PoseBlocks: A Toolkit for Creating (and Dancing) with AI

Brian Jordan, Nisha Devasia, Jenna Hong, Randi Williams, Cynthia Breazeal
MIT Media Lab
bcjordan@media.mit.edu, ndevasia@mit.edu, jhong19@mit.edu, randiw12@media.mit.edu, cynthiab@media.mit.edu

Abstract

Body-tracking artificial intelligence (AI) systems like Kinect games, Snapchat Augmented Reality (AR) Lenses, and Instagram AR Filters are some of the most engaging ways students experience AI in their everyday lives. Additionally, many students have existing interests in physical hobbies like sports and dance. In this paper, we present PoseBlocks; a suite of block-based programming tools which enable students to build compelling body-interactive AI projects in any web browser, integrating camera/microphone inputs and body-sensing user interactions. To accomplish this, we provide a custom block-based programming environment building on the open source Scratch project, introducing new AI-model-powered blocks supporting body, hand, and face tracking, emotion recognition, and the ability to integrate custom image/pose/audio models from the online transfer learning tool Teachable Machine. We introduce editor functionality such as a project video recorder, pre-computed video loops, and integration with curriculum materials. We discuss deploying this toolkit with an accompanying curriculum in a series of synchronous online pilots with 46 students, aged 9-14. In analyzing class projects and discussions, we find that students learned to design, train, and integrate machine learning models in projects of their own devising while exploring ethical considerations such as stakeholder values and algorithmic bias in their interactive AI systems.

Introduction

Physical movement is one of the most engaging and increasingly common mechanisms for interacting with AI systems, but students rarely have the opportunity to engage with it in K-12 AI curricula. Many middle school students have passionate interests in dance, art, physical movement in sports, and video games that involve physical motion which aren't easy to construct within the typical block-based coding environments found in classrooms. With the PoseBlocks toolkit, we introduce a suite of AI-integrated block-based coding tools used to complement learning activities in which students conceptualize, design, build, and reflect on interactive physical-movement-based multimedia experiences. Students can build interactive AI projects with two new sets of AI-powered blocks: (1) hand/body/face position-tracking and expression-detecting blocks, and (2) blocks that allow students to import their own custom trained image and pose recognition models from Google's Teachable Machine (Carney et al. 2020). With these new tools, students can design and build naturally interactive AI-powered projects that align with their interests, train their own supervised machine learning models, and reason about the ethics and presence of AI systems in their everyday lives.

In this paper we discuss the design and development of these blocks, additional affordances added to the PoseBlocks editor, materials developed to support use in classrooms, and the results of piloting these tools with 46 students aged 9-14. We examine the projects students created with the toolkit and reflections from students and their teachers who participated in the pilots. Students quickly learned to design, train, and integrate custom machine learning models into their drag-and-drop programming projects, and they designed projects that incorporated real-world elements relevant to their interests and surroundings.

Background

AI Education Many children growing up today have access to AI-enabled devices from an early age. However, studies have shown that children lack understanding of how their smart toys and AI assistants work (Druga et al. 2017; McReynolds et al. 2017). Education and awareness are key to resolving such misunderstandings. Understanding how AI works affects students' perceptions of smart devices; a better understanding leads to belief that AI devices are not as smart as humans (Williams, Park, and Breazeal 2019), and gives children a framework to safely interact with the smart devices in their lives.

The concept of teaching AI to children dates back many decades (Papert et al. 1971), but researchers have only just begun to systematically study and develop standards for AI education more broadly (AAAI 2018; Long and Magerko 2020). Many researchers and teachers have created workshops and curricula to teach AI concepts to K-12 students (Karaman et al. 2017; Tang 2019; Van Brummelen 2019; Lin et al. 2020; Payne 2020; Williams and Breazeal 2020) and even preschoolers (Williams et al. 2019). However, no existing learning experiences substantially support the creation of body-movement-focused interactive AI projects.

AI Learning Tools A variety of tools exist for students to interact with and design AI systems. Several extend block-based programming environments to support building projects that incorporate various AI algorithms and machine learning models. Machine Learning for Kids[1] and Cognimates (Druga 2018) add AI-related blocks to the block-based programming environment Scratch (Resnick et al. 2009), and AI Programming with eCraft2Learn (Kahn and Winters 2018) builds on the Snap! environment (Harvey et al. 2013). MIT App Inventor, which also features a block programming language, has a PoseNet v1 extension[2]. Scratch team members Eric Rosenbaum and Katya Bulovic also prototyped Scratch blocks for PoseNet v1 (Resnick et al. 2009).

