Lee et al.
International Journal of STEM Education (2023) 10:52
https://doi.org/10.1186/s40594-023-00444-5

International Journal of STEM Education

RESEARCH Open Access

Precision education via timely intervention in K-12 computer programming course to enhance programming skill and affective-domain learning objectives

Hsin-Yu Lee[1], Chia-Ju Lin[1], Wei-Sheng Wang[1], Wei-Cyun Chang[1] and Yueh-Min Huang[1*]

Abstract

Background In the realm of Science, Technology, Engineering, and Mathematic (STEM) education, computer programming stands as a vital discipline, amalgamating cross-disciplinary knowledge and fostering the capacity to solve real-world problems via fundamental concepts and logical methodologies inherent to computer science. Recognizing the important of computer programming, numerous countries have mandated it as a compulsory course to augment the competitiveness of K-12 learners. Nevertheless, the inherent complexity of computer programming for K-12 learners often goes unacknowledged. Constraints imposed by the course format, coupled with a low instructor–learner ratio, frequently inhibit learners' ability to resolve course-related issues promptly, thereby creating difficulties in the affective domain. While precision education tools do exist to ascertain learners' needs, they are largely research-oriented, thereby constraining their suitability for deployment in pragmatic educational settings. Addressing this issue, our study introduces the precision education-based timely intervention system (PETIS), an innovative tool conceived to enhance both programming skills and affective learning in K-12 learners. Our research investigates the influence of PETIS on learners' performance and evaluate its efficacy in facilitating computer programming education in K-12 environments.

Results Quantitative results demonstrate that the application of the precision education-based timely intervention system (PETIS) proposed by this research significantly improves programming skills and affective-domain learning objectives for K-12 learners. Similarly, qualitative results indicate that PETIS is beneficial for both teaching and learning in K-12 computer programming courses.

Conclusions These results not only confirm that timely intervention and feedback improve K-12 learners' programming skills and affective-domain learning objectives in computer programming courses, but also yield implications as to the feasibility of applying precision education in real-world STEM scenarios.

Keywords STEM education, Precision education, Affective domain, K-12 computer programming

Introduction

Computer programming is a critical component of Science, Technology, Engineering, and Mathematics (STEM) education, intertwining cross-disciplinary knowledge and enhancing learners' capacity to address real-world problems through the principles and methodologies of computer science (Gao et al., 2020; Hao et al., 2023; Sun et al.,

*Correspondence:
Yueh-Min Huang
huang@mail.ncku.edu.tw
[1] Department of Engineering Science, National Cheng Kung University, No. 1, University Road, Tainan 701, Taiwan, ROC

2021). Its significant role in promoting cognitive development, enhancing problem-solving skills, and preparing learners for a technology-driven future has received substantial recognition (Ouyang et al., 2022; Sung et al., 2023). Consequently, countries like Taiwan and the United Kingdom have integrated it as a compulsory subject within the K-12 curriculum to foster the cultivation of higher-order cognitive skills such as problem-solving (Tsai et al., 2020). Yet, most research contends that programming presents substantial challenges for K-12 students (Perera et al., 2021; Raj et al., 2018), predominantly due to affective domain deficiencies during the programming learning process (Medeiros et al., 2018).

To mitigate the effects of these deficiencies, pedagogical strategies such as game-based learning, unplugged programming, and immersive learning have been introduced to K-12 computer programming education. These methods aim to improve students' affective domain, thus facilitating positive learning outcomes (Lindberg et al., 2019). They simplify the programming process and introduce engaging elements, fostering motivation and engagement, thereby altering perceptions of programming as a tedious task (DemİRkiran & Tansu Hocanin, 2021; Zhao et al., 2022).

Beyond these methods, Medeiros et al. (2018) underscore the significance of instructor–learner interaction and feedback for improving the affective domain in K-12 computer programming education. They posit that instructors who can swiftly identify and address student difficulties, offering constructive feedback, significantly enhance the affective domain of programming learners—a perspective confirmed by Pordelan et al. (2020). Yet, some research highlights that, particularly in Asia, educators often find it challenging to provide timely assistance due to a low instructor-to-learner ratio and learners' reticence, thereby impeding the efficacy of computer programming education (Medeiros et al., 2018). Yiu (2013) corroborates that K-12 learners in Asia generally maintain the most distance from their instructors, often refraining from asking questions due to shyness or concern about peer judgment (Rapee et al., 2011).

In response, researchers have begun to incorporate precision education into curricula, enabling efficient identification of learner difficulties (Cook et al., 2018; Luan & Tsai, 2021; Yang, 2021). Leveraging artificial intelligence and learning analytics, precision education diagnoses and predicts learner performance, formulating targeted learning strategies to provide timely assistance to struggling students (Tempelaar et al., 2021). However, a systematic review by Luan and Tsai (2021) reveals a dearth of studies on the application of precision education in K-12 computer programming courses. Moreover, Gao et al. (2020) argue that the analytic tools employed in precision education are often research-oriented and lack practical applicability in classroom environments.

It is widely acknowledged that cognition and the affective domain are interlinked (Jeong et al., 2021; Lee et al., 2023; Makransky & Petersen, 2021; Wu et al., 2022). Bloom's taxonomy identifies the affective domain as a paramount learning objective, emphasizing the internalization of values concerning learners' feelings, emotions, attitudes, interests, and motivations towards people, events, and objects (Bloom, 1956). However, the majority of existing research primarily explores the cognitive aspect of learners, overlooking the influence of the affective domain (Cheng et al., 2022; Noroozi et al., 2020). While a few studies do consider the affective domain, they mostly focus on learner emotions and seldom scrutinize the affective domain in relation to Bloom's taxonomy levels (Yadegaridehkordi et al., 2019; Yun & Cho, 2021).

Based on Bloom's definition of the affective domain, we employ learning motivation, attitude, and self-efficacy as indicators to examine changes in learners' affective domain in this study. Learning motivation fluctuates based on whether students are passively receiving knowledge (Receiving) or actively participating in discussions (Responding). Learning attitude refers to learners' feelings or opinions regarding certain concepts, which contribute to the development and internalization of values associated with these concepts (Valuing and Organizing). Self-efficacy reflects learners' beliefs about their abilities and the outcomes of their efforts, with high self-efficacy indicating that concepts have been internalized as part of the learners' identity (Characterizing).

In conclusion, while the introduction of precision education has demonstrated a positive impact on learning, its effect on the learners' affective domain remains less understood. Consequently, this study will precisely define indicators for affective-domain learning objectives to elucidate the relationship between these and cognition. To this end, we have developed the precision education-based timely intervention system (PETIS), leveraging deep learning and image processing technologies to assist instructors in identifying when K-12 learners require support during computer programming courses. A quasi-experimental design is adopted to validate the use of PETIS for enhancing K-12 learners' programming skill and affective-domain learning objectives. We pose the following research questions:

1. How significantly does the application of PETIS influence the programming skill of K-12 learners?
2. To what extent does PETIS affect the affective-domain learning objectives of K-12 learners?

3. How significantly do affective-domain learning objectives influence K-12 learners' programming skill when PETIS is integrated into the computer programming course?
4. Is PETIS a potent and beneficial instrument for advancing K-12 programming education?

Related work

Programming for K-12 STEM education

Some studies underscore the importance of cultivating programming skills among K-12 learners (Lee et al., 2020; Xu et al., 2019). Consequently, several nations have incorporated programming as a mandatory component of their K-12 curriculum to enhance students' problem-solving abilities (Yun & Cho, 2021). As an integral element of STEM education, programming equips students with the aptitude to tackle real-world issues by utilizing basic computer science concepts and logical methodologies (Gao et al., 2020; Sun et al., 2021). Nevertheless, the syntactic and semantic complexities of programming languages present a significant hurdle to K-12 learners (Medeiros et al., 2018; Perera et al., 2021). Several studies have attempted to address this by incorporating varied teaching methodologies into K-12 computer programming instruction. For example, Xu et al. (2019) have corroborated the efficacy of visual programming teaching methods in achieving cognitive and affective learning outcomes. Similarly, Panskyi et al. (2019) have implemented game elements in computer programming coursework, demonstrating a boost in students' computational thinking, problem-solving, and abstract thinking capabilities. Moreover, Sun et al. (2021) have amalgamated unplugged game elements into the programming syllabus, revealing enhancements in students' knowledge, behavior, and attitudes through a mixed method.

In essence, contemporary research prioritizes the amalgamation of various instructional elements to alleviate the learning burden for K-12 learners. Importantly, Medeiros et al. (2018) in their systematic review highlight the crucial role of teacher–student interaction and feedback in computer programming courses. However, few studies probe the influence of apt and timely teacher feedback on K-12 computer programming instruction. While prompt and suitable teacher feedback has been shown to enhance cognitive performance considerably, its impact on the affective learning domain of K-12 learners remains largely unexplored. To bridge this research gap, we propose PETIS—a mechanism for instructors to identify student difficulties promptly, provide immediate feedback, and thereby investigate the impact of timely teacher feedback on the affective learning outcomes in K-12 programming education.

