Biomedical Engineering Education, Vol. 2, No. 1, January 2022 (© 2021) pp. 41–59
https://doi.org/10.1007/s43683-021-00057-w

BMES BIOMEDICAL ENGINEERING SOCIETY

Innovation Article

Applying Research-Based Teaching Strategies in a Biomedical Engineering Programming Course: Introduction to Computer Aided Diagnosis

R. Rosario,[1] T. S. Hopper,[2] and A. Huang-Saad[2,3]

[1]Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA; [2]Biomedical Engineering, University of Michigan, Ann Arbor, MI, USA; and [3]The Roux Institute at Northeastern University, 100 Fore Street, Portland, ME 04101, USA

(Received 25 February 2021; accepted 14 September 2021; published online 26 October 2021)

Abstract—There are increasing calls for the use of research-based teaching strategies to improve engagement and learning in engineering. In this innovation paper, we detail the application of research-based teaching strategies in a computer programming focused biomedical engineering module. This four-week, one-credit undergraduate biomedical engineering (BME) programming-based image processing module consisted of a blend of lectures, active learning exercises, guided labs, and a final project. Students completed surveys and generated concept maps at three time points in the module (pre, mid, and post) to document the impact of integrating research-based teaching strategies. Students demonstrated a significant ($p < 0.05$) increase in conceptual knowledge, confidence with material, and belief in the usefulness of material from the beginning to end of the module. Students also had high ($>$ 4 out of 5) perceptions of gains in knowledge and attitudes toward instructor support. Overall, the novel design utilized multiple research-based pedagogies and increased students' conceptual knowledge, self-efficacy, and perceived usefulness of material. The proposed design is an example of how multiple research-based instructional strategies can be integrated into an undergraduate biomedical engineering course.

Keywords—Project-based learning, Scaffolding, Concept maps, Self-efficacy, Research-based instructional strategies, Conceptual knowledge.

INTRODUCTION

Over the past two decades, there have been calls by international and professional organizations to transform the state of undergraduate engineering education from theory-focused, teacher-centered instruction to practice-focused, student-centered instruction.[18,35] To heed this call, Litzinger, Lattuca, Hadgraft, and Newstetter proposed the incorporation of research-based instructional strategies that have been shown to promote the development of student expertise, deep conceptual knowledge, and professional and technical skill fluency.[21]

The call for research-based instructional strategies has been echoed in programming education. Research by Borrego, Froyd, and Hall determined that both the awareness and adoption of research-based instructional strategies were lowest in computer science education.[5] This is despite the clear benefits of such strategies when teaching computer science, specifically introductory programming.[12,44] This gap highlights a clear need to increase the incorporation of research-based instructional strategies when teaching coding skills.

Computer programming is fundamental for the academic and professional success of biomedical engineers. Many undergraduate biomedical engineering (BME) programs across the United States have a computing component.[20] Additionally, a recent survey of BME faculty identified programming as the second most important skill for the future careers of biomedical engineers, just behind statistics and more important than design, regulatory materials, biomate-

---

Address correspondence to A. Huang-Saad, The Roux Institute at Northeastern University, 100 Fore Street, Portland, ME 04101, USA. Electronic mail: a.huang-saad@northeastern.edu

Rosario and Hopper contributed equally to this work.

rials, and system processing.[41] Thus, developing strong programming fundamentals is critical to the success of BME undergraduates.

The purpose of this innovation paper is to describe several research-based instructional strategies and provide an example of how the strategies were executed in a computer programming module for biomedical engineering students. The impact of these strategies on student attitudes and learning are also presented.

LITERATURE

Project-Based Learning

Project-based learning (PBL) is a common student-centered learning practice that has been used in engineering education, specifically to promote student engagement with design and increase professional skills.[19,24,32] In this method, students apply their knowledge to an open-ended, authentic problem in teams, which is meant be similar to the professional engineering experience.[19,24,34] Capraro and Slough defined PBL as "an ill-defined task within a well-defined outcome situated with a contextually rich task requiring students to solve several problems which when considered in their entirety showcase student mastery […]".[7] Critical components of PBL include: well-defined outcomes, ill-defined tasks to promote self-directed learning, students working in cooperative groups to complete the task, instructors acting as a facilitator rather than provide explicit instruction, and projects having real-life applications.[7] PBL also often includes an end product as a summative assessment, often in the form of a device, program, design, report, and/or presentation.[2,19] PBL is known to increase student's motivation, satisfaction with their work and learning, long-term knowledge, professional skills, self-directed learning skills, and engagement.[11]

The implementation of PBL has been discussed and assessed in undergraduate engineering education. For example, at Massey University, the engineering curriculum has been redesigned with a focus on PBL in an effort to develop key professional competencies (knowledge acquisition, communication, problem definition, teamwork, system thinking, decision making, professionalism, and design process) in students.[34] Within PBL courses, students follow the project stages of comprehension of problem, creation of solution, critique, and communication. The benefits of PBL from this program include an increased learning of design principles, application of theory to practical problems, deep learning, and decreased rates of plagiarism. Students also had high satisfaction with PBL and its effects on their learning.[34] Furthermore, through a PBL-based civil engineering capstone class, Gavin suggests that PBL increased student confidence with group work, time management, and technical skills.[13] Students also had high satisfaction with PBL and high perceptions of their learning in a PBL course compared to lecture-based courses.[13]

Instructor support is an important factor to consider when implementing PBL.[19,24,34] Scaffolding, described below, is one method that can be used in PBL classes to structure instructor facilitation. Pleiss, Perry, and Zastavker found poor student outcomes (low self-efficacy, negative view of instructor support, and poor motivation) in a PBL-based course that did not implement scaffolding compared to a PBL-based course that implemented scaffolding.[28]

Scaffolding

The term "scaffolding," in the context of education, was first described by Wood *et al.*[43] and generally associated with the socio-cultural work by Vygostky.[29] In Wood *et al.*'s work, scaffolding was described as a process tutors use to support students in solving problems that is beyond the student's individual ability.[29] While scaffolding is generally accepted as an effective teaching practice, scaffolding strategies are generally ill-defined.[29] Van de Pol, Volman, and Beishuizen's 2010 literature review sought to leverage education research to describe how teachers scaffold student learning experiences in the classroom and to rigorously define the process.[29] Their synthesis of 66 articles identified three characteristics of scaffolding in the classroom: (1) contingency, (2) fading, and (3) transfer of responsibility. Contingency is the adaptation of support to the student, which must be responsive and tailored to student needs. This requires the instructor to determine a student's current competence before providing appropriate support. Fading is the gradual decrease of support. To be fading, the level of support must gradually decrease over time. This is closely linked to the final component, transfer of responsibility. To have a transfer of responsibility, the student must gradually take ownership over their learning.

Education research in scaffolding is largely based in the K-12 context. In one of the few studies addressing the effects of scaffolding on undergraduate engineering education, Mayer, Moeller, Kaliwata, Zweber, Stone, and Frank found that scaffolding single problem-based learning class sessions increased student performance on short-answer concept questions.[23] In this study, Mayer *et al.* operationalized scaffolding through ten-minute lectures on key concepts and guided handouts. The lecture and handouts introduced key concepts and guided students through the calculations necessary to solve the problem presented in the course. When compared to students in a problem-based learning session without scaffolding, the students in the scaffolded problem-based learning session scored higher on a post-course, short-answer concept test.[23]

Implications for Design and Evaluation

In this paper, we present the design of a four-week, one-credit module on introductory computer programming for BME students. The module was a product of the U-M Biomedical Engineering Incubator.[17] Because students had no prior image processing experience, this module was designed to focus on both the acquisition and application of knowledge. We used traditional lectures, to promote gains in conceptual knowledge, particularly in the short-term,[37] and PBL to mimic engineering professional realities by having students engage in the self-directed application of knowledge.[24] To strengthen the role of instructors as facilitators while implementing PBL and increase perceptions of instructor support and self-efficacy, scaffolding was incorporated throughout the module.[28] We defined four design goals for the module:

1. Students should demonstrate gains in programming skills as applied to the content
2. Students should be able to identify clear applications for skills and knowledge from the module
3. Students should demonstrate gains in conceptual knowledge
4. Students should have positive attitudes toward instructor support

IMPLEMENTATION

Overview

Introduction to Computer-Aided Diagnoses (IntroCAD) was a one-credit, undergraduate module that met for three-hours, twice per week for four weeks in the winter semester of 2020. The module was designed by three graduate students, the first two authors and a classmate, enrolled in the BME Instructional Incubator in Fall 2019. The first two authors (the graduate student teaching team) elected to co-teach the module in winter 2020 with the third author as their faculty mentor. The goal of the module was to provide students with a basic knowledge of digital image processing in the context of medical applications and to increase students' skills in image processing, computer programming, teamwork, and communication, as characterized by the learning outcomes in Table 1. There were three fundamental components: (1) lectures with active learning exercises, (2) labs, and (3) a final project (Fig. 1). Lectures and labs were designed to introduce foundational coding and image processing skills and concepts. The final project, a PBL exercise, allowed students to apply that knowledge to an authentic problem.

