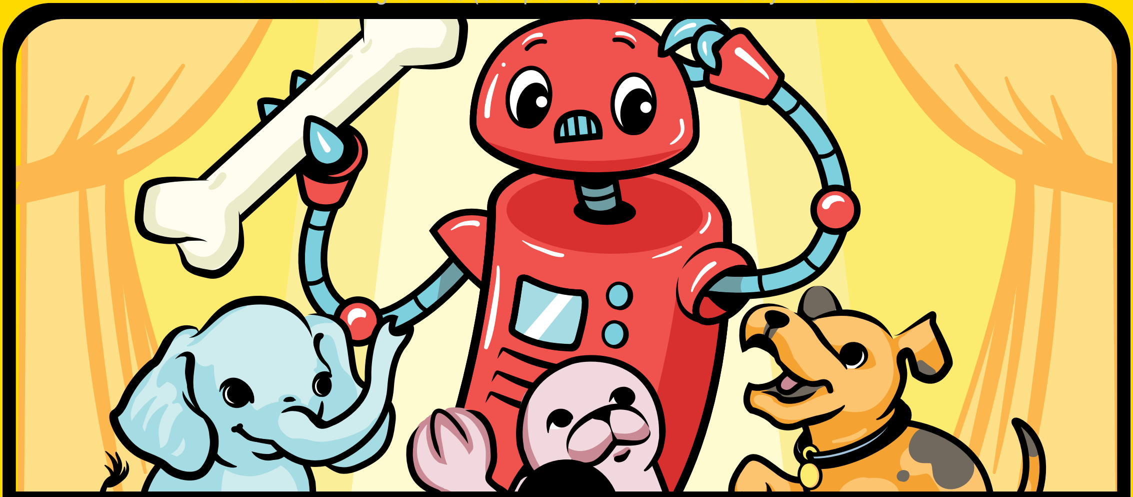


3

SORTING ANIMAL PICTURES

We all love pictures. We take over a trillion digital photos every year, and that doesn't even include other types of pictures that we make, like drawings and paintings.

Using computers to sort pictures and help us find the ones we need is called *image recognition*. To create an image recognition system, we collect lots of pictures of the same subject. Then we use those pictures to train an ML *model*, which identifies what those pictures have in common and uses that to recognize new pictures.

For example, if we want to train a computer to recognize photos of kittens, we collect lots of photos of kittens. ML systems use these photos to learn the shapes, patterns, and colors that often show up in kitten photos. The model can then recognize whether a photo has a kitten in it.

People use image recognition every day. Online photo-sharing tools use it to help sort the photos that we upload. Websites use it to describe the contents of photos to help people with visual disabilities know what's in photos even if they can't see them. Social media websites use it to recognize the faces of our friends and family in the photos we post. Companies use it to track when their logos or products are included in photos posted online so they know when they're being talked about on social media. More importantly, doctors use it to help them recognize medical conditions in patients' scans and photos. Doctors need to remember many different symptoms and signs of illness, so image recognition systems can help them by identifying things like skin tumors in photographs or cancer in a microscopic picture of a cell.

In this chapter, you'll make your own image recognition system by training an ML model to recognize and automatically sort photos of animals. Let's get started!

BUILD YOUR PROJECT

First, choose two types of animals that you want the computer to recognize. For this chapter, I chose cows and sheep to make a farm-themed Scratch project (see Figure 3-1). You can choose any two animals you'd like, as long as you can easily find lots of photos of them.

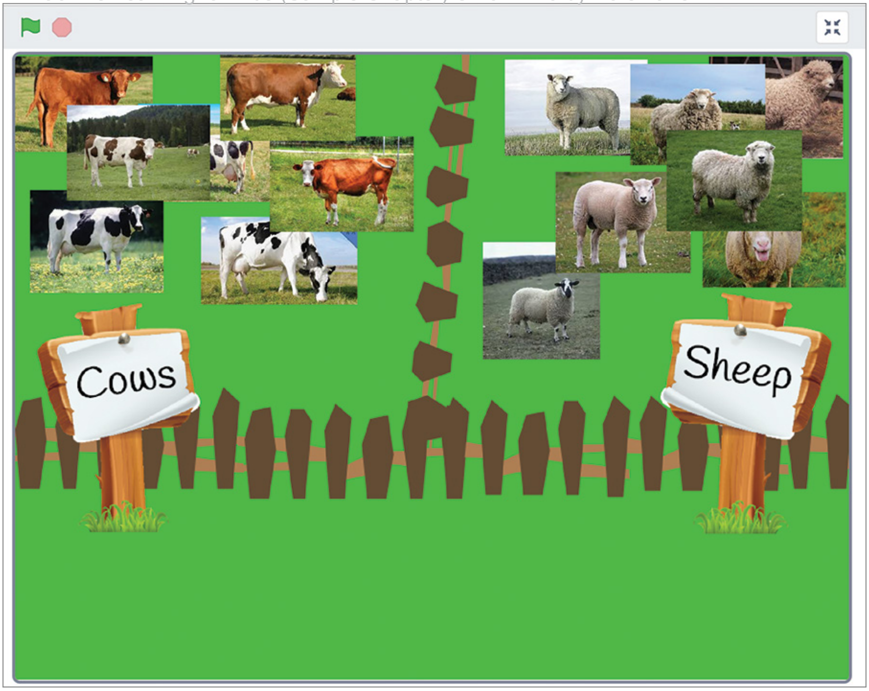


Figure 3-1: Sorting photos of animals into groups

TRAIN YOUR MODEL

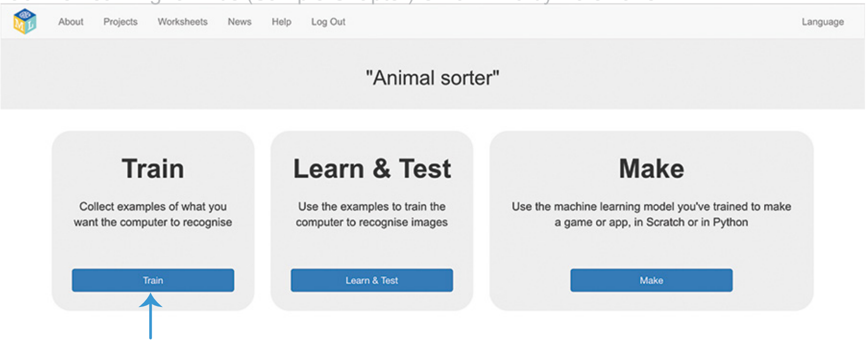
To train the computer to recognize different pictures of your two animals, you'll need to collect lots of images of those animals and use them to train an ML model.

