DoodleIt: A Novel Tool and Approach for Teaching How CNNs Perform Image Recognition

Vaishali Mahipal
vaishali_mahipal@student.uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

Srija Ghosh
srijaghosh2023@gmail.com
University of Massachusetts Lowell
Lowell, MA, USA

Ismaila Temitayo Sanusi
ismaila.sanusi@uef.fi
University of Eastern Finland
Joensuu, Finland
ismaila_sanusi@uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

Ruizhe Ma
ruizhe_ma@uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

Joseph E. Gonzales
joseph_gonzales@uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

Fred G. Martin
fred_martin@uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

ABSTRACT

To introduce middle school students to key concepts in image recognition, we created an interactive web application that performs sketch recognition and an afterschool curriculum for its use. Our app, called DoodleIt, was inspired by Google's Quick, Draw!, and makes use of its accompanying open-source sketch library. With DoodleIt, students make simple line drawings on a canvas area and a previously-trained convolutional neural network (CNN) identifies the object drawn. The application dynamically visualizes the different layers that are involved in the process of CNNs, including a display of kernels, the resulting feature maps, and the percentage of match at output neurons. We used DoodleIt in an 18-hour curriculum to introduce middle school students to artificial intelligence, machine learning, and data science. Four hours of content were related to image recognition and the ethics of using AI. Here, we describe the design of the DoodleIt application, the approach we used to introduce the associated ideas to the students, and how we assessed student learning. Qualitative data collected from students are presented and discussed. Our findings indicate that students were able to understand the functionality of the kernels and feature maps involved in the CNN to perform rudimentary image recognition.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; • **Human-centered computing** → *Visualization systems and tools*; • **Computing methodologies** → *Computer vision.*

ACE '23, January 30-February 3, 2023, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-9941-8/23/01…$15.00
https://doi.org/10.1145/3576123.3576127

KEYWORDS

Image Recognition, Convolution Neural Networks, Artificial Intelligence, K-12 Students, Middle School Students, Kernels, Feature Maps

ACM Reference Format:

Vaishali Mahipal, Srija Ghosh, Ismaila Temitayo Sanusi, Ruizhe Ma, Joseph E. Gonzales, and Fred G. Martin. 2023. DoodleIt: A Novel Tool and Approach for Teaching How CNNs Perform Image Recognition. In *Australasian Computing Education Conference (ACE '23), January 30-February 3, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3576123.3576127

1 INTRODUCTION

Machine Learning (ML) and Artificial Intelligence (AI) are key drivers of growth and innovation across all industries, and the education sector is no different [11]. According to Hao, "Children today are growing up in a world surrounded by AI: algorithms determine what information they see, help select the videos they watch, and shape how they learn to talk" [8]. Children could become critical consumers of technology if they understood how algorithms are created and how they impact society [17].

AI training in K-12 schools, particularly public schools, is a great way to equip a diverse citizenry with 21^{st}-century skills they need to thrive in an AI-powered world [26]. Thus, teaching AI should be a part of K-12 education [25]. The underlying processes of machine learning are rarely exposed to children and thus they lack the opportunity to acquire accurate ML mental models [9]. It is further important to go beyond engaging children in using AI algorithms; we must unpack the "black boxes" of how these algorithms works [10]. However, resources and support for this are limited. These needs are the motivation for our work.

In this study, we introduced middle school students to image recognition and taught them the processes a computer uses when recognizing images. Our intent was to create a tool with the following functionalities:

- Easily accessible to a large audience and does not require any installation.

- Unveils the functionality hidden behind the black box of neural networks in a way that is understandable by the user with no prior knowledge of AI.
- Is an interactive tool that can engage the user and generate curiosity about AI.

Our accompanying curriculum makes a connection with four of the five "AI4K12 Big Ideas" [25]:

- Computers perceive the world using sensors (Big Idea #1).
- Agents maintain models/representations of the world and use them for reasoning (Big Idea #2).
- Computers can learn from data (Big Idea #3).
- AI applications can impact society in both positive and negative ways (Big Idea #5).

We conducted a pilot study in the context of an after school program. The full program taught the following topics to students over 11 sessions conducted over six weeks:

(1) Introduction to AI and Image Recognition.
(2) Machine Learning Techniques.
(3) Data Creation, Exploration, and Visualization.
(4) Ethical Consideration and Societal Implications.

This paper presents our work on topics 1 and 4. Our work on 2 and 3 is published in [18].

2 RELATED WORK

One of the first steps in democratizing AI to include K-12 students is developing appropriate resources to demystify the concepts. Accordingly, several curricula, tools and approaches have been created to ensure the young learners have access to AI learning [16, 21, 27]. Owing to important role of technology in introducing concepts to learners, several tools have emerged for teaching different AI related concepts [22, 23]. Such tools include Google's Teachable Machine (GTM) [5], LearningML [20], Machine learning for Kids [14] and VotestratesML [12]. Growing empirical evidence from use of these tools suggests they are helpful in supporting student learning of AI concepts. For instance, GTM has been found to be useful in introducing classification and pattern recognition [3]. LearningML has also been effective to communicate how image recognition works among other supervised machine learning concepts [20]. VotestratesML is a tool for supporting K-12 students in building models and making predictions based on real world voting data [12].

