Smart Motor: A Low-Cost Hardware and Software Toolkit for Introducing Supervised Machine Learning to Elementary School Students

Tanushree Burman[1], **Milan Dahal**[2], **Geling Xu**[3],
Chris Rogers [2], **Jennifer Cross** [4], **Jivko Sinapov**[1]

[1]Tufts University, Department of Computer Science
[2]Tufts University, Department of Mechanical Engineering
[3]Tufts University, Department of STEM Education
[4]Tufts University, Center for Engineering Education and Outreach
{tanushree.burman, milan.dahal, geling.xu, chris.rogers, jennifer.cross, jivko.sinapov}@tufts.edu

Abstract

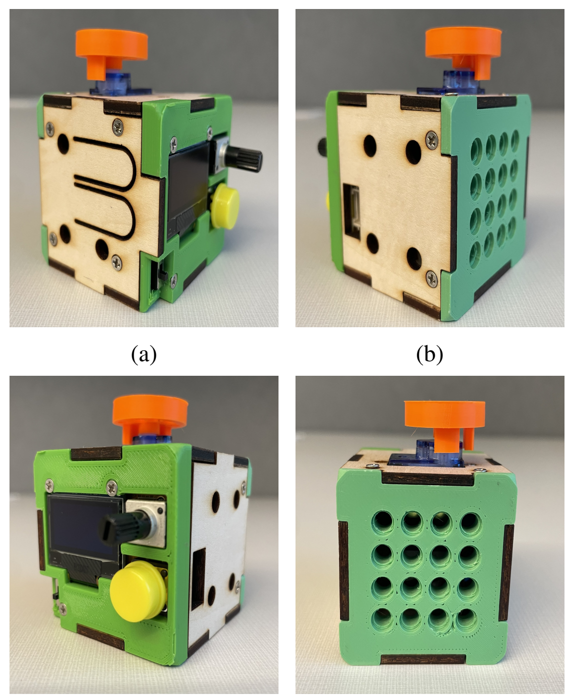
With the rise of Artificial Intelligence (AI) systems in society, our children have routine interactions with these technologies. It has become increasingly important for them to understand how these technologies are trained, what their limitations are and how they work. To introduce children to AI and Machine Learning (ML) concepts, recent efforts introduce tools that integrate ML concepts with physical computing and robotics. However, some of these tools cannot be easily integrated into building projects and the high price of robotics kits can be a limiting factor to many schools. We address these limitations by offering a low-cost hardware and software toolkit that we call the Smart Motor to introduce supervised machine learning to elementary school students. Our Smart Motor uses the nearest neighbor algorithm and utilizes visualizations to highlight the underlying decision-making of the model. We conducted a one week long study using Smart Motors with 9- to 12- year old students and measured their learning through observation, questioning and examining what they built. We found that students were able to integrate the Smart Motors into their building projects but some students struggled with understanding how the underlying model functioned. In this paper we discuss these findings and insights for future directions for the Smart Motor.

Smart Motor — https://smartmotors.notion.site/
Smart App — https://smart-motors.web.app/

Introduction

The integration of Artificial Intelligence (AI) systems into everyday life has increased in recent years. Children are introduced to these systems through virtual assistants such as Siri and Amazon Alexa, AI-powered recommendation systems for content platforms, and AI-powered toys. As these technologies become more common in our daily lives, it is crucial to understand how they are trained, what their limitations are, and how they work.

To integrate AI education into K-12 school curricula, initiatives like AI4K12 have been developed, outlining key concepts or "Big Ideas" of AI that all students should know (Touretzky et al. 2019; Broll and Grover 2023). Currently, pre-college AI education programs that introduce supervised machine learning concepts are very limited but rapidly growing in number. Some of the publicly accessible educational AI tools include *Teachable Machine*, *Cognimates*, and *AI for Oceans* (Yang 2022; Sabuncuoglu 2020). *Teachable Machine* allows users to train models to recognize images and sounds and is one of the most popular methods of teaching ML (Li, Fengchao, and Zhang 2024). Similarly, *AI for Oceans* introduces training datasets where users play a game to provide training data.



(a) (b) (c) (d)

Figure 1: (a) Smart Motor hardware where the left face has navigation buttons and (b) the right face has a sensor port. (c) The front face has an OLED screen and (d) the back face has LEGO® compatible holes and a motor on top. The button and dial shown in (c) were included for other operation modes not used in this workshop.

Although these software are good baselines for introducing ML concepts to children, they are typically limited to the virtual world. In our work, we seek to enhance child engagement through tangible exploration with robotics. Robotics has been used to encourage critical thinking, problem-solving, and collaborative skills among students while introducing technological skills (López-Belmonte et al. 2021). Our goal is to combine ML concepts with robotics activities to boost engagement and develop technical skills.

In this paper, we introduce a low-cost hardware tool that we call Smart Motor with a complementary web-based user interface called Smart App. We pilot-tested the Smart Motor and the Smart App system in a 5-day workshop to determine how the system would support students' understanding of Machine Learning concepts. This experience report discusses the integration of the devices into classroom settings, and how the activities influenced students' understanding of Machine Learning. Our next steps involve adding new features to the Smart Motor system and developing more machine-learning-integrated activities for students.