Another class of AI learning tools allows students to tinker with training machine learning models of their own devising. Some of these harness transfer learning, which allows students to quickly train machine learning models on top of more sophisticated pre-trained machine learning models. This enables students to train models in less time and with fewer resources than typically required to train effective classifiers. Google's Teachable Machine 2.0 (Carney et al. 2020), and App Inventor's Personal Image and Audio Classifiers (Tang 2019) allow students and educators to train, test, and export machine learning models using the transfer learning technique. Machine Learning for Kids and Cognimates include the ability to train machine learning models using cloud machine learning service credentials and use them in Scratch projects.

Natural Interaction Natural interaction occurs when people interact with technology just as they would in everyday life: through gestures, expressions, and movements (Valli 2008). For designers of AI, creating natural interaction between a human and an agent poses a substantial challenge (Touretzky et al. 2019); "agents must be able to converse in human languages, recognize facial expressions and emotions, and draw upon knowledge of culture and social conventions to infer intentions from observed behavior"[3].

The Scratch video sensing extensions (Hwang 2012) were developed to provide real-time video interactivity to a younger audience. The Kinect is a widely used pose-tracking tool, and has been used for education in a variety of settings, including Mikumikudance (MMD), a program that allows young children to move with the Kinect to create their own 3D animations (Hsu 2011), and Kinect2Scratch[4], a tool that integrates the Kinect with Scratch 2.0 projects (Howell 2012). Scratch Nodes ML introduced a gesture interaction device that students hold while gesturing to train classification models for use in Scratch projects (Agassi et al. 2019), and AlpacaML allows students to train and test gesture recognition models on a mobile device connected to wearable sensors (Zimmermann-Niefield et al. 2019).

[1] https://machinelearningforkids.co.uk

[2] https://github.com/mit-cml/appinventor-extensions/tree/extension/posenet

[3] AI4K12 Big Ideas: https://github.com/touretzkyds/ai4k12/wiki

[4] https://stephenhowell.github.io/kinect2scratch/

PoseBlocks Toolkit

Overview

PoseBlocks are a suite of drag-and-drop coding blocks that enable students to build projects which integrate machine learning models. These blocks are made available in a special forked version of the Scratch GUI which can be accessed with a web browser. The project creation environment additionally introduces affordances for recording video clips of projects and loading sample projects from a file system.

Learning Principles

Our approach to developing curriculum supports draws heavily on the theory of constructionism (Papert et al. 1971), and the 4 P's of Creative Learning (Resnick and Robinson 2017). We were especially guided by the following principles and practices:

- **Tinkerability** is the ability of a student to "experiment with new ideas incrementally and iteratively" (Resnick and Rosenbaum 2013). Scratch itself provides many tinkerability-focused affordances, such as allowing students to trigger block behaviors from the toolbox without integrating them into a script. For PoseBlocks, block behavior was carefully designed to also enable instantaneous feedback. For example, instead of providing `nose X` and `[nose] Y` blocks, PoseBlocks gives students a `go to [nose]` block which produces an on-screen behavior with one click. This additionally fits well with Scratch's focus on Sprite behavior.
- **Project-Based Learning**. PoseBlocks are designed to support project-based learning; students can design and build projects which are aligned with their personal interests. Building on top of Scratch, students have the ability to incorporate drawings, music, animations, and a wide array of code blocks with their machine learning models.
- **Alignment with Passions**. Students are more engaged when working on projects that are meaningful to them (Resnick and Robinson 2017). PoseBlocks enables students to build and understand the types of naturally interactive AI systems they use regularly in apps like TikTok, Instagram and Snapchat, and bring their personal interests to bear on project ideation.

Technical Design Considerations

We took the following technical design considerations into account when developing the toolkit.

- **Student Data Privacy** We designed the PoseBlocks Toolkit to keep all training and test data, including camera images and audio clips, private. No audio or video data is ever sent from users to the back-end. All machine learning models are executed locally in the web browser, with the hand, body, and Teachable Machine models leveraging TensorFlow.js (Smilkov et al. 2019), and the face (Affectiva Affdex) model using the client-side web SDK (McDuff et al. 2016; Pitre 2016).

  Google's Teachable Machine, which students use for custom model training, also trains completely client-side and in-browser. It additionally allows users to publish their trained models to the web, but publishing only stores trained model weights, not any training data (Carney et al. 2020).

- **Minimal Technical Requirements** To support the wide variety of devices found in classrooms, PoseBlocks are designed to run entirely in the web browser. They have been tested to work across Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge browsers, as well as on Chromebooks. Users can take recordings of their project on Chrome, Firefox, and Edge, but not Safari, which does not yet implement the W3C MediaRecorder specification (Barnett, Casas-Sanchez, and Leithead 2017).

