Exploring Machine Teaching with Children

Utkarsh Dwivedi, Jaina Gandhi, Raj Parikh, Merijke Coenraad, Elizabeth Bonsignore, and Hernisa Kacorri
College of Information Studies, University of Maryland, College Park {udwivedi, ebonsign, hernisa}@umd.edu

***Abstract*—Iteratively building and testing machine learning models can help children develop creativity, flexibility, and comfort with machine learning and artificial intelligence. We explore how children use machine teaching interfaces with a team of 14 children (aged 7-13 years) and adult co-designers. Children trained image classifiers and tested each other's models for robustness. Our study illuminates how children reason about ML concepts, offering these insights for designing machine teaching experiences for children: (i) ML metrics (*e.g.* confidence scores) should be visible for experimentation; (ii) ML activities should enable children to exchange models for promoting reflection and pattern recognition; and (iii) the interface should allow quick data inspection (*e.g.* images vs. gestures).**

***Index Terms*—child-computer interaction, machine learning, machine teaching, informal learning, AI education**

I. Introduction

Consider the problem of classifying data as positive or negative based on a threshold. In this context, Zhu *et al.* [1] define *machine teaching* as a method or algorithm that involves a teacher who knows the true threshold for separating positive and negative data and designs an optimal training set for the learner to learn to classify. In this work, we invite children to be the teachers and a machine learning algorithm to be the learner. We explore machine teaching with children using Google Teachable Machines [2], "*an experiment that makes it easier for anyone to start exploring how machine learning works, live in the browser.*" Children are called to design training sets of images to teach the underlying model, which leverages neural networks, how to classify their images.

Why explore machine teaching with children? Machine teaching can be a great vehicle for exposing children early on to machine learning and AI concepts. This work is aligned with recent government declarations (*e.g.* [3, 4, 5, 6]) and initiatives calling for an AI curriculum as early as the first five years of schooling [7, 8]. Similar to us, researchers and educators have early on seen the opportunity to expose children to AI and machine learning black boxes. When categorizing prior efforts, we see two main threads: (a) those that require some programming and (b) those that do not. In the first thread we see the use of block-based visual programming languages such as Scratch [9, 10] through worksheets [11], robotics camps [12, 13], and interactive systems [14, 15]. Such approaches assume prior exposure to block-based programming, which may not apply to younger children or those who do not have access to early computer science (CS) education. For example, in the US, only 47% of schools teach CS, with disparities across ethnicity, race, gender, disability, and socioeconomic status [8]. Perhaps this can explain why learning objectives for AI education in K12 [7] highlight the use of interactive systems before Scratch-based ones [16]. We see such efforts in the second thread, where children train and test AI black boxes through *e.g.* interactive spreadsheets [17] or *e.g.* accelerometer-based gesture recognizers [18, 19]. In this paper, we present findings from one of the earliest efforts that fall under this second thread, focusing on a more diverse group of children from a younger age group.

The overarching goal of this work is to inform the design of teachable machines, a paradigm of machine teaching, for introducing children to machine learning concepts. As a first step in designing such applications, we investigate the following research question: *What are the key behaviors characterizing children's interactions with a teachable interface?*

We explore this question through co-design with a university-based intergenerational design team of 14 children and 12 adult co-designers. As shown in Figure 1, during circle time children answer a warm-up question about teaching others and watch videos introducing them to Google's Teachable Machine [2]. For the design activity, they are split into pairs, where they design their teaching and testing sets for the classifier and swap classifiers with each other to see whether their models generalize. Children then demo their final classifier during a whole group presentation, reflecting on challenges they faced and workarounds they attempted.

We found that metrics such as confidence scores tend to serve as proxy for children to judge whether the model was confused on unstable. Also, inviting children to swap and test their classifiers elicits collaborative observations and reflections and promotes experimentation. Last, having classification tasks such as image recognition can enable children to quickly inspect the data and uncover patterns, though, they assume that children are sighted. More research is needed on making such activities accessible for children with visual impairments.

These findings contribute to our understanding of how children interact with machine teaching interfaces. Specifically, we offer the following insights to designers and educators for the design of AI-related learning experiences and interactive systems to support them: (1) making metrics such as confidence scores visible and dynamic can enable experimentation, reasoning, and discussion; (2) enabling pair activities where children can compare training sets and strategies with others can support reasoning and recognition of patterns for improving models; and (3) employing modalities accessible to children (*e.g.*, images rather than gestures for sighted children) in machine teaching activities can promote pattern recognition and enable quick data inspection and training adjustment.

978-1-6654-4592-4/21/$31.00 ©2021 IEEE

Fig. 1. Our sessions include (a) circle time, (b) design activities, and (c) final presentation with reflections. A screenshot from one of the children's classifiers (d), indicates what is captured by the camera, the number of training examples and images, and the confidence bar for the triggered class and its output.

II. Related Work

We discuss recent efforts in AI education for children with a focus on studies that employ machine teaching. Prior work involving adults is briefly mentioned as it informs our analysis.

A. Adult Non-experts Training Machine Learning Models

There is a rich literature on adult non-experts[1] and interactive machine learning. We look at efforts that, similarly to this work, employ machine teaching [1, 20]; where non-experts train and evaluate supervised classification models using interfaces that abstract the complexities of the algorithm as the children do in our study. The arguments for such applications are many. For example, Amershi *et al.* [21] argues that involving people more actively in the training process can lead to higher acceptance of AI and more robust models. Kacorri [22] also underlines the potential of teachable interfaces for accessibility, where training data are sparse and highly variable. We see assistive applications such as teachable sound and object recognizers [23, 24] falling under this paradigm.

Given that machine teaching "focuses on the efficacy of the teachers" [20], early work in this field has put an emphasis on understanding the underlying concepts that non-experts are able to grasp as well as misconceptions and other pitfalls that they may be susceptible to [25]. Our paper shares this goal. Looking at adults, prior work has shown that non-experts tend to teach with clear representative examples and sometimes incorporate examples that are closer to the decision boundary through variation [26]. Some seem to grasp the concept of overfitting [27] and there is anecdotal evidence that they learn to balance class proportions in training after multiple iterations [28]. However, they also tend to be more satisfied and trusting toward their models compared to experts [29]. Beyond class imbalance, they are susceptible to disparate treatments such as being inconsistent in the way they introduce variation [26]. Common misconceptions relate to accuracy being a sole measure of performance [29], consistency entailing teaching over and over with the same example [26], and the machine possessing reasoning capabilities [26]. Such misconceptions and pitfalls led to problematic deployments.

