



# Fooled

In this project you will learn about how machine learning can make mistakes.

You'll train a machine learning model to recognise pictures of apples and tomatoes.

Then you'll see how you fool it so that it confuses the two.

The image shows a Scratch project interface. The stage features a red tomato sprite. A script is attached to it:

```
when I start as a clone
  go to x: 170 y: -110
  show
  check model [ML recognise image image (label)]
  set size to [20 %]

```

The script uses the `check model` block from the `ML` category. Below the stage, a `define` block is defined:

```
define [check model <recognized>]
  if [recognition = APPLE then
    
```

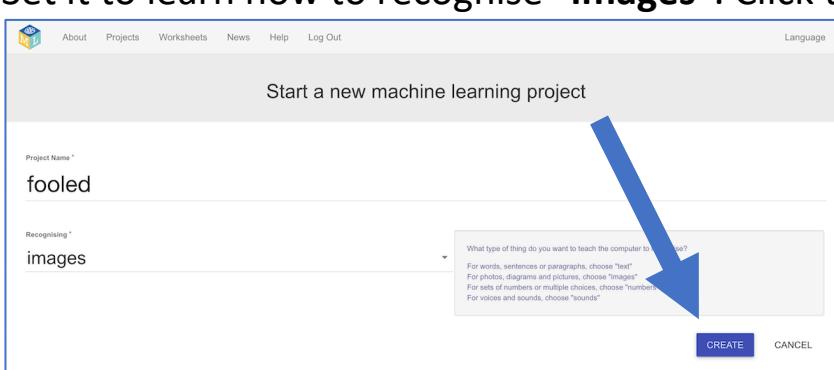
The `ML` category in the blocks palette contains the following blocks:

- costume image
- backdrop image
- save screenshot to costume
- fooled
- recognise image image (label)
- recognise image image (confidence)
- apple
- tomato
- add training data image apple

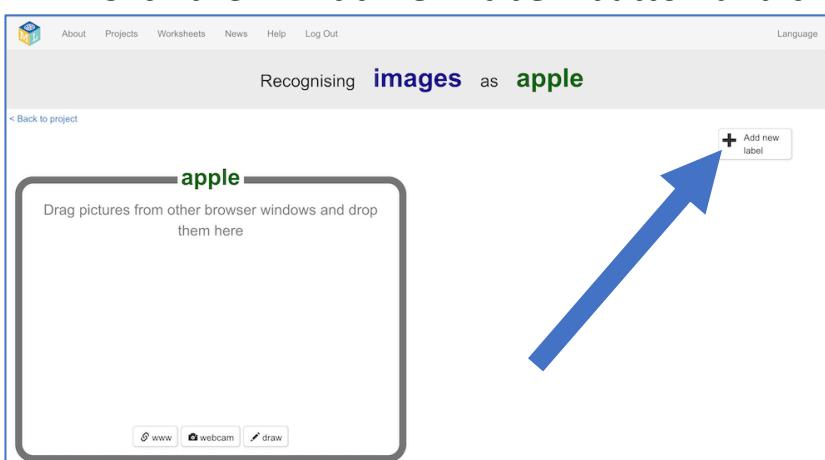


This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License  
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

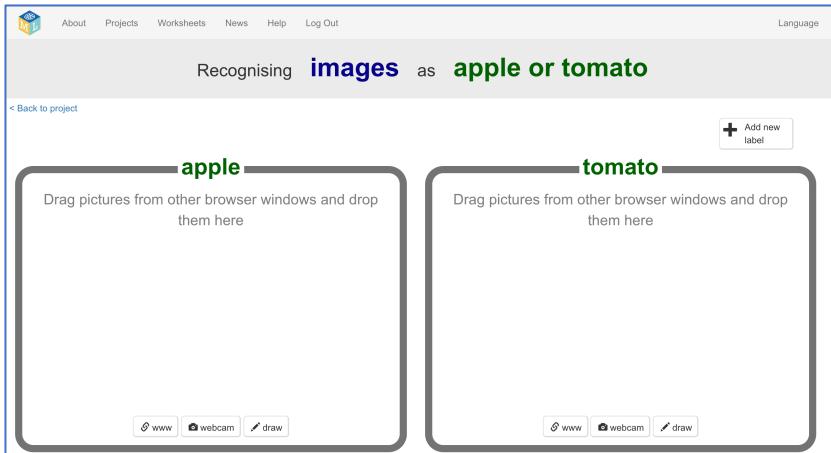
- 1.** Go to <https://machinelearningforkids.co.uk/> in a web browser
- 2.** Click on “**Log In**” and type in your username and password  
*If you don't have a username, ask your teacher to create one for you.  
If you can't remember your password, ask your teacher to reset it for you.*
- 3.** Click on “**Projects**” on the top menu bar
- 4.** Click the “**+ Add a new project**” button.
- 5.** Name your project “**fooled**”.  
Set it to learn how to recognise “**images**”. Click the “**Create**” button



