

makeblock

Lesson plan template

Cross curricular links: Developing skills to script block based programs Learning to train an artificial intelligence in a simple way Coding	Age group: 10-13	Duration: 2 hours
--	-------------------------	--------------------------

Objectives: <hr/> Understand what an AI is and use machine learning to identify objects and indicate how to recycle them
Resources: <hr/> <ul style="list-style-type: none">• Desktop or laptop computers every 2 students• powerpoint presentation• mBlock 5 (browser based version or downloaded version)• Extension Teachable machine

Duration 30 min	Mini lesson (introduction): Waste pollution is an important problem today. Separate waste collection can help to overcome the problems related to pollution of the seas, rivers and our planet in general. Introduce the topic of pollution by showing the video in your class: https://www.youtube.com/watch?v=OqHp03RRTDs Guide a short discussion on the topic through pilot questions that stimulate reflection Introduce now the subject of the lesson, using the extension of mBlock 5 Teachable Machines to help us in the separate waste collection
--------------------------------------	---



Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

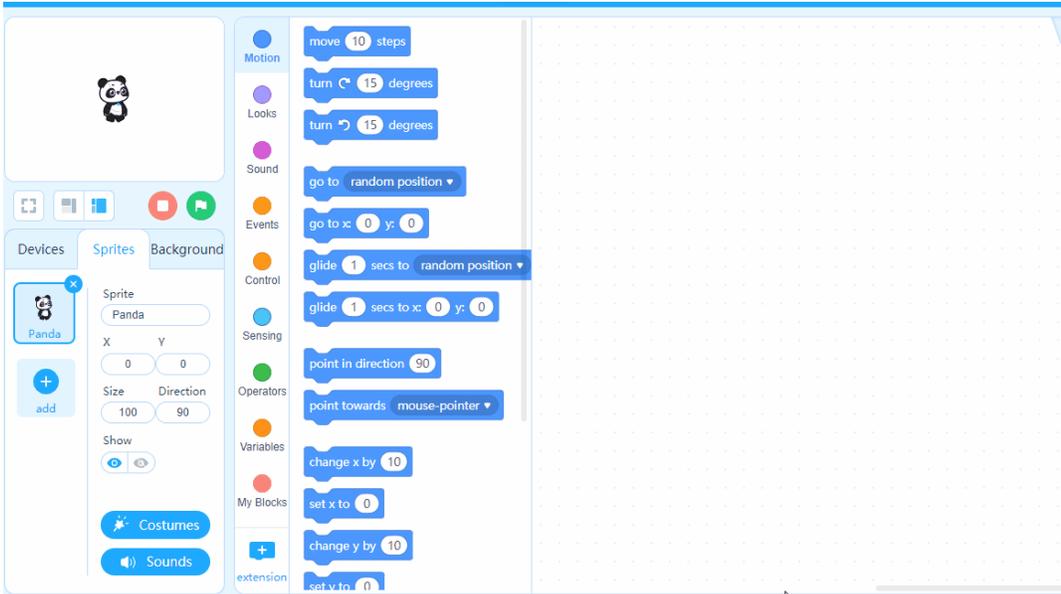
We build a system capable of recognizing between 3 types of waste and show us where to throw it away for recycling.

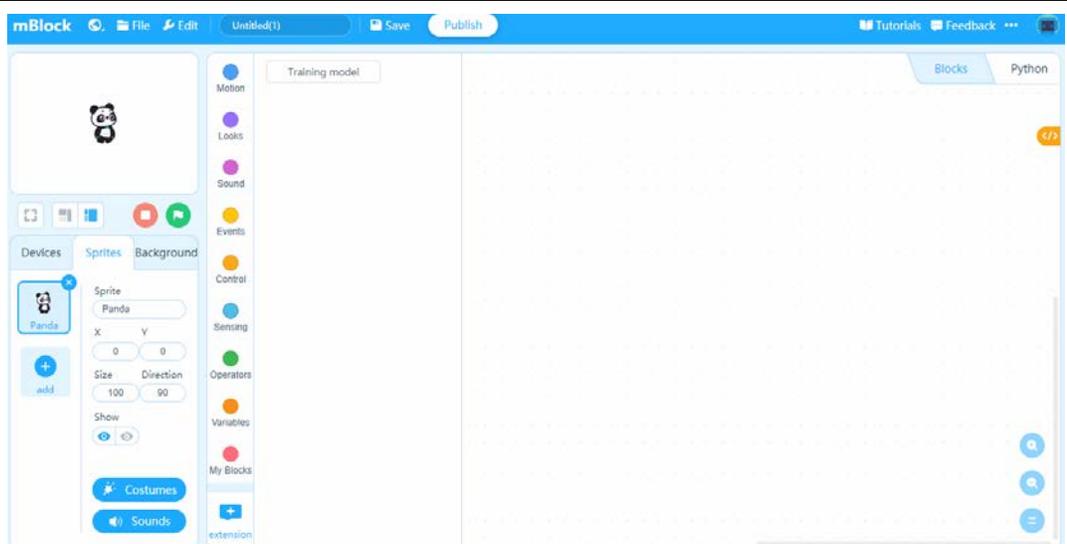
If your students are very good, you can try explaining how learning for machines works:

Teaching a machine is very similar to how you teach people, namely learning through experience. You provide a series of examples that enable the machine to compare until it learned what we wanted him to

More elaborate explanation:

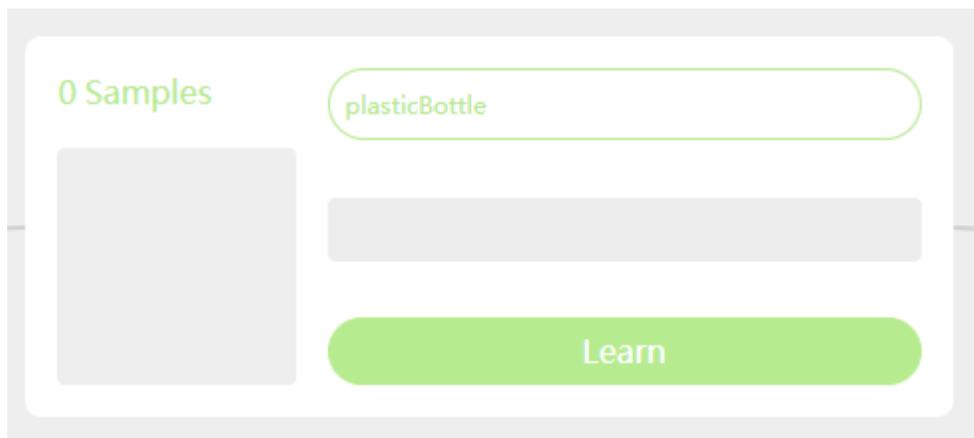
<https://youtu.be/3bJ7RChxMWQ>

<p>Duration</p> <p>1 hour</p>	<p>Main activity:</p> <p>As preparation you need to collect several objects of the same type: plastic bottles, sheets of paper, cans. In the example I have chosen 3 objects from each category, but remember that the more different examples you give, the more precise the recognition will be!</p> <p>We open mBlock 5 and add the extension Teachable Machines:</p>  <p>The first step is the training of our machine. As mentioned the objective is to show as many examples as possible, in order for the computer to derive from enough examples for comparison to the samples that will be presented in the future.</p> <p>Let's open the training model</p>

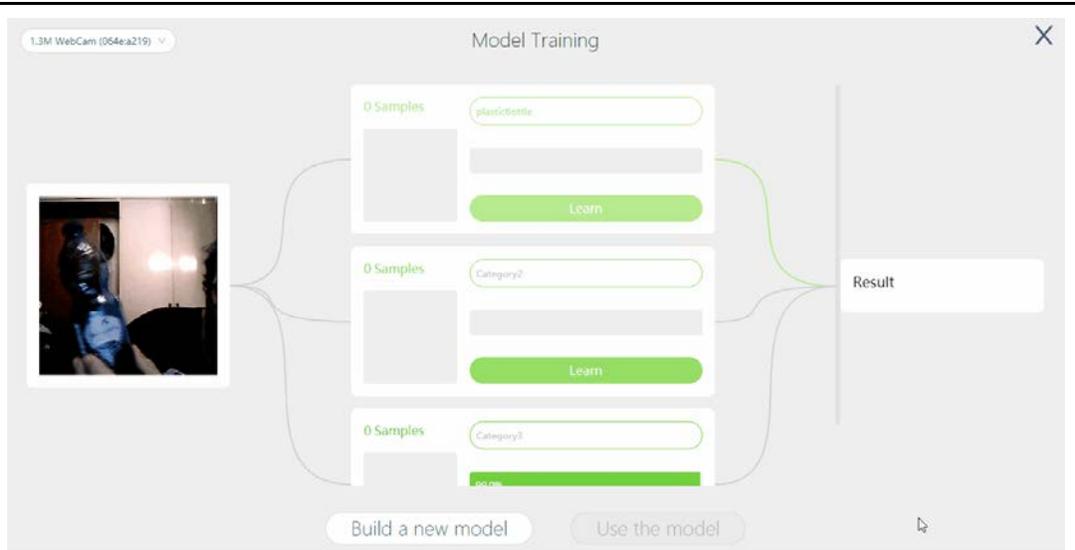


The training consists off placing the object in front of the webcam and saving it by pressing the **"learn"** button

We put the name in the category: **"plasticBottle"**.



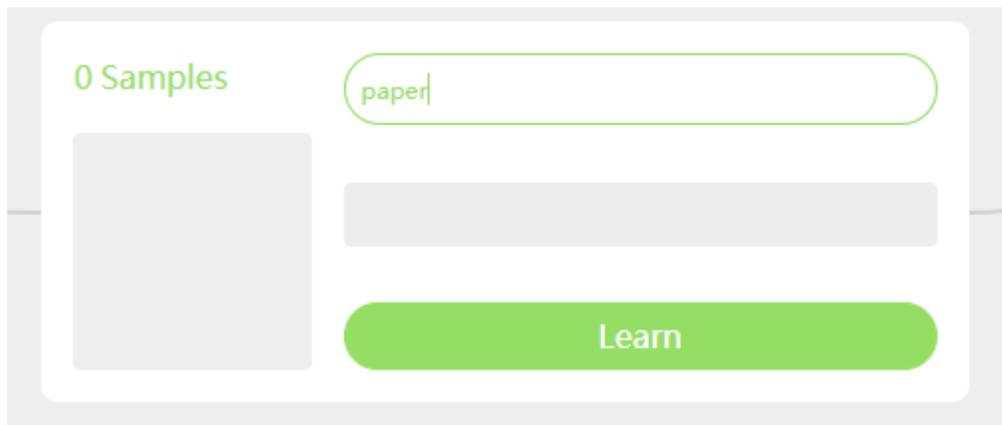
Now we start the training



I've used 100 examples.
Remember, the more examples you use, the more accurate your program will be.



Create the category : "**paper**"



Train the category 'Paper', use different sheets of paper and maybe someone

bundled up in a pile of paper.

The screenshot shows a 'Model Training' window with a camera feed on the left. Three training cards are visible:

- plasticbottle:** 110 Samples, 99.0% accuracy, with a 'Learn' button.
- paper:** 0 Samples, with a 'Learn' button.
- Category3:** 0 Samples, with a 'Learn' button.