Affective domain in STEM education

Bloom's taxonomy posits three learning domains in education: cognitive, affective, and psychomotor. This theory emphasizes that effective instruction should incorporate strategies to foster cognitive, emotional, and behavioral development in learners (Bloom, 1956; Wu et al., 2019). Specifically, the affective domain, which centers on learners' emotional responses, attitudes, and values, is of great significance in the pedagogical process. According to Bloom's taxonomy, this domain comprises five levels of internalization: receiving, responding, valuing, organizing, and characterizing (Bloom, 1956; Kranch, 2012; Wu et al., 2019), as described below:

- Receiving: This concerns the learner's perception of a phenomenon or stimulus and the selection of the stimulus to attend to. At this level, the learner is willing to listen to the voices of others.
- Responding: Once the learner has mastered receiving, he or she actively participates in discussions and asks questions to show others what knowledge and information he or she has.
- Valuing: The learners are aware of the value of a phenomenon, thing, or action. Once they have mastered receiving and responding, they develop their own values that inform how they use their thinking to take action.
- Organizing: The learners internalize values. By comparing different perspectives and creating their own unique system, they evaluate what is happening according to their own values.
- Characterizing: The learners internalize the values at the organizational level, which then become part of their personalities and begin to become a philosophy of life. The learners act according to their resultant unique affective system.

The importance of the affective domain is increasingly recognized in STEM education, which emphasizes learner-centered, hands-on activities and peer collaboration (Guzey et al., 2016). Affective outcomes such as interests, attitudes, motivation, and values have been extensively explored in STEM research (Gao et al., 2020). De Loof et al. (2021) argue that affective outcomes such as motivation, learning attitudes, and self-efficacy are vital in STEM education.

Learning motivation, as defined by Lin and Chen (2017), is dependent on the learner's personal perception of learning. We categorize it as a lower-level affective outcome, influenced by whether learners are passively receiving knowledge (Receiving) or actively participating in discussions (Responding). Tzafilkou et al. (2021) posit that learning attitude, which encapsulates the learners' feelings or opinions about specific concepts (Valuing), is a mid-level affective outcome that emerges once learners have internalized their personal value of a concept (Organizing). Lastly, Bandura (1977) suggests that self-efficacy, reflecting learners' confidence in their skills and the outcomes of their efforts, is a high-level affective outcome apparent when concepts have been integrated into learners' personalities (Characterizing).

This trajectory from initial receptivity to active engagement, formation of values, assimilation of these beliefs, and final internalization reflects the progression of affective development in STEM learners (Krathwohl, 2002; Sharunova et al., 2022). Motivation is typically initiated in the early stages (Receiving and Responding), learning attitudes are cultivated during the intermediate phase (Valuing and Organizing), and self-efficacy emerges in the advanced stage (Characterizing). Given our research goal to measure these three components, Bloom's taxonomy serves as a valuable framework for assessing the effects of strategic pedagogical interventions in STEM activities. It enables us to appraise the efficacy of various teaching strategies at each level of the affective domain, thereby informing the development of more effective interventions to enhance motivation, learning attitudes, and self-efficacy in STEM education.

In view of these trends and theories, we employ questionnaires to assess the impact of timely interventions in K-12 computer programming courses on learner motivation, attitude, and self-efficacy as key affective-domain learning objectives.

Precision education

Traditional education has often been characterized by a 'one-size-fits-all' approach, constraining individualized learning due to low instructor–learner ratios and rigid educational policies. Consequently, instructors are typically unable to adopt pedagogical strategies that are responsive to the unique learning styles of each student (Cook et al., 2018; Hu, 2022; Snow, 1986).

However, recent advancements in artificial intelligence (AI) and educational data mining technologies have drawn attention to the potential of precision education (Hu, 2022). Borrowing from precision medicine, which considers each patient's unique genetic makeup, living environment, and lifestyle for targeted prevention and treatment (Collins & Varmus, 2015), precision education endeavors to accommodate individual differences in learning. By leveraging AI and educational data mining, precision education can provide prompt, individualized intervention aimed at enhancing learning effectiveness, thus mitigating the limitations of the conventional educational model (Luan & Tsai, 2021; Yang, 2021).

Existing research underscores the efficacy of precision education in enhancing learners' outcomes. For instance, Qushem et al. (2021) reported that blending precision education into online learning paradigms positively impacts student performance, achievement, and well-being. This is primarily attributed to the ability of precision education to optimize the pedagogical potential of educational platforms and tools, thereby facilitating knowledge acquisition and skills development. Similarly, Liu (2022) deployment of the Taiwan Adaptive Learning Platform (TALP), a precision education tool, demonstrated improvement in student engagement and math performance through precise identification of learners' knowledge gaps, provision of diverse learning resources, and delivery of feedback.

Precision education aims to identify learners at risk of distraction early and to provide timely interventions through prediction, diagnosis, prevention, and treatment. For example, Tsai et al. (2020) developed a precision education model integrating deep learning and educational statistical analysis to *predict* the dropout rates of university students in Taiwan. It was determined that factors such as student loan applications, absenteeism, and subjects of concern significantly impacted dropout rates. Lee et al. (2023) applied deep learning and image recognition to *diagnose* learners' behaviors and processes, revealing correlations between learning processes and performance in STEM education.

Nonetheless, there is a noticeable gap in the field. Luan and Tsai (2021) noted that few existing tools effectively integrate all precision education objectives, with most only capable of predicting learner performance. Moreover, Gao et al. (2020) indicated that many analytical tools in precision education are research-oriented, conducting post-event analysis, which does not facilitate real-time interventions in classroom scenarios. To address these deficiencies, we propose PETIS. This tool utilizes image processing and deep learning to discern when K-12 students encounter difficulties during computer programming courses. Moreover, PETIS offers an intuitive interface, integrating diagnostic, therapeutic, and preventative functions, thereby addressing the limitations of existing tools.

ICAP framework

Chi and Wylie (2014) have proposed a distinctive framework known as ICAP, which categorizes learning engagement into four discrete modes. This innovative model provides a foundation to map learner behaviors, thereby assisting in understanding the nature of engagement leading to changes in learner's knowledge. The ICAP framework specifically classifies learning engagement into the following quadrants:

- Passive: Learners in this mode receive information passively from educational resources without significant engagement in their learning trajectory (Chi & Wylie, 2014).
- Active: This mode involves learners demonstrating observable behavior and actively engaging in physical interactions with learning materials (Chi & Wylie, 2014).
- Constructive: Learners in this category generate or produce externalized outputs or products that extend beyond the learning materials provided, signifying an active creation of new knowledge (Chi & Wylie, 2014).
- Interactive: The definition of interactive behaviors is operationalized through two principal criteria: the learners' utterances must be predominantly constructive and there must be a significant amount of turn-taking evident in their interaction (Chi & Wylie, 2014).

The practicality of the ICAP framework's methodology facilitates a comprehensive understanding of learner engagement in the context of instructional courses. To further this approach, we have developed a system, PETIS, that is rooted in the ICAP model. It aims to provide educators and academics with profound insights into student engagement in STEM education, thereby establishing a solid foundation for an evaluative mechanism in this field.

The design of PETIS

According to Yang (2021), the objectives of precision education are to identify at-risk learners as early as possible and provide a timely intervention through prediction, diagnosis, and treatment. We present the Precision Education based Timely Intervention System (PETIS) developed in alignment with this definition of precision education. As outlined in Fig. 1, PETIS enables instructors to understand learners' individual learning engagement by evaluating their learning behaviors (diagnosis), offering timely feedback or guidance (treatment) to address the deficiencies in traditional K-12 computer programming courses. Furthermore, the system allows instructors to modulate the lesson's difficulty by reviewing each learner's record (prevention). Lastly, we investigate whether the implementation of precision education enhances K-12 learners' learning outcomes in both the affective domain and programming skills. Further details about the design of PETIS are presented below.

As noted by Scott and Ghinea (2014), K-12 learners often experience a sense of helplessness when encountering difficulties during programming lessons. Therefore, learners who are stagnating or inactive for an extended period are likely to be facing challenges. Moreover, recent findings from Lee et al. (2023) suggest that the interaction between learners' hands and learning materials can serve as an insightful indicator to understand the learning engagement in STEM education. Consequently, PETIS integrates image processing and deep learning

