-21_09h04.jpg_0</filename>
  <path>augmented_vFlip/35_0_Downsized_CnD_B1_19-03-21_09h04.jpg_0.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>2700</width>
    <height>2700</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Worm</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <confidence>1.0</confidence>
    <bndbox>
      <xmin>220</xmin>
      <ymin>186</ymin>
      <xmax>302</xmax>
      <ymax>211</ymax>
    </bndbox>
  </object>
</annotation>
```

Figure 67: .xml file produced after using the RectLabel application on an image. *Highlighted in yellow, from top to bottom, are the width and height of the image, the class of an object, and the minimum and maximum x and y values (i.e. corners) of the bounding box.*

This format, however, is unintelligible by the You Only Look Once (YOLO)v5 training model that we used, as it requires a text (.txt) file where each line is an identified object holding the class, x, y, width, and height values of the bounding box in that order (Figure 68). The coordinates and size of the bounding box in this .txt format are relative to the size of the image (e.g. 2700 × 2700 in the example above).



class	x	y	w	h
1	0.096667	0.073518	0.030370	0.009260

Figure 68: .txt formatted file example with one detected object. *This is the format compatible with YOLOv5 (i.e. used as input training data as well as output).*

In Figure 68, the class of the object is either 0 for Egg or 1 for Worm,

x is the x-coordinate of the center point of the bounding box,

y is the y-coordinate of the center point of the bounding box,

w is the width of the bounding box relative to the width of the image, and

h is the height of the bounding box relative to the height of the image.

The model was then fed smaller sized chunks of the original image with its corresponding bounding boxes .txt files. Those smaller images were 640px × 640px as recommended by the training model. I wrote a script, siwo.py, that divided the original image into smaller images using a step interval with overlap (Figure 69). The entire collection of those resulting images represents the dataset on which the model was trained. I used a remote server to perform the training using the paid service of a Google notebook. This gave me access to a more powerful computer than my own, with a better CPU and GPU.

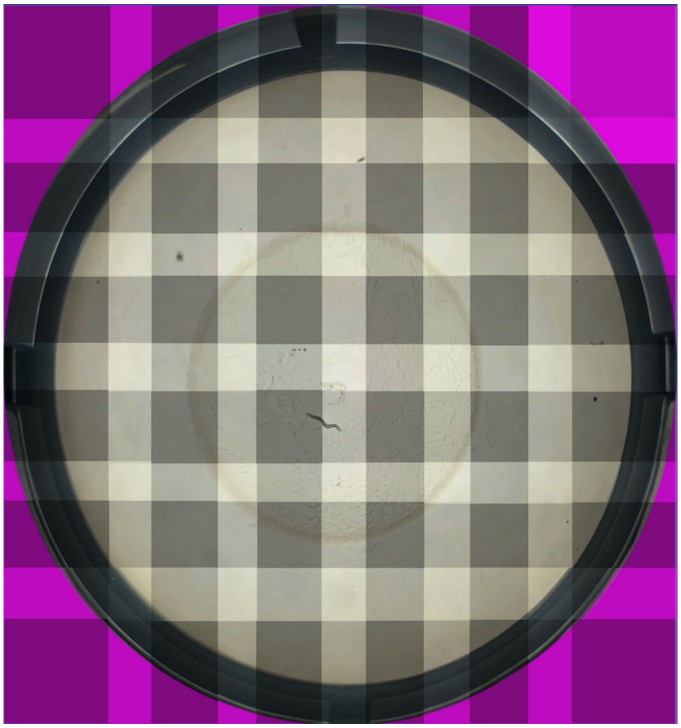


Figure 69: Subdivision of an original image into 640px × 640px images to train the neural network.

To increase the robustness of the neural network after training, I used data augmentation on the training dataset. This artificially increases the size of the training dataset by applying a vertical mirroring transformation to it (Figure 70). This was done using the PlantCV Python library.

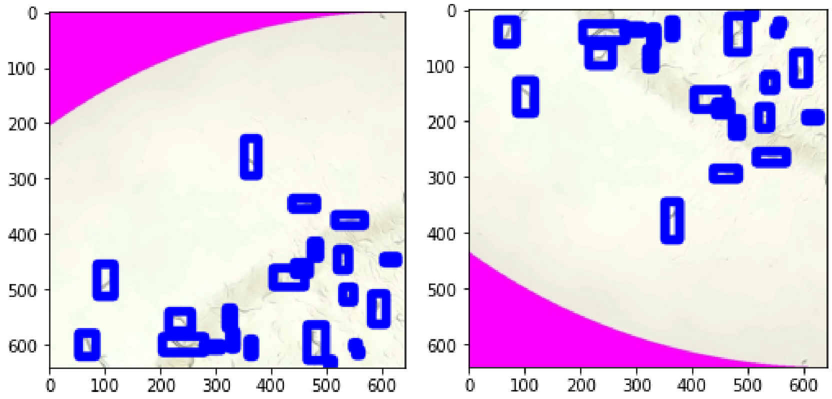


Figure 70: Vertical flip transformation used as data augmentation to increase training data size.

When training a neural network, a key question is whether the object proposition is correct or not (i.e. a true or false positive respectively). To answer this question, a mathematical approximation is used: the intersection over union (IoU, Figure 71). It represents the area of overlap between the ground truth (i.e. what we know to be positive in our training data) and the prediction, divided by their union. For instance, if the IoU is above 0.5, the inference is considered a hit. Otherwise, it is considered a miss. This makes intuitive sense as the bigger the overlap relative to the union, the more area is shared between the prediction and ground truth. It therefore follows that the IoU is proportional to the quality of the inferences: the better the inference, the higher the IoU.

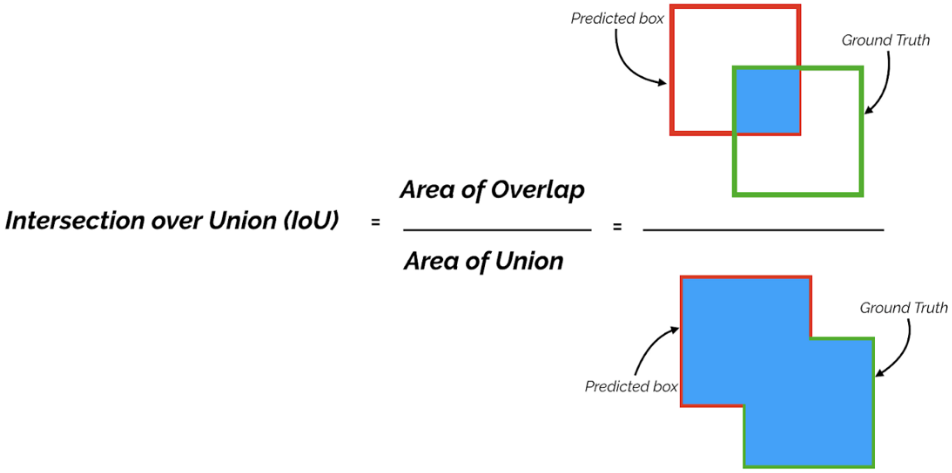


Figure 71: Graphical representation of the intersection over union (IoU) in the context of object detection training *(source: https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2).*

Having false positive (FP) or true positive (TP) for each class then allows for the calculation of average precision for that class. It is calculated as follows:

$$AP_{class} = \frac{\#TP_{class}}{\#TP_{class} + \#FP_{class}}$$

Where AP_{class} is the average precision for class,

$\#TP_{class}$ is the number of true positives for class, and

$\#FP_{class}$ is the number of false positives for class.

The common metric for estimating the strength of a neural network can then be calculated as the mean average precision (mAP). It essentially is the average of the AP_{class} for every class the neural network has been trained on. Using this training, I designed the process through which the AI would be used to identify the animals. I started by programming the most efficient route (Figure 72) that satisfied two rules: be closest to the camera at home position and have the smallest distance between the current well and the next one.