Background

Initiatives like AI4K12 establish foundational concepts that all students should grasp when learning about AI. These key ideas include understanding how computers perceive the world through sensors, how they represent their surroundings, and how they learn from data (Touretzky et al. 2019). The design of the Smart Motor aims to address these learning objectives through the constructivist education theory that students construct their own knowledge through interactive experiences (Mota-Valtierra, Rodríguez-Reséndiz, and Herrera-Ruiz 2019). We particularly build upon the concepts of providing multiple representations and scaffolding. Recent research has shown that providing multiple representations allows students to better connect abstract and tangible concepts and apply these skills in the future (Uttal et al. 2009).

There have been efforts to integrate the concepts of ML with physical computing. The micro:bit and ml-machine.org Web page allows students to train supervised machine learning models for classification and redesign of everyday objects (Bilstrup et al. 2022). In a similar effort, PlushPal is a web-based design for children to make interactive plush toys with ML. One of the limitations of the micro:bit is that integrating robotics components into the hardware, such as a motor or external sensors, requires additional materials and wiring skills. To address this limitation, a key design consideration for the Smart Motor was to ensure that the hardware includes a motor and can interface with a variety of sensors.

Studies have shown that educational robotics can integrate technical subjects, such as engineering and physics, with social topics (Anwar et al. 2019). They have shown that educational robotics supports teaching STEM concepts (Khanlari and Mansourkiaie 2015; Williams et al. 2007; Ortiz 2010) while fostering soft skills like collaboration, communication, creativity (Sahin, Ayar, and Adiguzel 2014; Okita 2014), critical thinking, problem-solving (Okita 2014) and inquiry (Ganesh et al. 2010). Among the various commercial robotic kits available, LEGO® Robotics kits are one of the more popular kits due to their wide range of applications (Souza et al. 2018; Takacs et al. 2016). Although numerous researchers have used LEGO® Robotics kits to demonstrate robotics education in classrooms, the cost of a kit is a limiting factor that hinders broader impact (Couceiro et al. 2012; Gorjup and Liarokapis 2020). Therefore, a key objective in designing our Smart Motor was to minimize cost while maintaining versatility.

In efforts to incorporate ML with robotics, LEGO® introduced machine learning through the LEGO® MINDSTORMS® Robot Inventor App in 2022 (Karalekas, Vologiannidis, and Kalomiros 2023). Machine learning is an extension of this app that allows the use of the microphone or camera to identify different objects or sounds. Although students are introduced to ML concepts through platforms like this, many current AI educational tools teach ML as a black box model. According to a survey evaluating the instructional modules dedicated to teaching ML concepts, it was reported that several of the units present ML concepts but only on an abstract level such that some of the underlying ML processes were obscure to reduce complexity for students (Marques, Gresse von Wangenheim, and Hauck 2020).

In summary, previous educational robotics tools that incorporate ML concepts are often costly and not easily integrated into building projects. Our primary goals when designing the Smart Motor were to create a system that is (1) affordable, (2) versatile for building projects to foster creativity, and (3) introduces ML concepts in a clear, transparent way through simple algorithms and multiple representations. Our Smart Motor, which includes a motor, sensor, and screen, is priced at $30.

Smart Motor

Early Smart Motor prototypes were developed using inexpensive, off-the-shelf components that could be substituted with locally available materials for adaptation and adoption in international settings by collaborators (Dahal et al. 2023). The version of the Smart Motor system used in this experience report was refined to reduce assembly time and was enhanced with a web-based user interface (Smart App) for the exploration of supervised learning.

When using a Smart Motor, the objective for the user is to train their motor behavior based on sensor input. Users can connect different sensors to the Smart Motor, such as a distance sensor, a light sensor, and a rotary, and slide potentiometer. The sensor reading is the input of the Smart Motor, and the output is the motor's position. Users can switch the Smart Motor into Train mode and train the Smart Motor by selecting motor positions corresponding to the sensor readings. During the Play mode, these sensor-motor pairings and current sensor values are used to determine the position the motor should move to in real time.

Examples of projects that students can use Smart Motors for include training the Smart Motor with a light sensor to open and close a window shade when it is sunny or dark outside. Similarly, students can use a distance sensor to open and close a door when someone is near. Our goal is for students to understand how sensors receive input from the environment, how this data can be represented, and how the Smart Motor can learn from this data.

In classrooms, the Smart Motor can be integrated with structural and decorative materials when building projects. The Smart Motors have LEGO® compatible holes so that they can be easily attached to LEGO® components. Additionally, the Smart Motors can be integrated in building with arts and crafts materials such as cardboard and construction paper.