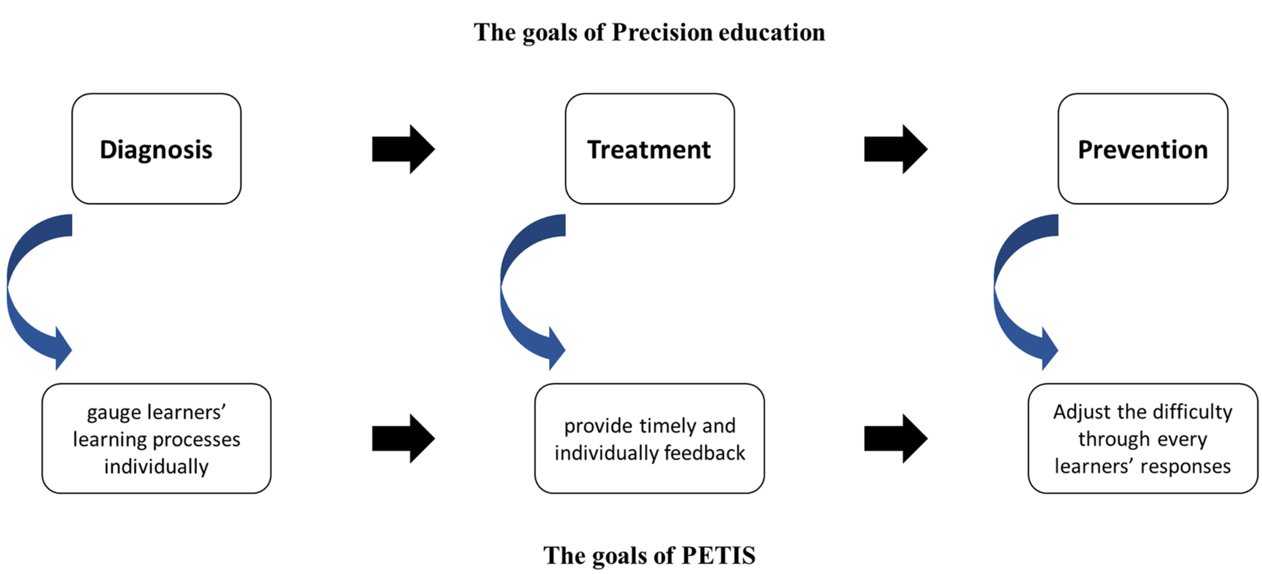

Fig. 1 The goals of precision education and PETIS

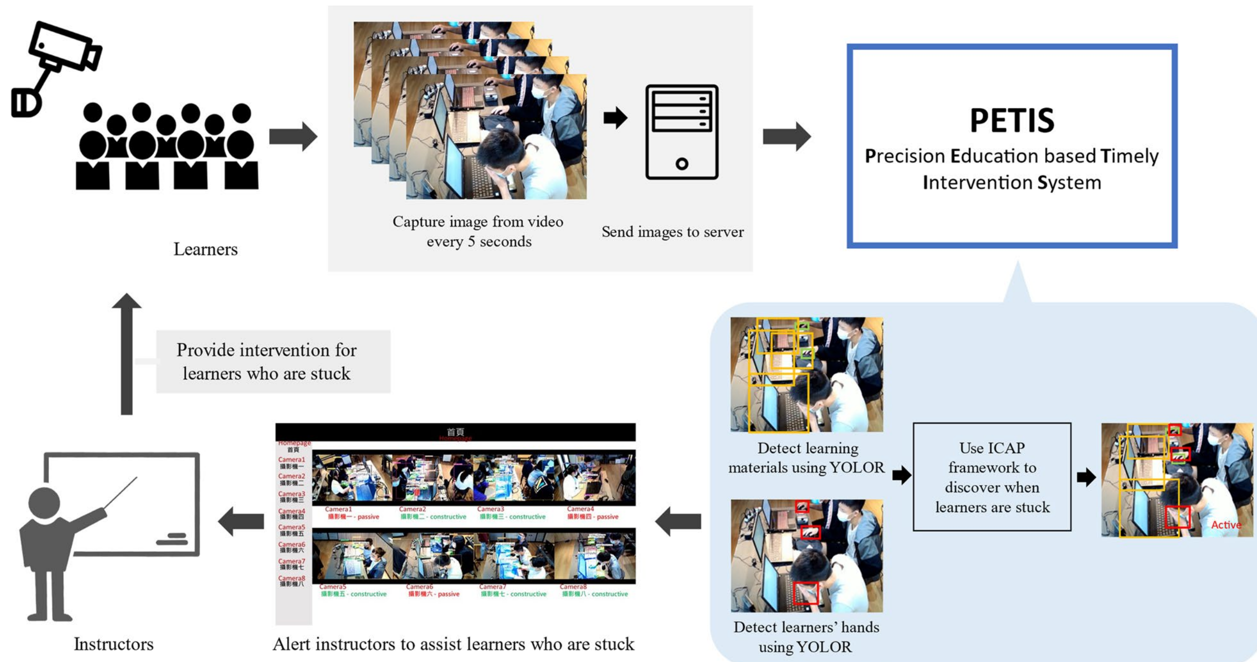

Fig. 2 The workflow of PETIS

technologies to detect instances where learners may be struggling, as discerned through the interaction between their hands and the learning materials. The system architecture is depicted in Fig. 2, with each section elaborated further below.

Data collection

In an endeavor to amass a sufficient dataset to construct the system, four graduate-level teaching assistants acted out all conceivable behaviors during a computer programming course, all the while capturing their actions with a video camera. In an effort to mitigate the risk of overfitting, each teaching assistant produced between seven and eight videos at varied intervals (Lin et al., 2021a, 2021b). Each video was recorded at a resolution of 1920 × 1080, capturing 30 frames per second (fps). The specific camera angle employed for recording is depicted in Fig. 3. The collected corpus consisted of 30 simulated videos. From these videos, still images were extracted at 5-s intervals, intended for use as training data for the model (Lee et al., 2023; Sun et al., 2021).

Fig. 3 Camera angle during data simulation

Precision education-based timely intervention system (PETIS)

The construction of PETIS

In the construction of the PETIS, the initial step necessitates the cloning and establishment of the YOLOR project as proposed by Wong (2022). The operating system utilized is Ubuntu 20.04, complemented by Python and Pytorch, versioned 3.9 and 1.7.0, respectively. For a comprehensive understanding of PETIS's functionality, a pseudo-code is provided in Table 1. The initial phase of the PETIS operation entails the extraction of the learners' hands and learning materials using the YOLOR, performed every 5 s. This serves as the fundamental input in understanding the learning engagement through the ICAP framework. On the collection of 60 data points, corresponding to 5 min, PETIS decides whether the learners are stuck or not, and subsequently records both video and statistical data on the dedicated webpage. Based on this data, instructors can make informed decisions on whether immediate feedback and assistance are required.

Table 1 The pseudo-code of PETIS

```
Algorithm: PETIS
    Input: Video V
    Output: ProvideFeedback P, where P = 0 or 1
initial BehaviorArray B
repeat
    Capture an Image I from V every 5 seconds
    Detect learning materials M by YOLOR
    Detect learners'hands H_i by YOLOR, where i is the number of learners
    if H_i overlap on M then
        if M = tablet then
            B.append("Active")
        if M = laptop, mouse, raspberry pi then
            B.append("Constructive")
        else
            B.append("Other")
        endif
    else
        B.append("Passive")
    endif
    if H_i overlapped then
        B.append("Interacitve")
    endif
    if the length of B = 60 then
        Output the MostBehavior MB in B
        if MB = "Active" or "Constructive" or "Interactive" then
            return P = 0
        else
            return P = 1
        endif
        Save B and I
        Clean B
    endif
until the end of the V
```

Detecting learning materials and learners' hands using YOLOR

We have employed the YOLOR model, as detailed by Wang et al. (2021), to detect the learner's hands and learning materials. The YOLOR model learns generic representations by integrating implicit and explicit knowledge, enhancing model performance and facilitating reasoning across multiple computer vision tasks

Table 2 YOLOR training parameters

Parameter	Batch size	Epochs	Height	Width	Class
Value	16	300	640	640	6

Fig. 4 YOLOR detection results

with a significant reduction in parameters and computational effort compared to previous models (Wang et al., 2021). However, as learning materials commonly used in computer programming courses were not included in the pre-training phase of YOLOR, we utilized transfer learning to re-train YOLOR, enabling it to recognize the common learning materials used in these courses. The training parameters for YOLOR are detailed in Table 2.

The retrained YOLOR is now capable of recognizing six common objects in computer programming courses: hand, tablet, laptop, mouse, Raspberry Pi, and cellphone, as depicted in Fig. 4.

Detecting learner difficulty and facilitating timely intervention

Our proposal for the identification of learners who may be struggling in class involves a timely intervention mechanism. This is based on the interaction between hands and learning materials in the captured images. Such interactions are utilized to assess the learners' progress and to identify those who may be stuck. Building upon the work of Lee et al. (2023), we have integrated the ICAP framework into PETIS. This framework serves as an indicator of the learning engagement, determining whether learners are stuck. The relationship between these learning engagement indicators and a student's progress is illustrated in Table 3. An example of an active indicator is a learner touching a tablet to read the learning materials, indicating that the learner is likely not stuck.

The PETIS records the learning process every five seconds and exports these learning engagement indicators in CSV format every 5 min (totaling 60 records). It also counts the instances when a learner is stuck within these five minutes. If the learner is stuck for more than half of this time (more than 30 records), an assumption is made that timely intervention is required.

Interface design

Figure 5 illustrates the interactive interface we devised, designed to indicate to instructors whether students required assistance within the preceding 5-min interval, as symbolized by the color-coding of the text—green for do not need assistance, red for assistance needed.

Instructors can access group options located on the left of the screen, leading them to the group page. This page presents the high-resolution video of recognition results (as shown on the left-hand side of Fig. 6), the latest indicators of each student's learning process (top right), and a comprehensive breakdown of the learning engagement (bottom right).

Table 3 Relationship between learning engagement indicators and difficulties

Is the learner stuck?	Indicator	Definition	Material being manipulated
No	Active	• Learners actively operates tablet containing learning materials to solve their questions instead of passively receiving knowledge from instructors	• *Tablet*
	Constructive	• Learners use laptop and mouse to complete programming Learners assemble Pi components to solve task	• *Laptop* • *Mouse* • *Raspberry Pi*
	Interactive	• Learners ask peers questions when they encounter difficulty while constructing knowledge	• *Hands*
Yes	Passive	• Learners' hands do not touch objects related to programming course	
	Other	• PETIS fails to recognize learners' hands • Learners use cellphone to do something unrelated to STEM workshop	• *Cellphone*