[1]We refer to those not formally trained in machine learning as non-experts.

B. Children Training Machine Learning Models

Looking at recent work in AI for K-12, we see many efforts requiring familiarity with Excel [30], Rapidminer [31] or block-based programming [11, 12, 13, 14, 32]. In contrast, our paper focused on efforts that employed machine teaching[2] and did not make assumptions about children's familiarity with programming.

Table I presents representative examples of studies from 2018-2021 with classification tasks involving either multiclass gesture recognition [33, 34, 35, 36] or, as in our study, multiclass image recognition [37, 38]. Surprisingly, only four reported the number of children involved and their gender distribution, which is skewed towards boys except for Vartiainen [38] which has equal distribution. Being committed to broadening participation in computing, in our study, we tried to balance the number of boys and girls and report demographic data on children's race and ethnicity. Similar to our study, children's ages in these prior efforts ranged from 8 to 14 years old; Agassi *et al.* [35] did not report children's age, and Scheidt *et al.* [37] only report the age group of invited children, not necessarily the ones who actually experienced their system. Typically, they involve middle and high school children at least 10 years old; exceptions being Cognimates [14] with a younger age group 7-11 years old and Vartiainen [38] with 6-12 years old. Limited methodological information is included in these two publications, perhaps due to their limited page length. For the other studies, researchers employ both quantitative methods such as pretest post-test [33] or within-subject [18] designs and qualitative methods such as design experiments through workshops and semi-structured focus groups [34]. Similar to Vartiainen [38], our study employs a participatory design approach known as Cooperative Inquiry (CI) [39, 40]. In CI (also, "co-design"), children and adults act as full partners, assuming various roles throughout the design process [41, 42, 43]. For example, children can act as end-users testing prototypes, evaluate low-fidelity mockups, or have an equal voice in sharing ideas with adult co-designers [41, 42]. In contrast to related work, we used pair-testing of the trained model, where children test each other's models and investigate its machine learning properties of generalizability.

When looking at the underlying machine learning models that children interacted within these studies, two of them, Scheidt *et al.* [37] and Vartiainen *et al.* [38], included neural

[2]A term perhaps not originally used by the authors in the publications.

	Characteristics	[33]	[18]	[35]	[36]	[37]	[38]	Ours
Child	ages (years)	10-12	10-13	n/a	8-14	~ 6-12	3-9	7-13
	gender	9b	20b, 10g	n/a	4b, 2g	n/a	3b, 3g	8b, 6g
Study	pretest posttest design	•						
	within-subjects design		•					
	design experiment				•			
	co-design						•	•
	not available			•		•		
Model	Wizard of Oz	•						
	dynamic time warping		•	•	•			
	neural networks					•	•	•
Input	images					•	•	•
	gesture	•	•	•	•			

TABLE I

CHARACTERISTICS OF STUDIES THAT EXPLORE TEACHABLE MACHINES WITH CHILDREN JUXTAPOSED WITH OUR WORK.

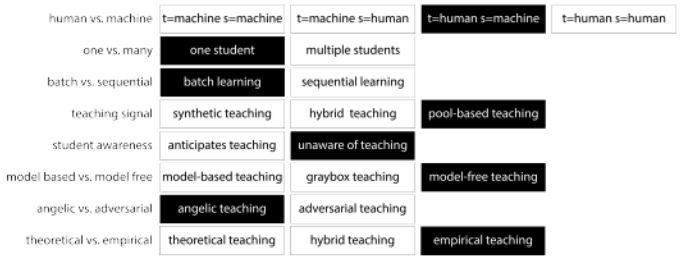

Fig. 2. Characterizing GTeach in the machine teaching space by Zhu *et al.* [1], where t stands for teacher and s for student.

networks. Others opted for Wizard of Oz [33] or dynamic time warping algorithms [18, 34, 35, 36]. When interacting with these algorithms, children only saw the top prediction. In contrast, in our study children are exposed simultaneously to the top prediction, and the confidence scores across the classes, which allow them to gain more insights should their model fail.

III. METHODS: CO-DESIGN

In our study, we work with youth 7-13 years old who are members of an intergenerational co-design team [39, 40]. We aimed to explore how youth with no prior programming experience (as in [33, 34]) might consider teaching a machine to recognize object and image classes that they themselves designed from everyday low-fidelity prototyping materials (*e.g.*, colored paper, popsicle sticks). Informed by prior work, our exploration process prioritizes recent guidelines for supporting K-12 students [7] and the ISTE standards [44][3] more generally with a focus on children who are 7-13 years old, to demonstrate how the iterative nature of teachable machines can promote children's understanding of AI. In our study, child and adult co-designers engaged with an existing teachable interface provided by Google abbreviated as GTeach [2]. Children's design session goal was to compose input images (*e.g.*, origami shapes) and explore issues they might encounter while training a well-performing image classifier. They "designed" the classes for the teachable image classifier to recognize and experimented with various training examples to present to the GTeach interface. They devised their input classes from a paper-based prototyping kit that included a variety of shapes and colors of origami, which they could personalize with stickers and markers. Children were invited to select types of shapes and colors that they thought would be easy to teach or that they would want to teach, and also, how they would teach them (*e.g.*, how many training images, what sort of background, how close to/how far from the camera, and more).

A. Machine Teaching Testbed: GTeach

GTeach [2] is a popular demo exposing how machine learning works to non-experts. We used its version 1.0 that allows people to quickly train an image classifier by using a webcam. As shown in Figure 1d, the input can fall under one of three classes (green, purple, and orange) that the user can train. The output can be either GIFs, sounds, or spoken text. For each class, the interface displays the total number of training examples, thumbnails of nine last examples, and the classification confidence when recognizing a video frame. The user can test the recognition performance of their classifiers interactively. The underlying recognition model is the SqueezeNet [45], a neural network architecture small enough to run locally on the browser. In this case, SqueezeNet was pretrained on ImageNet [46] to recognize 1,000 different classes (such as animals, plants, and everyday objects), thus developing internal representations for recognizing color, edges, and shapes in images. Through transfer learning [47], these internal representations are used to quickly learn how to recognize a new class that the network has not seen before just by providing a few training examples. Basically, SqueezeNet provides an embedding vector for every training image, a numerical representation that serves as a descriptor for that image. The underlying assumption here is that similar images also have similar embedding vectors. Thus, to recognize a new image, the teachable machine simply compares the embedding vector of the new image with the embedding vectors of the previous training examples to see which class is the closest.