Both formative and summative assessment were used to provide feedback, gauge student perceptions, evaluate growth in conceptual knowledge and skills, and assess completion of learning outcomes (Table 1). Formative assessments included responses to lab questions, in-class activities, three surveys and daily muddiest points,[8]which accounted for 30% of the total grade. For each lab, students were asked to submit scripts and answer 5–6 questions, which is described in more detail in the lab section below. Lab responses were designed to address learning outcomes 1, 2, and 4 (Table 1). Students were graded on activities from lectures, which were designed to check for understanding of the content that was covered in class. The surveys helped the graduate student teaching team assess completion of design goals and adjust class session plans based on students' expectations, areas of confusion and interest, and general feedback. Students were asked to submit a muddiest point for every class session, which asked students to identify the most confusing part of the class which instructors would address at the beginning of the next class.[8] These muddiest points and surveys helped the graduate student teaching team to adjust class plans to address areas of greatest confusion, enabling the contingency required when implementing scaffolding. Summative assessment from the final project accounted for the remaining 70% of the total grade, consisting of a script (30%), final report (20%), and presentation (20%). The script assessed learning outcomes 1, 2, and 3, while the report and presentation assessed outcomes 4 and 5. Due to unexpected impact of COVID-19, the last class (Week 4, Thursday) was cancelled.

Thirteen students enrolled in the module (Table 2). Although the module was initially designed for sophomore BME students, 8 out of 13 students had already completed their sophomore year. In a pre-module survey, all students expressed that they had at least some prior introductory-level coding experience and had previously used MATLAB.

Lectures

Over the eight class sessions, five lectures (Fig. 1) were presented. Lectures 1, 4, and 5 focused on general coding skills, supplementing the university's introductory coding course for engineers. Lectures 2 and 3 introduced concepts relevant to image processing. All lectures except the first lasted 30 minutes or less. Lec-

TABLE 1. Learning outcomes and the corresponding module element designed to meet that outcome.

Number	Learning outcomes	Portion of module covered in
1	To apply automated image processing techniques to medical images	Lecture, labs, final project
2	To implement industry best practices to create organized, efficient, and understandable code	Lecture, labs, final project
3	To work as a team to design an algorithm to identify and describe illness or injury	Final project
4	To critically evaluate methods used in scripts	Labs, final project
5	To communicate the motivation for creating their scripts, methods, results, and broader implications and future extensions of their final scripts.	Final project

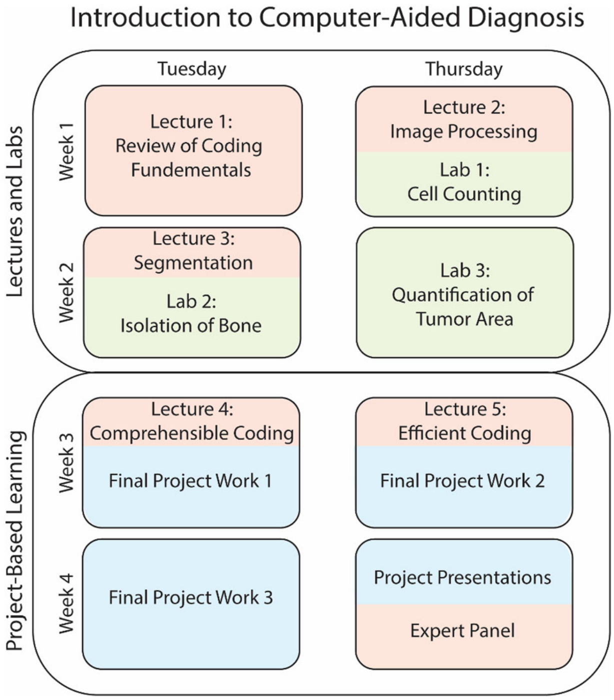

FIGURE 1. Schematic of module schedule with lectures (orange), labs (green), and final project (blue) indicated.

TABLE 2. Student demographics for those enrolled in IntroCAD.

- Class level
  - 1st year: 0
  - 2nd year: 5
  - 3rd year: 4
  - 4th year and higher: 4
- Gender
  - Male: 5
  - Female: 8
  - Formal programming experience
  - None: 1
  - Only introductory courses: 6
  - Higher level programing courses: 6
- Confidence with Image processing
  - Strongly agree: 2
  - Somewhat agree: 1
  - Neither agree nor disagree: 3
  - Somewhat disagree: 5
  - Strongly disagree: 2

tures incorporated active learning exercises and facilitated discussions. Each class started with a clear explanation of the daily learning objectives and ended with a muddiest point exercise.[8]

Lectures 1–5 were scaffolded; lecture content was informed by student survey responses and designed to address lab questions and muddiest points, demonstrating contingency. Lectures 1 and 2 included more thorough descriptions of concepts, and lectures 3–5 evolved to be more open-ended and included student-led discussions about problems in the field and issues students were facing in their final project. This progression illustrates fading and transfer of responsibility, two critical components of scaffolding.

Lectures 1, 4, and 5 heavily incorporated active learning exercises. Lecture content was presented to students in 3–10 min intervals, which were immediately followed by sample problems. Students completed the sample problems, asking their peers for help as needed. For lecture 1, which reviewed coding fundamentals, a phone-based application was used to poll student responses to concept-based questions throughout the lecture. If most students answered incorrectly, then additional instruction was provided, thus demonstrating contingency. For lectures 4 and 5, which covered coding for comprehensibility and efficiency, the graduate student teaching team led a discussion where students discussed their evaluations of sample scripts in terms of the criteria for comprehensibility and efficiency that were reviewed in lecture. To incorporate active learning in all lectures, students were often asked to consider questions individually before discussing in small groups and sharing to the class (think-pair-share), think of real-world applications, critique methods, and summarize key concepts from labs and lectures.[10]

Labs

Labs were self-paced image processing exercises that students completed individually. There were three labs in total, which introduced loading image data, segmenting images, and quantifying features (Fig. 1). As an example, the workflow from the third lab is shown is Fig. 2. Each lab was designed to meet the following five criteria: (1) a focus on a real-world problem, (2) a beginner-level set of instructions and questions to address the problem, (3) advanced-level open-ended extra credit questions to address a nuance on the problem, (4) an accumulative build-up of knowledge with progressively less detailed instructions, and (5) a low-stakes checklist for formative assessment.

Real-World Problem

For each lab, students were tasked with solving an authentic task associated with medical images, which has been shown to increase engagement with material and learning.[36] This allowed for unexpected errors that required students to engage in basic inquiry to correct. These errors occur because real-world data has inher-

