



Hand gestures

In this project you will train the computer to recognize different gestures you can make with your hands.

You will assemble a collection of different machine learning models so that they can work together to identify the shape of your hand in the webcam.

The screenshot shows the Scratch programming interface with the title bar "Hand Gestures". The stage area displays a man's face with his right hand raised, palm facing forward, with several colored dots (red, green, blue) tracked on his fingers and thumb. The script area contains the following scripts:

- A main script starting with a "when green flag clicked" hat:
- An "if then" branch:
 - "define recognise gesture" hat
 - "set gesture to [fist v]" control block
 - "if gesture then [recognise numbers (wrist-x) (wrist-y) v]" control block
 - "if gesture then [recognise numbers (wrist-x) (wrist-y) v]" control block
- "define store training data" hat
- "add training data (wrist-x) (wrist-y) (thumb-1-x) (thumb-1-y)" control block

The script palette on the left shows the "Hand gestures" category selected, containing blocks for "recognise numbers", "fist", "okay", "point", "add training data", and "train new machine learning model". The bottom palette shows "Video Sensing" blocks for "when video motion > 10", "video motion on sprite", "turn video on", and "set video transparency to 50".



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

In this project, you will use a pre-trained machine learning model that has been trained to find your hand in a view from your webcam.

This pre-trained model will give you the x,y coordinates of your wrist, and the top and bottom of each of your fingers.



For example:

	x	y
top of thumb	-82	2
base of thumb	-31	-107
top of index finger	-45	119
base of index finger	-25	-11
top of middle finger	-2	138
base of middle finger	9	-11

	x	y
top of ring finger	51	107
base of ring finger	39	-23
top of pinky finger	110	45
base of pinky finger	66	-46
wrist	17	-121

You will use these coordinates to see if your computer can learn to recognize different hand gestures, based on the coordinates of your fingers when you make those gestures.

For example:



fist



okay

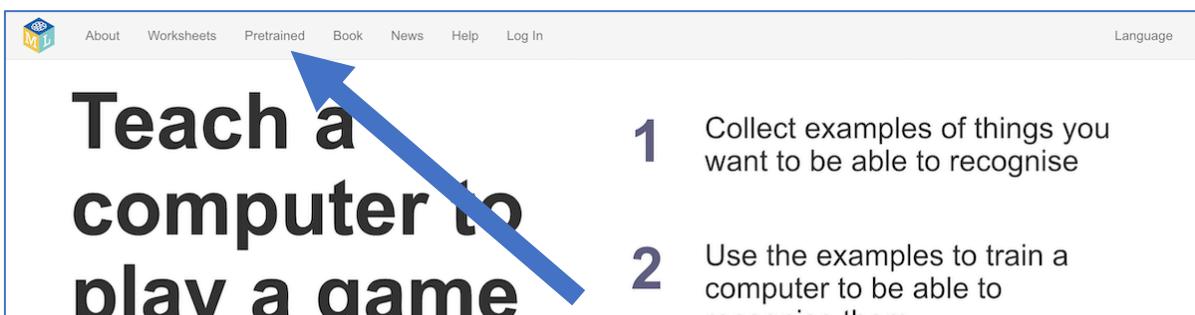


point

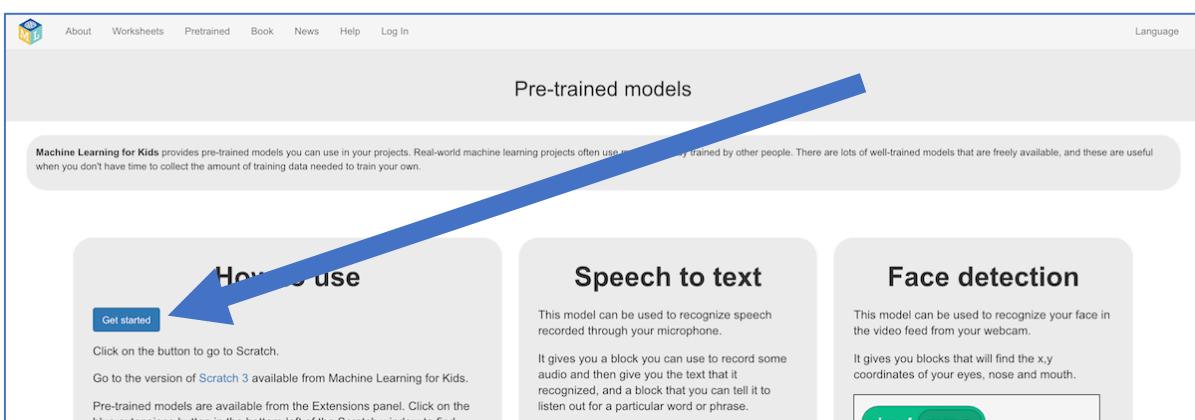
Let's start by trying out the pre-trained model to see what it can do.

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

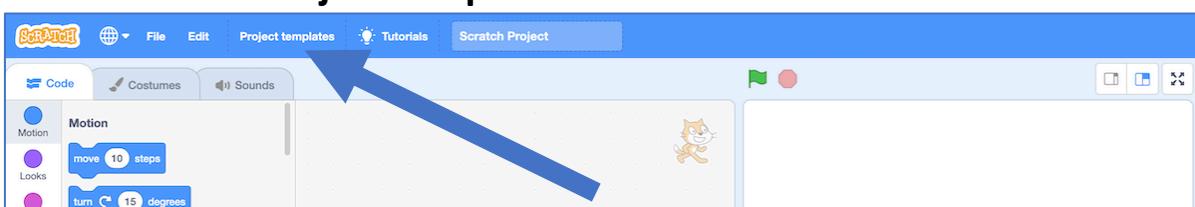
2. Click on “Pretrained”