Fig. 5 Main user interface

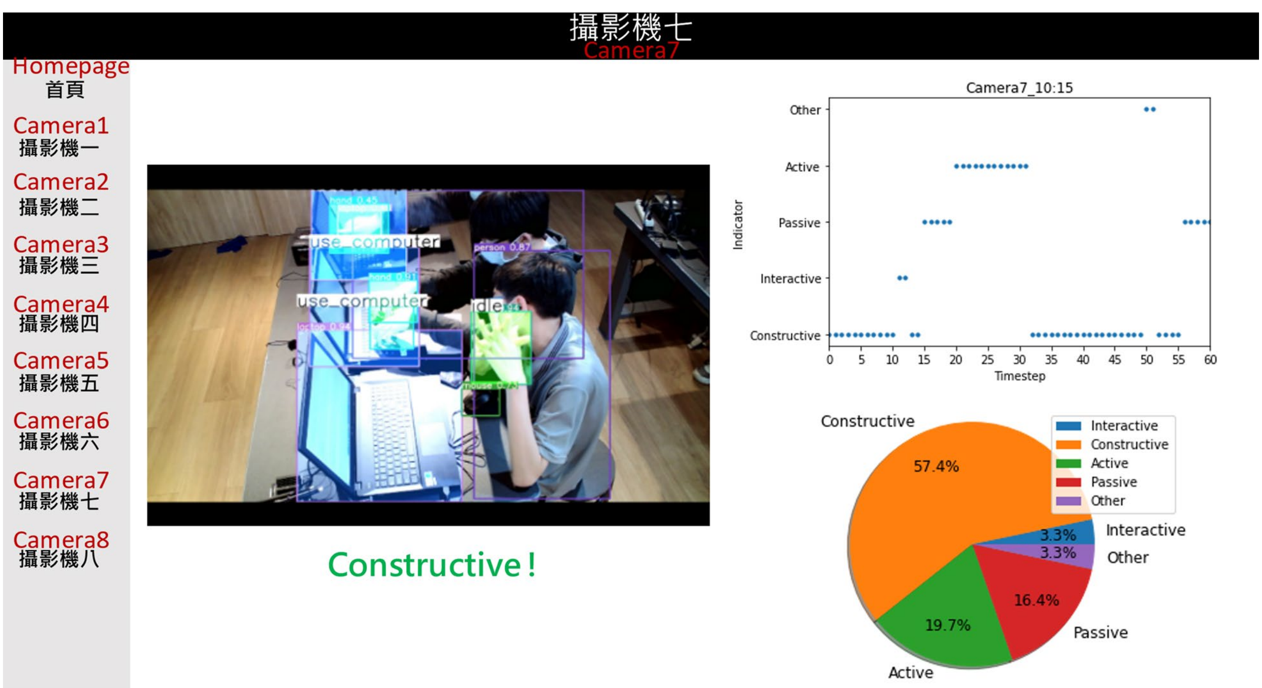

Fig. 6 Group page

Our system, PETIS, therefore delivers not only an incisive analysis of learners' impediments, but also identifies the parts of the curriculum that learners find challenging. This information could be employed by instructors as a foundation for future curricular improvements.

Methodology

Participants

We organized an exploratory activity termed "STEM Workshop: Introduction to Python Programming", where we recruited 60K-12 participants (34 males and 26 females) from a central Taiwan high school. The participants were randomly allocated to either the experimental group (EG) or the control group (CG). In the EG, consisting of 16 males and 14 females, instructors and teaching assistants (TAs) offered real-time intervention and assistance predicated on PETIS results. Conversely, in the CG, with 18 males and 12 females, intervention and assistance were dispensed based on instructor's and TA's discretion, primarily when they perceived learners to be struggling. Participants were duly informed about the experiment and data collection permissions were secured before commencement. To mitigate downtime, both groups shared the same set of instructors and TAs. To further maintain the equivalence, both EG and CG were recorded and analyzed; however, the analyzed information was exclusively provided to EG instructors for swift intervention, while CG instructors depended on their pedagogical experience for learner assistance.

Experimental procedure

We employed a quasi-experimental design and semi-structured interviews to ascertain whether PETIS enhanced learners' programming skills and impacted affective-domain learning objectives, including learning motivation, learning attitudes, and self-efficacy. The experimental procedure is depicted in Fig. 7. The EG and CG undertook identical activities throughout the experiment, with the exception that in the EG, instructors and TAs utilized PETIS to identify instances where learners were struggling, while in the CG, they relied on their professional judgment or awaited student-initiated assistance requests. In the EG, the instructors could actively monitor the learners' progress with PETIS support. Meanwhile, in the CG, instructors awaited student queries passively.

At the onset of the experiment, participants completed a 10-min pre-test to measure their prior programming skill. Subsequently, they spent 30 min familiarizing themselves with the development environment and Python principles. The foundational Python syntax and programming logic were presented to participants, with each lesson encompassing 20 min of instruction and 20 min of practice. In the EG, PETIS was employed during practice sessions to aid instructors and TAs in recognizing when learners were experiencing difficulties. Following the lessons, learners applied their newly acquired skills to complete the Raspberry Pi code by filling in the blanks to successfully run the code. This constituted tangible learning, where participants were expected to solidify the intangible programming concepts acquired in instructional activities by manipulating the tangible Raspberry Pi (Marshall, 2007). Upon conclusion of the activities, programming skills post-tests and post-questionnaires regarding learning motivation, attitudes, and self-efficacy were administered. Researchers then conducted semi-structured interviews with the instructors, TAs, and learners.

Research tools

Programming skill pre- and post-tests

We administered pre- and post-tests to record the participants' Python programming skills. These tests, each comprising 20 single-choice questions worth five points apiece, were developed by two esteemed professors of computer science. To affirm the validity and appropriacy of the test, we ascertained the internal consistency (Cronbach's α) to be 0.75—a value high enough to yield reliable results (Nunnally, 1978).

Scale of affective-domain learning objectives

In the absence of established scales for measuring affective-domain learning objectives, we adopted and adapted questionnaires measuring learning motivation, learning attitude, and self-efficacy to gauge the low- to high-level objectives delineated in section "Affective domain in STEM education". We selected four items from the Situational Motivation Scale (SIMS) by Guay et al. (2000) to measure low-level affective-domain learning objectives. We extracted four items from the Learning Computer Programming Attitude Scale (LeCoPAS) by Cetin and Ozden (2015) to measure mid-level objectives. Lastly, we chose four items from the New General Self-Efficacy Scale by Chen et al. (2001) to measure high-level objectives. These 12 items were assessed using a 5-point Likert scale, ranging from 1 (strongly disagree) to 5 (strongly agree). Validity and reliability tests on each scale level were performed to verify the appropriateness of our study. Table 4 presents these results. All three levels of affective-domain learning objectives exhibited robust validity and reliability, with alpha values of 0.732, 0.783, and 0.791 for low-, mid-, and high-levels, respectively, endorsing this as a credible measure of affective-domain learning objectives.

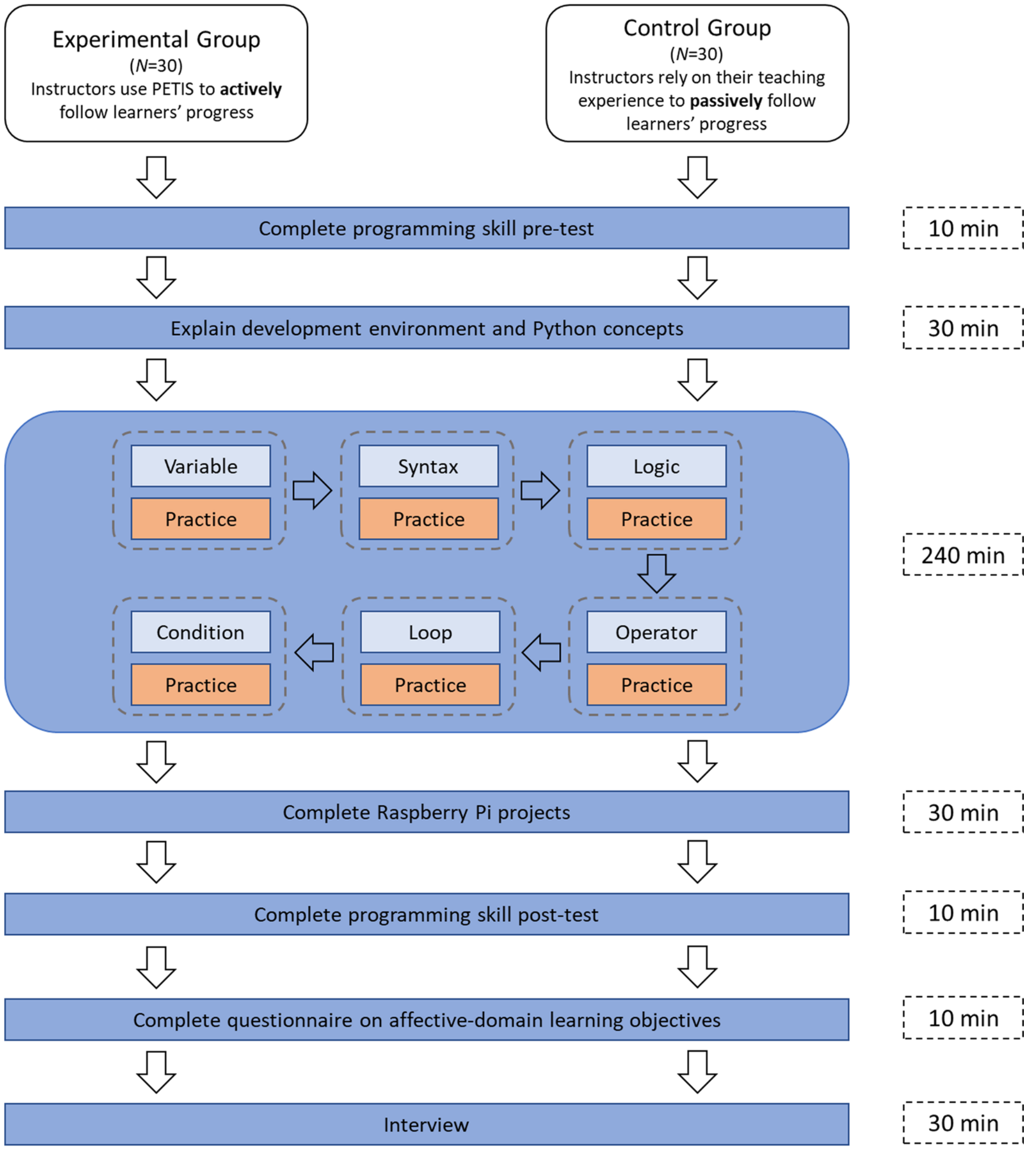

Fig. 7 Experimental procedure

Semi-structured interviews and thematic analysis

Post-activity, we selected six participants each from the experimental and control groups and two teachers (including teaching assistants) for semi-structured interviews. The learner interviews aimed to gauge intervention immediacy during the activity, knowledge retention, and the learners' emotional experience during the learning process. The instructor interviews concentrated on assessing the usability of PETIS, its ease of operation, and proposals for future enhancements. The interview questions were as follows:

Learners:

- How often do you think instructors provide assistance during the activity?
- Can you recall what you learned during the activity?
- What do you think was the most impressive part of the activity?
- After this activity, what are your future plans for programming?

