The Biophysicist

Teaching Image Processing and Optical Engineering to University Biology Students

Thomas Zimmerman[1,2,3,*], Raymond Esquerra[1,2], Yee-Hung Mark Chan[1,2], Anagha Kulkarni[1], Nicole Adelstein[1], Ashley Albright[1,2,4], Jiayu Luo[1], Ziah Dean[1], Salma Ahmed[1], Michelle Phillips[5], Simone Bianco[2,3], Sara Capponi[2,3]

[1]San Francisco State University, San Francisco, CA, USA
[2]Center for Cellular Construction, San Francisco, CA, USA
[3]IBM Research–Almaden, Almaden, CA, USA
[4]University of California San Francisco, San Francisco, CA, USA
[5]Phillips & Associates, San Francisco, CA, USA

ABSTRACT Biophysics is an interdisciplinary pursuit requiring researchers with knowledge and skills in several areas. Optical instruments and computers are fundamental tools in biophysics research to collect and analyze data. We developed a 1-semester Optical Engineering Laboratory course to teach image processing, optical engineering, and research skills to undergraduate students majoring in biology and biochemistry. With the use of development systems on students' laptops and in the cloud, students learned image processing with Python and OpenCV. Each student constructed a microprocessor-based lensless holographic microscope, gaining hands-on experience with optical engineering. The class culminated in original, student-designed research projects. All lectures, hands-on labs, and student research projects were performed both in person and remotely, in response to the COVID-19 pandemic.

KEY WORDS biophysics; CURE; holography

I. INTRODUCTION

Microscopes are an essential tool in biology. Today's microscopes comprise optical components and a digital image sensor, with much of the analysis of captured images processed with computers, for example to identify, count, and measure cell features. Commercial and open source software exists to perform many of these tasks, such as ImageJ (version 1.53t, Wayne Rasband, National Institute of Mental Health, Bethesda, MD). Scientists continually adapt and enhance existing microscopy hardware and image processing tools to expand their compatibility. For instance, light-sheet microscopy was adapted to monitor neurological activity of live zebrafish (1), and fluorescent microscopy was enhanced to exceed the resolution limit set by the diffraction of light (2). These "super-resolution" methods are achieved by teams with a broad range of skills, not only in biology, but in hardware, software, and "soft skills," such as teamwork and leadership (3–5).

The value of educating students in at least one field with a workable level of understanding in another field is well recognized in engineering, computer science, and management (6–10). In addition

"*" corresponding author

Received: 31 December 2022
Accepted: 30 May 2023
Published: 8 August 2023

to mastering academic knowledge and participating in laboratory courses that teach hands-on technical skills like pipetting, cell culturing, and spectroscopy, students need to develop communication and interpersonal skills to be productive members of a team (11). Students can learn these soft skills with cooperative learning, characterized by students working in small groups with shared goals, collaborative behavior, positive interdependence, and individual and group accountability and responsibility (12). For engineering students, a capstone project is an opportunity to perform independent group research, often with an industrial partner, providing a real-world context for the project and preparing them for the transition from academia to industry (13).

Course-based undergraduate research experiences (CUREs) have emerged as an effective approach to expand research skills for undergraduates with many benefits to students, including enhancing self-confidence (14), increasing diversity (15), and engaging in authentic inquiry-based research (16). Often, student research projects originate from outside the student's home institution (17). We implement a CURE laboratory-based learning course at San Francisco State University where students use microscopy and image processing to address a research question motivated by their scientific curiosity. Student projects often involve collaboration with researchers from Center for Cellular Construction (18) institutions, where collaborators help students formulate projects, share code, and provide datasets developed from their work.

The paper is organized as follows: section II describes the pedagogical principles of our Optical Engineering Laboratory syllabus which is composed of 3 sections that take place in one semester: (section II.A) a self-paced online bootcamp to teach basic programming skills necessary for image processing, (section II.B) an image processing pipeline, and (section II.C) a lab-based hands-on section in which each student builds a holographic microscope and performs original research, often with their microscope. In section III, we review the materials and methods used to teach the 3 parts described in section II. In section IV, we present the results from several student research projects. In section V, we discuss our experiences and observations from teaching the course and students' feedback from pre- and postcourse interviews. We provide a summary of the motivation and intended effect of the course on students' education and professional careers in section VI.

II. PEDAGOGICAL BACKGROUND

Our approach to the Optical Engineering Laboratory syllabus is threefold: first, teach students the basic programming skills they will need to understand and write image processing code; second, teach students image processing coding with a pipeline and dataset developed for a real research problem; third, engage students in hands-on learning of basic optical theory and engineering skills by building a holographic microscope and using it to conduct original research projects.

A. Image processing bootcamp

We selected Python (version 3.7, Python Software Foundation, Beaverton, OR) as the programming language because it is one of the most common open source languages used in science disciplines, it is easy to learn, and it has many versatile libraries covering many disciplines (19). However, many of our students are not proficient in Python and are new to image processing; therefore, we provide a self-study "bootcamp" to teach the basics of Python programming and image processing. The self-paced instructional modules are run by the students on Google's Colaboratory (Colab) platform (version 2022/6/10, Google, Mountain View, CA), a web-based development environment for Python with instructional material and code exchanged on Google Drive, a cloud-based file storage and sharing platform (20). The modules are publicly available in a GitHub repository (21). Each module is a self-contained Jupyter notebook (release 4.10.0, Jupyter community; https://jupyter.org/) with text and images explaining the purpose of the module, links to papers, videos covering the topic, and a sequence of code segments and their output to teach programming concepts and methods.

The Colab platform is well suited for self-paced learning because all the content is available online. No software needs to be installed on the student's laptop, avoiding configuration and device-specific incompatibility problems often experienced when loading an integrated development environment and libraries on a personal computer. The notebooks provide the experience of an interactive textbook, combining text and links to videos and other references, with code, images, and plots that can be run and modified by the student. The modules focus on (a) Python skills commonly used in image processing, including representing and manipulating images with the NumPy library (version 1.21.5) and plotting with the Matplotlib library (version 3.5.1) (22), and (b) image processing pipeline components from OpenCV (version 3.4.2, OpenCV team, Palo Alto, CA) and the scikit-learn library (version 1.0.1) (23).

B. Image processing pipeline

The image processing pipeline gives students a practical understanding and sufficient skills to use and write code to process the types of images they may encounter in their professional careers. To do this, we use an image processing pipeline we developed to detect and classify plankton (24, 25). The pipeline is composed of components that receive and output CSV (comma-separated value) files, allowing each component to be studied and modified by the student for their research needs. The pipeline is written entirely in Python using the OpenCV library, which students loaded onto their personal laptops with the Anaconda distribution tool (version 2.1.4) (26). The installation is more complex than using Colab but provides several advantages. Code runs much faster on a laptop, speeding up holographic reconstruction, detection, and tracking. Video can stream from the microscope to the laptop, providing real-time feedback, which is essential for adjusting camera exposure settings. Local running code can access the mouse and keyboard, enabling students to write graphical user interfaces (GUIs) to control image processing functions.

Use of the image processing pipeline as a component of the syllabus template provides 3 benefits: (a) a logical progression for the course, because the pipeline performs a sequential series of operations on objects in the image; (b) reuse of code and datasets already developed for research; and (c) teaching students practical skills directly applicable to research and providing collaboration, internship, and employment opportunities.

The pipeline covers a considerable amount of content from image capture to object classification. It was originally developed and taught as a 1-semester lecture course. When used in the Optical Engineering Laboratory course, we teach the Detect and Track components, and leave the remaining pipeline components optional for those students who have sufficient Python programming background and a need for their research project.

Rapid innovations in computing and technology have greatly reduced the cost and increased the accessibility of these tools for biophysical research. To thrive in their scientific careers, the next generation of biophysicists will need to have computational skills, and an understanding of engineering and 3-dimensional (3D) design. Future scientists need to nurture their creativity and problem-solving skills that will drive their future career success. We designed CUREs for students wherein they learn and apply research and engineering skills to answer student-driven research questions. Our pedagogical approach is modeled after curricula designed to promote inclusivity in computing (27) and provides an affordable and scalable model to teach engineering and coding to biology and biochemistry majors.

C. Hands-on learning

Laboratory courses serve a multitude of purposes for undergraduates (28). They teach the necessary hands-on skills essential for biologists who want to engage in clinical or wet bench research. They also present a variety of learning styles, including visual, auditory, kinesthetic, tactile, group, and individual learning (29). Building an instrument and using it to collect data provides perceptually rich and realistic educational experiences (30).

Table 1. Optical Engineering Laboratory course syllabus. Each week consists of one 50-min lecture and two 165-min active learning and laboratory classes. Lecture slides are available on a public GitHub repository (https://github.com/CCCofficial/OpticalEngineeringLabCourse).

Week	Topic and activities	Assessment
1	Build holographic microscope Introduction to Raspberry Pi (single-board computer), digital cameras, mechanical design, soldering, circuits, light and lasers, holography Install Python and OpenCV with Anaconda on student laptops Lecture slides: basic microscope optics	Quiz
2	Test holographic microscope USAF calibration slide, hair sample, ImageJ, image processing with Python on Colab bootcamp Lecture slides: image processing pipeline	Presentation, report
3	Plankton investigations Capture plankton holographic videos, holographic reconstruction and object detection, and tracking code on laptop Lecture slides: holographic reconstruction	Presentation
4, 5	Research planning Scientific literature review, research design, presentation, proposals	Presentation, proposals
6–14	Student-driven research projects Weekly progress presentations; peer, group, and external collaborations Feedback and research adjustments, scientific literature review, practice communication skills, develop image processing skills for projects	Weekly research progress reports
15	Final student research project presentations	Paper, presentation, departmental seminar

Additionally, incorporating computer science and engineering experiences into a laboratory course for biology majors is an opportunity to expose more women to engineering, because 3 times as many female graduates have bachelor's degrees in Biology than in Engineering and Computer Science (31). Introducing hands-on engineering experiences to undergraduate women can help decrease the gender bias around tools and machines (32–34).

In the first week of the syllabus (Table 1), students build and test a low-cost holographic microscope (Fig 1) and install Python and the OpenCV library on their personal laptops with Anaconda (35). We start this process the first week because sometimes installation problems are encountered because of the different operating systems (e.g., Windows, Mac, Linux). Code can also use functions and features that are depreciated (no longer supported) with new versions of libraries. We found the question-and-answer website Stack Overflow (36) a good source of assistance in these matters. With a peer-mentored and self-driven learning methodology, students learn the fundamentals of image processing in Python with the Colab notebooks in week 2. In week 3, students apply the skills learned in the first 2 wk to detect and track plankton. They then compare plankton features to see how different species can be distinguished quantitatively. For most of our students, all the material is new and novel.

