

Machine Learning Workshop



**University of
Sunderland**

In this activity you will train the computer to recognise images from a webcam.

We will use a web browser application called *Teachable Machine* which is a fun, easy way to create machine learning models - no coding required and by the end of this activity **you** will have trained the computer to recognise different sweets.



WARNING

Allergy statement: During this activity you may come into contact with WHEAT, EGGS, TREE NUTS, and MILK. For more information, please speak with a member of staff.

Let's get going!



**University of
Sunderland**

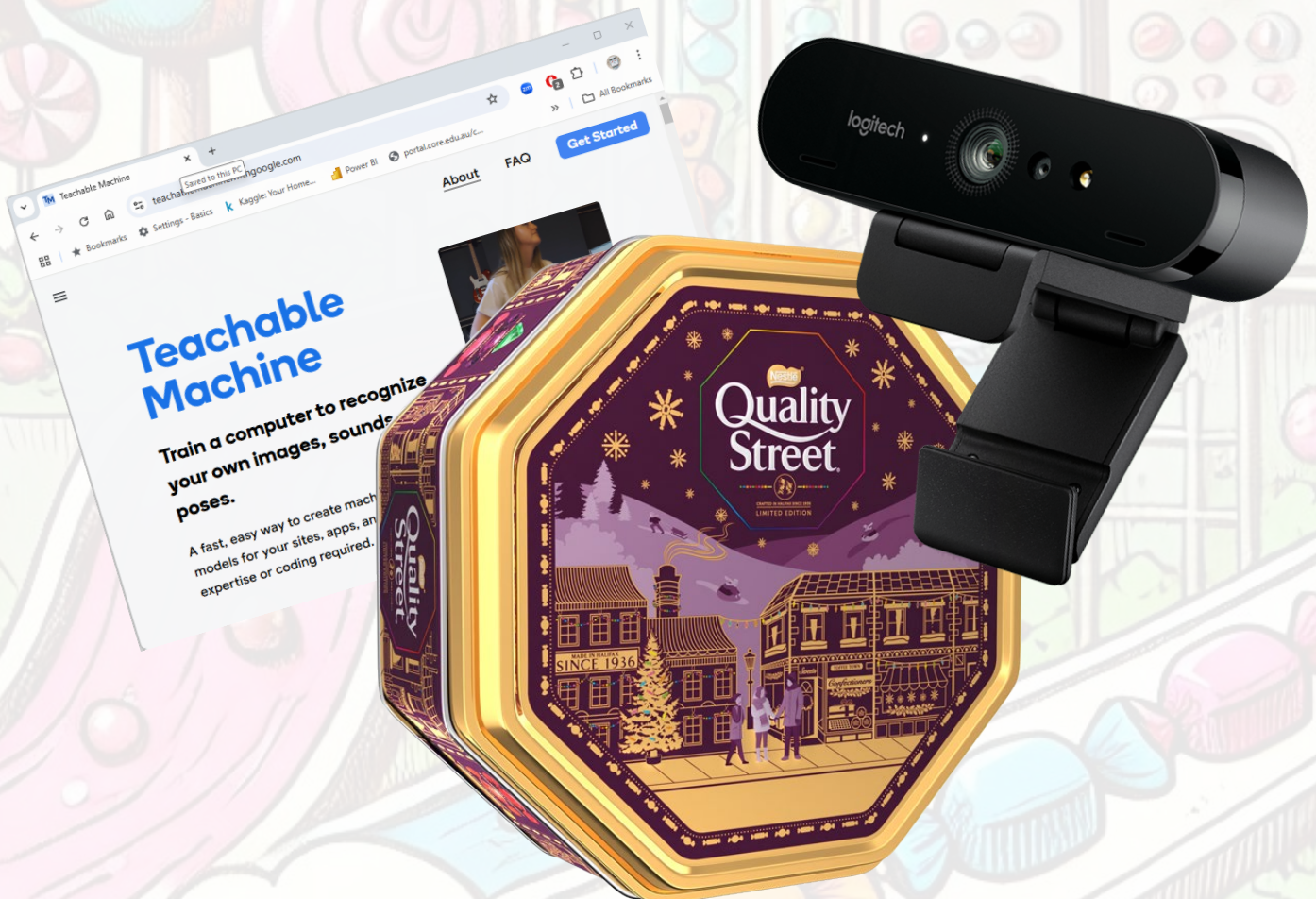
What do we need!

For this activity we need the following:

- A **web browser** (we recommend you use Chrome).
- A **webcam** connected to the computer.
- Some **sweets**. 🍬

The lab computers should already be setup, but if you get stuck at any point, ask for help. We are here to help you and don't worry if you don't get everything finished in this session.