Table 4 Validity and reliability of affective-domain learning objectives

Item	Validity	Reliability
Low-level affective domain (Learning motivation)	0.643	0.732
• I think the STEM Workshop: Introduction to Python Programming was interesting		
• I think the workshop activity will prove helpful for my future		
• I think the workshop is necessary for me		
• I think attending the workshop was a good use of time		
Mid-level affective domain (Learning attitude)	0.675	0.783
• I think programming is a distinct skill		
• I think programming can make human life more convenient		
• I will do much research to better understand programming		
• I will work hard to become a better programmer		
High-level affective domain (Self-efficacy)	0.712	0.791
• I can achieve the goals I set for myself during the workshop		
• When I encounter difficulty in the workshop activity, I know I will be able to resolve it		
• Compared to others, I complete the tasks in the workshop better than others		
• I feel confident that I will be able to effectively complete the different tasks in the workshop activity		

Instructors:

- How do you think the PETIS helps in teaching?
- What suggestions do you have for PETIS' functions?

Data analysis

To clearly address research question 1, we must first verify the homogeneity of the programming skill pre-tests given to the EG and CG to ensure the effectiveness of subsequent analysis. After confirming homogeneity, we use ANCOVA to identify significant differences between the EG and the CG in terms of post-test programming skill. The programming skill pre-test score was used as a covariate to eliminate the influence of pre-test differences on the post-test significance. To clearly address research question 2, we adopted an independent sample t-test to identify significant differences in participants' affective-domain learning objectives between the EG and CG. To clearly address research question 3, we calculated the Pearson correlation coefficient to examine the relationship between each level of affective-domain learning objectives and the post-test programming skill for all participants. To clearly address research question 4, we adopted six thematic analysis steps to analyze the interview content (Cohen et al., 2000): data formatting, separate data coding by each coder, recording specific coded data segments, comparing segments with like codes, code integration, and double-checking the final coded themes. We then converted the frequencies of the qualitative codes into quantitative data, as shown in tables and graphs.

Results

Impact of the timely intervention through PETIS on programming skill

To evaluate whether a significant difference exists between the experimental group (EG) and control group (CG) in terms of programming skill, we utilized analysis of covariance (ANCOVA) with the pre-test scores serving as the covariate and the post-test scores as the independent variable. This statistical methodology allowed

Table 5 ANCOVA for programming skill

	Sum of squares	*df*	Mean squares	*F*	*p*	Partial η^2
Pre-test scores	777	1	777	4.69	**0.034***	0.076
Group	3382	1	3382	20.41	**< 0.001*****	0.264
Error	9443	57	166			

*$p < 0.05$, ***$p < 0.001$

Bold values represent significant difference us to scrutinize the disparities in post-test programming skill scores between the EG and CG while mitigating the influence of pre-existing knowledge differences. Before employing ANCOVA for data analysis, we conducted a Levene's test to confirm the homogeneity of variances between the EG and CG. The results demonstrated that the measured variances did not have a significant impact ($F=0.549$, $p=0.462>0.05$), thereby supporting the use of ANCOVA. Table 5 outlines the ANCOVA results, revealing a significant discrepancy in post-test programming skill scores between the two groups ($F=20.41$, $p<0.001$). Upon comparing the group means, it was evident that the EG ($M=64.8$, SD = 11.7) significantly outperformed the CG ($M=50.3$, SD = 14.7) in post-test programming skill. Hence, it can be inferred that the application of PETIS for timely identification and assistance of learners' needs can significantly enhance their programming skill compared to relying solely on instructors' spontaneous judgments.

Impact of the timely intervention through PETIS on affective-domain learning objectives

We further scrutinized significant differences in affective-domain learning objectives between the EG and the CG, applying an independent sample t-test. As depicted in Table 6, there is a significant difference in overall affective-domain learning objectives between the two groups ($t=4.64$, $p<0.001$). Significant disparities exist at the low ($t=3.47$, $p<0.001$), mid ($t=3.24$, $p<0.01$), and high levels ($t=2.42$, $p<0.05$). Comparing the group means indicates that the EG performs significantly better than the CG across all levels and in overall affective-domain learning objectives. As a result, using PETIS for timely learner assistance significantly improves affective-domain learning objectives, as opposed to traditional instructor-dependent judgment calls.

Table 6 Independent sample *t*-test for affective-domain learning objectives

	Group	*M*	SD	*t*
Low-level affective domain	EG	11.90	3.73	**3.47*** **
	CG	8.63	3.57	
Mid-level affective domain	EG	11.10	2.87	**3.24** **
	CG	8.77	2.79	
High-level affective domain	EG	11.70	4.04	**2.42***
	CG	9.36	3.30	
Affective domain	EG	34.70	7.15	**4.64*** **
	CG	26.80	6.04	

*$p<0.05$, **$p<0.01$, ***$p<0.001$

Bold values represent significant difference

Table 7 Pearson correlation coefficient analysis of affective domain and programming skill

	Low-level	Mid-level	High-level	Post-test programming skill
Low-level	1			
Mid-level	0.190	1		
High-level	0.248	0.305*	1	
Post-test programming skill	0.311*	0.366**	0.442***	1

*$p<0.05$, **$p<0.01$, ***$p<0.001$

The relationship between programming skill and affective-domain learning objectives

To deepen our understanding of the impact of each level of affective-domain learning objectives on programming skill within the computer programming course, we computed the Pearson correlation coefficient. As shown in Table 7, there exists a significantly positive correlation between the low-level affective domain and programming skills ($r=0.311$, $p<0.05$), the middle-level affective domain and programming skills ($r=0.366$, $p<0.01$), and the high-level affective domain and programming skills ($r=0.442$, $p<0.001$). Therefore, at all levels, affective-domain learning objectives significantly influence the programming skill of K-12 learners in computer programming courses, with the extent of this influence progressively increasing from the low to high level.

Is PETIS a potent and beneficial instrument for advancing K-12 programming education?

To address the question of usefulness in K-12 programming education, we first need to define a "useful" tool. For learners, it should augment learning outcomes (including programming skills and affective domain) in programming courses by providing instructors with the opportunity for timely intervention when learners are stuck. For instructors, it should offer an understanding of learner struggles via a quick and straightforward examination of the PETIS interface. We employed semi-structured interviews and thematic analysis in our study to assess PETIS's helpfulness for learners and instructors. Thematic analysis of the learner's interview data produced three themes: instructor intervention (providing timely intervention), cognitive issues (enhancement of programming skill) in programming, and affective issues (enhancement of affective domain) in programming.

Table 8 Thematic analysis of learner interviews

Theme/sub-theme	EG (*N*)	CG (*N*)
Instructor intervention		
• Instructors did not help me to resolve my difficulties	0	2
• Instructors helped me only after I raised my hand	2	3
• Instructors helped me immediately when I got stuck	4	1
Cognitive issues in programming		
• It was difficult to understand the programming concepts	1	3
• I was confused about the programming concepts	1	2
• I understood most of the programming concepts	4	1
Affective issues in programming		
• I was concerned about how difficult programming is	0	2
• I had a positive attitude towards programming	2	3
• I was confident about programming	4	1

The results of the thematic analysis, depicted in Table 8, highlight instructor intervention. Most EG learners felt they received immediate help from the instructors ($N=4$). One learner reported, "Sometimes when I stopped to think about the problem, the instructors would approach me and provide assistance so that I would not be interrupted by difficulties when learning." In contrast, most CG learners felt that the instructors only came to their aid after they had raised their hand ($N=3$), or sometimes not at all ($N=2$). One CG learner stated, "When I have a problem, I must raise my hand to ask for help, but sometimes the TAs are helping someone else, so I have to wait until the TAs are free to resolve my difficulties." Overall, EG learners perceived their problems in the course as being better and more immediately resolved than those of CG learners. Although most CG learners believed that instructors would not provide immediate assistance ($N=5$), most of them still felt they could receive help after raising their hands ($N=3$). This underscores the importance of a sufficient number of teaching assistants (TAs).

Regarding cognitive issues in programming, most EG learners believed they had mastered all the programming concepts taught in the course ($N=4$). One learner expressed, "Because I don't get interrupted when learning, I am very consistent in my learning and do not get interrupted by difficulties." Conversely, most CG learners felt confused ($N=2$) and found it challenging to understand the programming concepts taught in the activity ($N=3$). One learner confessed, "I felt like I spent a lot of time waiting for help from the TAs during the activity, so there was a lack of coherence in learning, and I had to continue learning the next concepts before I had really learned the previous ones." Overall, EG learners felt they had a better understanding of programming concepts during the activity than CG learners. The introduction of PETIS reduces waiting time for learners, allowing those in EG to absorb more knowledge than their counterparts in CG.