1. Create a new ML project, enter **Animal sorter** as its name, and set it to learn to recognize images.

NOTE

If you're not sure how to create an ML project, read the section "Creating a New ML Project" on page XX in Chapter 2.

2. Click **Train**, as shown in Figure 3-2.



"Animal sorter"

Train

Collect examples of what you want the computer to recognise

Learn & Test

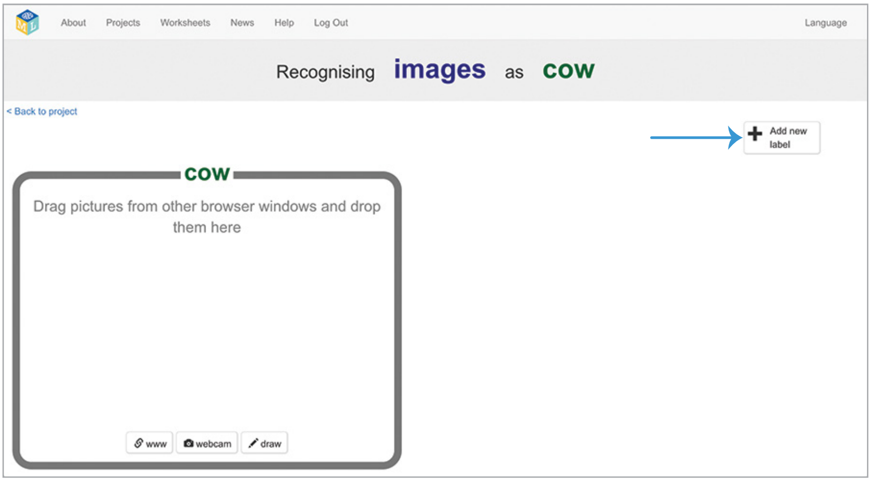
Use the examples to train the computer to recognise images

Make

Use the machine learning model you've trained to make a game or app, in Scratch or in Python

Figure 3-2: Train is the first phase of an ML project.

3. Click **Add new label** (see Figure 3-3). Then enter the name of your first type of animal.



Recognising **images** as **cow**

Drag pictures from other browser windows and drop them here

*Figure 3-3: Click **Add new label** to create a new bucket of training examples.*

4. Open a second window in your web browser (usually by selecting **File▶New window**) and arrange your two windows side by side, as shown in Figure 3-4. In the second window, search for photos of the first type of animal. In my case, I've searched for photos of cows.

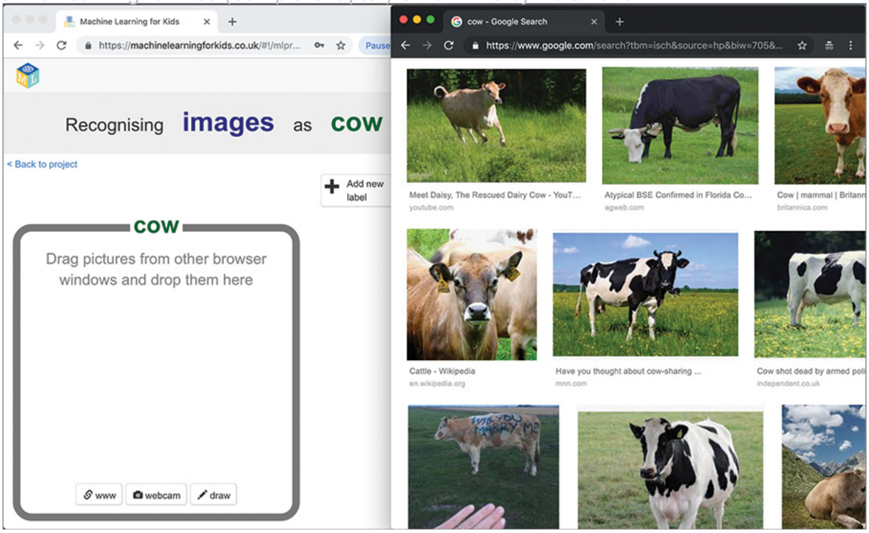

Machine Learning for Kids
https://machinelearningforkids.co.uk/#!/mlpr...
Pause
Recognising images as cow
< Back to project
Add new label
cow
Drag pictures from other browser windows and drop them here
www
webcam
draw
cow - Google Search
https://www.google.com/search?tbm=isch&source=hp&biw=705&...
Meet Daisy, The Rescued Dairy Cow - YouT...
youtube.com
Atypical BSE Confirmed in Florida Co...
agweb.com
Cow | mammal | Britann
britannica.com
Cattle - Wikipedia
en.wikipedia.org
Have you thought about cow-sharing ...
mnn.com
Cow shot dead by armed poli
independent.co.uk

Figure 3-4: Arrange your two web browser windows side by side.

5. Drag a photo from the search window and drop it into the training bucket for your first type of animal. You should see a thumbnail version of the photo in your training bucket, as shown in Figure 3-5. If you don't see it, try dragging and dropping the photo again.