Fig 1. Microscope construction. Each student builds their own holographic microscope in class from a kit of parts with hand drill, soldering iron, hot glue gun, and hand tools.

Nearly all the students have never drilled, soldered, or modified electronics. Most have limited coding experience in Python or R, and for some this is their first coding experience. To support these students, peer mentors and online resources help students better understand the material. Importantly, students learn and develop strategies that work for them to learn how to seek answers to overcome challenges. Additionally, to providing a support framework of peer learning, it also helps promote self-efficacy that drives success in the independent research project that wraps up the remainder of the course.

In weeks 4 and 5, students are taught research methodologies and how to write a scientific paper. Students perform literature searches of potential projects. These discoveries are shared, discussed, and refined. Most projects are driven by student interest in scientific questions of global importance, such as microplastics, chemical pollution, climate change, and artificial intelligence. In this section, students come up with a testable hypothesis based on a novel question, or they focus on a need-based research question with the use of sound scientific methods.

For the remainder of the course (weeks 6–15), students begin their independent research project that was proposed in the previous week. Students build, modify, and design new instrumentation to research their question. They collect initial data and determine how to progress in their research. These projects are usually done in groups, but sometimes students elect to work independently. A fundamental part of this project is that each student takes ownership of their project regardless of whether they are part of a team or work independently. Each week, data are collected and shared with the class and feedback is given. Often, several projects using multiple approaches address a larger research question. Some examples include: (a) How do small amounts of chemical pollutants affect plankton? (b) What is the effect of microplastics on plankton? (c) In what ways can we enhance the research performance of the microscope? (d) How can we classify plankton? In these cases, students can work together on one project or break the project into several complementary subprojects, negotiated during the research planning phase of the course syllabus (weeks 4 and 5; Table 1). For case 1, one student studied the movement of plankton when subjected to common chemicals used in the home and garden (disinfectants and pesticides), another focused on monitoring the natural variation of plankton count and shape sampled from a pond with a net they built, and a third worked on automating sample collection (reported here). For case 2, the students decided to work together to distinguish natural and synthetic microfibers, defining and dividing the tasks among the team (reported here). For case 3, students created several projects to improve the microscope, including measuring resolution under different conditions, exploring 2 approaches to changing magnification, and creating a GUI to improve usability, all reported here. For case 4, 2 students decided to work together, dividing up their work. One student curated plankton sample images, preparing them for feature extraction, and the other performed unsupervised classification, modifying the image processing pipeline code we provided to support student research projects. To demonstrate how a big project can be distributed among many, we had every class member use their holographic microscopes to capture images of plankton (131020, Carolina Biological Supply, Burlington, NC, USA) and provide them to the student curating plankton samples, who learned about the challenges of receiving data (images) of varying quality from multiples sources. In addition to fostering collaboration within and between teams, students were encouraged to consult with faculty outside the classroom and at other institutions. For example, one student worked in a university research lab studying *Caenorhabditis elegans* and brought in samples to view with their microscope. These activities help students realize that scientific progress is a community effort, and they are part of this community. The course culminates with students giving a final presentation of their research at a departmental seminar.

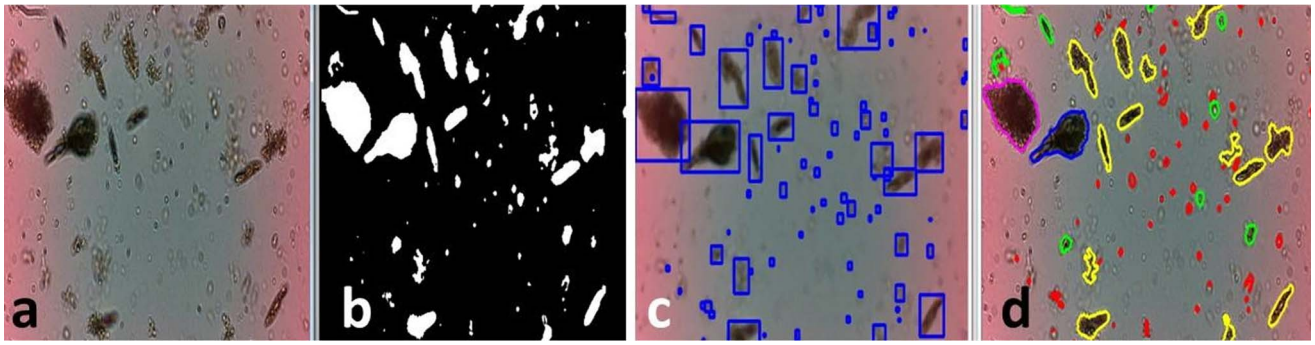

Fig 2. Images from the image processing pipeline. (a) A raw image from a lensless microscope is captured. (b) A threshold is applied to the raw image to create a binary image of white objects on a black background. (c) The OpenCV function cv2.findContours detects white objects on a black background and provides contour points and bounding box coordinates. (d) The objects are classified by area, indicated by the color of the contour points, demonstrating a simple means of classifying plankton.

III. MATERIALS AND METHODS

The following are the 3 syllabus components of the 1-semester Optical Engineering Laboratory course.

A. Image processing bootcamp

All students participate in a bootcamp to learn fundamental skills in image processing. The bootcamp consists of interactive Python scripts to teach Python and OpenCV library skills necessary for image processing written in Jupyter scripts (https://github.com/CCCofficial/Python_Image_Processing_Bootcamp).

(a) Introduction to Python (all students)
(b) Working with images (all students)
(c) Detecting objects in images (all students)
(d) Tracking detected objects in videos (all students)
(e) Extracting features from objects (optional, based on research project needs)
(f) Classifying objects by feature (optional, based on research project needs)

The scripts provide step-by-step demonstrations of applying Python and OpenCV functions. Some include questions for students, with worked out answers in separate scripts.

B. Image processing pipeline

The image processing pipeline (Fig 2) is available for students whose research projects require tasks like feature extraction and machine learning classification. The image processing pipeline components cover image capture through object classification. Learning and using the Detect and Track component code is mandatory for all students. Students can watch instructional videos and review code on any other components they need for their research project. The videos are from a 1-semester Image Processing Pipeline lecture course we previously taught and recorded with Zoom during the COVID-19 pandemic, edited into 5–10 min segments and organized into 3 YouTube Playlists. For the Optical Engineering Laboratory course, we made the Detect and Track pipeline components mandatory and the remaining components optional according to the needs of the student's research project.

(a) Image Processing with Python and OpenCV (37–45).

An introduction to image processing for biologists. Provides an overview of detection, tracking, feature extraction, unsupervised and supervised classification, and the use of a confusion matrix to investigate classification performance.

(b) Detecting Objects Using Python and OpenCV (46–51). How to use Python and OpenCV functions to detect and track objects in a video. Explains how images are stored in data arrays, how to convert color images into binary images, and how to detect objects by their contour.
(c) Unsupervised Clustering with Python and Scikit-learn (52–58). How *k*-means clustering

Table 2. Repository of code for coursework and student research projects.

Repository and code	URL and function
DetectTrack	https://github.com/CCCofficial/DetectTrack
Detect.py	Detects objects by removing background with a median filter, blur to remove holes in objects, binary quantize image with threshold, FindContour to detect objects. Calls Track.py to track objects and saves results in a CSV file.
Track.py	Tracks objects by pairing up object in current video frame with nearest object in previous frame. Assigns new IDs to new objects that appear in a frame.
detectImageGUI.py	GUI to detect objects in images in a directory.
detectVideoGUI.py	GUI to detect objects in frames of a video.
blep_14sec.mp4	Short video taken with a lensless microscope of multiple moving *Blepharisma* to test, detect, and track programs.
Holo1	https://github.com/CCCofficial/Holo1
goldHolo.zip	Cropped holographic images of 48 plankton and 16 microfibers from dryer lint.
holoVideoReco.py	Interactive holographic reconstruction with Tkinter GUI to open a video and view frame-by-frame, selecting area to crop and reconstruct.
darkPixReco.py	Detects plankton in video, optimized for detecting tiny plankton that produce fringes with low or no contrast center, making detection very difficult. Create composite image by selecting darkest pixel of several *Z* reconstructions.
Feature.py	Calculates shape, texture, grayscale histogram, local binary patterns, and several moment features of an object.
Cluster.py	Clusters objects in video blep1.mp4 into 5 clusters with extracted features.
3D_Cluster_Plot.py	Displays scatterplot of area, texture, and aspect ratio.
ConfusionMatrix.py	Displays normalized confusion matrix and accuracy calculation.
ViewFeatures.py	Displays scatterplot of object features while viewing video.
blep1.mp4	Video used to test clustering.
piClassifier.pdf	Evaluates 13 classifiers on 9 classes of plankton running on Raspberry Pi 3.
Holo2	https://github.com/CCCofficial/Holo2
findZ.py	Loads images from a directory and uses keyboard to adjust reconstruction *Z* and save reconstructed image with *Z* value embedded in file name.
findZ_histogram.py	The findZ.py program with a histogram plot of pixel intensity.
holoVideoReco.py	Reconstructs holographic images from MP4 videos.
examineReco.py	Demonstrates how reconstruction works.
streamRecoLine.py	Interactive holographic reconstruction with line contrast display.
reco.py	Performs holographic reconstruction.
detectBlur.py	Detects program with blur to prevent multiple bounding boxes.
HSV_colorSpace.py	Represents and manipulates images in HSV color space.
bouncingSquare.py	Creates and manipulates images in NumPy arrays.
playChords.py	Plays notes and arpeggiated chords with pyGame and MIDI files.
keyboard.py	Changes variables with keyboard while program is running.
USB_CAM.docx	How to convert Raspberry Pi into a USB camera.

is performed to classify objects automatically, including a review of working code.

The slides used to teach image processing components and holographic reconstruction referenced in Table 1 are publicly available in GitHub repositories (59). The Python code used to teach and implement image the processing of pipeline components, holographic reconstruction, and other techniques to support student research projects is summarized in Table 2. The table includes links to public GitHub repositories that contain the listed code. We shall now discuss how these materials are used in the course syllabus.

Referring to Table 2, the DetectTrack repository contains Python code to detect and track multiple objects in a video, including 2 GUI programs written by a student (co-author Salma Ahmed) to select and save objects from images and videos. A short video of plankton (blep_14sec.mp4), captured with a lensless microscope with a single blue LED located 100 mm from the image sensor, is included for testing. The light source is not coherent enough to produce an interference pattern, so reconstruction is not required, albeit the resolution is lower than the holographic microscope.