- 6.** You should see “**fooled**” in the list of your projects. Click on it.
- 7.** Click the **Train** button.
- 8.** Click the “**+ Add new label**” button and create a label called “apple”



## 9. Click “+ Add new label” again and create a label called “tomato”

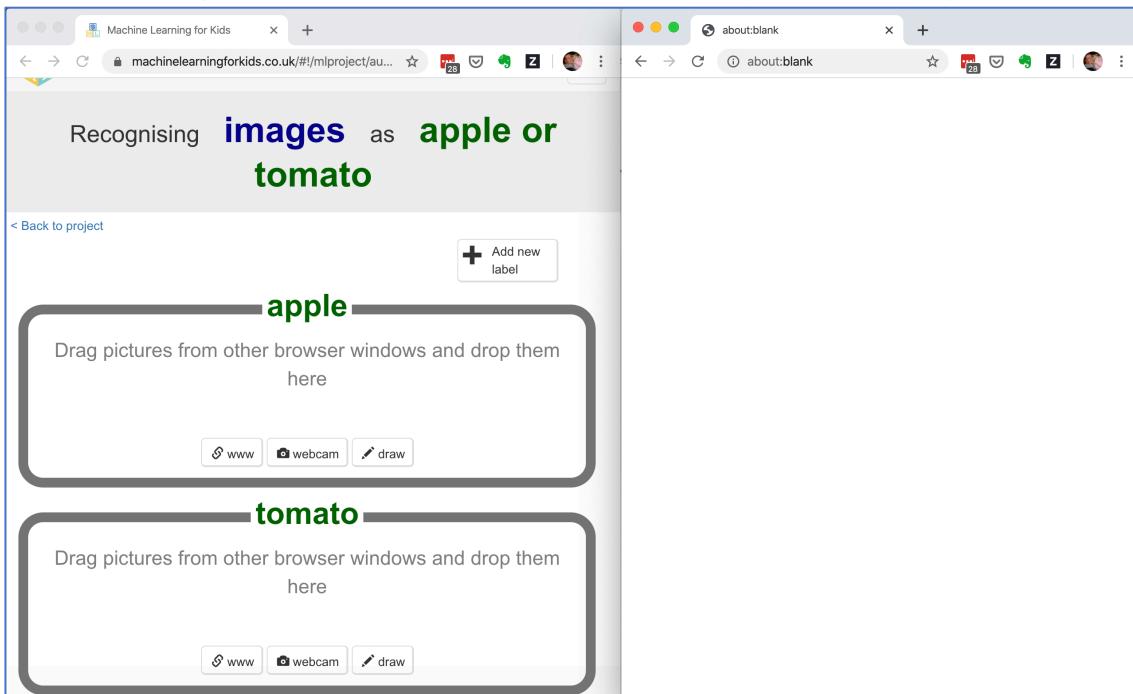


## 10. Open a new browser window

*How to do this will depend on what web browser you're using, but it's probably going to be a menu like “File -> New Window”*