3. Click on “Get started”



4. Click on “Project templates”



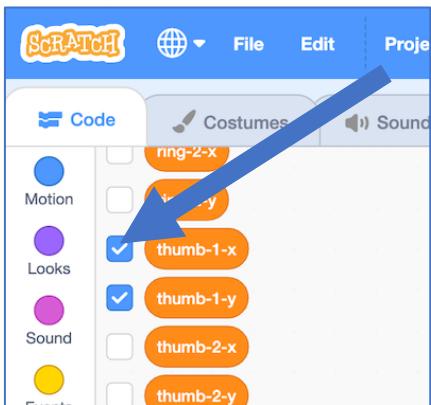
5. Click on the “Hand gestures” template

6. Click on the Green Flag

There might be a short delay while the machine learning model loads

7. Hold one hand up to your webcam

- 8.** Try ticking variables to see the values that the model gives you
Move your hand in the webcam to see how the numbers change



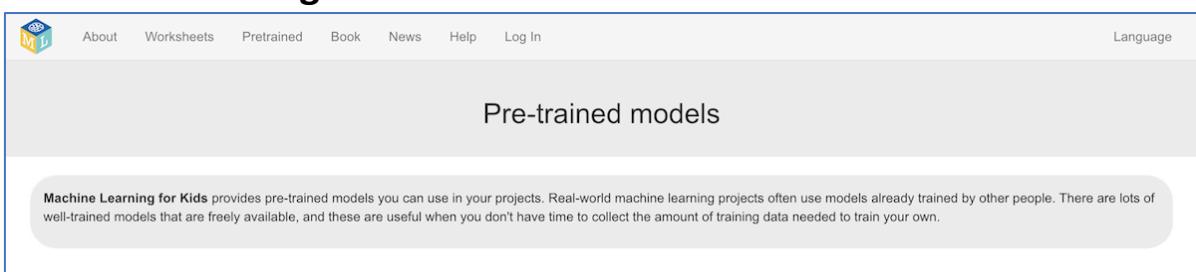
These are the numbers that you will get getting the computer to learn from to recognize different hand shapes and gestures.

Next, you'll prepare somewhere to collect these numbers as training examples.

- 9.** Close the Scratch window

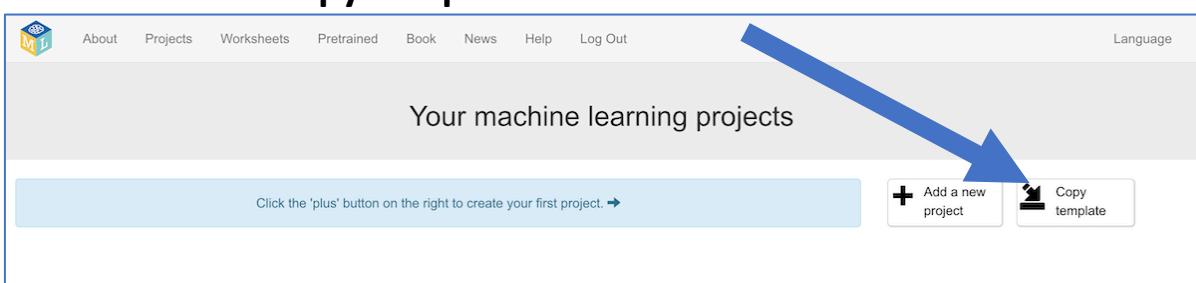
Go back to <https://machinelearningforkids.co.uk>

- 10.** Click on “Log In”

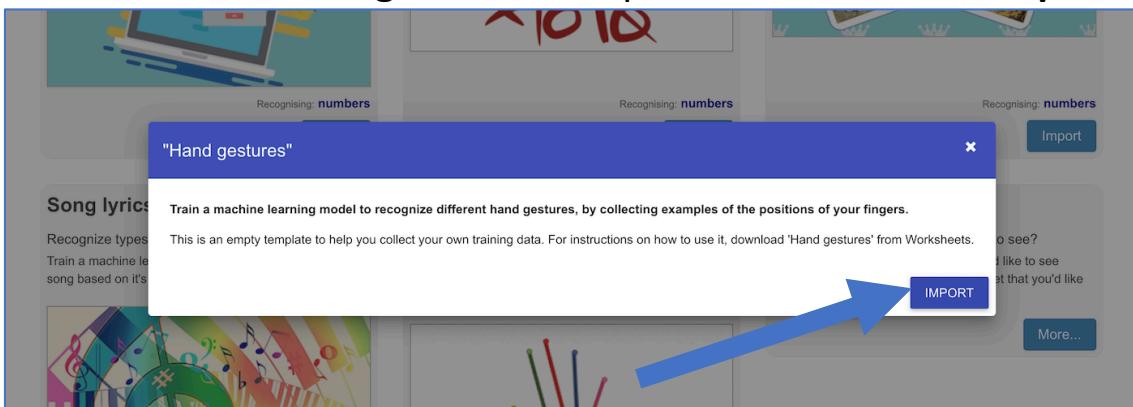


- 11.** Click on “Try it now”

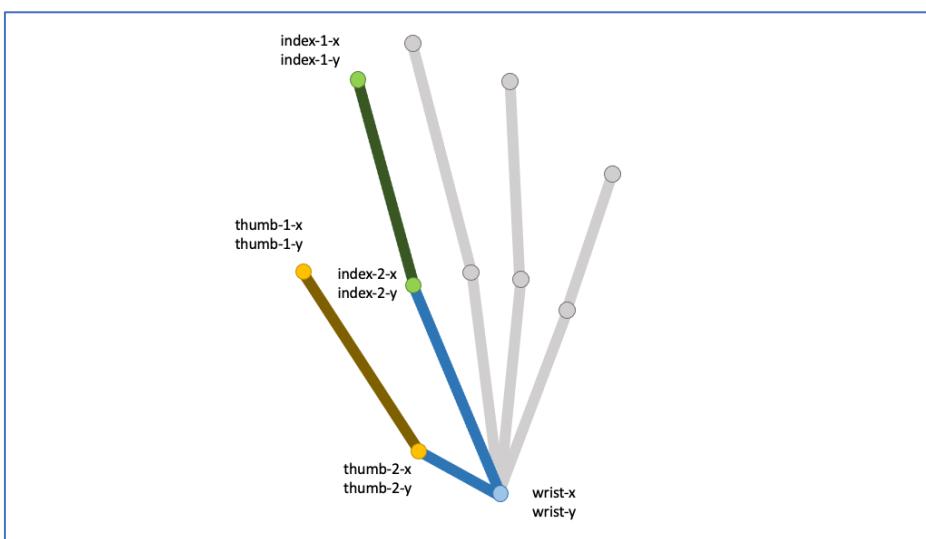
- 12.** Click the “Copy template” button