A 'Result' box on the right displays 'plasticbottle'. At the bottom, there are buttons for 'Build a new model' and 'Use the model'.

Again, roughly 100 samples were used.

This is a close-up of the 'paper' training card. It shows:

- 101 Samples:** A grid of 12 images of paper rolls with a close icon (X).
- paper:** The category name in a text input field.
- 99.0%:** The accuracy percentage shown in a green bar.
- Learn:** A large green button to train the model.

Create the category : "can"

This is a close-up of the 'can' training card. It shows:

- 0 Samples:** A placeholder for the sample count.
- can:** The category name in a text input field.
- Learn:** A large green button to train the model.

Also here we use different types of aluminium cans to improve the training model

The screenshot shows a 'Model Training' window with a camera feed on the left. The interface is divided into three sections for different classes:

- 0 Samples:** A text input field contains 'can', a progress bar is empty, and a 'Learn' button is visible.
- 101 Samples:** A text input field contains 'paper', a progress bar is filled to 99.0%, and a 'Learn' button is visible.
- 0 Samples:** A text input field is empty, a progress bar is empty, and a 'Learn' button is visible.

At the bottom, there are two buttons: 'Build a new model' and 'Use the model'. On the right side, a 'Result' box displays 'paper'.

I gave 100 examples

The screenshot shows a detailed view of the training process for the 'can' class:

- 123 Samples:** A grid of 12 small images shows various views of aluminium cans. A white 'X' is overlaid on the top-right image.
- can:** The text input field contains the word 'can'.
- 40.0%:** A progress bar is partially filled, indicating 40.0% completion.
- Learn:** A large green button labeled 'Learn' is positioned below the progress bar.

Let's use the model

1.3M WebCam (064ea219) Model Training X

Learn

101 Samples paper 60.0% Learn

101 Samples can 40.0% Learn

Result
paper

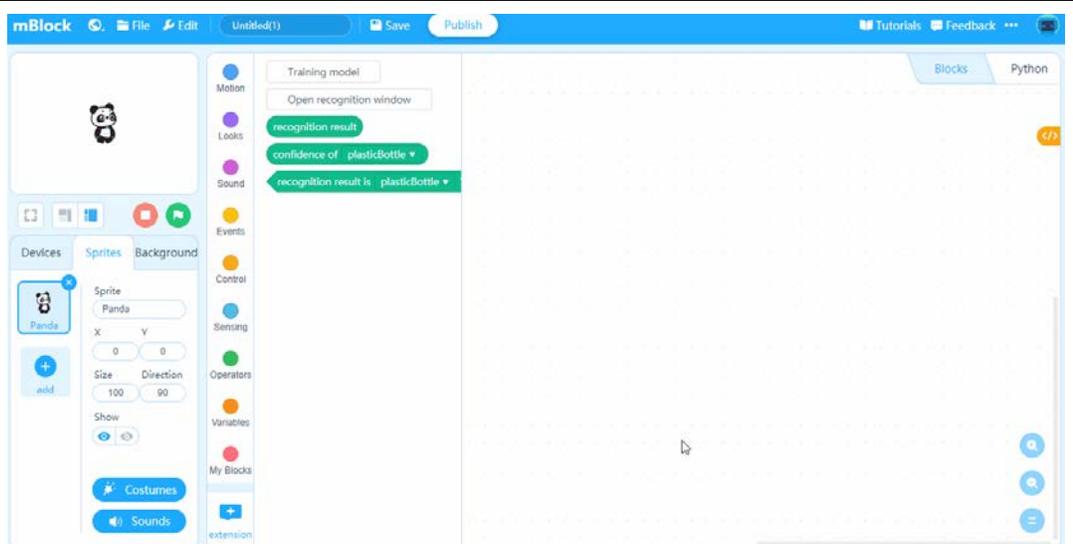
Build a new model Use the model

This is the result

The image shows a Scratch script editor window. On the left is a vertical palette with categories: Looks (purple), Sound (pink), Events (yellow), Control (orange), Sensing (blue), Operators (green), Variables (orange), My Blocks (red), TM (green), and extension (blue with a plus sign). The main workspace contains a script with the following blocks:

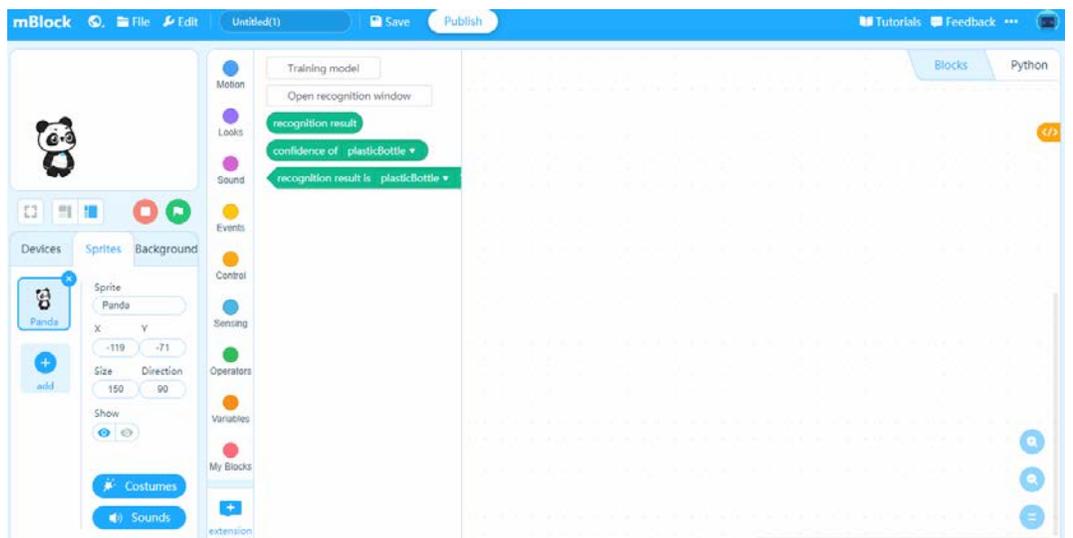
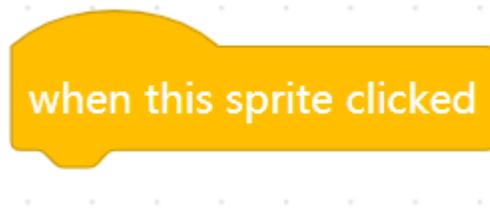
- Training model (white)
- Open recognition window (white)
- recognition result (green)
- confidence of plasticBottle (green)
- recognition result is plasticBottle (green)