*Ask your teacher or group leader if you need help.*

## 11. Arrange the two windows so that they are side-by-side



## 12. In the blank window, search for photos of green apples.

### 13. Drag one photo to the training bucket for apples Remember – green apples for now.

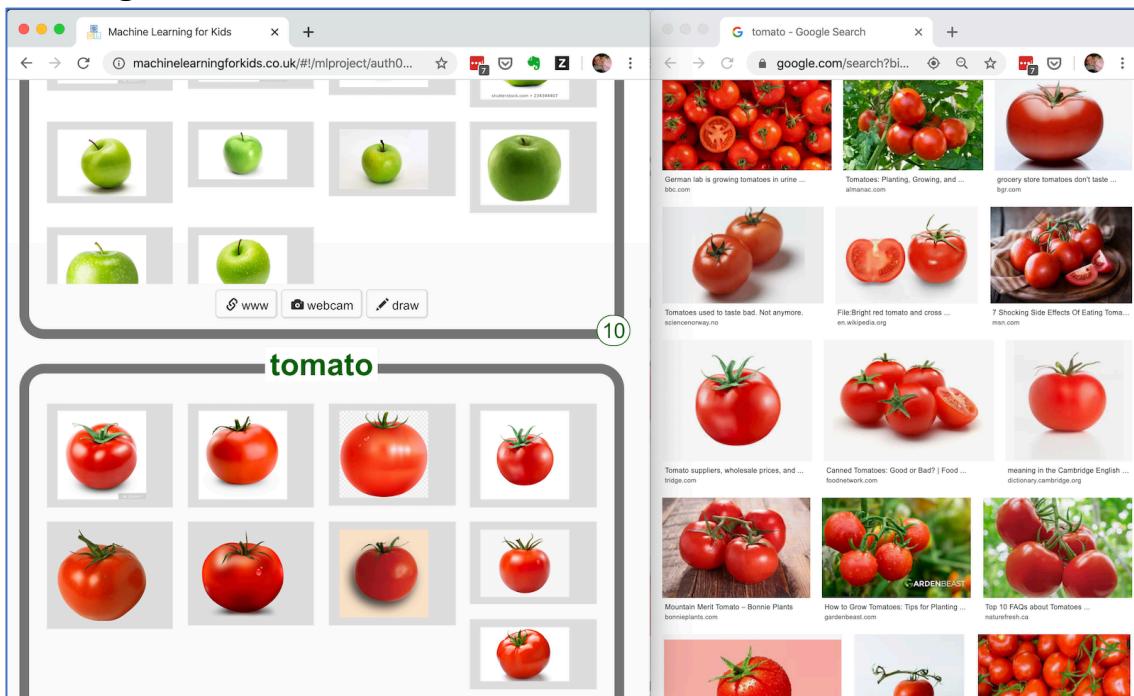
The screenshot shows the 'Machine Learning for Kids' application interface. On the left, there is a 'Machine Learning for Kids' tab and a 'green apple - Google Search' tab. The main area has two sections: 'apple' and 'tomato'. The 'apple' section contains a single green apple image with a bounding box around it. Below the image are three buttons: 'www', 'webcam', and 'draw'. A circular counter labeled '1' is positioned next to the 'apple' label. The 'tomato' section has a placeholder text 'Drag pictures from other browser windows and drop them here' and three buttons: 'www', 'webcam', and 'draw'. To the right of the application is a Google search results page for 'green apple' showing various images of green apples.

### 14. Repeat until you have at least ten examples of green apples

This screenshot shows the same application after adding more green apple images to the 'apple' bucket. There are now ten green apple images in the bucket, each with a bounding box. A circular counter labeled '10' is positioned next to the 'apple' label. The 'tomato' section remains empty. The right side of the screen still displays the Google search results for 'green apple'.

### 15. Search for photos of red tomatoes

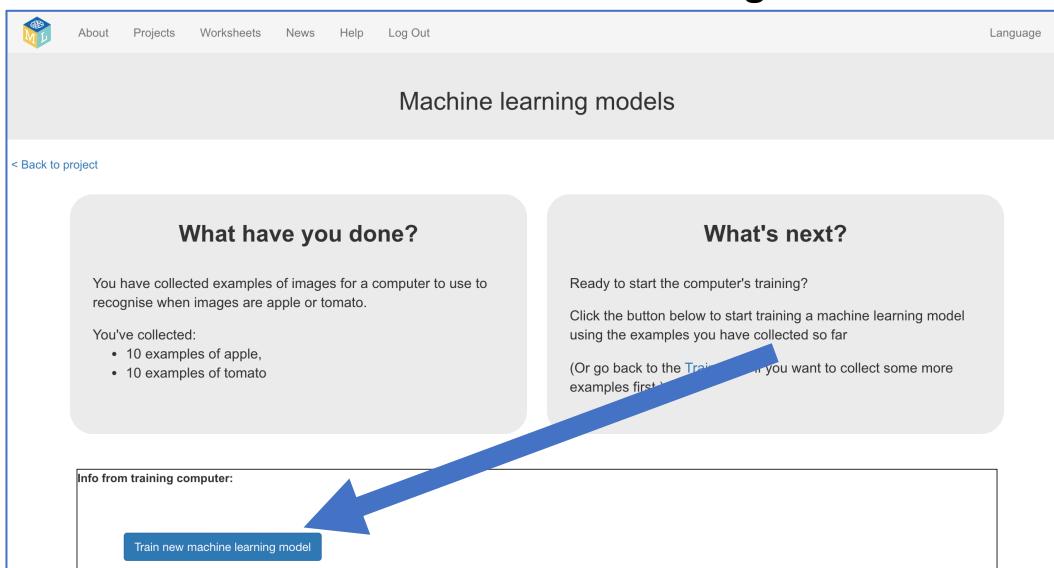
- 16.** Drag at least **ten** examples of red tomato photos to the “tomato” training bucket.