Hardware

The Smart Motor hardware consists of a discrete servo motor connected to a Seeed Studio XIAO ESP32C3 microcontroller on a custom-printed circuit board. The Smart Motor has a cubical form factor (47mm X 47mm X 47mm) with 3D-printed front and back faces, and the rest of the faces are made of laser-cut plywood. Figure 1a and b shows that there are two navigation buttons on the left face and a Grove-compatible sensor port on the right face. Figure 1c shows the Smart Motor's front face with a button (selection button), a dial, a power switch, and a 0.96-inch monochrome OLED screen. The top face has a servo motor connected to a LEGO®-compatible wheel. There are LEGO®-compatible holes for mounting LEGO® pieces on the box's left, right bottom, and back faces. The dial and buttons were included for other operation modes not used in the workshop discussed in this paper.

Supervised Machine Learning

Supervised learning is a category of machine learning where a computational model is trained on a dataset that is composed of both input values, called examples, and matching desired outcomes, called labels. The training data of the Smart Motor is composed of a single input value from a sensor connected to the Grove-compatible port and a single value label that corresponds to the desired angular position of the servo motor.

The k-Nearest Neighbor method is one of the simpler algorithms for predicting the class of a test example, which makes it well-suited for introducing concepts of ML decision-making (Elkan 2011). For the training phase, each training example is stored with its label. To generate a predicted label for a test example, its Euclidean distance to every training example is first computed. The k closest training examples are kept, where $k \geq 1$ is a fixed integer. The label that is most common among these examples then becomes the prediction for this test example.

The Smart App use k-NN with a k of 1. We do this to reduce the complexity of the algorithm for users. Users train the Smart Motor by entering a dataset composed of sensor-motor pairings. During Play mode, the current sensor reading is used as a test example, and its Euclidean distance to every sensor reading in the dataset is calculated. The closest sensor-motor pair is kept, the motor position is used as the prediction for this test example, and the motor moves to this position. This calculation happens in real time, and the motor moves in real time depending on the current sensor reading and the training data the user enters.

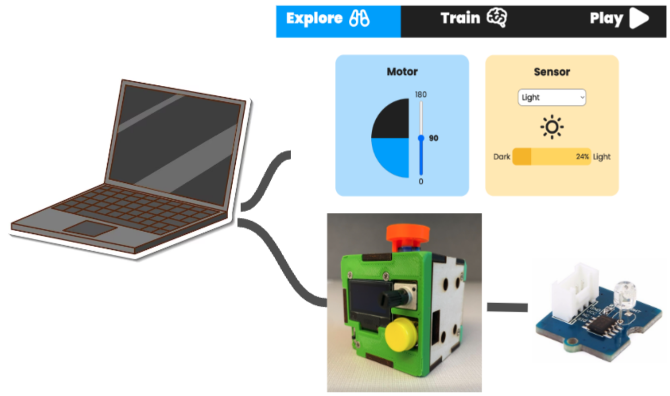


Figure 2: Smart App: Users connect the Smart Motor to their laptop and navigate to a web page that allows them to connect to the Smart Motor device, control the motor, and view sensor readings in real time.

User Interface

The Smart Motor and Smart App system allows users to train and test the Smart Motor and add and delete individual training data points. As shown in Figure 2, students can connect their Smart Motor to a computer or other personal device via USB cable and navigate to a web page that will allow them to connect to the Smart Motor. Students can then use this web page to control the motor and view sensor readings in real-time. Since the computer interface is provided as a web page, it is compatible with many platforms, including smartphones and tablets, enabling the system to work in various classroom technology environments.

The Smart App is designed to give students more detailed and user-friendly visualizations of the training data and the sensor and motor values. The design of the Smart App interface uses multiple representations to support student development of conceptual understandings of ML. It provides visually distinct representations with a blue pie chart and slider representing the motor position and a yellow progress bar for the sensor reading. The sensor and motor readings are then represented together through a scatter plot and a table that contains the sensor and motor training data. These numerical and graphical representations are available to students to help them better connect how the motor and sensor readings correlate to the real-time motor and sensor states. These features provide multiple representations of how the sensor and motor values are perceived. Finally, the scatter plot represents the underlying model and during Play mode highlights how the current sensor reading informs the algorithm's decision-making in real time.

The Smart App consists of an Explore tab to encourage users to experiment with the sensor and motor (Figure 3), a Train tab where users can add sensor and motor values (Figure 4-top), and finally, a Play tab where they can let the Smart Motor run on the training values they input (Figure 4-bottom). In the graph, the yellow vertical line moves in real time, which is associated with the sensor reading, and the blue horizontal line can be manually moved to change the

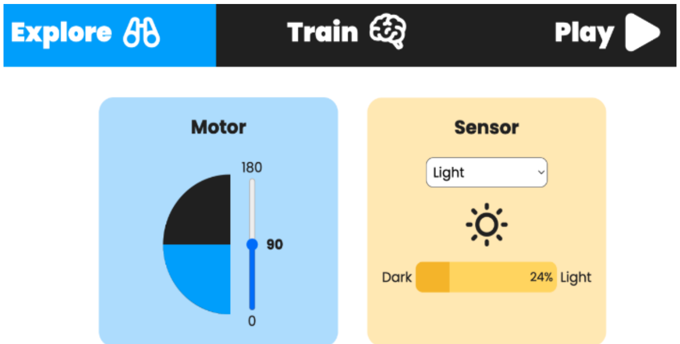


Figure 3: Smart App Explore: Users can choose their sensor and drag the motor slider to change the angle of the motor.

motor position. Any added data points are saved such that disconnecting the Smart Motor from the Smart App preserves added data values.