We adopt Zhu *et al.* [1] machine teaching problem space to characterize GTeach as a system where the child is the *teacher* and machine is the *only student* (Figure 2). A child provides *batches* of images that are *pooled labels* as the teaching signal. The model (neural network) *does not anticipate* this signal, *i.e.* assumes training examples are error-free, independent, and identically distributed. The child takes a *model free* approach, treating the model as a black box, and is considered a friend, i.e., *no adversarial training*. We assume that the child uses *empirical teaching* methods to improve model performance.

B. Design Session and Participants

Our study comprised two iterations, conducted with two different groups of children: one in Feb 2018 (8 children); the other in Oct 2019 (6 children). In total, 14 children aged 7-13 years old and 12 adult co-designers participated in the design session (6 girls, 8 boys) over two years. In the first session, girls and boys were evenly numbered (3 girls, 3 boys); in the second, 4 boys and 2 girls participated. Many children (64%, or 9 of 14) are second generation immigrants to the

[3]Specifically: 3d) Building solutions for real-world problems, 4c) Design, test and redesign solutions, and 5b) where children analyze data to look for similarities and patterns

Pseudonym	Gender	Age	Race/Ethnicity	S1	S2
Gina	F	8	Black/African American	•	
Jeremy	M	8	Black/African American	•	
John	M	8	Black/African American	•	
Matt	M	10	Asian American	•	
Amber	F	11	Hispanic / Latina	•	
Caleb	M	11	Black/African American	•	
Rina	F	11	White / Caucasian	•	
Sandy	F	11	Asian/Asian American	•	
Ben	M	7	Black/African American		•
Brian	M	8	Black/African American		•
Penny	F	11	Black/African American		•
Alan	M	11	Black/African American		•
Denny	F	11	Black/African American		•
Kevin	M	13	Black/African American		•

TABLE II
DISTRIBUTION OF CHILDREN ACROSS OUR CO-DESIGN SESSIONS.

USA. Children are recruited through word of mouth, based on family interest. All participants and activities are approved by the university's IRB. We obtain signed parental consent and child assent, including consent for audio/video recording. All personally identifiable data is removed to protect the children's anonymity. Of the adult co-designers, 3 had a machine learning background, and others had an education background. Children's pseudonyms, gender, age, race/ethnicity, and the session they participated in are shown in Table II.

The specific CI technique we employed is *technology immersion* [48, 49], which exposes children to novel technologies with little or no experience to raise their awareness of the design potential of those technologies [48]. Children mainly assumed the role of informants and evaluators, providing feedback on their observations [48, 50]. Children also saw themselves as designers of the paper-based models they used to train and collaboratively test the GTeach classifier. The adults assumed facilitating roles [43] as the children designed their classifier models, scaffolding them with guiding questions as needed. A session had three parts:

Circle Time: As a warm-up, all co-designers (children and adults) answered the question *what's one thing you've taught someone else?* The question provided context for the design session and enabled the team to discuss challenges and successes related to people teaching people. For example, Denny shared that she taught her young "*baby cousin to say the word, 'eat'*". When asked how many times she had to model or repeat saying the word, 'eat', Denny replied, '*a LOT.*' Denny's every day teaching observation afforded adult co-designers opportunities later in the session to connect similar familiar experiences to key components of machine teaching. The team then watched videos about GTeach [51] and a fun example to pique their creativity [52].

Train, Test, Deploy: During the main co-design activity, children selected origami shapes they wished to train and worked in pairs to train GTeach. After each child had trained their model with specific shapes and/or colors, they switched with their partner, to test one another's model.

Presentation: At the end, each child presented their classifier to the group. Children elaborated upon their training strategies and explained how to tackle problems they faced and possible workarounds to the problems shared by others. Salient themes from the children's descriptions and discussion were captured as "Big Ideas" by an adult on a whiteboard.

C. Data Collection and Analysis

We collected videos through screen recordings and cameras, photos, and field notes. Specifically, screen (and mic) recordings captured children's interactions with the GTeach, including the number of training examples, confidence scores, and the video from the webcam. Thus, they also captured the use of props and children's interactions with their pairs and adult co-designers. Static cameras also captured group interactions in circle time and final presentations. Screen recordings from 5 children were not saved properly; we rely on complementary data when available *e.g.* screen recordings of their pair, static camera, photos, and field notes. Follow up questions included: (1) *How many examples did you have to give it before it learned what to do?* (2) *What did you find challenging? Was it hard to teach the computer? Why or why not?* (3) *Why does it make mistakes? If it made mistakes, how did you make it work better? What trick did you do to stop it from making mistakes?*, and (4) *Does it work all the time? Why not or why?*

After the session ended, 5.5 hours of videos, 2 sets of in-situ, "Big Idea" design themes (*i.e.*, photos of whiteboard themes) and 2 session notes from researchers were analyzed and coded using thematic analysis [53]. While open coding was used for inducing codes, some codes were constructed beforehand related to machine learning and teaching for deductive analysis. The a priori codes were: sample sizes used for training each class, presence of balanced and unbalanced classes, errors, and children's responses to errors. First, we coded the screen recording from each child's computer in the *train, test, deploy* followed by corresponding videos in the *circle time* and *presentation*. Codes per child were merged and tagged with time. Then codes were merged across all children; quotes and video activities were grouped within similar codes, and they were merged again based on evidence. Thematic analysis helped us decouple codes by actions, quotes, and text in the notes and whiteboard, so patterns reflected in quotes could be corroborated with actions if quotes were absent.

D. Study Design Limitations

While our co-design study was one of the first efforts that explores machine teaching with children (early 2018), due to its subjective nature, it is not conclusive. It provides a rich set of observations and insights, generating many hypotheses that need to be further investigated, *e.g.* through mixed methods.

We made a conscious effort to balance gender and focus on underrepresented communities in computing in our study. However, we are aware that our participatory design team of children may not be representative of children with similar demographics. Our child participants had prior experience in evaluating novel technology and training in design thinking. This experience helps them articulate their thoughts better and

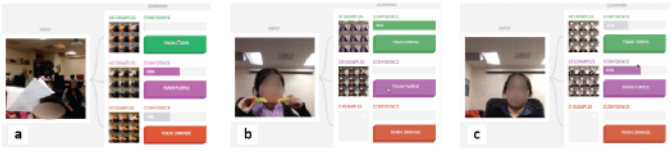

Fig. 3. Children's interpretation of the confidence score, (a) Amber found that the classifier would trigger the wrong class and fluctuate between classes without reaching 100%, (b) Sandy interpreted the classifier as confused as it showed a confidence score of 100% for the wrong class, and (c) she used the score to probe the classifier, by moving towards and away from the camera.

be more aware of their role in the design process. So, they have more experience with technology (not machine learning), which does not make them an ideal representative of children.