- 17.** Click the “[< Back to project](#)” link

- 18.** Click the “[Learn & Test](#)” button

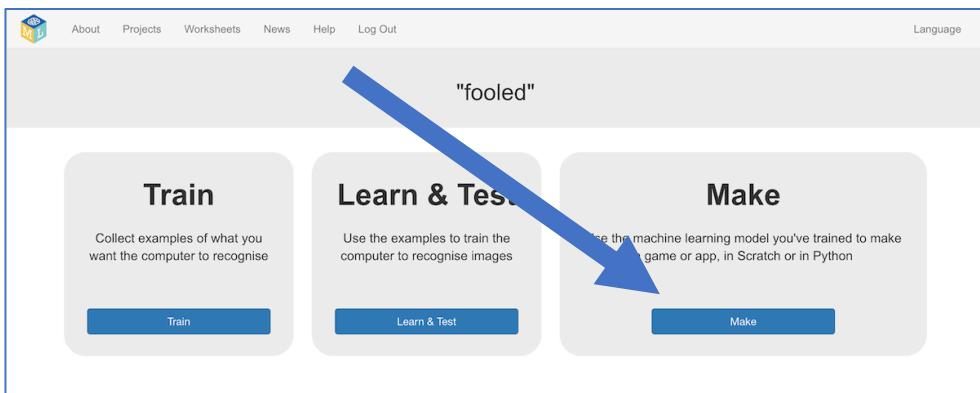
- 19.** Click the “[Train new machine learning model](#)” button



- 20.** Wait for the training to complete. This might take a few minutes.

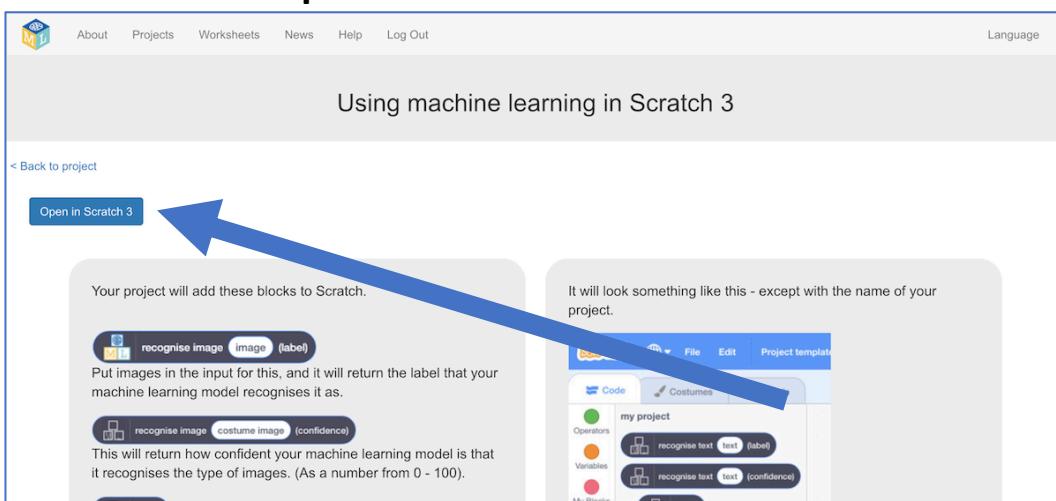
**21.** Click the “< Back to project” link