- 13.** Click the “Hand gestures” template, then click the “Import” button

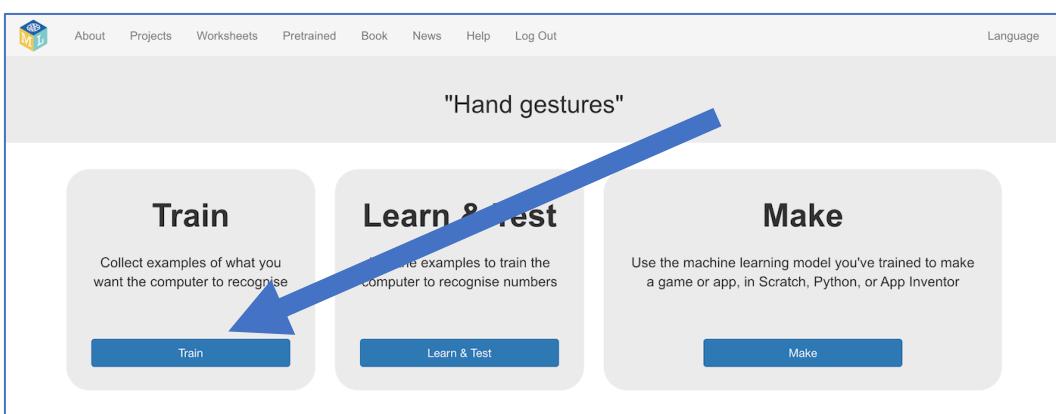


To make your Scratch coding simpler, we'll just be using the coordinates for two of your fingers. That will still give you ten numbers to use in your code!

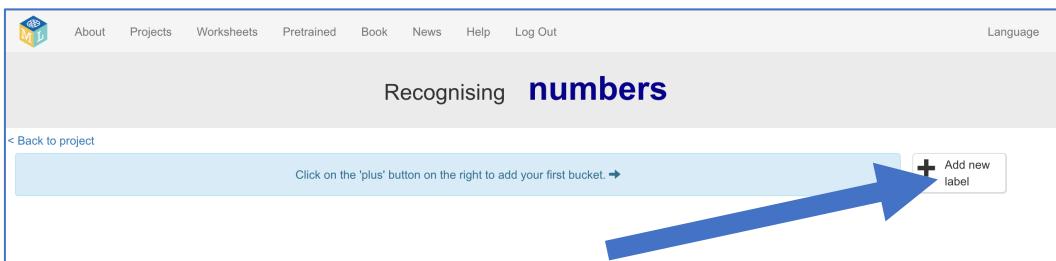


- 14.** You should see “Hand gestures” in your list of projects. Click on it.

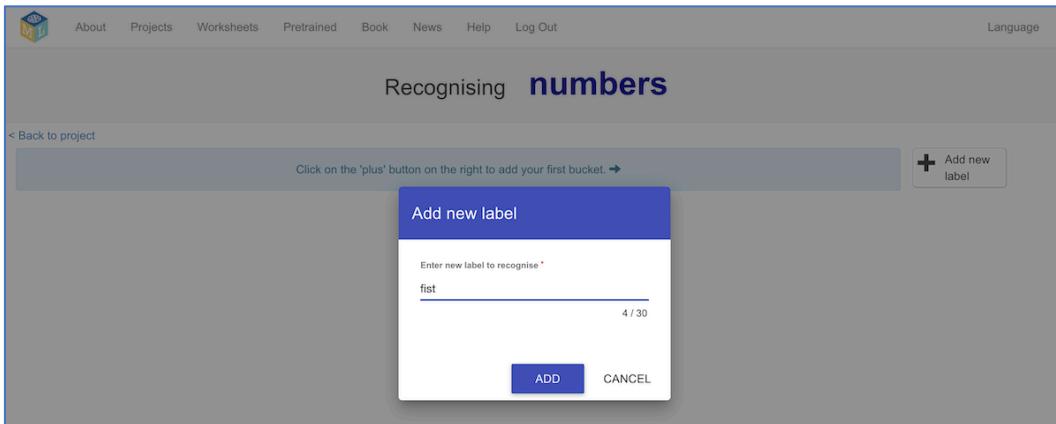
- 15.** Click the Train button.



