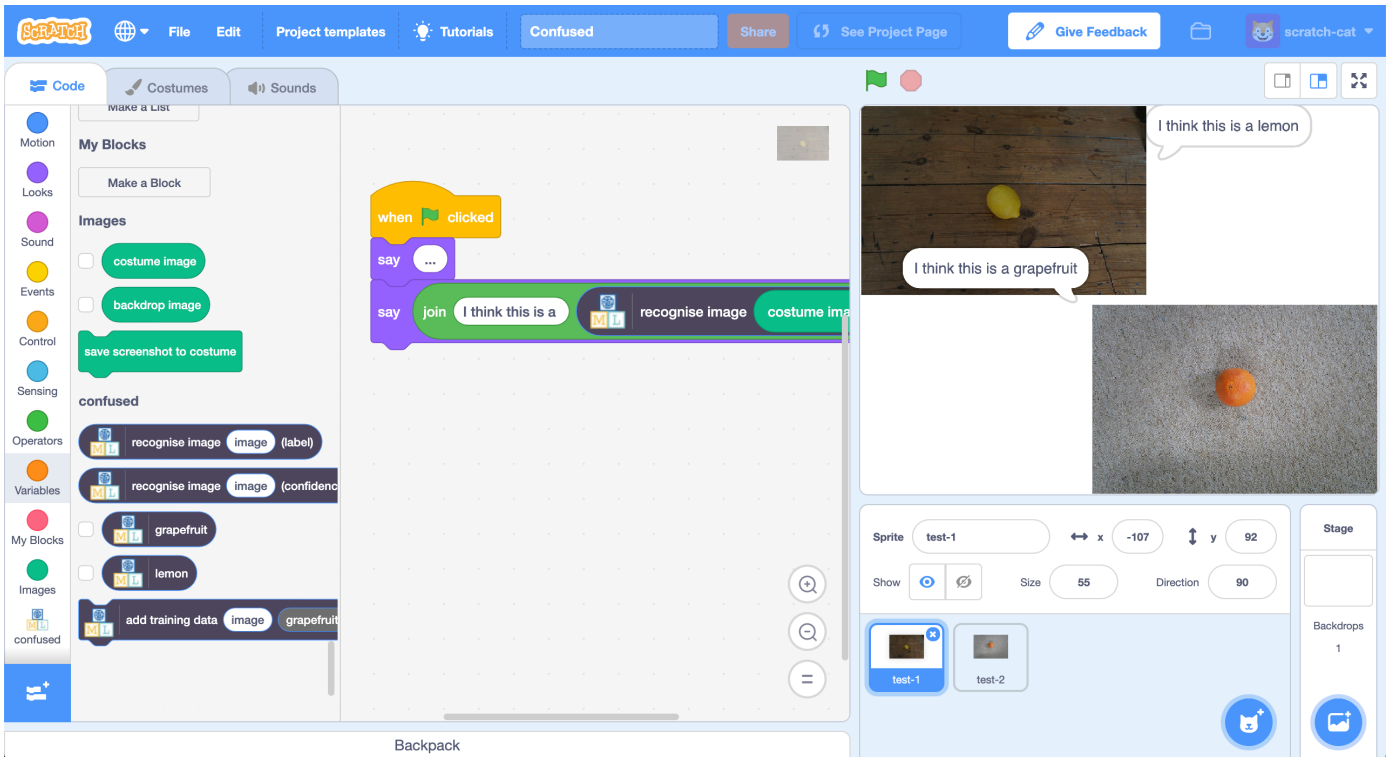
Confused

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

You'll train a machine learning model to recognise lemons and grapefruits.

To start with, you'll train it badly so that it confuses the two.

When you understand why that happened, you'll train it again so that it is harder to confuse.



1. Go to https://machinelearningforkids.co.uk/ in a web browser

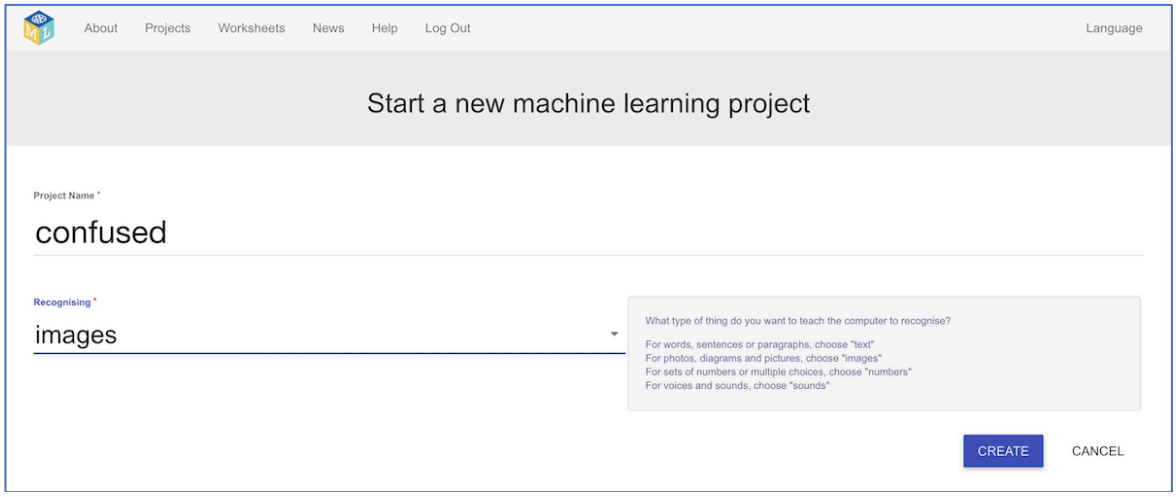
2. Click on "**Log In**" and type in your username and password

If you don't have a username, ask your teacher to create one for you.
If you can't remember password, ask your teacher to reset it for you.