IV. Results

In this section, we consider how children approached the process of training their teachable machines. We present the children's quotes and corresponding screenshots of their model as evidence to support our findings. The quotes specify child and classifier actions or responses italicized and encapsulated within asterisks, *e.g.*, {*she wears her glasses*} or *e.g.*, {*the classifier triggers the purple class*}, and the rest of the quote is the statement made, where square brackets denote any addition to complete missing words, *e.g.*, [the].

A. Interpreting the Confidence Score

Metrics like accuracy, precision-recall, F1-scores are typically used to evaluate models [54], and experts have a clear understanding of how to interpret and use them in the context of the training pipeline. Amershi *et al.* [21] showed how adult non-experts could use metrics such as confidence scores[4] and confusion matrices[5] to assess and improve the model's quality. However, prior work with children (except for Vartiainen [38] in Table I) has not presented confidence scores along with the prediction; instead, only the prediction is shown. In contrast, the GTeach interface dynamically displays its three outputs with a confidence score for each. It highlights the selected output, making visible AI decision-making mechanisms opaque in the "black-box" approaches of prior studies. This feature helped surface the children's efforts to understand as well. We could observe how they interpreted confidence scores and the model's output decisions and how they used these metrics to improve and test their training approaches.

Children interpreted confidence scores in three ways: (1) if the classifier predicted a correct class with a 100% and remained stable, they saw it as a success; (2) if the classifier shifted between classes rapidly, triggering different outputs, they interpreted it to be confused; and (3) if the classifier remained 100% on a wrong class, they viewed it as a mistake or that something had gone wrong. Since the maximum score for a class would trigger the corresponding output, children viewed that as a cue that the classifier had selected a class. The following examples show how children interpreted (or misinterpreted) confidence scores, enabling us to gain insight into the potential for such metrics to scaffold exploration, experimentation and promote understanding of AI concepts. For John, Sandy, and Amber, the confidence level served as the primary threshold they used in training: they trained their inputs until they registered a confidence level of 100%. When Amber demonstrated the classifier to the whole group, she saw that the classifier was not performing well (see Figure 3a). When training her classifier, she had included her head in the frame; however, when testing it, the camera only captured a small portion of her head, resulting in a correct classification but a low confidence score. When asked about the disparity, she observed, "It reaches 60 the more I put my head up".

Sandy's experience offers another example demonstrating how in-situ metrics help children dynamically attend to and apprehend the classification process. As she tested, she noticed her classifier predicted the right class at less than 100%, often jumping to other classes during training. She wondered aloud, "It's not 100%, should I do this again then?" (see Figure 3b).

> **Adult**: Maybe it recognizes your glasses and my scarf as other objects. [The] top one is me, oh, {*the classifier fluctuates*} because my scarf is off it's not sure which one's which, {*Adult wears her scarf back*} now it's really sure, and let's do glasses {*Adult wears Sandy's glasses*} Oh {*the green class is triggered when she wears the glasses*}
> **Sandy**: {*she wears her glasses*} now it's really sure {*the purple class is triggered*} let's see about this {*she wears the adult's scarf*}, well it's clearly focusing on the glasses and the scarf. It is so confused. Maybe it takes one of the objects to recognize.

During the presentation, after Sandy reflected on her observations, she reasoned how the classifier might be working, *"I think it chooses one inanimate object for it to focus on."*

Children actively employed the confidence score to track how well their classifiers performed and varied according to variations in their training models. Often they would test even if they had only trained two out of three classes. Then they would attempt to "fix" the low confidence scores with more samples or switch to a completely different set of objects. When children themselves or their child-partner tested their classifier, they wrestled with the errors and adjusted accordingly. Similar to adult non-experts, who aimed to get to accuracy that "looked good" [29], children rested training decisions on the prediction metric until deployment, where partners devised examples confusing the classifier, or they noticed that a new background affecting prediction and confidence.

B. Varying the Size of the Training Set

In this subsection, we examine how children fix and iterate on their models by adding or removing training data. We compare their reasoning with findings from Yang *et al.* [29], who provided adult non-experts the choice of adding or splitting datasets when experimenting with the models. They observed that in contrast to experts, non-experts tend to choose the maximum size of the dataset that can be used to train [29] but not the data quality or class balance. To compare children's

[4]A score that denotes the uncertainty a model has on any given output.
[5]A matrix that shows the distribution of predicted versus the correct classes.

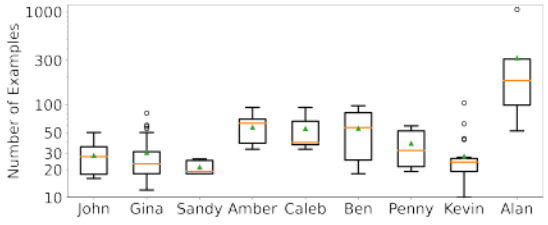

Fig. 4. Box-plot distribution of training image for all the classes each time the children added or reset a training class. Alan went upto 1000 to test how far the classifier goes and Kevin mostly tested a set of 10-12 images for all three classes. Other children stayed withing the 15-100 range of images.

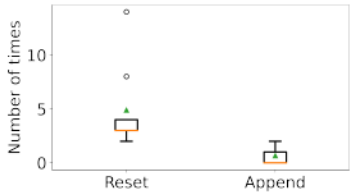

Fig. 5. Box-plot distribution of times children employed a reset versus append strategy. All children reset at least two times with a median of three. Only four children choose to append.

efforts with adults, we noted the number of images that children collected for each class, whether they reset the class (start with a new set of images) or append to the existing examples. We calculate whether the dataset is imbalanced (if it has too few of a certain class).

The number of training examples varied across children, which collected on average 61 (min-max=10-1058, sd=130) examples per class (see Figure 4). John, Ben, and Alan specifically mentioned they needed to train a class with 4, 100, and 300 examples, respectively. When asked during the presentation about the effect of the training size, Sandy commented that she didn't focus on the number of examples while Gina answered, "*Yeah, I think it did, a little bit.*"

In contrast to adult non-experts [29], providing more examples was not a primary strategy among children for overcoming prediction errors. John and Sandy only employed this strategy after having accidentally provided a few negative examples. John saw the problem and tried fixing it by adding multiple positive examples to the class showing fluctuations. In contrast, Sandy, who saw that her classifier was still predicting the correct class after her mistake, was unsure if there was a problem. When asked if the wrong images had any effect, she replied,

> **Sandy**: Probably, actually, I don't know, {*moves towards and away from the camera*} I don't think it did, I don't think it did anything. Because it doesn't seem [that] anything changed by the way it [the confidence] goes on and up.