16. Click on “+ Add new label”

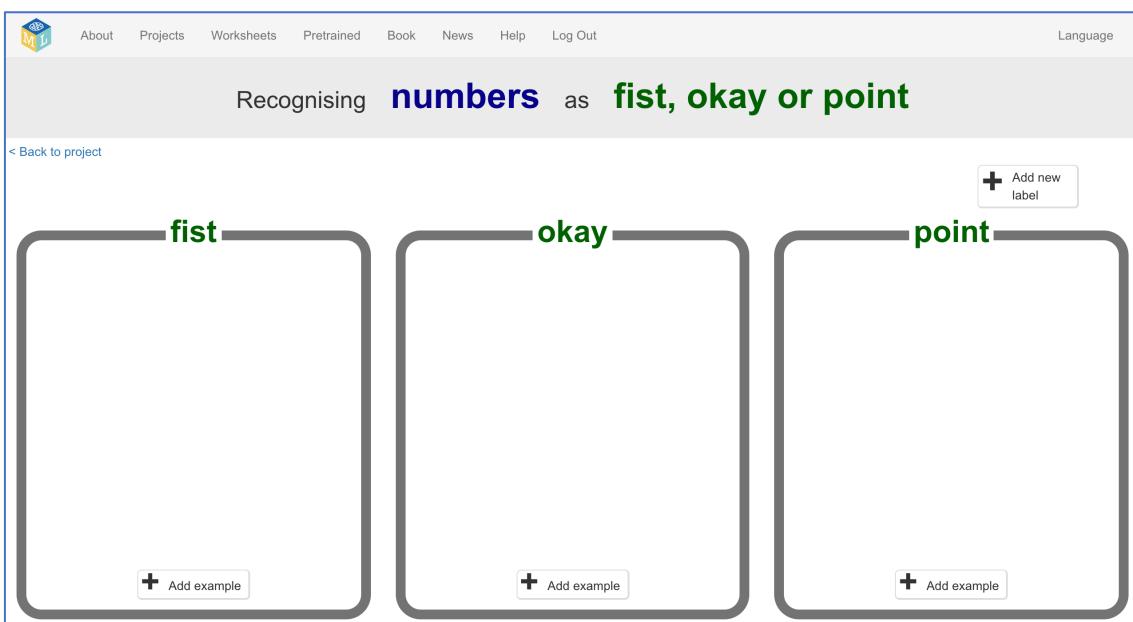


17. Type in “fist” and then click on “Add”.



18. Do that again to create a second bucket called “okay”. Do that a third time to create a bucket called “point”.

If you use different hand-gestures for your project, choose hand-shapes that can be recognised from the two fingers we’re using (thumb and index finger).



These are where you store training examples for the computer to learn from.

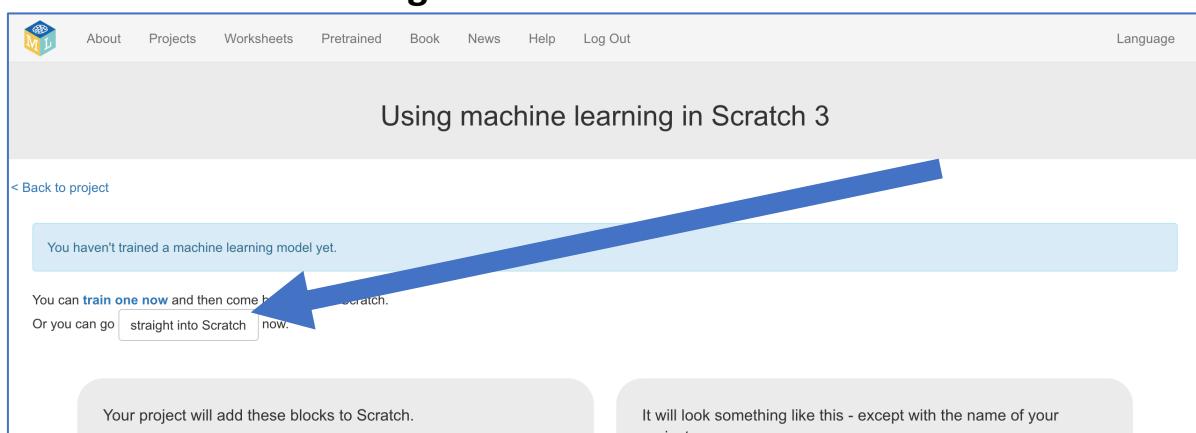
Next, you'll use Scratch to collect some examples.

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

20. Click on the “Make” button

21. Click on the “Scratch 3” button

22. Click on the “straight into Scratch” button



23. Click on “Project templates” (as you did before)

The difference is that this time you will have extra blocks to use for your machine learning buckets.

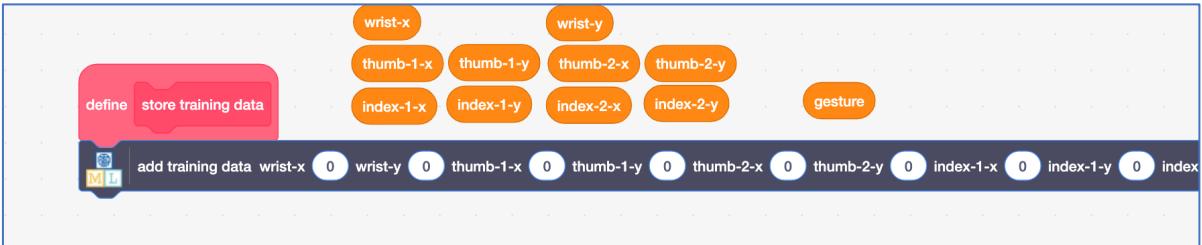
24. Click on the “Hand gestures” template (as you did before)

25. Find the placeholder for the “store training data” script

You should find it on the sprite called “main”



26. Add a block from your machine learning project to store examples



27. Copy in the orange variable blocks into the matching spaces on the "add training data" block

Do this slowly and carefully!

There are a lot of blocks to add. (This is why we're only using ten numbers for this project instead of all twenty-two)



Don't forget to use the gesture block for the last menu space.