Now it's time to use the new blocks created with our sprites, let's increase the size of the panda to 150%.

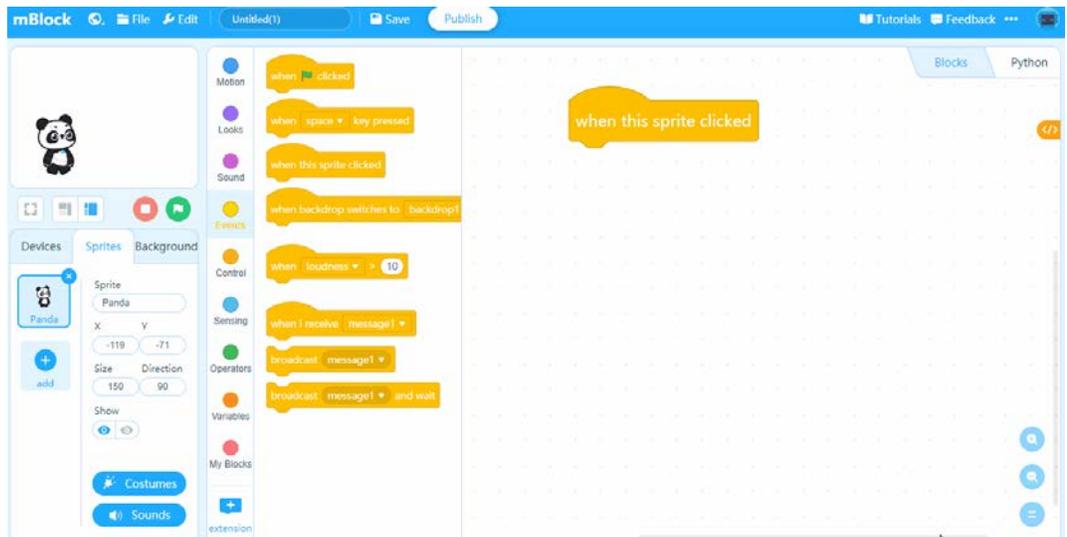


Let's start programming:

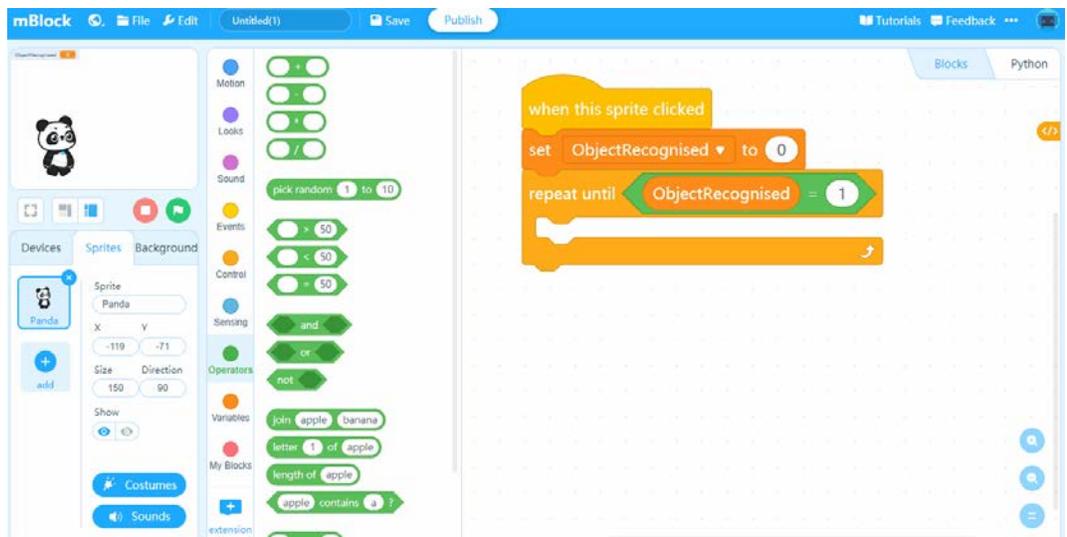
we insert the block "When the sprite is clicked"