  A webcam and microphone can be used to train and use image, pose, and audio models. However, students who would prefer not to use their webcams, or whose CPU/GPU speed prohibits high-framerate execution, may replace the camera feed with video loops that have pre-cached face, hand, and pose model outputs. Since models are downloaded prior to use, an Internet connection capable of downloading 10-50 megabytes in a reasonable period of time is recommended.

- **Low Latency** A benefit of executing machine learning models client-side in browser is low-latency — no round trips to a web server are required to go from a change in input to result (Smilkov et al. 2019). This enables interactions like body part following and rapid reaction to changes in streams of camera, audio, and pose input.

System Design

The process for using PoseBlocks in a project is as follows:

1. Students select a model from the list of AI extensions, and the model downloads to the student's browser
2. A student clicks the drag-and-drop coding environment's "Add Blocks" button, which presents a list of AI-powered extensions
3. Upon loading the extension, the user's browser requests access to sample the user's camera feed and display it on the stage area
4. The machine learning model is downloaded asynchronously, while an icon in the block toolbox indicates whether it has loaded
5. Once the model is loaded into memory, the browser begins running new camera (or video) frames through the model, updating an internal cache of the latest frame's model results
6. As the student tinkers with the PoseBlocks, the blocks return cached results from the models; the responsiveness and execution time of blocks is not dependent on evaluating the machine learning model

The model download process differs for the Teachable Machine blocks: the model is imported dynamically by the student (see Teachable Machine Blocks section). When using pre-computed video loops, a timestamped cache of serialized model results are downloaded along with the media file. Instead of evaluating video frames in real-time, the system returns the pre-computed model output for the current time in the clip.

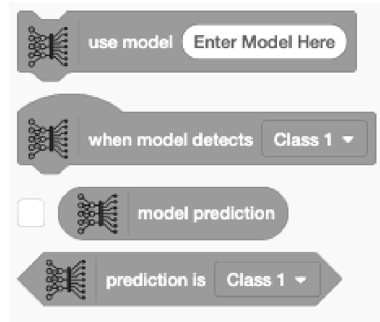

Figure 1: Blocks which allow for importing and use of Teachable Machine models in student projects

Figure 2: Blocks which respond to hand motion

Teachable Machine Blocks

The Teachable Machine PoseBlocks allow students to use Pose, Image, and Audio classification models they trained and published on Google's Teachable Machine interface in their projects. A `use model [url]` block imports the model and begins evaluating the browser's camera or microphone feed with the student's custom trained model. The blocks' class selector dropdowns automatically update to show the student's custom class names when the model is imported.

The `when model detects [class]` block is an "Event" block which triggers the attached stack of blocks underneath when the model changes its current prediction from one to another.

The `model prediction` block is a "Value" emitting block that always outputs the current prediction. Clicking this block shows what the current prediction is, and checking the box next to it continuously displays the current prediction above the project staging area. This can be helpful in debugging what the model is currently predicting.

Figure 3: Blocks which respond to facial expression, emotion recognition, and facial movement

Hand Sensing Blocks

The Hand Sensing PoseBlocks enable students to use the MediaPipe Hand tracking model (Zhang et al. 2020) in their projects. This model provides a number of finger tracking landmarks, which we expose by way of the dropdowns seen in Figure 2.

Face Sensing Blocks

The Face Sensing PoseBlocks (Figure 3) enable use of the Affectiva Affdex SDK (McDuff et al. 2016), an expression recognition toolkit, in student projects. The `when [expression] detected` block triggers when a given expression (such as a smile, eyebrow furrow, or eye closure) crosses a minimum confidence threshold to count as detected. This enables the creation of projects that respond to various facial expressions in just a few blocks.

Since the Affdex SDK provides the locations of faces, the same model evaluation is re-used to provide a `go to [face part]` block. This behaves similarly to the body and hand sensing blocks, enabling students to attach objects to parts of their face, just as is done with Snapchat and Instagram AR Lenses.

Body Sensing Blocks

The Body Sensing PoseBlocks, seen in Figure 4, provide a block-based interface for the TensorFlow.js PoseNet model (Smilkov et al. 2019). A `go to [body part]` block moves the current Scratch Sprite to overlay the corresponding body part location in the scene. The full-body PoseNet includes locations for a few facial features (nose, eye, ear), but the Affectiva Affdex model used for the face blocks provide a higher-resolution of facial feature location.

Clip Recorder

Since many projects created with PoseBlocks are interactive and can result in quite a bit of variance, we additionally provide a Record button to capture video clips of projects. We utilize the W3C MediaRecorder API (Barnett, Casas-Sanchez, and Leithead 2017) to record project clips locally in-browser, and then save the clip to the user's local computer. This allows students to use project clips as a private, local journal of their project's evolution over time.