**22.** Click the “Make” button

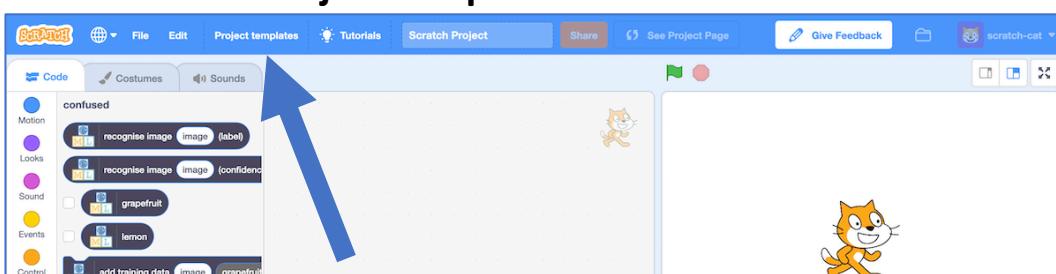


**23.** Click the “Scratch 3” button

**24.** Click the “Open in Scratch 3” button

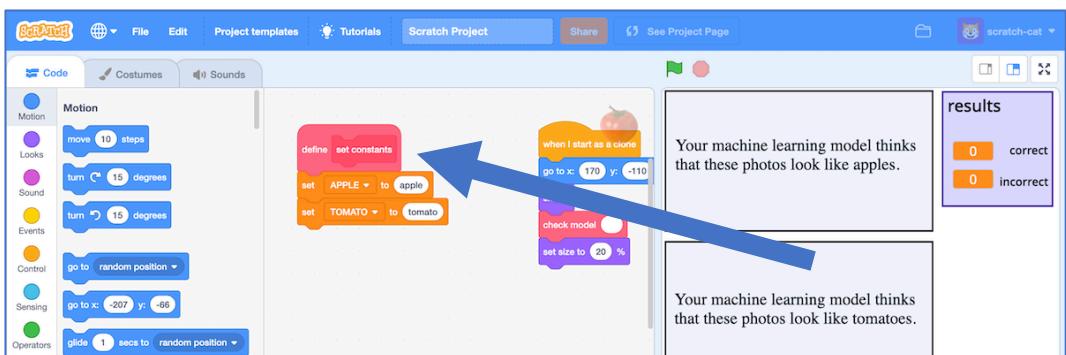


**25.** Click on “Project templates”

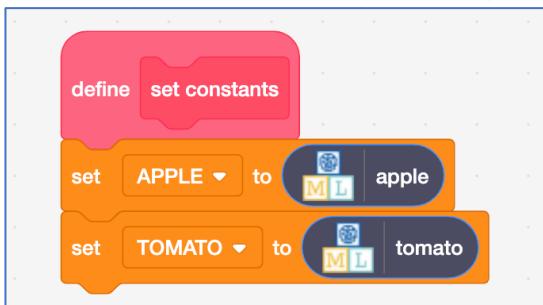