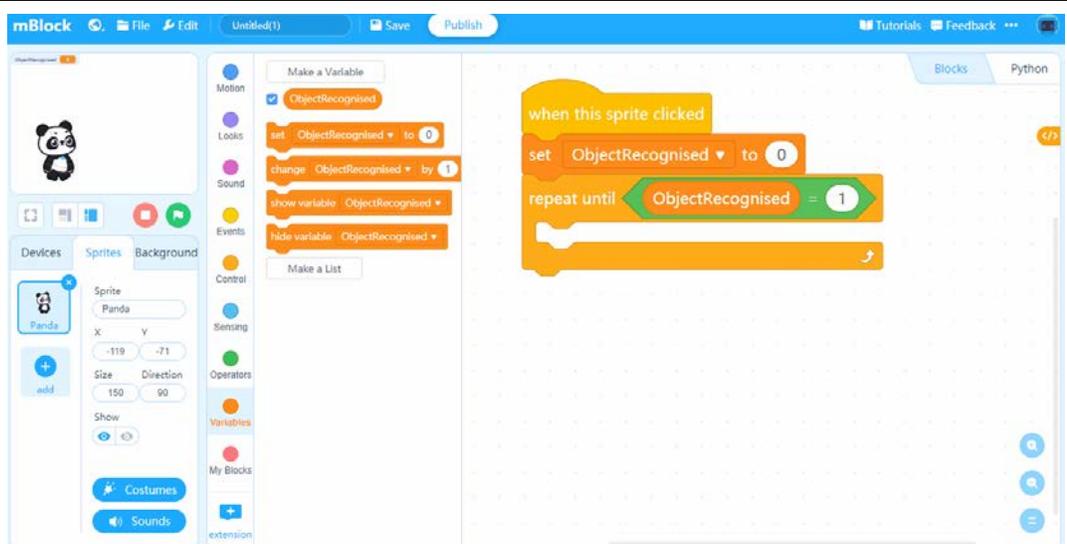
create a "**ObjectRecognised**" control variable that tells us when the object has been recognised and set its value to 0. Its value will become one after recognition has taken place.



Insert the exit condition with 'control' and 'operators'

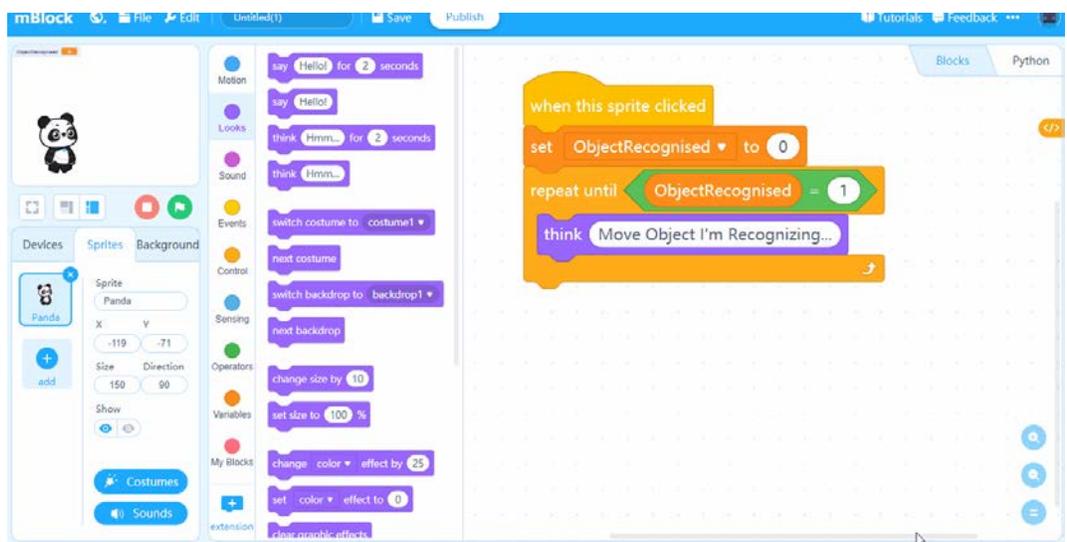
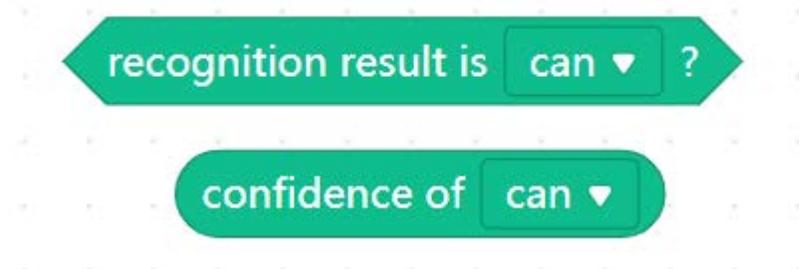


Let's put in a message to see if the recognition has started: something like "**Move Object I'm Recognizing...**"



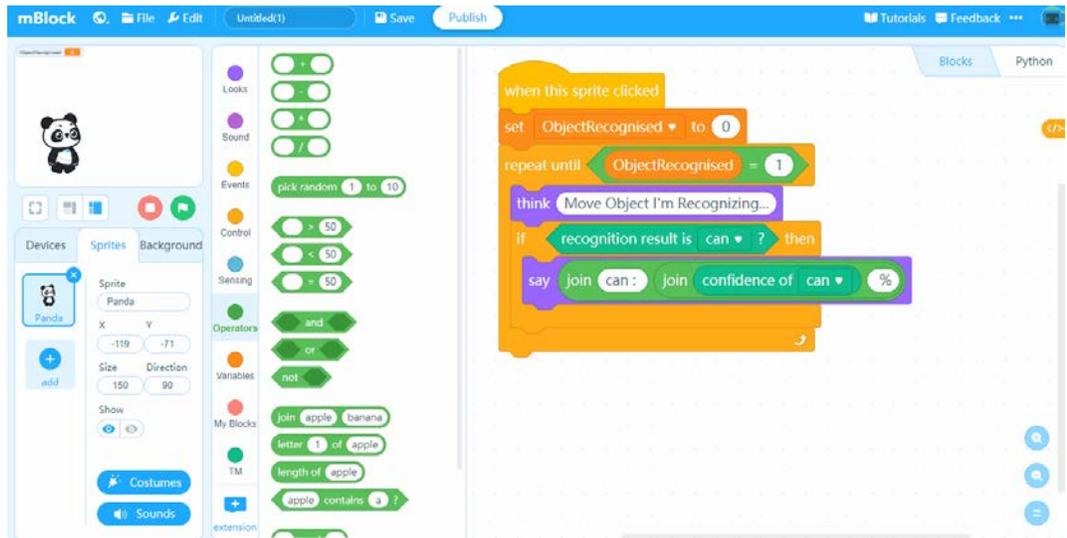
Now let's insert three "If... Then..." blocks to manage the three conditioners. First create the conditioner for the **can**