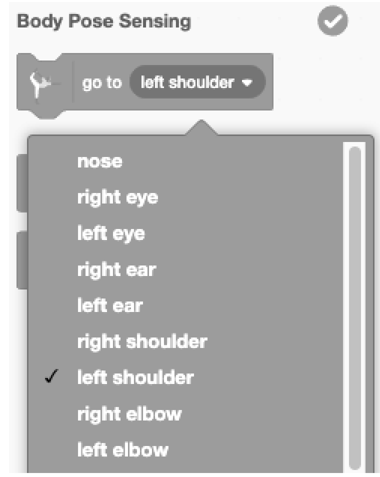

Figure 4: Blocks which respond to body part movement

Precomputed Video Loops

While great advances have been made in model optimization, continuous in-browser local model execution can be taxing on users' GPUs and CPUs. PoseBlocks's automatic model execution throttling allows students with slower computers to still build projects for real-time webcam interaction, but at a lower evaluation framerate, which can be suboptimal for tinkering and less satisfying to watch.

The toolkit's pre-computed video loops provide a solution for rapid iteration on slower computers. Unlike camera feeds, video loops always have the same frame contents, and can therefore have their model outputs pre-computed and cached. We provide a selection of sample videos (well-suited for each of the PoseBlocks tracking blocks) for which we have pre-evaluated the body, hand, and face (Affdex) outputs. Model evaluation samples are taken to build up roughly 30 model outputs per second, stored with their video timestamp, and then serialized and exported as a "video AI metadata" JSON file. On the import side, these video AI metadata files are downloaded along with the video clip, deserialized into a JavaScript object, and matched with the nearest timestamp. This allows the creation of video-reactive projects that run at a much higher framerate than possible with local evaluation.

Educational Support Materials

To support the use of PoseBlocks within machine learning curricula in classrooms, we provide a set of accompanying scaffolding materials:

PoseBlock Cards Inspired by Scratch Cards (Rusk 2019) and Bricoleur Cards (Hickey 2019), PoseBlock Cards are a series of examples of block combinations for students to try with their PoseBlocks. One card introduces, for example, how to place a `go to [thumb] [tip]` inside of a `forever` loop to cause a sprite to continuously follow your thumb. Another card introduces how to make a sprite say something in response to a facial expression using `when [smile] detected`.

Sample Projects and Activities We additionally provide students with a variety of sample projects using PoseBlocks, which include in-line comments describing how the program functions. Students can launch these projects in the PoseBlocks editor by clicking “Open Project”, or download the `.sb3` Scratch file by clicking “Download”. Each project is aligned to the different days of the workshop, and covers a different facet of building projects with AI.

Ethical Design Supports Inspired by (Payne 2020), we provide an Ethical Matrix (Figure 5) for students to use during PoseBlocks project design which allows them to analyze who their project’s stakeholders are, and what project attributes they would value.

PoseBlocks provide a unique opportunity to apply lessons from machine learning curriculum content and discussions about dataset bias to hands-on student projects. In the Dancing with AI curriculum pilot (Williams et al. 2020) which utilizes PoseBlocks, students watch and discuss Joy Buolamwini’s TED Talk “How I’m Fighting Bias in Algorithms” (Buolamwini 2016), in which Buolamwini speaks about her work in eliminating dataset bias against people of color. They then design and curate datasets for their own machine learning models with Teachable Machine, incorporate them into their PoseBlocks projects, and analyze the ethical implications of what they have built.

Tool Deployment

The PoseBlocks Toolkit was deployed as part of the ”Dancing with AI” curriculum pilot. While that curriculum’s content and learning outcomes are more fully presented in (Williams et al. 2020), we will present lessons learned and impressions of PoseBlocks in the context of this curriculum. This curriculum was piloted as a series of three synchronous remote workshops with 46 students aged 9-14. Of these, 34 had guardian consent and assented themselves to use of recordings and surveys for research purposes. The full curriculum is a week long module, consisting of 2.5 hour sessions daily from Monday to Friday. Sessions were conducted over Zoom, where instructors presented short lectures before sending the students interactive activities to complete.

