Enhancing Teachers' Programming and Artificial Intelligence Competencies

through a Symbol-Based Robot Car Control Workshop

***Abstract*—This research aimed to develop a workshop on programming and artificial intelligence for controlling robotic cars using pictorial symbols, comparing pre- and post-training knowledge, and study participant satisfaction. The target group consisted of 19 teachers and educational personnel who participated in a 6-hour training session integrating microcontrollers, IoT systems, programming, motor control, and image classification. Participants assembled robotic cars using ESP8266 and L298N, practiced classifying six symbol classes using Teachable Machine, and developed a Python program to send HTTP commands via Wi-Fi. Research instruments included a 29-item knowledge test and a 5-level Likert scale satisfaction questionnaire. The results indicated that the average post-training score was 18.11, significantly higher than the pre-training average of 13.21 ($p < .05$), indicating a large effect size. The number of participants achieving at least 60% increased from 3 to 10. The area showing the greatest improvement was programming and the use of the Arduino IDE. Overall participant satisfaction was very high, and participants perceived an increase in post-training knowledge. The study concludes that teachers can use the developed activities as a guideline to enhance their competencies in artificial intelligence, robotics, automation, and computational thinking. However, the duration of the training should be increased, training manuals should be prepared, and the application of the acquired knowledge in the classroom should be monitored in subsequent studies.**

Keywords—Artificial Intelligence, Robot Car, Microcontroller, Computational Thinking, Teacher Professional Development

I. INTRODUCTION

The advances in artificial intelligence (AI) and automation have had a significant impact on teaching and the role of the teacher. In this age of digital transformation, teachers need to be more than users of technology. They need to have a thorough understanding of AI concepts, be able to craft relevant and contextualized learning experiences, and use technology ethically, safely, and responsibly, with an awareness of their potential impacts. The UNESCO AI Competency Framework for Educators identifies 15 competencies across five dimensions: human-centric reasoning, ethical considerations of AI, foundational knowledge and applications of AI, AI-enhanced learning, and AI for professional growth. Professional development for educators should, in consequence, foster an understanding of how tools and systems are applied, the processes involved, and appropriate oversight in the use of AI [1].

Computational thinking is a prerequisite for learning programming, artificial intelligence, and automation. It covers processes like problem decomposition, abstraction and modeling, algorithm design, testing, and debugging. Therefore, computational thinking is not only restricted to programming skills but also a systematic reasoning process to formulate solutions that can be realized by computers or digital devices. The use of educational robots as a learning tool makes this process visible in a tangible way because learners can directly analyze the relationship between commands, programs, electronic circuits, and the behavior of real devices. If the program or device connections are incorrect, the robot may move in the wrong direction, fail to respond to commands, or continue moving under specified conditions. The immediate feedback allows learners to analyze causes, test hypotheses, debug the program, and cyclically improve the system.

Previous studies indicated that professional development of computational thinking can improve teachers' knowledge, attitudes, beliefs, and confidence to promote learning, especially activities that enable teachers to actively engage, create lesson plans, share ideas with peers, and reflect on the learning outcomes. Professional development that connects computational thinking with the curriculum can enhance teachers' knowledge and confidence in teaching [2]. Meanwhile, Kılıç and Çakıroğlu [3] concluded that professional development through the use of robots with real-world problems, peer assessment, and experimental teaching activities can improve both content knowledge and the knowledge of managing computational thinking.

However, literature reviews have identified that most programs of teacher professional development are constrained by relatively short training durations, self-assessment of perception, and limited follow-up of knowledge application in the classroom. Furthermore, many programs have been designed to evaluate the knowledge, understanding, attitudes, or confidence of teachers in the short term, with little evidence about their ability to design activities, apply them at school, or their impact on students [4], [5]. So, the teacher's professional development activities should consist of a combination of conceptual knowledge and practical exercises and discussions on the application in the classroom, as well as the evaluation of the acquired knowledge and the satisfaction of the participants.

The stages of integrating artificial intelligence into embedded systems and robotics include several learning modules, starting from problem definition, collection and preparation of image data, class definition, model training and testing, programming and processing of the results, network command execution, and motor driver control. This level of complexity can put a learning burden on beginners, especially those with different backgrounds in programming and electronics. Therefore, it is important to choose tools that make the technical complexities simpler for short-term training.