3. Click on "**Projects**" on the top menu bar

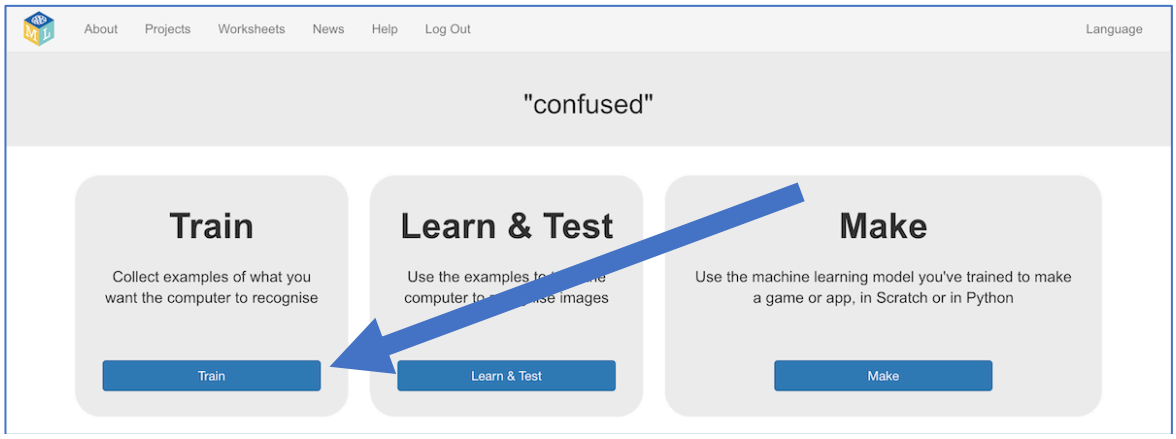
4. Click the **"+ Add a new project**" button.

5. Name your project "confused". Set it to learn how to recognise "**images**". Click the "**Create**" button



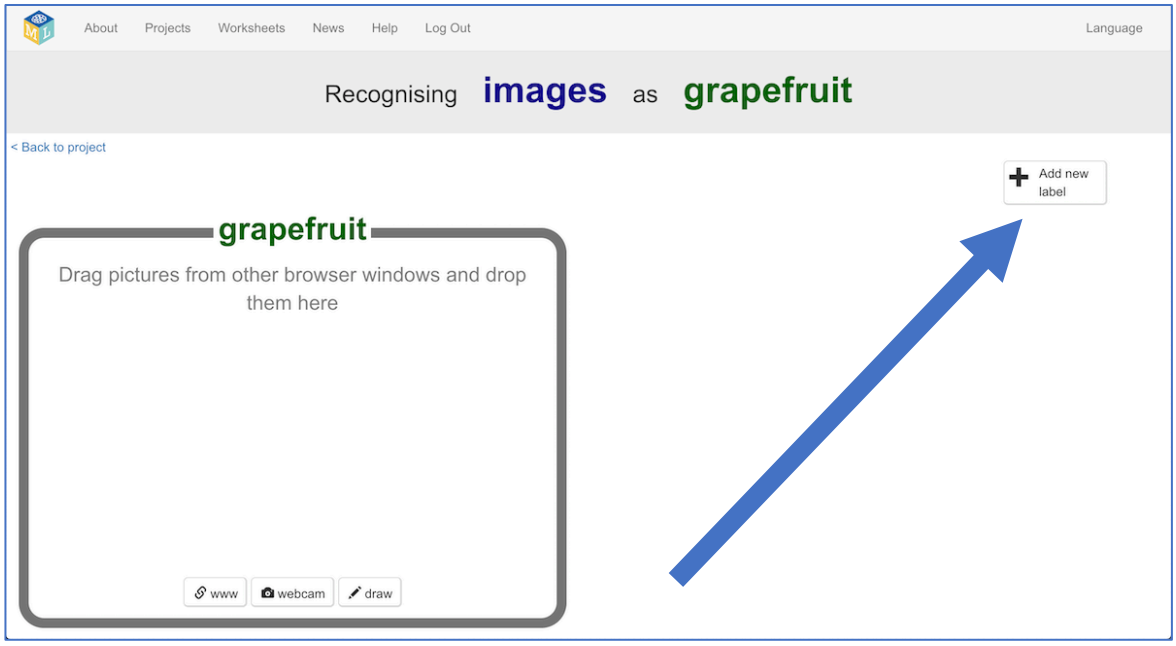
6. You should see "**confused**" in the list of your projects. Click on it.