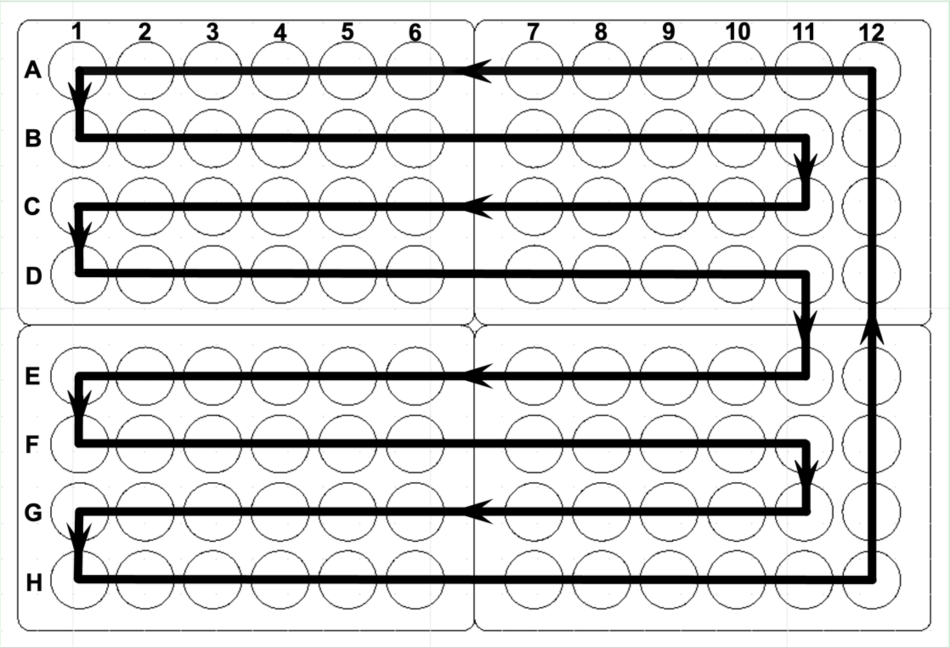
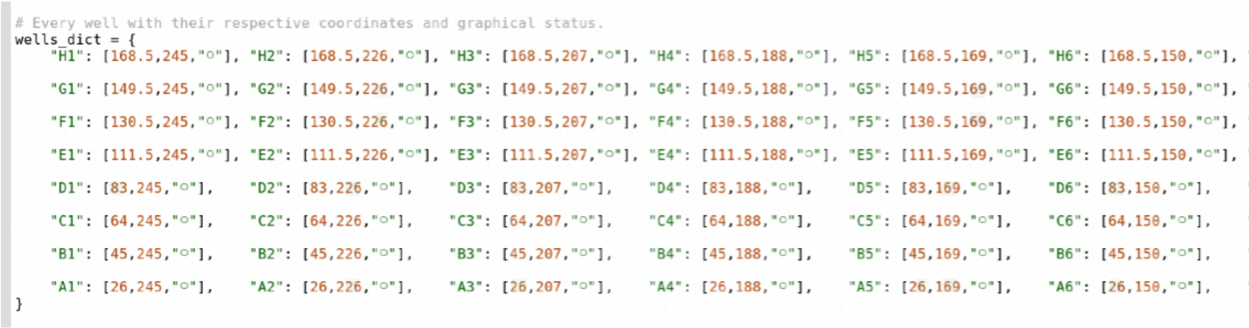


Figure 72: Designed path for the robot to take to be as efficient as possible. Although user-programmable, by default, it starts at the A12 well.

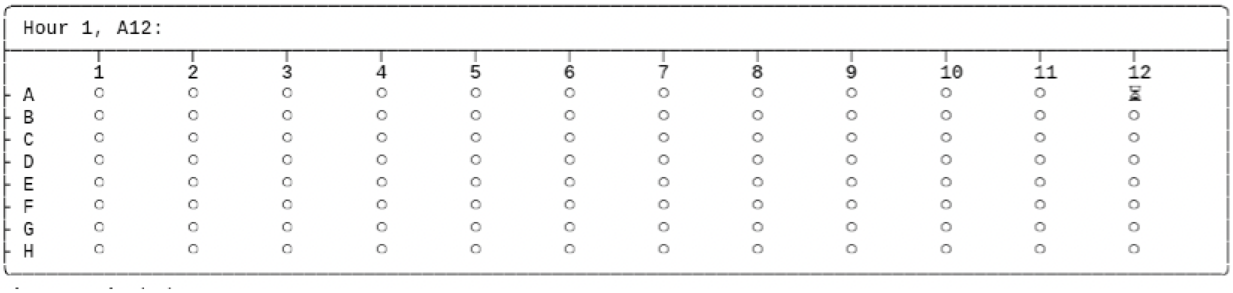
The coordinate values from origin were estimated using the plates' specification sheet. In practice, these values had to be adjusted by trial and error to account for a small system skewness. Every well position was then hard-coded in a dictionary variable to be used with the well name (e.g. "E7") as reference (Figure 73).



```
# Every well with their respective coordinates and graphical status.
wells_dict = {
    "H1": [168.5,245,"○"], "H2": [168.5,226,"○"], "H3": [168.5,207,"○"], "H4": [168.5,188,"○"], "H5": [168.5,169,"○"], "H6": [168.5,150,"○"], '

    "G1": [149.5,245,"○"], "G2": [149.5,226,"○"], "G3": [149.5,207,"○"], "G4": [149.5,188,"○"], "G5": [149.5,169,"○"], "G6": [149.5,150,"○"], '

    "F1": [130.5,245,"○"], "F2": [130.5,226,"○"], "F3": [130.5,207,"○"], "F4": [130.5,188,"○"], "F5": [130.5,169,"○"], "F6": [130.5,150,"○"], '

    "E1": [111.5,245,"○"], "E2": [111.5,226,"○"], "E3": [111.5,207,"○"], "E4": [111.5,188,"○"], "E5": [111.5,169,"○"], "E6": [111.5,150,"○"], '

    "D1": [83,245,"○"],    "D2": [83,226,"○"],    "D3": [83,207,"○"],    "D4": [83,188,"○"],    "D5": [83,169,"○"],    "D6": [83,150,"○"],    '

    "C1": [64,245,"○"],    "C2": [64,226,"○"],    "C3": [64,207,"○"],    "C4": [64,188,"○"],    "C5": [64,169,"○"],    "C6": [64,150,"○"],    '

    "B1": [45,245,"○"],    "B2": [45,226,"○"],    "B3": [45,207,"○"],    "B4": [45,188,"○"],    "B5": [45,169,"○"],    "B6": [45,150,"○"],    '

    "A1": [26,245,"○"],    "A2": [26,226,"○"],    "A3": [26,207,"○"],    "A4": [26,188,"○"],    "A5": [26,169,"○"],    "A6": [26,150,"○"],    '
}
```

Figure 73: Dictionary variable named wells_dict used to associate each well with its X and Y coordinates as well as a graphical American Standard Code for Information Interchange (ASCII) marker showing its current status. *In this case, every well is initialized to be unevaluated.*

I then created a graphical representation of the progress of wells surveillance, which shows surveilled, being surveilled, and to be surveilled wells. This representation is an updated string variable made of ASCII characters that appears in the console as the main script runs (Figure 74).



```
Hour 1, A12:

     1    2    3    4    5    6    7    8    9    10   11   12
A    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ⧗
B    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
C    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
D    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
E    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
F    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
G    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
H    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○    ○
```

Figure 74: Graphical representation of the current progress on wells surveillance. *A circle signifies that the well is unevaluated. An hourglass represents which well is currently being evaluated, as is A12 in this case. A colored disk signifies that the well has been evaluated (none it this case).*

I first tried to have one parent, P_0, per well. However, that caused a high population density (Figure 75) that was difficult to reliably count by the AI and could potentially be high enough to produce significant daumone concentrations. This dauer-inducing pheromone lengthens the egg-laying process and induces dauer formation[57].

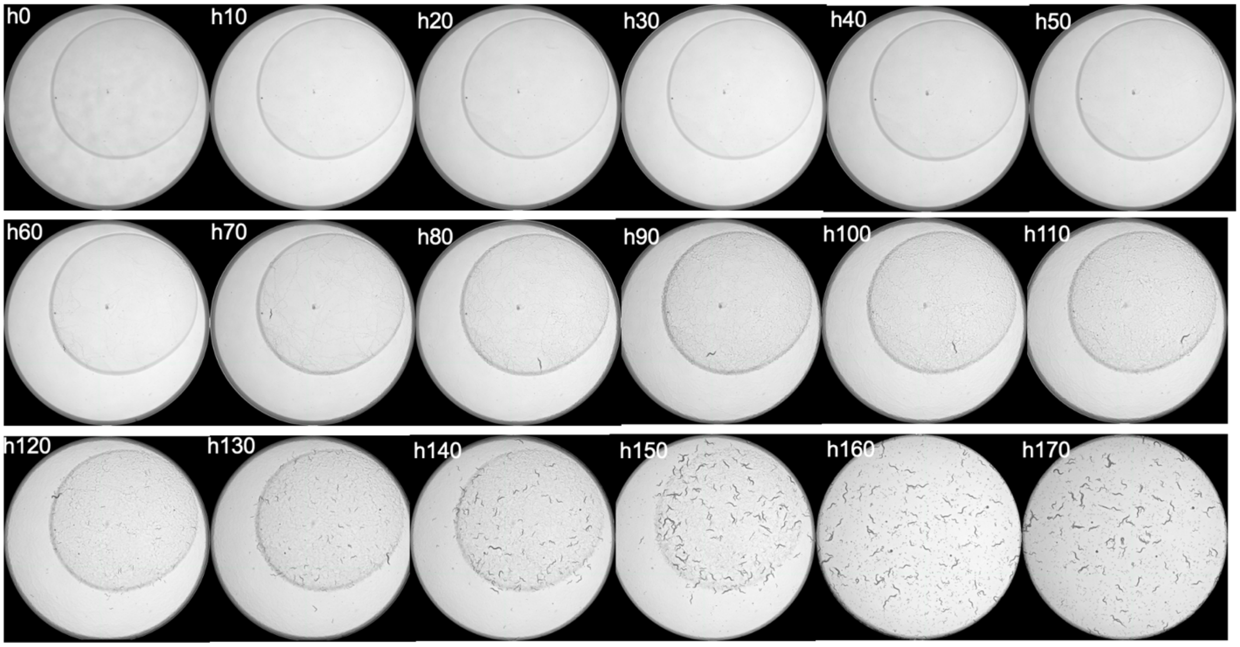
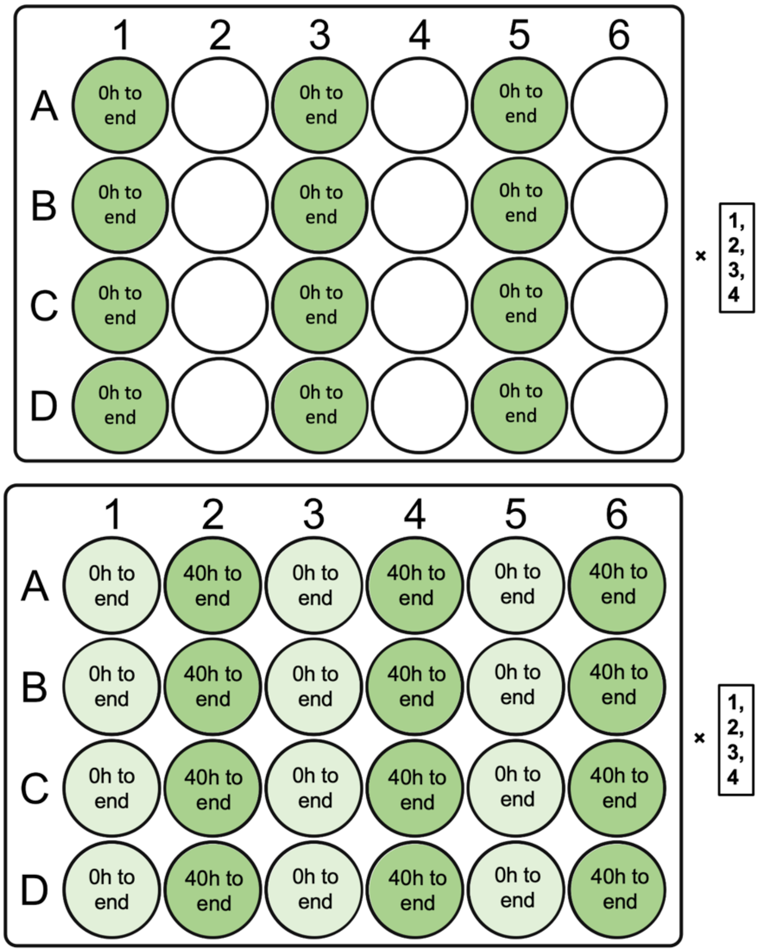


Figure 75: 7 day timelapse of C. elegans from one P_0 egg to its full first filial (F_1) and start of second filial (F_2). *At hour 50, roughly 2 days after the egg-laying stage, the well gets crowded and food is decreasing rapidly.*

Since having a high population density could represent a future lack of food, this response enables the species to delay development until more favorable environmental conditions. Having only one well per P_0 also caused a premature lack of food once the progeny, F_1, were mostly developed to the fourth and last larval stage, L4. This is why I decided to divide the sampling power of the robot by two: to increase data quality. This, however, is a drawback as it means that the experimenter has to transfer the parent animal from one well to another. This implies two human manipulations: the initial and mid-way transfers of P_0.