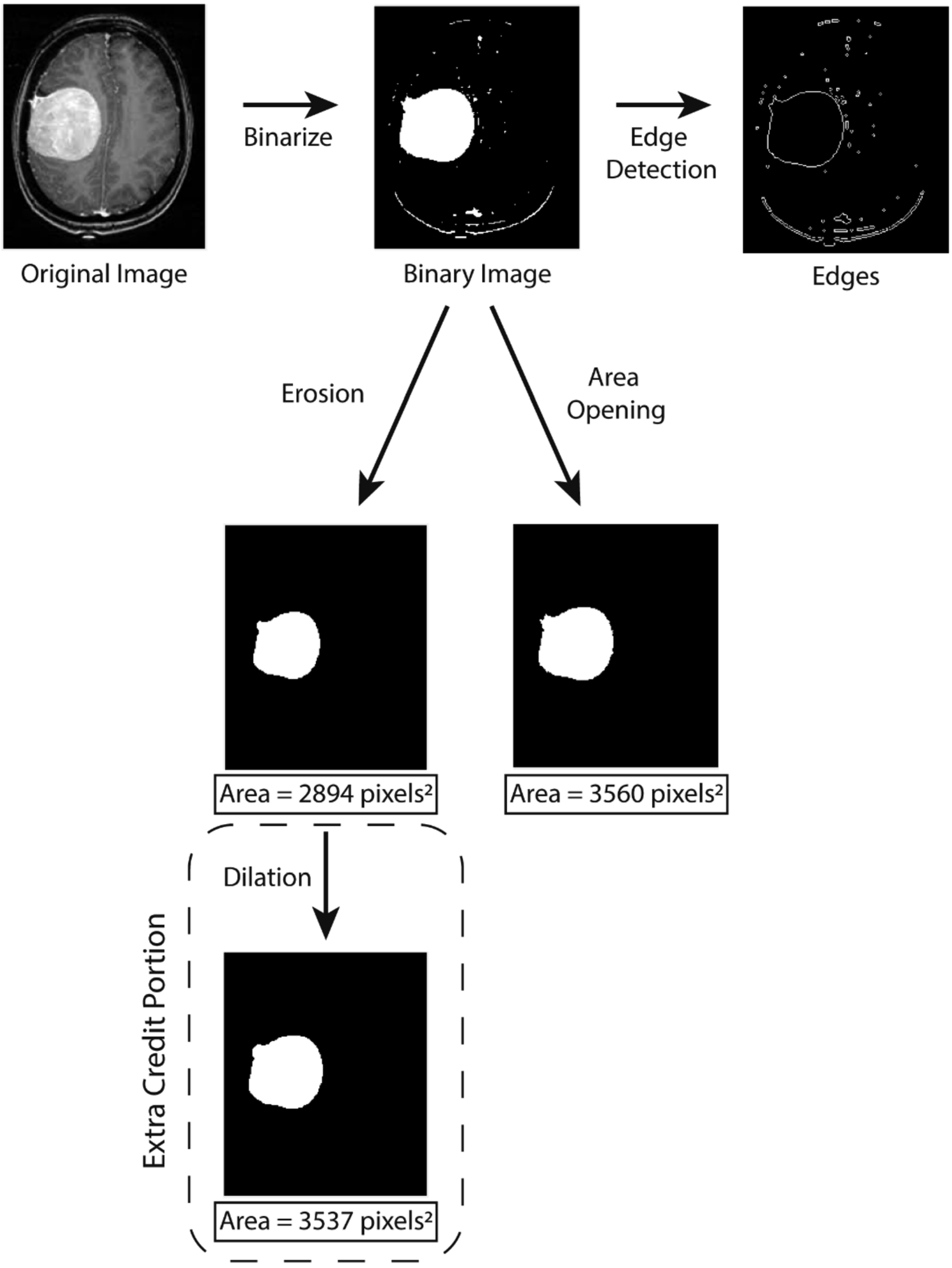

FIGURE 2. Workflow of Lab 3: Quantifying Tumor Area. Students were given the original image and needed to conduct sequential code-based image processing steps to isolate and quantify the area of a brain tumor using two different methods. Students then answered questions related to the validity of both methods.

ent noise and deficiencies that can make pre-set tasks difficult. Finally, using real medical images prepared students for their final projects, where students applied concepts from labs to a larger, unstructured real-world problem.

Beginner-Level Instructions and Questions

Each lab included beginner-level instructions and questions that students with minimal coding experience should have been able to complete. These instructions and questions were designed for the student who can write a simple script using mathematical operations, variables, conditionals, loops, and function calls; all skills that are learned in the first-year coding course for engineers at the institution. Student confidence with these skills was also assessed using the pre-module survey, demonstrating the scaffolding component of contingency. Using only that introductory knowledge and instructions included in handouts, students were expected to be able to complete the exercises. In some cases, this was accomplished by providing several lines of code that students could directly insert into their scripts. When code was provided to students, a corresponding follow-up question that asked students to explain the function was included to confirm comprehension. Lab instructions and questions were reviewed by four BME faculty, one Ph.D. student in engineering education, and one postdoctoral fellow in engineering education for clarity prior to the launch of the module.

Expert-Level Extra Credit Questions

Recognizing that students of varying levels may enroll in the module, complex, open-ended extra credit questions that required more advanced knowledge and inquiry to solve in the lab handouts were also included. These questions were designed for students with more experience in computer programming and covered concepts that were not explicitly introduced in lecture. The extra credit questions took two forms, either asking a student to explain a concept that was introduced but not explored in the lab activity or implementing a new piece of code that necessitated the use of functions not discussed in the module. To complete the extra credit assignment, students often needed to search MATLAB documentation to understand the nuances of functions or to identify a function that would meet their needs. The more advanced extra credit questions required outlining and iteration to complete. By including questions that were targeted at both beginner- and advanced-level students, the lab activities incorporated contingency.

Progressively Less Detailed Instructions

Fading was explicitly incorporated in the design of the labs. Each successive lab activity required the use of skills and knowledge that were introduced in prior lab activities with the goal of introducing students to a range of skills and serve as a starting point for their final projects. This was accomplished in the lab handouts through careful wording of the instructions. Lab handouts included more detailed instructions when first introducing a skill or concept with less detail in subsequent labs. For example, in lab 1, one step had students convert an image to grayscale: "Since this is an RGB image, we need to convert to grayscale before we can binarize or perform our other operations. Use the function rgb2gray to create a new image matrix." Much less detail was provided for this step in Lab 2, where a similar step instructed students to "convert the image to grayscale," with no additional instruction. If students were unsure how to perform any of these steps, they were referred to prior lab materials or the MATLAB help directory. The lab handouts demonstrated two of the critical components of scaffolding: fading and transfer of responsibility. Table 3 provides specific examples for how each lab incorporated these first four lab design criteria.

Formative Assessment

Lastly, lab handouts provided a low-stakes opportunity for students to identify their current knowledge level and gaps that needed to be addressed. Incorrect responses on the lab handout resulted in a small (1–2%) point reduction. Full credit was awarded if the script compiled, all steps of the lab were followed, all questions were answered in 1–2 sentences, and all figures were created with descriptive labels. Lab assignments were graded and returned before the due date of the next lab, which allowed students to address their issues in the next lab. The labs provided beginner-level students with a low-stakes opportunity to acquire the skills necessary to effectively contribute to the final projects.

Additional Considerations

While not a key component of the design process, it should be noted that students were asked to sit with their project teams when completing the labs to encourage peer-to-peer learning.[16] When a student had a question about the lab, the graduate student teaching team encouraged the students to first discuss the question with their project teams. If the team was unable to answer their question through discussion, instructors would re-enter the discussion. By the end of the third lab, most student questions were answered by

TABLE 3. Description of how four of the lab core components were included in each activity.

Component	Lab 1	Lab 2	Lab 3
Real-world problem	Count cells from fluorescent microscope images	Segment bones from knee x-ray images	Quantify tumor size from brain MRI scans
Beginner-level instructions and questions (contingency)	Count three isolated cells from a high-contrast image with step-by-step, highly detailed instructions. Requires explanation of provided code with minimal independent implementation. Learning objectives: (1) Define image properties (2) Manipulate images using arithmetic operations and built-in functions (3) Identify basic process of segmenting images	Isolate bones from x-rays. Uses images with less contrast between the region-of-interest and the background. Requires some independent code implementation. Learning objectives: (1) Define and implement image morphological operations (2) Identify issues caused by morphological operations	Isolate and quantify tumor size from MRI scans with very low contrast between the region-of-interest and background. Requires nearly independent code implementation. Learning objectives: (1) Identify image processing difficulties caused by low contrast (2) Interpret MATLAB help documentation (3) Quantify properties of a segmented image
Advanced extra-credit (contingency)	Count cells from an image with many highly clustered cells. Requires logic and/or functions not used in the beginner lab.	Redo the lab using built-in functions that were not introduced in the lab instructions.	Create a metric and implement a script to identify whether the tumor is likely to be malignant based on its shape.
Example of progressively less detailed instructions (fading)	Explicit instructions are provided for grayscale conversion and binarization: *Since this is an rgb image, we need to convert to grayscale before we can binarize or perform our other operations. Use the function rgb2gray to create a new image matrix.* *Use the imbinarize function to binarize the image.*	Explicit instructions are provided only for binarization because a more complex process is used: *Convert the image to grayscale* *We will create a binary version of the image with a threshold value calculated by the 'sobel' operator. Use this threshold value with the edge function to create a binary image. Use the following code to do so: [~, threshold] = edge(<grayscale image>, 'sobel');*	Because (1) grayscale conversion and binarization were previously used and (2) binarization requires a grayscale image, only the final instruction was provided: *Binarize the image.*

their peers without instructor guidance. Lab handouts in their entirety are provided in the Supplemental Materials.

Final Project

The last half of the module was devoted to the final project, where students extended what they learned in the first four sessions and incorporated knowledge from other biomedical engineering domains. Premodule survey results were used to create groups with evenly distributed coding ability. Groups were tasked with selecting one of eight problems. Each problem instructed students to use image processing to quantify a clinical parameter, such as ''quantify the age and size of a fetus.'' Students were also responsible for finding their own radiographic images from medical databases. None of the problems were previously used as an example in the module. Each problem provided a well-defined task with clear real-life applications.

To solve the problem, students needed to seek out new knowledge. Specifically, they needed to understand the clinical problem, to understand the corresponding physiology, and gain additional imagine processing skills. As a result, the graduate student teaching team observed the students engage in self-directed learning. For example, one student group chose to analyze a computed tomography (CT) scan to determine whether a patient had kidney stones (Fig. 3). The students were unfamiliar with the causes of kidney stones, their appearance in medical imaging, and relevant clinical markers when making a diagnosis. To address this knowledge gap, the students looked to general online sources and academic papers on kidney stones. After sharing knowledge among group members, the students applied this conceptual knowledge of the pathology to their iterative algorithm development, where they suggested potential identification methods, acquired missing imaging processing skills-based knowledge, implemented their identification method in MATLAB, and evaluated the algorithm's performance. The students' process was emblematic of cooperative, self-directed learning that occurs during PBL when the project is ill-defined.[7]

Consistent with the definition of PBL by Capraro and Slough, the module instructors facilitated the