7. Click the **Train** button.

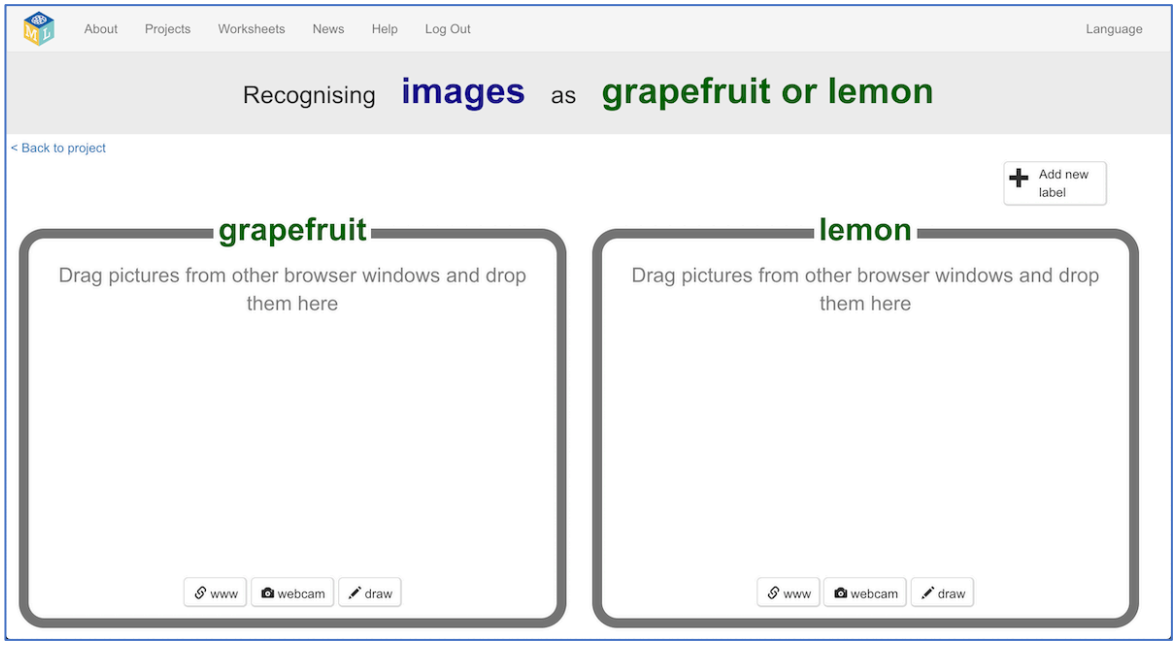


8. Click the **"+ Add new label"** button.

9. Create a new label called "grapefruit"



10. Click "**+ Add new label**" again and create a label called "lemon"

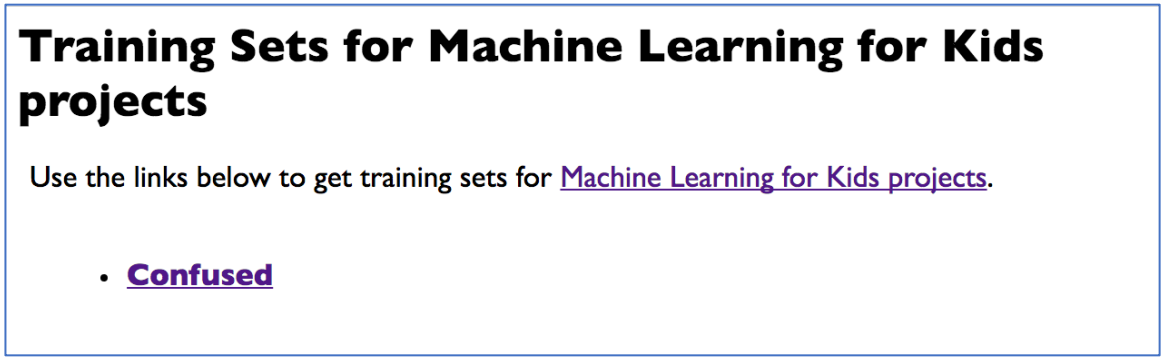


11. Open a new browser window

*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.

12. Go to https://machinelearningforkids.co.uk/datasets **in the new window**.

That's the same web address as before, with "data" "sets" on the end.



13. Click on the "**Confused**" link

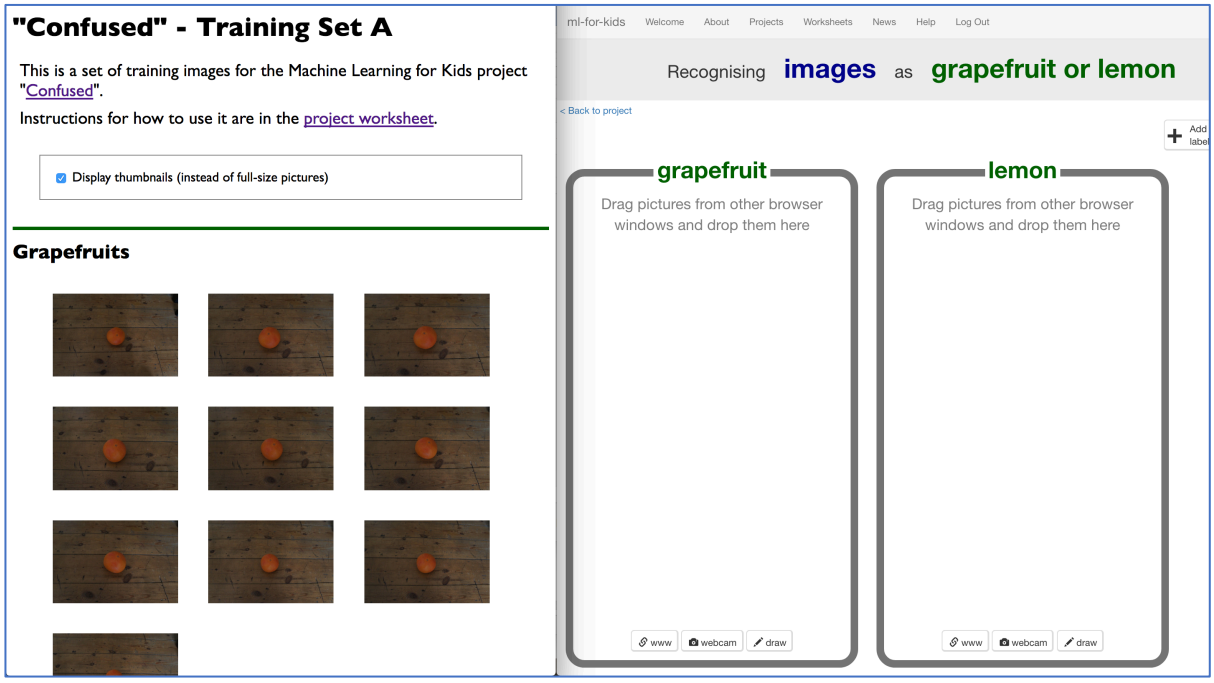
14. Click on "**First** Training Set"