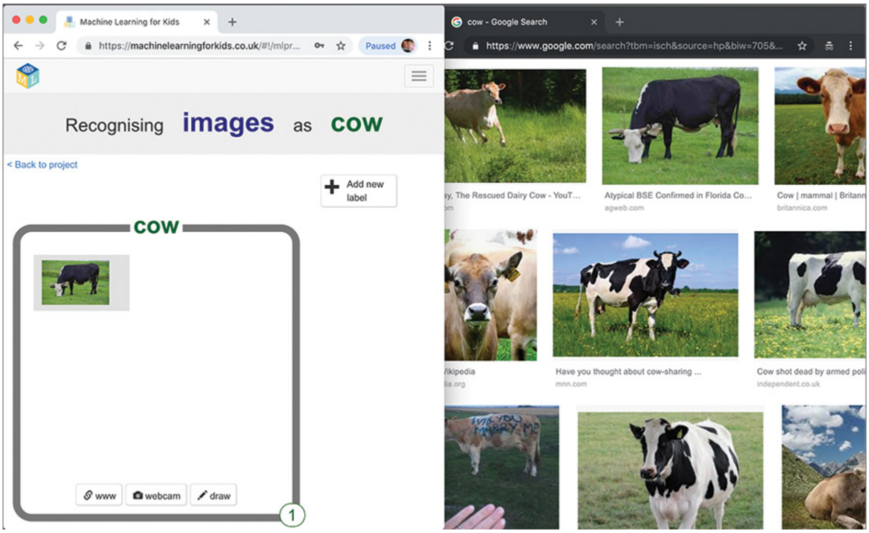

Machine Learning for Kids
https://machinelearningforkids.co.uk/#!/mlpr...
Paused
Recognising images as cow
< Back to project
Add new label
cow
www
webcam
draw
1
cow - Google Search
https://www.google.com/search?tbm=isch&source=hp&biw=705&...
y, The Rescued Dairy Cow - YouT...
Atypical BSE Confirmed in Florida Co...
agweb.com
Cow | mammal | Britann
britannica.com
ikipedia
Have you thought about cow-sharing ...
mnn.com
Cow shot dead by armed poli
independent.co.uk

Figure 3-5: Dropping photos of cows into my training bucket

6. Repeat step 5 until you have at least 10 different photos of that animal, as shown in Figure 3-6.

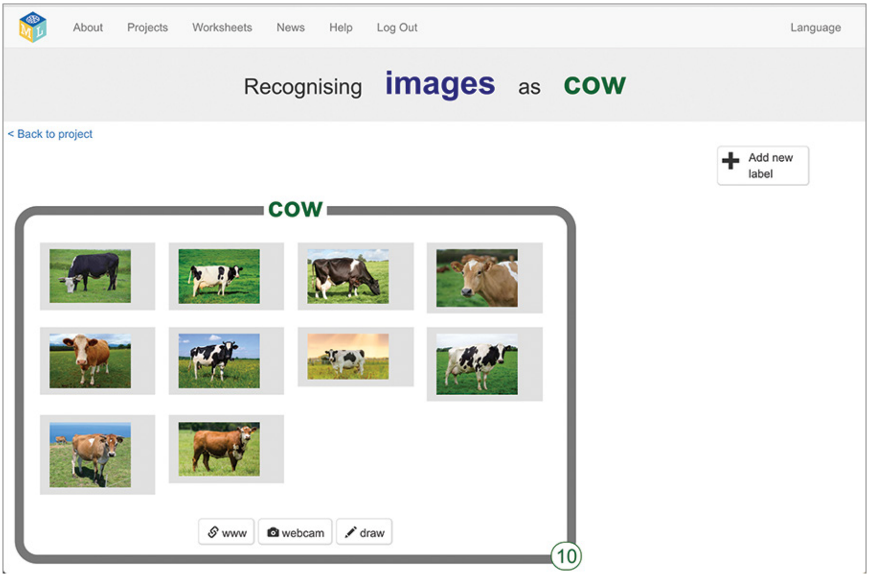


Figure 3-6: My training data for recognizing photos of cows

7. Repeat steps 3 through 6 until you have at least 10 different photos for each type of animal, as shown in Figure 3-7.

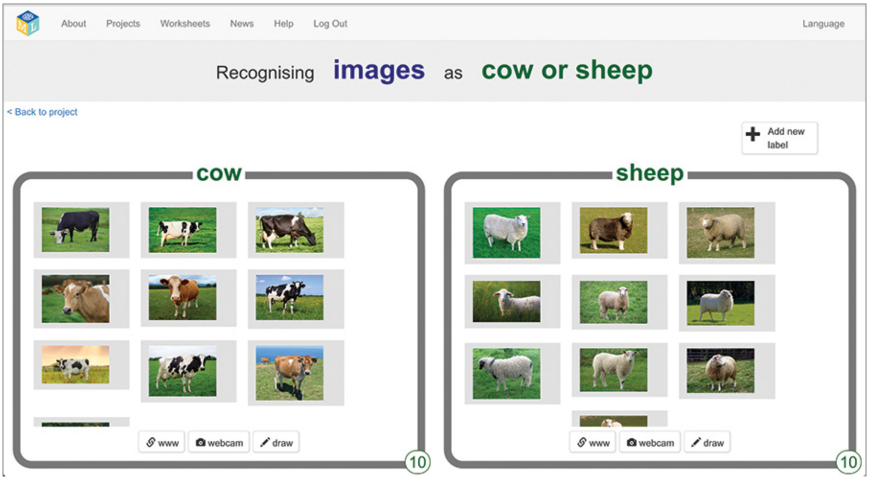
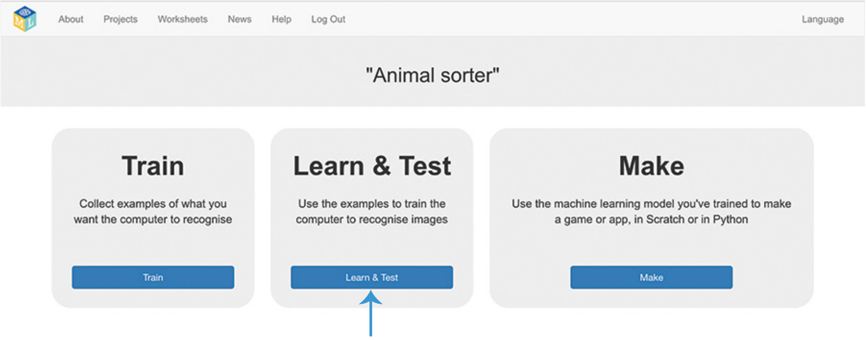


