makeblock education

# Lesson 9:
# mBot2 in the wild

**Subject:** STEAM
**Grade(s):** 5th and up
**Duration:** 45 minutes
**Difficulty:** beginner

## Lesson objectives

*By the end of the lesson, students will be able to:*

- Integrate Machine Learning into programming the mBot2
- Make multiple technical features of the mBot2 work together
- Establish a communication between the robot and the computer, even though the robot is running independently in upload mode for interaction between physical hardware (robot) and sprite stage of mBlock software

## Overview

Machine Learning is a weak form of Artificial Intelligence (AI). Artificial Intelligence refers to devices or machines that mimic human intelligence. Based on collected data, Machine Learning enables computer or robot systems to choose actions based on history of data and probability without the need to explicitly formulate an algorithm for it.

## Focus

*By the end of the lesson, students will know:*

- What Machine Learning is and how to apply it,
- How to make mBot2 work based on conditions,
- How the different technical features of mBot2 work together in an ecosystem,
- Establish a communication protocol between the robot running programs independently and the computer

## Pre-lesson Checklist

What do you need?

- PC or laptop (with USB output) with the mBlock software installed, the web version (also for ChromeBook)
- The mBot2 with a CyberPi
- A USB-C cable or Makeblock Bluetooth dongle
- Webcam, on your laptop or external
- Printer (optional) or paper, pencils and markers for drawing or smartphone to display pictures

## Lesson Plan

This lesson consists of four steps and takes a total of 45 minutes.

| Duration | Contents |
|---|---|
| 5 minutes | 1. Warming up<br>• Machine Learning in everyday life<br>• Getting to know the Machine Learning |
| 25 minutes | 2. Hands-on<br>• Applying Machine Learning<br>• Machine Learning in combination with the mBot2 in mBlock<br>• Establishing a communication protocol from inside running programs |
| 10 minutes | 3. Trying out<br>• Final task: ecosystem of a mouse |
| 5 minutes | 4. Wrap-up<br>• Showtime: show your ecosystem in a fun, short movie for later discussion.<br>• If your teacher allows, share the end result on social media with the hashtag #mBot2<br>• Reflection: What are you most proud of? What would you like to improve about your robot? |

makeblock education

Activities

# 1. Warming up (5 min)

**Step 1: Warming up**

This step consists of three parts:

1. Machine Learning in everyday life
2. Getting to know the Machine Learning extension in mBlock
3. Applying Machine Learning

**1. Machine Learning in everyday life**

Machine Learning works on a statistical basis: training data is provided to the algorithm, like pictures of different objects plus their descriptions. The learning phase then tries to distinguish best between these pictures and match each picture to the correct category. After this phase, the learned patterns are tested against new sets, and the learning can be reinforced. The difference to "classical" sorting programs is that there is no clear indication beforehand on how to decide between the different categories, like you can differentiate between colors.

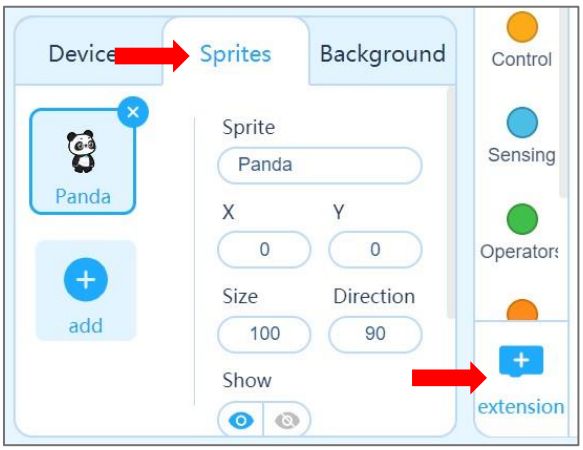
By "learning from past data" the computer program can better adjust to changing situations. Think about facial recognition on your phone: the first times after setting this up, it is harder for the phone to correctly identify you, but over time, the algorithm "learns" to identify your face in different environments as well. Machine Learning is broadly used in image classification (e.g., on social media) to automatically analyze what pictures uploaded show: a cat, a dog, furniture, people etc.