***Don't** pick "Final Training Set"– you'll use that later.*

15. You should see pictures of grapefruits and lemons that you can use to train the computer.

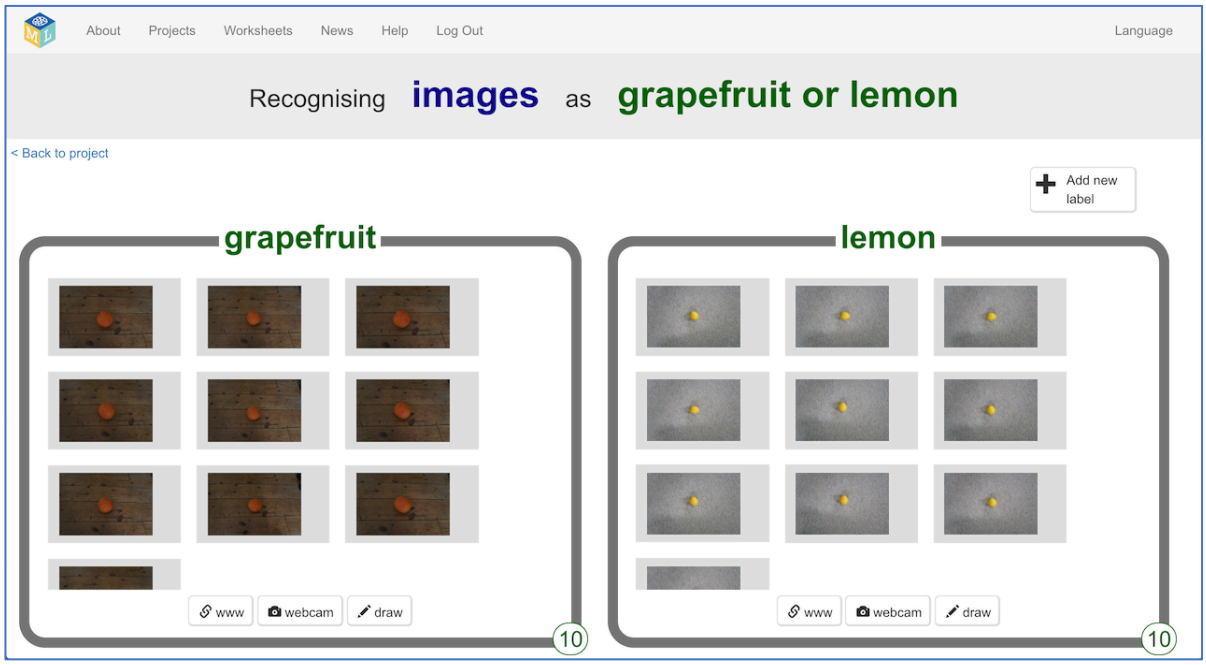
Click on the "**Display thumbnails**" checkbox so you can see them all.

16. Arrange the two web browser windows so they're side by side



17. Drag the ten grapefruit pictures into the “grapefruit” bucket.

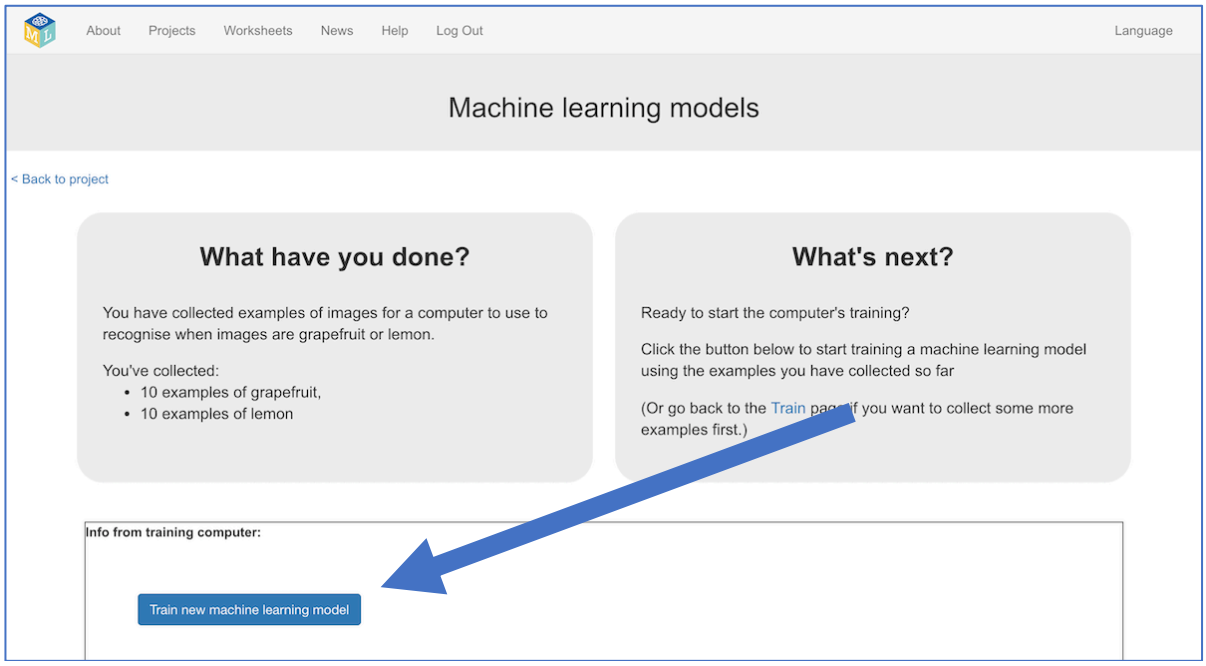
18. Drag the ten lemon pictures into the “lemon” bucket.



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

20. Click the “**Learn & Test**” button

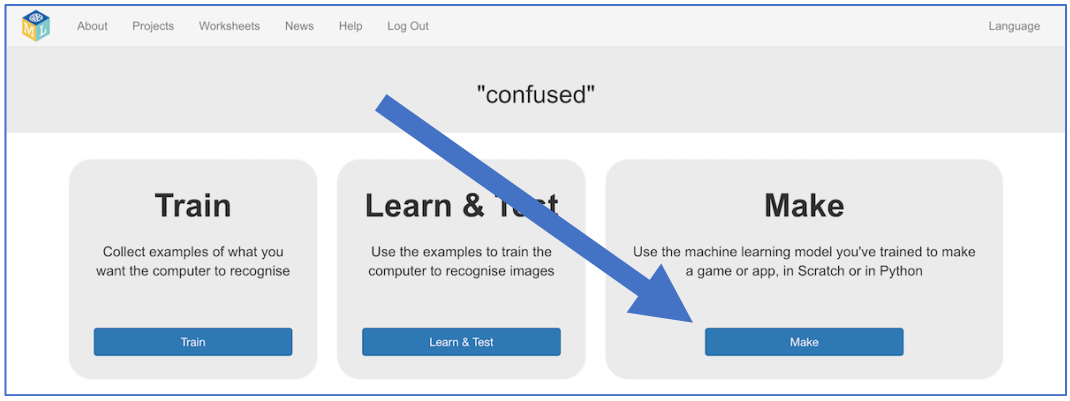
21. Click the “**Train new machine learning model**” button