Figure 3-7: Training data for my farm-themed project

8. Click **Back to project** in the top-left corner of the screen.
9. Click **Learn & Test** (see Figure 3-8).



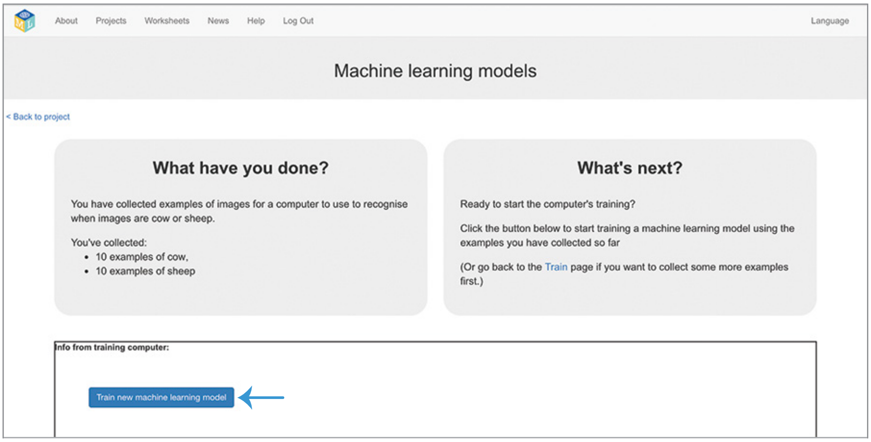
Collect examples of what you want the computer to recognise

Use the examples to train the computer to recognise images

Use the machine learning model you've trained to make a game or app, in Scratch or in Python

Figure 3-8: Learn & Test is the second phase of an ML project.

10. Click **Train new machine learning model** (see Figure 3-9).



What have you done?

You have collected examples of images for a computer to use to recognise when images are cow or sheep.

You've collected:

- 10 examples of cow,
- 10 examples of sheep

What's next?

Ready to start the computer's training?

Click the button below to start training a machine learning model using the examples you have collected so far

(Or go back to the Train page if you want to collect some more examples first.)

Figure 3-9: Click ***Train new machine learning model*** *to start the training.*

The computer will use the examples you've collected to learn what the photos of each animal have in common. This can take a few minutes, but you can continue to the next step in your second web browser window while you wait.

PREPARE YOUR GAME

To test your ML model, you'll need some new photos that you haven't used for training. The computer will use what it learns from your training examples to try to recognize your two animals in these new photos. Then you'll build a project in Scratch that tests how well your model performs.

1. Search for more photos of the animals you've chosen, and save them to your computer. To save a photo, right-click it and select **Save Picture** or **Save Image As**, as shown in Figure 3-10.

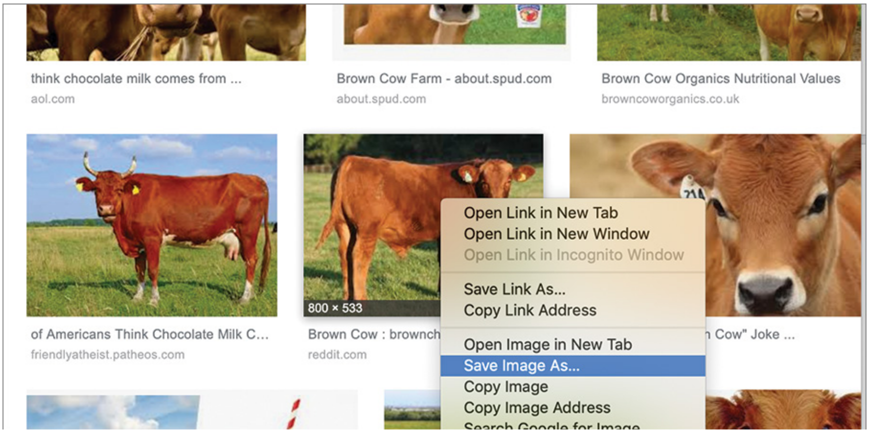


Figure 3-10: Saving test photos to the computer

NOTE

Don't choose the same photos you used for training the model. You want to test how good the computer is at recognizing new photos, not how good it is at remembering them.