Methodology

Workshop Overview

We conducted a workshop with 24 9- to 12-year-old students (13 male and 11 female). Our goal with this workshop was to determine how the Smart Motor and Smart App system supported students' projects and their understanding of ML concepts. The students were part of a 5-day educational summer program in St. Louis, MO. The workshop was conducted during a robotics program with a total of 10 hours of instructional time. During this workshop, we provided students with LEGO® pieces and craft materials such as cardboard, construction paper, and felt fabric sheets. Each student was provided with a laptop, USB cable and Smart Motor. They also had four compatible sensors made available to them: a distance sensor, a light sensor, and a rotary, and a slide potentiometer. Two elementary school teachers led the workshops, and three researchers were present as facilitators and collected data. Two months before this workshop, the elementary school teachers participated in a co-design work to develop the curriculum and Smart Motor system during a two-day workshop led by a member of the research team (Xu, Dahal, and Gravel 2024).

Data Collection

The primary data collection method was video footage of students working on their projects. Cameras were mounted over the group workstations to record the conversation and activity for the duration of the workshop. Researchers also recorded observations and field notes of students working. We recorded students' design choices and conversations while building and training with Smart Motors. All the video footage was compiled and played back. Video footage containing students' conversations about the design process, building, and training were identified. These clips were used to analyze students' progression and understanding of the

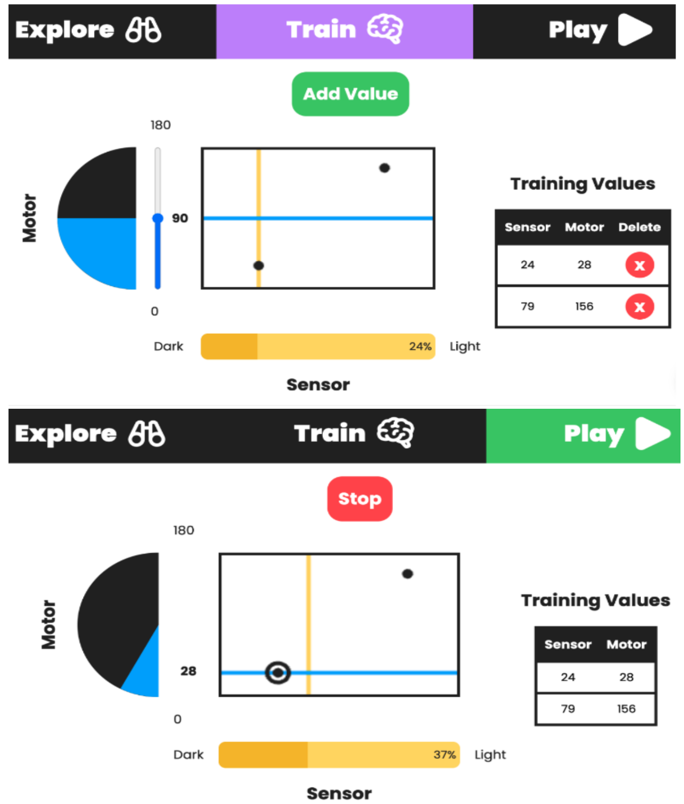


Figure 4: Smart App Train (top): 'Add Value' allows users to add sensor and motor pairs to their training data. Users can also delete data points from the table where the corresponding data point on the graph is highlighted. The yellow vertical line refers to the sensor reading, and the blue horizontal line refers to the angle of the motor in real life. Smart App Play (bottom): After adding all training data, the 'Play' button allows users to test the data. The graph highlights the data point used to inform the algorithm's decision-making.

Smart Motors system through the design, build, and training process. After the initial analysis of the videos by the first author, these clips and notes were shared with a second co-author to confirm the conversation topics and details. Identified conversations among students were organized into the following categories: (1) conversations surrounding the training of the Smart Motor and (2) design choices for each activity.

Novel Engineering-inspired Literacy Integration with *MunchA! MuncHa! MuNcha!*

The workshop had an activity inspired by the concept of Novel Engineering (Portsmore and Milto 2018). In a Novel Engineering curriculum, students are given reading materials and tasked to identify problems the characters face. They then have to use the engineering design process to build a solution for a character of their choice. During the workshop, students watched a video reading of the book *MunchA! MuncHa! MuNcha!* In this book, Mr. McGreely is a farmer who owns a garden, and every night, three rabbits come to his garden and eat his vegetables. Every day, he is upset, and he builds contraptions for his garden to protect his vegetables from the three rabbits. Students were instructed to determine the main characters in the story and identify the problems each character faces. They were then asked to design and build a solution to solve one of the problems they noticed using the Smart Motor system.

Workshop Design

The workshop was conducted in four sessions: a discussion of Artificial Intelligence and Machine Learning, an introduction activity with the Smart Motors and Smart App system, a Novel Engineering-inspired *MunchA! MuncHa! MuNcha!* activity, and a final engineering activity.