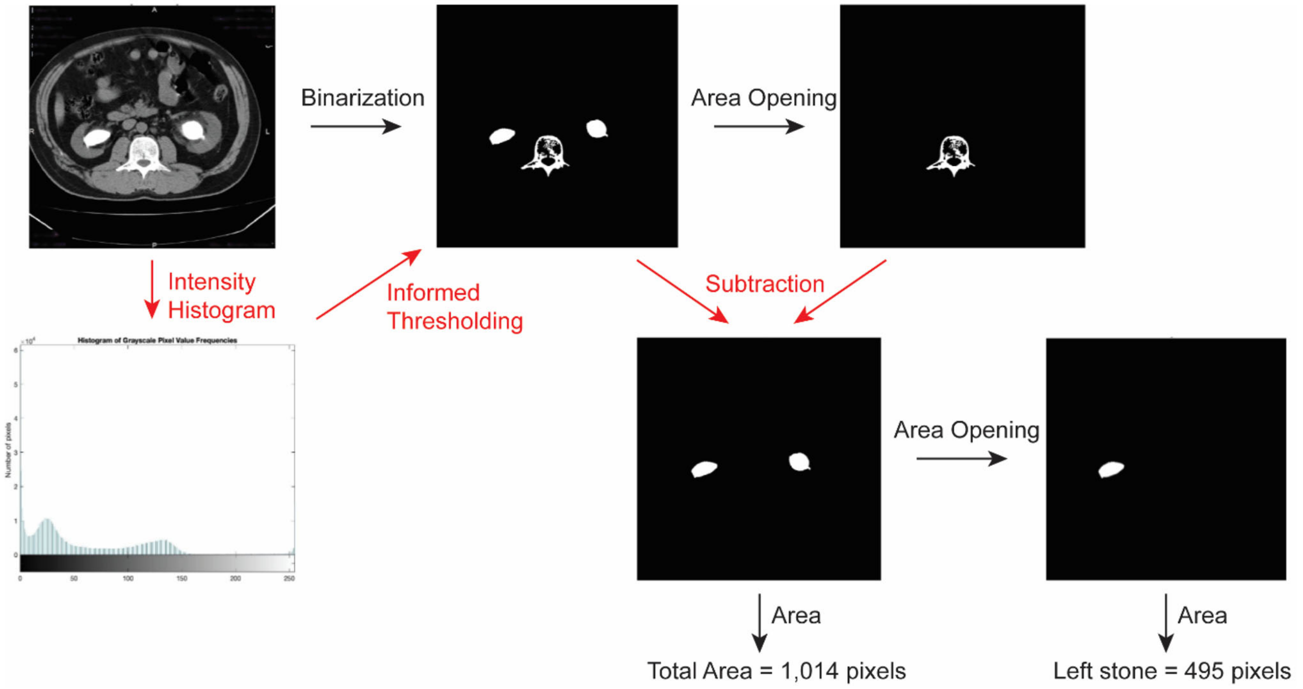

FIGURE 3. Representative final project submission. For this project, the team segmented and quantified cystic kidney stones. The team used thresholding based on average intensity values to isolate the kidney stones and spine from the original image, opening to isolate the spine, and then image subtraction to isolate the kidney stones. Image processing techniques not covered in class lectures or labs are shown in red

learning process and modeled reasoning strategies rather than provide explicit instruction during the final project portion of the module. In their role as facilitators, the instructors asked questions and pointed students toward relevant resources rather than directly answering their questions. Again, looking at the kidney stone group, the students wanted to create a more versatile script by using an automated thresholding method rather than a hard-coded value as was done in the guided labs. Initially, the student group asked the instructors for guidance. Rather than provide an algorithm, the instructors prompted the students to brainstorm and evaluate potential automated thresholding methods. Ultimately, the students created their own algorithm that used an intensity histogram to choose an informed threshold value for their images. Through this self-directed learning process, the students discovered informed thresholding techniques that went beyond the skills covered in the first two weeks of the module. Without instructor facilitation, the students would have been unlikely to engage in self-directed learning and move beyond the materials covered in the initial lab sections.

In addition to facilitating group discussion and self-directed learning, the graduate student teaching team guided students through the PBL project phases used at Aalborg University to model effective project management skills, as seen in Table 4.[19] Students were assigned a project planning worksheet, which prompted to complete project phases 1–5 by the end of the second week of the module. In phases 1–3, students defined the problem by providing a brief background on the body system and/or pathology and by identifying what they plan to quantify in their chosen medical image. In phases 4–5, students began to solve the problem by identifying which image processing techniques they will likely need to use and by developing a pseudocode outline of their final script. Completing this worksheet prepared students for the final project work sessions, where they focused on implementing, iteratively improving, and evaluating the performance of their final scripts.

At the end of the module, students were asked to turn in three assignments for their project: a script, report, and presentation. The final script demonstrated students' skills-based knowledge gains from the project. The final report provided students an opportunity to evaluate how well their script addressed the posed problem and to abstract the knowledge acquired during from the specific problem and toward more general problems of image processing. This reflection is a key characteristic of PBL defined by Kolmos and de Graaff.[19] The final presentation allowed students to share learnings from their project across student teams and with our expert panel, who could then model expert-level image processing thinking to the students.

TABLE 4. Implementation of Project Phases framework from Aalborg University in IntroCAD.[19]

Project phases	Implementation	Portion of module
1. Initiating the Problem	Description of problem statement and identification of pathologies	Project handout and planning sheet
2. Problem Analysis	Motivation of problem statement and introductions of report and presentation	Final report
3. Definition and formulation of problem	Description of problem statement	Project planning sheet
4. Problem solving methodologies	Lectures of image processing basics and implementation in labs; self-directed learning during project-based learning	Project planning sheet and final report
5. Demarcation	Discussions within project groups and guided question from instructors	Project work time
6. Solving the problem	Iterative development of scripts	Project work time
7. Implementation	Project scripts	Final script
8. Evaluation and reflection	Critical analysis within written report and presentation	Project work time and final report

TABLE 5. Deliverables for the team-based final project along with the learning outcome they fulfill, guidelines given to students, and grading criteria.

Final project component	Relation to learning outcomes	Deliverable	Grading criteria
Script	1, 2, 3	A MATLAB script that segments, quantifies, and creates figures relating to an injury or illness	Meets specifications, readability, documentation, and efficiency
Presentation	5	A 10-15-minute presentation on the background and motivation, methods, results, and discussion	Content, oral communication, and organization
Report	4, 5	A $<$ 3-page report with background and motivation, methods, results, and discussion	Content and formatting

Because the week 4, Thursday class was cancelled, teams were not graded on oral presentations, but only on their digital presentations. Summative assessments are summarized in Table 5, along with the grading criteria.

EVALUATING STUDENT PROGRESSION

Surveys and student-generated concept maps were used to document student progression throughout the module. Specifically, students' changes in perceived skills-based knowledge, perceived applicability of content, attitudes toward instructor support, and conceptual knowledge were measured. These modes of evaluation addressed the four goals that were established during module design. Data collection was approved by Internal Review Board (HUM00176990).

Survey Design, Collection, and Analysis

Survey questions were based on the previously validated scale for measuring attitude towards computer programing (AStCP)[1,42] and adapted by Baser.[1] Survey items assessed usefulness of and confidence in learning computer programming (pre- and post-) and with image processing (pre-, mid-, and post-) with a 5-point Likert-type scale, which asked students to rate their agreement with statements with provided answers *strongly disagree, somewhat disagree, neither agree or disagree, somewhat agree,* and *strongly agree*. In addition to the Likert-type questions in the mid-module survey, students were also asked to describe what was and was not working well in the module. Similarly, post-module, students were asked about perceptions toward PBL, instructor support, and knowledge and confidence with computer programming and image processing. Students were asked to describe their expectations and whether those expectations were met. Each of the three surveys can be seen in the appendix. Surveys were created and distributed using Qualtrics.

Twelve out of 13 students completed the survey at all three time points. Student responses were paired and analyzed with a non-parametric Wilcoxon Rank Sum test to measure the effect of IntroCAD on student attitudes toward the usefulness of and confidence in computer programming between the pre- and post-module. A non-parametric Wilcoxon Rank Sum test was used because of the small sample size. Population statistics rather than sample statistics were used for all tests because the analyses were designed to measure changes in attitudes for students in this module rather than a general population of biomedical engineering students.

Concept Maps

Concept maps were used to examine changes in conceptual knowledge over time. A concept map is a diagram where distinct concepts are connected by propositions. From a constructivist perspective, a concept map is a physical depiction of conceptual knowledge in the form of an interconnected web of facts, structures, and ideas.[14,26] This diagrammatic method of organizing conceptual knowledge was formalized by Novak and Gowin, originally as a method of organizing clinical interview data.[27] Since their creation, the uses of concept maps have multiplied, and are now used as a method of assessing depth and breadth of conceptual knowledge, interconnectedness of concepts, and student misconceptions.[3,39] The use of concept maps as an assessment method in engineering education has grown over the past two decades, providing insight beyond what is provided by traditional assessment methods into how students generate conceptual knowledge.[4,9,22,33,38,40]

In this module, students generated concept maps during the first class session, mid-module, and post-module. At the first time point, the process for creating concept maps was reviewed and an example was discussed as a group. After the initial instructional session, students independently generated concept maps given the initial bubble "Image processing". Examples of student-generated concept maps can be seen in Figs. 5 and 6. No time limit was given to generate concept maps, but most students completed their maps within 10–15 min.