Teachable Machine is a web-based tool that makes creating models to classify images, sounds, or gestures simple, even if you don't know anything about complex machine learning. Users can gather data, train models, test classification on new data, and simultaneously observe the confidence level of the results. Research by Kurz et al. [6] has shown that Teachable Machine can support teacher professional development and make concepts of machine learning more accessible to teachers. Thus, the tool is appropriate as a basis for activities where trainees would have to see the AI workflow from data preparation, model training, and testing to applying classification results to control real devices.

Considering the importance and limitations mentioned above, the present study aimed to develop a workshop for teachers at the basic education and vocational education levels. Participants learned to develop a robotic car that received commands from six classes of visual symbols: forward, backward, left turn, right turn, stop, and no symbol found. The system consists of a camera to capture images, an image classification model trained with Teachable Machine, a Python program to process the classification results and convert them to HTTP commands, an ESP8266 board, and an L298N motor driver module to control the direction and speed of the motor. The system is also designed to stop the car when it reaches a stop sign or when no symbol is detected, thereby improving safety during use.

This activity is designed to help teachers grasp how different parts of a complete system work together, including data, artificial intelligence models, processing programs, network communication, and control of physical devices. It also allows them to apply these concepts in learning activities in science, technology, engineering, and mathematics (STEM), programming, robotics, and artificial intelligence in schools. The present study aims to assess participants' knowledge before and after training, as well as their satisfaction, perceived skills, and feedback on the activity format. The results will inform the development of more effective teacher training on artificial intelligence and automation.

II. Background and Related Works

A. *Professional Development for Teachers in AI and Computational Thinking*

Teacher development in AI must include grounding in the fundamentals, assessment of the limitations of systems, responsible use of data, and learning design that engages the learner in asking questions, experimenting, and monitoring the results. It should not be confined to demonstrations of ready-made tools. According to Espinal et al. [4] , the training of computational thinking should include assessment of projects, design of activity, and observation of practical application. UNESCO proposes a developmental trajectory from awareness and use of tools to the application and development of new approaches [1].

B. *Robots for Education and Hands-on Learning*

Robotics is a way to link abstract concepts to physical outcomes such as command sequences, conditions, variables, sensor readings, and motor control. In this way, the use of robotics activities allows for the facilitation of problem-solving and reflective learning. Kılıç and Çakıroğlu [3] found that activities based on real-life problems, peer assessment, lesson plans, and short lessons contributed to the development of teachers' knowledge on robotics and computational thinking. Recent work in the context of robotics instruction integrating AI has also shown positive trends in teacher-learner computational thinking skills and acceptance of robots for educational purposes [7].

C. *Accessible Modeling Tools*

Modeling is made accessible through graphical interfaces with Teachable Machine. Users can easily save images, share classes, train, and export models [8]. Such features are ideal for creating an initial experience. But teachers still have to explain the balance of the data. Training and test data division, model confidence, errors from lighting and camera angles, and the design of safety conditions when AI controls real devices.

III. Objective

The objectives of this study are as follows:

- Develop a seminar on coding and artificial intelligence for robotic vehicle control using visual cues.
- To assess the knowledge of the participants before and after their participation in the workshop.
- To assess the satisfaction and feedback of the participants regarding the above-mentioned workshop.

IV. RESEARCH METHODOLOGY

A. Research Design

The present study was conducted as a quasi-experimental study with a single group pre- and post-training to investigate the effect of a workshop on programming and artificial intelligence to control a robotic vehicle using visual symbols. The research framework included four components: input, process, output, and outcome. The independent variable was the 6-hour workshop. The dependent variables were knowledge about microcontrollers, automation, programming, control of motors and artificial intelligence, and satisfaction with and perceived ability to apply this knowledge.

The research process began with the pre-training knowledge assessment. Then the participants learned the basic concepts and developed the robotic vehicles as prescribed. At the end of the activity, a post-training knowledge assessment and satisfaction with the activity were conducted. The collected data were analyzed to see the changes in knowledge, the initial effectiveness of the activity, and areas for improvement for future training. The research process is presented in Fig. 1.