2. Aim for at least five test photos of each type of animal that you chose, as shown in Figure 3-11.

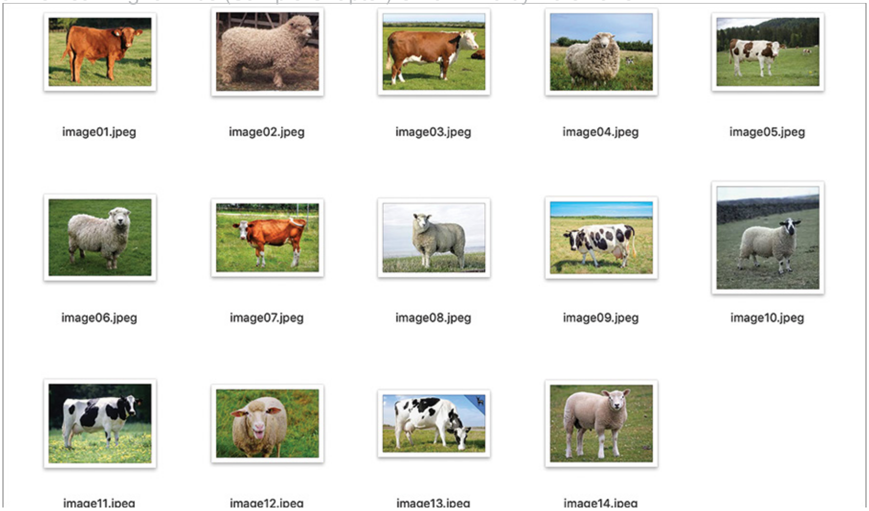
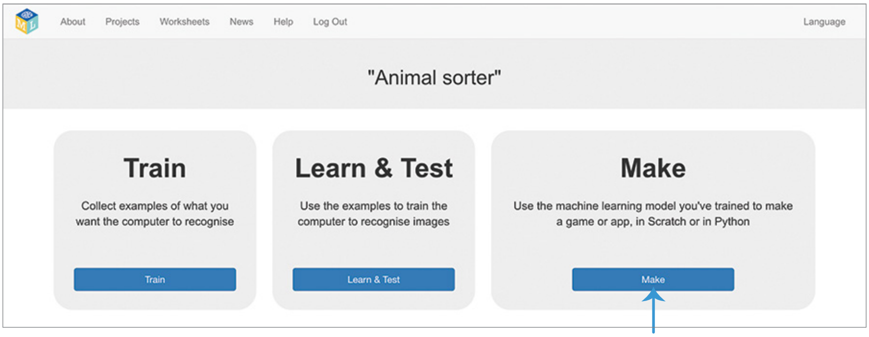


Figure 3-11: My folder for cow and sheep test photos

3. Click **Back to project** in the top-left corner of the screen.
4. Click **Make** (see Figure 3-12).



"Animal sorter"

Train

Collect examples of what you want the computer to recognise

Learn & Test

Use the examples to train the computer to recognise images

Make

Use the machine learning model you've trained to make a game or app, in Scratch or in Python

Figure 3-12: Make is the third phase of an ML project.

5. Click **Scratch 3**, and then click **Open in Scratch 3** to open a new window with Scratch.

   You should see new blocks representing your ML model in the Toolbox, as shown in Figure 3-13.

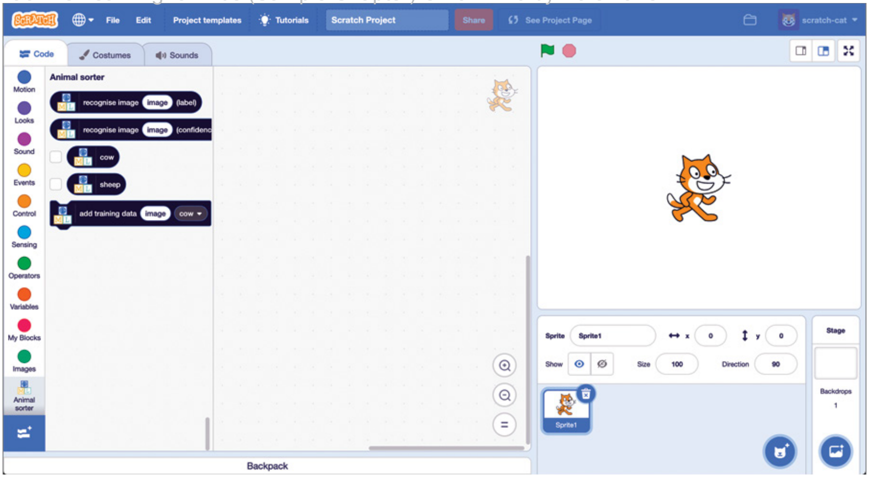
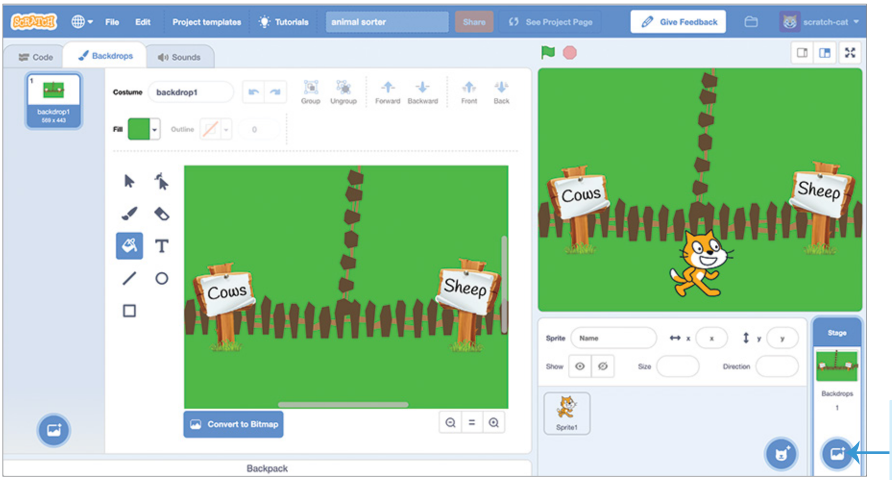


Figure 3-13: New blocks for your ML project will automatically be added to the Scratch Toolbox.

6. Create a backdrop for your game.

   In the bottom-right corner of the Scratch window, move your mouse pointer over the Choose a Backdrop icon, shown in Figure 3-14.



Choose a backdrop

Figure 3-14: My farm background for sorting cows and sheep into two groups

You have a few options here. If you don't like drawing, you can click **Choose a Backdrop** to pick a default background, or click **Upload Backdrop** to use a picture from the internet. To design your own background for your animals, click **Paint** and use the drawing and coloring tools in the *paint editor* to the left of the Code canvas.

No matter which option you choose, be sure to add clearly labeled sections to your backdrop for each type of animal.

I chose farm animals, so I drew a farm scene with signs labeled Cows and Sheep. You can draw something that fits with the animals you chose. For example, if you chose dogs and cats, you could draw a pet shop. If you chose lions and elephants, you could draw a zoo.