However, perhaps because she was prompted, she added a few more positive examples. But then, she decided to drop the whole training set and start from scratch.

Children who reset a class, *i.e.* discarding all previous examples, tend to retrain with a similar number of images. As shown in Figure 5, we find that the children more commonly employed the resetting strategy; all children reset at least once, but only four appended. When looking at the number of times children reset, we see a median of 3, with Gina and Kevin being outliers with 14 and 8 times, respectively.

Class imbalance, a phenomenon explored with adult non-experts [28], is hardly observed among children. We examine the presence of skewed class proportions adopting a definition of mild, moderate, and extreme imbalance with ratios 1:100 or worse between any two classes indicating extreme imbalance [55]. We found that none of the classifiers trained by the children had an extreme imbalance. Only 9 had moderate, and 3 had a mild imbalance. The majority of the classifiers (51 out of 63) were balanced. The most skewed class proportions, observed in one of Alan's classifiers, had ratios of 309:89:1058.

C. Composing Examples with Variation

Diversity plays an important role in machine learning [56]. Prior work on teachable object recognizers showed that adult non-experts draw from parallels to how humans recognize objects independent of size, viewpoint, location, and illumination to incorporate diversity in their training examples [26]. Similarly, we see children composing examples that incorporate variations in terms of perspective, size, color, and backgrounds when training and testing their own and their partners' classifiers. Similar to adults in [26], they would first test with examples that were similar to the training set. But then, they would explore the boundaries by confusing it with high variation examples *e.g.* incorporating new faces, hands, or different origami in the background. These investigations often led to comments and experiments indicating that children had noticed how objects can become noise if they share similarities with the other classes of objects, how backgrounds impact performance, and how the placement of the objects in the camera frame can impact the accuracy of the classifier.

For instance, when Alan tried out Ben's classifier, which was trained with various origami and Ben's face as he moved around in the frame, he found that the classifier is confused. He tried out different positions of the origami with respect to the frame and varied the distance from the camera (see Figure 6a)..

> **Alan**: Ooh, it's recognizing if it's [the flower origami] on my head or not look {*he puts flower up close the camera*} whenever it comes to my head. {*He puts the flower on his head, and removes it, sees the confidence fluctuate without it. It only triggers green class when it is on his forehead.*} It malfunctions there, but when I put my face in {*it works as the classifier triggered green class correctly*}.

In a similar exploration, Sandy composed examples that used her face and an origami. She ensured that her training and testing examples were similar by bending down while training to keep her face out of the frame. While testing the classifier, she argued that the classifier predicted the class for her face because the images had a high resemblance to her face leading to this comment (see Figure 6b).

Fig. 6. Children tried out different configurations and orientations of origami and faces, (a) Alan brings the flower origami towards and away from the camera and observed fluctuations, (b) an adult co-designer tested Sandy's classifer with her face, and (c) Ben tested out Penny's classifier, and finds that it works even when it's not trained with his face.

Fig. 7. Children revealed and introduced different forms of noise to their classifiers, (a) Penny found her classifier does not work on all hands and faces, (b) John built a color classifier because he found that it would not work on slightly different positions and new faces, and (c) Caleb added wave-like motions to the classes because he thought that would work better

Sandy: I have a question, isn't, aren't we already a face aren't we already a shape? Let's see if it recognizes her [adult co-designer].
Adult: {*Adult shows the exact "model like" face, face as Sandy called it, with the origami on top of her head*}
Sandy: I think it is the shape [origami] I think that it should only recognize one face and so someone else can set their own face.

In another example, Ben used Penny's classifier with origami that had been trained for but with Penny's face and shirt in the background. The classifier gets confused, switching erratically between classes. So, Ben tries yellow and ghost-shaped origami that is not the same color as trained on (see Figure 6c). He's asked why this could be happening,

Ben: They are both the same shapes that's why {*He is referring to the purple and orange classes*}
Adult: Okay, because they are both the same shape. Now, why is it getting these two confused?
Ben: Because they are both yellow. {*He is referring to the green and orange classes shown.*}

When children swapped their classifiers with child- or adult-partners, they would use the same origami, but the classifiers would not work right. This is how children would find that the classifier they trained with faces was not generalizable to other faces or similar origami; the color would cause confusion. They would be asked to train a classifier without their faces to make it possible for others to use the classifier.

D. Reasoning About Noise in the Classifier

Noise in children's data can be due to wrong labeling or the corruption of the data features [57]. Feature noise can be contextual; *e.g.*, low light, partially object, and objects with their discriminatory portions not adequately captured [22, 26]. We discuss the kind of noise the children found during testing. For instance, Kevin found that his classifier was confused even when he added more examples. He reset the erroneous training samples 8 times to try to fix this issue. His new examples had his blue coat and a blue origami in the images, which were driving the predictions. When it is pointed out to him, he says, *"It is? Oh because the ghost is also blue."* He added new examples by bending out of the frame, constituting more of the background, leading to noise as other examples remained same. When asked why the system is confused, he answered,

Kevin: That's because of the color, because of the light, because it's the main, the light has all the colors of the rainbow, right? The light you know messes with it.
Adult: So you think the color and the background messes with it. How do you think we can improve that? [by] having a black background?
Kevin: I feel like we should cancel out the light like the app cancels the light. Any white light gets rid off. It will still see with the camera, but it cancels out the white light.

Kevin gave the wrong reason even though he saw the color of his coat trigger his classifier for the blue origami; having fixed that, he did not see the background contributing noise.

Alan noticed how changes in the background triggered the wrong class with Denny wearing a colorful tie-dye shirt.

Adult: Yeah, it still showing your face. Maybe there is a lot of background, so it's still capturing the background, and it's giving the same answer.
Alan: So whenever I leave it stays the same
Adult: Oh you know why because there was Denny in the background when you were training it. So it recognized Denny and not on anything else. That's kinda funny.
Alan: Yeah, it's not getting 100, but when it sees someone else in the background, then it goes to a 100, see?

With scaffolding about his partner's colorful shirt, Alan reasoned that color and background became noise.