22. Wait for the training to complete. This might take a few minutes.

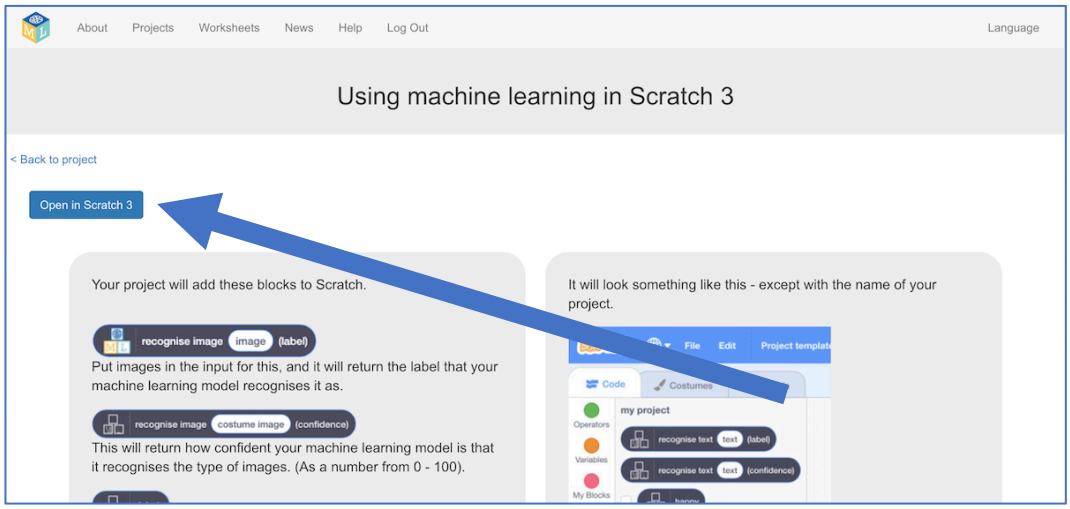
23. Click the **"< Back to project**" link