Machine Learning is considered as "weak Artificial Intelligence", since the computer program is not aware of itself nor can "think" like we do – but its current behavior is influenced by data collected and analyzed.

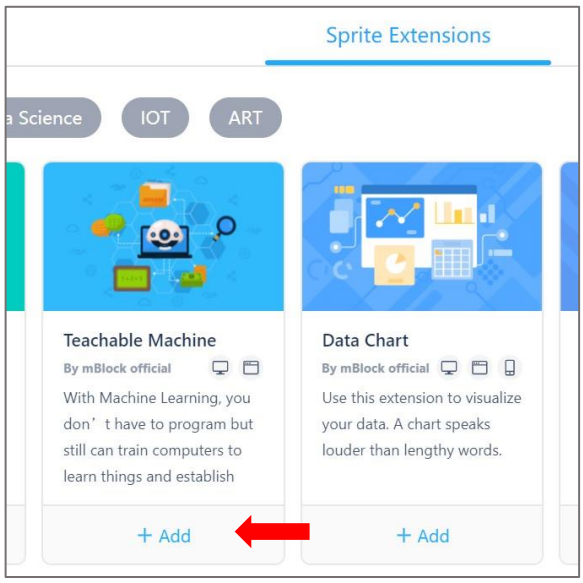
## 2. Getting to know the Machine Learning extension in mBlock

In mBlock there is a Machine Learning tool. You need to add this tool from the extension library. Proceed as follows:

- Click on 'Sprites'.
- Click on 'extensions' (at the bottom in the middle of the screen).




- You will now enter the extension library. At the top of the screen, click 'Sprite Extensions'.
- Find the extension 'Teachable Machine' and add it.




- The extension will now be added to the code blocks. You can recognize the extension by the abbreviation TM in green.
- Make sure you have a working webcam.
- Click on the 'Training model' button.


The extension consists of a training model in which you can 'teach' to differentiate between sets of pictures taken with a webcam. The basic model you see consists of 3 categories. If you want more, click on "build a new model" and choose the number of categories.

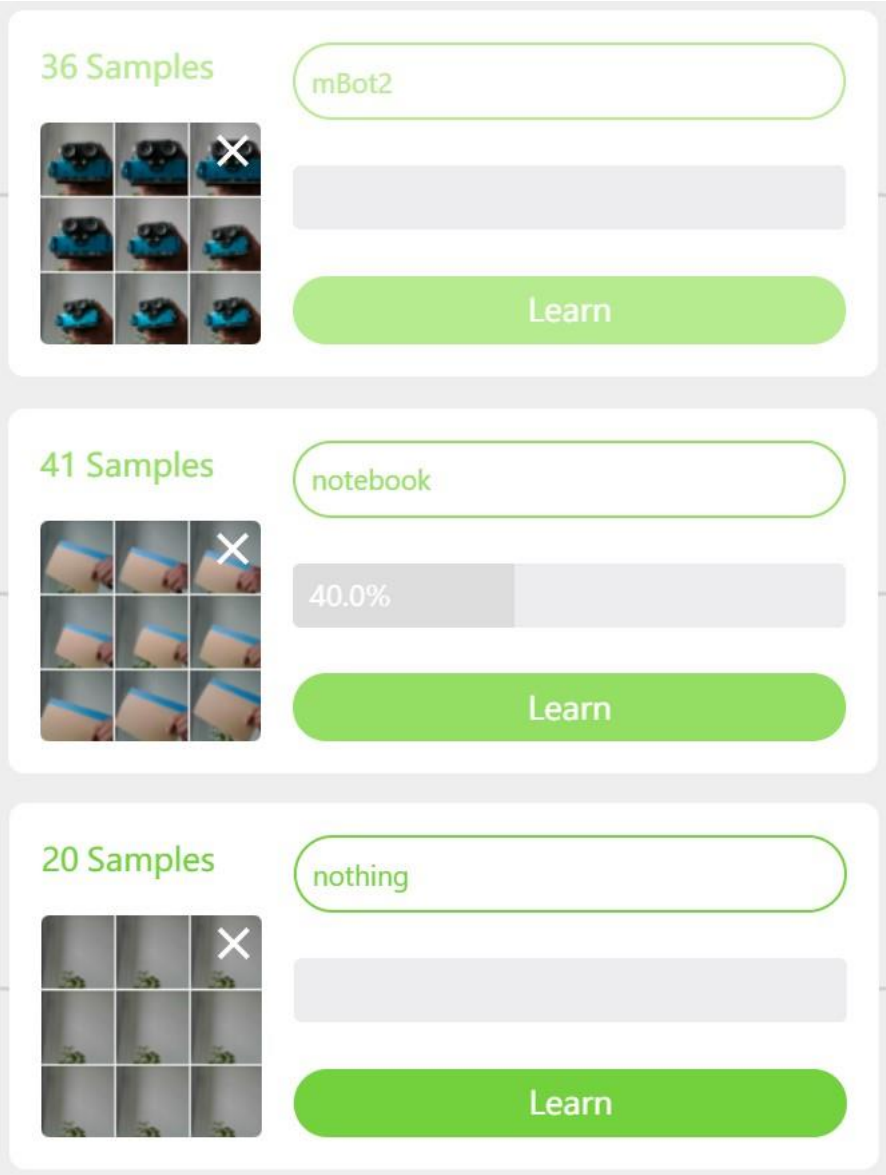
Some advice:

- Think about the number of categories and the recognition beforehand. You cannot add a category later, you would need to start all over again.
- Choose 1 extra category to be used as "background". So, if you want to recognize 2 different animals by pictures you hold in front of the webcam, choose 3 categories: The third one will be the one where there is no picture of an animal shown. Otherwise, you "force" the model to decide e.g., between cat and dog, even if the webcam shows the empty background.
- Background should generally be simple and static (not: classroom with pupils, webcam looking at the pupils etc.) with a clear contrast to the objects to be recognized.

The images will be analyzed just on the computer and the result can be stored. The pictures themselves will not be kept! So, if you save the program file in mBlock and reopen it later, you won't see the original pictures the model was trained on. You can always add new pictures to existing categories, but not delete single ones (only all pictures and learnings from a category).

You can directly 'teach' the categories using images taken with the camera. You will use the standard 3 categories for this exercise:

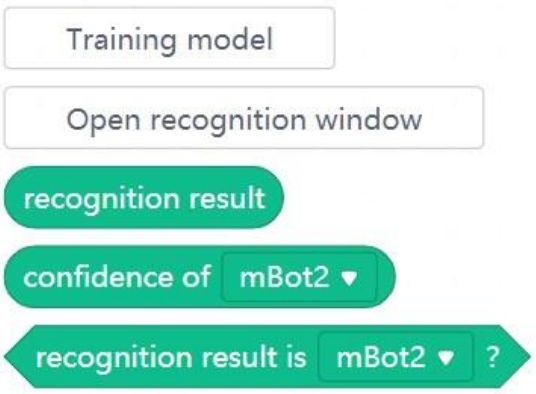
1. You can start with the mBot2. Place it on the desk and align the camera to get a good view – or hold it steady in your hand if you use a laptop with a built-in camera. Then click the 'Learn' button. Hold the button until the camera has recorded at least 20 samples. Above the mosaic of samples, you can see the number of samples stored for this category. Give this category a name, for example 'mBot2'.
2. Now choose something else, e.g., a notebook, hold it in front of the camera and click on the 'Learn' button. Hold the button until the camera has recorded at least 20 samples. Give this category a different name, for example "notebook".
3. Now record the "background". This should be used whenever none of the other objects are presented to the camera. Click and hold the 'Learn' button until the camera has recorded at least 20 samples. Give this category a different name, for example "background".




You have now 'trained' the Machine Learning different objects (mBot2, notebook and background) that the algorithm will now recognize in the camera picture. For every category you can see a bar measuring a percentage. This represents the confidence that the algorithm has in the recognition result. You can increase the percentage of confidence by increasing the number of samples and doing some slight variations like changing the angle of the object. This will help the camera recognize the objects even better and faster.