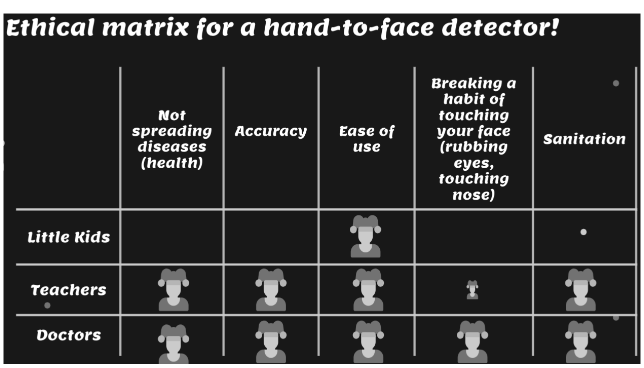

Figure 5: Ethical Matrix students may use to evaluate the stakeholders of their PoseBlocks projects

We partnered with four middle schools, as well as a summer program, to offer this curriculum to students physically located across three timezones, with 15 students attending Title 1 schools. We targeted middle school students due to their increased likelihood of having prior Scratch experience to build on, as well as their increased cognitive ability to morally reason about societal problems (Kohlberg 1981). Through the four schools we partnered with, we requested that each teacher recruit at least three non-male students to ensure a gender balance throughout each individual pilot. This resulted in a count of 18 females and 16 males of the 34 students participating in the research portion, of which 21 completed the entire week-long pilot.

Pilot Results

The use of PoseBlocks within the curriculum pilot resulted in a wide variety of student projects, each utilizing different subsets of the PoseBlocks made available to students.

Student Projects To evaluate students’ engagement with the interactive AI components, we examined the final projects students created with the PoseBlocks toolkit. Out of the 21 students who completed the week-long pilot, 13 submitted at least partial projects. Ten submitted and/or demoed their final project, and out of these, 5 used the Teachable Machine integrated blocks, 4 used the hand/face/body sensing blocks, and 1 made a non-AI related project. Applications of student projects included health and well-being (3), games (3), education (2), emotion recognition (1), and chores (1). Further analysis of the projects as they relate to the curriculum content can be found in an upcoming submission (Williams et al. 2020).

Of note, many students utilized PoseBlocks to create projects intended to positively impact their surrounding environment. One student created a project that integrated Teachable Machine’s Image Classification with Scratch’s built-in Text-to-Speech blocks to help their younger sibling learn their letters:

> “My little brother put his ABCs up and I put the letter he was struggling with up on the image and then got 100 or so samples of each. So then it was very accurate. So when my little brother put up the ABCs, I used the text to speech feature and then it said like A for apple.”

Figure 6 shows one student’s project where she has a character help her clean her room by identifying trash. She then described possible extensions to this project by introducing more input samples to the training data:

> “If you wanted to identify different things, you could add in samples to Teachable Machines of like 30 different pieces, so when you’re going through your room and trying to see if its clean, it could go into things like,

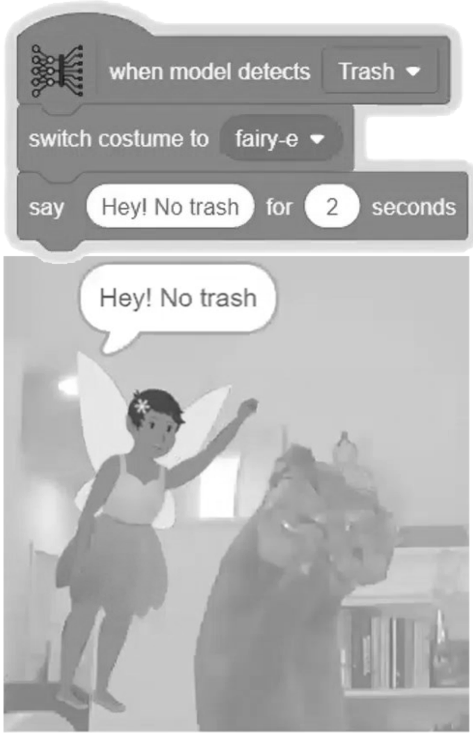

Figure 6: Student project using Teachable Machine blocks to detect trash and report it with a sprite.

‘no you still have this and this over here, you still need to take it out’. This is the most basic model.”

One student project, shown in Figure 7, was described succinctly by the student:

“If it detects you’re sad, it tries to encourage you to be happy, basically.”

Students also used PoseBlocks to create entertaining interactive experiences. The game project shown in Figure 8 uses the Hand Sensing PoseBlocks to attach a different fruit to each one of the user’s fingers. The user then brings the fruit to the astronaut to eat.

While the video playback blocks were designed to serve as a way to loop videos instead of using a live camera feed, one student surprised the researchers and figured out they could place the `play video` block within a `when [facial expression]` Face Sensing PoseBlock. Beginning with their camera on, it would wait until they smiled, and then start a video playing. Since the video would then replace their camera feed, it would begin reacting to the facial expressions in the video.