	1	2	3	4	5	6
A	0h to end		0h to end		0h to end	
B	0h to end		0h to end		0h to end	
C	0h to end		0h to end		0h to end	
D	0h to end		0h to end		0h to end	

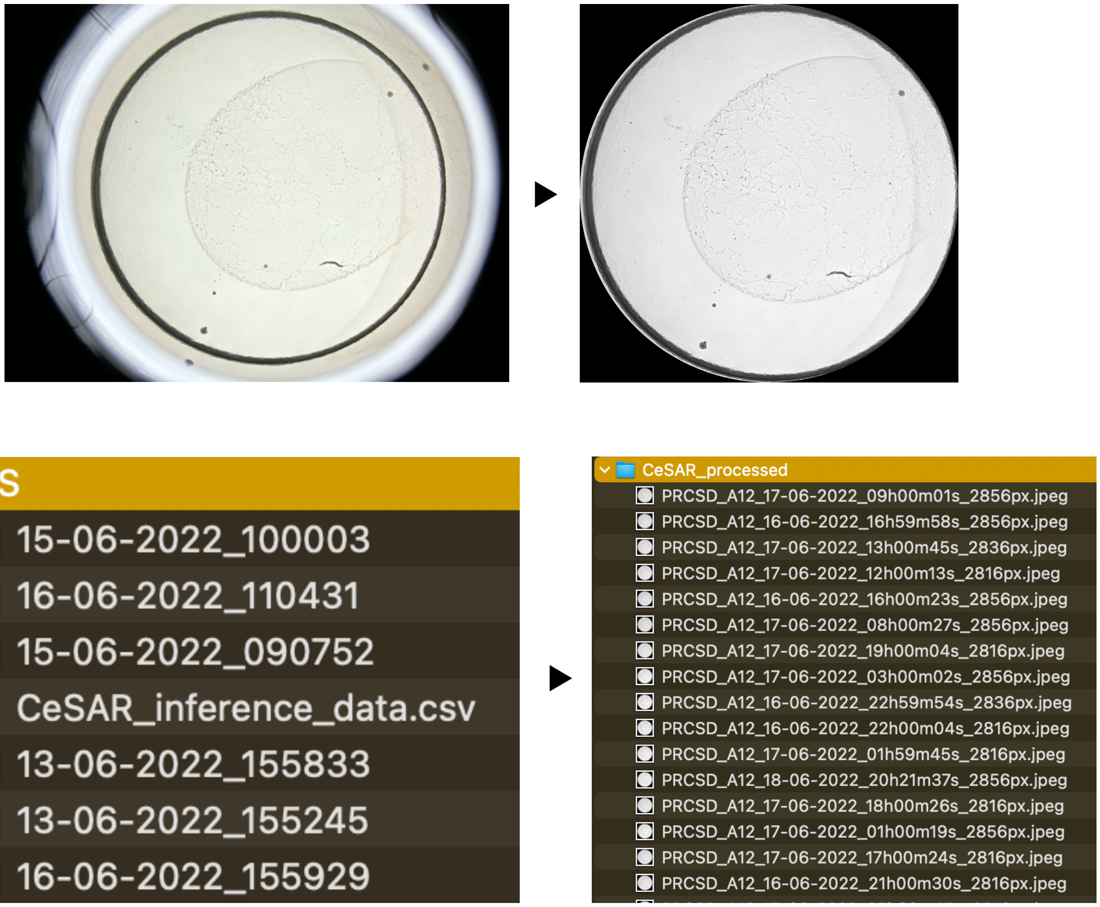
	1	2	3	4	5	6
A	0h to end	40h to end	0h to end	40h to end	0h to end	40h to end
B	0h to end	40h to end	0h to end	40h to end	0h to end	40h to end
C	0h to end	40h to end	0h to end	40h to end	0h to end	40h to end
D	0h to end	40h to end	0h to end	40h to end	0h to end	40h to end

Figure 76: Diagram of the data acquisition per well regarding the first part of the experiment (above) and the second (below).

This requires transferring the adult P_0 from the X(n) to the X(n+1) well, which should be done gently and as efficiently as possible, so as not to hurt the animals or waste time.

Each of the initial 40h and 40h to end surveillances is launched by a separated Python script. The first script, in addition to taking hourly photographs of each well, also records a 1FPS film of each worm at the very beginning for two minutes. This yields 120 images that are analyzed to determine the mobility and path the animal took during this time. This script along with the second one is saved on the Raspberry pi computer. The images they produce, however, are stored on a separate 1 terabyte hard-disk drive (HDD). The images, totaling ~50GB per experiment, must then be treated in a similar fashion as the training data before being inferred upon. From original image file to fully processed, they go through 3 distinct Python scripts.