Similar to Sandy's observation about faces, Penny stated that hands are not same even when placed in the same position in the frame (see Figure 7a). When asked why this could be happening, she replied,

Penny: Um I think it happened because, like, um, it does not recognize every single detail. It just recognizes like if it's a hand it just by the looks of it, like it doesn't like, take every detail, like if my hand is smaller or your hand is bigger. It doesn't take every detail. ...
Adult: So how would you do this better? ...
Penny: I think I would design it with more detail just by making it like pick up what like maybe they can like. If someone is like doing some action {*raises hand to trigger purple*} they can like zoom in to just like scan it, I guess.

Her idea is similar to labeling parts of an image using bounding boxes, and she adds that choosing an important portion could improve the classifier.

Children experimented with the similarity in color, shape, and portion of camera frame that an object takes. They found

that objects with distinct appearance are easier to train on in contrast to similar-looking objects or changing backgrounds.

E. Tackling Noise in the Classifier

We explore how children used confidence scores and props to avoid noise. For example, John began training the classifier with his face and origami, but he noticed it fluctuated when his adult partner tried it. To fix it, he completely covered the camera with a sticker such that it did not have any background (see Figure 7b). When asked why he used stickers and not his face or props and replied, "*It was easier because, with the face, you have to do it exactly.*" John simplified the task with a training set that is easy to learn and generalize; however, it was not an origami recognizer. It was merely a color identifier.

Amber, Caleb, and Ben trained by adding images at various angles such that a series of images would appear like a wave (see Figure 7c and 3a for Caleb and Amber). When Caleb was asked to explain what he had done, he replied,

> **Caleb**: Basically, I was trying to focus on the different moves, with the same colored shapes, and I was trying to do movements with them. ... It did better with motion.
> **Adult**: Did you try different movements with the same one, and it recognized it?
> **Caleb**: yes

Similar to adult non-experts [26], children had misconceptions related to model capabilities for reasoning. For example, some believed that the classifier recognized movement and improved its performance. All three children that added a wave-like motion found no problems when testing their classifier, but they faced problems when demonstrating them to the group.

V. Discussion and Design Implications

Our study helps characterizes key behaviors of children as young as 7 when interacting with a teachable interface. Given prior work's tendency towards block-based programming, our in-depth analysis of co-design sessions with children provides new insights into approaches that effectively expose a broader group of children to basic machine learning and AI concepts without a programming background.

Our results extend and reaffirm evidence from prior work and reveal new understandings regarding children's interactions with machine teaching. These insights hold the potential to guide the design of future teachable interfaces and early educational experiences about machine learning. We highlight some of them with the following suggestions:

Reveal confidence scores. Following a debugging first approach [58], our observations suggest that children could benefit from being exposed to the model's confidence scores, which have not been explored previously with children. This metric was the output of a softmax function in our study, denoting the distribution of probabilities over the three classes. It became a proxy for children to judge whether the model was confused or unstable. It also led to an emerging practice of building 100% confident models with classes that were easier to classify (*i.e.* by choosing more distinct objects or by zooming in to eliminate any background noise).

Allow for model swapping. The AI for K-12 Initiative [7] recommends that machine teaching applications like GTeach be used in grades K to 5 [59, 60]. When designing such learning activities, we recommend that teachers build-in model swapping opportunities. Our results indicate that inviting children to swap and test their classifiers elicits collaborative observations and reflections and promotes experimentation. Model swapping also exemplifies a core design tenet of constructionist learning: young learners gain opportunities to construct personally meaningful objects and to share them publicly with others [61]. In our study, the children engaged in their knowledge construction process ("in the head") as they actively experimented with their tangible object classifiers ("in the world"), discussing, evaluating, and iteratively expanding their models with their fellow co-designers. The iterative nature of the children's efforts to test and reflect on new hypotheses also echoes the reflect-imagine-create aspect of Resnick's Creative learning spiral [62]. In this way, the iterative nature of teachable machines is well-aligned with the process of learning by trial and error observed in Scratch communities [63]. Moreover, swapping can introduce children to standard machine learning practices that employ train-validate-test steps before system deployment.

Enable quick data inspection. Most prior studies exploring teachable interfaces with children have opted for gesture recognition tasks [18, 34, 35, 36]. As multivariate time series, such data can be difficult to visualize, inspect, and contrast. We suggest that future teachable interfaces include classification tasks that allow children to quickly inspect the data and uncover patterns in a modality that is accessible to them *e.g.*, image classification for sighted children or texture recognition for blind children. When reasoning about classification errors and instability in their models, children in our study often referred to the notion of similarity in shape and color of an image and their training set. This affordance for quick inspection can be further leveraged to incorporate concepts around *variance* and *bias* as well as existing mechanisms around *explainability* in teachable interfaces for children.

VI. Conclusion

Aligned with more recent efforts for an AI curriculum in early education, we explore how machine teaching can expose children to machine learning concepts. We employ co-design sessions with youth 7-13 years old. Our findings and insights can contribute to the ongoing discussion on how children conceptualize, experience, and reflect on their engagement with machine teaching. We discuss how they can guide the design of future teachable interfaces to anticipate children's tendencies, misconceptions, and assumptions. Our findings are being incorporated in developing a teachable interface that exposes children to common barriers for teaching machines.

VII. Acknowledgments

We thank the KidsTeam children and adult co-designers. Dwivedi, U., Gandhi, J., and Kacorri, H. were partially supported by NSF (#1816380) and NIDILRR (#90REGE0008).

References

[1] X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty, "An Overview of Machine Teaching," *arXiv preprint*, 2018. [Online]. Available: http://arxiv.org/abs/1801.05927

[2] Google, "Teachable machine." [Online]. Available: https://teachablemachine.withgoogle.com/v1/

[3] W. House, "Artificial Intelligence for the American People," 2016. [Online]. Available: https://www.whitehouse.gov/ai/

[4] D. Bonjean, "Digital Education Action Plan - Action 10 Artificial intelligence and analytics," Sep. 2018. [Online]. Available: https://ec.europa.eu/education/digital-education-action-plan-action-10-artificial-intelligence-and-analytics_en

[5] D. of International Cooperation Ministry of Science and T. (MOST), "Next generation artificial intelligence development plan," 2017. [Online]. Available: http://fi.china-embassy.org/eng/kxjs/P020171025789108009001.pdf

[6] N. I. for Transforming India (NITI Aayog), "National strategy for ai," 2017. [Online]. Available: https://niti.gov.in/writereaddata/files/document_publication/NationalStrategy-for-AI-Discussion-Paper.pdf