7. Click the cat sprite, and then in the bottom-left corner of the screen, move your mouse pointer over the Choose a Costume icon. Click **Upload Costume** as shown in Figure 3-15.
8. Select all of the test photos you saved in step 2 to upload all of them at once.

NOTE *Make sure you're doing this for the sprite, not the backdrop.*

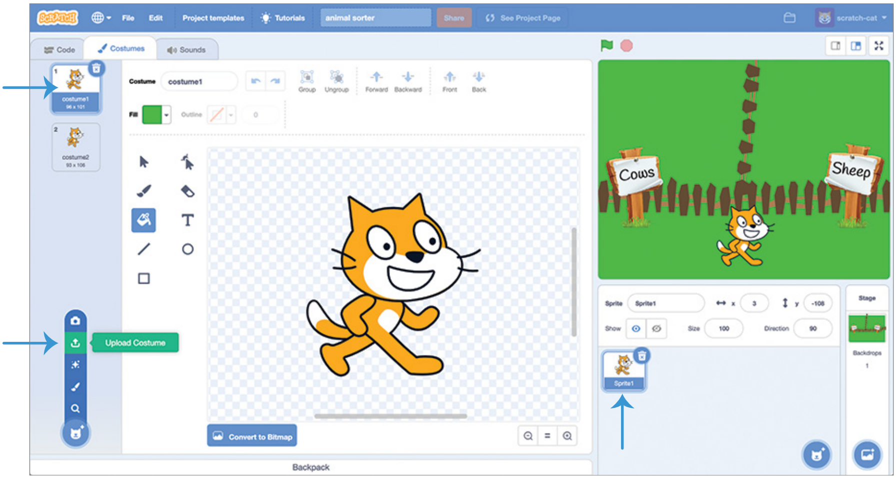


*Figure 3-15: Click **Upload Costume** to add your test photos. You can delete the cat costumes from the costume pane at the left.*

9. If you missed any of your test photos, click Upload Costume again and repeat until you've uploaded all of the test photos from step 2.

We won't need the Scratch cat costumes, so you can delete them. In the *costume pane* at the far left (see Figure 3-15), click the costume and then click the trash can in its top-right corner to delete it.

Make sure that you upload costumes for the same single sprite, as shown in Figure 3-16. Don't upload each photo as a new sprite.

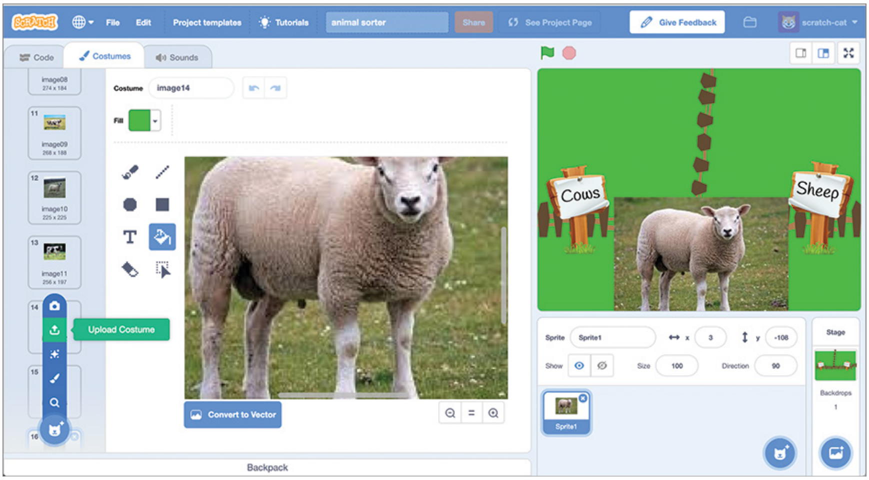
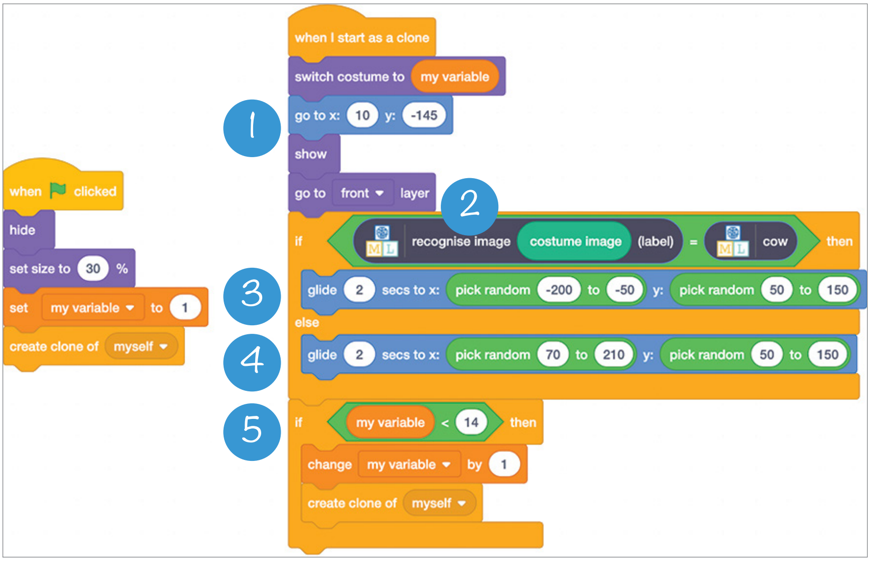


Figure 3-16: Upload additional costumes to the same sprite.