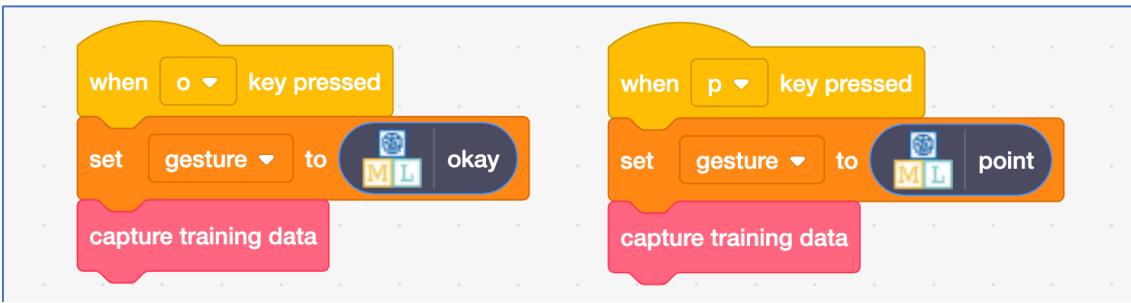


28. Create a script to collect a training example for your first gesture

I've used the "f" key because "fist" starts with f. If you're using different hand gestures for your project, you could pick a different key.

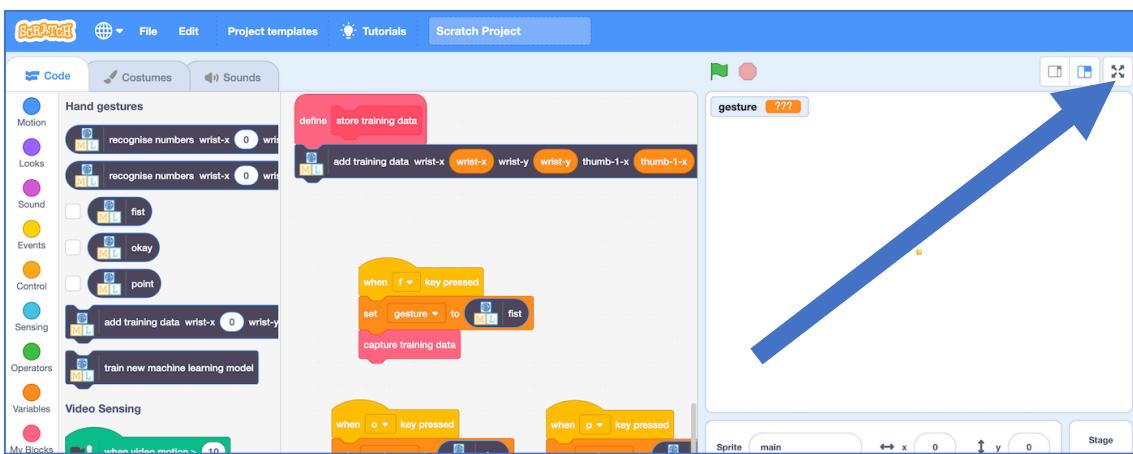


29. Create similar scripts for your other hand gestures



It's time to collect some training examples.

30. Click the full-screen button



31. Click on the Green Flag

32. Make your hand into the shape of a fist (or your first gesture) in the webcam, and wait until you think the dots have correctly found your hand *It works better if only one of your hands is visible in the webcam.*

33. Press the F button on your keyboard

(Or whichever key you picked for your first gesture.)

The dots stay still while coordinates are added to the “fist” training bucket

34. Repeat steps 32-33 nine more times

Try to vary the position of your hand in the webcam (for example, get some examples with your hand on the left side, and some on the right side) and vary how close your hand is to the webcam in each example.

35. Check the training tool page where you created the buckets to see the examples you've collected

Click on the “< Back to project” link and then click on “Train” (Don’t close the Scratch window!)

The screenshot shows the 'Recognising numbers as fist, okay or point' interface. There are three main sections: 'fist', 'okay', and 'point'. The 'fist' section contains 10 examples with detailed coordinate data for each hand joint. The 'okay' section contains 10 examples. The 'point' section is empty. Each example has a 'Delete' button and a '+ Add example' button.

36. Go back to the Scratch window

37. Collect training examples for your second hand gesture

*As before, use your Scratch project but this time press the **O** key on your keyboard while making an “okay” gesture (see page 2 for an example). Collect ten training examples of this hand shape.*

The screenshot shows the 'Recognising numbers as fist, okay or point' interface. The 'okay' section now contains 10 examples, while the 'fist' and 'point' sections remain empty. Each example has a 'Delete' button and a '+ Add example' button.

38. Repeat for your final hand gesture

Use the **P** key to collect examples while pointing (see page 2 for example)
As before, try to get a variety of locations (where your hand is in the screen) and sizes (how close your hand is to the webcam) in your training examples.

The screenshot shows the training tool interface with three main sections: 'fist', 'okay', and 'point'. Each section contains a grid of hand gesture examples. Each example is represented by a small thumbnail and a set of numerical coordinates for the wrist and fingers. At the bottom of each section, there is a '+ Add example' button. The 'okay' section is highlighted with a large blue arrow pointing towards the 'Learn & Test' button in the 'Learn & Test' section below.