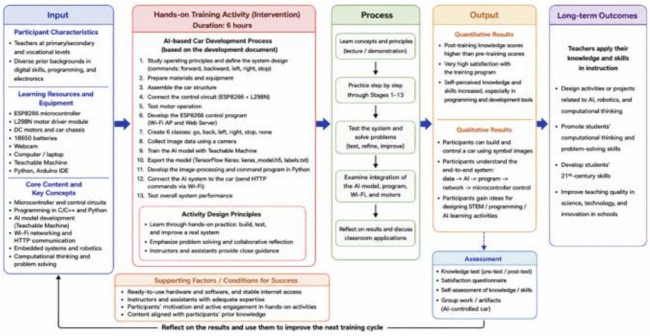

Fig. 1. Conception framework

B. Designing Training Activities

The training activities were designed according to the concept of hands-on learning by combining the knowledge of microcontrollers, programming, electronic circuits, network communication, artificial intelligence image classification, and robot control into one activity. Participants need to make a robotic car that can take commands from image symbols like forward, backward, left turn, right turn, and stop. The prototype system uses an ESP8266 board coupled with an L298N motor driver module, a DC motor, a camera, and an image classification model developed with Teachable Machine.

The learning process involves the study of principles, demonstration, step-by-step practice, system tests, error analysis, and project improvement. Participants are challenged to explore the correlation between image data, classification results, control commands, network communication, and the movement behavior of the car to fully understand the functioning of artificial intelligence systems and physical devices.

C. Development and Integration of AI-controlled Robotic Systems

The robotic system was developed by connecting an ESP8266 board with an L298N motor driver module for the forward, backward, left turn, right turn, and stop movements through a Wi-Fi network. Then the model was trained with Teachable Machine. The dataset consisted of six symbol classes: go, back, left, right, stop, and none. The model was output in the TensorFlow Keras format.

The Python program takes images from the camera, classifies the symbols, checks the confidence level, and converts the results to HTTP commands, which are sent to the ESP8266 for controlling the motors. The stop and no-class designs are designed to stop the vehicle for safety. The process includes image classification, programming, network communication, and controlling a physical device in one system.

Fig. 2. Activities for the workshop on developing artificial intelligence-the controlled robotic vehicles.

D. Conducting Workshops and Collecting Data

The workshop was 6 hours and started with an explanation of the objectives and a pre-workshop test. Then, the participants underwent a series of activities such as learning basic principles of microcontrollers, motor control, programming, and image classification with artificial intelligence, assembling a robotic vehicle and connecting the ESP8266 to the L298N module, testing motors, creating image datasets and training with Teachable Machine, and making a Python program to control the vehicle via Wi-Fi.

The instructors led the participants during the activities, and they collaborated to analyze the mistakes and improve the system. At the end of the workshop, participants took a post-workshop test and filled out a satisfaction survey, providing feedback on the content, duration, equipment, activity format, and application of the knowledge. The data collected were examined for completeness before statistical analysis. Fig. 2 shows an example of the workshop activities.

E. Data Analysis

The training began with a clarification of the objectives and a pre-training test. Afterwards, a series of activities were performed, such as basic concepts, circuit assembly and connection, testing the motor, control programming for ESP8266, creating image datasets, training the model, developing Python programming, and testing the whole system. During the activities, the instructors guided the participants, and the participants exchanged problem-solving approaches within the groups. At the end of training, participants completed a post-training test and satisfaction survey (content, duration, equipment, activity format, application of knowledge). Completeness of data was checked before statistical analysis was performed.

V. RESEARCH RESULTS

A. Participant Demographics Overview

The training involved a total of nineteen participants: ten men and nine women. Most of them were aged 30 and above. Thirteen of the participants were in basic education, three in vocational education, two in the private sector, and one was a staff member in a business. Participants came from diverse backgrounds in programming, electronics, automation, and the IoT. Selection was voluntary, as the activity was designed for professional development for teachers and those interested in applying technology to learning.

Before the activity started, participants were given explanations of the objectives, training procedures, equipment use, and safety guidelines. Participants were organized into collaborative learning activities to facilitate knowledge sharing, circuit testing, and problem-solving during system development.

B. Results of Pre- and Post-Training Knowledge Tests

A pre- and post-training test was conducted among 19 trainees using a 29-item test to assess the change in their knowledge. The data were analyzed by means, standard deviation, percentage of mean scores, median score range, and number of trainees who scored at least 60%. The analysis results are shown in Table 1.