**26.** Click on “Fooled” to open the project template

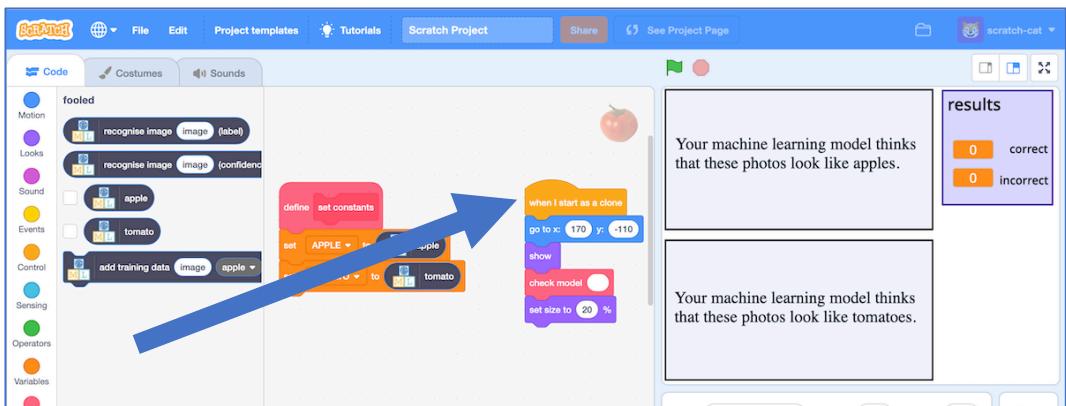
## 27. Find the “set constants” code



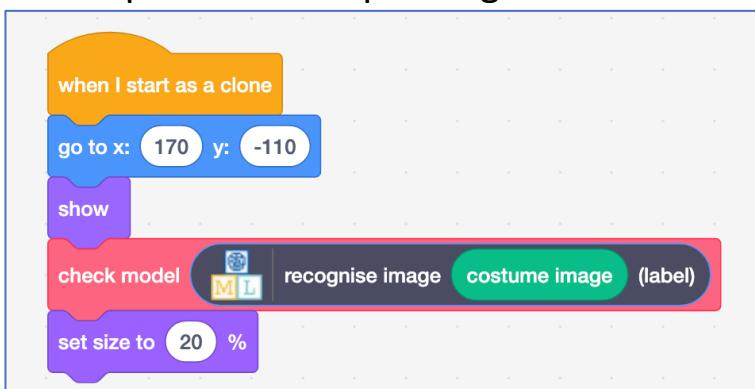
## 28. Update the script using the blocks from your project



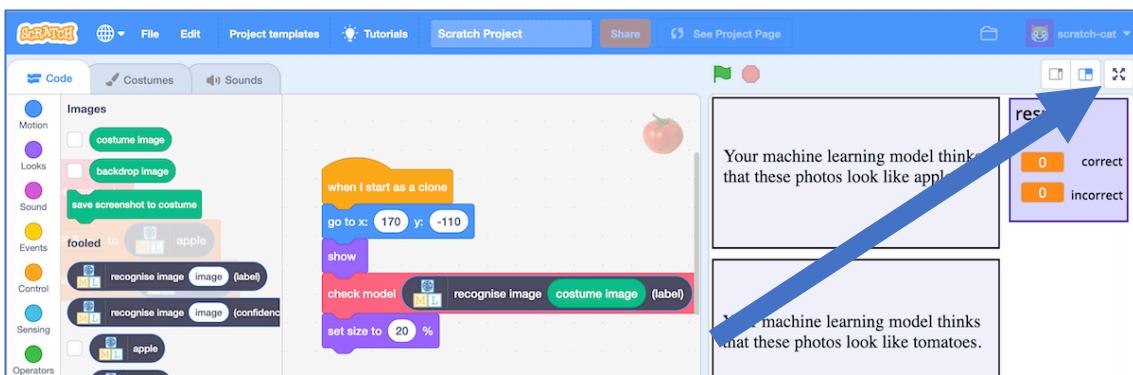
## 29. Find the “when I start as a clone” code