The first one processes all images:

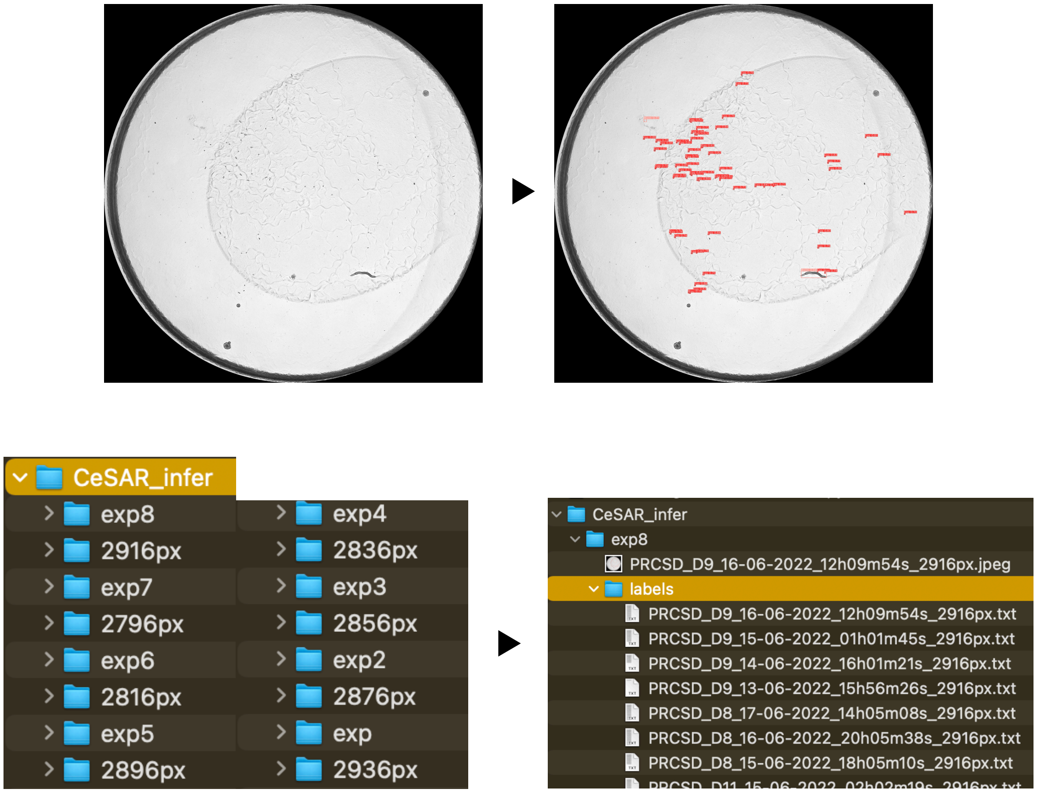


BS
- 15-06-2022_100003
- 16-06-2022_110431
- 15-06-2022_090752
- CeSAR_inference_data.csv
- 13-06-2022_155833
- 13-06-2022_155245
- 16-06-2022_155929

CeSAR_processed
- PRCSD_A12_17-06-2022_09h00m01s_2856px.jpeg
- PRCSD_A12_16-06-2022_16h59m58s_2856px.jpeg
- PRCSD_A12_17-06-2022_13h00m45s_2836px.jpeg
- PRCSD_A12_17-06-2022_12h00m13s_2816px.jpeg
- PRCSD_A12_16-06-2022_16h00m23s_2856px.jpeg
- PRCSD_A12_17-06-2022_08h00m27s_2856px.jpeg
- PRCSD_A12_17-06-2022_19h00m04s_2816px.jpeg
- PRCSD_A12_17-06-2022_03h00m02s_2856px.jpeg
- PRCSD_A12_16-06-2022_22h59m54s_2836px.jpeg
- PRCSD_A12_16-06-2022_22h00m04s_2816px.jpeg
- PRCSD_A12_17-06-2022_01h59m45s_2816px.jpeg
- PRCSD_A12_18-06-2022_20h21m37s_2856px.jpeg
- PRCSD_A12_17-06-2022_18h00m26s_2816px.jpeg
- PRCSD_A12_17-06-2022_01h00m19s_2856px.jpeg
- PRCSD_A12_17-06-2022_17h00m24s_2816px.jpeg
- PRCSD_A12_16-06-2022_21h00m30s_2816px.jpeg

Figure 77: First processing of the result images (i.e. preparing for inference). *Every image is treated to be in grayscale, approximated to a disk and contrast enhanced (above). The files are named such that they contain all relevant information for the next processing (below): the well name, the date of acquisition, and the size of the image.*

The resulting ~16,000 images (i.e. ~6,000 for the 48 initial movies of the individual parents and ~10,000 for their brood size analysis) of one experiment are then ready to be inferred on.



```
CeSAR_infer
  exp8        exp4
  2916px      2836px
  exp7        exp3
  2796px      2856px
  exp6        exp2
  2816px      2876px
  exp5        exp
  2896px      2936px
```

```
CeSAR_infer
  exp8
    PRCSD_D9_16-06-2022_12h09m54s_2916px.jpeg
    labels
      PRCSD_D9_16-06-2022_12h09m54s_2916px.txt
      PRCSD_D9_15-06-2022_01h01m45s_2916px.txt
      PRCSD_D9_14-06-2022_16h01m21s_2916px.txt
      PRCSD_D9_13-06-2022_15h56m26s_2916px.txt
      PRCSD_D8_17-06-2022_14h05m08s_2916px.txt
      PRCSD_D8_16-06-2022_20h05m38s_2916px.txt
      PRCSD_D8_15-06-2022_18h05m10s_2916px.txt
      PRCSD_D11_15-06-2022_02h02m19s_2916px.txt
```

Figure 78: Second processing of the result images (i.e. inference on the processed images). *Every processed image is inferred on (above) and a corresponding .txt file containing the inference results is generated (below). In the right image, the inferences are represented as red bounding-boxes with a red label. These boxes are easier to see in the next Figure.*

From this step on, the images that are used for the analysis of the P_0 path and the ones used to determine the brood size are processed differently. The former's P_0 is skeletonized to parametrize it (i.e. length, curvature) before analyzing its complete path. This involves a series of transformation from the OpenCV Python library where the animal is first blurred to remove small, insignificant details and make its body uniformly darker. A threshold is then used so that only the darkest pixels (i.e. the body of the worm) remain as white pixels. The form is then eroded and dilated to remove potential details and only keep the meaningful curvature of the animal. The contour of the shape is then calculated to ultimately derive its median line, which completes the skeletonization process (Figure 79).

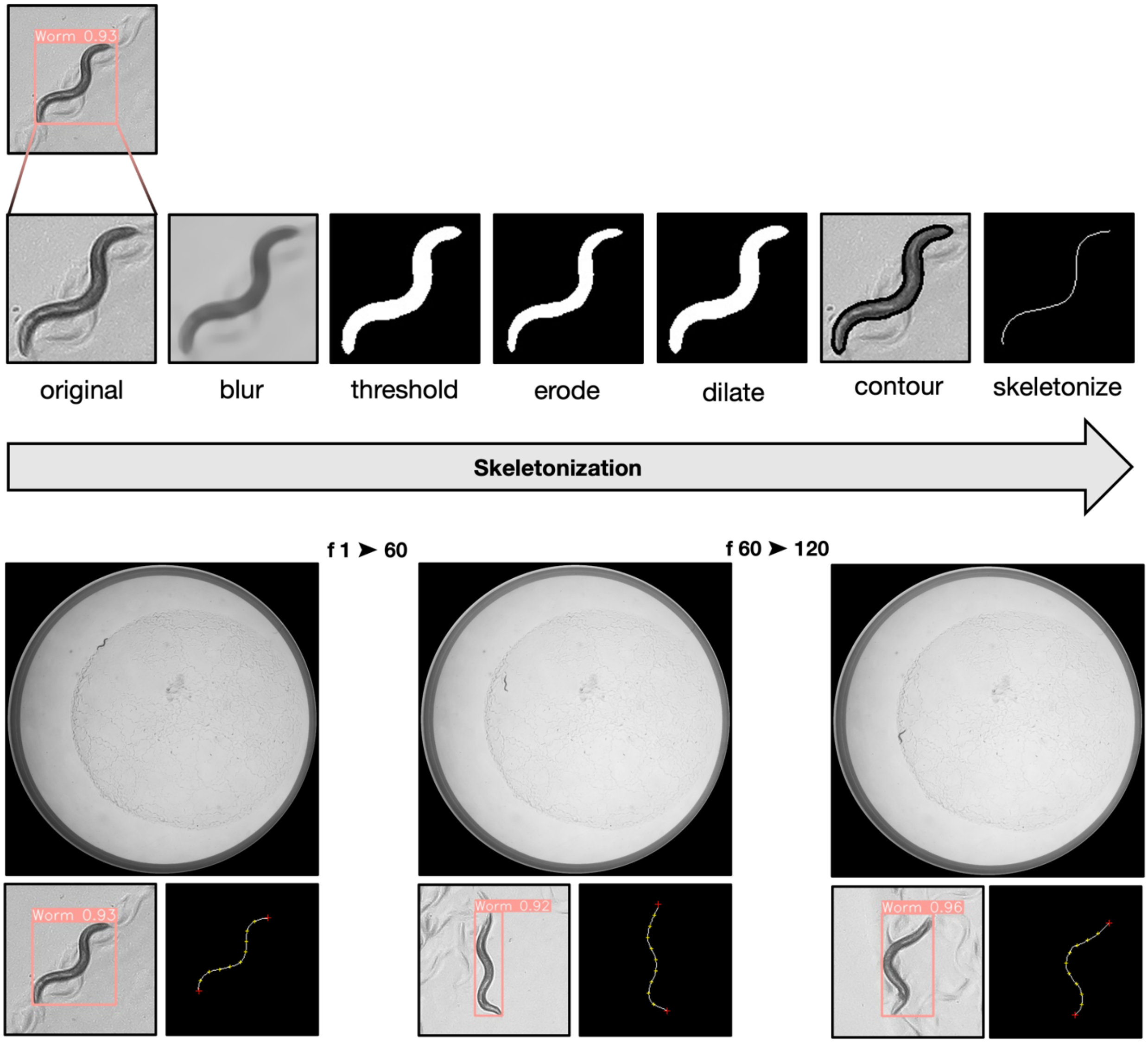


Figure 79: Skeletonization of the P0 (above) during the first 120s of surveillance (below).

The resulting path is used to determine the speed of the animal, as well as its average position from all data points compared to the median point of the shortest start-to-end path (Figure 80).