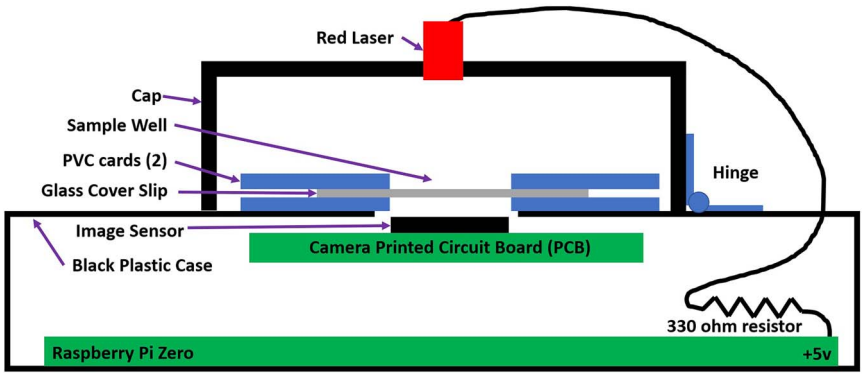

Fig 3. Cross section of holographic microscope (not drawn to scale). A sample well is created by gluing a round glass slide between 2 sheets of PVC plastic with a 3/4-inch (2 cm) hole punched in the center. The sample holder is placed on top of the plastic case so it can be removed and cleaned, along with the image sensor below, if needed.

The Holo1 repository contains a collection of cropped holographic images of plankton and microfibers (goldHolo.zip) and Python code to perform reconstruction and feature extraction. A paper is included (piClassifer.pdf) that evaluates the feature set performance and run time of 13 classifiers on a Raspberry Pi 3 computer. One student research project (Plankton Classification) combined elements of the Feature.py and Cluster.py programs to perform unsupervised classification. Objects can be at different *Z* planes, so each must be reconstructed separately. Detecting and tracking holograms of objects, particularly small objects, is often difficult because the fringes spread out the object, reducing the contrast. To address this problem, the program darkestPixelVideo_5.py in the Holo1 repository reconstructs an entire video frame at several *Z* planes, then selects the darkest pixel from the set of reconstructed *Z* planes, darkening every object, making them easier to detect.

The Holo2 repository contains Python code and files of holographic video and images and of reconstructed images. Instructions are provided to convert the Raspberry Pi into a USB microscope (USB_CAM.docx), eliminating the need for a video monitor, keyboard, mouse, and power supply. Programs are provided to reconstruct holographic images (findZ.py) and video frames (holoVideoReco.py) and demonstrate how reconstruction works (examineReco.py and phaseAbbScan.py). Several programs are useful utilities and functions (renameFiles.py, keyboard.py, and reco.py). Two programs play notes and chords, created to support a student's research project to sonify plankton by size and location.

C. Holographic microscope

Unlike conventional microscopes that require critical alignment of multiple optical components (60), the holographic microscope has only 2 components: an inexpensive laser that is hand aligned above an image sensor (Fig 3). Focusing is performed mathematically by using Fourier transforms (61) applied at the height of the object (*Z*) over the image sensor (Fig 4). The holographic microscope is based on Dennis Gabor's design published in 1948 (62), wherein he describes how it is possible to eliminate an objective lens and record on photographic film the amplitude and phases of an interference pattern produced by a small object in a field of coherent electromagnetic waves. Gabor used the combination of a mercury arc lamp, narrow-band green filter, and pinhole to create a coherent light source. In our version, we use a red laser (635 nm) and image sensor accessed by removing the lens of a Pi V1 camera (Raspberry Pi Ltd., https://www.raspberrypi.com/documentation/accessories/camera.html), to capture the interference pattern (Fig 4b). However, the red laser is too bright and coherent, saturating the image sensor and filling the image with speckle, a random interference pattern produced by coherent light diffusely reflecting off surfaces. To address this, we use a resistor (330 ohms; Fig 3) to limit the current below the lasing threshold current, so it operates not as a laser, but as a tiny ($\sim$5 μm $\times$ $\sim$100 μm) LED (63). The small size approximates a point light source, emitting continuous light that is acceptable to the image sensor and spatially

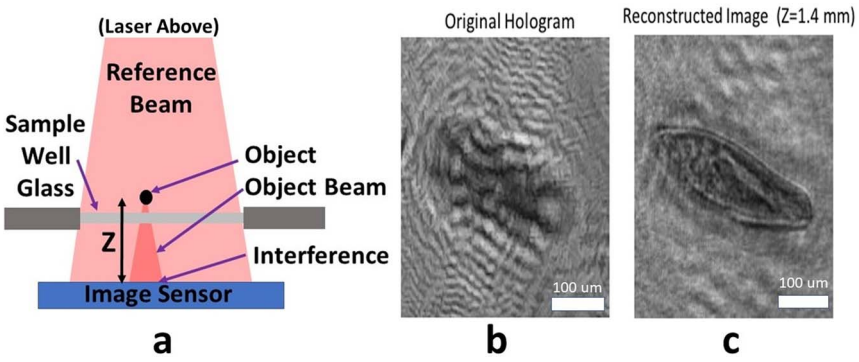

Fig 4. Lensless holographic microscope. (a) Most of the coherent light from the red laser reaches the image sensor without encountering any objects (reference beam), while some light is diffracted by the objects (object beam) floating on the glass in the sample well. (b) An interference pattern (hologram) is formed at the image sensor where these 2 beams meet, causing characteristic fringes around the object. (c) The hologram is reconstructed at the plane of the object (*Z*, the object-to-image sensor distance), found iteratively by reconstructing over a sequence of *Z* values and manually selecting the image most in focus.

coherent enough to produce an interference pattern without generating unwanted speckle.

Most of the parts are sourced directly from China (Table 3) and are very inexpensive but can take up to 1 mo to arrive in the United States. The following common accessories are needed to construct the microscope: hot glue gun, soldering iron, diagonal cutters, wire strippers, utility knife, Dremel tool, painter's tape, hand drill, wire, safety googles, and aquarium-safe silicone sealer. A mini-HDMI to HDMI cable is needed to connect the microscope to an external monitor. A USB mouse and keyboard are also needed to control the Pi computer.

The Optical Engineering Laboratory course was offered 4 times between 2019 and 2022: 3 times in-person and 1 time remotely because of Covid-19. For clarity, we will separately describe these 2 modalities.

D. Microscope construction in the classroom

To optimize class time for construction and research, students reviewed video instructions at home that demonstrate the entire build process (64–67). In class, each student was provided a kit containing the components listed in Table 3 and access to a hand drill (e.g., a Dremel tool), soldering iron, hot glue gun, black silicone sealer, safety glasses, and a few hand tools. The microscope was constructed in a plastic box, oriented upside down, with the thin lid on the bottom to reduce water penetration and simplify construction (Fig 5, right). A sample holder was fabricated by drilling a 3/4-inch

Table 3. Microscope parts list.

Item	Cost (US$)	Supplier
Raspberry Pi Zero W	10	adafruit.com
Raspberry Pi version 1 camera module	4	aliexpress.com
Adjustable laser dot diode module, 650 nm, 6 mm, 5 V, 5 mW	0.20	aliexpress.com
Raspberry Pi power supply, 5 V, 3 A (micro-USB)	6	robotshop.com
USB mini hub with power switch—OTG (micro-USB)	5	adafruit.com
SanDisk Ultra 32 GB microSD	5	aliexpress.com
Black plastic furniture leg plug end caps insert, 35 mm	0.11	aliexpress.com
Mini hinge, 10 × 8 mm	0.12	aliexpress.com
Black plastic electronic project box, 100 × 60 × 25 mm	0.67	aliexpress.com
White blank PVC card, standard size = 85.6 × 54 × 0.8 mm	0.27	aliexpress.com
Round microscope glass slide coverslip, 25 mm	0.04	aliexpress.com
Four nylon machine screws and nuts, 4 × 40 1/4-inch (0.6 cm) length	0.08	aliexpress.com
Resistor (limits laser current), 330 ohm 1/4 or 1/8 W	0.02	aliexpress.com

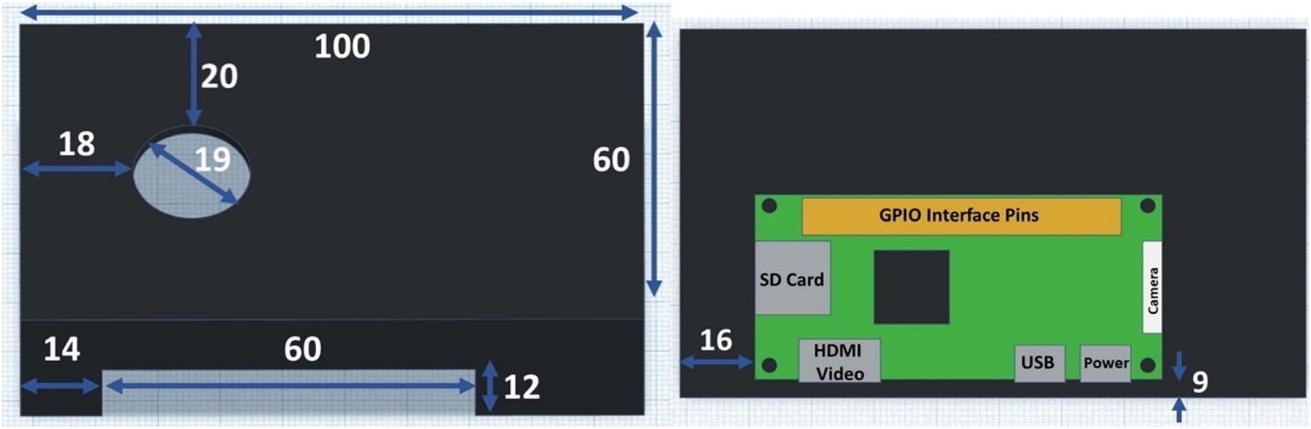

Fig 5. Microscope box dimensions (in mm). (Left) Case top with a drilled hole for the image sensor and case side with a slot created with a Dremel tool for Raspberry Pi connectors. (Right) Raspberry Pi mounted to bottom of case with 4 × 40 nylon screws.

(2 cm) hole in 2 polyvinyl chloride (PVC) cards and gluing a 1-inch-diameter (2.5 cm) glass coverslip between the 2 cards with aquarium-safe silicone sealer (Fig 6d), which takes 24 h to dry. This construction created a well to hold a liquid sample. A 3/4-inch (2 cm) hole was drilled in the top of the inverted plastic box, and the sample well was glued above the hole, after determining that the underside of the glass was perfectly clean, determined by operating the microscope. The molded lens of the camera was carefully removed with diagonal cutters to expose the image sensor. A notch was cut in the side of the box with the Dremel tool to provide access to connectors on the side of the Raspberry Pi computer. An SD card containing the free, open source, Linux-based Raspberry Pi OS (version 5.10, Raspberry Pi Foundation, Cambridge, UK) was inserted into the Raspberry Pi, and the Pi board was attached to the plastic case lid with 4 M2.5 (metric) nylon screws. The laser collimating lens was removed and inserted into a 1/8-inch (0.3 cm) hole drilled in the center of a 35-mm square black plastic end cap. The lens was removed to reduce the light intensity that would otherwise saturate the image sensor.