24. Click the "**Make**" button

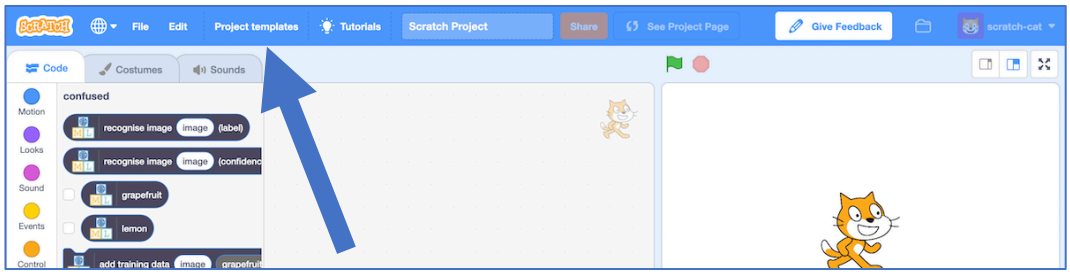


25. Click the "**Scratch 3**" button

26. Click the "**Open in Scratch 3**" button

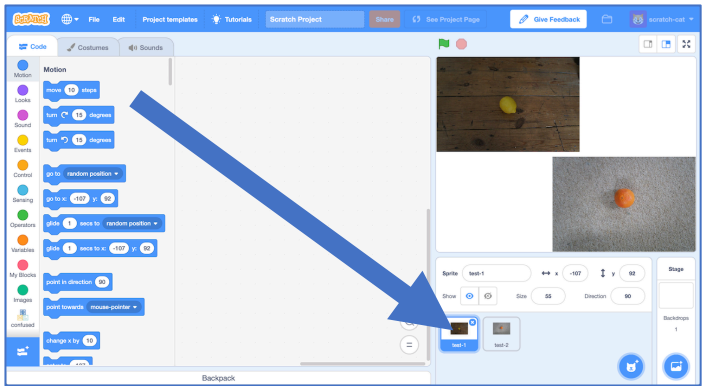


27. Click on "**Project templates**"

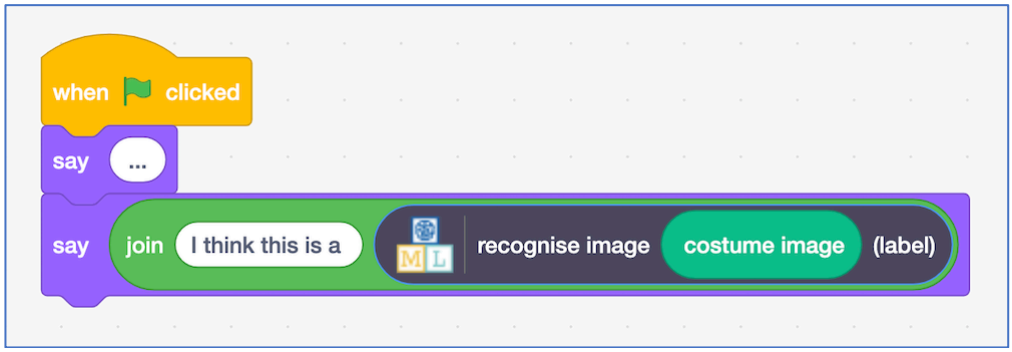


28. Click on "Confused" to open the project template

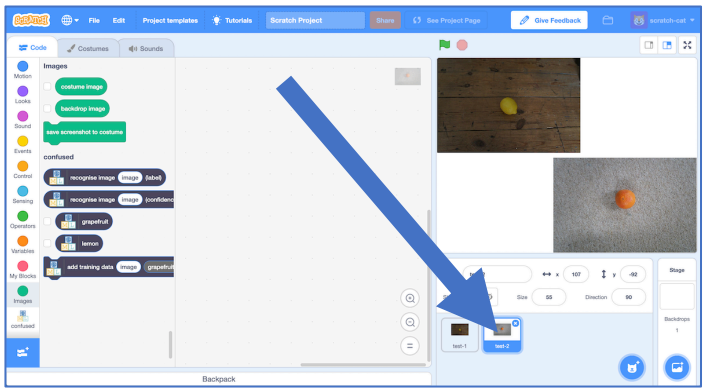
29. Click the “**test-1**” sprite



30. Create this script in the “**test-1**” sprite

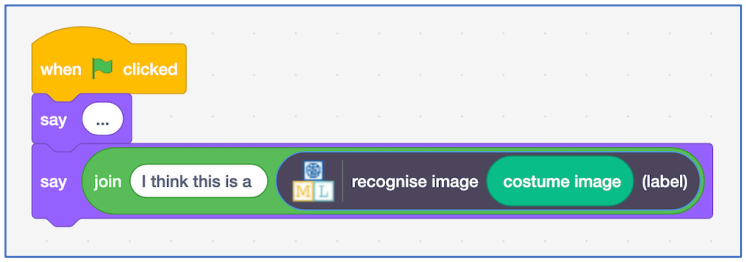


31. Click on the “**test-2**” sprite

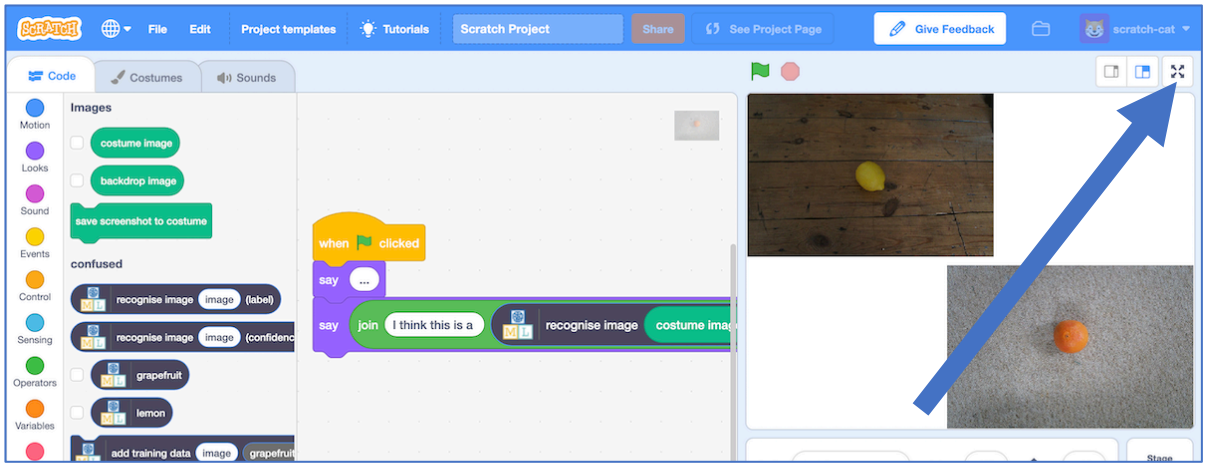


32. Enter the same script as before, this time in the “**test-2**” sprite

You can save time by dragging the script you wrote before onto the test-2 sprite to make a copy of it.

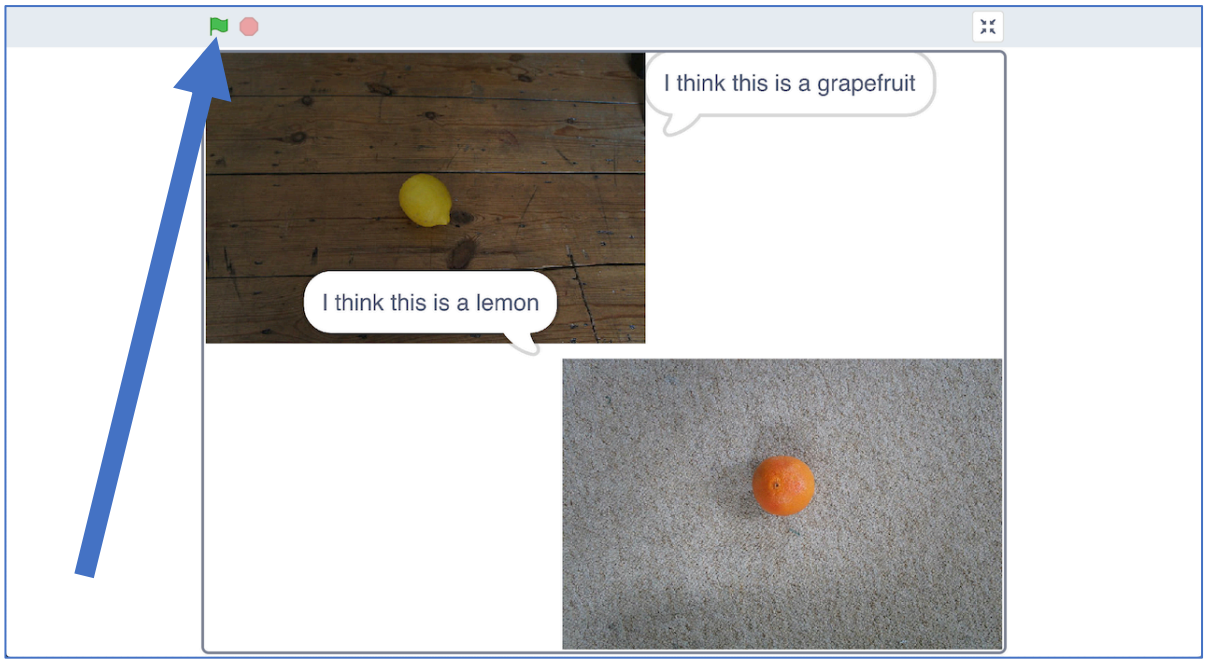


33. Click the "**full-screen**" button



34. Click the green flag

Your script will use the machine learning model you trained to recognise the two photos.