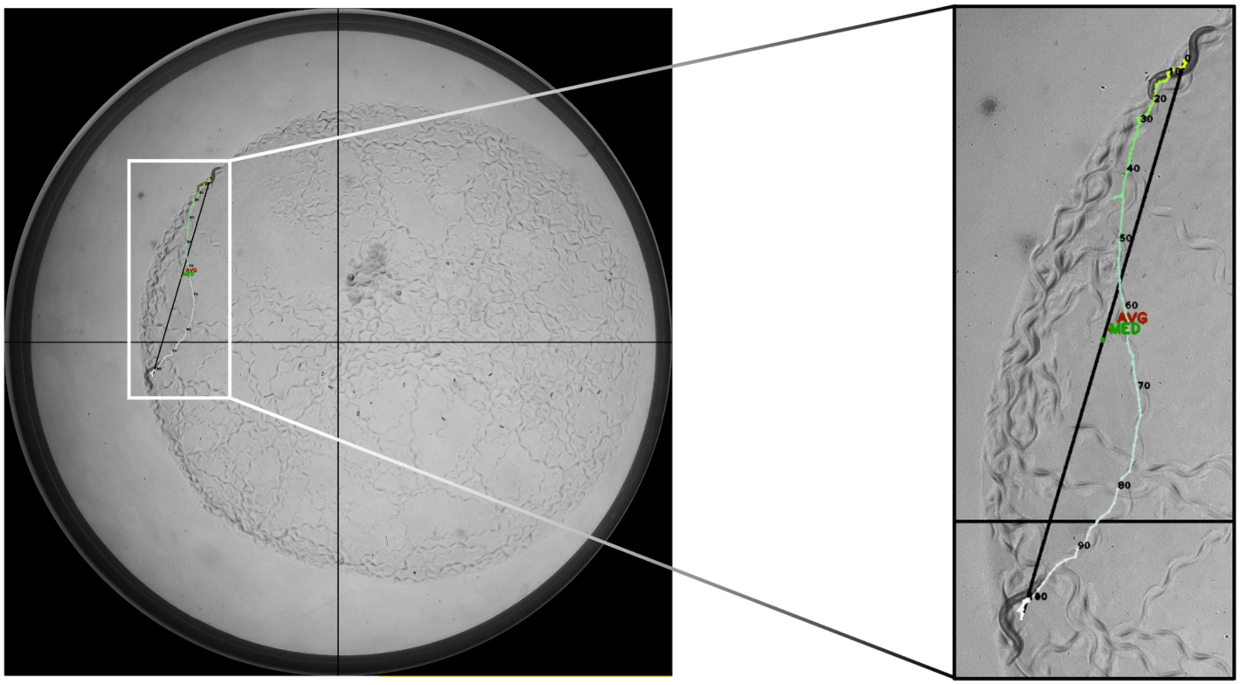
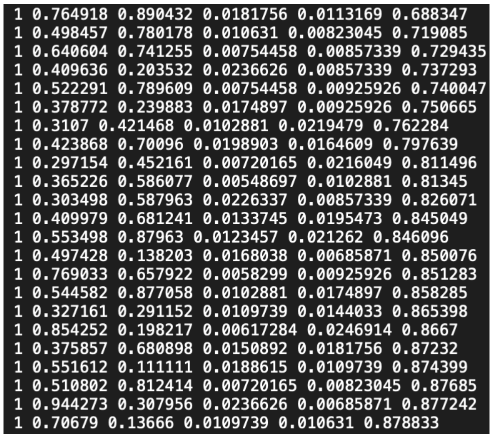


Figure 80: Path analysis of the P0 during the first 120s of surveillance. *Every 10th position is notated on the path, along with the average position and the median point of the shortest start-to-end path (i.e. black line from position 1 to 120).*

The images used to determine the brood size (Figure 81) have many more detected objects than the ones used for estimating the path traveled by the parent. The data derived from these images do not span 2mins but rather ~10 days.



```
1 0.764918 0.890432 0.0181756 0.0113169 0.688347
1 0.498457 0.780178 0.010631 0.00823045 0.719085
1 0.640604 0.741255 0.00754458 0.00857339 0.729435
1 0.409636 0.203532 0.0236626 0.00857339 0.737293
1 0.522291 0.789609 0.00754458 0.00925926 0.740047
1 0.378772 0.239883 0.0174897 0.00925926 0.750665
1 0.3107 0.421468 0.0102881 0.0219479 0.762284
1 0.423868 0.70096 0.0198903 0.0164609 0.797639
1 0.297154 0.452161 0.00720165 0.0216049 0.811496
1 0.365226 0.586077 0.00548697 0.0102881 0.81345
1 0.303498 0.587963 0.0226337 0.00857339 0.826071
1 0.409979 0.681241 0.0133745 0.0195473 0.845049
1 0.553498 0.87963 0.0123457 0.021262 0.846096
1 0.497428 0.138203 0.0168038 0.00685871 0.850076
1 0.769033 0.657922 0.0058299 0.00925926 0.851283
1 0.544582 0.877058 0.0102881 0.0174897 0.858285
1 0.327161 0.291152 0.0109739 0.0144033 0.865398
1 0.854252 0.198217 0.00617284 0.0246914 0.8667
1 0.375857 0.680898 0.0150892 0.0181756 0.87232
1 0.551612 0.111111 0.0188615 0.0109739 0.874399
1 0.510802 0.812414 0.00720165 0.00823045 0.87685
1 0.944273 0.307956 0.0236626 0.00685871 0.877242
1 0.70679 0.13666 0.0109739 0.010631 0.878833
```

Figure 81: .txt file generated by the trained neural network with one detected object per line. *Each line represents a bounding box with, from left to right, a class, an (X, Y) center, a width, and a height.*

One of the ways to visualize this new data is by looking at the distribution of eggs and worms at any given hour (Figure 82).

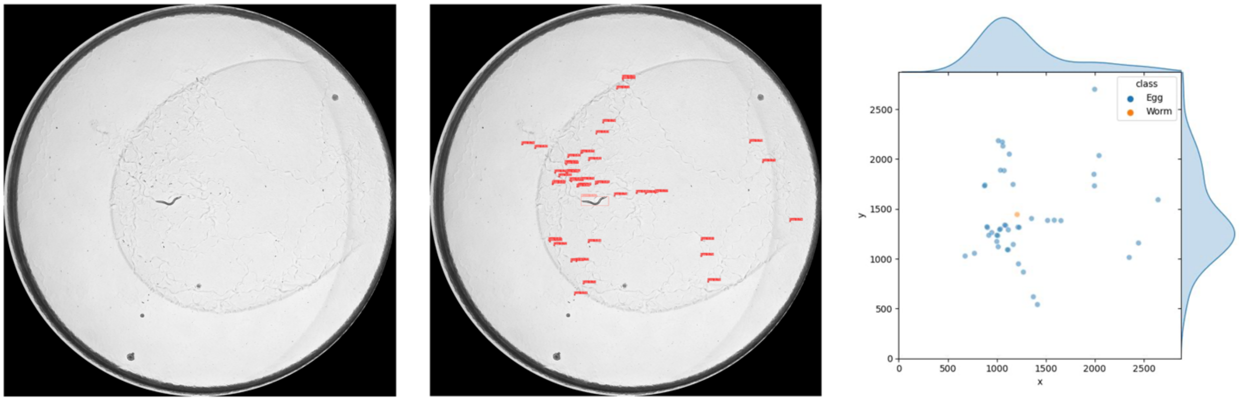


Figure 82: Visualization of the analysis of a well, with the distribution of its Egg and Worm populations (right), where each dot has transparency and will appear darker when on top of another dot. *The image on the left is the original image and the one in the middle is the one after inference by the trained model.*

The above data already represents a lot of data, and yet, it only represents a fraction (i.e. one given well at a given hour) of the overall data which we are most interested in: the evaluation of the brood size. Perhaps this demographic density of Egg vs. Worm could be studied over time to evaluate how it changes but, at this stage, we focused on the evaluation of all the images with respect to the brood size. It must also be said that the precise treatment of ~10,000 of images, compounded in data variables to produce meaningful data for researchers was definitely one of the more challenging data object and data abstraction aspects of this project. However, the result is a very rewarding one as it is the principal marker to validate whether my robot is viable.

4.3 Project Validation

To validate our model, software, and hardware, I set up an experiment of brood size counting opposing CeSAR to laboratory members of different expertise levels. My robot, students at the *B.Sc.*, *M.Sc.*, and *Ph.D.* level, and our professor (i.e. postdoctorate level) had to perform brood size measurements on n = 10 wildtype animals. Interestingly, questioning each experimenter revealed that each had their own slightly different method for evaluating the brood size, with the postdoc including dead eggs and arrested larvae in his dataset.