Session 1: Discussion of Artificial Intelligence and Machine Learning To introduce students to the concept of Artificial Intelligence, students were first given AI games to play with. These include "Quick, Draw!" where a neural network attempts to guess a drawing, and "Shadow Art." The teachers also led a class discussion on what AI is and examples of AI they may have encountered in real life.

Session 2: Introduction to the Smart Motors and Smart App system In the second session, students were introduced to Smart Motors. They were given a laptop, a Smart Motor, and a sensor of their choice. Students explored the Smart Motors and Smart App to understand how everything worked together at their own pace. The teachers then led a discussion on how AI is used with the Smart Motors and the different parts of the Smart Motor and Smart App system.

Session 3: Novel Engineering-inspired Design with *MunchA! MuncHa! MuNcha!* In this session, students watched a video of the book. They were then asked to pick either Team Rabbit or Team Farmer. Students were given time to brainstorm problems the characters encounter and how they can help the character solve those problems. They were then tasked to design and build solutions for the characters' problems.

Students were also provided with cardboard and arts and crafts materials to help build their structures. For this activity, students were given a Sphero Mini (small robot ball) that they controlled using a tablet. They placed a cup over the Sphero Mini to give the illusion of a rabbit moving around. The teachers demonstrated example projects using the distance and light sensor for team Farmer. The teacher also demonstrated the process of training the Smart Motor. Session 3 was conducted on the second and third day with 4 hours of instruction. Students formed groups of 2-3 for this activity.

Session 4: Final Engineering Activity For the final activity, students were instructed to choose a topic for a problem they are passionate about. These included building a community garden, a future city, a maze, or an escape room. The goal was to incorporate the Smart Motors system into their project. The teacher demonstrated an example maze project incorporating a Smart Motor and rotary potentiometer as the sensor. Students formed groups of 2-3 for this activity and had the opportunity to build upon their projects from Session 3. Session 4 was conducted on the fourth and fifth day with 4 hours of instruction.

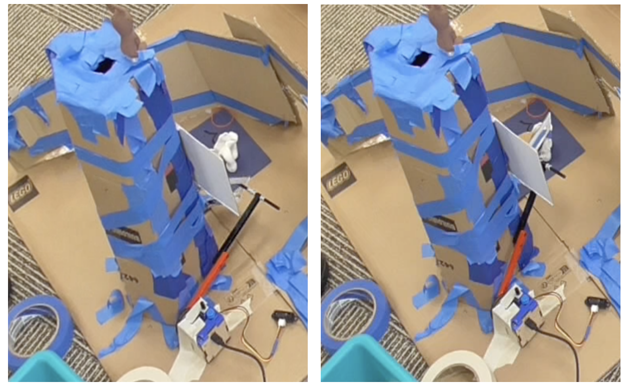


Figure 5: Group 1 used a Smart Motor to open and close a door to capture rabbits with a distance sensor for the Novel Engineering-inspired activity in session 3.

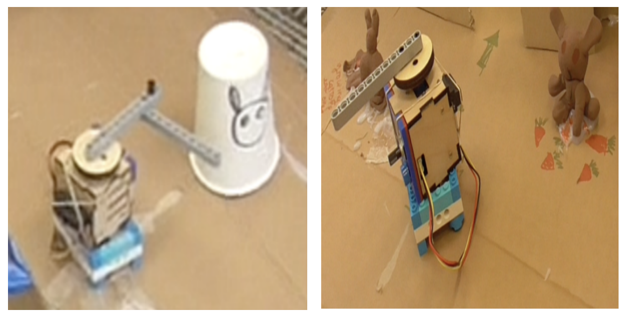


Figure 6: Group 1 used a Smart Motor and distance sensor to hit rabbits for the Novel Engineering-inspired activity in session 3.

Results

We present case studies of two groups of students and their design development throughout the workshop. We selected groups who were more verbose throughout the workshops and volunteered to participate with researchers when prompted about the progress of their projects. We focused on the design and build process for sessions 3 and 4 because these activities involved open-ended use of the Smart Motors. In the sections below, we discuss the sensors students used and how they integrated the Smart Motor system with their projects. We also discuss conversations students had about Machine Learning, concepts they struggled with, and future directions for integrating Machine Learning with Smart Motors.

Group 1: During the Novel Engineering-inspired activity in session 3, this group discussed they would like to be Team Farmer and build mechanisms to capture the rabbits. They used a Smart Motor to open and close a door to capture rabbits and a distance sensor to locate when the door should open and close (Figure 5). In addition to capturing the rab-

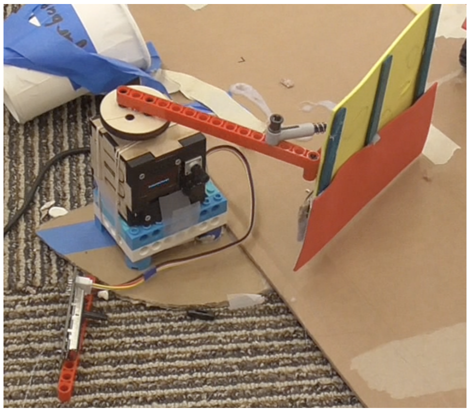


Figure 7: Future City: Group 1 used a Smart Motor to scoop trash for the final engineering activity.