[7] D. Touretzky, "Ai for k-12," 2019. [Online]. Available: https://github.com/touretzkyds/ai4k12/

[8] C. S. T. Association, "2020 state of computer science education: Illuminating disparities." [Online]. Available: https://advocacy.code.org/2020_state_of_cs.pdf

[9] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The scratch programming language and environment," *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4, pp. 1–15, 2010. [Online]. Available: http://dx.doi.org/10.1145/1868358.1868363

[10] L. K. G. at MIT Media Lab, "scratch.mit.edu," 2021. [Online]. Available: https://scratch.mit.edu/

[11] D. Lane, "Machine Learning for Kids," 2017. [Online]. Available: https://machinelearningforkids.co.uk

[12] "AmeriDuo - Teach School Students Robotics & Artificial Intelligence." [Online]. Available: http://ameriduo.com/

[13] D. M. Academy, "Coding + Artificial Intelligence Summer Camp Course - Ages 9 -12," 2020. [Online]. Available: https://www.digitalmediaacademy.org/programming-app-development-camps/adventures-in-artificial-intelligence/

[14] S. Druga, "Growing up with ai: Cognimates: from coding to teaching machines," Ph.D. dissertation, Massachusetts Institute of Technology, 2018. [Online]. Available: http://hdl.handle.net/1721.1/120691

[15] S. Dasgupta and B. M. Hill, "Scratch community blocks: Supporting children as data scientists," in *Proceedings of the 2017 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2017, p. 3620–3631. [Online]. Available: https://doi.org/10.1145/3025453.3025847

[16] D. Touretzky, "The ai4k12 initiative: Developing national guidelines for teaching ai in k-12." [Online]. Available: https://raw.githubusercontent.com/touretzkyds/ai4k12/master/documents/GlobalSWEdu2020_Touretzky.pdf

[17] A. Sarkar, M. Jamnik, A. F. Blackwell, and M. Spott, "Interactive visual machine learning in spreadsheets," in *Proceedings of the 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2015, pp. 159–163. [Online]. Available: https://doi.org/10.1109/VLHCC.2015.7357211

[18] T. Hitron, Y. Orlev, I. Wald, A. Shamir, H. Erel, and O. Zuckerman, "Can children understand machine learning concepts? the effect of uncovering black boxes," in *Proceedings of the 2019 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3290605.3300645

[19] A. Zimmermann-Niefield, R. B. Shapiro, and S. Kane, "Sports and machine learning: How young people can use data from their own bodies to learn about machine learning," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 25, no. 4, p. 44–49, Jul. 2019. [Online]. Available: https://doi.org/10.1145/3331071

[20] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. A. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing, "Machine teaching: A new paradigm for building machine learning systems," *CoRR*, vol. abs/1707.06742, 2017. [Online]. Available: http://arxiv.org/abs/1707.06742

[21] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the People: The Role of Humans in Interactive Machine Learning," *AI Magazine*, vol. 35, no. 4, pp. 105–120, Dec. 2014. [Online]. Available: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2513

[22] H. Kacorri, "Teachable machines for accessibility," *SIGACCESS Accessible Computing*, no. 119, p. 10–18, Nov. 2017. [Online]. Available: https://doi.org/10.1145/3167902.3167904

[23] H. Kacorri, K. M. Kitani, J. P. Bigham, and C. Asakawa, "People with visual impairment training personal object recognizers: Feasibility and challenges," in *Proceedings of the 2017 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2017, p. 5839–5849. [Online]. Available: https://doi.org/10.1145/3025453.3025899

[24] D. Bragg, N. Huynh, and R. E. Ladner, "A personalizable mobile sound detector app design for deaf and hard-of-hearing users," in *Proceedings of the 2016 International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*. New York, NY, USA: Association for Computing Machinery, 2016, p. 3–13. [Online]. Available: https://doi.org/10.1145/2982142.2982171

[25] C. J. Cai and P. J. Guo, "Software developers learning machine learning: Motivations, hurdles, and desires," in *Proceedings of the 2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2019, pp. 25–34. [Online]. Available: https://doi.org/10.1109/VLHCC.2019.8818751

[26] J. Hong, K. Lee, J. Xu, and H. Kacorri, "Crowdsourcing the perception of machine teaching," in *Proceedings of the 2020 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3313831.3376428

[27] E. Wall, S. Ghorashi, and G. A. Ramos, "Using expert patterns in assisted interactive machine learning: A study in machine teaching," in *Proceedings of the 2019 IFIP Human-Computer Interaction Conference (INTERACT)*, ser. Lecture Notes in Computer Science, vol. 11748. Springer, 2019, pp. 578–599. [Online]. Available: https://doi.org/10.1007/978-3-030-29387-1_34

[28] R. Fiebrink, P. R. Cook, and D. Trueman, "Human model evaluation in interactive supervised learning," in *Proceedings of the 2011 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2011, p. 147–156. [Online]. Available: https://doi.org/10.1145/1978942.1978965

[29] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, "Grounding interactive machine learning tool design in how non-experts actually build models," in *Proceedings of the 2018 ACM Conference on Designing Interactive Systems Conference (DIS)*. Hong Kong, China: Association for Computing Machinery, 2018, pp. 573–584. [Online]. Available: https://doi.org/10.1145/3196709.3196729

[30] S. Srikant and V. Aggarwal, "Introducing Data Science to School Kids," in *Proceedings of the 2017 ACM Technical Symposium on Computer Science Education*. Seattle, Washington, USA: Association for Computing Machinery, 2017, pp. 561–566. [Online]. Available: https://doi.org/10.1145/3017680.3017717

[31] A. Dryer, N. Walia, and A. Chattopadhyay, "A Middle-School Module for Introducing Data-Mining, Big-Data, Ethics and Privacy Using RapidMiner and a Hollywood Theme," in *Proceedings of the ACM Technical Symposium on Computer Science Education*. Baltimore, Maryland, USA: Association for Computing Machinery, 2018, pp. 753–758. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3159450.3159553

[32] A. Rao, A. Bihani, and M. Nair, "Milo: A visual programming environment for data science education," in *Proceedings of the 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2018, pp. 211–215. [Online]. Available: https://doi.org/10.1109/VLHCC.2018.8506504

[33] T. Hitron, I. Wald, H. Erel, and O. Zuckerman, "Introducing children to machine learning concepts through hands-on experience," in *Proceedings of the 2018 ACM Conference on Interaction Design and Children (IDC)*. Trondheim, Norway: Association for Computing Machinery, 2018, pp. 563–568. [Online]. Available: https://doi.org/10.1145/3202185.3210776

[34] A. Zimmermann-Niefield, M. Turner, B. Murphy, S. K. Kane, and R. B. Shapiro, "Youth learning machine learning through building models of athletic moves," in *Proceedings of the 2019 ACM International Conference on Interaction Design and Children (IDC)*. New York, NY, USA: Association for Computing Machinery, 2019, p. 121–132. [Online]. Available: https://doi.org/10.1145/3311927.3323139

[35] A. Agassi, I. Y. Wald, H. Erel, and O. Zuckerman, "Scratch Nodes ML: A Playful System for Children to Create Gesture Recognition Classifiers," in *Proceedings of the 2019 ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI)*, 2019, p. 7.