Why do you think the computer is getting this wrong?

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 fruits, 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 many other things.

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

First Training Set

"grapefruits" – a set of **dark** images on a **wooden** background

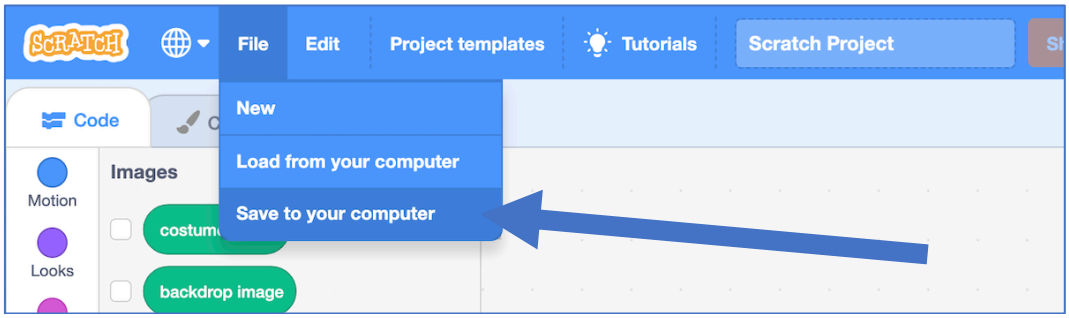
"lemons" – a set of **light** images on a **cream** carpet background

Confused by testing with:

- A **dark** photo of a lemon on a **wooden** background
- A **light** photo of a grapefruit on a **cream** carpet background

35. Save your Scratch project

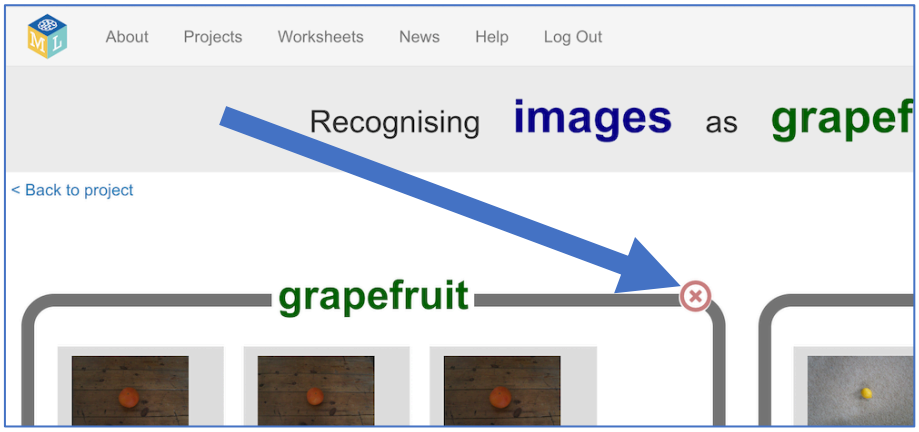
*Click on "**File**" -> "**Save to your computer**"*



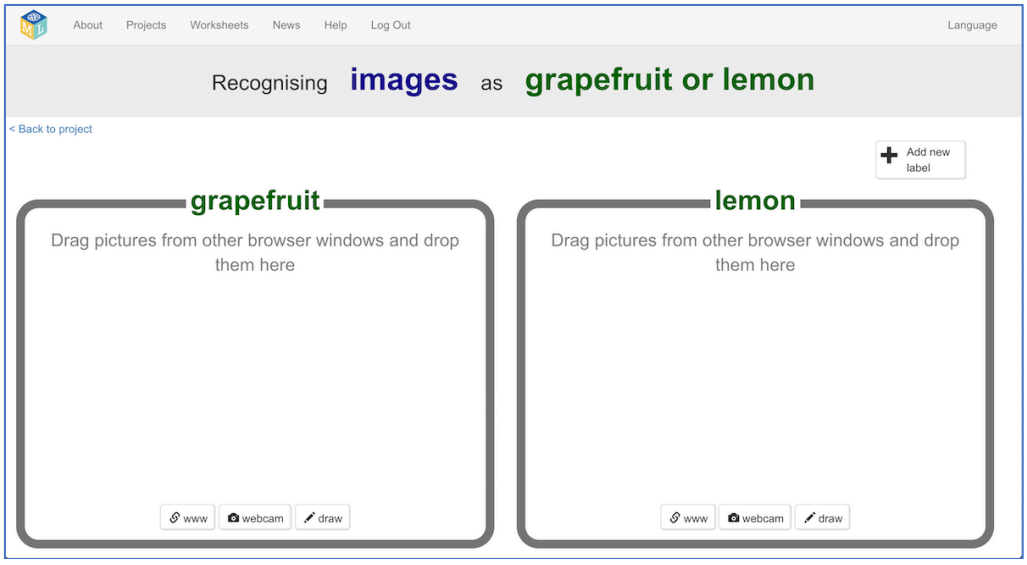
36. Switch back to the training tool window