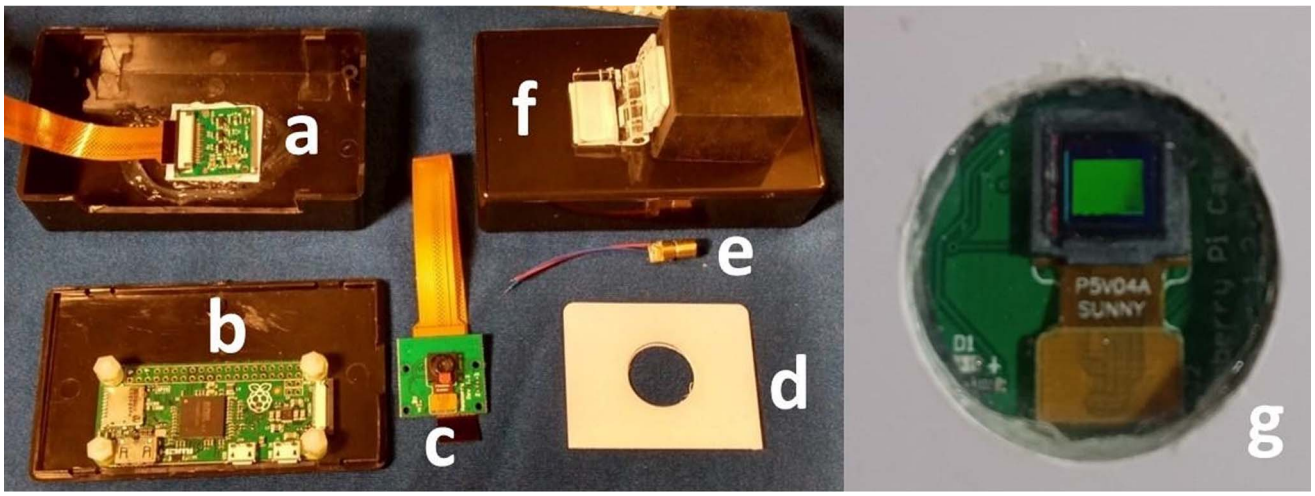

Fig 6. Components of the student-built holographic microscope. (a) Image sensor board attached to the underside of the plastic case with glue or double-stick foam tape (preferred for easy removal, if necessary). (b) Raspberry Pi Zero attached to the bottom of the plastic case with 4 × 40 nylon (nonconductive) screws and nuts. (c) Camera before the lens is removed. (d) Sample well comprising 1-inch-diameter (2.5 cm) glass cover slip glued between 2 PVC cards (cut in half) with 3/4-inch-diameter (2 cm) hole. (e) Red laser before the lens is removed. (f) Assembled microscope with the square lid hinged to the case, attached with double-stick tape. (g) Closeup of the sample well mounted on top of the plastic case and above the image sensor.

Removing the lens also spreads the beam about 18°, simplifying optical alignment with the image sensor. Wires from the laser were passed through a 1/16-inch (1.6 mm) hole in the top of the case and soldered to the 5V and ground pads of the Pi, with a 330-ohm resistor in series to limit current, providing continuous power to the laser. The current was below the lasing threshold (63), operating the laser as a point-light source LED, because lasing would produce too much speckle and brightness. In the final alignment step, the student would turn on the system and slide the cap until a bright uniformly illuminated image was observed on the monitor, then fix the cap location to the case with double-stick foam tape. The assembled microscope is displayed in Figure 6f.

To capture holograms, a sample was placed in the sample well (Fig 6d), the cap was closed, and a 30-s video was recorded by the raspivid utility included in the Raspberry Pi OS (68). Students transferred holographic videos to their laptop with a USB flash drive. The Raspberry Pi–native h.264 video format was converted to MP4 format by the FFmpeg utility (version N-67100-g6dc99fd, FFmpeg team; https://ffmpeg.org) to enable random frame access under program control. We wrote an application for the students to view the video frame-by-frame, select a plankton of interest, then manually step through reconstruction distance values (Z) to select and save the best looking (in focus) image (findZ.py; Table 2). The Raspberry Pi can run the Python reconstruction software, but a laptop is much faster, has more resources (e.g., memory and hard drive), and provides a more familiar programming environment.

E. Microscope construction at home

Because of the COVID-19 lockdown of the San Francisco State University campus in Fall 2020, students had to assemble their microscopes at home. Mailing each of the 22 students the set of tools described in the previous paragraph was financially impractical and raised safety issues (e.g., hot soldering iron and sharp tools) because they would not be used under faculty supervision. To compensate, we prebuilt microscope components, performing the necessary drilling and soldering that required tools, and mailed each student a kit with double-stick tape and silicone sealer to finish the assembly and perform the laser-to-image sensor alignment. Assembly consisted of gluing a glass slide between 2 PVC cards with a 3/4-inch (2 cm) center hole that acts as a sample well, gluing it to the top of the predrilled case, then adjusting the cap for the best laser-to-image sensor alignment.

Instead of relying on the apps raspistill and raspivid (69), provided in the Raspberry Pi OS distribution to capture images and video, respectively, which would require a separate USB keyboard, mouse, power supply, and HDMI monitor, we provided instructions on how to install showmewebcam (70), which is an alternative way to collect images and video. Students download an open source disk-image (collection of files) onto an SD card and insert it in the microscope's Pi single-board computer. The disk-image contains the minimum necessary Raspberry Pi OS functionality to run an application that turns the Pi's camera into a USB video device. When plugged into a laptop's USB port, the microscope boots in 3 s and operates as a web camera, allowing any web camera software to be used, including streaming holographic videos over a Zoom call. Additionally, the code on the SD card includes a lightweight application to control camera parameters, including exposure and contrast, essential for producing holographic images with many fringes to produce good reconstructions.

IV. RESULTS

After successfully building their personal microscope, all students used samples of their hair to practice capturing good-quality holograms and perform measurements. Two strands of hair were mounted to a glass microscope slide with clear nail polish in an "X" pattern (Fig 7a). The sharpness of the hair intersection provides a visual guide to select the best sample-to-image sensor reconstruction Z distance. The resulting hologram was reconstructed, and the pixel brightness of a cross section of the hair (Fig 7b) was plotted with ImageJ to determine the hair diameter (Fig 7c).

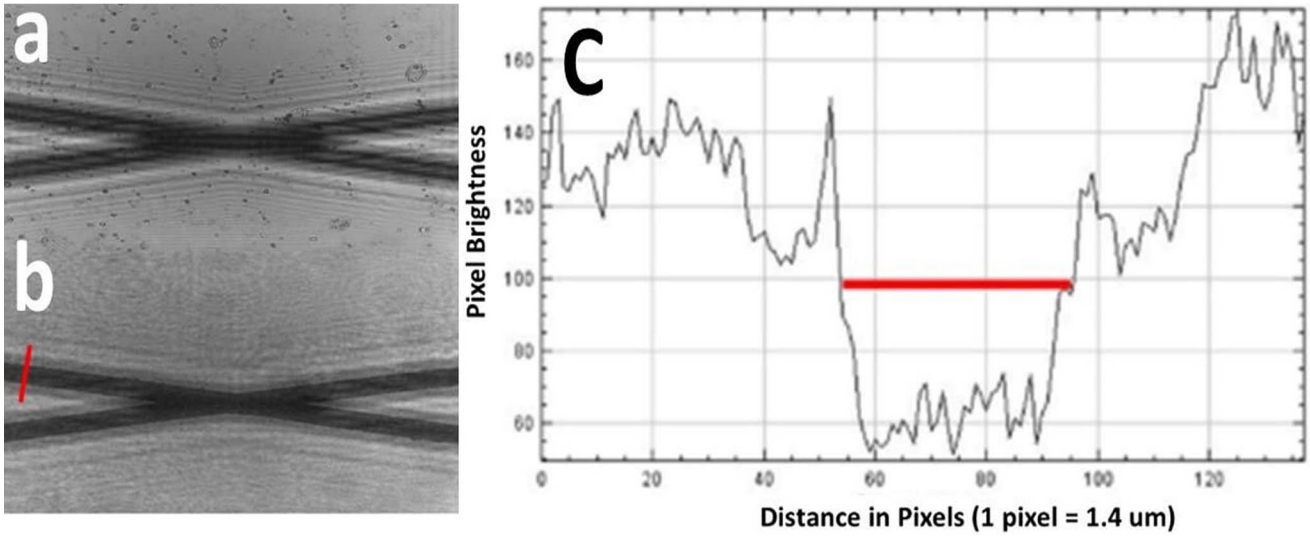

Fig 7. Measuring the diameter of a human hair. (a) Raw hologram of 2 strands of hair arranged in an "X" pattern. (b) Reconstruction was achieved when a *Z* value is chosen such that the intersection of the strands becomes well defined. (c) The pixel intensity of the hair's cross section (red line) was plotted. The diameter was determined by counting the number of pixels in the half-amplitude interval (red line) and multiplying by the pixel size (1.4 μm). In this example, the interval was 41 (95 − 54) resulting in a hair diameter of 57.4 μm.

Once they successfully measured hair diameter, each student developed a project driven by their own questions, in which they measured the microscope's performance, modified microscope hardware or software, or applied the microscope as a tool to answer their research question. In subsequent terms of teaching the course, we allowed students to use other microscopes and image datasets from the public domain, research labs they worked in, or our Center for Cellular Construction partner labs. We shall now share several research projects to demonstrate the variety of work the Optical Engineering Laboratory students conducted.

A. Optimizing holographic microscope resolution

Two students (J. Luo, Z. Dean) examined the resolution of the holographic microscope and what improvements could be made to increase performance. The project aimed to contribute to a larger class goal of identifying cost-effective approaches to enhance the performance of the microscope. Their hypothesis was that shorter wavelength and use of a pinhole would improve resolution from previous work by Amann *et al*. (71). In the cited work, the authors built and compared holographic microscopes with a blue laser (405 nm) coupled into a single-mode fiber and a blue LED (430 nm) filtered with a 25-μm pinhole, resulting in a lateral spatial resolution of 1.55 μm and 3.91 μm, respectively, determined with a 1951 USAF resolution test target slide (R1DS1P, Thorlabs, Newton, NJ). With the goal of maintaining the low cost and construction simplicity of our holographic microscope design, the students tested 3 configurations and compared their resolution results to Amann's work.