You can test the recognition live without the need for any coding at this stage. Add pictures until the recognition is fast and accurate (not switching between results and showing a high percentage >90%). Are you satisfied? Then click on 'Use the model'. You will now return to the screen for coding on sprite level.

There are three new code blocks now. One block will report the recognition result (as you typed it), the second one is the confidence for a given category and the third one is the Boolean block for conditional statements. It lets you choose the category by a drop-down menu. If you click on one of the code blocks, a new window will appear. This is the recognition window, which displays the camera image and the recognition result. Now your webcam functions as a smart sensor, identifying different objects or pictures.




## 2. Hands-on (25 min)

**Step 2: Hands-on**

This step consists of three parts:

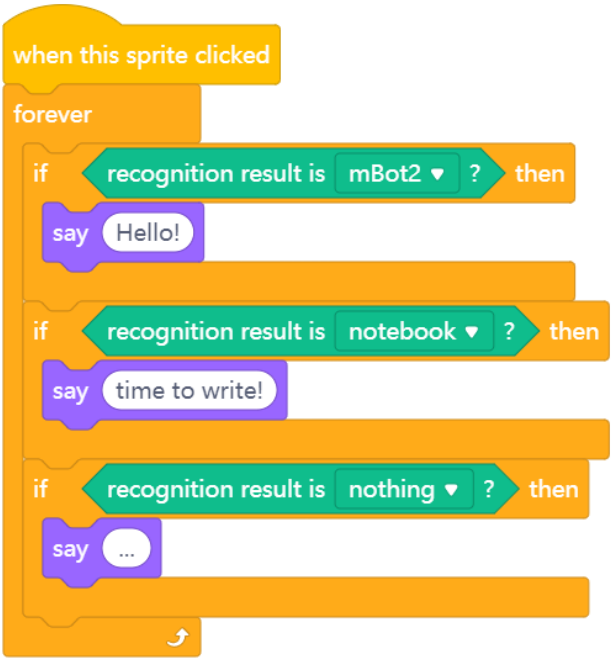
1. Applying Machine Learning
2. Machine Learning in combination with the mBot2
3. Establishing a LAN connection

**1. Applying Machine Learning**

You have 'taught' two different objects, the mBot2 and a notebook, plus the background to the Machine Learning model. Now you can use the output of the recognition like any other sensor blocks in your program code. You can use conditional statements to differentiate between the objects and assign a certain action separately to each result.

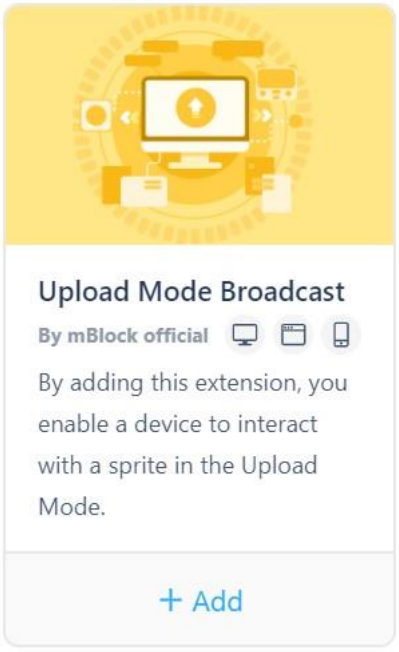
You are now going to try to make the sprite (in this case the panda) talk when the webcam recognizes a learned object.

On the right you see a programming example. Reproduce the programming example and see what the panda does when the webcam recognizes your learned objects.
Was it successful? Then investigate what other possibilities there are. Can you make the panda jump or walk?