Use the following blocks:

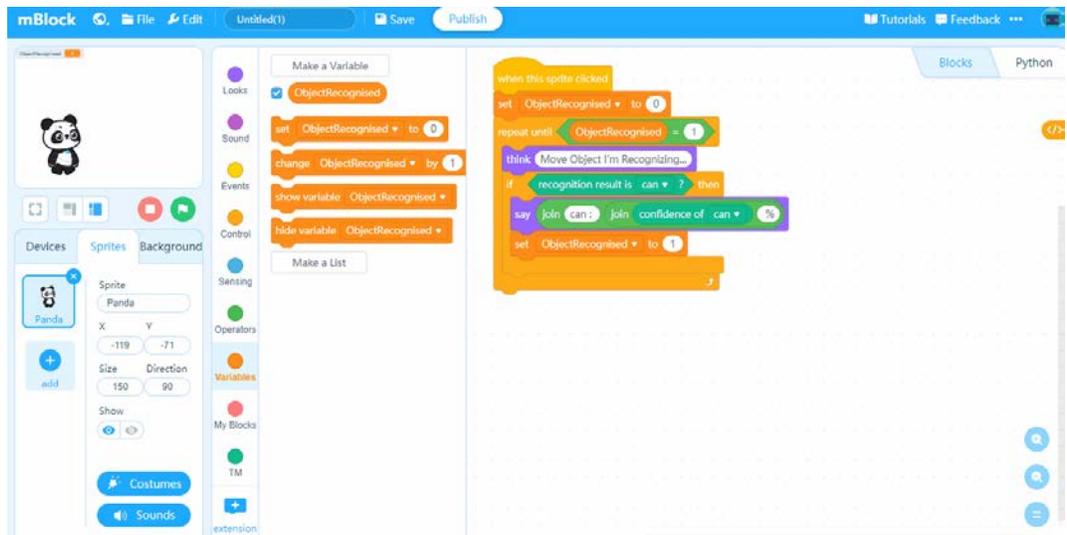
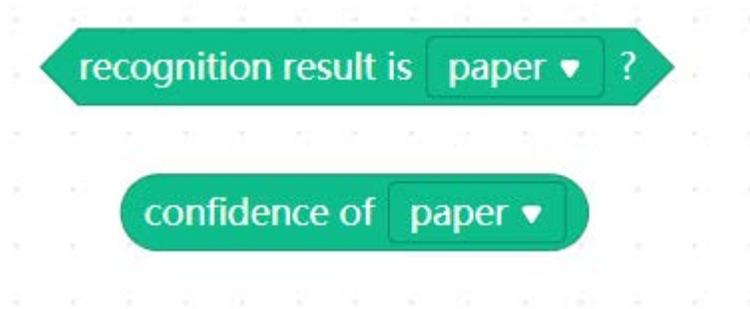


Remember to change the status of the variable "ObjectRecognised" to 1 to exit

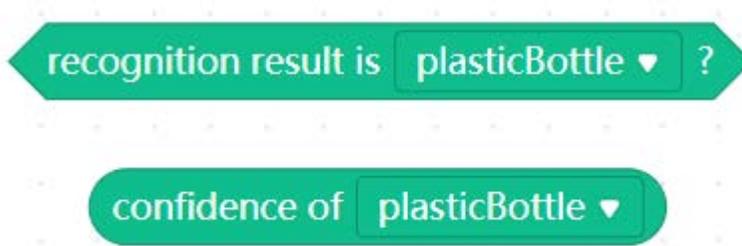
the cycle and interrupt the execution of the program



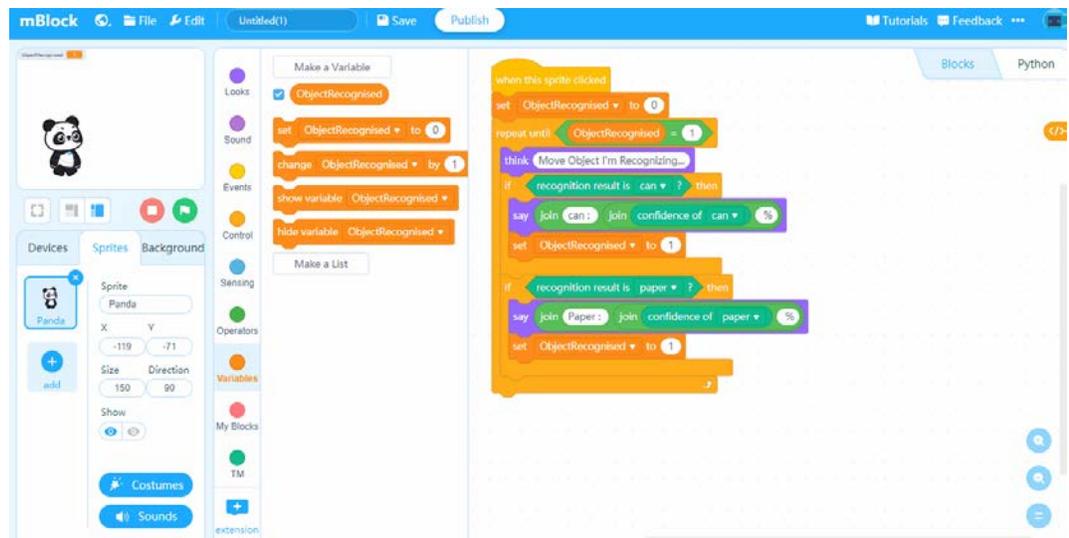
Now let's create the **paper** one, we use the blocks:



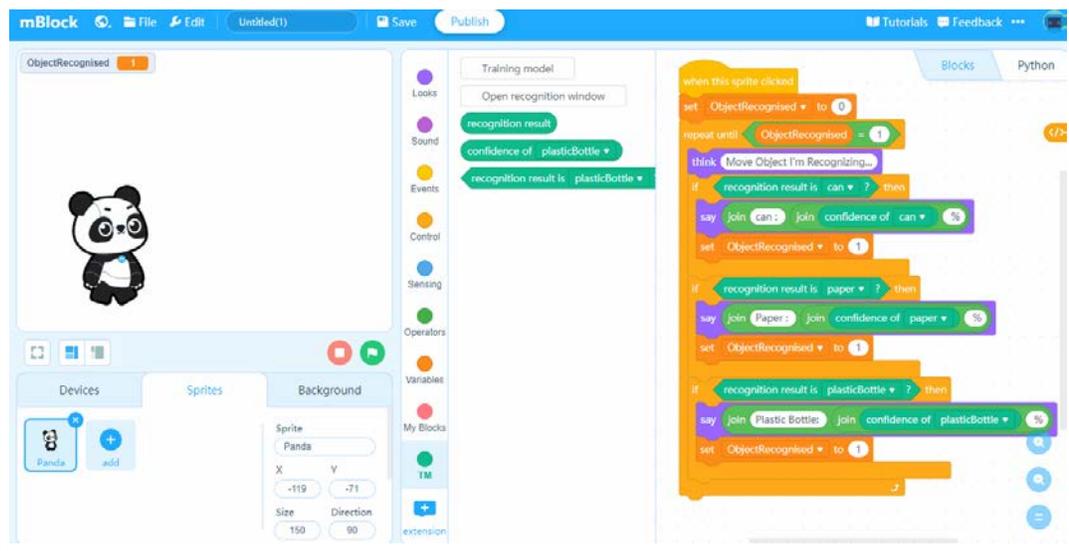
Finally, for the **plastic bottle**, we use the blocks:



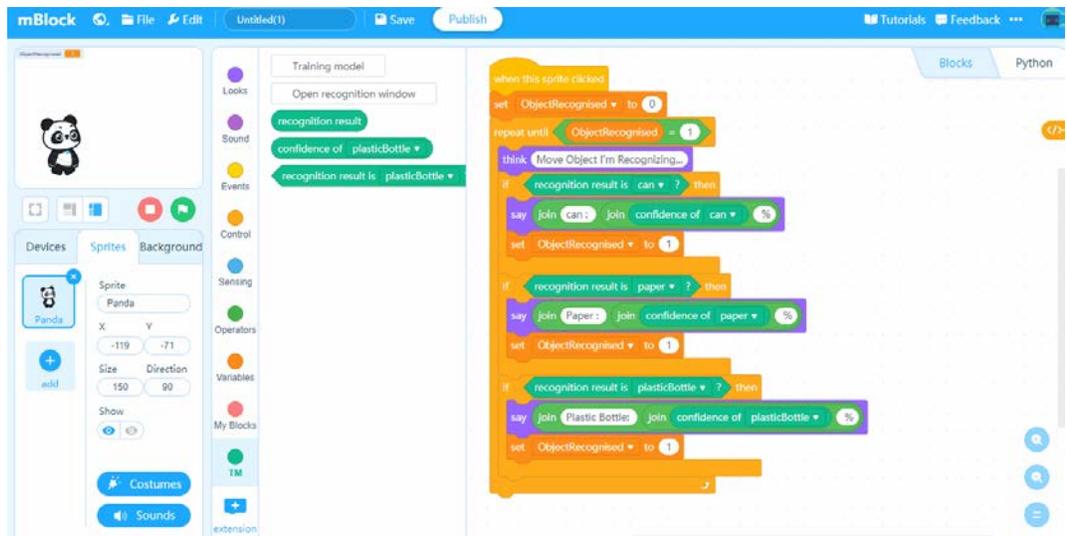
You can duplicate the precedents to do faster



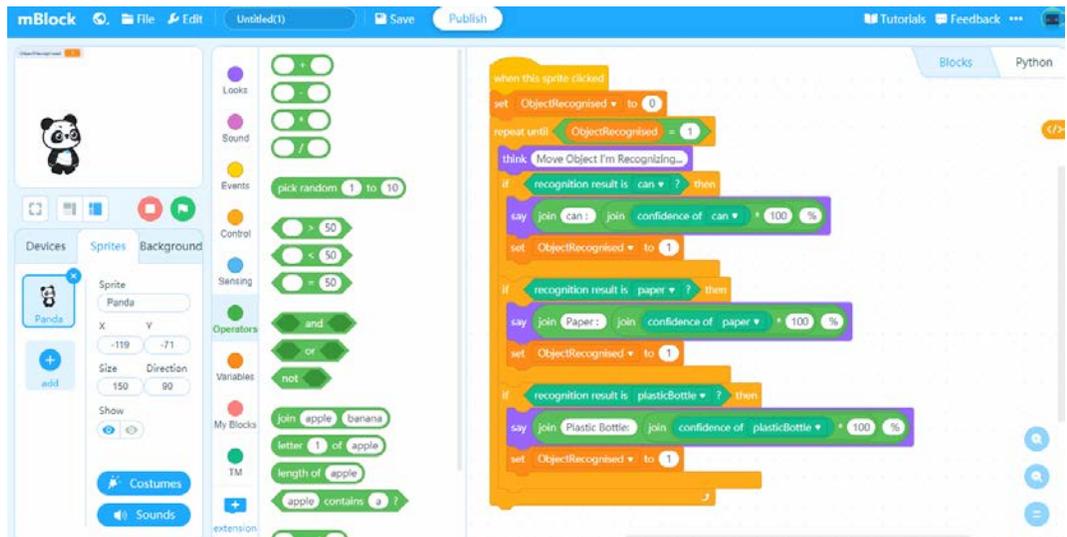
Notice that all the percentages are in decimals.