Students also created projects that would positively affect the world at large. In one project, the student trained a model to “detect if your mask was good enough”, categorizing N95 masks as the best mask, surgical and cloth masks as good masks, and bandanas and masks with filter [valves] as bad masks. He researched these recommendations on the web, and used images displayed on his phone shown to the web cam to train a Teachable Machine classifier.

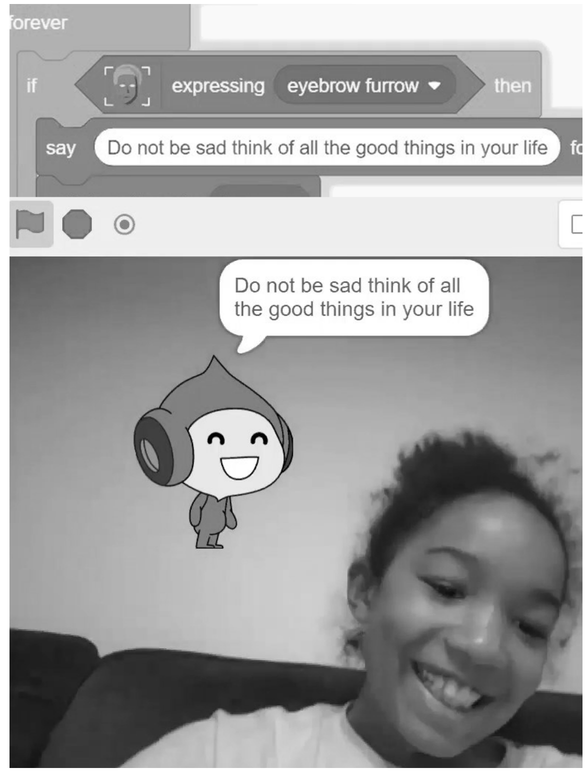

Figure 7: Student project using Face Sensing blocks to encourage user to cheer up if they are frowning.

Teacher Impressions of PoseBlocks In Pilots 1 and 3, we had the benefit of piloting alongside teachers who already knew the students we were working with. In addition to their valuable feedback and advice on pacing and remote classroom management, teachers provided daily feedback on the tools we were using.

Teachers found the project-based learning aspect especially engaging:

“I really enjoyed seeing all of the projects that they were able to come up with in a relatively short amount of time.”

“[Students] took on an entire new perspective on how AI is infused with the real world. They also had the opportunity to experience hand-on projects using JavaScript, Scratch, and various machine learning platforms and how integrating these platforms can be used to create something magical.”

Discussion

Student use of the PoseBlocks Toolkit during these three pilot workshops suggests that they were indeed able to use PoseBlocks to combine machine learning models with constructionist projects of their own devising. Student projects incorporated the entire range of developed PoseBlocks. They also learned to train machine learning models on their own, as well as take data representation and dataset curation into consideration. Additionally, students demonstrated their

Figure 8: Student project using Hand Sensing blocks to feed a sprite various food items, resulting in different soundbites.

computational thinking skills by using event-based actions, conditionals, and loops in their projects.

One method we used in the pilots was introducing a non-machine learning Scratch programming approach first, then introducing the machine learning model to the program. This was especially helpful to students who were newer to Scratch. PoseBlock Cards were also helpful to novice coders. One teacher who joined for both Pilot 1 and Pilot 3 underscored the importance of the PoseBlock Cards scaffolding, adding that they were "super easy to follow".

Teacher feedback supports existing theory that the constructionist project-based learning approach is especially effective in engaging students (Resnick and Robinson 2017). One teacher, who was also the parent of a student, related a humorous story that her son would sneak off to work on these projects even when he was not supposed to be using a screen.

Another teacher related:

> "I asked [a student] if she wanted to use some of these tools in our math class this year, and she said 'YES!'"

Our theory that students and teachers would find interactive AI and ethics relevant to their daily lives is supported by teacher feedback. At the end of the session, in response to the question "Do you think teaching artificial intelligence in the classroom is beneficial for students? Why or why not?", one teacher responded:

> "Yes... students need to be aware of how much AI impacts their daily life. They should learn these tools because it is what future jobs may require. The tools we learned about also sparked their creativity and gave them a new way to express themselves and how they think about the world, including solving problems."

Limitations

Two device-specific technical issues arose while piloting the toolkit which are worth considering for this and any web-based camera-interactive tools in the classroom context. One issue we found on many student and teacher Windows PCs is that webcams could only be used by one program at a time. The fix is to temporarily disable your camera on any video call software, close any other browser tabs which may be using the camera, and then request camera access in the Scratch editor again. This accounted for the majority of camera issues, and seems worth documenting for troubleshooting within the tool.