37. Click the "**< Back to project**" link and then click the "**Train**" button

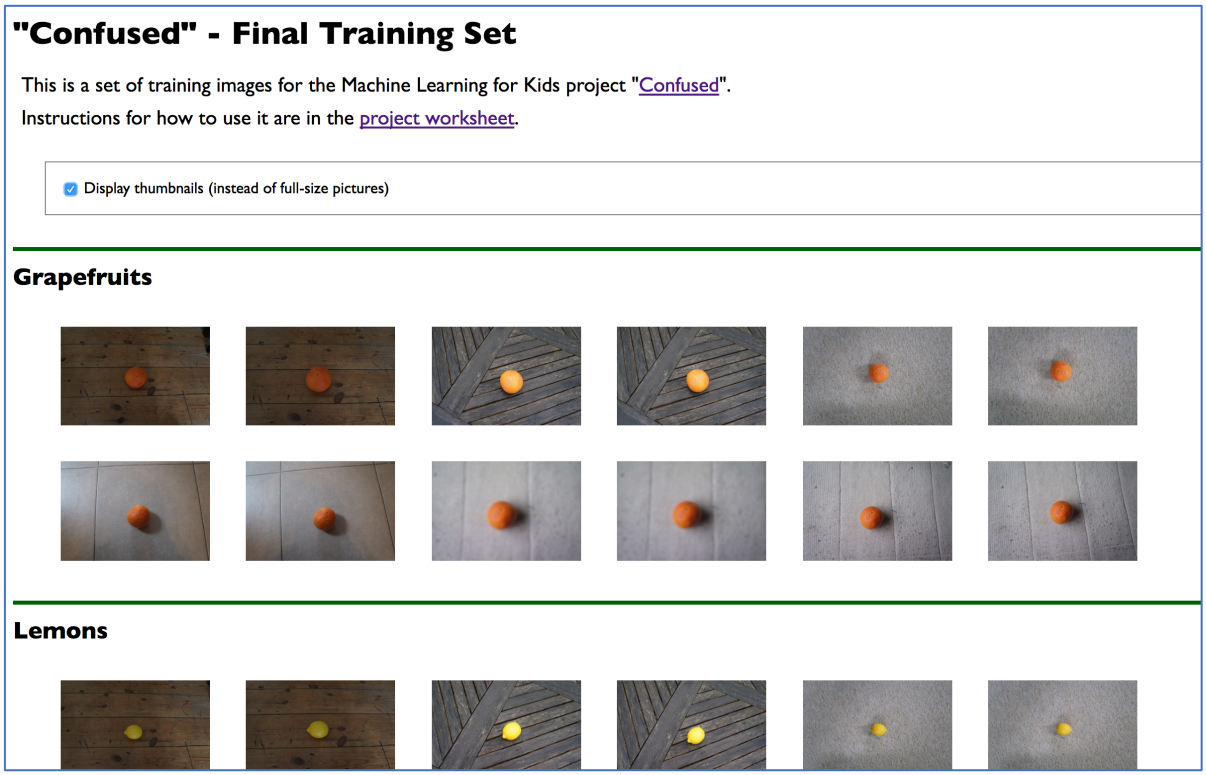
38. Click the red crosses to delete the training pictures you used before



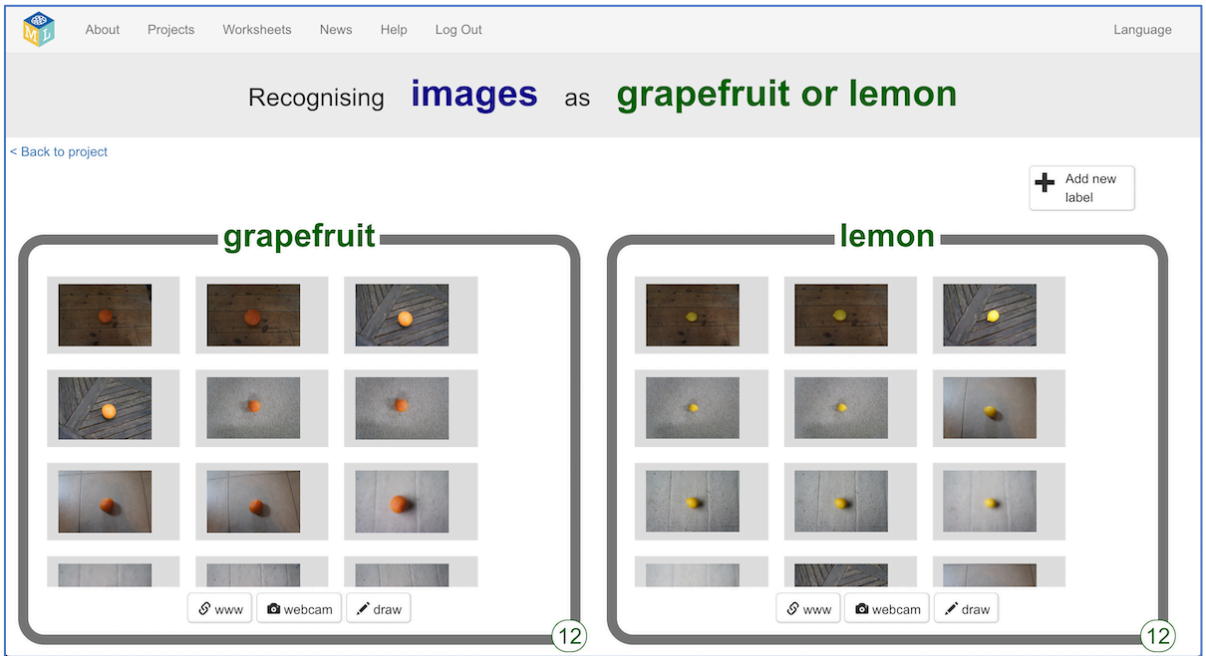
39. Recreate the training buckets so you're ready to start again



40. In the browser window with the training set pictures, click Back. This time, choose "**Final** Training Set"
If you didn't leave the window open before, open a new window now, go to https://machinelearningforkids.co.uk/datasets and go to "Confused"



41. Drag the new training images into your "lemon" and "grapefruit" buckets, like you did before.



42. Click the "**< Back to project**" link

43. Click the "**Learn & Test**" button

44. Click the "**Train new machine learning model**" button and wait for the model to finish training.

45. 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**"*

46. Run the scripts again with the new model

Click full-screen, then click the Green Flag. Does it get it right 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 focus level – 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 "lemon" training photos were taken on different backgrounds, taken inside and outside, in light and dark, some in focus and some blurry. The only thing they all had in common was that they all had a lemon in there.

This meant it was much more likely that the pattern the computer spotted in the training photos was that there was a yellow fruit in the middle.

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.