Category	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8	Example 9	Example 10								
fist	wrist-x: 100.91887196383067 wrist-y: -50.5127610244686 thumb-1-x: 44.4226811383638 thumb-1-y: 41.74769420774538 thumb-2-x: 52.03814486731397 thumb-2-y: 50.20703542330101 index-1-x: 52.4914571347390101 index-1-y: 42.142829422065546 index-2-x: 55.215359911285645 index-2-y: 27.8068737731748	wrist-x: 124.15745423978603 wrist-y: -58.030855993713144 thumb-1-x: 44.2226811383632 thumb-1-y: 49.5716742145588 thumb-2-x: 51.2550371628568 thumb-2-y: 50.207035423301146 index-1-x: 52.4914571347390101 index-1-y: 42.142829422065546 index-2-x: 55.215359911285645 index-2-y: 27.8068737731748	wrist-x: 74.289455929776163 wrist-y: -44.52370996565658 thumb-1-x: 47.40538023377916 thumb-1-y: 45.047303216424325 thumb-2-x: 52.03814486731397 thumb-2-y: 50.207035423301693 index-1-x: 51.733675301008986 index-1-y: 45.865243421809886 index-2-x: 52.237624420606935 index-2-y: 52.3599339983808	wrist-x: 106.45303783513913 wrist-y: -39.522216712655904 thumb-1-x: 46.3991278059956 thumb-1-y: 45.23750372282195 thumb-2-x: 53.09787159890106 thumb-2-y: 51.021310203521693 index-1-x: 51.733675301008986 index-1-y: 47.567093217105986 index-2-x: 53.1633075726009 index-2-y: 52.3599339983808	wrist-x: 81.21413801425444 wrist-y: -54.10749898036778 thumb-1-x: 45.23750372282195 thumb-1-y: 53.1712413598725054 thumb-2-x: 51.021310203521693 thumb-2-y: 52.3599339983808 index-1-x: 51.58013723309664 index-1-y: 74.3489969194358 index-2-x: 52.3599339983808 index-2-y: 52.3599339983808	wrist-x: -27.881915464619993 wrist-y: -117.10509201779428 thumb-1-x: 45.23750372282195 thumb-1-y: 54.076865212598915 thumb-2-x: 59.24480116940859 thumb-2-y: 59.57416496764709 index-1-x: 55.9117632670602034 index-1-y: 78.4703346853174 index-2-x: 61.68356218806068 index-2-y: 83.5689294321778	wrist-x: -108.43741640565366 wrist-y: -113.6168441438319 thumb-1-x: -101.01153881891849 thumb-1-y: -38.82773800178936 thumb-2-x: -138.60309316310344 thumb-2-y: -97.59813777121837 index-1-x: -157.87867652360688 index-1-y: -47.065386219624 index-2-x: -141.61181847826413 index-2-y: -43.45432878517482	wrist-x: -42.01028020098538 wrist-y: -15.699173148842428 thumb-1-x: -101.01153881891849 thumb-1-y: -71.540701622385473 thumb-2-x: -72.76534752273228 thumb-2-y: -1.06271900873557 index-1-x: -15.886858533607622 index-1-y: -75.08160212817291 index-2-x: -85.44808735773222 index-2-y: -60.3791744380346	wrist-x: -108.78695711972637 wrist-y: -191.6891673148842428 thumb-1-x: -101.01153881891849 thumb-1-y: -71.540701622385473 thumb-2-x: -131.31167699473886 thumb-2-y: -112.131167699473886 index-1-x: -168.40127807070675 index-1-y: -15.9651876828178627 index-2-x: -122.30798211902482 index-2-y: -37.9147013516323	wrist-x: -102.616941692368941 wrist-y: -191.6891673148842428 thumb-1-x: -176.04769887934984 thumb-1-y: -43.4445759275337 thumb-2-x: -131.31167699473886 thumb-2-y: -127.131167699473886 index-1-x: -168.40127807070675 index-1-y: -15.9651876828178627 index-2-x: -19.634279375188723 index-2-y: -37.9147013516323	wrist-x: -152.480101296537048 wrist-y: -191.6891673148842428 thumb-1-x: -161.03756479372039 thumb-1-y: -27.04911159357064 thumb-2-x: -124.54936852353869 thumb-2-y: -58.880411465116225 index-1-x: -15.948841814450134 index-1-y: -48.30753560364354 index-2-x: -16.31391561492462 index-2-y: -81.37104893712382	wrist-x: -37.37715392086576 wrist-y: -20.712664548680336 thumb-1-x: -89.0428280310329 thumb-1-y: -106.08552362469697 thumb-2-x: -55.41552313890148 thumb-2-y: -50.30895823473611 index-1-x: -180.2987847065593 index-1-y: -58.79691285184613 index-2-x: -77.34273676049423 index-2-y: -61.16764352768354	wrist-x: 74.78406338013536 wrist-y: -93.84297972883343 thumb-1-x: 55.042648577431294 thumb-1-y: 2.97157620626658 thumb-2-x: 50.21837690738668 thumb-2-y: -64.26149803017321	wrist-x: -37.48430978203687 wrist-y: -48.28430978203687 thumb-1-x: 7.28802340595455 thumb-2-x: 19.3661949037864815 thumb-2-y: -22.3408601397937010	wrist-x: -46.26959474404025 wrist-y: -57.24228771268678 thumb-1-x: 54.65801211925444 thumb-1-y: -11.792716891092482 thumb-2-x: -22.3408601397937010 thumb-2-y: -22.3408601397937010	wrist-x: -55.106513258379294 wrist-y: -24.613119894901894 thumb-1-x: -18.28727306230988 thumb-1-y: -67.8108427018567 thumb-2-x: -82.19667411530378 thumb-2-y: -1.192335857198072	wrist-x: 141.7539905192138 wrist-y: -33.82658599502145 thumb-1-x: 75.41653723939343 thumb-1-y: -49.974130879702045 thumb-2-x: 123.07556949243269 thumb-2-y: -4.75032132665657	wrist-x: -37.4886380157086576 wrist-y: -20.712664548680336 thumb-1-x: -89.0428280310329 thumb-1-y: -74.6474216712799 thumb-2-x: -72.2879352285664 thumb-2-y: -116.17116714609546 index-2-y: 16.57827007436299

You've collected enough training examples to get started.

Next, you'll get the computer to learn to recognize hand gestures by using these examples to train a machine learning model.

39. In the training tool window, click on the “< Back to project” link Leave the Scratch window open, as you'll be using it again in a moment.