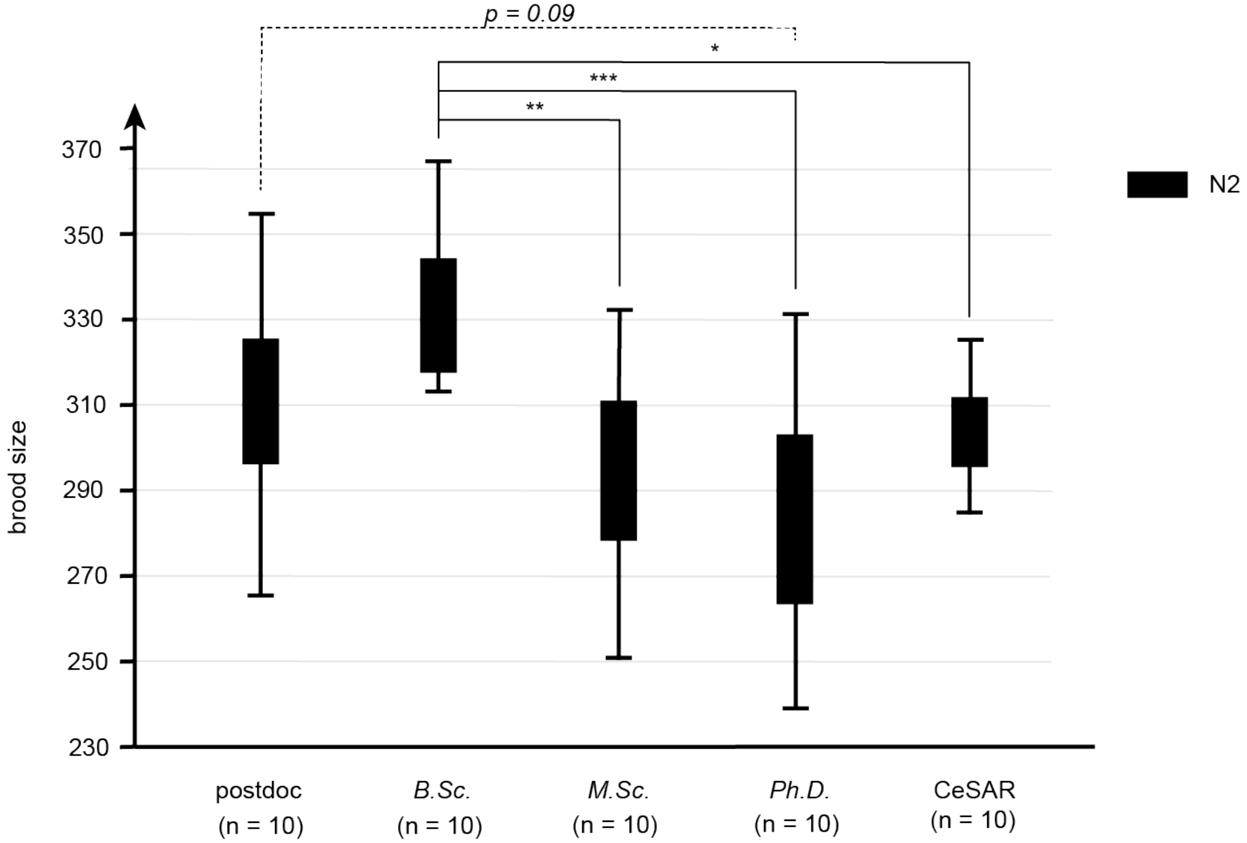


Figure 83: Validation experiment of manual brood size counting of n = 10 N2 C. elegans by four experimenters of different expertise level, and automatic counting by CeSAR.

There were no significant difference between CeSAR's results and the postdoctorate results, demonstrating its accuracy. There however were some significant differences between some of the experimenters, with the undergraduate intern being slightly off from everyone else. This suggests that some level of training is essential in correctly asserting the *C. elegans* brood size. Second, we noticed that the worm-to-worm brood size variability was lowest in CeSAR's dataset. This may result from the reduced handling of the parents (i.e. manual brood size evaluation require re-picking of the parent to a new plate every day), and highly-controlled and stable environment. From these promising results, it would be interesting to go further and conduct two additional experiments. The first would be used to further validate CeSAR by comparing *daf-2*(*e1370*) temperature-sensitive mutants, which have decreased Insulin/IGF-1 (Insulin-like Growth Factor) receptor signaling to the wild-type at 20°C, a temperature that mildly inactivates *daf-2*[58]. At this temperature, *daf-2* animals and move slower than wildtype[59] and also show slower oogenesis (Ref: Lopez III and Arur, Dev Cell paper). CeSAR should be able to detect the reduced brood size of *daf-2* mutants. The second, and perhaps most exciting, experiment is to gather data for n > 100 wildtype animals and look for phenotypical traits from our parametrization that would be predictive of an animal's brood size.

CHAPTER V: DISCUSSION

We wanted to evaluate the possibility of developing a better brood size characterization method than the manual one used ubiquitously in *C. elegans* laboratories. Our work has successfully improved the methodology for characterizing *C. elegans*, not only allowing to normalize and perform complex brood size measurements using a custom-made robot (CeSAR), but also managed to characterize major motor programs in the parent. Indeed, we designed and constructed an entirely novel surveillance robot that can fit in an incubator and support four 24-wells plates (n=96 wells). We used readily available materials and 3D-printed pieces that were hard to build manually to keep the entire cost to build under $1,500 USD. We developed open source softwares to pilot the robot remotely, perform the image analyses, and use a region-based convolutional neural network (R-CNN) to infer egg and worm positions. By using both a macroscopic and a microscopic lens, we were able to simultaneously study a parent worm while studying its brood characteristics (*e.g.* size, egg and larval lethality). Our results suggest that our method for studying *C. elegans* is more reliable than the current manual method used and taught in laboratories and is certainly much less time-consuming. Indeed, CeSAR required the preparation of the wells and the two transfers of the parent animal, while the experimenters had to count multiple at least n = 10, and at most n = 30 (i.e. the current batch, in addition to the ones from 24h and 72h) every day. This translates to a couple of hours vs. easily more than 10 hours. It is good to keep in mind that all of the time used by the experimenters to each measure the brood size of n = 10 animals could have been done at the same time by CeSAR, which was only using 10 of its 48 available wells.

Our surveillance system is built around the two lenses such that each have their own light source. Made from PVC, the cylinders containing the light sources serve as inexpensive implementations and allow the price of the robot to be highly affordable (Figure 37A). However, those tubes are rather bulky and inflexible. This is why it would be interesting to develop two light chambers that allowed better control over their positioning to illuminate the worms. Our system uses two stepping motors in a coreXY configuration to move the center stage. This allows for the maintenance of constant tension from both sides, on the X-axis, of the center stage. It also reduces lateral and vertical space that would be used with the other existing systems: the motors are below the stage instead of outside and both timing belts can be vertically very close together, as long as they do not touch (Figure 37B). The linear motion in both the X and Y directions is permitted by linear rails along which bronze sleeves slide. Those sleeves are glued using epoxy glue to the corresponding stage. Although this yields excellent precision (*i*.e. < 100µm of absolute precision when moving the stage), it creates potential weak points that are constantly subject to forces. It would, therefore, be interesting to test a different railing system to achieve smoother linear motion with perfect orthogonality. Also, since epoxy glue cannot be easily removed, it makes the entire process very unforgiving: once the glue has set, there is no possibility of readjustments. An example of a design with extremely smooth linear motion and control would be from Cermak *et al.*[42], although it only focuses on one worm in one well for 6h and does not have high amplitude of motorization in both axes.

The inner stage in our design is used to house the two lenses and two corresponding motors to make them autofocus and is built out of four horizontal levels. This allows the experimenter to move each one of them to the desired height to get better images. However, this a very specialized implementation that does not leave much room for other hardware choices such as different sensors or different lenses. Although minor, this could be improved by having lenses with built-in autofocus such that only two stages would need to be moved instead of four. This reveals a potential weakness in our robot, which lies in the way the lenses turn in their sockets to autofocus. This could create enough friction in the long run to perhaps damage the threading over time and make the lens fit more loosely. Regarding the lenses, since they are upside down, an important consideration is the dampening of vibrations that might get to it. Indeed, without dampening devices, the microscopic imaging could get blurry from time to time in our setting due to fluctuating building vibrations, making it difficult to see key anatomical features such as the grinder, which, located at the base of *C. elegans*' pharynx, and which moves every time it sucks in bacteria.

Likewise, we chose the official and readily available 12.3Mpx IMX477 sensor as cameras. Since starting this R&D project, the available hardware has evolved significantly to the point where new sensors are currently available with 64Mpx and autofocus. Such a sensor would be an obvious upgrade for the next-generation CeSAR.

The software architecture and flow of information allows for both a hierarchical and clear communication between its components (Figure 37C). That way, for instance, both RPI's can communicate as client-server: the server remains always open while the client can send requests to it at any time point. To simplify this design, it would be beneficial to connect each lens to the same RPI as it would remove the need for a second RPI with which to establish and maintain proper communication. Currently, the modules available to connect multiple camera sensors to the

RPI use too many GPIO (General Purpose Input/Output) pins, which are necessary to monitor environmental conditions, which is why we did not consider this option. Should a more versatile RPI come out on the market however, it would provide another obvious upgrade to CeSAR.

With this bare structure, we developed the software to infer the position of every egg and worm in the image as well as characterize the parental phenotypical traits. The object detection, although accurate, does not discriminate against different larval stages, which is why the analysis of the larval stage must be done after the original inference has taken place. Indeed, during initial testing, the neural network was taught, with my own experimenter biases, to discriminate between Egg, L1, L2, L3, L4 and Adult objects. However, by doing this, I unknowingly created juxtaposition of these classes and taught the resulting inaccuracies to the network. This produced inference bounding boxes of adjacent larval stages, most notably L1 and L2, on the same object. In other words, whether the postdoc experimenter or myself, we were unable to label every object in an image with the same labels twice in a row. This meant that our method of labeling was unreliable and de facto, unproperly teachable to a neural network. Therefore, the algorithmic design of having only the easily differentiable Egg and Worm objects is easily teachable both to a new experimenter and the network. The biases are now reduced to ones of false negatives and positives, where the experimenter might falsely not identify the object when it actually is either an Egg or a Worm, or falsely identify an Egg or Worm when it isn't one, respectively. With clean arenas, however, this is unlikely, and the most potential false negative remains L1 animals that are the most difficult to identify. Therefore, although we do not expect large inter-experimenters differences, one must admit that having multiple experimenters could be useful in properly exhaustively determining biases in the training process. Analyzing the parent microscopically in addition to measuring its brood size has the potential to perform highly accurate lifespan and healthspan assays. This is key because in past platforms, such as for the WormBot[35], worms were studied macroscopically specifically, which means a lot of important healthspan traits were lost such as the defecation rate and pumping rate. Indeed, in that work, as well as in the WorMotel's[55], the worms are studied at such a distance that they were acquired at a relatively low resolution, which rendered those kinds of healthspan assays rather difficult, if not impossible.