TABLE I. COMPARISON OF PRE-TRAINING AND POST-TRAINING KNOWLEDGE TEST SCORES OF PARTICIPANTS

n=19	**The average Score**	**Standard Deviation**	**Percentage The average Score**	**Median**	**Minimum/Maximum Score**	**Participants with at least 60% score**
Before Training	13.21	4.45	45.55	14.00	1–21	3 (15.79%)
After Training	18.11	4.58	62.43	18.00	9–27	10 (52.63%)

As shown in Table 1, the analysis of knowledge scores indicated that there were 19 participants with a mean score of 13.21 (SD = 4.45) out of a possible 29 points, or 45.55% in the pre-test. Also, 19 participants scored an average of 18.11 points (SD = 4.58) in the post-training test, which corresponds to 62.43%. The average test score after training was up by 4.89 points. Also, the number of participants who scored 60% or more increased from 3 (15.79%) to 10 (52.63%). The results indicate a significant increase in the participants' knowledge of programming, microcontrollers, motor control devices, and artificial intelligence after the training. This demonstrates that a portion of the participants were able to develop their knowledge to achieve the required score level after completing the training.

TABLE II. COMPARISON OF PRE-TRAINING AND POST-TRAINING CORRECT RESPONSE RATES BY KNOWLEDGE DOMAIN

Knowledge Domain	**Knowledge Domain**	**Knowledge Domain**	**Knowledge Domain**
Automation and AI	56.14	71.93	15.79
Microcontrollers and the IoT	41.23	55.26	14.04
Devices, Sensors, and Motor Control	33.97	50.72	16.75
Programming and Arduino IDE	59.06	78.36	19.30

TABLE III. COMPARISON OF PRE-TRAINING AND POST-TRAINING CORRECT RESPONSE RATES BY KNOWLEDGE DOMAIN

Item	**Mean**	**SD**	**Level**
Clarity of the project content and objectives	4.53,,	0.51	Highest
Appropriateness of the training duration	4.37	0.90	High
Benefits gained from the training	4.63	0.50	Highest
Ability to apply the acquired knowledge in practice	4.53	0.51	Highest
Quality of the instructor and instruction	4.63	0.50	Highest
Appropriateness of the equipment and instructional media	4.68	0.48	Highest
Benefits gained from the workshop	4.68	0.48	Highest
Understanding of automation and IoT systems	4.16	0.76	High
Electronic circuit connection skills	4.16	0.83	High
Microcontroller programming skills	4.00	0.94	High
Ability to apply logical reasoning in problem-solving	4.00	1.00	High
Ability to design automation systems appropriate for practical applications	3.95	0.78	High
Knowledge and understanding before the training	3.32	1.29	Moderate
Knowledge and understanding after the training	4.32	0.58	High
Ability to organize and transform ideas into systematic work development	4.21	0.63	High
Ability to organize and transform ideas into systematic development of AI-related work	4.26	0.65	High

Table 2 shows post-training scores increased across all areas. The area that improved the most was programming and using the Arduino IDE, with a 19.30% increase. This was followed by hardware, sensors, and motor control, up 16.75%, and automation and AI, up 15.79%. However, the post-training score on hardware, sensors, and motors is still the lowest at 50.72%. This means that the trainees still need practical training and more time to troubleshoot hardware issues such as circuit connection, signal pin selection, power supply, motor control, and device error checking.

C. *Participant Satisfaction and Perceived Learning Outcomes*

To evaluate the trainees' opinion about the organization of the activities. The evaluation included satisfaction and perception of learning outcomes, using a 5-level rating scale that assessed the clarity of content and objectives. The time is appropriate. Benefits and the ability to use knowledge. Average analysis for assessment level and standard deviation, as shown in Table 3

Table 3 revealed the overall assessment of the training and satisfaction with a mean score of 4.58 (standard deviation of 0.42), which indicated the highest level of satisfaction. 97.74% of the responses in this area were rated high to very high. The highest-scored items were the appropriateness of equipment and teaching materials and the benefits received from the workshop activities, with a mean score of 4.68. The next highest mean score of 4.63 was for the benefits of the training and the quality of the trainers. This reflects the fact that workshop-style activities in which participants can build circuits, write programs, and test out robot control using symbols are effective in promoting engagement and in enabling participants to see the benefits of applying technology.