On iOS mobile devices such as iPads and iPhones, we found both Teachable Machine and Scratch 3.0's video sensing are unable to successfully request camera access. Additionally, Teachable Machine's model training UI does not load on iOS. We found that even with guidance to prepare each day to use laptops and not tablets/phones, students attending remote classes would often only have access to iOS devices. A workaround devised during the pilot was to have a student "drive" an instructor's computer over screenshare. A teacher remarked that this worked well to keep these students engaged.

Conclusion and Future Work

This paper discusses the design, development, and teacher/student pilot testing of a novel drag-and-drop coding toolkit for anyone to build projects that feature natural interactions with AI. PoseBlocks has the distinction of being first of its kind to support face tracking, hand tracking, facial emotion and expression recognition, and introduces the technique of providing students with pre-computed machine learning model results for videos. Through student observation and teacher feedback, we found that students were highly engaged in designing natural interactions, were able to devise and build personally meaningful projects, and were able to use these tools as a capstone to apply lessons on ethics from curriculum material on dataset bias and ethical design of AI systems.

Further work is needed to develop and assess materials utilizing PoseBlocks, and incorporate them with other student-friendly AI and robotics tools. We are excited by the promise of these tools to help students feel empowered to be active creators and informed users of AI, and even envision new career possibilities. As one student shared in their reflection at the end of the pilot:

> "I came into this class not really wanting to do a job with like computers and coding and things like that, but as this class goes on, it made me like coding and actually like working with AI more than I thought I liked, so I might end up doing something having to do with AI when I grow up."

Acknowledgements

We'd like to thank Amazon Future Engineer for funding Pilots 1 and 3 and allowing us to recruit incredible teachers through their network of Title I schools, and STEAM Ahead Boston for allowing us to pilot the toolkit during their summer program.

References

AAAI. 2018. AI for K-12” Initiative in Collaboration with the Computer Science Teachers Association (CSTA) and AI4All. URL https://aaai.org/Pressroom/Releases/release-18-0515.php.

Agassi, A.; Erel, H.; Wald, I. Y.; and Zuckerman, O. 2019. Scratch Nodes ML: A Playful System for Children to Create Gesture Recognition Classifiers. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, number Paper LBW0111 in CHI EA ’19, 1–6. New York, NY, USA: Association for Computing Machinery.

Barnett, J.; Casas-Sanchez, M.; and Leithead, T. 2017. MediaStream Recording. W3C working draft, W3C. Https://www.w3.org/TR/2017/WD-mediastream-recording-20170621/.

Buolamwini, J. 2016. How I’m fighting bias in algorithms. *TED talk* Accessed: 2020-9-8.

Carney, M.; Webster, B.; Alvarado, I.; Phillips, K.; Howell, N.; Griffith, J.; Jongejan, J.; Pitaru, A.; and Chen, A. 2020. 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*, CHI EA ’20, 1–8. New York, NY, USA: Association for Computing Machinery.

Druga, S. 2018. *Growing up with AI : Cognimates : from coding to teaching machines*. Master’s thesis, Massachusetts Institute of Technology.

Druga, S.; Williams, R.; Breazeal, C.; and Resnick, M. 2017. “ Hey Google is it OK if I eat you?” Initial Explorations in Child-Agent Interaction. In *Proceedings of the 2017 Conference on Interaction Design and Children*, 595–600. dl.acm.org.

Harvey, B.; Garcia, D. D.; Barnes, T.; Titterton, N.; Armendariz, D.; Segars, L.; Lemon, E.; Morris, S.; and Paley, J. 2013. Snap!(build your own blocks). In *Proceeding of the 44th ACM technical symposium on Computer science education*, 759–759. dl.acm.org.

Hickey, S. M. 2019. *Bricoleur : creative learning through video and computation*. Ph.D. thesis, Massachusetts Institute of Technology.

Howell, S. 2012. Kinect2Scratch (Version 2.5)[Computer Software]. 2012. https://stephenhowell.github.io/kinect2scratch/.

Hsu, H.-M. J. 2011. The potential of kinect in education. *International Journal of Information and Education Technology* 1(5): 365.

Hwang, T.-H. T. 2012. *Exploring real-time video interactivity with Scratch*. Ph.D. thesis, Massachusetts Institute of Technology.

Kahn, K. M.; and Winters, N. 2018. AI programming by children. *Constructionism* 5: 315–325.

Karaman, S.; Anders, A.; Boulet, M.; Connor, J.; Gregson, K.; Guerra, W.; Guldner, O.; Mohamoud, M.; Plancher, B.; Shin, R.; and Vivilecchia, J. 2017. Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT. In *2017 IEEE Integrated STEM Education Conference (ISEC)*, 195–203. ieeexplore.ieee.org.