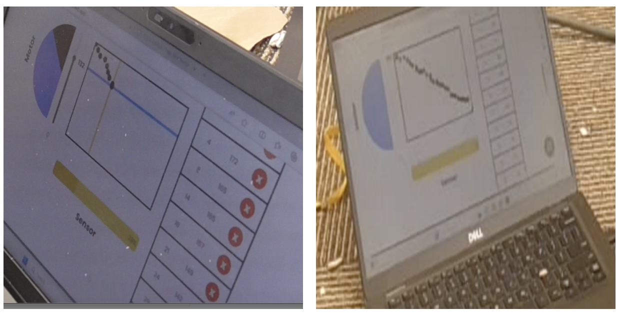


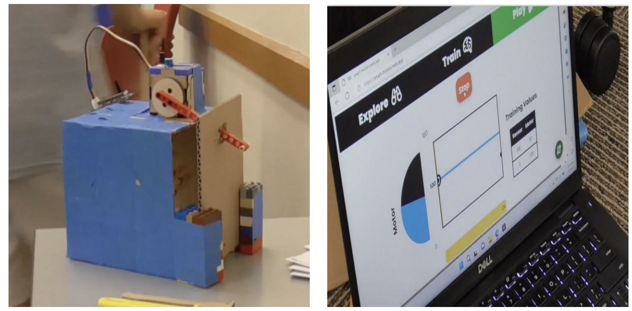
Figure 8: Future City: Training data for garbage scooper that Group 1 built for the final engineering activity.

bits, they used a second Smart Motor and distance sensor to hit the rabbits when they came close to the Smart Motor (Figure 6).

"We're farmers, so we have two motors, one that opens the door and one that hits. For both of them, I use a distance sensor because this is an automatic door, and this is to hit the bunnies." The group trained the Smart Motors using two data points where a lower sensor value corresponded to one extreme motor position and a higher sensor value corresponded to the other.

For the final engineering activity, the group decided to build a future city where they built upon their team Farmer project. In this future city, there was trash all over the streets, and they decided to use Smart Motors to help clean the city. "We'll make a robo thingy that will clean it (the trash)." "Let's attach something to that (Smart Motor) that can pick up the trash."

The group decided to build something that could scoop the trash from the city street into a cup. To build the garbage scooper, the group attached a beam to the motor, which was then attached to a cardboard piece that allowed for a scooping motion when the motor moved (Figure 7). When training the garbage scooper, the team decided to use a slider to move the scooper back and forth. However, unlike the two mechanisms from the Novel Engineering-inspired activity, they did not use only two data points (Figure 8).



(a) Smart Motor opens and closes the door vertically using a slider.

(b) Two data points are used when training the Smart Motor.

Figure 9: Group 2 used a Smart Motor to capture rabbits in a box for the Novel Engineering-inspired activity in session 3.

"So right now I'm just using the slider, I'm coding, kind of training it... I want to use a lot of values." When asked what using a lot of values meant and why do that, the student said "It's more specific. If you just use two data values then its more of a on and off switch." "If I have a lot of data points, when I move a tiny little bit, it will still move but if you have 3 or 2 values then it wouldn't move specific like this. . . " The student showed how the scooper moved even though they only moved the slider a little bit. The way this group was able to recognize the impact of several training data points versus only two and how these data points affect the movement of the motor was noteworthy.

Group 2: During the Novel Engineering-inspired activity in session 3, this group discussed they would also like to be Team Farmer and build a contraption to capture the bunnies. "We are going to use a distance sensor so that it will go down (referring to cardboard door) when a rabbit comes." This group used a Smart Motor to open and close a vertical door that they attached to a beam on the motor. Similar to Group 1, they also used two data points to train their Smart Motor but towards the end of their project they opted to use a slider instead of a distance sensor (Figure 9).

For the final engineering activity, Group 2 chose to build a maze by adding on to their project from session 3 (Figure 10). This group used a second Smart Motor that had two LEGO® beams attached to the motor that would move up and down to act as an obstacle in the maze. A user would navigate the maze by using a tablet to move a Sphero Mini and would have to try to navigate the ball through the obstacle. To build the obstacle, the students in this group chose to use a distance sensor that was attached to the bottom of the maze facing the ceiling. They trained the Smart Motor by inputting data points of extreme motor positions with very close sensor readings (Figure 11). Since the distance sensor value was often noisy, training the Smart Motor in this way caused random movements and this random behavior was what they intended for the beam to do.