With a USAF test slide to measure resolution, the students tested a blue laser (405 nm), with and without a 25-μm pinhole (P25k, Thorlabs, Newton, NJ), and the red laser (650 nm) used in the student microscope. They varied the object-to-image sensor distance to achieve the maximum resolution, fixing the laser-to-image sensor to 25 mm. They achieved a 2.76-μm resolution with the blue laser and the pinhole, 4.38-μm resolution with just the blue laser, and 4.92-μm resolution with the red laser (the original student microscope design), verifying their hypothesis (Fig 8). Considering the student microscope did not use a single-mode fiber or pinhole to increase the coherence, both of which would require careful alignment and additional expense, achieving 4.92-μm resolution was an impressive result.

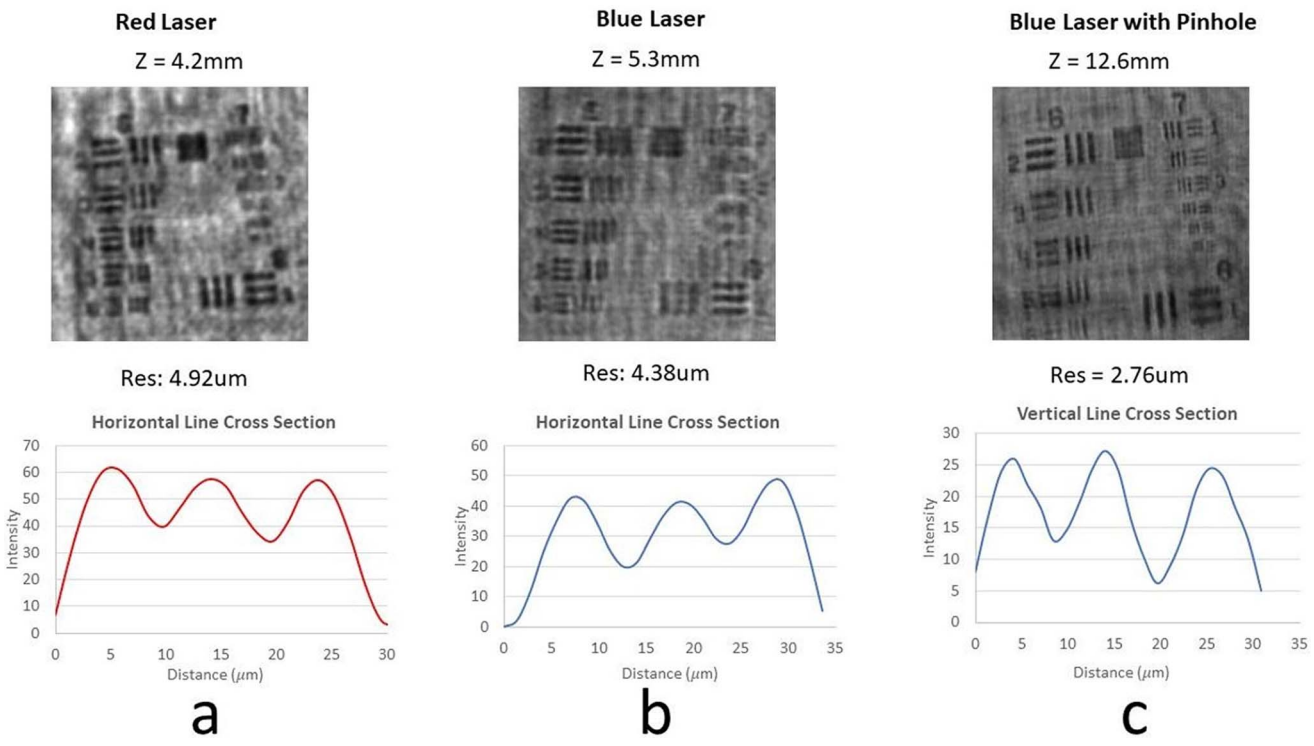

Fig 8. Holographic microscope resolution under 3 conditions: (a) 4.92-μm resolution with a red laser and sample-to-image sensor distance of 4.2 mm, (b) 4.38-μm resolution with a blue laser and sample-to-image sensor distance of 5.3 mm, and (c) 2.76-μm resolution with a blue laser, 15-μm pinhole, and sample-to-image sensor distance of 12.6 mm. The top row consists of the reconstructed target images. The bottom row consists of the cross-section intensity profiles used to verify whether line contrast was sufficient to resolve the target lines. Images are of a 1951 USAF resolution test slide.

B. Automatic sample loader

The goal of this project was to automate the loading of samples into the microscope's viewing chamber, for example, to monitor the plankton in a fish tank or from a lake. The student's (D. Ruiz) hypothesis was that a single electric pump under computer control could clear out old samples and load new samples for observation. The system, illustrated in Figure 9, operated as follows: A MOSFET transistor (72) enabled the microscope's Raspberry Pi to control a 12 V peristaltic pump (NKP-12V-S10B, Kamoer Fluid Tech Co., Shanghai, China) (Fig 9b).

The microscope turned on the pump for 30 s to draw water from a plankton reservoir (Fig 9a), delivering a new sample to a 3D-printed flow cell (green box in Fig 9c) while washing out the previous sample into a waste container. After a 5-s pause to allow the water in the flow cell to settle, a 30-s holographic video was recorded. The cycle was repeated often enough to prevent water from drying and fouling the sample well glass, verifying the student's hypothesis. This project was designed and built by a nonengineering biochemistry major and was their first experience with soldering, mechanical design, 3D printing, circuit design, and assembly. The student learned and used Fusion 360 (version V.2.0.6263, Autodesk Inc., San Francisco, CA) to design and 3D print the flow cell (Fig 9c).

C. Increasing the field of view

The viewing area of the lensless microscope is typically the size of the image sensor (3.76 × 2.74 mm). One student (R. Cisneros) investigated a method to increase the viewing area of the microscope using multiple light sources. The hypothesis was that a horizontal displacement of the light source would expose the edges of the viewing area, and by using multiple light sources under computer control, a mosaic of images could be collected, increasing the field of view. To simplify the design, an array of LEDs manufactured in an epoxy case were used

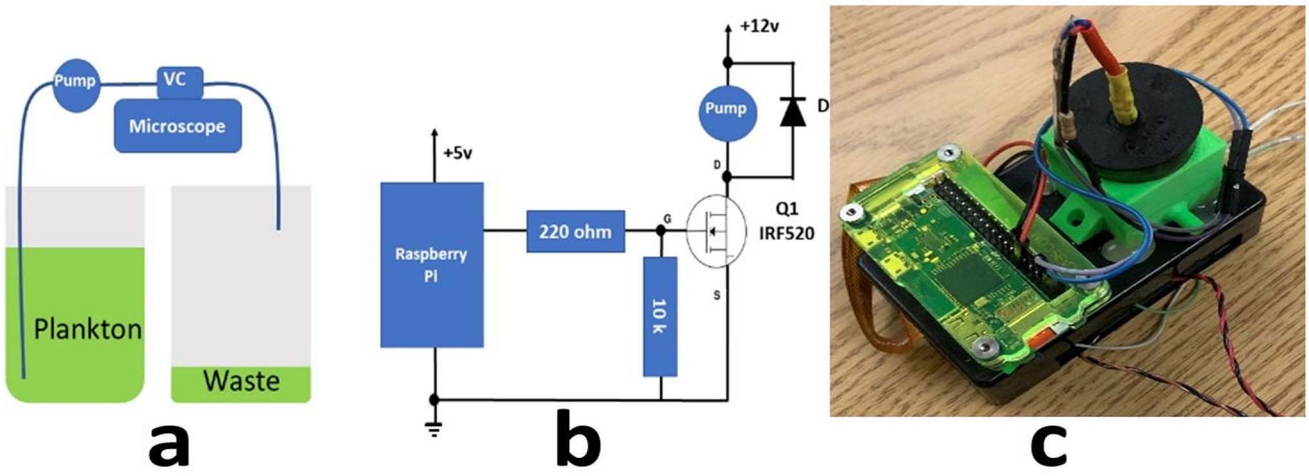

Fig 9. Automatic plankton sampler. (a) Plankton was pumped from a container, simulating a lake, into the holographic microscope, with excess water spilling into a waste container. (b) The microscope's Raspberry Pi energized a MOSFET transistor (IRF520) circuit periodically, washing away the previous sample while loading a new sample into the microscope. (c) A 3D-printed flow cell (green), mounted between the glass sample well and laser, includes an inlet and outlet port to receive and discharge pumped plankton samples.

instead of multiple lasers. An 8 × 8 array of red LEDs (Fig 10a), each with a 2-mm-diameter on a 2.5-mm pitch arranged in a plane (KWM-30881CVB, Adafruit, New York, NY), was mounted above the image sensor (Fig 10b,d). By sequentially energizing selected LEDs one at a time and capturing an image, the microscope's effective field of view was enlarged (Fig 10e), verifying the student's hypothesis. The student also demonstrated that increasing the distance between the LEDs and image sensor increased the image contrast (Fig 10c), because a distant LED better approximated a point light source.

D. 3D-printed cap

The original microscope design had a hard rubber cap (25-mm cube) to hold the red laser directly above the image sensor and block ambient light. The cap was hinged to the case to allow samples to be loaded into the sample well. The hypothesis was that by placing the sample at fixed distances from the sensor, discrete magnifications could be selected. To test this hypothesis the student (C. Guzman) learned Fusion 360 and used the program to design and 3D print a cap with multiple pairs of slots to accept conventional 1 × 3 inch (2.5 × 7.6 cm) microscope slides (Fig 11). Because the size of the interference pattern increases with the distance between the sample and image sensor, multiple pairs of slots provided a set of discrete magnifications. The student tested the cap with a variety of samples, demonstrating discrete magnifications and proving his hypothesis.

E. Variable magnification

The previous project provided slots to select discrete magnifications. Two students (C. Ly, A. Martin) took different approaches to build an apparatus to adjust magnification continuously. The hypothesis was that by continually varying the sample-to-image sensor distance, a continuous zooming feature could be achieved. The first student constructed an apparatus at home (because of COVID-19) with material available around the house (Fig 12a,b). A cardboard frame held a pulley made from a plastic straw and bottle cap. Four strings suspended the image sensor below the fixed sample, continually changing the image-to-sample distance.