While evidence supports the influence of the existing tools to teach basic AI concepts, more tools are required at earlier educational stages, such as middle school (students between ages twelve and thirteen), to further popularize AI knowledge. Inspired by this need, we created DoodleIt, for introducing concepts in convolution neural networks (CNN) to students and uncover the processes hidden behind the black box of image recognition.

Very few tools and studies exist that focus on internal working of CNNs for K-12 students. One of such is the browser-based game, "Contour to Classification," which introduces the structure, components, and functioning of a neural network [15].

DoodleIt differs from this prior work as it visualizes the different layers that are involved in the process of CNNs: the display of kernels, the resulting feature maps, and the percentage of the match at output neurons. DoodleIt was inspired by Google's Quick, Draw! [13] and makes use of the open-source sketch library provided by this project. As observed by Gresse von Wangenheim [6], there is lack of information on how AI tools have been developed and evaluated. Here, we describe the design of the DoodleIt application, the approach we used to introduce the associated ideas to the students, and how we assessed student learning.

3 METHODOLOGY

DoodleIt was created as a web-based application that recognizes the drawings of different objects made on a canvas. The tool identifies what drawing is being made on the canvas while the user is actively drawing. It displays the filters (also known as kernels) that were trained in the convolution neural network model. It visualizes their application to the source image, producing respective feature maps in real time. DoodleIt was trained to recognize six different objects: cat, sheep, door, cake, apple, and triangle. These six categories were chosen based on their ease of drawing from the 345 categories in Google's Quick, Draw! dataset [13].

3.1 Motivation

Today's youth are continuously exposed to a multitude of image recognition applications. Examples include Face ID for phone security, visual-image search engines to identify certain products and logos, road navigation for self-driving cars, and many others. Since youth likely have encountered some form of image recognition, we thought it to be beneficial that they understand the processes behind how it works. Through this, we could help students gain a better understanding of the world around them and develop their curiosity about AI.

3.2 Implementation of DoodleIt

We use a convolutional neural network (CNN) for the image classification model in DoodleIt. Our CNN model was trained in Python and consists of six classes (cat, sheep, apple, door, cake, triangle). We created our model using Keras, a high-level API of TensorFlow 2, which provides the necessary abstractions and building blocks to build sophisticated machine learning models [4]. As mentioned, we used Google's Quick, Draw! [13] dataset in order to train our predictive CNN. This dataset consists of 50 million drawings amongst 345 categories with each image formatted as a 28*28 grayscale NumPy image. Each of DoodleIt's six classes comprised 50K images from the dataset, totaling 300K images altogether. Our six-class dataset is divided into training images (180K) and testing images (120K). A sample of the training dataset is shown in Fig. 1.

Image inputs are normalized such that the pixel values of each image are between zero and one, speeding up the training process. Our CNN architecture consists of six layers and one convolutional layer. As shown in Fig. 2, our model has:

- A convolution layer with eight filters of three by three in size with a ReLU activation function;
- A max-pooling Layer with a pool size of two by two;
- A flatten layer;
- A dropout layer with a probability of 0.2;
- A dense layer with 32 neurons and ReLU activation function;
- A output/softmax Layer.

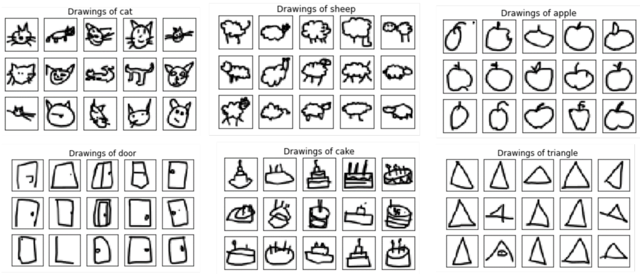

Figure 1: Sample Drawings from the Training Dataset

To explain CNNs to students, we focused on the fully-connected network between the feature maps and the output neurons. We chose this "one-layer" CNN model because it would be easier to understand than the typical multi-layer CNN designs. Also, our one-layer, eight-filter model produced simple, intelligible kernels. As shown in Fig. 3 (a), one kernel specialized in detecting diagonal lines while another detected vertical lines. We did test a three-layer CNN model; while it performed better, its kernels were ambiguous in terms of their feature extraction (Fig. 3 (b)).

Our one-layer CNN model was exported as a Keras H5 file and converted to a JSON format which contained the model's architecture, weight values, and compiled information. The JSON format is compatible with HTML, CSS, JavaScript, and PHP, which we used to build DoodleIt's front- and back-ends. We imported the TensorFlow.JS library [1] and used it for importing the JSON model into the front-end itself, along with visualizing the filter and feature maps.

We based our front-end design on open-source code shared by Yining [24]. We added several important features to our front-end, including visualizing the filters and feature maps, implementing prompts for student exploration, and color-coding output neurons to signify different levels of accuracy percentage. We have designed DoodleIt in such a way that it represents the actual CNN architecture.

The full DoodleIt app is shown in Fig. 4 :

- On the left side, the interactive canvas allows users to use their mouse to make drawings.
- Next to the canvas, filters extracted by our trained CNN model are displayed.
- As the user makes a drawing, feature maps resulting from application of the filter to the input image are visualized, and update in real-time as the user draws on the canvas. Through this, users gain insight into how a computer interprets their drawing using filters and feature maps.
- Six output neurons display a percentage associated of match associated with each image class. The neurons are color-coded based on this value. For example, green denotes percentage matches greater than 85% and red, less than 15%.
- Below the canvas, a prompt encourages user exploration.
- For research purposes, a screenshot is captured each time the user finishes their work (upon clicking the clear button). A code name field records the anonymous unique identifier of research participants.

Source code for DoodleIt is available at [7] , and the live demo is available at [19].

4 THE CURRICULUM

Our curriculum for image recognition was part of an 18-hour long AI after school program for middle school students. The curriculum had the following components: (1) Use of Google's Teachable Machine and discussion of ethics of face recognition; (2) a presentation on neural networks; (3) use of DoodleIt; (4) a presentation of kernels/filters and convolution, followed by a paper and pencil kernel activity; and (5) videos, conversations, and interviews about the ethics of using image recognition in the world.

4.1 Google Teachable Machine & AI Ethics

Through Google's Teachable Machine [5], students gained an understanding of classification problems and the importance of training dataset. Our discussion of the ethics of image recognition was guided by videos, including introducing algorithmic bias via the Gender Shades work [2] (Big Idea #5).

4.2 Introduction to Image Recognition

A slide presentation introduced the students to some basic concepts of image recognition, neural networks, and convolutional neural networks. Children were shown that the computer reads an image in the form of numbers (Big Idea #1) and feeds them into the neural network in order to recognize images (Big Idea #2).

The visual approach to explaining neural networks generated interest and discussion. One student asked, "Why does it says 15 percent dog when it is clearly a cat?" as shown in Fig. 5. We explained to them the cat and dog have some similar features which make the computer think that it might be a dog. They were also introduced to the concepts of training and testing data.

4.3 Interaction with DoodleIt

After students were shown a demo of DoodleIt, they were invited to use it. They used the prompts in the drop-down menu and experimented with the application freestyle. We helped them understand how filters and feature maps worked. We helped them understand that the filters recognize particular features in the images drawn by them, such as diagonal and vertical lines.

After students interacted with the tool, we showed them sample images from the training dataset (Fig. 1). This helped them understand why the computer could or could not properly identify their drawing (Big Idea #3).

We then returned to the slide presentation to show how the filters and their respective feature maps are involved in the CNN process. We focused on the feature extraction process and the output layer of the CNN as shown in Fig. 2.

We showed them how image drawn by them is passed to the different layers with the help of the diagram shown in Fig. 2. The drawn image is first passed to eight filters which results in their respective feature maps and those feature maps pass through different layers like max pooling, dropout layer and fully connected layer for further processing. We briefly introduced these terms and did not explain in depth as it could be overwhelming for the beginners.

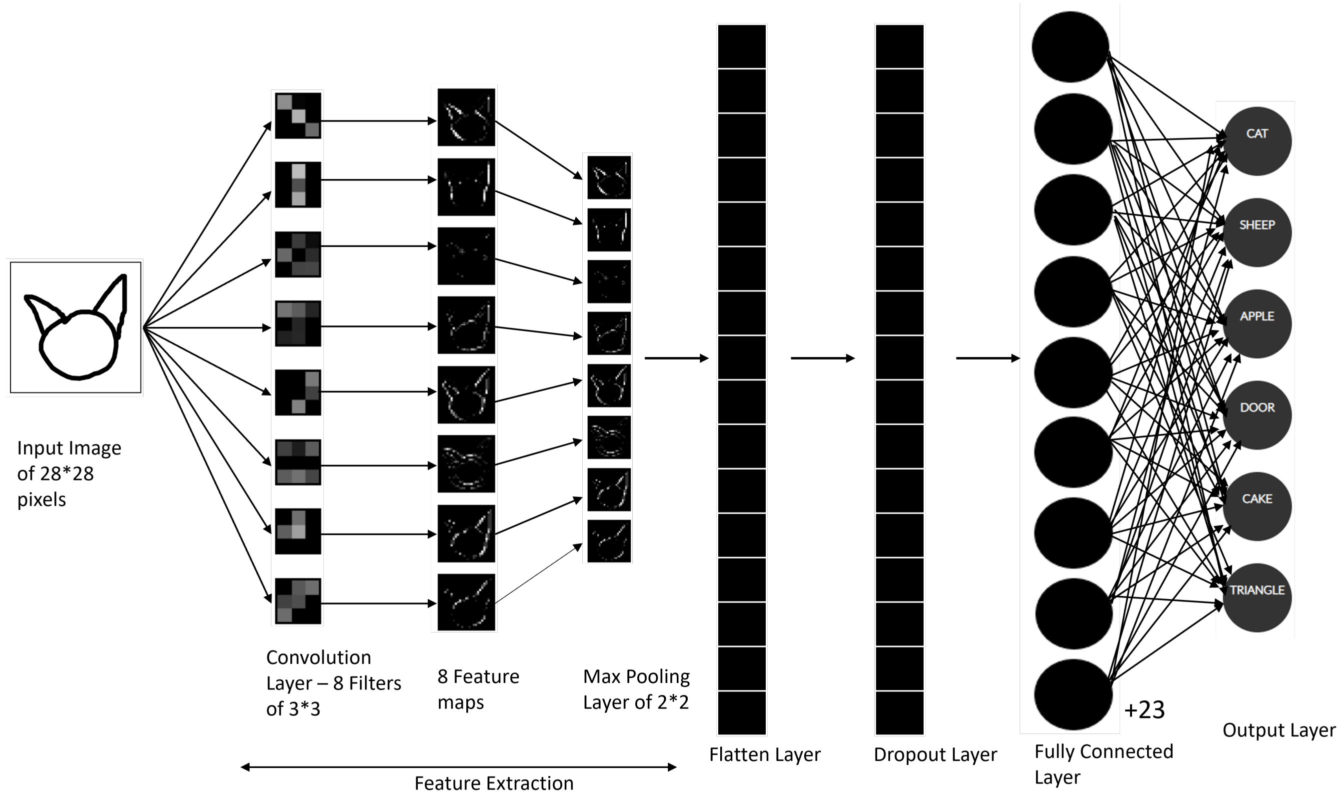

Figure 2: CNN Architecture for DoodleIt

(a) Diagonal and Vertical filters **(b) Ambiguous Filters**

Figure 3: Comparison between filters trained in different layered models

And the last layer is the output layer that gives the probability for each object.

4.4 Kernel Paper Activity

After the interaction with the DoodleIt application, we did a kernel paper activity where we explained to them how filters are capturing some features in an image. In terms of logistics, this is a straightforward pen and paper exercise that is inexpensively and simply printable on paper and a translucent sheet. This activity was designed in such a way that no prior knowledge of matrix multiplication or complicated computation is needed. We gave them the worksheet with an image of one digit and an empty box for the feature map for them to fill out and an acetate sheet that had a filter made on it with a cut-out.

In the worksheet, black cells represented a binary 1 value and white cells represented 0 as shown in Fig. 6. The acetate sheet had translucent, pink-colored squares to represent the values of 1 in the filter as shown in Fig. 6. Students were first asked to position the acetate (representing the filter) on the upper-left 3×3 corner of the input image and count the number of overlapping black and pink boxes. (This enacted the matrix multiply step of convolution, which was not explained in detail to the students.)

Then students recorded this sum in a corresponding box of the feature map, which was revealed by the cut-out square of the acetate. As the students moved the acetate filter, they filled in different boxes on the feature map. Representative student work is shown in Fig. 6.

4.5 Assessments

Student performance was evaluated through a series of assessments, including a post-survey questionnaire, interviews, and written assessments.

- The post-survey questionnaire had four multiple choice questions, some of which are as follows:
  - How successful do you think the model was in recognizing the images?
  - Which filter is the diagonal filter?
- Some of the interview prompts used were:

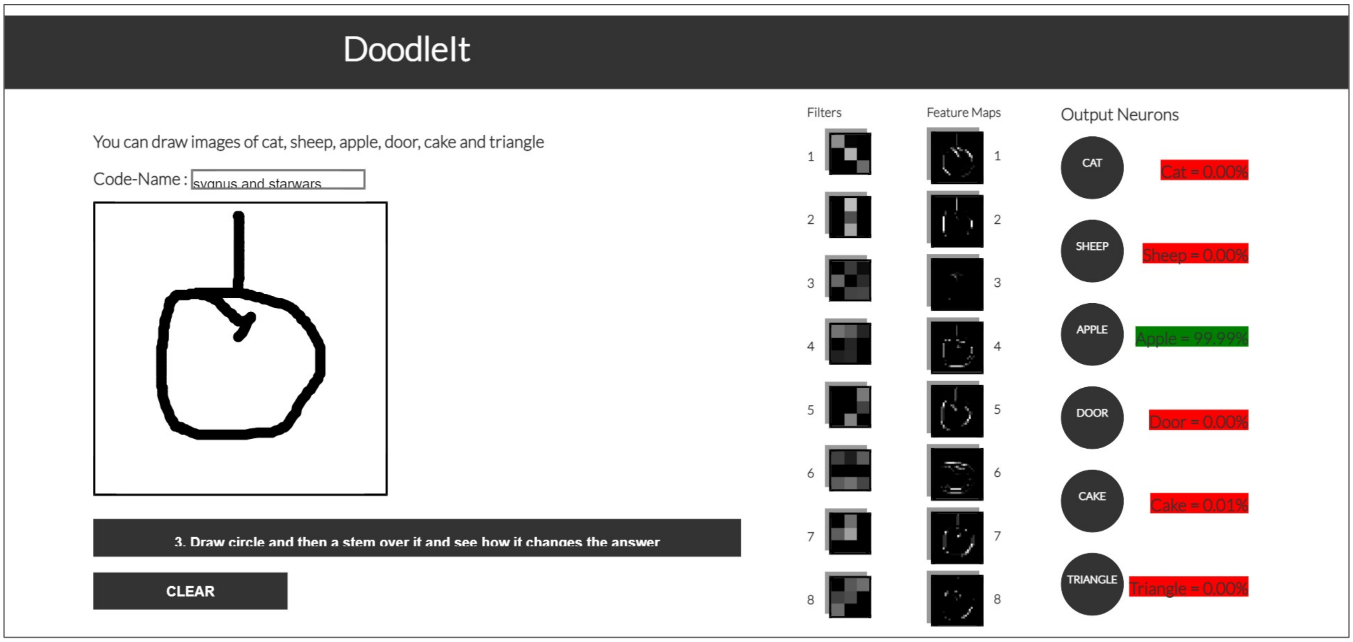

Figure 4: Screenshot of DoodletIt Application

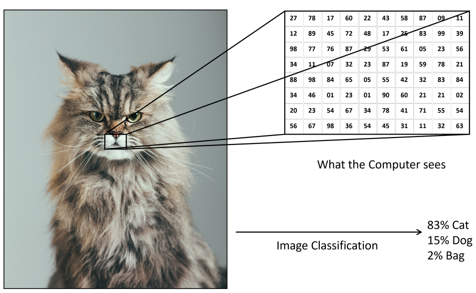

Figure 5: Slide to introduce image digitization and categorization

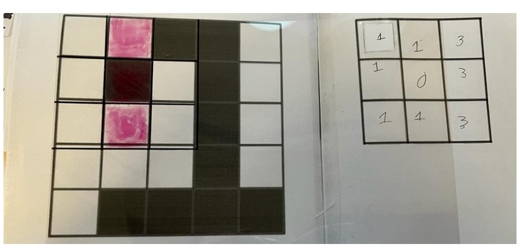

Figure 6: Example of Kernel Activity Worksheet

- How do you think DoodleIt recognizes images?
- What objects does DoodleIt always guess correctly? Why do you think this is so?
- Do some of the filters recognize different types of features in the image?
- If you draw an object other than these 6, then what happens and why do you think it happens?
- What object does the demo guesses wrongly most often and why do you think this is so?

- A written assessment asked students to predict how a neural net would identify a drawing of a dog when the net was not trained on dog images, and to match filters with resulting feature maps (Fig. 9).

5 DATA ANALYSIS AND RESULTS

We collected data from four students and to analyze this data we have done a single-subject design for our four subjects. We also made observations while students completed their activities; i.e., we noted their questions, observations, and understanding of the activities as they went through them. We first analyzed all the data collected from the students according to the sequence of the performed activities. The sequence of the activities performed was : doodleIt interaction, kernel activity, post-survey questionnaire, interviews, and written assessment.

We present our findings in the form of student stories which tie together the data from every activity done by an individual or student group. To maintain student anonymity, we refer to the students by number. The students included three boys and one girl, aged 12 to 13 years old.

Student 1. This student interacted with the application free-style and only used a few prompts from the drop-down as he did not find the challenges drop-down helpful. He believed the application used a filter and compared the drawn image with the training data (which he referred to as a "box of code"). He observed that the application recognized some images correctly but never all of them. He observed that

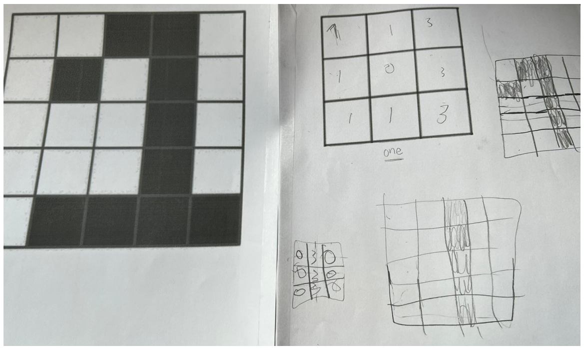

Figure 7: Student 1 response to Kernel Paper Activity

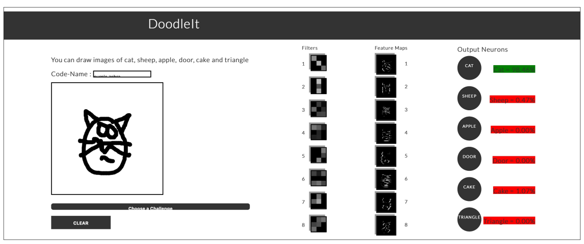

Figure 8: Student 2 interaction with DoodleIt

cat doodles were recognized the most and opined that it may be because it most closely resembles a circle, and it was the most common "box of code" which made it easy for the application. He showed understanding of the fact that the application is learning from what it has been trained on (referred to as "it has seen"). He determined that if an object is drawn which is not part of the dataset, the application will try to guess something from the dataset which closely resembles the doodle. He believed that the application got the apple wrong most often because it resembles a circle which is similar to other objects (e.g., cat or cake) which the application has been trained on the most.

He was able to do the kernel paper activity correctly and he made his own image of number one and applied the vertical filter and realized that it picked up the same vertical line feature as the one in the worksheet as shown in Fig. 7.

Overall, he found the activity interesting and had fun. His overall responses in the interview and the activities suggested that he understood that the computer is able to recognize the images because it has been pre-trained with images of those objects and because of that it looks for those features while making a prediction.

Student 2. This student started interacting with the application using the prompts in the drop-down and found them helpful and later explored on her own as shown in Fig. 8. In one of the post-questionnaires, she answered that she got the cake mostly correct; it could be because she got mostly cake as the prediction, even if the object drawn was not a cake. She understood that some filters were capturing some particular features in the image and were clearly able to explain the features that make an object different from another. For example, when she was asked what makes a sheep different from a cake she said, "Sheep has legs upside down, not on the top like the candle on the cake." She thought that model could not detect the image of a door because whenever she drew a door it was predicted as cake and she thought it could be because of the doorknob which could be interpreted as cake. Initially, she thought that the computer chooses randomly if any object other than the six objects were drawn but her results from the written assessment suggest that she understood computer will predict the object that it is most similar with.

Students 3 and 4. This group interacted with the application using some prompts and sometimes of their own. The group tried making objects different from these six objects which always gave them a cake that clearly was not a cake and it made them think that the computer always guesses cake wrong. The observations from this group were in stark contrast with Student 1. The overall consensus among the group was negative as they felt the application could not recognize most of their drawings correctly. Student 3 did not answer one of the questions in the post-survey questionnaire when asked which object the applications guesses correctly mostly because he thought there was no object that the application mostly guesses correctly. However, they liked the kernel activity and through this activity, they understood the functionality of the filters. Responses from the written assessment of Student 3 as shown in Fig. 9 suggested that he understood the functionality of the filters and how training data is important in predicting the images.

6 DISCUSSION AND LIMITATIONS

We found that DoodleIt often did not perform well in terms of identifying images. We did not experience the same performance while testing the application, where it was almost always correct. We think one of the reasons might be the way the drawings were made on the canvas. In our testing, we realized we made all our drawings to fill the canvas, whereas students tried to make drawings in all parts of the canvas. In subsequent work with different groups of users, we have coached them to fill the whole canvas with their drawings and, consequently, DoodleIt performed much better.

The application worked in terms of explaining the concepts of filter and feature maps to the students.

A three-layer convolution model would have performed better in terms of image recognition, but we wanted to be transparent with the students in terms of the technology we used for the image recognition and the filters in one layer convolution model were more visually representative. Keeping the balance between performance and difficulty was important to us. Our study suggests that the application was helpful in educating all these concepts.

Although this activity was designed to help middle school students understand image recognition, it can be helpful to others with varying prior knowledge of AI. A graduate machine learning class was also introduced to the DoodleIt application. After viewing the interface, students could gain a deeper understanding of

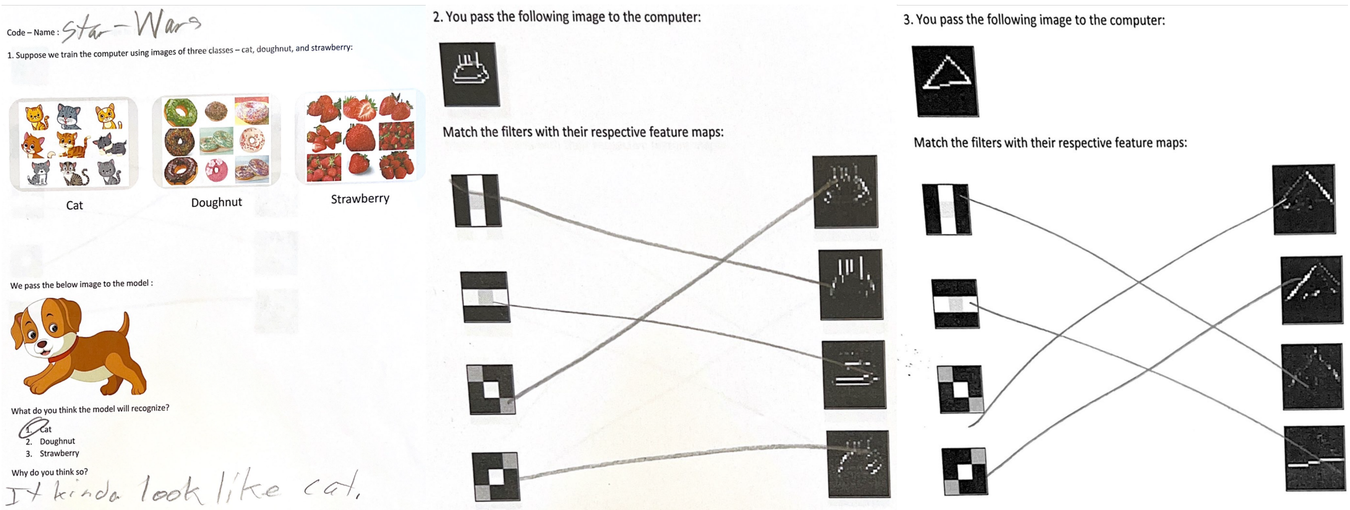

Figure 9: Student 3 response to written assessment

neural networks and lay a foundation for constructing more complex neural networks. The interactive process of real-time drawing can better engage students. The simple filters and feature maps make it easy to explain and comprehend what the filters do, and the output is straightforward. Although convolutional neural networks (CNN) are trained and applied using matrix computation and back-propagation, visual aids are very beneficial in helping students comprehend the inner workings of a CNN.

During an AI workshop with K–12 teachers, we completed the DoodleIt curriculum, including the kernel pen–paper activity. One of the instructors noticed that the vertical filter functions as a detector for the number "1" (which was an accurate observation). Overall, they felt the DoodleIt connected with them, and they wanted to share it with their pupils.

The kernel activity was very beneficial in teaching them the functionality of the filters and how feature maps are generated. We were able to unveil some of the functionalities used in the black box of neural networks. Our students had no prior knowledge of machine learning or artificial intelligence, and these activities were able to help them understand the concepts of the feature extraction and training data used in image recognition.

Our work was successful in generating curiosity about AI in these students. The participant group was too small to draw definitive statistical results, but it shows promise. We intend to conduct this study with a larger group of students to generate more insights and to allow the generalizability of the findings.

7 CONCLUSION AND FUTURE DIRECTIONS

Evidence collected from the users suggests that students understood aspects of convolution neural networks (CNNs): filters/kernels, feature maps, and combining results to get a probability of identification. Students were also able to understand the importance of the training data and that the neural network will only give an answer based on the categories it is trained on. They understood that CNNs work by feature extraction.

While two students were frustrated because of incorrect recognition at the beginning, the other two students found the activity fun. But they all felt that they learned something. Overall, everyone enjoyed the title "DoodleIt" and collectively came to the playful conclusion that "people make too many cakes."

We can add different versions of CNNs in the future, such as one layer and three layers, in order to allow students to interact with the one layer first, then with the three layers, and compare the performance of the two models.

This study suggests that the combination of tutorial presentations, activities with the DoodleIt app, and kernel exercises was a meaningful introduction to CNNs for fifth and sixth-grade students. It helped them understand the concepts of CNNs and also generated curiosity about AI among them.

ACKNOWLEDGMENTS

We would like to thank the teachers and students involved in carrying out this study. Sam Zuk provided helpful support in conducting the afterschool activities. We would like to thank Stephanie Rosenthal for allowing us to share DoodleIt with a K-12 teacher workshop she organized. This material is based upon work supported by the National Science Foundation under Grant No. 2112633.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Joy Buolamwini. 2018. *Gender Shades*. https://www.media.mit.edu/projects/gender-shades/overview/

[3] U. Dwivedi, J. Gandhi, R. Parikh, M. Coenraad, E. Bonsignore, and H. Kacorri. 2021. Exploring Machine Teaching with Children. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–11. https://doi.org/10.1109/VL/HCC51201.2021.9576171

[4] Google. 2015. *About Keras*. https://keras.io/about/

[5] Google. 2020. *Google Teachable Machine*. https://teachablemachine.withgoogle.com/

[6] Christiane Gresse von Wangenheim, Jean C. R. Hauck, Fernando S. Pacheco, and Matheus F. Bertonceli Bueno. 2021. Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping. *Education and Information Technologies* 26, 5 (sep 2021), 5733–5778. https://doi.org/10.1007/s10639-021-10570-8

[7] Engaging Computing Group. 2022. *DoodleIt*. https://github.com/engaging-computing/DoodleIt

[8] Karen Hao. 2019. *Kids are surrounded by AI. They should know how it works.* https://www.technologyreview.com/2019/09/13/133056/kids-are-surrounded-by-ai-they-should-know-how-it-works/

[9] Tom Hitron, Yoav Orlev, Iddo Wald, Ariel Shamir, Hadas Erel, and Oren Zuckerman. 2019. Can Children Understand Machine Learning Concepts? The Effect of Uncovering Black Boxes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300645

[10] Cindy E. Hmelo and Mark Guzdial. 1996. Of Black and Glass Boxes: Scaffolding for Doing and Learning. In *Proceedings of the 1996 International Conference on Learning Sciences* (Evanston, Illinois) *(ICLS '96)*. International Society of the Learning Sciences, 128–134.

[11] David Karandish. 2021. *7 Benefits of AI in Education*. https://thejournal.com/articles/2021/06/23/7-benefits-of-ai-in-education.aspx

[12] Magnus Hoeholt Kaspersen, Karl-Emil Kjaer Bilstrup, Maarten Van Mechelen, Arthur Hjorth, Niels Olof Bouvin, and Marianne Graves Petersen. 2021. VotestratesML: A High School Learning Tool for Exploring Machine Learning and Its Societal Implications. In *FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education* (St. Gallen, Switzerland) *(FabLearn Europe / MakeEd 2021)*. Association for Computing Machinery, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3466725.3466728

[13] Google Creative Lab. 2017. *Quick, Draw!* https://experiments.withgoogle.com/quick-draw

[14] Dale Lane. 2020. *Machine Learning for Kids*. https://machinelearningforkids.co.uk

[15] Irene Lee and Safinah Ali. 2021. The Contour to Classification Game. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 17 (May 2021), 15583–15590. https://ojs.aaai.org/index.php/AAAI/article/view/17835

[16] Irene Lee, Safinah Ali, Helen Zhang, Daniella DiPaola, and Cynthia Breazeal. 2021. Developing Middle School Students' AI Literacy. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 191–197. https://doi.org/10.1145/3408877.3432513

[17] Duri Long and Brian Magerko. 2020. What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI conference on human factors in computing systems.* 1–16.

[18] Ruizhe Ma, Ismaila Temitayo Sanusi, Vaishali Mahipal, Joseph E. Gonzales, and Fred G. Martin. 2023. Developing Machine Learning Algorithm Literacy with Novel Plugged and Unplugged Approaches. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3545945.3569772

[19] Vaishali Mahipal, Srija Ghosh, and Fred Martin. 2022. *DoodleIt*. https://tinyurl.com/mydoodleit

[20] Juan Rodríguez, Jesús Moreno-León, Marcos Román-González, and Gregorio Robles. 2020. LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects. *Revista de Educación a Distancia (RED)* 20 (04 2020). https://doi.org/10.6018/red.410121

[21] Ismaila Temitayo Sanusi and Solomon Sunday Oyelere. 2020. Pedagogies of Machine Learning in K-12 Context. *2020 IEEE Frontiers in Education Conference (FIE)* (2020), 1–8.

[22] Ismaila Temitayo Sanusi, Solomon Sunday Oyelere, Friday Joseph Agbo, and Jarkko Suhonen. 2021. Survey of Resources for Introducing Machine Learning in K-12 Context. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE49875.2021.9637393

[23] Ismaila Temitayo Sanusi, Solomon Sunday Oyelere, Henriikka Vartiainen, Jarkko Suhonen, and Markku Tukiainen. 2022. A systematic review of teaching and learning machine learning in K-12 education. *Education and Information Technologies* (2022), 1–31.

[24] Yining Shi. 2019. *Yining1023/doodleNet*. https://github.com/yining1023/doodleNet

[25] David Touretzky, Christina Gardner-McCune, Fred Martin, and Deborah Seehorn. 2019. Envisioning AI for K-12: What Should Every Child Know about AI? *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 9795–9799. https://doi.org/10.1609/aaai.v33i01.33019795

[26] Randi Williams. 2021. How to Train Your Robot: Project-Based AI and Ethics Education for Middle School Classrooms. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 1382. https://doi.org/10.1145/3408877.3439690

[27] Qi Xia, Thomas K.F. Chiu, Min Lee, Ismaila Temitayo Sanusi, Yun Dai, and Ching Sing Chai. 2022. A self-determination theory (SDT) design approach for inclusive and diverse artificial intelligence (AI) education. *Computers & Education* 189 (2022), 104582. https://doi.org/10.1016/j.compedu.2022.104582