In terms of CeSAR's repeatability and result accuracy, the fact that both the focus and XY recalibration of the system are controlled by the experimenter makes it a particularly strong feature of the system. Indeed, it allows for flexibility in the initial precision of either positioning or focus before getting a more accurate level using software. This means that even if the system had an original XY precision of 100µm, it could artificially be increased by using software. It would, however, be interesting to build an inexpensive system that attains a high level of accuracy from its construction alone, rather than relying on software recalibration. A step towards achieving this would be to rethink the construction design and not rely on a uniquely manual assembly of the different stages using epoxy glue. This step is key and would be the first one to tackle to ensure system orthogonality and could greatly benefit from having a precise grid for assembly as well as mechanical means to ensure that all parts are held perfectly before gluing them to one another. Another step would be to have more precise motors and a structure that allows for more taut belts. However, this would require some creativity not to add too much cost to the design.

We showed that, regardless of the expertise level, CeSAR has a higher frequency of analysis, seems to have a smaller interquartile range (Figure 83) and can produce highly detailed reports on every studied well. Although we must recognize a potential bias coming from myself toward favouring the success of CeSAR over our own labour, the data clearly showed no significant difference in the counting of the brood size between our system and our reference postdoc experimenter. A potential flaw to this experiment's design, however, is that it involves only one experimenter for each category of expertise level. This could bias the results (i.e. to be either a smaller or bigger brood size), depending solely on that experimenter's skill at measuring a brood size. On the other hand, increasing the sample size would have likely yielded averages that would tend to equalize as the sample sizes increased, defeating the purpose of this analysis, which was to make sure CeSAR was working properly, and to find out whether there were brood size variability that dependent on the experimenter.

A clear overall strength of this surveillance system is its modularity. Indeed, even though it was built originally to accurately measure brood sizes, it can be used for a variety of other purposes. It can be used for survival[60], chemotaxis, pathogen aversion[61], lifespan, healthspan, or behavioral[40,41] assays for instance, without having to change the robot fundamentally. Only some programming would be required. Similarly, in its inexpensive structural design, a team could easily make necessary modifications to CeSAR to explore other specific research question, such as one that would require the implementation of fluorescence microscopy to monitor neuronal activity or using optogenetics. For example, it would seem especially interesting to study *C. elegans*' lethargic sleep-like state[62] by observing the parent worm from egg to dying adult and derive patterns for different mutant strains. The greater sample size and multiple parameters that CeSAR can measure open the field for revealing potential correlations between brood characteristics and parental phenotypical traits. For instance, it would allow to verify if some of these traits could be used to predict brood size. Through these kinds of correlations, we could better understand the influence of certain genes that may affect not only the brood size but simultaneously influence other parameters.

An example of this modularity advantage is that it can benefit from technological advancement. For instance, in the past few months, several new iterations of hardware and software components used in this project have come out, publicly available for sale. These improvements include a new 64Mpx camera from Arducam with autofocus capability and a new version of RPI, which is expected to be released in 2024. The former will allow, albeit being more computationally costly, for much more detailed images and will not require any external mechanical forces to adjust the focus. These features should further enable the use of the additional microscopic camera that was initially envisioned for this project, as it could then be more solidly anchored to CeSAR's frame to prevent it from vibrating. There is also the Google Coral USB Edge Tensor Processing Unit (TPU) Machine Learning (ML) Accelerator coprocessor which would make it possible to perform the inferences directly from the RPI. Since the start of this project, newer versions of the Ultralytics YOLO model have been made available, namely YOLOv8. Training with it and having different lighting and centering conditions should help increase the mAP of the neural network and reduce some of its potential biases, respectively.

By studying some egg-laying combined with behavioral study on *C. elegans* mutants, we could, for instance, better understand its nervous system development[63]. We also believe our robot constitutes a step towards automation of critical tasks, namely evaluation of the brood size to characterize new alleles more accurately. Our surveillance system also couples the phenotypical analysis of the parent responsible for the brood with the brood itself as it may give precious information potentially related to the studied mutation. We believe that a strong surveillance system has the potential of being used by several other research teams that use *C. elegans* as a model system. This was the case for Jushaj et al. (2020)[64], who used the WorMotel to propose better criteria for healthspan assays based on locomotion. We estimate that our robot will allow a great number of research projects, related to chemotaxis, brood size or healthspan to take place thanks to its great flexibility in closely monitoring the entire well for a specific time frame at a high image resolution.

REFERENCES

1 Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in Drosophila. *Nat Methods* **6**, 297-303, doi:10.1038/nmeth.1310 (2009).

2 Ahloy-Dallaire, J., Klein, J. D., Davis, J. K. & Garner, J. P. Automated monitoring of mouse feeding and body weight for continuous health assessment. *Lab Anim* **53**, 342-351, doi:10.1177/0023677218797974 (2019).

3 Maupas, E. La mue et l'enkystement chez les nématodes. . *Arch. Zool. Exp. Gen.* **7**, 563-628 (1899).

4 Dusenbery, D. B. Using a microcomputer and video camera to simultaneously track 25 animals. *Comput Biol Med* **15**, 169-175, doi:10.1016/0010-4825(85)90058-7 (1985).

5 Dhawan, R., Dusenbery, D. B. & Williams, P. L. Comparison of lethality, reproduction, and behavior as toxicological endpoints in the nematode Caenorhabditis elegans. *J Toxicol Environ Health A* **58**, 451-462, doi:10.1080/009841099157179 (1999).

6 Hardaker, L. A., Singer, E., Kerr, R., Zhou, G. & Schafer, W. R. Serotonin modulates locomotory behavior and coordinates egg-laying and movement in Caenorhabditis elegans. *J Neurobiol* **49**, 303-313, doi:10.1002/neu.10014 (2001).

7 Cronin, C. J. *et al.* An automated system for measuring parameters of nematode sinusoidal movement. *BMC Genet* **6**, 5, doi:10.1186/1471-2156-6-5 (2005).

8 Li, S., Stone, H. A. & Murphy, C. T. A microfluidic device and automatic counting system for the study of C. elegans reproductive aging. *Lab Chip* **15**, 524-531, doi:10.1039/c4lc01028k (2015).

9 Buckingham, S. D. & Sattelle, D. B. Strategies for automated analysis of C. elegans locomotion. *Invert Neurosci* **8**, 121-131, doi:10.1007/s10158-008-0077-3 (2008).

10 Husson, S. J., Costa, W. S., Schmitt, C. & Gottschalk, A. Keeping track of worm trackers. *WormBook*, 1-17, doi:10.1895/wormbook.1.156.1 (2013).

11 Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115-118, doi:10.1038/nature21056 (2017).

12 Bolton, F. J., Bernat, A. S., Bar-Am, K., Levitz, D. & Jacques, S. Portable, low-cost multispectral imaging system: design, development, validation, and utilization. *J Biomed Opt* **23**, 1-11, doi:10.1117/1.Jbo.23.12.121612 (2018).

13 Tristan-Landin, S. B., Gonzalez-Suarez, A. M., Jimenez-Valdes, R. J. & Garcia-Cordero, J. L. Facile assembly of an affordable miniature multicolor fluorescence microscope made of 3D-printed parts enables detection of single cells. *PLoS One* **14**, e0215114, doi:10.1371/journal.pone.0215114 (2019).

14 Stewart, C. & Giannini, J. Inexpensive, Open Source Epifluorescence Microscopes. *Journal of Chemical Education* **93**, 1310-1315, doi:10.1021/acs.jchemed.5b00984 (2016).

15 Miller, A. R. *et al.* Portable, battery-operated, low-cost, bright field and fluorescence microscope. *PLoS One* **5**, e11890 (2010).

16 Needs, S. H. *et al.* Exploiting open source 3D printer architecture for laboratory robotics to automate high-throughput time-lapse imaging for analytical microbiology. *PLoS One* **14**, e0224878, doi:10.1371/journal.pone.0224878 (2019).

17 Maia Chagas, A., Prieto-Godino, L. L., Arrenberg, A. B. & Baden, T. The €100 lab: A 3D-printable open-source platform for fluorescence microscopy, optogenetics, and accurate temperature control during behaviour of zebrafish, Drosophila, and Caenorhabditis elegans. *PLoS Biol* **15**, e2002702, doi:10.1371/journal.pbio.2002702 (2017).

18 Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput ethomics in large groups of Drosophila. *Nat Methods* **6**, 451-457, doi:10.1038/nmeth.1328 (2009).

19 Diederich, B. *et al.* A versatile and customizable low-cost 3D-printed open standard for microscopic imaging. *Nat Commun* **11**, 5979, doi:10.1038/s41467-020-19447-9 (2020).

20 Wright, L. Laser De-Bayering of Low-Cost Image Sensors. *IEEE Sensors Letters* **7**, 1-4, doi:10.1109/LSENS.2022.3231931 (2023).

21 Lukas, R. & Plataniotis, K. N. in *23rd Biennial Symposium on Communications, 2006.* 352-355.

22 Eckelt, K., Masanas, H., Llobet, A. & Gorostiza, P.

23 Ishaq, O. *et al.* AUTOMATED QUANTIFICATION OF ZEBRAFISH TAIL DEFORMATION FOR HIGH-THROUGHPUT DRUG SCREENING. *Proc IEEE Int Symp Biomed Imaging*, 902-905, doi:10.1109/isbi.2013.6556621 (2013).