The second student built an apparatus at school with a 3D design provided online from the Universidad de Colombia (73) (Fig 12c). The 3D-printed structure used a threaded bolt to move the image sensor above the sample, with the laser fixed below the sample. The student modified the design to accommodate the Raspberry Pi camera and red laser. Both designs verified the hypothesis that a variable zoom

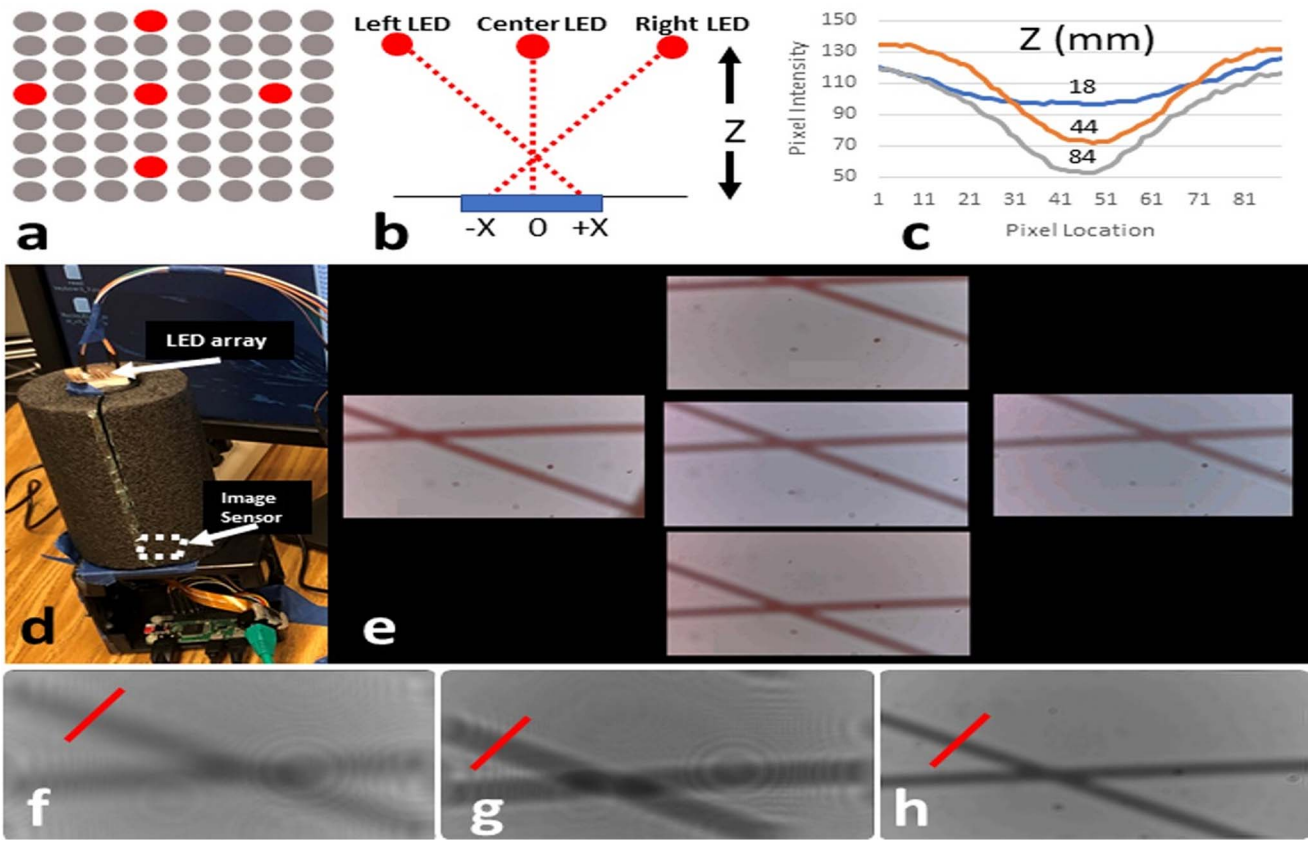

Fig 10. Increasing the microscope's field of view. (a) An 8 × 8 matrix of red LEDs was (d) mounted on top of a foam tube above the image sensor. A slide containing a pair of crossed human hairs was mounted on glass and placed directly above the image sensor. (a, b) Five LEDs (red colored) were sequentially energized to produce (e) 1 centered and 4 off-centered images, increasing the viewing area. (c) The resolution of the image produced at 3 different sample-to-LED (*Z*) distances (blue = 18 mm, orange = 44 mm, gray = 84 mm) was measured by plotting the pixel intensity of hair cross sections (red line) of the images produced at progressively greater *Z* distances (f, g, h, respectively), demonstrating that image contrast increases with *Z* distance because distant LEDs better approximate a point light source.

could be implemented by mechanisms that moved the sample-to-image sensor distance in a continuous motion.

F. User interface design

We provided students image processing code to reconstruct holographic images and detect objects with a simple keyboard interface to change program variables. One class lecture demonstrated how to use TKinter (version 8.6, Python Software Foundation), a Python GUI library to design a mouse-driven GUI. Inspired by this exercise, the student's (S. M. Ahmed) hypothesis was that a GUI would improve the user experience of the reconstruction code. One result was an application (Fig 13, left) that allowed students to use sliders to vary the quantizing threshold, minimum and maximum area of detected objects, and label and save the objects. Another application (Fig 13, right) enabled students to use sliders to view their holographic videos frame-by-frame and modify detection parameters. The student's GUIs were adopted by the class as the preferred method of processing holographic images, proving the hypothesis.

G. Plankton classification

Two students (E. Samperio, A. Barrera-Velasquez) teamed up to classify plankton. One student created a labeled training set of 11 plankton classes (131020, Carolina Biological Supply, Burlington, NC) imaged by the entire class with their holographic microscopes by cropping and reconstructing 360 images from videos recorded by all class members. The other student performed object segmentation (Fig 14a–d), feature extraction, and *k*-means

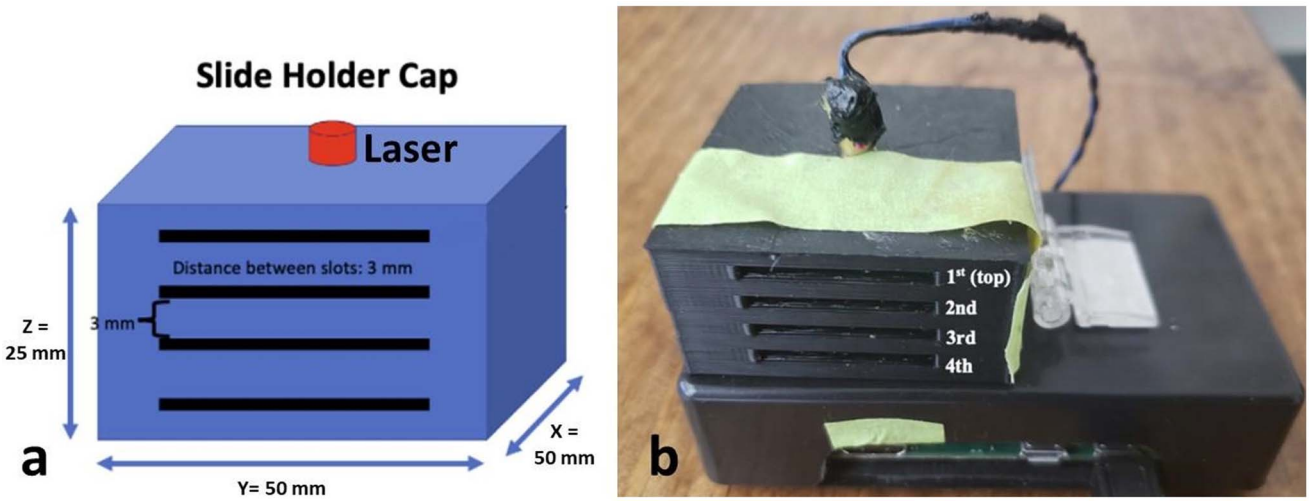

Fig 11. Microscope with variable magnification. (a) A 3D-printed cap with 40-mm-wide × 2-mm-high slots to receive standard glass microscope slides. A matching set of slots on the opposite side allowed a slide to be passed through the cap in the *X* direction. Because the magnification increases with the sample-to-image sensor distance, the slots allowed the user to select discrete magnifications. (b) Modified cap attached to the holographic microscope.

clustering (Scikit-learn) by modifying code from the image processing pipeline.

The hypothesis was that the plankton classes could be classified by area, aspect ratio, and circularity. The segmentation process separated the object (Fig 14a) from the background by blurring the image (Fig 14b) to remove object details, which might otherwise make one object look like several smaller objects. A quantization threshold was applied to the blurred image and the OpenCV contour function determined the object contour (Fig 14c,d). The contours were used to calculate the geometric features of area, aspect ratio, and circularity. These 3 features were used to cluster the objects into 11 classes (Fig 14e) by the *k*-means clustering algorithm. Although the results did show that the classes presented sensitivity to the features, the combination of the 3 features were not sufficient for classification, disproving the hypothesis.

H. Laundry microfibers

A group of students (A. Nelson, J. Suarez, G. Alarcon-Cruz) concerned about the dangers of microfibers, primarily from clothing, in waterways designed a method to distinguish cotton from synthetic microfibers. Their hypothesis was that the holographic microscope could be used as a low-cost tool to monitor microplastic pollution in the environment. They collected lint from several clothing dryers, separated out the tiny fibers, suspended them in a dilute solution with water, then captured images with their holographic microscope. They used ImageJ to measure the path and endpoint lengths, and used the ratio to distinguish the 2 types of microfibers (Fig 15). They determined

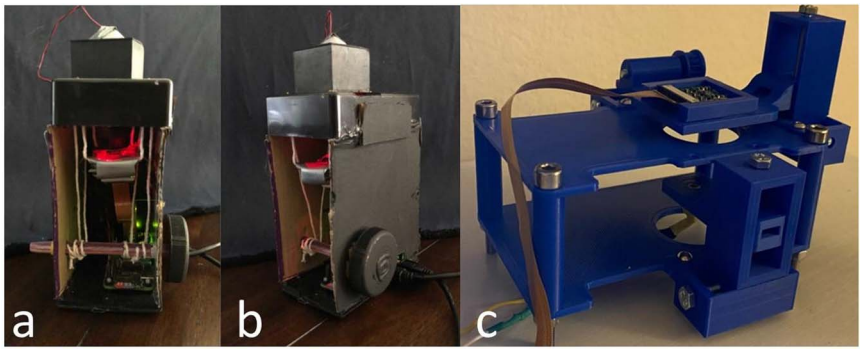

Fig 12. Microscopes to adjust magnification continuously. (a, b) An apparatus made at home from cardboard, plastic straw, string, bottle cap, and hot glue. (c) A 3D-printed apparatus based on a design from the Universidad de Colombia (36).

Fig 13. GUI for pipeline functions. (Left) Application to adjust detection parameters and label and save cropped images of objects. (Right) Application to adjust detection parameters to track selected objects in a video.

that cotton fibers had greater curvature than the synthetic fibers and can be observed with the low-cost holographic microscope, proving their hypothesis.