Multiply the value of the blocks "confidence of ()" by 100 to have the values in percentages.



Insert the background of a park to make everything more green



Test your program: click on the panda to start the recognition

Duration

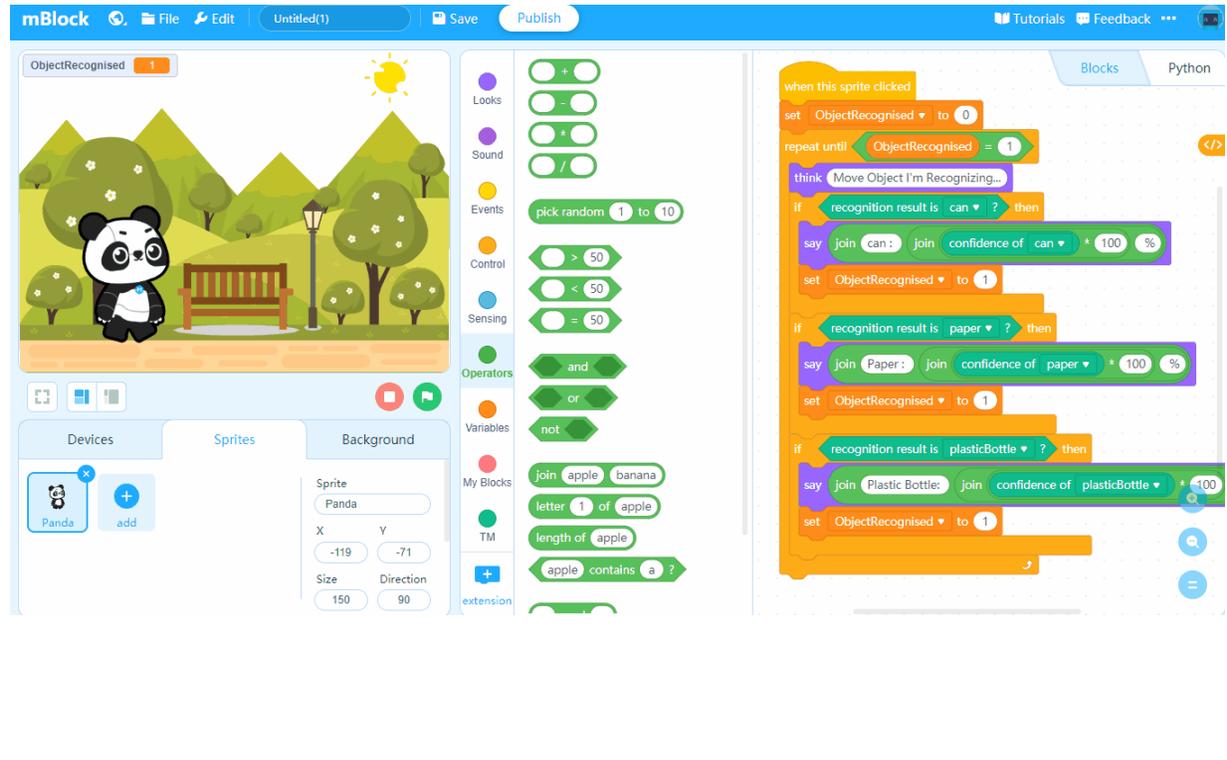
30 min

Conclusion:

Allow students to 'test' other edited scripts by running the script and showing different objects. They should check for:
 Is the recognition correct?
 Does the panda correctly recognize the displayed objects?
 Try with students to decide which will be the next recycling category to add

Notes:

To improve recognition you can at any time add new images or create new categories. Take a look at the example to see how to do it



The screenshot displays the mBlock software interface. On the left, a stage shows a panda sprite in a park setting with a sun, trees, and a bench. The 'Sprites' panel shows the 'Panda' sprite selected. The 'Background' panel shows the 'Panda' sprite's position (X: -119, Y: -71) and size (150). The 'Scripts' panel shows a script for object recognition:

```
when this sprite clicked
  set ObjectRecognised to 0
  repeat until ObjectRecognised = 1
    think Move Object I'm Recognizing...
    if recognition result is can ? then
      say join can: join confidence of can * 100 %
      set ObjectRecognised to 1
    if recognition result is paper ? then
      say join Paper: join confidence of paper * 100 %
      set ObjectRecognised to 1
    if recognition result is plasticBottle ? then
      say join Plastic Bottle: join confidence of plasticBottle * 100 %
      set ObjectRecognised to 1
```

Opportunity to reflect:

Discuss with the students what they found challenging with regards to editing the program

- What would they do differently?
- Can they enter information about how to recycle this type of object after it is indicated what it is?

Can students think of any other simple AI projects that they could create with this type of machine learning?