10. Click the **Code** tab and copy the scripts shown in Figure 3-17.



```
when 🏁 clicked
hide
set size to 30 %
set my variable to 1
create clone of myself
```

```
when I start as a clone
switch costume to my variable
go to x: 10 y: -145
show
go to front layer
if recognise image costume image (label) = cow then
    glide 2 secs to x: pick random -200 to -50 y: pick random 50 to 150
else
    glide 2 secs to x: pick random 70 to 210 y: pick random 50 to 150
if my variable < 14 then
    change my variable by 1
    create clone of myself
```

Figure 3-17: Example code for sorting photos of animals

NOTE *If you're not sure how to code in Scratch, read "Coding in Scratch" on page XX in the introduction to this book.*

This code will go through each of your test photo costumes and use your ML model to recognize the animal and move it to the correct section.

The go to x: 10 y: -145 block ❶ sets the starting position for each photo. This script will start each photo at the bottom middle of the screen. Adjust these coordinates to a starting position that fits with your backdrop.

The recognise image block ❷ uses your ML model to recognize the photo.

The glide to x: pick random -200 to -50 y: pick random 50 to 150 block ❸ moves the photo to a random position in the top left of the screen. Adjust these coordinates to the location on your backdrop where you want the photos of your first type of animal to go.

The glide to x: pick random 70 to 210 y: pick random 50 to 150 block ❹ moves the photo to a random position in the top right of the screen. Adjust these coordinates to the location on your backdrop where you want the photos of your second type of animal to go.

The number in if my variable is < 14 ❺ refers to how many test photos you have. Set this value to the number of test costumes you saved in step 2 and uploaded in step 8. I have 14 costumes in my test sprite, so my script goes through 14 test photos.

TEST YOUR MODEL

To test the model you've created, click the Green Flag in the top-left corner, as shown in Figure 3-18. Your model will sort your test photos into two groups of animals.

Count how many photos your model moves to the correct side. That's a simple way to measure how good your project is at sorting photos of your two animals.

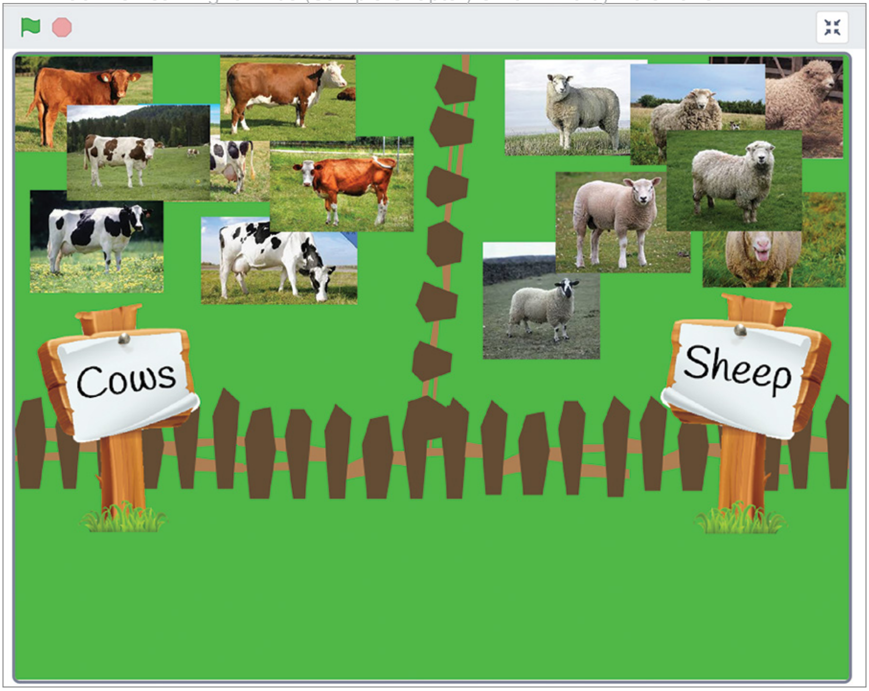


Figure 3-18: Recognizing photos and sorting them into groups

If your model is getting a lot wrong, you can try to improve it using more training examples. Go back to the Train phase and drag in more photos of each of your animals. Then go back to the Learn & Test phase and train the new, improved ML model. Run your Scratch script again to see if the new model sorts more of your test photos correctly.

REVIEW AND IMPROVE YOUR PROJECT

You've successfully trained an ML model to recognize photos of animals! This project isn't based on rules. You haven't described how different animals look or given the computer specific instructions on how to recognize them. Instead you've used ML to train the computer to do it. This approach is known as *supervised learning* because you supervised the process by preparing sets of training examples for the computer to use.

As long as your test photos are similar to your training photos, your model should work. However, if you test your model with pictures that differ somehow from your training photos, you'll probably get different results.

For example, I tried replacing the costumes in my Scratch project with cartoon illustrations of cows and sheep instead of photographs. Then I ran my code again by clicking the Green Flag. As you can see in Figure 3-19, my new model got a lot wrong.