V. DISCUSSION

Our objectives in designing and teaching the Optical Engineering Laboratory course are to introduce biology and biochemistry students to image processing, instrumentation, and research methods and skills. The software side has 3 levels of achievement: knowing the basics of how to program in Python, learning how to use OpenCV to perform image processing tasks, and composing a program that solves a problem based on this acquired knowledge. The hardware side has similar levels of achievement: getting to know how to use basic tools, applying them to build the microscope, and using them to conduct original research. Most importantly, to conduct research, students need to learn how to define a project that is interesting scientifically and challenging but narrow enough in scope to be completed in one semester. Lastly, communication and collaboration skills are essential for future scientific careers. Students learn how to present results in a variety of formats and collaborate with faculty, fellow students, and researchers outside the classroom. In this section we discuss our impressions of the effect of the course on these goals and share students' impressions, based on their responses to pre- and post-class surveys.

For the Optical Engineering Laboratory course, we were pleasantly surprised how fast students could construct the microscopes and how much they enjoyed doing hands-on work. For many, it was the first time they used a soldering iron and Dremel tool. We received

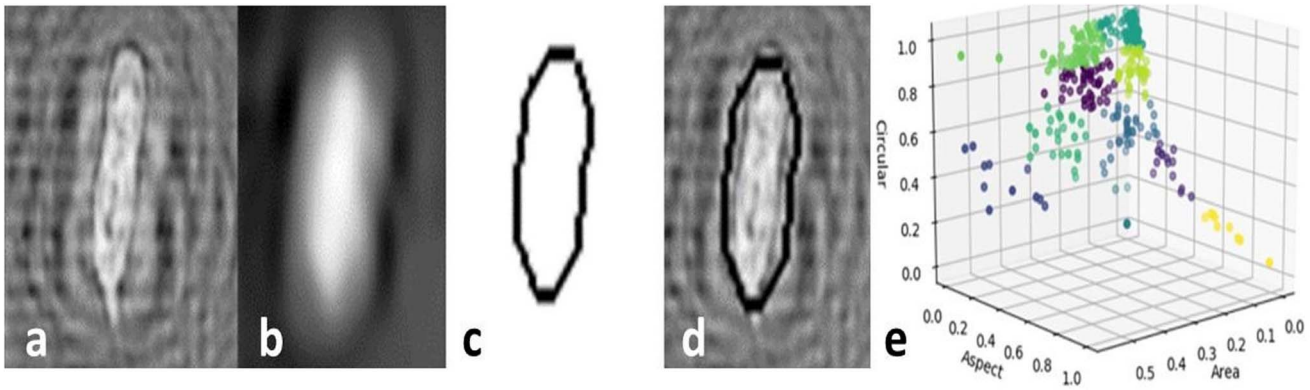

Fig 14. Plankton classification. (a) A reconstructed plankton image (b) was blurred and binary quantized (c, d) to detect the plankton contour. (e) Geometric features of area, aspect ratio, and circularity were extracted from the contour of the plankton and clustered into 11 classes, indicated by color, by the *k*-means clustering algorithm.

Fig 15. (Left) Cotton ball fibers are observed to have greater curvature and more kinks than (right) polyester fibers.

encouraging comments like, "I never knew I liked engineering," "I really liked soldering," and "I never knew I could do engineering." We were particularly impressed by the ingenuity and resourcefulness of the student who built an apparatus for continuously adjusting magnification out of household items. This mirrors our own experience of adapting existing equipment and materials to perform novel experiments in our professional research.

The big challenge in the lab was maintaining student's progress in their research projects. To address this, each week students would stand before the class and give a progress report. We began taking on the role of project managers, identifying critical-path tasks and experiments and setting these as deliverable goals for the next week. We also had to teach students how to present novel research in oral and written presentations. This task is more difficult than writing up the results of an assigned lab experiment, in which procedures are defined and the results are well known. To address this challenge, we devoted a class session in which each student would read and review a research paper published on bioRχiv (74). This activity helped students see how figures, tables, and other parts of a paper must be clear to the reader and provide all the essential elements of the research, so the results can be reproduced by others.

VI. CONCLUSION

The holographic microscope provides a platform for hands-on learning of several scientific and engineering disciplines. These include physics, mathematics, electrical and mechanical engineering, signal processing, and computer science. In academia, these disciplines are often taught independently in separate departments and different majors. The holographic microscope provides an affordable, accessible, and extendable platform to study and apply all these disciplines in one project-based interdisciplinary class. For many of the biology and biochemistry majors this was their first exposure to engineering and hands-on equipment building, and many were surprised to find how much they enjoyed engineering.

The image processing pipeline provides a flexible structure for inclusion in a syllabus. Each pipeline component can be taught individually because they had defined inputs and outputs and were used in sequence to perform a complex function (i.e., classify objects). This modular feature made it possible to select a few components and integrate them into the Optical Engineering Laboratory course. Many students used individual components in their research projects. Providing working code demonstrated how to take a complex program and break it into functions. The modules taught Python programming skills and introduced students to the power of the OpenCV computer vision library and Scikit-learn machine learning library. An ideal 1-yr curriculum would be a semester-long Image Processing lecture course followed by the Optical Engineering Laboratory course. This combination would provide enough time for students first to learn the theory and gain coding experience with the entire pipeline in the Image Processing lecture course, followed by learning and applying engineering and research skills in the Optical Engineering Laboratory course.

In research, especially in biology and biophysics, we are often faced with buying expensive equipment, or more often modifying and adapting equipment we have, and sometimes building it from scratch. Research labs need graduates who are versed in more than one field,

who have depth (the traditional product of education) but also breadth, basic knowledge, and skills in complementary disciplines. The multidisciplinary laboratory and project-based course we developed, supported by lessons in programming and image processing, provided opportunities for students to broaden their knowledge, experience, abilities, and opportunities.

AUTHOR CONTRIBUTIONS

TZ, RE, and Y-HMC conceived and designed the courses. TZ and RE drafted the manuscript. TZ, RE, Y-HMC, NA, AA, JL, and SC edited the manuscript. TZ, RE, Y-HMC, AK, NA, SA, and AA were course instructors. JL and ZD performed microscope performance analysis. TZ and SA wrote code for the course. MP performed student evaluations. SB and SC provided project oversight. The student consortium comprised Gilbert Alarcon-Cruz, Adrian Barrera-Velasquez, Rocio Cisneros, Carlos Guzman, Cara Ly, Adrian Martin, Alicia Nelson, Donovan Ruiz, Emily Samperio, and Jessenia Suarez.

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation (DBI-1548297). We thank the students listed in the Acknowledgments for their participation, hard work, and enthusiasm to learn and discover. Disclaimer: Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors declare no competing interests.

REFERENCES

1. Randlett, O., C. L. Wee, E. A. Naumann, O. Nnaemeka, D. Schoppik, J. E. Fitzgerald, R. Portugues, A. M. B. Lacoste, C. Riegler, F. Engert, and A. Schier. 2015. Whole-brain activity mapping onto a zebrafish brain atlas. *Nat Methods* 12:1039–1046. https://doi.org/10.1038/nmeth.3581.
2. Henriques, R., C. Griffiths, E. Hesper Rego, and M. M. Mhlanga. 2011. PALM and STORM: unlocking live-cell super-resolution. *Biopolymers* 95:322–331. https://doi.org/10.1002/bip.21586.
3. Neiles, K. Y., and P. S. Mertz. 2020. Professional skills in chemistry and biochemistry curricula: a call to action, chap. 1. In ACS Symposium Series, vol 1365. Integrating Professional Skills into Undergraduate Chemistry Curricula. K. Y. Neiles, P. S. Mertz, and J. Fair, editors. American Chemical Society, Washington, DC, pp. 3–15. https://doi.org/10.1021/bk-2020-1365.ch001.
4. US Bureau of Labor Statistics. How to become a biochemist or biophysicist. Accessed 29 November 2022. https://www.bls.gov/ooh/life-physical-and-social-science/biochemists-and-biophysicists.htm#tab-4.
5. Whitcomb, C., and L. E. Whitcomb. 2013. Effective interpersonal and team communication skills for engineers. Wiley-IEEE Press. Online ISBN: 9781118514283. https://ieeexplore.ieee.org/book/6480475.
6. Boehm, B., and S. K. Mobasser. 2015. System thinking: educating T-shaped software engineers. In Proceedings of the IEEE/ACM 37th IEEE International Conference on Software Engineering, vol. 2. Florence, Italy, 16–24 May 2015. IEEE Computer Society, Los Alamitos, CA, pp. 333–342. https://doi.org/10.1109/ICSE.2015.166.
7. King, S. O. 2019. Producing "T-shaped" engineering graduates: the impact of student clubs as learning communities. In Proceedings of the 10th IEEE Global Engineering Education Conference (EDUCON 2019). A. K. Ashmawy, and S. Schreiter, editors. Dubai, UAE, 8–11 April 2019. IEEE, New York, NY, pp. 271–275. https://doi.org//10.1109/EDUCON.2019.8725241.
8. Boehm, B., and S. Koolmonojwong. 2019. Educating I-shaped computer science students to become T-shaped system engineers. *Procedia Comp Sci* 153:71–79. https://doi.org/10.1016/j.procs.2019.05.057.
9. Ryan, P., L. M. Lye, and A. Hsiao. 2011. When engineers become managers: learning from current engineering managers to advance engineering management education. In Proceedings of the 2nd Annual Canadian Engineering Education Association (CEEA) Conference. St. John's, Newfoundland, Canada. 6–8 June 2011. CEEA, St. Peters Bay, PE, Canada. https://doi.org://10.24908/pceea.v0i0.3612.
10. Schaffer, C. 2021. "Both/and" leadership: combining the benefits of I- and T-shaped leaders. Harvard Business Publishing. Accessed 22 November 2022. https://www.harvardbusiness.org/both-and-leadership-combining-the-benefits-of-i-and-t-shaped-leaders.
11. Gilbert, A., W. Tozer, and M. Westoby. 2017. Teamwork, soft skills, and research training. *Sci Life* 32:81–84. https://doi.org/10.1016/j.tree.2016.11.004.
12. LeJeune, N. 2003. Critical components for successful collaborative learning in CS1. *J Comput Sci Coll* 19:275–285.
13. Shurin, A., N. Davidovitch, and S. Shoval. 2021. The role of the capstone project in engineering education in the age of industry 4.0—a case study. *Eur Educ Res* 4:63–84. https://doi.org/10.31757/euer.414.
14. McLaughlin, J. Developing course-based undergraduate research experiences. Accessed 22 November 2022. https://urfm.psu.edu/mentors/developing-course-based-undergraduate-research-experiences.
15. Bangera, G., and S. E. Brownell. 2014. Course-based undergraduate research experiences can make scientific research more inclusive. *CBE Life Sci Educ* 13:602–606. https://doi.org/10.1187/cbe.14-06-0099.
16. Weaver, G., C. Russell, and D. Wink. 2008. Inquiry-based and research-based laboratory pedagogies in undergraduate science. *Nat Chem Biol* 4:577–580. https://doi.org/10.1038/nchembio1008-577.
17. Brownell, S. E., M. Kloser, T. Fukami, and R. J. Shavelson. 2012. Undergraduate biology lab courses: comparing the impact of traditionally based "cookbook" and authentic research-based courses on student lab experiences. *J Coll Sci Teach* 41:36–45.
18. Center for Cellular Construction, An NSF Science + Technology Center. Accessed 20 June 2023. https://centerforcellularconstruction.org.
19. Data Flair. Python libraries—Python standard library & list of important libraries. Accessed 14 December 2022. https://data-flair.training/blogs/python-libraries/.
20. Google Drive. Accessed 20 June 2023. https://www.google.com/drive
21. Zimmerman, T. 2023. CCCofficial/Python_Image_Processing_Bootcamp. Accessed 20 June 2023. https://github.com/CCCofficial/Python_Image_Processing_Bootcamp.
22. Hunter, J. D. 2007. Matplotlib: a 2D graphics environment. *Comput Sci Eng* 9:90–95.
23. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830.
24. Pastore, V. P., T. G. Zimmerman, S. K. Biswas, and S. Bianco. 2020. Annotation-free learning of plankton for classification and anomaly detection. *Sci Rep* 10:1–15.
25. Zimmerman, T. G., V. P. Pastore, S. K. Biswas, and S. Bianco. Embedded system to detect, track and classify plankton using a lensless video microscope. arXiv:2005.13064, https://doi.org/10.48550/arXiv.2005.13064 (preprint posted 26 May 2020).
26. Anaconda Inc. 2023. Accessed 20 June 2023. https://www.anaconda.com.
27. San Francisco State University. 2023. PINC Promoting Inclusivity in Computing. Accessed 20 June 2023. https://pinc.sfsu.edu/pinc.
28. Feisel, L. D., and A. J. Rosa. 2005. The role of the laboratory in undergraduate engineering education. *J Eng Educ* 94:121–130. https://doi.org/10.1002/j.2168-9830.2005.tb00833.x.
29. Sener, S., and A. Çokçalışkan. 2018. An investigation between multiple intelligences and learning styles. *J Educ Train Stud* 6:125–132. https://doi.org/10.11114/jets.v6i2.2643.
30. Justo, E., A. Delgado, C. Llorente-Cejudo, R. Aguilar, and J. Cabero-Almenara. 2022. The effectiveness of physical and virtual manipulatives on learning and motivation in structural engineering. *J Eng Educ* 111:813–851. https://doi.org/10.1002/jee.20482.