Finally, regarding affective issues in programming, most EG learners felt more confident about future programming learning ($N=4$) and exhibited a positive attitude towards learning programming ($N=2$). One learner shared, "After this activity, I would like to pursue a programming-related major in the future." In contrast, although most CG learners also had positive attitudes toward learning programming ($N=3$), some were concerned that the level of programming difficulty would pose a barrier to learning ($N=2$). One student admitted, "I think this activity taught me basic programming concepts, but I still don't have a clear understanding of logic and loops, which makes me apprehensive about having to program in the future." Overall, EG learners felt positive about future programming and more confident about mastering programming concepts than CG learners. The immediate resolution of difficulties in EG due to the intervention of PETIS resulted in no EG learners perceiving programming learning as difficult ($N=0$). In contrast, two learners in CG considered learning programming to be challenging ($N=2$). The faster learners' difficulties were resolved, the more positive attitudes and confidence were established in programming courses.

Interview data from the two instructors and the TAs showed that PETIS enhances teaching. One instructor praised, "The PETIS user interface is simple and easy to use, and you quickly notice when learners run into trouble. Additionally, the functions for past imagery and statistical data analysis make it easy to understand when learners are encountering difficulties, which I can use to decide how to adjust the difficulty of the teaching materials." Another instructor noted, "Some more introverted learners are afraid to raise their hands to ask questions, so they may not get anything out of the activity; this system helps such learners perform better in the activity." Overall, two TAs believed they could quickly and directly understand what learners were doing and whether they were stuck by examining the interface of PETIS.

We engaged two STEM experts to further explore the effectiveness of PETIS in identifying difficulties encountered by students during their learning process. They compared their observations with the output generated by PETIS, using a standardized set of coding guidelines. Five 20-min classroom videos were analyzed by both the experts and PETIS to determine instances where learners faced challenges. The coding guidelines and PETIS system output were uniformly set to generate reports every

5 min, indicating whether learners were experiencing problems. Cronbach's alpha score was employed to assess the consistency between the codings, with a consistent result suggesting that PETIS has the potential to replace expert coding. The Cronbach's alpha results revealed scores of 0.86 between Expert A and Expert B, 0.79 between Expert A and PETIS, and 0.73 between Expert B and PETIS. All values exceed the threshold of 0.7, confirming the accuracy of PETIS in identifying learning difficulties.

In summary, for both K-12 learners and instructors, PETIS significantly facilitates programming education. The interviews indicate that the EG significantly outperformed the CG in terms of cognitive and affective outcomes, as well as in the perception of intervention immediacy. This finding is in line with the quantitative data from the questionnaires. In addition, instructors also perceived PETIS as a useful tool for their teaching.

Discussion

We introduced PETIS to scrutinize its impact on programming skill and the affective domain in K-12 computer programming courses and to analyze the relationship between the cognitive and affective domains.

Impact on programming skill

Acknowledging the significance of programming, some nations have incorporated it as a mandatory K-12 course to instill twenty-first century skills (Tikva & Tambouris, 2021). Yet, the challenges faced by K-12 learners in programming are often disregarded (Perera et al., 2021; Raj et al., 2018). Specifically, the predominantly English grammatical and developmental contexts of programming languages pose barriers to knowledge construction for K-12 learners who are English as a Foreign Language (EFL) speakers (Perera et al., 2021). Moreover, the interconnected nature of programming concepts can obstruct subsequent learning if a single concept remains unclear (Kranch, 2012; Nikula et al., 2011). Low instructor–learner ratios also contribute to learners discontinuing their studies (Nikula et al., 2011; Ott et al., 2016). We proposed PETIS as a solution, aiding instructors in identifying learners' difficulties and providing support promptly. As displayed in Table 5, instructors who utilized PETIS significantly enhanced K-12 learners' programming skill mastery compared to those who relied on conventional methods. By leveraging deep learning and image processing technologies, we developed PETIS—a system that facilitates real-time assessment of learners' progress. This timely evaluation empowers educators to intervene when students struggle, reducing learning interruptions and fostering an engaging learning environment. The positive impact on learners' programming skills proficiency reaffirms the findings of Medeiros et al. (2018), stressing the importance of immediate intervention. It also aligns with previous studies endorsing the advantages of precision education on students' knowledge construction (Hu, 2022; Tsai et al., 2020).

Impact on affective-domain learning objectives

Over recent years, a growing emphasis has been placed on the affective domain in K-12 computer programming education (Yun & Cho, 2021). As a crucial component of STEM education, programming also highlights the importance of integrating interdisciplinary knowledge to cultivate learners' problem-solving abilities (Hsiao et al., 2022). Consequently, most contemporary research positions the enhancement of the affective domain as a key learning outcome within both programming and STEM education (Apedoe et al., 2008; Guzey et al., 2016; Hsiao et al., 2022). Mainstream methodologies often incorporate gaming elements or various pedagogical theories to promote affective-domain learning objectives. Yet, one of the most direct and intuitive approaches is frequently overlooked—providing learners with timely and relevant feedback and assistance when they encounter obstacles. Historically, technological limitations and disproportionate instructor–learner ratios made it challenging for educators to promptly discern when learners were struggling. This often led to student demoralization and subsequent disengagement from the learning process (Nikula et al., 2011; Ott et al., 2016). To address this issue, we have developed PETIS—a tool designed to assist instructors in rapidly identifying and rectifying learner difficulties. As shown in Table 6, instructors who employ PETIS to detect and address learning challenges significantly bolster the affective-domain learning outcomes of K-12 students compared to traditional, ad hoc methods. Experimental Group (EG) learners markedly outshine Control Group (CG) learners in achieving low-level affective-domain learning objectives, such as learning motivation—a finding that resonates with most studies on the implementation of precision education (Liu, 2022; Ross et al., 2018). The intervention of PETIS helps ensure that learners encounter fewer difficulties, thereby fostering an environment conducive to learning and enhancing their motivation (Lin & Chen, 2017). Likewise, EG learners significantly outperform CG learners in achieving mid-level objectives, such as learning attitudes—a result that aligns with the majority of research in precision education (Hu, 2022; Lin & Lai, 2021). Owing to the timely assistance provided by instructors using PETIS, learners are less apprehensive about confronting unfamiliar or cognitively demanding content, hence fostering positive sentiments and perspectives towards the learning material and ultimately deepening their learning attitudes (Tzafilkou et al., 2021). Finally, with respect to high-level objectives, such as self-efficacy, EG learners significantly surpass CG learners—a result consistent with prior studies on the integration of precision education (Hwang et al., 2020; Lin et al., 2021a, 2021b). The supportive intervention of PETIS enables learners to engage with course content without disruption, thereby allowing them to experience their progress, build confidence in their abilities, and fortify their self-efficacy (Bandura, 1977).

Relationship between programming skill and affective domain

As indicated in Table 7, a significant relationship is present between all levels of affective-domain learning objectives and programming skills. Additionally, Table 7 demonstrates that every level of affective-domain learning objectives is significantly and positively correlated with programming skills. Notably, the higher-level affective-domain learning objective, self-efficacy, shows a particularly strong correlation. That is, an increase in learner's self-efficacy corresponds to enhanced programming skills. This observation aligns with previous studies that have linked cognition with the affective domain (Lee et al., 2023; Liu, 2022). Yusuf (2011) proposed that learners possessing greater self-efficacy can confidently establish suitable self-regulation strategies based on their learning activities, which in turn results in improved outcomes. Furthermore, correlation coefficients displayed in Table 7 suggest the following hierarchy of influence on programming skills: high-level affective-domain objectives exert the greatest influence, followed by mid-level, and then low-level objectives. This finding resonates with Bloom's taxonomy of affective-domain learning objectives (Bloom, 1956; Krathwohl, 2002). Taken together, these results indirectly validate the practicality of assessing learners' affective-domain learning objectives—from low to high levels—encompassing aspects such as motivation, attitude, and self-efficacy in the context of programming education.

The enhancement recommendations of PETIS

The semi-structured interviews conducted with instructors have yielded insightful perspectives. One instructor suggested that the system's functionality could be expanded by incorporating a scoring mechanism predicated on the pattern of learner behaviors. This proposition aligns with another instructor's view who emphasized the potential benefits of identifying not just learners who are struggling, but also those exhibiting traits such as indolence or inattentiveness. Such capability, the instructor posits, would allow instructors to prioritize assistance based on urgency, thus enhancing the efficacy of interventions. These views collectively underscore the instructors' prioritization of recognizing learner engagement, and further emphasize the pivotal role of engagement in programming education (Hosseini et al., 2020; Yildiz Durak, 2020).

Conclusion

In light of the critical necessity for immediate intervention and the absence of precision education applications in the real-world (Gao et al., 2020), we have developed the Precision Education based Timely Intervention System (PETIS) using deep learning and image processing. This innovative system aids instructors in promptly identifying when learners encounter difficulties, providing swift and appropriate assistance. We utilize a quasi-experimental design to examine the enhancement in programming abilities and affective-domain learning objectives subsequent to PETIS implementation in a K-12 computer programming curriculum. Our results show that PETIS introduction substantially augments learners' programming skill and affective-domain learning outcomes. Furthermore, qualitative data gathered through interviews reveal a consensus among instructors and learners alike that this tool positively impacts K-12 computer programming education.

Programming, a vital facet of STEM education, promotes an integrated teaching and learning approach where discipline-specific content is not segregated but is treated as a dynamic and fluid area of study (Hsiao et al., 2022). Compared to traditional courses, the immediate identification and detection of learning obstacles in the programming process are notably challenging (Gao et al., 2020; Lee et al., 2023). Hence, under the premise that all learning behaviors occur through interactions between learners and their learning materials, we develop PETIS. This system contributes significant insights toward the development of real-world precision education tools for STEM education. By integrating the ICAP framework with learning behavior, PETIS's capability to identify learning engagement also holds implications for the automated measurement of engagement in STEM education.