**2. Machine Learning in combination with the mBot2**

Now you know how to make use of the Machine Learning extension for image recognition and control a sprite with it, for example the panda. You are now going to learn how to communicate from the computer to the mBot2 running its own program. For this the mBot2 needs to stay "connected" to the computer – this is done via USB cable or the Makeblock Bluetooth Dongle. The dongle allows the robot to move independently from the computer; if you use a USB-cable, make sure you only perform slight movements of the mBot2, like small turns and distances.
With this connection in place, the sprite (in this case the panda) can send data to the mBot2. The mBot2 receives this data and the program running on the robot can react to it.
You need to add a new extension to the code blocks for the sprite and the mBot2, called "Upload Mode Broadcast".

| | Extension | Add |
|---|---|---|
| For the sprite |  | • Click on 'Sprites'<br>• Click on 'Extensions'<br>• Click on 'Sprite extensions'<br>• Find the extension labelled "Upload Mode Broadcast" and add it |
| For mBot2 | | • Click on 'Devices'<br>• Click on 'Extensions'<br>• Click on 'Device extensions'<br>• Find the extension labelled "Upload Mode Broadcast" and add it (might not be on the first page of the Extensions library) |

In this way, the extension will be added both for the mBot2 and the sprite. In mBlock it appears in dark blue. Click on the new extension and see the different code blocks associated with it. Basically, the extension adds a communication protocol. It allows you to send and receive messages between the sprite and the robot while the robot is in Upload mode.

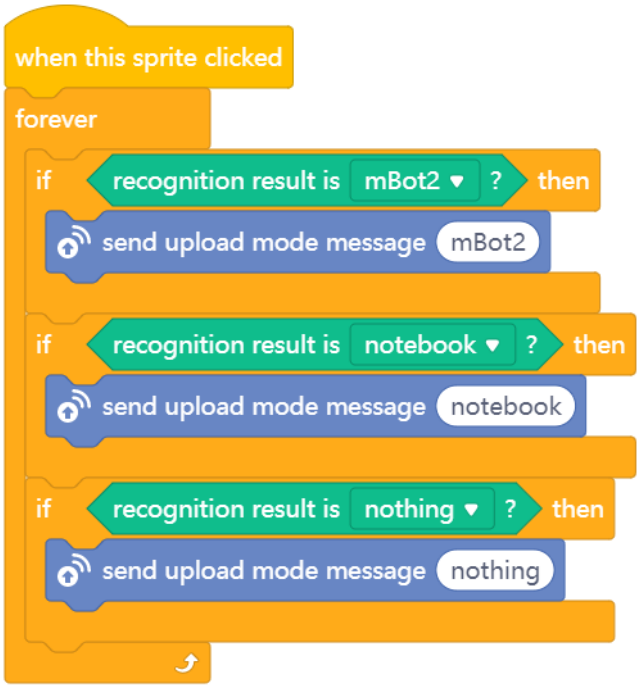
**3. Establishing the connection with the mBot2**

You have now installed all the necessary extensions for the hardware and software to be able to communicate with each other. Now you will program the mBot2 to be controlled by Machine Learning. The idea is that the mBot2 will perform actions based on recognition results. You programmed a similar task earlier just for the sprite (mBot2, notebook, and background).

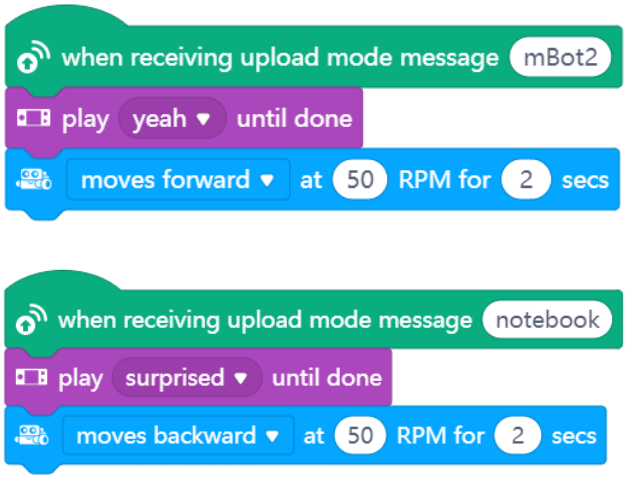
Now, the recognition results must be sent and received over the connection between the Bot2 and the computer (either cable or wirelessly via Bluetooth Dongle). The receiving code uses events, so the corresponding code is executed, once the message is received. With events there is no need to check for a sensor or message in a loop, making the code more efficient and easier to understand.