Concept maps were scored using an adapted version of the validated rubric proposed by Besterfield-Sacre *et al.*, which assesses concept maps based on comprehensiveness, organization, and correctness (Table 6).[3] Besterfield-Sacre *et al.* used this rubric to quantify discipline-specific conceptual knowledge growth of engineering students over time.[3] Comprehensiveness describes the breadth and depth of a concept map. In this context, we used comprehensiveness to assess whether students sufficiently included the module content. For this module, students were expected to include concepts related to image acquisition, image properties, fundamental coding skills, segmenting, morphological operations, and quantification for a medical diagnosis. Organization describes the physical layout of a concept map, based on hierarchy structure and interconnectedness of knowledge. A higher organization score indicates expert-level conceptual knowledge, highlighting the hierarchical nature and interconnectedness of concepts. Correctness measures the validity of concepts and links between concepts. Comprehensiveness and organization are measures of content coverage, where students are awarded points for including additional complexity. Correctness is a measure validity, where points are taken away for incorrect usage of terms or links between concepts.

Prior to scoring the student-generated concept maps, the first two authors scored a set of six concept maps generated by doctoral students and faculty that use image processing techniques in their research. This was done to obtain consistency when scoring concept maps with the rubric. After the initial training, the instructors independently scored all student-generated concept maps, which were de-identified and scored in a random order. The first two authors then met to discuss scores and reach a consensus score for each concept map and criterion. A composite score was generated for each concept map by taking the sum of scores across rubric criteria.

To assess changes in conceptual knowledge as demonstrated by concept maps, the concept map composite score was examined over time. The median composite scores for module pre- to post-module were compared using a one-sided paired Wilcoxon rank sum test. Following quantitative analysis of concept map scores, qualitative document analysis of the concept maps was conducted for a select number of students. Two concept maps from two students that were representative of the changes in holistic scores observed from pre- to post-module were chosen for document analysis. One student had minimal growth in holistic score, while the other had substantial growth. Student in-class assignments and responses to survey questions were used to inform conclusions on changes in conceptual knowledge.

FINDINGS

Surveys

Results of the Wilcoxon statistical analysis for student's attitudes toward the usefulness and confidence with computer programming are presented in Table 7. Significant increases were found between pre- and post- time points for confidence with image processing and coding ($p < 0.05$). Mean responses for perception of increase in knowledge and confidence from IntroCAD can be seen in Table 8, along with the average mean score for the 29 questions relating to attitudes toward instructor support.

In the mid-module survey, students identified three main areas that were working well: the utility of lectures and labs, helpfulness of in-class work time, and small group work. Students mentioned that labs and

TABLE 6. Holistic scoring rubric adapted from Besterfield-Sacre *et al.*[4]

Criteria	3	2	1
Comprehensiveness: Covering content completely or broadly	The knowledge is very simple and/or limited. Minimal coverage of content. No extensions beyond what was covered in the module	Some content is covered. There is one extension beyond what was covered in the module, but it is not fully develope	Covers nearly all content and includes at least one fully developed extension (i.e., there is hierarchy level below that extension)
Organization: Arranging by systematic planning and united effort	Hierarchies have no cross-links between concepts and no branch structure	There is at least one cross-link between concepts and at least one branching hierarchy	There are multiple cross-links and branching hierarchies. Or uses a net-like structure with multiple feedback loops
Correctness: Conforming to or agreeing with fact, logic, or known truth	The map is naïve and contains misconceptions about the subject area; inappropriate words or terms are used. The map documents an inaccurate understanding of some subject matter	The map has some subject matter inaccuracies; most links are correct	The map integrates concepts properly and reflects an accurate understanding of subject matter meaning with few or no misconceptions

TABLE 7. Results of usefulness and confidence survey questions and Wilcoxon analysis from Likert-scale questions out of 5. Mean +/- standard deviation and median are shown.

Question	Pre-mean	Pre- median	Post-mean	Post- median	Wilcoxon comparison pre-to-post
Coding skills are important for biomedical engineers in industry	4.17 (± 0.90)	4	4.42 (± 0.64)	4.5	*p* = 0.14
It is important for me to learn coding skills	4.54 (± 0.50)	5	4.75 (± 0.43)	5	*p* = 0.14
I feel confident in my ability to digitally manipulate medical images	2.75 (± 1.3)	2.5	4.5 (± 0.65)	5	*p* = 0.0103
I can use computer programming to solve BME problems	3.42 (± 1.3)	4	4.5 (± 0.50)	4.5	*p* = 0.0119

TABLE 8. Results of perceived increase in knowledge and confidence from Likert-scale questions out of 5 from the post-module survey. Mean +/- standard deviation and median are shown.

Question	Mean	Median
IntroCAD increased my knowledge of image processing	4.9 (± 0.28)	5
IntroCAD increased my knowledge of computer programming	4.7 (± 0.62)	5
IntroCAD increased my confidence in computer programming	4.6 (± 0.76)	5
The PBL techniques used in IntroCAD increased my learning	4.5 (± 0.76)	5
Average attitudes toward instructor support	4.5 (± 1.12)	5

lectures, specifically "real-world examples" were useful and helpful for them, showing that they could see the applicability of the material. Regarding the module structure, one student responded, "the class format, which includes a 1-h lecture and 2 hours of interactive lab time, gives a good balance of learning and practice of the material," while another said "I like the short lectures before the lab. Also, having help throughout the lab definitely helps me understand it more." Students had high perceptions of the proposed pedagogy at the mid-module timepoint. When asked what they would change about the module going forward, seven students said nothing, and the other responses identified two weakness of the module: the simplicity of content covered and rushed nature. Multiple responses said the module was not difficult enough or was too general, while other students said it moved too quickly through material.

In the post-module survey, 11 students said their expectations were met while one said they were "kind of" met. The reasons students gave for why their expectations were met included learning, practicing, and increasing confidence in coding and image processing. The student who did not have their expectations met, critiqued the module by saying "it was more of an image processing class disguised as BME related. Most of the material could have been self-taught with MATLAB." This is related to the identified weaknesses of the module from the mid-module survey. When asked what they would change about the module, five students did not have suggestions. The other students' suggestions fell into the following categories: increasing the difficulty, increasing the depth of material, and increasing the amount of critical thinking in the module. One response said, "make this [module] harder or go deeper, it's an interesting topic that we didn't get into much." Another response said, "making the labs more open-ended and having students think more critically about how to process the images."

Concept Maps

Results for student holistic scores can be seen in Fig. 4

. Nine students completed both the pre- and post-module concept map. Student concept map holistic scores steadily increased throughout the semester. There was a statistically significant increase in median concept map score from pre- to post-module ($p = 0.021$).

Document Analysis

Student concept maps were analyzed to better understand what differentiated a student with high pre-to-post gains in concept map holistic score from a student with low gains. One student from each group was chosen. The maps were then examined for how connections between concepts evolved and how that evolution related to student survey data.

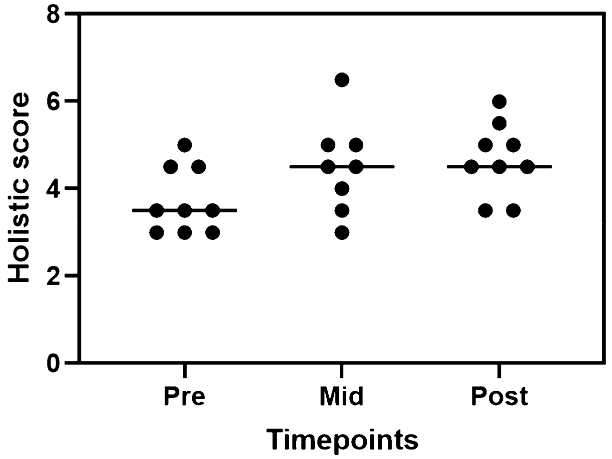

FIGURE 4. Concept map scores for all students that completed the assessment pre-, mid-, or postmodule. A significant increase was found between pre- and post-module.

Student 1: Alicia

Alicia demonstrated high growth in concept map holistic score from pre- to post-module (Fig. 5). Alicia was a second-year woman in the BME department. She took AP Computer Science in high school and an introductory engineering programming course prior to enrolling, and she was co-enrolled in a 200-level programming course taught through the university's computer science department that focused on data structures and did not cover image processing, which gave her more coding experience than her peers in IntroCAD.

Alicia's more extensive coding background was evident in her pre-module concept map. She incorporated multiple foundational coding topics that were covered throughout the module and necessary for a high score in the comprehensiveness section of the rubric. In addition to foundational coding knowledge, Alicia included several concepts that related to the medical images used as inputs, another key idea covered in IntroCAD. Alicia's pre-module concept map did not include any concepts related to image processing methods or outputs, which was reasonable given that she had no prior experience with image processing.

In her mid-module map, Alicia demonstrated a large amount of growth in content knowledge related to image processing. Alicia's mid-module map included concepts related to image processing methods and properties of digital images, using field-specific language. This growth was similar to that of other students without Alicia's background in computer science, highlighting that a more extensive computer science background was not needed to grasp module concepts. Despite gains in comprehensiveness, Alicia had minimal gains in correctness because many of the connections contained within the map were naïve. For example, she connected "MATLAB," "matrices," "segmentation," and "manually coding" to the central bubble of "image processing,“ even though all these concepts were specific examples rather than general higher order concepts.