Kohlberg, L. 1981. *The Philosophy of Moral Development: Moral Stages and the Idea of Justice (Essays on Moral Development, Volume 1)*. Harper and Row.

Lin, P.; Van Brummelen, J.; Lukin, G.; Williams, R.; and Breazeal, C. 2020. Zhorai: Designing a Conversational Agent for Children to Explore Machine Learning Concepts. *AAAI* 34(09): 13381–13388.

Long, D.; and Magerko, B. 2020. What is AI Literacy? Competencies and Design Considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI ’20, 1–16. New York, NY, USA: Association for Computing Machinery.

McDuff, D.; Mahmoud, A.; Mavadati, M.; Amr, M.; Turcot, J.; and Kaliouby, R. e. 2016. AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’16, 3723–3726. New York, NY, USA: Association for Computing Machinery.

McReynolds, E.; Hubbard, S.; Lau, T.; Saraf, A.; Cakmak, M.; and Roesner, F. 2017. Toys that Listen: A Study of Parents, Children, and Internet-Connected Toys. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI ’17, 5197–5207. New York, NY, USA: Association for Computing Machinery.

Papert, S.; Solomon, C.; Soloway, E.; and Spohrer, J. C. 1971. Twenty things to do with a computer. In *Studying the Novice Programmer*, 3–28. Lawrence Erlbaum Associates, Inc.

Payne, B. H. 2020. *Can my algorithm be my opinion?: An AI + Ethics Curriculum for Middle School Students*. Master’s thesis, Massachusetts Institute of Technology.

Pitre, B. 2016. Emotions In Your Web Browser. https://blog.affectiva.com/emotions-in-your-web-browser. Accessed: 2020-09-08.

Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; and Kafai, Y. 2009. Scratch: programming for all. *Commun. ACM* 52(11): 60–67.

Resnick, M.; and Robinson, K. 2017. *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press.

Resnick, M.; and Rosenbaum, E. 2013. Designing for tinkerability. *Design, make, play: Growing the next generation of STEM innovators* 163–181.

Rusk, N. 2019. *Scratch 3.0 Coding Cards*. No Starch Press, Incorporated.

Smilkov, D.; Thorat, N.; Assogba, Y.; Yuan, A.; Kreeger, N.; Yu, P.; Zhang, K.; Cai, S.; Nielsen, E.; Soergel, D.; Bileschi, S.; Terry, M.; Nicholson, C.; Gupta, S. N.; Sirajuddin, S.; Sculley, D.; Monga, R.; Corrado, G.; Viégas, F. B.; and Wattenberg, M. 2019. TensorFlow.js: Machine Learning for the Web and Beyond .

Tang, D. 2019. *Empowering novices to understand and use machine learning with personalized image classification models, intuitive analysis tools, and MIT App Inventor*. Master's thesis, Massachusetts Institute of Technology.

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 AAAI Conference on Artificial Intelligence*, volume 33, 9795–9799. aaai.org.

Valli, A. 2008. The design of natural interaction. *Multimed. Tools Appl.* 38(3): 295–305.

Van Brummelen, J. 2019. Conversational Agents to Democratize Artificial Intelligence. In *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 239–240. ieeexplore.ieee.org.

Williams, R.; Ali, S.; Devasia, N.; DiPaola, D.; Hong, J.; Jordan, B.; and Kaputsos, S. 2020. Deploying project-based, AI curricula in virtual, middle school classrooms. *"Int. J. AI Ed."* "Manuscript in progress".

Williams, R.; and Breazeal, C. 2020. How to Train Your Robot: A Middle School AI and Ethics Curriculum. *IJCAI* .

Williams, R.; Park, H. W.; and Breazeal, C. 2019. A is for Artificial Intelligence: The Impact of Artificial Intelligence Activities on Young Children's Perceptions of Robots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, number Paper 447 in CHI '19, 1–11. New York, NY, USA: Association for Computing Machinery.

Williams, R.; Park, H. W.; Oh, L.; and Breazeal, C. 2019. PopBots: Designing an Artificial Intelligence Curriculum for Early Childhood Education. *AAAI* 33: 9729–9736.

Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.-L.; and Grundmann, M. 2020. MediaPipe Hands: On-device Real-time Hand Tracking .

Zimmermann-Niefield, A.; Turner, M.; Murphy, B.; Kane, S. K.; and Shapiro, R. B. 2019. Youth Learning Machine Learning through Building Models of Athletic Moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, 121–132. New York, NY, USA: Association for Computing Machinery.