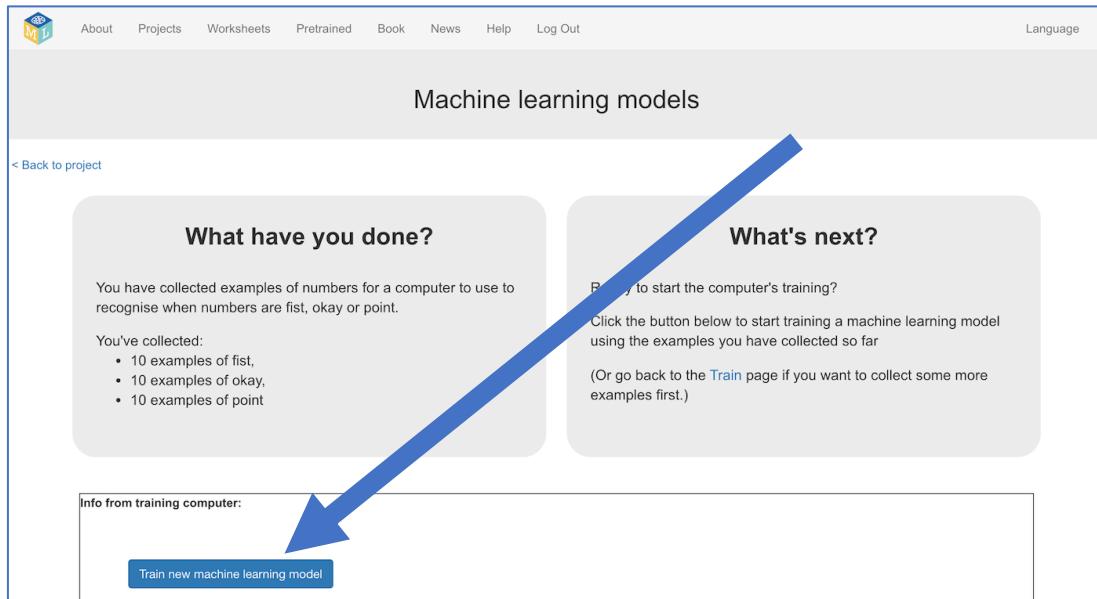
40. Click on the “Learn & Test” button

The screenshot shows the training tool interface with three main sections: 'Train', 'Learn & Test', and 'Make'. The 'Learn & Test' section is highlighted with a large blue arrow pointing towards the 'Learn & Test' button. Each section has a brief description and a corresponding button: 'Train', 'Learn & Test', and 'Make'.

"Hand gestures"

Section	Description	Button
Train	Collect examples of what you want the computer to recognise	Train
Learn & Test	Use the examples to train the computer to recognise numbers	Learn & Test
Make	Use the machine learning model you've trained to make a game or app, in Scratch, Python, or App Inventor	Make

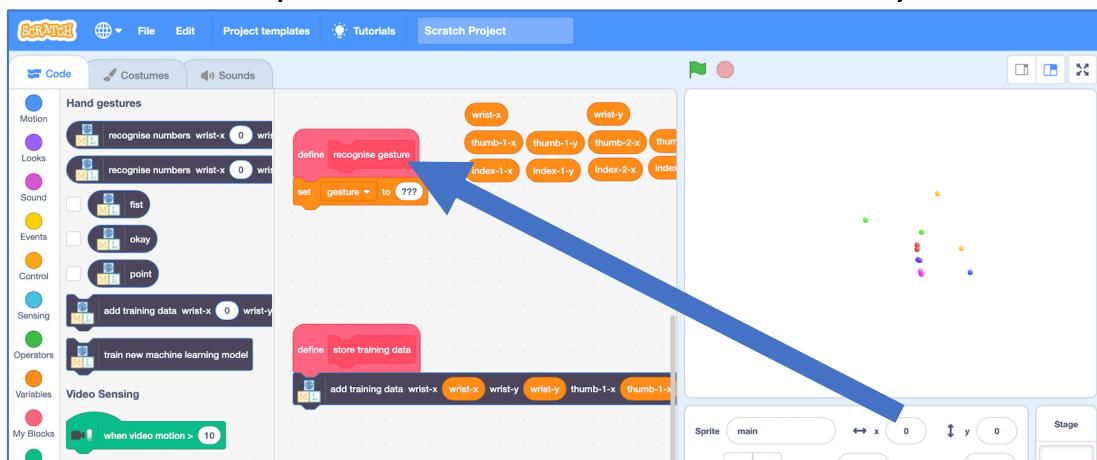
41. Click on the “Train new machine learning” button