[36] A. Zimmermann-Niefield, S. Polson, C. Moreno, and R. B. Shapiro, "Youth making machine learning models for gesture-controlled interactive media," in *Proceedings of the 2020 ACM International Conference on Interaction Design and Children (IDC)*. New York, NY, USA: Association for Computing Machinery, 2020, p. 63–74. [Online]. Available: https://doi.org/10.1145/3392063.3394438

[37] A. Scheidt and T. Pulver, "Any-cubes: A children's toy for learning ai: Enhanced play with deep learning and mqtt," in *Proceedings of 2019 ACM Conference on Mensch Und Computer (MuC)*. New York, NY, USA: Association for Computing Machinery, 2019, p. 893–895. [Online]. Available: https://doi.org/10.1145/3340764.3345375

[38] H. Vartiainen, M. Tedre, and T. Valtonen, "Learning machine learning with very young children: Who is teaching whom?" *International Journal of Child-Computer Interaction*, vol. 25, pp. 1–11, Sep. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2212868920300155

[39] A. Druin, "Cooperative inquiry: developing new technologies for children with children," in *Proceedings of the 1999 ACM Conference on Human Factors in Computing Systems (CHI)*, Pittsburgh, Pennsylvania, United States, 1999, pp. 592–599. [Online]. Available: https://doi.org/10.1145/302979.303166

[40] M. L. Guha, A. Druin, and J. A. Fails, "Cooperative Inquiry revisited: Reflections of the past and guidelines for the future of intergenerational co-design," *International Journal of Child-Computer Interaction*, vol. 1, no. 1, pp. 14–23, Jan. 2013.

[41] A. Druin, "The role of children in the design of new technology," in *Behaviour & Information Technology*, vol. 21, no. 1, 2002, pp. 1–25. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01449290110108659

[42] E. Bonsignore, J. Ahn, T. L. Clegg, M. L. Guha, J. P. Hourcade, J. C. Yip, and A. Druin, "Embedding

participatory design into designs for learning: An untapped interdisciplinary resource?" in *Proceedings of 2013 International Conference on Computer-Supported Collaborative Learning, (CSCL)*. International Society of the Learning Sciences, 2013, pp. 549–556. [Online]. Available: https://repository.isls.org/handle/1/1963

[43] J. C. Yip, K. Sobel, C. Pitt, K. J. Lee, S. Chen, K. Nasu, and L. R. Pina, "Examining adult-child interactions in intergenerational participatory design," in *Proceedings of the 2017 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 2017, p. 5742–5754. [Online]. Available: https://doi.org/10.1145/3025453.3025787

[44] ISTE, "Iste standards for students," 2021. [Online]. Available: https://www.iste.org/standards/iste-standards-for-students

[45] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of 2009 IEEE conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[47] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[48] J. P. Hourcade, "Interaction design and children," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 4, pp. 277–392, 2007. [Online]. Available: https://doi.org/10.1561/1100000006

[49] V. Nesset and A. Large, "Children in the information technology design process: A review of theories and their applications," *Library & Information Science Research*, vol. 26, no. 2, pp. 140–161, Mar. 2004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0740818804000234

[50] M. Scaife, Y. Rogers, F. Aldrich, and M. Davies, "Designing for or designing with? informant design for interactive learning environments," in *Proceedings of the 1997 ACM Conference on Human Factors in Computing Systems (CHI)*. New York, NY, USA: Association for Computing Machinery, 1997, p. 343–350. [Online]. Available: https://doi.org/10.1145/258549.258789

[51] Google, "A.i. experiments: Teachable machine." [Online]. Available: https://www.youtube.com/watch?v=3BhkeY974Rg

[52] ——, "Teachable machine: Origami," 2017. [Online]. Available: https://youtu.be/_Al6E5Q3ah4

[53] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[54] A. Swalin, "Choosing the right metric for evaluating machine learning models - part 2," 2018. [Online]. Available: https://www.kdnuggets.com/2018/06/right-metric-evaluating-machine-learning-models-2.html

[55] G. Developers, "Imbalanced data," 2020. [Online]. Available: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

[56] Z. Gong, P. Zhong, and W. Hu, "Diversity in machine learning," *IEEE Access*, vol. 7, pp. 64 323–64 350, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2917620

[57] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study of their impacts," *Artificial Intelligence Review*, vol. 22, no. 3, p. 177–210, Nov. 2004. [Online]. Available: https://doi.org/10.1007/s10462-004-0751-8

[58] M. J. Lee, F. Bahmani, I. Kwan, J. LaFerte, P. Charters, A. Horvath, F. Luor, J. Cao, C. Law, M. Beswetherick, S. Long, M. Burnett, and A. J. Ko, "Principles of a debugging-first puzzle game for computing education," in *Proceedings of the 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2014, pp. 57–64. [Online]. Available: https://doi.org/10.1109/VLHCC.2014.6883023

[59] D. Touretzky, "Ai for k-12," 2019. [Online]. Available: https://github.com/touretzkyds/ai4k12/blob/master/documents/CSTA2020_Learning_Activities_K-5.pdf

[60] ——, "The ai4k12 initiative: Developing national guidelines for teaching ai in k-12." [Online]. Available: https://github.com/touretzkyds/ai4k12/blob/master/documents/CSTA_2019_How_To_Teach_AI_Across_K-12.pdf

[61] S. Papert, *Mindstorms : children, computers, and powerful ideas*, 2nd ed. New York: Basic Books, 1993.

[62] M. Resnick and K. Robinson, *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*, 2017. [Online]. Available: https://mitpress.mit.edu/books/lifelong-kindergarten

[63] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 Annual meeting of the American educational research association (AERA)*, vol. 1, 2012, p. 25. [Online]. Available: http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf