SEML: Designing Scratch Programming Extensions for Machine Learning

Mark Kocherovsky, CJ Chung, PhD
College of Arts and Sciences, Lawrence Technological University

INTRODUCTION

Machine learning is a subfield of computer science that studies algorithms capable of training predictive models from given sets of data and previous experience [1] [2] [3]. To make machine learning more accessible, Google offers a web-based tool called Teachable Machine to allow users to make their own models using image, pose, and sound-based training [4]. Teachable Machine itself uses TensorFlow.js [5], which uses JavaScript to create a neural network for training and output [6]. It allows users to upload the trained model to the cloud for later usage, which will be an integral component in SEML.

Scratch is a programming language and environment created by the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) Media Lab to teach young people programming concepts. It is a visual language that uses blocks to represent lines of code. The user can directly see and add objects on the output screen (such as Scratch's mascot, a cartoon cat, or the background), the user can import or record sounds, and allow the user to provide input through the keyboard and mouse. The system is very versatile, with the inclusion of variables, control features such as loops and conditional statements, logic, and the ability for the user to create their own functions. Using this environment, the user can easily program animations and games [7].

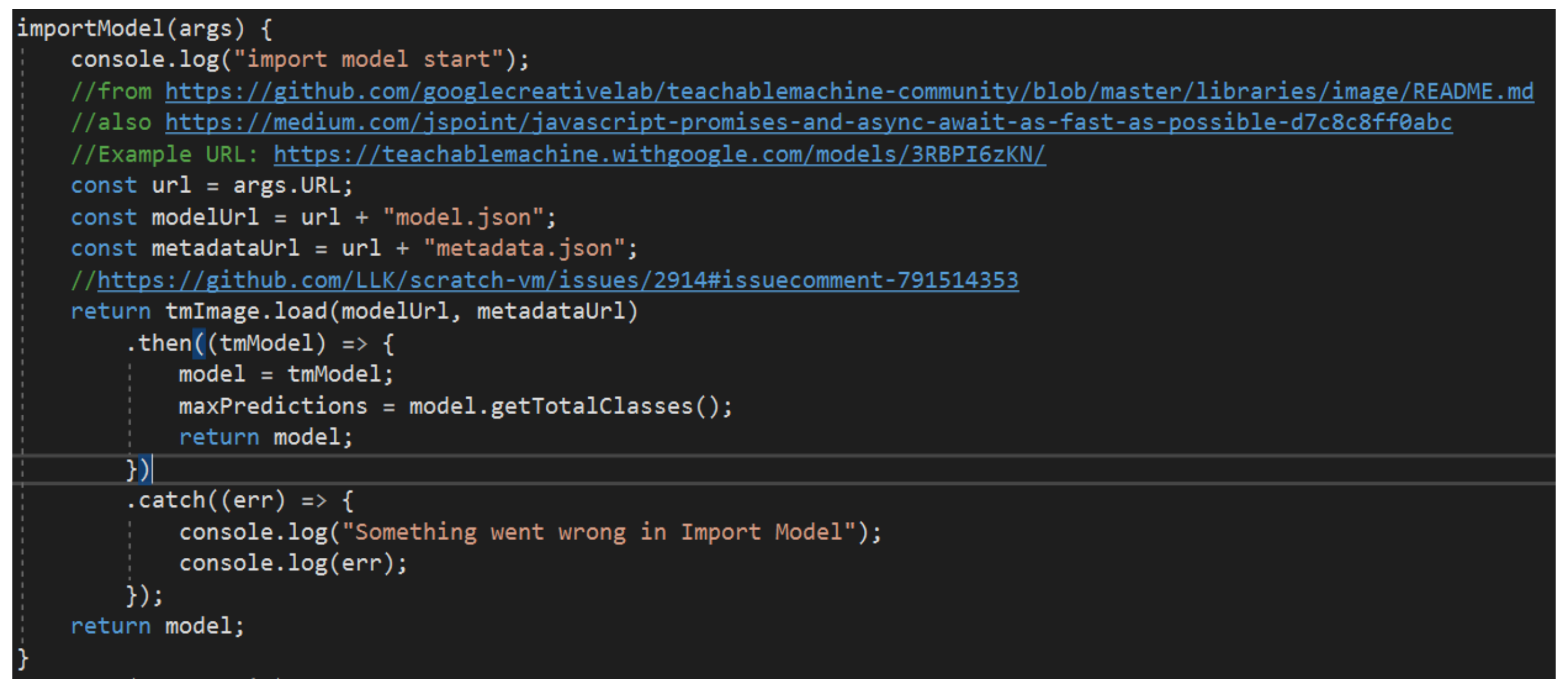
The SEML Program has two components. First, a test extension was made to detect the location of a user's palm using a Teachable Machine model. However, the main component of SEML is the ability to directly import a Teachable Machine model without the use of any intermediaries.

References

[1] C. Chung, Neural networks and deep learning with Python (2020), [PowerPoint Slides].
[2] C. Chung, Introduction to AI & ML with Scratch [Pt. 1] (2021), [PowerPoint Slides; Online; accessed 28-February-2021].
[3] C. Chung, Introduction to AI & ML with Scratch [Pt. 2] (2021), [PowerPoint Slides; Online; accessed 28-February-2021].
[4] Alphabet Incorporated, Teachable Machine (2021), [Online; accessed 02-February-2021].
[5] Alphabet Incorporated, Teachable Machine - FAQ (2021), [Online; accessed 02-February-2021].
[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah,M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Tal-war, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Víegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
[7] M. Marji, Learn to Program with Scratch, 1st ed. (No Starch Press, 2014) [Online; accessed 06-December-2020].
[8] K. Chadha, C. Willis-Ford, and u9g, Scratch 3.0 Extensions (2020), [Online; accessed 30-March-2021].
[9] I. Alverado and J. Jongejan, Teachable Machine Library - Image (2019), [Online; accessed 28-February-2021].
[10] C. Chung and M. Kocherovsky, CS+PA2: Learning computer science with physical activities and animation — A MathDance experiment, in 2018 IEEE Integrated STEM Education Conference (ISEC) (2018) pp. 262–267.
[11] M. Shamir, M. Kocherovsky, and C. Chung, A Paradigm for Teaching Math and Computer Science Concepts in K-12 Learning Environment by Integrating Coding, Animation, Dance, Music and Art, in 2019 IEEE Integrated STEM Education Conference (ISEC) (2019) pp. 62–68.

Other acknowledgements:

- Dr. Mitchel Resnick - MIT Media Lab
- Eric Rosenbaum - MIT Media Lab
- Dale Lane - IBM
- GitHub users apple502j and adroitwiz
- The Scratch Multimedia Lab at MIT for helping with technical issues.



```
importModel(args) {
    console.log("import model start");
    //from https://github.com/googlecreativelab/teachablemachine-community/blob/master/libraries/image/README.md
    //also https://medium.com/jspoint/javascript-promises-and-async-await-as-fast-as-possible-d7c8c8ff0abc
    //Example URL: https://teachablemachine.withgoogle.com/models/3RBPI6zKN/
    const url = args.URL;
    const modelUrl = url + "model.json";
    const metadataUrl = url + "metadata.json";
    //https://github.com/LLK/scratch-vm/issues/2914#issuecomment-791514353
    return tmImage.load(modelUrl, metadataUrl)
        .then((tmModel) => {
            model = tmModel;
            maxPredictions = model.getTotalClasses();
            return model;
        })
        .catch((err) => {
            console.log("Something went wrong in Import Model");
            console.log(err);
        });
    return model;
}
```

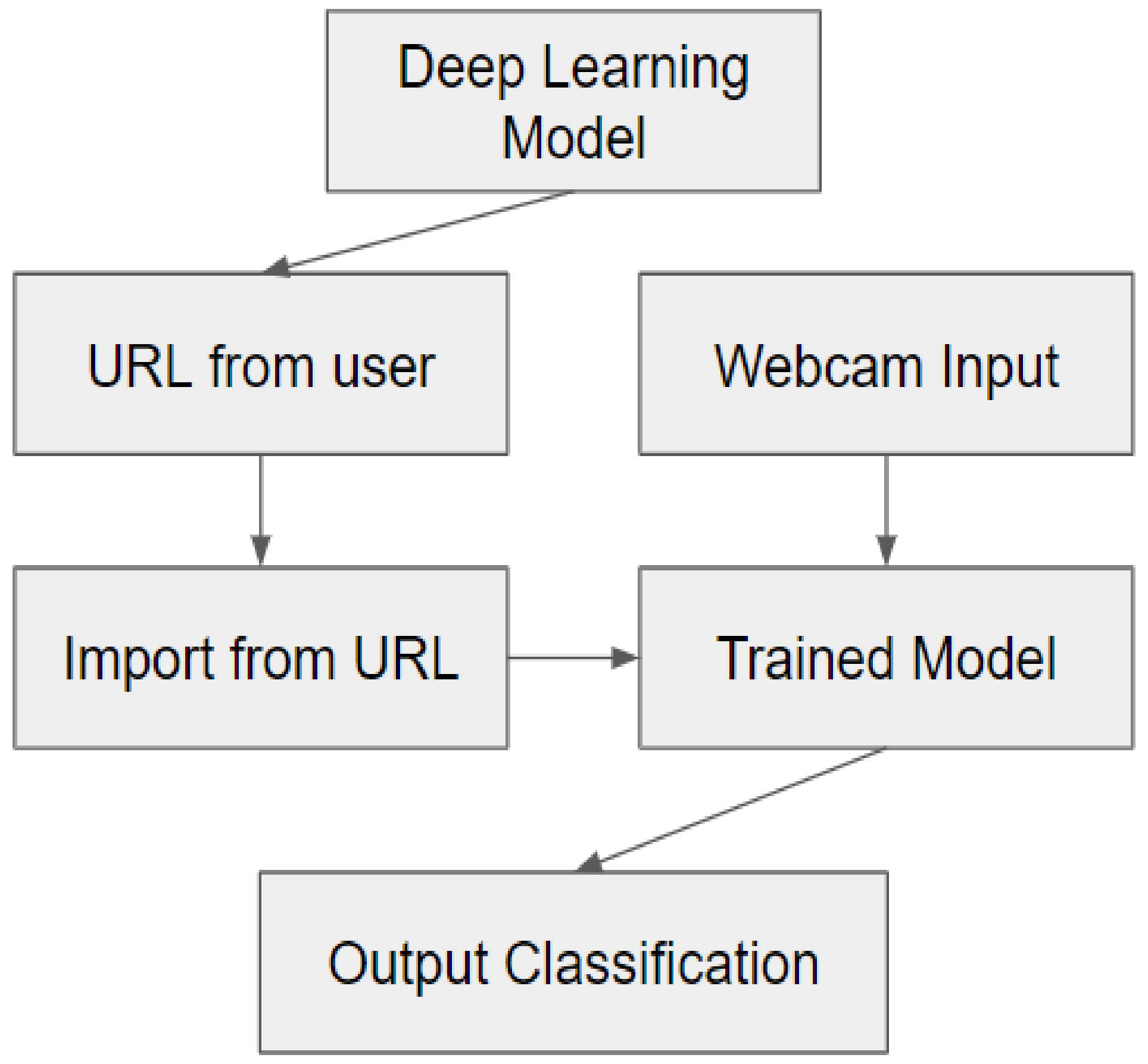
Code for SEML's Import Model block. The url is passed in as part of the "args" (arguments), and the model is fetched from the cloud based on the url.

DESIGN

Scratch 3.0 extensions are JavaScript files that contain classes that define the extensions themselves, each block, the function corresponding to each individual block, and various helper functions. Blocks can take arguments, include graphics, and contain menus [8].

Palm Detection
The palm detection test module will be able to detect the presence of the user's palm using their webcam and return its position, which the user can then export to a variable. This is a test component to gauge Scratch's ability to import Teachable Machine models from the cloud. The Palm Detection test module begins by requiring the user to import the model from the cloud. Since the Teachable Machine model already exists, its URL is hardcoded in the program's backend. The import command, "Import Block," then connects the user to the cloud, downloads the palm detection model, and initializes the model. The user can then select blocks entitled "X Coordinate" and "Y Coordinate", which will return the x-coordinate and the y-coordinate of the user's palm depending on the quadrant it is in. These blocks function by using the model to ascertain the location of the user's palm.

SEML:
The full SEML extension shall allow the user to import any Teachable Machine model from the cloud, provided that they have the URL. SEML will then allow the user to implement the model in the program, which will return output that can be exported into variables. The main SEML program will function nearly identically to the Palm Detection test module. However, there are two key differences. First, the Import Model command includes a text input field in which the user can place the URL of a Teachable Machine model. The block, when run, then automatically initializes the given model based on the URL, which uses TensorFLow libraries [9]. Secondly, instead of specific "X Coordinate" and "Y Coordinate" blocks, SEML only provides a single "Return Prediction" block, which allows the user to export the results of their model into a variable, as SEML requires a more general set of commands. The diagram below provides a high-level overview of the SEML module's design.



Deep Learning Model

URL from user

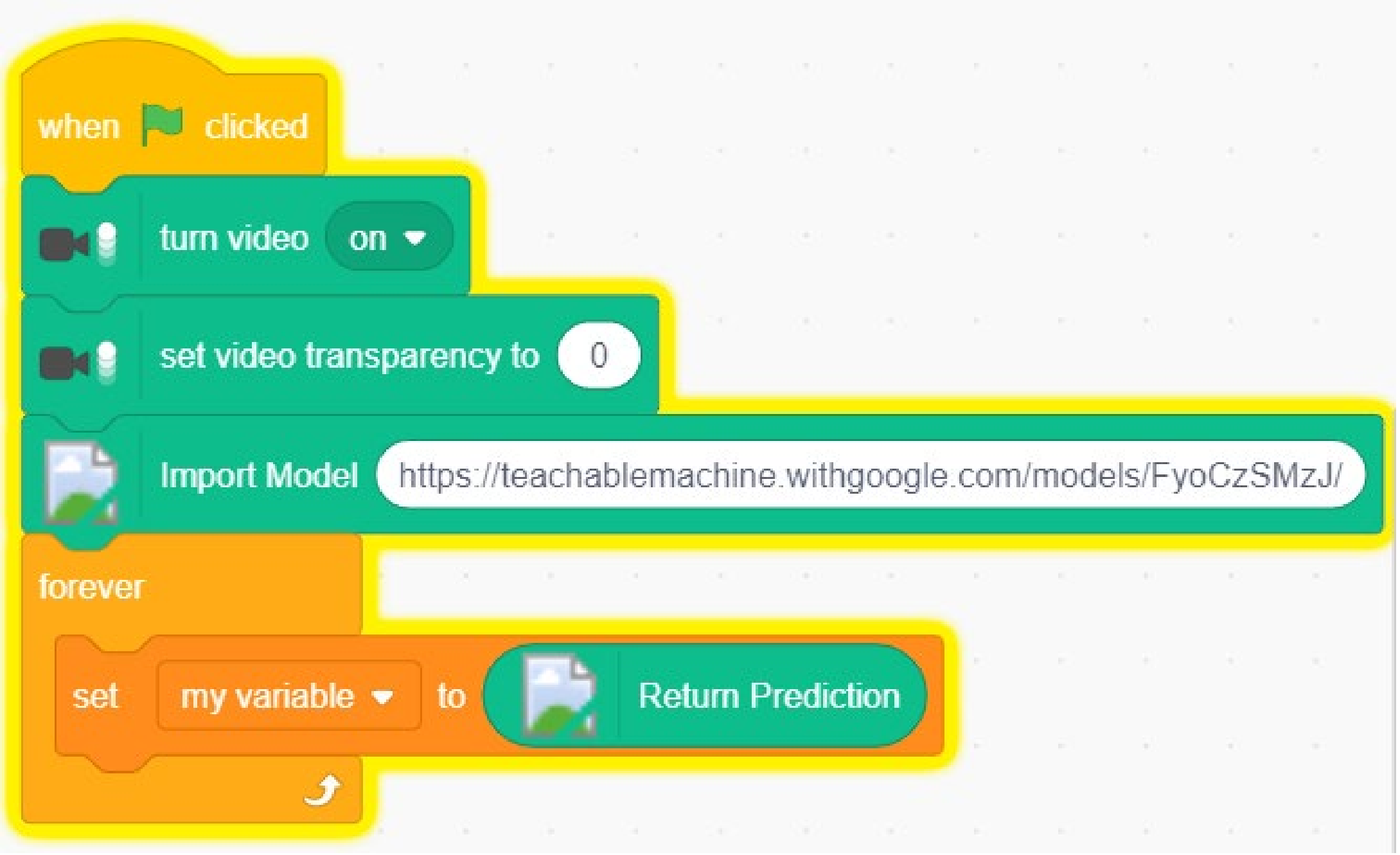
Webcam Input

Import from URL

Trained Model

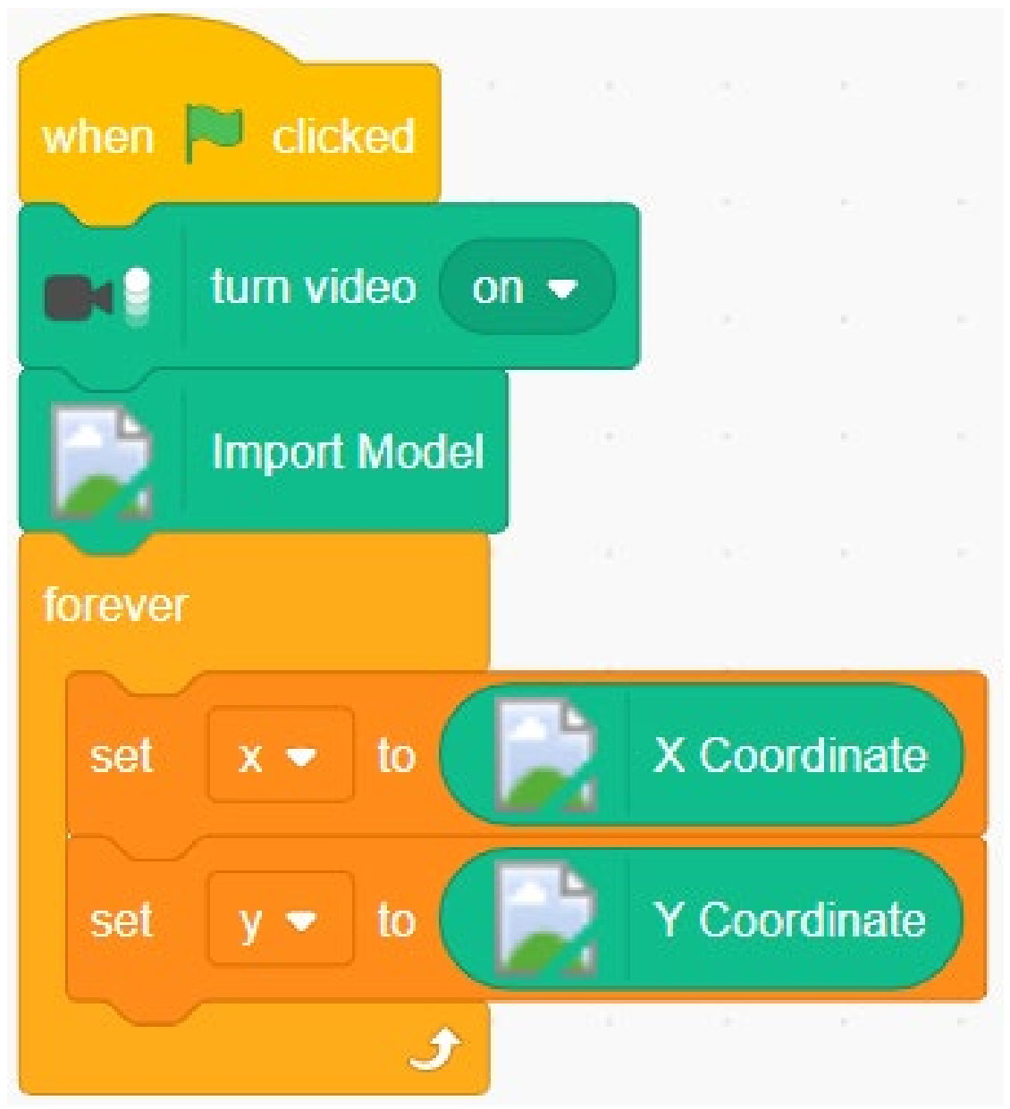
Output Classification

RESULTS



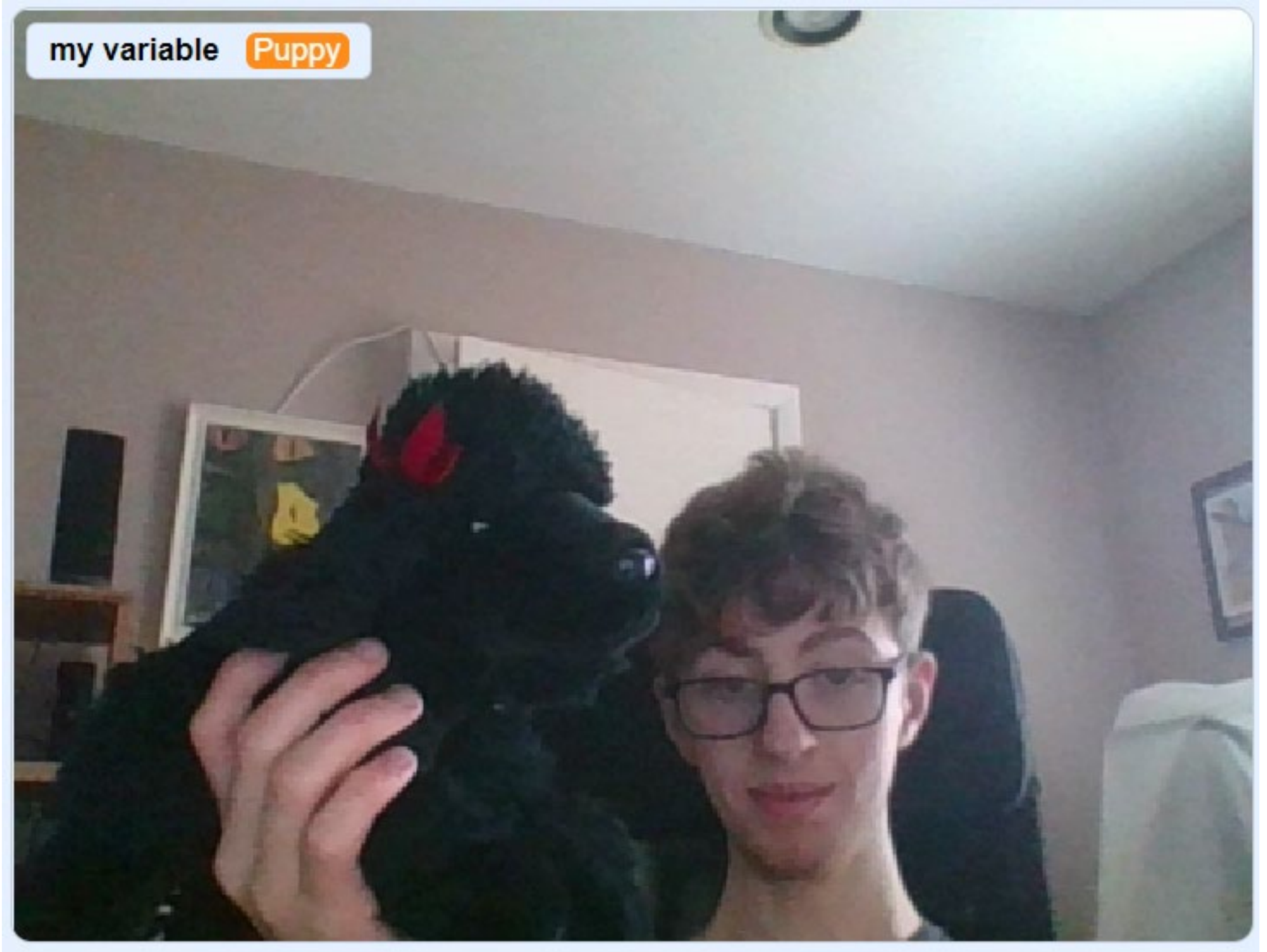
```
when ⚑ clicked
turn video (on ▾)
set video transparency to (0)
Import Model (https://teachablemachine.withgoogle.com/models/FyoCzSMzJ/)
forever
    set [my variable ▾] to (Return Prediction)
↻
```

A basic program using the SEML code. As you can see, a URL is given to the "Import Model" block that determines what it uses to judge the images given to it by the webcam and which result to output. In this case, it is using a Kitten-Puppy recognition model.

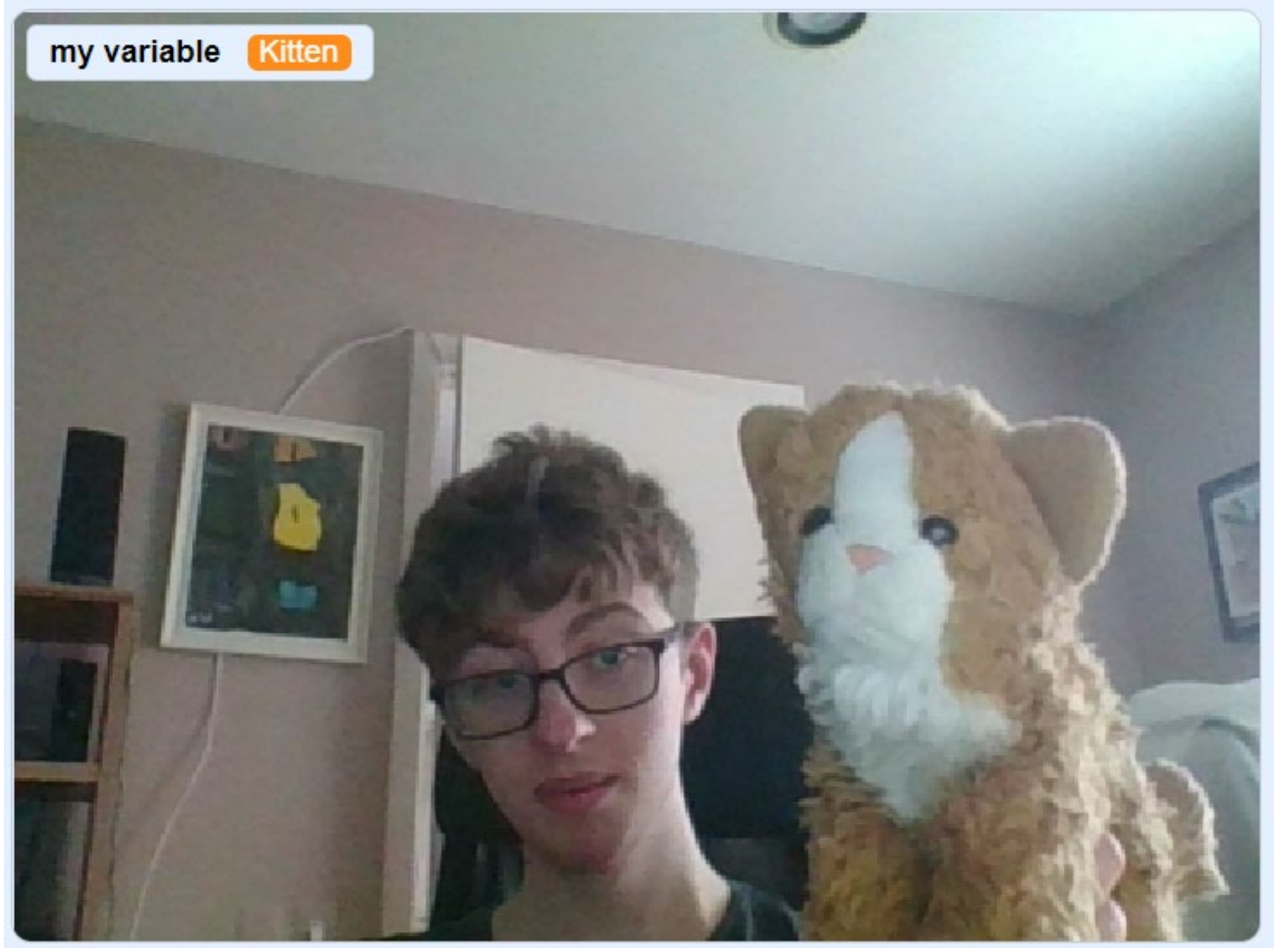


```
when ⚑ clicked
turn video (on ▾)
Import Model
forever
    set [x ▾] to (X Coordinate)
    set [y ▾] to (Y Coordinate)
↻
```

A basic Palm Reader program. The program imports the model from a **fixed** url given in the code, and then sets x and y according to the quadrant where it detects the user's palm.



An example result of the Kitten-Puppy recognition model when a plush puppy is held in front of the camera. In the top left, it is seen that the model correctly identifies that the author is showing it a puppy.



An example result of the Kitten-Puppy recognition model when a plush kitten is held in front of the camera. In the top left, it is seen that the model correctly identifies that the author is showing it a kitten.

DISCUSSION

SEML proves that it is possible to dynamically import trained JavaScript neural network models into Scratch 3.0 for use in a Scratch Program. As Scratch is meant to be an educational software, SEML thus also has educational applications. SEML can be used to teach students and professionals about the basics of neural networks by allowing them to train a model and apply it in a relatable manner. It has been shown that students may learn better through the use of a kinesthetic (hands-on) component [10][11], which SEML would provide if used with a Tensorflow model.

Future development of SEML should take several factors into account. First, the extension does not come with any graphics, which would need to be developed to make the extension visually appealing and easier to understand. Second, SEML so far can only handle image-based models. A more advanced version of SEML should be able to use Teachable Machine audio and pose models as well.