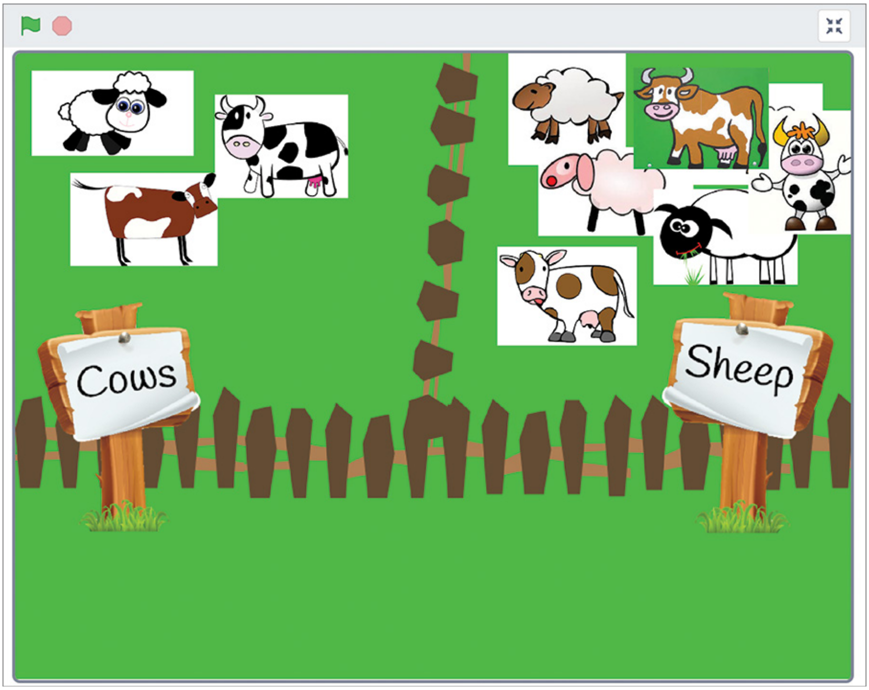


Figure 3-19: ML models get a lot wrong if the test inputs aren't similar to the training inputs.

I got these results because the patterns my model learned to recognize from the training photos weren't useful to help it recognize cartoon drawings. If you want a computer to be able to recognize photographs *and* cartoons, you need to train it with both.

Go back to the Train phase and create a new set of training examples that includes both photographs and cartoons, as shown in Figure 3-20. I collected 10 example photos and 10 example cartoons in each bucket.

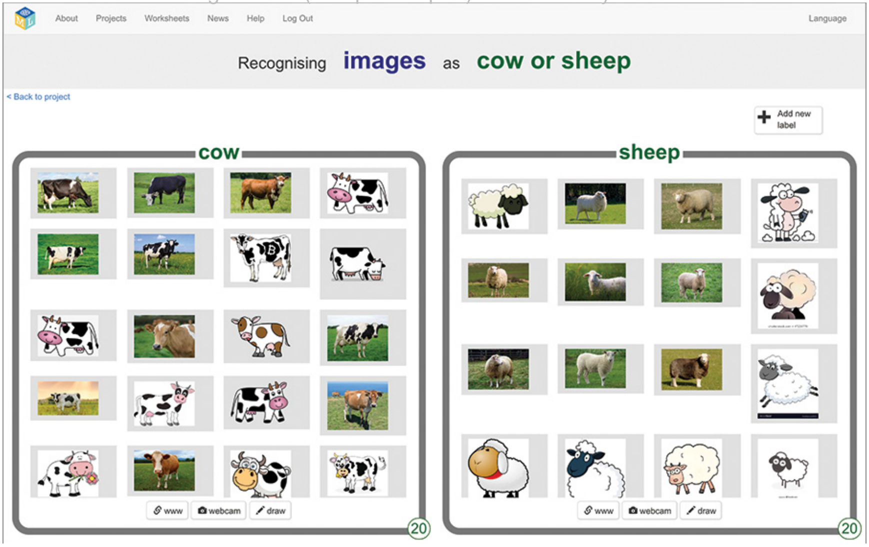


Figure 3-20: Training a computer to recognize both photos and cartoons

Then go back to the Learn & Test phase and train a new ML model using your new set of training examples. This new set should teach the computer to identify patterns in photographs and cartoons so it can recognize both. Figure 3-21 shows how much better my updated ML model performed.

As you can see, the more similar the test images are to the training images that the computer learns from, the better ML models perform.

What other changes could you make to improve your model?

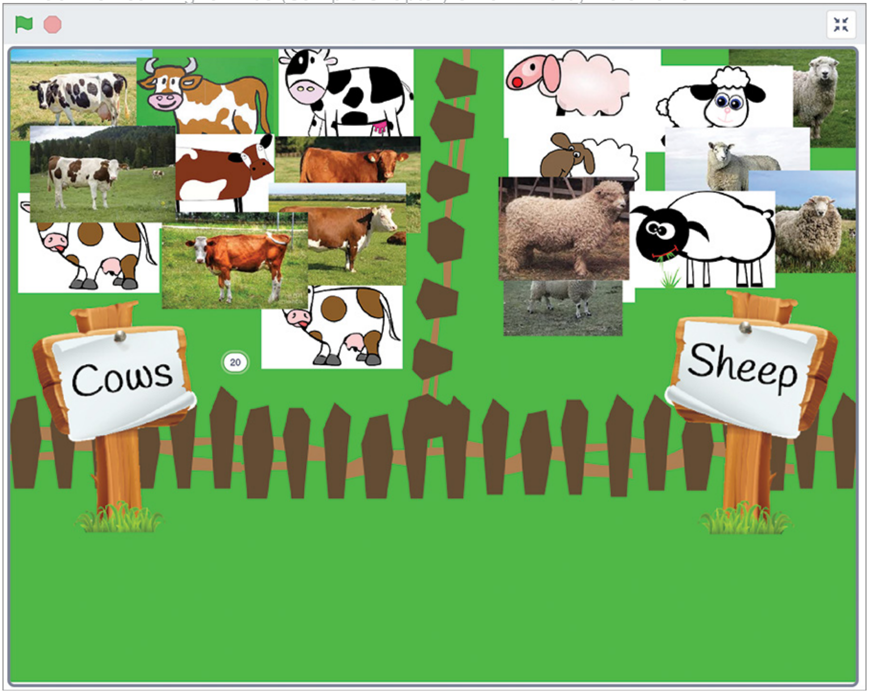


Figure 3-21: Testing with a mixture of photographs and drawings

WHAT YOU LEARNED

In this chapter, you used ML to create an image recognition system capable of recognizing and sorting pictures of animals. You learned some of the key principles in ML projects, such as improving your results by increasing the number of training images and making sure they're similar to the images you want the computer to recognize.

You also learned that we can measure how well an image recognition system performs by trying it out with test pictures and counting how many it identifies correctly. You tried this yourself by creating a project in Scratch to test how well your ML model sorted a group of animal photos.

In the next chapter, you'll train another image recognition system and use it to make a game. You'll also learn about some of the ways that ML projects can go wrong.