## 30. Update the script using the blocks from your project.



## 31. Click the “full-screen” button

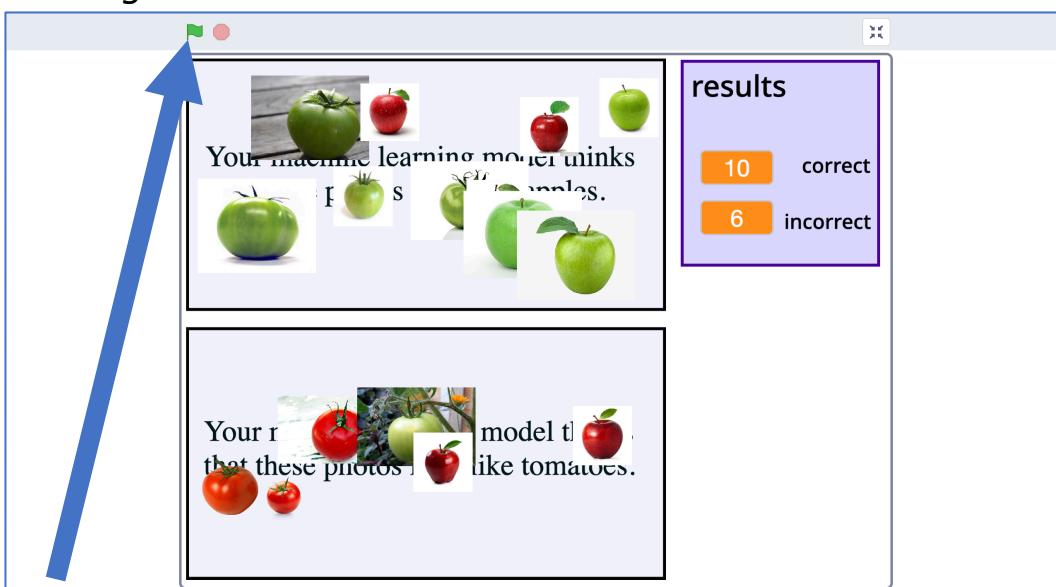


## 32. Click the green flag

Your script will use the machine learning model you trained to recognise a series of photos of apples and tomatoes.

Photos recognized as apples will be moved into the top box. Photos recognized as tomatoes will be moved into the bottom box.

The box on the right will display how many mistakes your machine learning model makes.



**Why do you think your machine learning model is making mistakes?**

Try to think of a reason for yourself before you read the next page!

You might find it helpful to look back at the training set you used and compare it with the test images in the Scratch project.

## What is happening?

When you train a machine learning model, you're asking the computer to look at sets of photos for patterns.

It looks for what photos in each set have in common and learns to recognise those patterns in new photos it is given.

You might want it to have recognise apples and tomatoes, but the computer doesn't know that. It could spot patterns about the colour of the background, or whether the photo is blurry or focused, or whether the lighting is dark or bright, or whether the object is large or small, or many other things.

When it makes decisions based on recognising those patterns in new photos, it can get the wrong answer.

### Your Training Set

“apples” – a set of photos of **green** objects

“tomatoes” – a set of photos of **red** objects

Confused by testing with:

- A photo of a red apple
- A photo of a green tomato

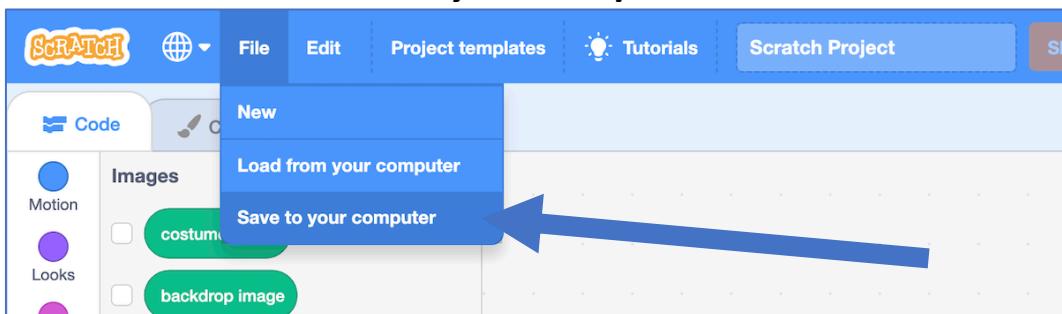
Your machine learning model might not have thought all red apples were tomatoes, or that all green tomatoes were apples.

There are other patterns that might have helped it get some correct, like the shape of the leaves.

But can you help it do better?