Despite our findings, this study has its limitations. Firstly, the relatively small number of participants in the experiment ($N=60$) could potentially affect the validity of our statistical analysis. Additionally, the experiment's brief duration necessitated learners to assimilate a substantial amount of programming knowledge within a limited timeframe, possibly inducing bias due to cognitive overload. Since PETIS relies on image processing technology, the system's effectiveness may be compromised by issues like camera angle and occlusion, leading to misrecognition. Consequently, future research should not only expand the sample size and extend the experimental period to mitigate bias and statistical error but also incorporate Multimodal Learning Analysis (MMLA) technology. MMLA's capacity to amalgamate log files, images, and discussion data could provide a more comprehensive understanding of learning behaviors and processes, thereby further enhancing the system's accuracy.

Acknowledgements
We are extremely grateful to the research assistants and students who participated in this study.

Author contributions
H-YL is the leader of this research, he is in charge of the research design, conducting teaching and learning experiment, data analysis, and writing the manuscript. C-JL is responsible for assisting in the conduct of experiments and surveying related literature and proofreading the manuscript. W-SW is responsible for assisting in the conduct of experiments and surveying related literature. W-CC is responsible for assisting in the conduct of experiments. Y-MH is responsible for designing research experiments, providing fundamental education theories and comments to this research, and he is also responsible for revising the manuscript. All authors spent more than 2 months to discuss and analyze the data. The author(s) read and approved the final manuscript.

Funding
This project was funded by the Ministry of Science and Technology (MOST), of the Republic of China, under Contract numbers MOST 109-2511-H-006-011-MY3, MOST 110-2511-H-006-012-MY3, MOST 110-2511-H-006-008-MY3.

Availability of data and materials
Not applicable.

Declarations

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no conflict of interest.

Received: 18 December 2022 Accepted: 24 July 2023
Published online: 08 August 2023

References
Apedoe, X. S., Reynolds, B., Ellefson, M. R., & Schunn, C. D. (2008). Bringing engineering design into high school science classrooms: The heating/cooling unit. *Journal of Science Education and Technology, 17*(5), 454–465. https://doi.org/10.1007/s10956-008-9114-6
Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*, 191–215. https://doi.org/10.1037/0033-295X.84.2.191
Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*.
Cetin, I., & Ozden, M. Y. (2015). Development of computer programming attitude scale for university students. *Computer Applications in Engineering Education, 23*(5), 667–672. https://doi.org/10.1002/cae.21639
Chen, G., Gully, S. M., & Eden, D. (2001). Validation of a new general self-efficacy scale. *Organizational Research Methods, 4*(1), 62–83. https://doi.org/10.1177/109442810141004
Cheng, B., Fan, C., Fu, H., Huang, J., Chen, H., & Luo, X. (2022). Measuring and computing cognitive statuses of construction workers based on electroencephalogram: A critical review. *IEEE Transactions on Computational Social Systems, 9*(6), 1644–1659. https://doi.org/10.1109/TCSS.2022.3158585
Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist, 49*(4), 219–243. https://doi.org/10.1080/00461520.2014.965823
Cohen, L., Manion, L., & Morrison, K. (2000). Research methods in education. *Routledge*. https://doi.org/10.4324/9780203224342
Collins, F. S., & Varmus, H. (2015). A new initiative on precision medicine. *New England Journal of Medicine, 372*(9), 793–795. https://doi.org/10.1056/NEJMp1500523
Cook, C. R., Kilgus, S. P., & Burns, M. K. (2018). Advancing the science and practice of precision education to enhance student outcomes. *Journal of School Psychology, 66*, 4–10. https://doi.org/10.1016/j.jsp.2017.11.004
De Loof, H., Boeve-de Pauw, J., & Van Petegem, P. (2021). Engaging students with integrated STEM education: A happy marriage or a failed engagement? *International Journal of Science and Mathematics Education*. https://doi.org/10.1007/s10763-021-10159-0
DemİRkiran, M. C., & Tansu Hocanin, F. (2021). An investigation on primary school students' dispositions towards programming with game-based learning. *Education and Information Technologies, 26*(4), 3871–3892. https://doi.org/10.1007/s10639-021-10430-5
Gao, X., Li, P., Shen, J., & Sun, H. (2020). Reviewing assessment of student learning in interdisciplinary STEM education. *International Journal of STEM Education, 7*(1), 1–14. https://doi.org/10.1186/s40594-020-00225-4
Guay, F., Vallerand, R. J., & Blanchard, C. (2000). On the assessment of situational intrinsic and extrinsic motivation: The Situational Motivation Scale (SIMS). *Motivation and Emotion, 24*(3), 175–213. https://doi.org/10.1023/A:1005614228250
Guzey, S. S., Moore, T. J., Harwell, M., & Moreno, M. (2016). STEM integration in middle school life science: Student learning and attitudes. *Journal of Science Education and Technology, 25*(4), 550–560. https://doi.org/10.1007/s10956-016-9612-x
Hao, X., Xu, Z., Guo, M., Hu, Y., & Geng, F. (2023). The effect of embedded structures on cognitive load for novice learners during block-based code comprehension. *International Journal of STEM Education, 10*(1), 42. https://doi.org/10.1186/s40594-023-00432-9
Hosseini, R., Akhuseyinoglu, K., Brusilovsky, P., Malmi, L., Pollari-Malmi, K., Schunn, C., & Sirkiä, T. (2020). Improving engagement in program construction examples for learning python programming. *International Journal of Artificial Intelligence in Education, 30*(2), 299–336. https://doi.org/10.1007/s40593-020-00197-0
Hsiao, J.-C., Chen, S.-K., Chen, W., & Lin, S. S. J. (2022). Developing a plugged-in class observation protocol in high-school blended STEM classes: Student engagement, teacher behaviors and student–teacher interaction patterns. *Computers & Education, 178*, 104403. https://doi.org/10.1016/j.compedu.2021.104403
Hu, Y.-H. (2022). Effects and acceptance of precision education in an AI-supported smart learning environment. *Education and Information Technologies, 27*(2), 2013–2037. https://doi.org/10.1007/s10639-021-10664-3
Hwang, G.-J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles and research issues of Artificial Intelligence in Education. *Computers and Education: Artificial Intelligence, 1*, 100001. https://doi.org/10.1016/j.caeai.2020.100001
Jeong, J. S., González-Gómez, D., & Cañada-Cañada, F. (2021). How does a flipped classroom course affect the affective domain toward science course? *Interactive Learning Environments, 29*(5), 707–719. https://doi.org/10.1080/10494820.2019.1636079
Kranch, D. A. (2012). Teaching the novice programmer: A study of instructional sequences and perception. *Education and Information Technologies, 17*(3), 291–313. https://doi.org/10.1007/s10639-011-9158-8
Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory into Practice, 41*(4), 212–218. https://doi.org/10.1207/s15430421tip4104_2
Lee, H.-Y., Cheng, Y.-P., Wang, W.-S., Lin, C.-J., & Huang, Y.-M. (2023). Exploring the learning process and effectiveness of STEM education via learning behavior analysis and the interactive-constructive- active-passive framework. *Journal of Educational Computing Research*. https://doi.org/10.1177/07356331221136888
Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