You need to create two computer programs – one for the device (mBot2) and one for the Sprite that uses the Machine Learning add-on:

*Sprite program*




*mBot2 program*




After reprogramming the above code blocks, make sure your computer stays connected to the mBot2 if you use a cable (don't disconnect the cable). Ideally, they would be connected via Bluetooth using the Makeblock Bluetooth Dongle, because this gives the mBot2 more flexibility in driving around.

| | |
|---|---|
| If you use the dongle:<br>  | Install the dongle on the computer and turn on the mBot2. You need to pair the mBot2 with the Bluetooth Dongle. You do this by clicking on the button on the dongle. The LED will start flashing and the dongle tries to connect to the closest Makeblock controller (like mBot2 and others). In a classroom situation, make sure this is done team by team, to ensure the corresponding match between Dongle and mBot2. Once the pairing is done, you will hear a beeping sound from the mBot2. This means that the computer and the mBot2 have made a wireless connection. |

The Makeblock Bluetooth Dongle will remember the last paired device and instantly reconnects.

In case you don't have the Bluetooth dongle from Makeblock, you can verify the compatibility of the Bluetooth in your computer via the following link: [https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility](https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility)

Now switch mBlock to Upload Mode and connect the mBot2 to mBlock. Click the 'Connect' button and upload the computer program. You can now start testing!

makeblock education

# 3. Trying out (25 min)

**Step 3: Trying out**

You have all the knowledge and skills to use the mBot2 in any situation. Now you are going to work with the ecosystem of a little mouse. A mouse lives in nature where it must look for food and water. There are also many dangers, such as predators who feed on mice.

In this assignment, you will program the mBot2 to act as a little mouse that searches for food and flees from predators.

Find a picture of a flying eagle and some corn. The corn is the food of the little mouse. The eagle is the hunter. You can print these images or use your smartphone to display them to the webcam. You can also hand-draw pictures for food and predator if you like. Just use the same method you decided for 'teaching' them to your algorithm using the Machine Learning extension and for testing the program.

Program the mBot2 so that if you show the corn to the camera, the mBot2 will move forward (if you use the cable, make a small movement), but if you show the image of the eagle, it moves backwards.

You can also add more code: if the little mouse sees the food, it says "yummy" and when it sees the eagle, we can hear its heartbeat.

When creating the computer program for this assignment, it is helpful to use the following step-by-step plan.

| | **Explanation** |
|---|---|
| Step 1: What do you want to do? | • What do you want to program? |
| Step 2: What do you need? | • What do you need in addition to the mBot2? |
| Step 3: What code blocks do you need? | • How are you going to make the mBot2 move?<br>• What code blocks will you use?<br>• Make a brief description on how your program works (pseudocode/natural language, flowchart or UML)<br>• If you need further explanation, you can discuss with your fellow students, the teacher, or do a research on the topic. There is help available for every coding block in mBlock as well. |

| | |
|---|---|
| Step 4: Testing and implementation | • First version ready? Test it! During the testing round, write down points of improvement<br>• Work on the improvement points until the mBot2 does exactly what you had in mind<br>• Successful? Film the end result and ask your teacher if you can post it on social media with the hashtag #mbot2 |

## 4. Wrap-up (5 min)

**Step 4: Wrap-up**

Were you able to recreate the mouse ecosystem?

In this lesson, you learned about Machine Learning and how to apply it. You also know how to use a communication (and why preferably wireless) between a sprite and the mBot2, and to make use of it to exchange data between the two. In addition, you have made various technical features of the mBot2 work together.

It is now time for a brief reflection. Think on your own and discuss with the group:

- What do you think turned out well?
- What could be even better?
- Which parts of the lesson did you find easy and which did you find more difficult?
- What would you like more explanation about?
- Who could help you with that?