31. Perry, M. J. 2019. Chart of the day: female shares of BA degrees by majors, 1971 to 2017. Accessed 29 November 2022. https://www.aei.org/carpe-diem/chart-of-the-day-female-shares-of-ba-degrees-by-major-1971-to-2017.
32. Blosser, E. 2021. Hack along with Goldieblox: gender messages in engineering YouTube videos for girls. *J Gend Stud* 32:427–440. https://doi.org/10.1080/09589236.2021.1995341.
33. Eccles, J. S. 2015. Gendered socialization of STEM interests in the family. *Int J Gend Sci Technol* 7:116–132.
34. Pillonton, E. 2020. To remake the world, give girls all the (power) tools. Accessed 22 November 2022. https://www.fastcompany.com/90512278/to-remake-the-world-give-girls-all-the-power-tools.
35. Ramamurthy, A. 2019. Installing Python and Anaconda. Accessed 20 June 2023. https://saas.berkeley.edu/education/installing-python-and-anaconda.
36. Stack Overflow. 2023. All Questions. Accessed 20 June 2023. https://stackoverflow.com./questions.
37. Zimmerman, T. 2023. 1 Monitoring the Health of the Biosphere. Accessed 14 May 2023. Video, 5:44. https://youtu.be/cN3_CuqOAv8.
38. Zimmerman, T. 2023. 2 Detection. Accessed 14 May 2023. Video, 1:28. https://youtu.be/o3ilxsVu_b0.
39. Zimmerman, T. 2023. 3 Tracking. Accessed 14 May 2023. Video, 2:21. https://youtu.be/nRr21_DpLv4.
40. Zimmerman, T. 2023. 4 Unsupervised Classification. Accessed 14 May 2023. Video, 3:56. https://youtu.be/fqhqmwEr-ZI.
41. Zimmerman, T. 2023. 5 Supervised Classification. Accessed 14 May 2023. Video, 2:51. https://youtu.be/jEG0LfQaK4Q.
42. Zimmerman, T. 2023. 6 Image Processing Pipeline. Accessed 14 May 2023. Video, 10:20. https://youtu.be/UdlZGYCzjAo.
43. Zimmerman, T. 2023. 7 Image Quantization. Accessed 14 May 2023. Video, 2:10. https://youtu.be/KUrV1viPYUQ.
44. Zimmerman, T. 2023. 8 Morphology Features. Accessed 14 May 2023. Video, 1:15. https://youtu.be/CFwqvXWpnQY.
45. Zimmerman, T. 2023. 9 Confusion Matrix. Accessed 14 May 2023. Video, 3:50. https://youtu.be/D0VWQxuFgS8.
46. Zimmerman, T. 2023. 1 Representing Images in Arrays. Accessed 14 May 2023. Video, 4:52. https://youtu.be/e1ShaCfJnD0.
47. Zimmerman, T. 2023. 2 Representing Color Images. Accessed 14 May 2023. Video, 5:07. https://youtu.be/ZaWrz6ep1Ig.
48. Zimmerman, T. 2023. 3 Image Quantization. Accessed 14 May 2023. Video, 2:54. https://youtu.be/JEtvs7uNX8A.
49. Zimmerman, T. 2023. 4 Advanced Thresholding. Accessed 14 May 2023. Video, 4:40. https://youtu.be/JqScvKS3FJY.
50. Zimmerman, T. 2023. 5 Blurring. Accessed 14 May 2023. Video, 2:55. https://youtu.be/_p-jWNPoq_w.
51. Zimmerman, T. 2023. 6 Finding Objects. Accessed 14 May 2023. Video, 4:04. https://youtu.be/rOnPkJX_92A.
52. Zimmerman, T. 2023. 1 k Means Clustering (Part 1). Accessed 15 May 2023. Video, 2:04. https://youtu.be/cdmKwzJm_uc. 15
53. Zimmerman, T. 2023. 2 k Means Clustering (Part 2). Accessed 15 May 2023. Video, 6:40. https://youtu.be/78rYpJht4fw.
54. Zimmerman, T. 2023. 3 Determining Number of Cluster. Accessed 15 May 2023. Video, 2:53. https://youtu.be/EpVapQ2KLVw.
55. Zimmerman, T. 2023. 4 Plot Clusters. Accessed 15 May 2023. Video, 7:38. https://youtu.be/kscz01Y6x44.
56. Zimmerman, T. 2023. 5 Cluster Code. Accessed 15 May 2023. Video, 9:59. https://youtu.be/R9VxuE47tqg.
57. Zimmerman, T. 2023. 6 Other Clustering Methods. Accessed 15 May 2023. Video, 1:21. https://youtu.be/8P19AoQNoLQ.
58. Zimmerman, T. 2023. 7 Detecting Mouse Events. Accessed 15 May 2023. Video, 6:55. https://youtu.be/_Ah6L2PLwgw.
59. Zimmerman, T. 2023. CCCofficial/OpticalEngineeringLabCourse. Accessed 20 June 2023. https://github.com/CCCofficial/OpticalEngineeringLabCourse.
60. Kemp, R., A. Chippendale, M. Harrelson, J. Shumway, A. Tan, S. Zuraw, and J. L. Ross. 2020. Hands-on curriculum in optics of microscopy. *Biophysicist* 1:6. https://doi.org/10.35459/tbp.2019.000114.
61. Zimmerman, T. 2023. CCCofficial/Holo1. Accessed 20 June 2023. https://github.com/CCCofficial/Holo1/blob/main/reco.py.
62. Gabor, D. 1948. A new microscopic principle. *Nature* 161:777–778. https://doi.org/10.1038/161777a0.
63. Hall, R. inventor; General Electric Co., assignee. Stimulated emission semiconductor devices. 1966. US Patent 3245002, filed 24 October 1962, issued 5 April 1966. Accessed 16 May 2022. https://patents.google.com/patent/US3245002A/en.
64. Zimmerman, T. 2023. Part 1 of 3. How to build a Digital InLine Holographic Microscope (DIHM). Accessed 18 May 2023. Video, 15:17. https://youtu.be/rgpuIB9MnaM.
65. Zimmerman, T. 2023. Part 2 of 3. How to build a Digital InLine Holographic Microscope (DIHM). Accessed 18 May 2023. Video, 20:54. https://youtu.be/QulLrAdbIeI.
66. Zimmerman, T. 2023. Part 3 of 3. How to build a Digital InLine Holographic Microscope (DIHM). Accessed 18 May 2023. Video, 36:29. https://youtu.be/HobFIiBkCfs.
67. Zimmerman, T. 2023. Removing Pi Camera Lens. Accessed 18 May 2023. Video, 3:31. https://youtu.be/ZgE-U9m7L54.
68. Ibex. Raspberry Pi Resources. Accessed 20 June 2023. https://raspberry-projects.com/pi/pi-hardware/raspberry-pi-camera/using-the-camera.
69. RaspiCam Documentation. Accessed 20 June 2023. https://www.raspberrypi.org/app/uploads/2013/07/RaspiCam-Documentation.pdf.
70. Github. 2021. Show-me Webcam Releases. 20 June 2023. https://github.com/showmewebcam/showmewebcam/releases.
71. Amann, S., M. von Witzleben and S. Breuer. 2019. 3D-printable portable open-source platform for low-cost lens-less holographic cellular imaging", Scientific Reports,9, Article #11260.
72. Vishay. 2021. Siliconix IRF520 Power MOSFET. Accessed 20 June 2023. https://www.vishay.com/docs/91017/irf520.pdf.
73. Tobon-Maya, H., S. Zapata-Valencia, E. Zora-Guzmán, C. Buitrago-Duque, and J. Garcia-Sucerquia. 2021. Opensource, cost-effective, portable, 3D-printed digital lensless holographic microscope. Applied Optics. 60(4): A205–A214.
74. bioRxiv, The Preprint Server for Biology. Accessed 20 June 2023. https://www.biorxiv.org.