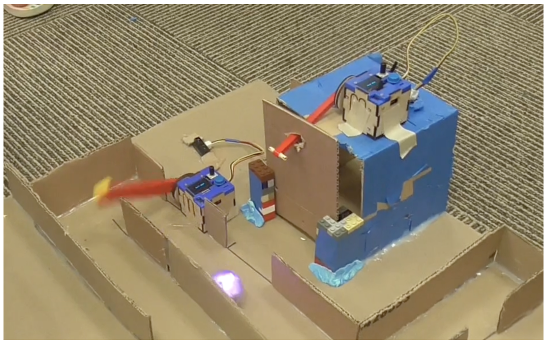


Figure 10: Maze: Group 2 built a maze for their final engineering activity using a Smart Motor as an obstacle.

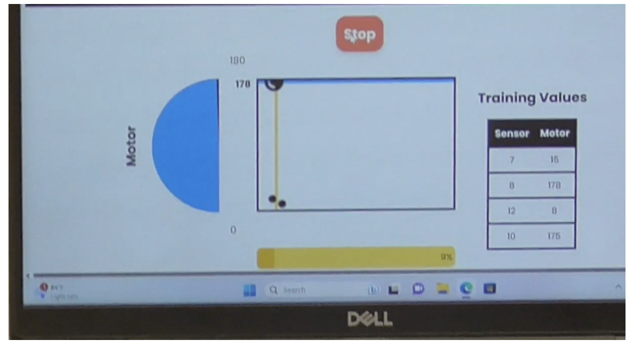


Figure 11: Maze: Group 2 training data for maze obstacle course in session 4.

Discussion

During this workshop experience, we found that students were excited to work with the Smart Motor system and had positive feedback regarding the user interface. In general, students were able to integrate the Smart Motor and Smart App system with their building projects. When training their Smart Motors, we found that most students chose to use two data points of extreme sensor and motor pairs which would create a sweeping motion back and forth by the motor. Since examples of training was demonstrated by the teacher using two data points, it was not clear if all students understood how the underlying algorithm worked or if they were following the demonstrations led by the teacher. We also found that students spent a majority of the time building the physical structure and less time on the training of the Smart Motors. To encourage students to explore more with the training of the Smart Motors, we will incorporate more guidelines in future activities such as requiring that the motor moves to at least three different positions. Additionally, to reduce the amount of time spent building, we will limit the size of the physical structure that students can build. We also plan to use a WiFi-connected app rather than using serial communication because we found that the USB cable limited the location that the Smart Motor could be mounted.

The conversation with students from Group 1 explaining why they used so many data points to train their garbage scooper rather than choosing only a couple was significant because they recognized that having several data points versus only a couple affects the movement of the motor. When they described that having two data points is like an on or off switch especially shows their understanding of having a few versus several training data points. At the same time, a careful look at the way this group trained their Smart Motor suggests that these students were hoping to have a sort of linear relationship between the sensor and motor data. For future directions with the Smart Motor, we hope to incorporate more Machine Learning algorithms so that students can learn about the different algorithms and when one may be preferred over the other. Additionally, our current algorithm uses a k value of 1 but we will allow students to change this value and develop activities that will allow them to explore this concept.

In Group 2, we observed how the students were able to use a limitation of the system to their advantage. The distance sensor reading in general can be noisy and it was creative on the students' part to use this noisy reading to their advantage by training the random behavior of the motor. Having training data with extreme motor positions with very close sensor readings allowed for the Smart Motor to move back and forth randomly. This worked well when other users navigated the Sphero Mini through the maze. The use of the sensors with the Smart Motors introduced students to how data can be noisy. Moving forward, we aim to develop Smart Motor activities that provide opportunities for students to investigate how noisy data can impact the performance of a Machine Learning algorithm and cause limitations.

Conclusion and Future Work

Our goal in developing the Smart Motor and Smart App system was to introduce a low-cost hardware tool that students can integrate into physical projects while learning about Machine Learning concepts. Specifically, we wanted for students to understand how computers understand their environment through sensors, how this data can be represented and how it can be used to learn. Through our pilot workshop, we found that students were able to integrate the Smart Motors with their projects and had conversations regarding Machine Learning, specifically about training data and how it affects the output of the Smart Motor. For future directions, we hope to have several Machine Learning algorithms available on the Smart Motor system.

One of the limitations of our workshop was that it was conducted during an out-of-school STEM summer program. As a result, the students likely had higher than average preexisting interest in STEM and robotics topics. Additionally, our participant numbers were limited by both the small pilot workshop size and the limited number of students assenting to be video recorded for this study. To collect more generalizable student data, we plan to integrate similar activities into in-school elementary classes.

Acknowledgements

We would like to thank Steven Coxon, Dustin Nadler, Karen Engelkenjohn and the teachers at Maryville University's

Center for Access and Achievement (CA2) for their work running the Smart Motor programs. We also thank Carlie Sylvan and Aengus Kennedy for their contributions to the software and hardware in this effort. This material is based upon work supported by the National Science Foundation under Grant No. IIS-2119174.

References

Anwar, S.; Bascou, N. A.; Menekse, M.; and Kardgar, A. 2019. A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, 9(2): 2.

Bilstrup, K.-E. K.; Kaspersen, M. H.; Sørensen, M.-L. S. K.; and Petersen, M. G. 2022. Opportunities and Challenges of Teaching Machine Learning as a Design Material with the micro:bit. In *Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference*, NordiCHI '22. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394482.