In her post-module map, Alicia demonstrated more extensive coverage of content and had more nuanced connections between concepts. Whereas her mid-module concept map demonstrated a naïve understanding of image processing by connecting specific examples to the central "image processing" bubble, her post-module concept map has some these specific examples branching off more general concepts (e.g., "Segmenting" is now a sub-concept of "Methods"). It is interesting to note that the organization of Alicia's concept map remained roughly constant throughout the module, increasing slightly in complexity mid-

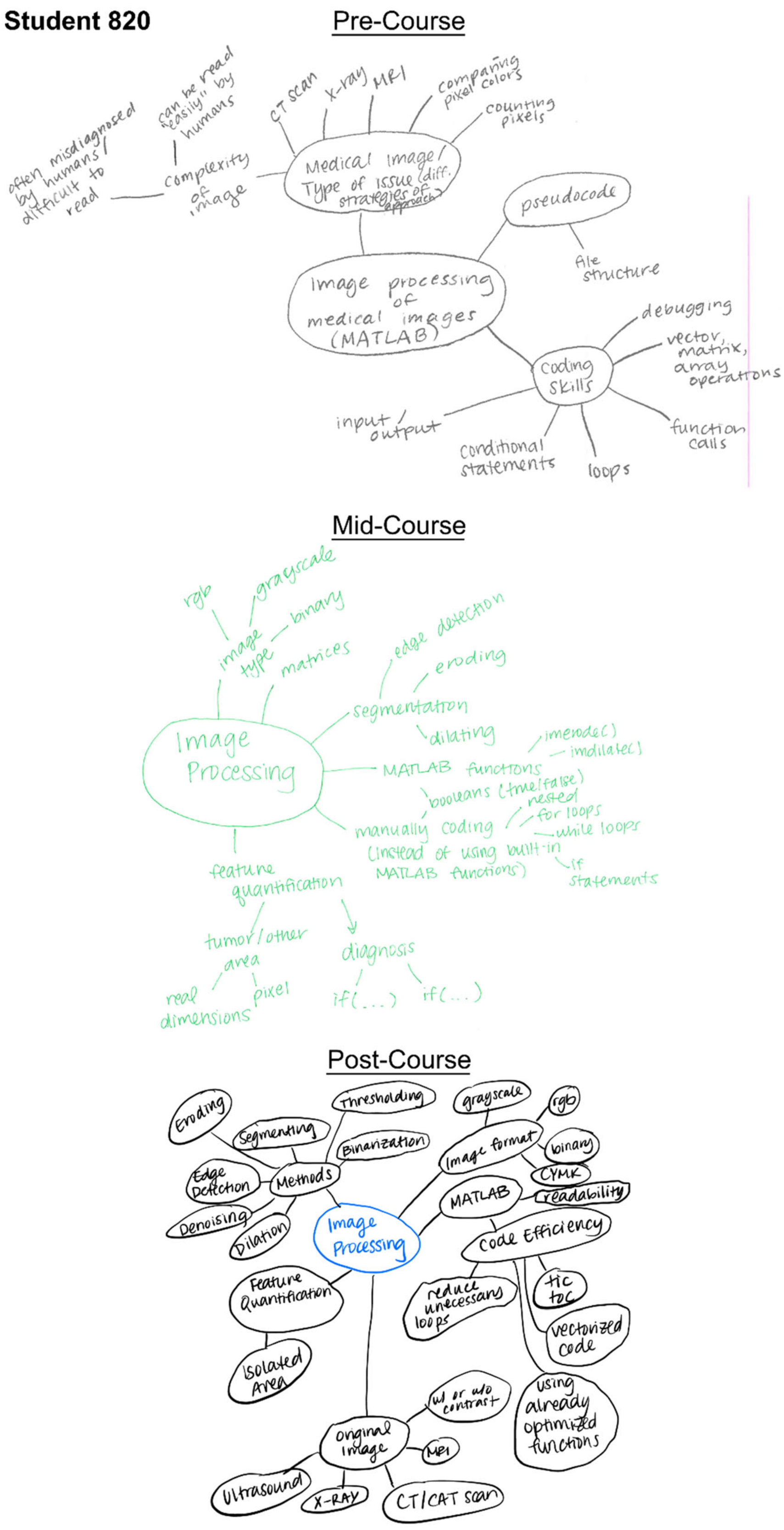

FIGURE 5. Concept maps from Alicia, who demonstrated high growth in concept map holistic score pre- to post-module.

module the inclusion of a cross-link (though that cross-link is arguably invalid), and that the organization was that of a novice. This constant organization could suggest that Alicia's knowledge structure is subject to rapid change as she learns more about image processing and works toward a stable, highly networked knowledge structure.

Student 2: Tara

Tara demonstrated minimal growth in concept map holistic score from pre- to post-module (Fig. 6). Tara was a third-year woman in the BME department. She had taken the introductory engineering programming course but had no other coding experience. Tara's primary motivation for taking IntroCAD was to increase her proficiency with MATLAB and was not related to the content material. In multiple short-response and Likert-type questions in the module pre-survey, Tara emphasized the importance of coding skills to her future career prospects in the BME industry.

Tara's pre-module concept map incorporated multiple concepts related to clinical uses of medical images, including concepts like clinicians, imaging modalities, and disease states that would require imaging. The inclusion of these concepts could have been enabled by Tara's more advanced position in the degree program; many BME undergraduates have finished general requirements and start taking more discipline-specific coursework in their third year. Tara's pre-module concept map did not include any concepts related to image processing methods, properties of digital images, or analysis of digital images. Beyond the inclusion of several MATLAB-specific image processing functions, Tara's concept map did not develop much past her pre-module map. This lack of development in conceptual knowledge could have been due to a greater focus on coding skills acquisition rather than increased conceptual understanding of image processing during the module.

DISCUSSION AND LESSONS LEARNED

In this innovation paper, we describe the design and implementation of a four-week, one-credit module that combined traditional lectures, scaffolding, and PBL to provide instructors with a roadmap for incorporating and assessing the impact of research-based instructional strategies into their teaching. The structure of this module included two weeks of traditional lectures and labs with scaffolding and active learning exercises and two weeks of final project work time. Critical components of both scaffolding and PBL were incorporated into the module, the implementation of which can be seen in Table 9 (for PBL) and Table 10 (for scaffolding). Overall, students positively engaged with the material, educational strategies, labs, activities, and PBL.

To evaluate the impact of the innovation in relation to our design goals, we used surveys and concept maps to demonstrate student gains in conceptual knowledge, perceived gains in skills-based knowledge, perceived applicability of skills, and high perceptions of instructors. Students indicated in the post-module survey that the module increased their knowledge in both image processing and computer programming (above 4.5 on a 5-point Likert-type scale, Table 8). Significant increases were found from pre- to post-module in confidence with image processing and use of computer programming to solve BME problems Students' attitudes toward instructor support and PBL had average scores of above 4 on a 5-point Likert-type scale, indicating that students had high perception of instructors and instruction techniques. The significant increase in concept map holistic scores from pre- to post-module time points shows that students gained conceptual knowledge from the module. Students in this module demonstrated clear gains in skills acquisition, self-efficacy, and beliefs in the applicability of knowledge, which suggests that our design, which had mini-lectures and labs preceding the final project, met our design goals and positively impacted students (Table 10).

Future iterations of this module could be adjusted to improve student learning outcomes, engagement, and satisfaction with the module. Three students indicated they were not satisfied with the difficulty of the module in the post-module survey, which could be addressed by increasing the number and difficulty of lab extra credit portions and by adjusting the difficulty of the final project script. The difficulty of the extra credit portions of labs could be increased by providing more open-ended questions or by providing more explicit correctness criteria for existing questions. The difficulty of the final project script deliverable could be altered by including an explicit requirement that students use methods outside of those covered in the lab portion of the module. Two of five submissions solely used functions that were introduced in labs; including this requirement would have pushed the students in those two groups to further extend their knowledge. Additionally, this requirement would promote self-directed learning.