Lin, H.-C., Tu, Y.-F., Hwang, G.-J., & Huang, H. (2021a). From precision education to precision medicine: Factors affecting medical staffs intention to learn to use AI applications in hospitals. *Educational Technology & Society, 24*(1), 123–137.
Lin, K.-C., Ko, C.-W., Hung, H.-C., & Chen, N.-S. (2021b). The effect of real-time pose recognition on badminton learning performance. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2021.1981396
Lin, M.-H., & Chen, H.-G. (2017). A study of the effects of digital learning on learning motivation and learning outcome. *Eurasia Journal of Mathematics, Science and Technology Education, 13*(7), 3553–3564. https://doi.org/10.12973/eurasia.2017.00744a
Lin, Y.-S., & Lai, Y.-H. (2021). Analysis of AI Precision education strategy for small private online courses. *Frontiers in Psychology, 1*, 2. https://doi.org/10.3389/fpsyg.2021.749629
Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology, 50*(4), 1979–1995. https://doi.org/10.1111/bjet.12685
Liu, T. C. (2022). A case study of the adaptive learning platform in a Taiwanese Elementary School: Precision education from teachers' perspectives. *Education and Information Technologies, 27*(5), 6295–6316. https://doi.org/10.1007/s10639-021-10851-2
Luan, H., & Tsai, C.-C. (2021). A review of using machine learning approaches for precision education. *Educational Technology & Society, 24*(1), 250–266.
Makransky, G., & Petersen, G. B. (2021). The cognitive affective model of immersive learning (CAMIL): A theoretical research-based model of learning in immersive virtual reality. *Educational Psychology Review, 33*(3), 937–958. https://doi.org/10.1007/s10648-020-09586-2
Marshall, P. (2007). *Do tangible interfaces enhance learning?* Proceedings of the 1st international conference on Tangible and embedded interaction, Baton Rouge, Louisiana. https://doi.org/10.1145/1226969.1227004
Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133
Nikula, U., Gotel, O., & Kasurinen, J. (2011). A motivation guided holistic rehabilitation of the first programming course. *ACM Transactions on Computing Education (TOCE), 11*(4), 1–38. https://doi.org/10.1145/2048931.2048935
Noroozi, O., Pijeira-Díaz, H. J., Sobocinski, M., Dindar, M., Järvelä, S., & Kirschner, P. A. (2020). Multimodal data indicators for capturing cognitive, motivational, and emotional learning processes: A systematic literature review. *Education and Information Technologies, 25*(6), 5499–5547. https://doi.org/10.1007/s10639-020-10229-w
Nunnally, J. C. (1978). An overview of psychological measurement. *Clinical diagnosis of mental disorders*, 97–146.
Ott, C., Robins, A., & Shephard, K. (2016). Translating principles of effective feedback for students into the CS1 context. *ACM Transactions on Computing Education (TOCE), 16*(1), 1–27. https://doi.org/10.1145/2737596
Ouyang, F., Dai, X., & Chen, S. (2022). Applying multimodal learning analytics to examine the immediate and delayed effects of instructor scaffoldings on small groups' collaborative programming. *International Journal of STEM Education, 9*(1), 45. https://doi.org/10.1186/s40594-022-00361-z
Panskyi, T., Rowinska, Z., & Biedron, S. (2019). Out-of-school assistance in the teaching of visual creative programming in the game-based environment–Case study: Poland. *Thinking Skills and Creativity, 34*, 100593. https://doi.org/10.1016/j.tsc.2019.100593
Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A systematic review of introductory programming languages for novice learners. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3089560
Pordelan, N., Sadeghi, A., Abedi, M. R., & Kaedi, M. (2020). Promoting student career decision-making self-efficacy: An online intervention. *Education and Information Technologies, 25*(2), 985–996. https://doi.org/10.1007/s10639-019-10003-7
Qushem, U. B., Christopoulos, A., Oyelere, S. S., Ogata, H., & Laakso, M.-J. (2021). Multimodal technologies in precision education: Providing new opportunities or adding more challenges? *Education Sciences*. https://doi.org/10.3390/educsci11070338
Raj, A. G. S., Ketsuriyonk, K., Patel, J. M., & Halverson, R. (2018). *Does native language play a role in learning a programming language?* Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA. https://doi.org/10.1145/3159450.3159531
Rapee, R. M., Kim, J., Wang, J., Liu, X., Hofmann, S. G., Chen, J., Oh, K. Y., Bögels, S. M., Arman, S., Heinrichs, N., & Alden, L. E. (2011). Perceived impact of socially anxious behaviors on individuals' lives in Western and East Asian Countries. *Behavior Therapy, 42*(3), 485–492. https://doi.org/10.1016/j.beth.2010.11.004
Ross, B., Chase, A.-M., Robbie, D., Oates, G., & Absalom, Y. (2018). Adaptive quizzes to increase motivation, engagement and learning outcomes in a first year accounting unit. *International Journal of Educational Technology in Higher Education, 15*(1), 30. https://doi.org/10.1186/s41239-018-0113-2
Scott, M. J., & Ghinea, G. (2014). On the domain-specificity of mindsets: The relationship between aptitude beliefs and programming practice. *IEEE Transactions on Education, 57*(3), 169–174. https://doi.org/10.1109/TE.2013.2288700
Sharunova, A., Wang, Y., Kowalski, M., & Qureshi, A. J. (2022). Applying Bloom's taxonomy in transdisciplinary engineering design education. *International Journal of Technology and Design Education, 32*(2), 987–999. https://doi.org/10.1007/s10798-020-09621-x
Snow, R. E. (1986). Individual differences and the design of educational programs. *American Psychologist, 41*, 1029–1039. https://doi.org/10.1037/0003-066X.41.10.1029
Sun, D., Ouyang, F., Li, Y., & Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education, 8*(1), 1–15. https://doi.org/10.1186/s40594-021-00311-1
Sung, J., Lee, J. Y., & Chun, H. Y. (2023). Short-term effects of a classroom-based STEAM program using robotic kits on children in South Korea. *International Journal of STEM Education, 10*(1), 26. https://doi.org/10.1186/s40594-023-00417-8
Tempelaar, D., Rienties, B., & Nguyen, Q. (2021). The contribution of dispositional learning analytics to precision education. *Educational Technology & Society, 24*(1), 109–122.
Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education, 162*, 104083. https://doi.org/10.1016/j.compedu.2020.104083
Tsai, S.-C., Chen, C.-H., Shiao, Y.-T., Ciou, J.-S., & Wu, T.-N. (2020). Precision education with statistical learning and deep learning: A case study in Taiwan. *International Journal of Educational Technology in Higher Education, 17*(1), 1–13. https://doi.org/10.1186/s41239-020-00186-2
Tzafilkou, K., Perifanou, M., & Economides, A. A. (2021). Development and validation of a students' remote learning attitude scale (RLAS) in higher education. *Education and Information Technologies, 26*(6), 7279–7305. https://doi.org/10.1007/s10639-021-10586-0
Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You only learn one representation: Unified network for multiple tasks. https://doi.org/10.48550/arXiv.2105.04206
Wong, K. (2022, May 5). *YOLOR*. Retrieved 26 December, from https://github.com/WongKinYiu/yolor
Wu, C.-H., Liu, C.-H., & Huang, Y.-M. (2022). The exploration of continuous learning intention in STEAM education through attitude, motivation, and cognitive load. *International Journal of STEM Education, 9*(1), 35. https://doi.org/10.1186/s40594-022-00346-y
Wu, W.-H., Kao, H.-Y., Wu, S.-H., & Wei, C.-W. (2019). Development and evaluation of affective domain using student's feedback in entrepreneurial Massive Open Online Courses. *Frontiers in Psychology, 10*, 1109. https://doi.org/10.3389/fpsyg.2019.01109
Xu, Z., Ritzhaupt, A. D., Tian, F., & Umapathy, K. (2019). Block-based versus text-based programming environments on novice student learning outcomes: A meta-analysis study. *Computer Science Education, 29*(2–3), 177–204. https://doi.org/10.1080/08993408.2019.1565233
Yadegaridehkordi, E., Noor, N. F. B. M., Ayub, M. N. B., Affal, H. B., & Hussin, N. B. (2019). Affective computing in education: A systematic review and future research. *Computers & Education, 142*, 103649. https://doi.org/10.1016/j.compedu.2019.103649
Yang, S. J. (2021). Guest Editorial: Precision education-A new challenge for AI in education. *Journal of Educational Technology & Society, 24*(1), 105–108.
Yildiz Durak, H. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational

thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning, 25*(1), 179–195. https://doi.org/10.1007/s10758-018-9391-y

Yiu, H. L. (2013). The influence of student–teacher racial match on student–teacher closeness: A focus on Asian and Asian American students. *Asian American Journal of Psychology, 4*, 126–135. https://doi.org/10.1037/a0027785

Yun, H. J., & Cho, J. (2021). Affective domain studies of K-12 computing education: A systematic review from a perspective on affective objectives. *Journal of Computers in Education*. https://doi.org/10.1007/s40692-021-00211-x

Yusuf, M. (2011). The impact of self-efficacy, achievement motivation, and self-regulated learning strategies on students' academic achievement. *Procedia: Social and Behavioral Sciences, 15*, 2623–2626. https://doi.org/10.1016/j.sbspro.2011.04.158

Zhao, D., Muntean, C. H., Chis, A. E., Rozinaj, G., & Muntean, G. M. (2022). Game-based learning: Enhancing student experience, knowledge gain, and usability in higher education programming courses. *IEEE Transactions on Education, 65*(4), 502–513. https://doi.org/10.1109/TE.2021.3136914

Publisher's Note

Hsin-Yu Lee received his B.S. degree in the Department of Technology Application And Human Resource Development, National Taiwan Normal University, Taiwan, R.O.C., in 2019, and received the M.S. degree in the Department of Engineering Science, National Cheng-Kung University, Taiwan, R.O.C., in 2021. He is currently pursuing his Ph.D. degree in engineering science at National Cheng Kung University, Taiwan, R.O.C. His research interests include educational technology, learning analytics, computer vision and artificial intelligence.

Chia-Ju Lin received her B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan, R.O.C., in 2019 and 2021, respectively. She is currently pursuing her Ph.D. degree in engineering science at National Cheng Kung University, Taiwan. Her research interests include educational technology, learning analytics, computer networks and artificial intelligence.

Wei-Sheng Wang received her B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan, R.O.C., in 2018 and 2020, respectively. He is currently pursuing his Ph.D. degree in engineering science at National Cheng Kung University, Taiwan. His research interests include virtual reality, educational technology, and optimization algorithm.

Wei-Cyun Chang received his B.S. degree in the Department of Electrical Engineering, National Taiwan Ocean University, Taiwan, R.O.C., in 2020, and received the M.S. degree in the Department of Engineering Science, National Cheng-Kung University, Taiwan, R.O.C., in 2022.

Yueh-Min Huang received the M.S. and Ph.D. degrees in Electrical Engineering from the University of Arizona in 1988 and 1991, respectively. He is a Chair Professor in Department of Engineering Science, National Cheng-Kung University, Taiwan. He has co-authored 3 books and has published more than 250 refereed journal research papers. His research interests include e-Learning, multimedia communications, wireless networks, and artificial intelligence. He is in the editorial board of several international journals in the area of educational technology, computer communications, and web intelligence, including 3 SSCI-indexed e-Learning ones. In e-Learning area, except of publishing more than 100 SSCI-indexed journal papers and editing 3 special issues in SSCI-indexed journals, he was also serving as the directors of Disciplines of Applied Science Education and Innovative Engineering Education in Taiwan's Ministry of Science and Technology. Dr. Huang is a senior member of the IEEE and became Fellow of British Computer Society in 2011. Dr. Huang has received many research awards, such as Taiwan's National Outstanding Research Award in 2011 and 2014, given to Taiwan's top 100 scholars. According to a paper published in BJET, he is ranked no. 3 in the world on terms of the number of educational technology papers published in the period 2012 to 2017.