The ability to apply the knowledge gained was 4.53, which is in the highest level. This points to the trainees' vision of the potential of the content for use in teaching or in the elaboration of activities for students, especially the integration of programming, microcontrollers, IoT systems, and artificial intelligence with the control of robots. However, this evaluation is the intention and awareness of the trainees; therefore, follow-up should be done after the training to assess how the trainees can apply knowledge in the classroom or develop learning materials.

The training evaluation results were high for all categories, but the training duration's suitability had the lowest average score of 4.37 with a standard deviation of 0.90. This indicates that trainees had a greater diversity of opinion about the adequacy of the training duration than any other item. This result is in line with the open-ended suggestion of at least four trainees to increase the duration of the training. The activities performed in the 6-hour period encompassed many steps, from basic knowledge and circuit assembly to programming, error checking, and application of artificial intelligence, which may make it impossible for those with limited experience to perform all the steps in time.

The overall average score on knowledge and skills acquired from the training was 4.05 with a standard deviation of 0.69, which indicates a high level of expertise. Automation and IoT Understanding and Skills in Electronic Circuit Design had the highest average score at 4.16. The highest average score was 4.00 for microcontroller programming and problem-solving logic skills, and the lowest average score of 3.95 was on the ability to design automated systems appropriate for real-world applications. These results, while still high, show that integrated system design is a complex skill requiring knowledge of hardware, software, and problem-solving processes. Short-term training can teach participants to understand principles and to follow examples, but not to design new systems on their own.

The self-assessment results indicated that the mean score of knowledge and understanding before training was 3.32 (SD = 1.29), which was at a moderate level. The score was improved to 4.32 (SD = 0.58) after training, indicating a high level, with an average increase of 1.00 points. Pairwise comparisons showed that post-training scores were significantly higher than pre-training scores at the .05 level. Eleven respondents reported increased knowledge, eight respondents reported no change in knowledge, and no respondents reported a decrease in their knowledge. This finding is consistent with the results of the knowledge test, which indicated that the average post-training score was higher than the pre-training score, indicating that the activity helped to promote the knowledge measured by the test and the perceived competence of the trainees.

The overall mean of organizational thinking and application was 4.24, indicating a high level of satisfaction. The mean score of the ability to organize thoughts for the development of AI was 4.26, and the mean score of the ability to organize thoughts for systematic development was 4.21. This suggested that the activity not only helped in the development of skills in the use of equipment or programming but also empowered the trainees to understand the sequence of system development processes, from problem definition, device connection, code writing, testing, and system improvement.

Open-ended suggestions supported the qualitative results, such as trainees' requests for longer practical training, more detailed circuit diagrams and manuals, a bigger screen size, online lessons on basic programming before the training, and the inclusion of Python content. Furthermore, there were suggestions to hold similar activities again, to develop new programming curricula, and to implement the activities for students at schools. These suggestions reflect the fact that trainees value and desire to use the knowledge learned in their learning context.

In conclusion, programming training on the control of robotic vehicles using symbols had the highest satisfaction level and increased the knowledge, skills, and perceived abilities of the trainees. The activity received high evaluations due to its interactive nature. However, we need to allocate more time and support for developing skills in integrated programming and system design. The next training should increase its duration. These activities are categorized into basic and advanced levels. Pre-training lessons are prepared, and a manual is written with circuit diagrams, sample code, and troubleshooting instructions to cater to trainees with different backgrounds.

VI. Discuss the Results

Results of the research show a tendency to improve the participants' knowledge through workshops connecting AI programming, networks, and hardware. The average score increased by 4.89 points, and the effect size was large. The analysis could not be matched individually. These findings are in line with a notable rise in self-assessment knowledge and a rise in the number of participants reaching at least 60% from 3 to 10.

The largest increase was in programming and development tools, likely due to activities that allowed trainees to immediately see the results of commands in the movement of the vehicle, checking results via the serial monitor, and iterative error correction. This process aligns with the components of computational thinking that emphasize algorithms, testing, and error correction and aligns with the study that found hands-on professional development improves teachers' knowledge and confidence in computational thinking [2].

The equipment, sensors, and motor control were clearly better, but the scores after training were the lowest. This is because the hardware problems can be solved in many ways, including pin connections, power supply, common ground, control signals, and motor direction. Participants from different backgrounds may spend different amounts of time checking circuits. This result is consistent with the lower score for the appropriateness of time compared to other items and suggests increasing the time and providing a practical guide.