Aside from adjusting the requirement for the final script, there are several other potential adjustments to the module that would promote acquisition of self-directed learning skills. To better guide students' self-directed learning, instructors could walk students

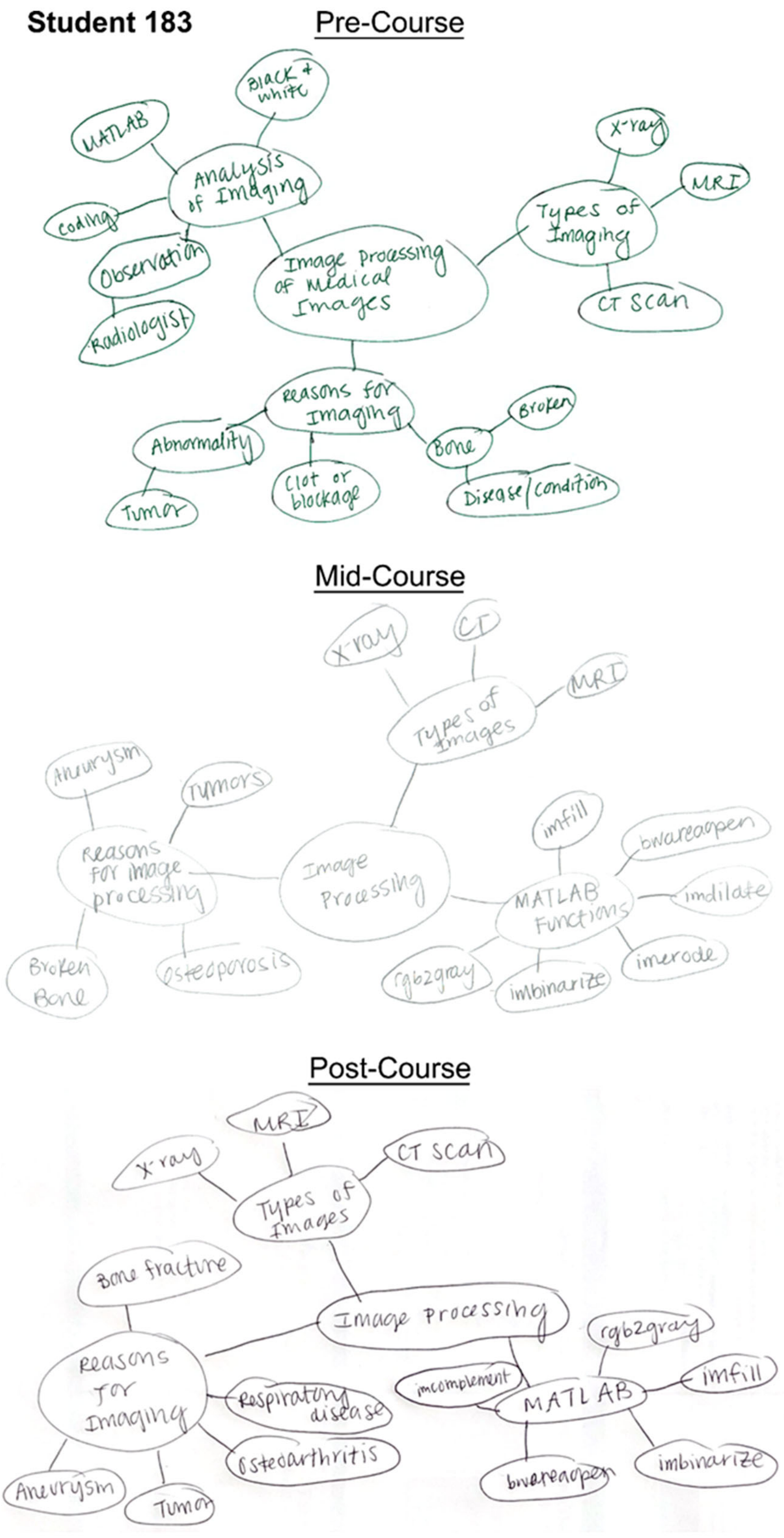

FIGURE 6. Concept maps from Tara, who demonstrated minimal growth in concept map holistic score pre- to post-module.

TABLE 9. Implementation of PBL components into the final design project.

Critical component of PBL	Implementation
Outcomes are well-defined	Project outcomes were defined based on rubric with clear specifications
Task is ill-defined to promote self-directed learning	Project focused on a body system and often required skills beyond what was covered in lecture and labs
Students work in cooperative groups to complete the task	Students worked in groups of 2-3
Instructors act as facilitators	Instructors asked questions and guided students toward resources rather than provide explicit instruction
Projects have real-life applications	Students used open-source radiology images from medical applications
Students engage in self-reflection	Students submitted a final report which critically evaluated their script's strengths and limitations

TABLE 10. Implementation of scaffolding into IntroCAD.

Critical component of scaffolding	Implementation
Contingency	Lectures, beginner- and advanced- level questions in lab, and discussions were tailored using survey responses and muddiest points; questions answered in class were based on responses to questions probing student background knowledge.
Fading	Instructors moved from providing explicit instruction to facilitating discussions; labs included progressively less detailed instructions ending with an open-ended final project.
Transfer of Responsibility	Students took more ownership of both discussions and project work as they transitioned from lectures and guided labs to discussions and an open-ended final project.

TABLE 11. Problems for additional project-based sections beyond what was covered in this module.

Problem focus	Skills	Conceptual knowledge
Using an ultrasound image to identify a breast tumor	Filtering, non-anatomical data	Breast cancer pathology
Analyzing a 3D CT scan of kidney stones	Analyzing and manipulating 3D images	CT scan properties, kidney stone pathology
Using machine learning to identify wrist fractures	Training and using neural networks	Neural network uses and function
Creating and exporting meshes from segmented knee MRIs	Segmenting, smoothing, mesh generation	Knee anatomy, mesh quality measures
Generalizing a script to scans from different people	Accounting for variability in scan acquisition and body structures	Sources of variability

through an explicit process, such as the problem-based learning cycle described by Hmelo-Silver.[15] In this process, students identify known data related to the problem, generate hypotheses or design ideas to address the problem, identify knowledge gaps, and lastly develop an action plan to cover those knowledge gaps and test their hypotheses or design ideas. When conducting this process, students write out key details on a white board or in a shared document to make the process more concrete. In the current iteration of this module, students conducted the self-directed learning process with *ad hoc*, unstructured guidance, which may have limited the types of solutions that they posed during the final project.

To promote abstraction from the specific problem and transfer to other contexts, more structured reflection opportunities could be given.[15] Currently, concept maps and written report were the only opportunities for reflection. However, concept maps were not graded, and the written report did not include any criteria related to reflecting on the knowledge acquired during the final project. Grading the concept maps would encourage students to take the assignment more seriously and could provide an opportunity to reflect on how conceptual knowledge grew because of the module.[25] A modified written report rubric could encourage students to explicitly identify the new conceptual and skills-based knowledge that they acquired and explain how this knowledge could be applied to different problems. Both modifications to the current module design would encourage critical reflection and abstraction away from the specific problem presented during the final project.

Furthermore, a study could be created to determine the effectiveness of the combination of educational strategies used in this innovation. The study would need to implement the innovation with more students and for a full-length course, since we describe a 4-week module with 13 students. A control course taught with traditional methodologies would also be needed. For example, a study could be run where the experimental group was taught using the design proposed in this innovation paper and the comparison group was taught using PBL only. Any differences between the two groups, focusing on conceptual knowledge gains and attitudes toward instructor support, would be caused the inclusion of brief lectures and guided labs. Such a study of the innovation described in this paper would enable scientific conclusions to be drawn.

There are multiple limitations to this paper. Since we created a four-week, one-credit module, it may not be directly transferable to other course formats. If other instructors were looking to implement the module described in this paper as a three- or four-credit course, we would recommend including additional project-based sections to address this limitation. The four-week sequence described in this innovation paper would provide students with a general introduction to image processing and to PBL. Students could then build on this introduction in subsequent project-based sections, which would require more complex conceptual and skills-based knowledge to address. This extended course format would enable students to move further beyond the introductory material than what was possible in this condensed, four-week format. Potential project topics for additional sections can be seen in Table 11.

Additionally, we sought to incorporate critical components of scaffolding in the module design and when answering student questions. However, we only collected data to evaluate the outcomes relevant to scaffolding rather than evaluate our implementation of the instructional strategy. Surveys should be adjusted to explicitly address the critical components of scaffolding. For a more detailed evaluation, class sessions could be recorded for later dialogue analysis.[31] If instructor support does not adhere to the critical components of scaffolding, additional training could be provided.[30]

This paper described the design, implementation, and evaluation of a module that combined multiple research-based instructional strategies. This was in response to multiple calls to apply the findings from engineering education research to engineering education practice. To heed this call, we identified and instructional strategies from that literature that were suitable for our learning objectives, incorporated those strategies into our design, and intentionally chose evaluation methods that would assess whether the instructional strategies would yield the benefits from literature. The module described in this paper was an image-processing-based computer programming module for biomedical engineering and shows the successful implementation of evidence-based practice from medical and engineering into computer science courses for engineers. We hope that this paper could serve as a guide for other biomedical engineering instructors when looking to incorporate research-based instructional strategies into their own course designs.

SUPPLEMENTARY INFORMATION

The online version contains supplementary material available at https://doi.org/10.1007/s43683-021-00057-w.

ACKNOWLEDGEMENTS

The authors would like to thank Keira Jeeyu Choi for her work on the development of the module course described in this paper. Recent work in several fields of science has identified a bias in citation practices such that papers from women and other minority scholars are under-cited relative to the number of such papers in the field.[6] We would like to acknowledge this citation bias in terms of both gender, racial, and ethnographic disparities. We hope to have appropriately cited references that reflect diversity from within the engineering education and biomedical engineering fields.

CONFLICTS OF INTEREST

N/A.

CONSENT TO PARTICIPATE

HUM00176990.

CONSENT FOR PUBLICATION

HUM00176990.

ETHICAL APPROVAL

Data collection was approved by Internal Review Board (HUM00176990).