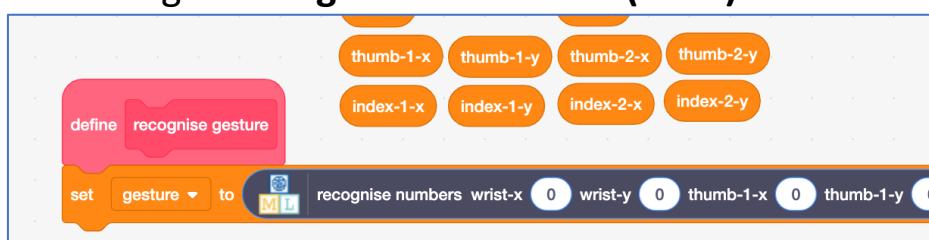
42. Go back to the Scratch window

43. Click on the full-screen button so you can see the code again

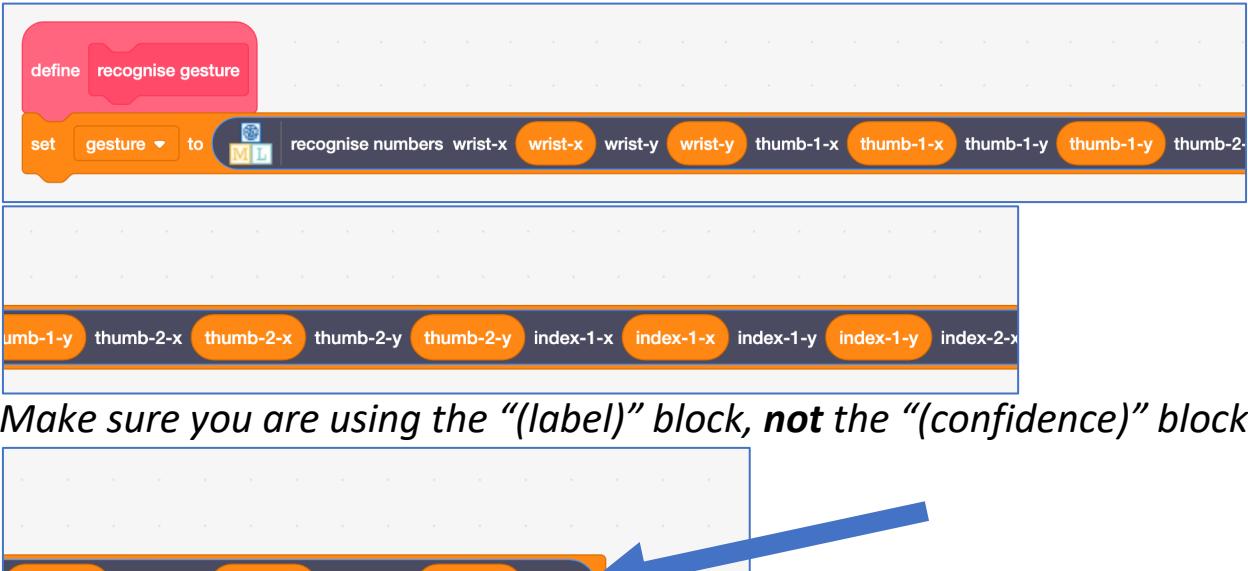
44. Find the placeholder for the “recognise gesture” script *It is also in the sprite called “main”, above the code you worked on before.*



45. Drag a “recognise numbers … (label)” block into the gap with ???



- 46.** Copy in the orange variable blocks into the matching spaces on the “recognise numbers” block
Do this slowly and carefully!



Your project is now ready to go. It is time to test your machine learning model!

- 47.** Click the **full-screen** button
- 48.** Click on the **Green Flag**
- 49.** Make one of your gestures and hold your hand in front of the webcam, and wait until you think the dots have correctly found your hand
- 50.** Press the **space bar** on your keyboard
The machine learning model's prediction will be displayed in the top-left.

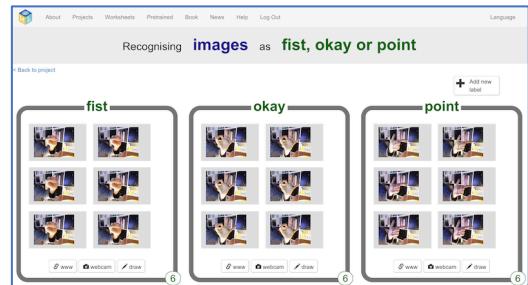
Does your machine learning model correctly recognize your hand gestures?

You might find that, with only ten examples of each shape, the model is still making mistakes.
If so, **repeat steps 31-42** (to collect twenty examples of each gesture and then train a new, improved machine learning model).

What have you done so far?

You've used a combination of different machine learning models (two pre-trained models, and one new custom model that you trained yourself) to make an artificial intelligence project that can do something new: recognize your hand gestures.

Instead of using a combination of machine learning models in this way, you could have just trained one new image machine learning model to recognize your hand shapes.



(If you'd like to see how that approach works, you can try the “Rock, Paper, Scissors” worksheet which uses this technique.)

Why do you think the technique you used in this project might be better?

Input	Model	Training challenge	Output
 picture of anything that includes a hand somewhere	Hand detection	Difficult Creating this model will involve a lot of training data with a large range of backgrounds (different types of indoors rooms, outdoor scenes and environments, etc.), with a variety of hands (different sizes, colours, etc.), in different positions Doing this is a lot of work!	 location of something that looks like a hand
 picture of a hand	Landmark detection	Difficult Creating this model will involve training data with a variety of hands in a variety of positions. Doing this is a lot of work!	 location of finger joints
 coordinates of finger joints	Gesture recognition	Simple Creating this model will involve recognizing relationship between a small collection of numbers. It doesn't need to deal with different backgrounds, or colours, or shadows, etc. This can be done with a small amount of training data.	“okay”

The table on the previous page describes what you did for this project.

Three machine learning models were used:

- A pre-trained hand detection model
- A pre-trained hand landmark detection model
- A custom gesture recognition model that you trained yourself

This is a common approach to machine learning projects:

- 1) break the overall thing that you want the project to do into separate steps
 - Tasks that other people have wanted to do (e.g. find a hand)
 - Tasks that other people have spent time and effort to train a machine learning model for
 - Tasks that aren't particularly unique to your project
- 2) use pre-trained models for the steps that are common tasks
- 3) train a new custom model for the steps that are unique to your project

Doing all of this in a single model means that your model has to learn how to do all of these separate steps by itself.

Input	Model	Training challenge	Output
 picture of anything that includes a hand somewhere	Hand detection	<p>Very difficult</p> <p>This model needs to recognize a variety of hands (different sizes, colours, etc.) in a variety of locations, when seen against a large range of backgrounds (different types of indoors rooms, outdoor scenes and environments, etc.), and making a variety of shapes/gestures.</p> <p>That is a lot of things for one model to learn, so a lot of training examples will be needed.</p>	"okay"

But if you only use pre-trained models, then this limits you to only being able to make AI projects that someone else has already thought of.

What have you done?

You've seen how artificial intelligence projects can combine pre-trained models with custom models to get the best of both worlds: the extensive training data and accuracy of pre-trained models, and ability to make new and unique models for new project ideas.

Ideas and Extensions

Now that you've finished, why not give one of these ideas a try?

Or come up with one of your own?

Try different hand gestures

Try making the project with your own ideas for hand gestures.

Try using an images model

Try making this project using a single, custom, images machine learning model. Which technique is easier for you to train? Which approach do you think works the best?

Use different fingers

Instead of using locations for your wrist, thumb and index fingers, you could create the numbers project yourself and choose the number fields that you want to use.