### 33. Save your Scratch project

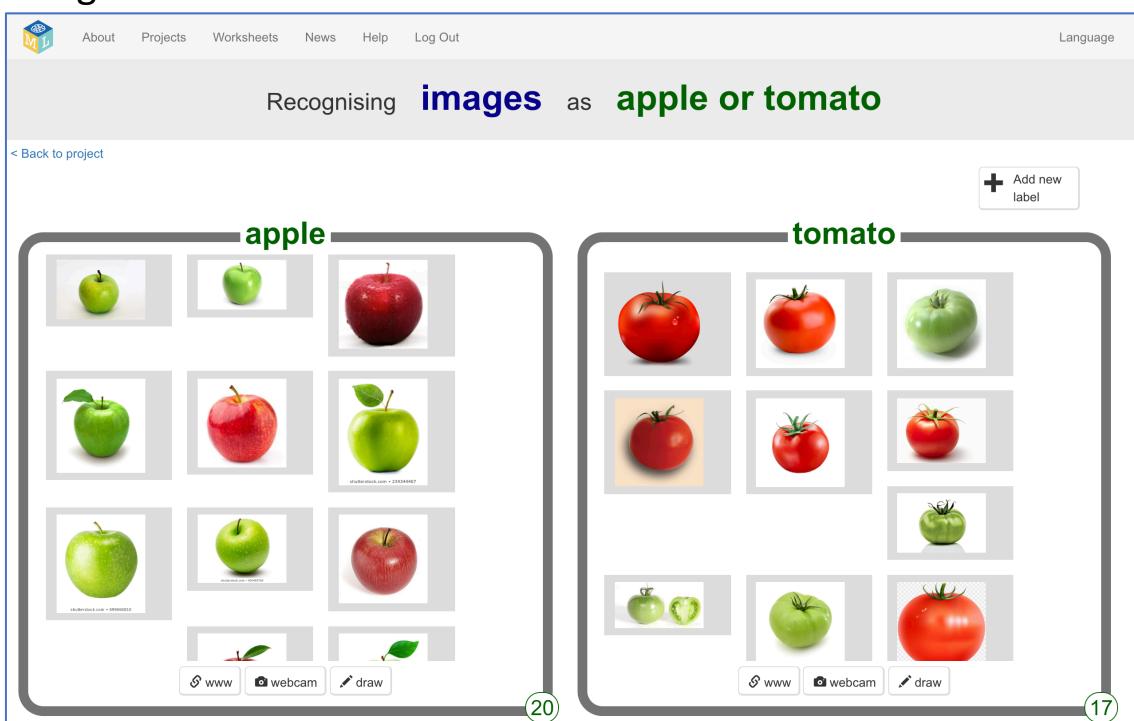
Click on “File” -> “Save to your computer”



### 34. Switch back to the training tool window

### 35. Click the “< Back to project” link and then click the “Train” button

### 36. Update your training data so that this time you have a mix of red and green fruit in both buckets.



### 37. Click the “< Back to project” link

### 38. Click the “Learn & Test” button

**39.** Click the “**Train new machine learning model**” button and wait for the model to finish training.

**40.** Switch back to the Scratch window.

*If you accidentally closed it, you can get back to it by doing this:*

- \* Click the “**< Back to project**” link
- \* Click the “**Make**” button
- \* Click the “**Scratch 3**” button
- \* Click the “**Open in Scratch 3**” button
- \* Open the project you saved before, with “**File**” -> “**Load from your computer**”

**41.** Run the scripts again with the new model

*Click full-screen, then click the Green Flag. Does it do better this time?*

## What have you done?

Machine learning models learn to recognise patterns in what you use to train it.

If all photos in a set have the same background, or the same lighting, or the same colour – then those can be patterns that the model uses to recognise pictures.

This time, you used a wider variety of photos to train the model.

For example, the “apple” training photos included both red and green apples. The only thing they all had in common was the shape and type of leaf.

This meant it was much more likely that the pattern the computer spotted in the training photos was based on the shape.

Variety in training data is essential when training a reliable model.

## The “Russian Tank” problem

This worksheet is based on an old story told to Artificial Intelligence students called “The Russian Tank problem”.

It’s unclear whether or not it’s a true story, as there are many different versions. Whether or not it’s true, it’s a useful way to teach an important lesson in training machine learning systems.

Here are two examples of how the story is told:

### Spotting camouflaged Russian tanks

Once upon a time, the US Army decided to use machine learning to recognize tanks hiding behind trees in the woods. Researchers trained a machine learning model using photos of a woods without tanks, and photos of the same woods with tanks sticking out from behind trees.

It seemed to work, but in tests the model didn’t do better than random guesses.

It turned out that in the researchers’ training data set, photos of camouflaged tanks had been taken on cloudy days, while photos of plain forest had been taken on sunny days. The machine learning model had learned to recognise cloudy days from sunny days, instead of recognising camouflaged tanks.

### Recognising American and Russian tanks

Once upon a time, the US Army tried training a computer to tell the difference between Russian and American tanks by the way they look. Researchers trained a machine learning model using photos they took of American tanks, and spy photos they collected of Russian tanks.

But when they tested it in the field, the machine learning model didn’t do any better than randomly guessing.

It turned out that the researchers had photos of American tanks which were large, high-resolution and high-quality. But the long-distance spy photos of Russian tanks they were able to get were all blurry, low-resolution and grainy.

The machine learning model had learned to recognise the difference between grainy photos and high-quality photos, instead of Russian or America.