The highest satisfaction on the usefulness of the workshop and teaching equipment/media shows that the piece of work helps create meaning to learning. The robot was a channel through which the trainees could see how the information was flowing from symbols to models, programs, networks, and motors. This result is in line with Kılıç and Çakıroğlu [3] who proposed the use of real-world problem activities, concrete examples, and reflection in the development of robotics teachers in order to link subject matter knowledge with learning management.

Teachable Machine enabled non-experts to build models for image classification within a limited timeframe. This aligns with the concept of creating tools for non-expert machine learning users [8]. But the use of simplified tools should not reduce the need for AI literacy. Future training should include activities on data balancing, splitting training and testing data, confusion matrices, model confidence, and where AI fails and gets it wrong. Also, emphasize that the none class and stop conditions serve as safety mechanisms rather than being merely general data classes.

The results of the study are in line with the suggestion of the review of Espinal et al. [4] that teacher development should not be limited to the increase of basic knowledge. Teachers should be able to plan, assess, and carry out the activities in real-life situations. The next step in the curriculum should be for trainees to develop lesson plans, robotics assignments, or assessment criteria relevant to their subjects and track classroom progress after the training.

VII. Summary of Results

The 6-hour workshop successfully merged knowledge of microcontrollers, programming, AI image classification, and robotic vehicle control into one activity, allowing participants to build, test, and refine real systems. The research results showed that there was a significant increase in knowledge scores after training compared to before training. The activity was rated at the highest level of satisfaction, and increased knowledge and understanding were reported by the participants, especially in the areas of programming and the use of system development tools. The findings show that the developed activity has the potential to be a model for the development of teachers' competencies in AI, automation, robotics, and computational thinking and applying the concepts to the design of learning activities in schools.

In terms of the activity itself, the curriculum should be divided into online preparation and in-person practical training phases or be extended to two days so that participants have sufficient time to learn, experiment, and correct errors. A manual containing circuit diagrams, sample programs, and basic troubleshooting checklists should be provided. Group facilitators should be available to assist participants of different levels of background. For system evaluation, different test datasets with respect to training datasets should be used, and accuracy, confusion tables, response times, and safety stop events should systematically be recorded. Further follow-up after training should be done to study the application of the knowledge in the classroom, possibly by using lesson plans, teacher projects, developed activities, or student learning outcomes. To gather more robust evidence of the effectiveness and sustainability of teacher professional development.

References

[1] F. Miao and M. Cukurova, *AI Competency Framework for Teachers*. Paris, France: UNESCO, 2024.

[2] A. Saxena and M. M. Chiu, "Developing preschool teachers' computational thinking knowledge, attitudes, beliefs, and teaching self-efficacies: A curriculum-based professional development program," *Frontiers in Education*, vol. 7, Art. no. 889116, 2022, doi: 10.3389/feduc.2022.889116.

[3] S. Kılıç and Ü. Çakıroğlu, "Design, implementation, and evaluation of a professional development program for teachers to teach computational thinking via robotics," Technology, Knowledge and Learning, vol. 28, no. 4, pp. 1539–1569, 2023, doi: 10.1007/s10758-022-09629-3.

[4] A. Espinal, C. Vieira, and A. J. Magana, "Professional development in computational thinking: A systematic literature review," *ACM Transactions on Computing Education*, vol. 24, no. 2, Art. no. 27, pp. 1–24, 2024, doi: 10.1145/3648477

[5] Z. Liu, Z. Gearty, E. Richard, C. H. Orrill, S. Kayumova, and R. Balasubramanian, "Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K–12 classrooms," *International Journal of STEM Education*, vol. 11, Art. no. 51, 2024, doi: 10.1186/s40594-024-00510-6.

[6] T. L. Kurz, S. Jayasuriya, K. Swisher, J. Mativo, R. Pidaparti, and D. T. Robinson, "The impact of Teachable Machine on middle school teachers' perceptions of science lessons after professional development," *Education Sciences*, vol. 14, no. 4, Art. no. 417, 2024, doi: 10.3390/educsci14040417.

[7] R. Hijón-Neira, C. Pizarro, O. Borrás-Gené, and S. Cavero, "AI-generated context for teaching robotics to improve computational thinking in early childhood education," *Education Sciences*, vol. 14, 2024.

[8] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen, "Teachable Machine: Approachable web-based tool for exploring machine learning classification," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2020, pp. 1–8, doi: 10.1145/3334480.3382839.