24 Barreiros, M. d. O. *et al.* Zebrafish automatic monitoring system for conditioning and behavioral analysis. *Scientific Reports* **11**, 9330, doi:10.1038/s41598-021-87502-6 (2021).

25 Early, J. J. *et al.* An automated high-resolution in vivo screen in zebrafish to identify chemical regulators of myelination. *Elife* **7**, doi:10.7554/eLife.35136 (2018).

26 Nouhaud, P., Mallard, F., Poupardin, R., Barghi, N. & Schlötterer, C. High-throughput fecundity measurements in Drosophila. *Sci Rep* **8**, 4469, doi:10.1038/s41598-018-22777-w (2018).

27 Schumann, I. & Triphan, T. The PEDtracker: An Automatic Staging Approach for Drosophila melanogaster Larvae. *Front Behav Neurosci* **14**, 612313, doi:10.3389/fnbeh.2020.612313 (2020).

28 Scaplen, K. M. *et al.* Automated real-time quantification of group locomotor activity in Drosophila melanogaster. *Sci Rep* **9**, 4427, doi:10.1038/s41598-019-40952-5 (2019).

29 Itskov, P. M. *et al.* Automated monitoring and quantitative analysis of feeding behaviour in Drosophila. *Nat Commun* **5**, 4560, doi:10.1038/ncomms5560 (2014).

30 Arend, D. *et al.* Quantitative monitoring of Arabidopsis thaliana growth and development using high-throughput plant phenotyping. *Scientific Data* **3**, 160055, doi:10.1038/sdata.2016.55 (2016).

31 Tovar, J. C. *et al.* Raspberry Pi-powered imaging for plant phenotyping. *Appl Plant Sci* **6**, e1031, doi:10.1002/aps3.1031 (2018).

32 Nanjappa, A. *et al.* Mouse Pose Estimation From Depth Images. *ArXiv* **abs/1511.07611** (2015).

33 Andresen, N. *et al.* Towards a fully automated surveillance of well-being status in laboratory mice using deep learning: Starting with facial expression analysis. *PLoS One* **15**, e0228059, doi:10.1371/journal.pone.0228059 (2020).

34 Akintayo, A. *et al.* A deep learning framework to discern and count microscopic nematode eggs. *Sci Rep* **8**, 9145, doi:10.1038/s41598-018-27272-w (2018).

35 Pitt, J. N. *et al.* WormBot, an open-source robotics platform for survival and behavior analysis in C. elegans. *Geroscience* **41**, 961-973, doi:10.1007/s11357-019-00124-9 (2019).

36 García Garví, A., Puchalt, J. C., Layana Castro, P. E., Navarro Moya, F. & Sánchez-Salmerón, A. J. Towards Lifespan Automation for Caenorhabditis elegans Based on Deep Learning: Analysing Convolutional and Recurrent Neural Networks for Dead or Live Classification. *Sensors (Basel)* **21**, doi:10.3390/s21144943 (2021).

37 Puchalt, J. C. *et al.* Small flexible automated system for monitoring Caenorhabditis elegans lifespan based on active vision and image processing techniques. *Sci Rep* **11**, 12289, doi:10.1038/s41598-021-91898-6 (2021).

38 Puchalt, J. C., Sánchez-Salmerón, A. J., Martorell Guerola, P. & Genovés Martínez, S. Active backlight for automating visual monitoring: An analysis of a lighting control technique for Caenorhabditis elegans cultured on standard Petri plates. *PLoS One* **14**, e0215548, doi:10.1371/journal.pone.0215548 (2019).

39 Jung, S. K., Aleman-Meza, B., Riepe, C. & Zhong, W. QuantWorm: a comprehensive software package for Caenorhabditis elegans phenotypic assays. *PLoS One* **9**, e84830, doi:10.1371/journal.pone.0084830 (2014).

40 Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. High-throughput behavioral analysis in C. elegans. *Nat Methods* **8**, 592-598, doi:10.1038/nmeth.1625 (2011).

41 Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. & Schafer, W. R. A database of Caenorhabditis elegans behavioral phenotypes. *Nat Methods* **10**, 877-879, doi:10.1038/nmeth.2560 (2013).

42 Cermak, N. *et al.* Whole-organism behavioral profiling reveals a role for dopamine in state-dependent motor program coupling in C. elegans. *Elife* **9**, doi:10.7554/eLife.57093 (2020).

43 Bornhorst, J., Nustede, E. J. & Fudickar, S. Mass Surveilance of C. elegans—Smartphone-Based DIY Microscope and Machine-Learning-Based Approach for Worm Detection. *Sensors* **19**, 1468 (2019).

44 Wang, S. J. & Wang, Z. W. Track-a-worm, an open-source system for quantitative assessment of C. elegans locomotory and bending behavior. *PLoS One* **8**, e69653, doi:10.1371/journal.pone.0069653 (2013).

45 Fukunaga, T. & Iwasaki, W. Inactivity periods and postural change speed can explain atypical postural change patterns of Caenorhabditis elegans mutants. *BMC Bioinformatics* **18**, 46, doi:10.1186/s12859-016-1408-8 (2017).

46 Krajacic, P., Shen, X., Purohit, P. K., Arratia, P. & Lamitina, T. Biomechanical profiling of Caenorhabditis elegans motility. *Genetics* **191**, 1015-1021, doi:10.1534/genetics.112.141176 (2012).

47 Roussel, N., Sprenger, J., Tappan, S. J. & Glaser, J. R. Robust tracking and quantification of C. elegans body shape and locomotion through coiling, entanglement, and omega bends. *Worm* **3**, e982437, doi:10.4161/21624054.2014.982437 (2014).

48 Hebert, L., Ahamed, T., Costa, A. C., O'Shaughnessy, L. & Stephens, G. J. WormPose: Image synthesis and convolutional networks for pose estimation in C. elegans. *PLoS Comput Biol* **17**, e1008914, doi:10.1371/journal.pcbi.1008914 (2021).

49 Wake, M. H. in *Encyclopedia of Reproduction (Second Edition)* (ed Michael K. Skinner) 18-22 (Academic Press, 2018).

50 T, M., Hari Dass, S. & Sharma, V. K. Egg-laying rhythm in Drosophila melanogaster. *J Genet* **87**, 495-504, doi:10.1007/s12041-008-0072-9 (2008).

51 Kannan, N. N., Reveendran, R., Hari Dass, S., Manjunatha, T. & Sharma, V. K. Temperature can entrain egg laying rhythm of Drosophila but may not be a stronger zeitgeber than light. *Journal of Insect Physiology* **58**, 245-255, doi:https://doi.org/10.1016/j.jinsphys.2011.11.012 (2012).

52 Hardin, P. E. The circadian timekeeping system of Drosophila. *Curr Biol* **15**, R714-722, doi:10.1016/j.cub.2005.08.019 (2005).

53 Girard, L. R. *et al.* WormBook: the online review of Caenorhabditis elegans biology. *Nucleic Acids Res* **35**, D472-475, doi:10.1093/nar/gkl894 (2007).

54 Altun, Z. F., Herndon, L.A., Wolkow, C.A., Crocker, C., Lints, R. and Hall, D.H. (ed.s) *WormAtlas*, <http://www.wormatlas.org> (2002-2023).

55 Churgin, M. A. *et al.* Longitudinal imaging of Caenorhabditis elegans in a microfabricated device reveals variation in behavioral decline during aging. *Elife* **6**, doi:10.7554/eLife.26652 (2017).

56 Brenner, S. The genetics of Caenorhabditis elegans. *Genetics* **77**, 71-94, doi:10.1093/genetics/77.1.71 (1974).

57 Wong, S. S., Yu, J., Schroeder, F. C. & Kim, D. H. Population Density Modulates the Duration of Reproduction of C. elegans. *Curr Biol* **30**, 2602-2607.e2602, doi:10.1016/j.cub.2020.04.056 (2020).

58 Gems, D. *et al.* Two pleiotropic classes of daf-2 mutation affect larval arrest, adult behavior, reproduction and longevity in Caenorhabditis elegans. *Genetics* **150**, 129-155, doi:10.1093/genetics/150.1.129 (1998).

59 Podshivalova, K., Kerr, R. A. & Kenyon, C. How a Mutation that Slows Aging Can Also Disproportionately Extend End-of-Life Decrepitude. *Cell Rep* **19**, 441-450, doi:10.1016/j.celrep.2017.03.062 (2017).

60 Park, H. H., Jung, Y. & Lee, S. V. Survival assays using Caenorhabditis elegans. *Mol Cells* **40**, 90-99, doi:10.14348/molcells.2017.0017 (2017).

61 Kaletsky, R. *et al.* C. elegans interprets bacterial non-coding RNAs to learn pathogenic avoidance. *Nature* **586**, 445-451, doi:10.1038/s41586-020-2699-5 (2020).

62 Raizen, D. M. *et al.* Lethargus is a Caenorhabditis elegans sleep-like state. *Nature* **451**, 569-572, doi:10.1038/nature06535 (2008).

63 Schafer, W. F. Genetics of egg-laying in worms. *Annu Rev Genet* **40**, 487-509, doi:10.1146/annurev.genet.40.110405.090527 (2006).

64 Jushaj, A. *et al.* Optimized criteria for locomotion-based healthspan evaluation in C. elegans using the WorMotel system. *PLoS One* **15**, e0229583, doi:10.1371/journal.pone.0229583 (2020).