Broll, B.; and Grover, S. 2023. Beyond black-boxes: teaching complex machine learning ideas through scaffolded interactive activities. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press. ISBN 978-1-57735-880-0.

Couceiro, M. S.; Figueiredo, C. M.; Luz, J. M. A.; Ferreira, N. M.; and Rocha, R. P. 2012. A Low-Cost Educational Platform for Swarm Robotics. *International Journal of Robots, Education & Art*, 2(1).

Dahal, M.; Kresin, L.; Peres, A.; Pereira, E.; and Rogers, C. 2023. International Collaboration to Increase Access to Educational Robotics for Students. 1–5.

Elkan, C. 2011. Nearest neighbor classification. *elkan@ cs. ucsd. edu‖,‖ January*, 11: 3.

Ganesh, T.; Thieken, J.; Baker, D.; Krause, S.; Roberts, C.; Elser, M.; Taylor, W.; Golden, J.; Middleton, J.; and Kurpius, S. R. 2010. Learning through engineering design and practice: Implementation and impact of a middle school engineering education program. In *2010 Annual Conference & Exposition*, 15–837.

Gorjup, G.; and Liarokapis, M. 2020. A low-cost, open-source, robotic airship for education and research. *IEEE Access*, 8: 70713–70721.

Karalekas, G.; Vologiannidis, S.; and Kalomiros, J. 2023. Teaching Machine Learning in K–12 Using Robotics. *Education Sciences*, 13(1).

Khanlari, A.; and Mansourkiaie, F. 2015. Using robotics for STEM education in primary/elementary schools: Teachers' perceptions. In *2015 10th International Conference on Computer Science & Education (ICCSE)*, 3–7. IEEE.

Li, L.; Fengchao, Y.; and Zhang, E. 2024. A systematic review of learning task design for K-12 AI education: Trends, challenges, and opportunities. *Computers and Education: Artificial Intelligence*, 100217.

López-Belmonte, J.; Segura-Robles, A.; Moreno-Guerrero, A.-J.; and Parra-González, M.-E. 2021. Robotics in Education: A Scientific Mapping of the Literature in Web of Science. *Electronics*, 10(3).

Marques, L. S.; Gresse von Wangenheim, C.; and Hauck, J. C. 2020. Teaching machine learning in school: A systematic mapping of the state of the art. *Informatics in Education*, 19(2): 283–321.

Mota-Valtierra, G.; Rodríguez-Reséndiz, J.; and Herrera-Ruiz, G. 2019. Constructivism-Based Methodology for Teaching Artificial Intelligence Topics Focused on Sustainable Development. *Sustainability*, 11(17).

Okita, S. Y. 2014. The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology*, 45(5): 844–862.

Ortiz, A. M. 2010. *Fifth grade students' understanding of ratio and proportion in an engineering robotics program.* Tufts University.

Portsmore, M.; and Milto, E. 2018. Novel engineering in early elementary classrooms. *Early engineering learning*, 203–223.

Sabuncuoglu, A. 2020. Designing one year curriculum to teach artificial intelligence for middle school. In *Proceedings of the 2020 ACM conference on innovation and technology in computer science education*, 96–102.

Sahin, A.; Ayar, M. C.; and Adiguzel, T. 2014. STEM Related After-School Program Activities and Associated Outcomes on Student Learning. *Educational Sciences: Theory and Practice*, 14(1): 309–322.

Souza, I. M. L.; Andrade, W. L.; Sampaio, L. M. R.; and Araujo, A. L. S. O. 2018. A Systematic Review on the use of LEGO® Robotics in Education. In *2018 IEEE Frontiers in Education Conference (FIE)*, 1–9.

Takacs, A.; Eigner, G.; Kovacs, L.; Rudas, I. J.; and Haidegger, T. 2016. Teacher's Kit: Development, Usability, and Communities of Modular Robotic Kits for Classroom Education. *IEEE Robotics Automation Magazine*, 23(2): 30–39.

Touretzky, D.; Gardner-McCune, C.; Martin, F.; and Seehorn, D. 2019. Envisioning AI for K-12: what should every child know about AI? In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. ISBN 978-1-57735-809-1.

Uttal, D. H.; O'Doherty, K.; Newland, R.; Hand, L. L.; and DeLoache, J. 2009. Dual Representation and the Linking of Concrete and Symbolic Representations. *Child Development Perspectives*, 3(3): 156–159.

Williams, D. C.; Ma, Y.; Prejean, L.; Ford, M. J.; and Lai, G. 2007. Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp. *Journal of research on Technology in Education*, 40(2): 201–216.

Xu, G.; Dahal, M.; and Gravel, B. 2024. Exploring K-12 Teachers' Confidence in Using Machine Learning Emerging Technologies through Co-design Workshop (RTP). In *2024 ASEE Annual Conference & Exposition*.

Yang, W. 2022. Artificial Intelligence education for young children: Why, what, and how in curriculum design and implementation. *Computers and Education: Artificial Intelligence*, 3: 100061.