AUTHOR CONTRIBUTION

RR and TSH co-developed the computer coding module and led the writing of the manuscript. AHS taught the Instructional Incubator, where the computer coding module was developed, and mentored the graduate students, RR and TSH, in the execution of the computer coding module. She also contributed to the evolution of the manuscript.

FUNDING

Funding for the design and development of the Instructional Incubator and resultant modules was provided by *NSF-EEC-1825669*.

DATA AVAILABILITY

N/A.

REFERENCES

[1] Baser, M. Attitude, gender and achievement in computer programming. *Middle East J Sci Res* 14(2):248–255.

[2] Bédard, D., C. Lison, D. Dalle, D. Côté, and N. Boutin. Problem-based and project-based learning in engineering and medicine: determinants of students' engagement and persistance. *Interdiscip J Probl Learn* 6(2):8.

[3] Besterfield-Sacre, M. E., J. Gerchak, M. Lyons, L. J. Shuman, and H. Wolfe. Scoring concept maps: an integrated. *J Eng Educ* 4:105–115.

[4] Bodnar, C., T. R. Christiani, K. Dahm, and A. J. Vernengo. Implementation and assessment of an undergraduate tissue engineering laboratory course. *Educ Chem Eng* 24:52–59.

[5] Borrego, M., J. E. Froyd, and T. S. Hall. Diffusion of engineering education innovations: a survey of awareness and adoption rates in U.S. engineering departments. *J Eng Educ* 99(3):185–207.

[6] Caplar, N., S. Tacchella, and S. Birrer. Quantitative evaluation of gender bias in astronomical publications from citation counts. *Nat Astron* 1(6):141.

[7] Capraro, R. M., and S. W. Slough. Capraro, slough-STEM PBL-integrated science Tech Eng Math approach-2013. Pdf. In: STEM Project-Based Learning: An Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach, edited by R. M. Capraro, M. M. Capraro, and J. Morgan. Rotterdam: Sense Publishers, 2013, pp. 1–5.

[8] Carberry, A., Krause, S., Ankeny, C., and Waters, C. Unmuddying' Course content using muddiest point reflections. In: Proceedings-Frontiers in Education Conference, pp. 937–942, 2013.

[9] Ellis, G. W., A. Rudnitsky, and B. Silverstein. Using concept maps to enhance understanding in engineering education. *Int J Eng Educ* 20(6):1012–1021.

[10] Felder, R. M., and R. Brent. Teaching and Learning STEM: A Practical Guide | Wiley. New York: Wiley-Blackwell Publishing Ltd, 2016.

[11] Frank, M., I. Lavy, and D. Elata. Implementing the project-based learning approach in an academic engineering course. *Int J Technol Des Educ* 13(3):273–288.

[12] Freeman, S., S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proc Natl Acad Sci USA* 111(23):8410–8415.

[13] Gavin, K. Case study of a project-based learning course in civil engineering design. *Eur J Eng Educ* 36(6):547–558.

[14] Von Glasersfeld, E. Cognition , construction of knowledge , and Teaching author (s): Ernst Von Glasersfeld Source: Synthese , Vol . 80 , No . 1 , History , Philosophy , and Science Teaching (Jul., 1989), Published by: Springer Stable URL: http:/wWww.Jstor.Org/St. *Synthese* 80(1):121–140.

[15] Hmelo-Silver, C. E. Problem-based learning: what and how do students learn? *Educ Psychol Rev* 16(3):235–266.

[16] Hsiung, C.-M. The effectiveness of cooperative learning. *J Eng Educ* 101(1):119–137.

[17] Huang-Saad, A., J. Stegemann, and L. Shea. Developing a model for integrating professional practice and evidence-based teaching practices into BME curriculum. *Ann Biomed Eng* 48(2):881–892.

[18] Jamieson, L. H., and J. R. Lohmann. Creating a Culture for Scholarly and Systematic Innovation in Engineering Education. Washington, DC: American Society of Engineering Education, 2009.

[19] Kolmos, A., and E. De Graaff. Problem-based and project-based learning in engineering education: merging models. *Cambridge Handb Eng Educ Res* 24:141–160.

[20] Krishnan, S. M. Academic program models for undergraduate biomedical engineering. In: 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC 2014, pp. 5145–5148, 2014.

[21] Litzinger, T. A., L. R. Lattuca, R. G. Hadgraft, and W. C. Newstetter. Engineering education and the development of expertise. *J Eng Educ* 100(1):123–150.

[22] Martínez, G., Á. L. Pérez, M. I. Suero, and P. J. Pardo. The effectiveness of concept maps in teaching physics concepts applied to engineering education: experimental comparison of the amount of learning achieved with and without concept maps. *J Sci Educ Technol* 22(2):204–214.

[23] Mayer, R., Moeller, B., Kaliwata, V., Zweber, B., Stone, R., and Frank, M., Educating engineering undergraduates: effects of scaffolding in a problem-based learning environment. In: Proceedings of the Human Factors and Ergonomics Society. pp. 2507–2511, 2012.

[24] Mills, J. E., and Treagust, D. F. Engineering education—is problem based or project based learning the answer?. Australasian Association for Engineering Education. pp. 1–16, 2014.

[25] Newstetter, W. C. Fostering integrative problem solving in biomedical engineering: the PBL approach. *Ann Biomed Eng* 34(2):217–225.

[26]Novak, J. D., and A. J. Cañas. Theoretical origins of concept maps, how to construct them, and uses in education. *Reflect Educ* 3(1):29–42.

[27]Novak, J. D., and D. B. Gowin. Learning How to Learn. New York: Cambridge University Press, 1984.

[28]Pleiss, G., Perry, M., and Zastavker, Y. V. Student self-efficacy in introductory project-based learning courses. In: Proceedings-Frontiers in Education Conference, pp. 8–13, 2012.

[29]van de Pol, J., M. Volman, and J. Beishuizen. Scaffolding in teacher-student interaction: a decade of research. *Educ Psychol Rev* 22(3):271–296.

[30]van de Pol, J., M. Volman, F. Oort, and J. Beishuizen. Teacher scaffolding in small-group work: an intervention study. *J Learn Sci* 23(4):600–650.

[31]van de Pol, J., M. Volman, F. Oort, and J. Beishuizen. The Effects of Scaffolding in the classroom: support contingency and student independent working time in relation to student achievement, task effort and appreciation of support. *Instr Sci* 43(5):615–641.

[32]Savin-Baden, M. 1960, Major, C. H., Education., S. for R. into H., ebrary, I., Savin-Baden, M., and Major, C. H., Foundations of Problem-Based Learning, Maidenhead, 2004.

[33]Segalàs, J., D. Ferrer-Balas, and K. F. Mulder. What do engineering students learn in sustainability courses? The effect of the pedagogical approach. *J Clean Prod* 18(3):275–284.

[34]Shekar, A. Project based learning in engineering design education: sharing best prac- tices project-based learning in engineering design educatio: sharing best practices. In: ASEE Annual Conference Proceeding, 2014.

[35]Sheppard, S. D., Macatangay, K., Colby, A., and Sullivan, W. M. Educating Engineers: Designing for the Future of the Field. 2009.

[36]Stanley, R. J., S. E. Watkins, and R. H. Moss. Integration of real-world problems into an image processing curriculum. *Int J Eng Educ* 21(1):318–326.

[37]Strobel, J., and A. van Barneveld. When is PBL more effective? A meta-synthesis of meta-analyses comparing PBL to conventional classrooms. interdiscip. *J Probl Learn*. https://doi.org/10.7771/1541-5015.1046.

[38]Tokdemir, G., and Cagiltay, N. E. A Concept map appraoch for introduction to computer engineering course curriculum. In: IEEE Global Engineering Education Conference, Madrid, pp. 243–250, 2010.

[39]Turns, J., C. J. Atman, and R. Adams. Concept maps for engineering education: a cognitively motivated tool supporting varied assessment functions. *IEEE Trans Educ* 43(2):164–173.

[40]Walker, J. M. T., and King, P. H. Concept mapping as a form of student assessment andinstruction. In: Proceedings-Frontiers in Education Conference, (April), pp. 7151–7166, 2002.

[41]White, J. A., D. P. Gaver, R. J. Butera, B. Choi, M. J. Dunlop, K. J. Grande-Allen, A. Grosberg, R. W. Hitchcock, A. Y. Huang-Saad, M. Kotche, A. M. Kyle, A. L. Lerner, J. H. Linehan, R. A. Linsenmeier, M. I. Miller, J. A. Papin, L. Setton, A. Sgro, M. L. Smith, M. Zaman, and A. P. Lee. Core competencies for undergraduates in bioengineering and biomedical engineering: findings, consequences, and recommendations. *Ann Biomed Eng* 48(3):905–912.

[42]Wiebe, E. N., L. Williams, K. Yang, and C. Miller. Computer science attitude survey. *Comput Sci* 41:5.

[43]Wood, D., J. S. Bruner, and G. Ross. The role of tutoring in problem solving. *J Child Psychol Psychiatry* 17(2):89–100.

[44]Zhang, X., C. Zhang, T. F. Stafford, and P. Zhang. Teaching introductory programming to IS students: the impact of teaching approaches on learning performance. *J Inf Syst Educ* 24(2):147–155.