Source files: <https://github.com/neliot/sweeties>



Let's get going!



**University of
Sunderland**

Gather your items

Gather up the sweets that you're using - you'll need one of each type of sweet. This is what we are going to use AI to carry out some classification!

In this guide we use Quality Street chocolates, but yours might be different.

Resist the temptation! You can have some later!



Let's get going!



**University of
Sunderland**

Gather your items

Identify each sweet and allocate an appropriate class/label.
You might need to look at the key on the side of the box.

No Really! - resist the temptation! You can have some later!

Item	Label/class
	Green Triangle
	Toffee Penny
	Coconut Eclair
	Strawberry Delight
	Orange Crunch
	Milk Choc Block
	Orange Cream
	Toffee Finger
	Purple One
	Fudge
	Caramel Swirl

Build the model!



**University of
Sunderland**

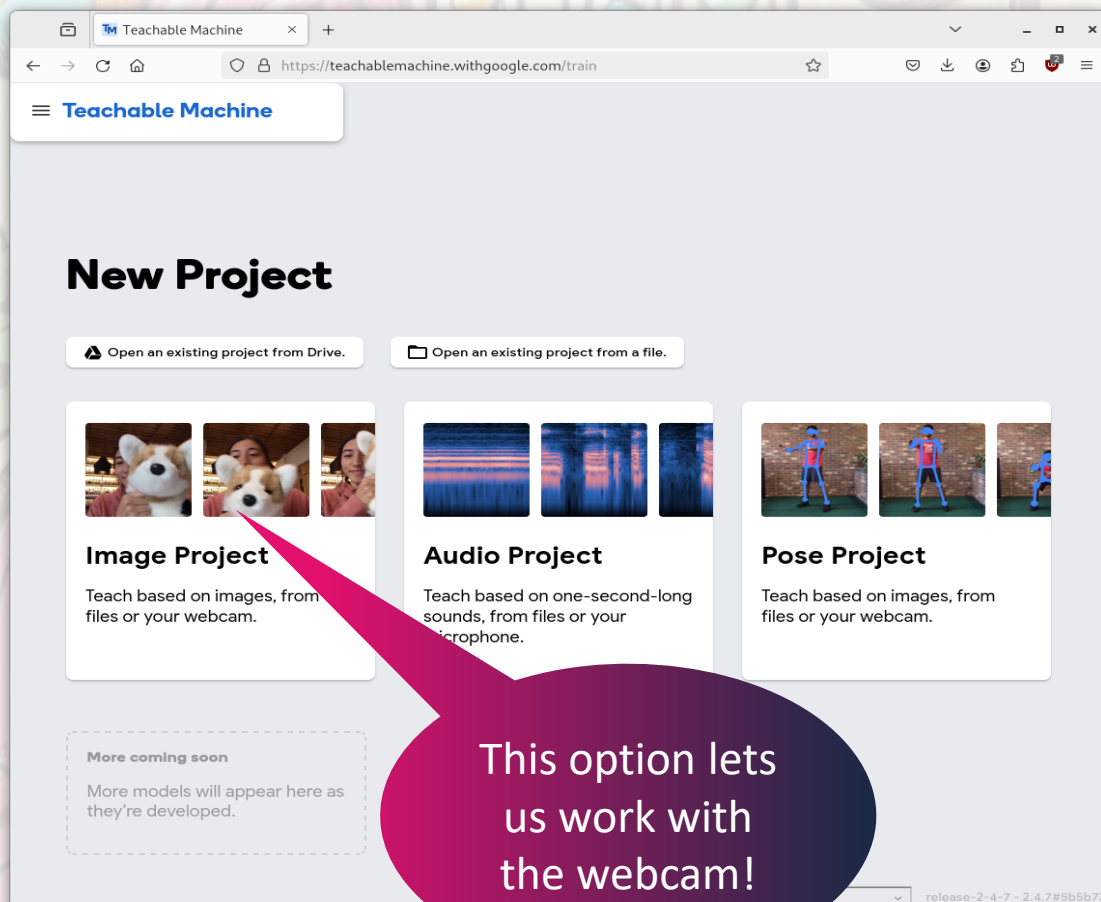
Let's get classifying

Time to show the computer the sweets and train a model. We're going to use a website called Teachable Machine which has been developed by Google. It makes it easy to create models.

Open Teachable Machine in another window by right clicking on the following link below and selecting open a new window.

<https://teachablemachine.withgoogle.com/>

You should see a screen that looks like this:



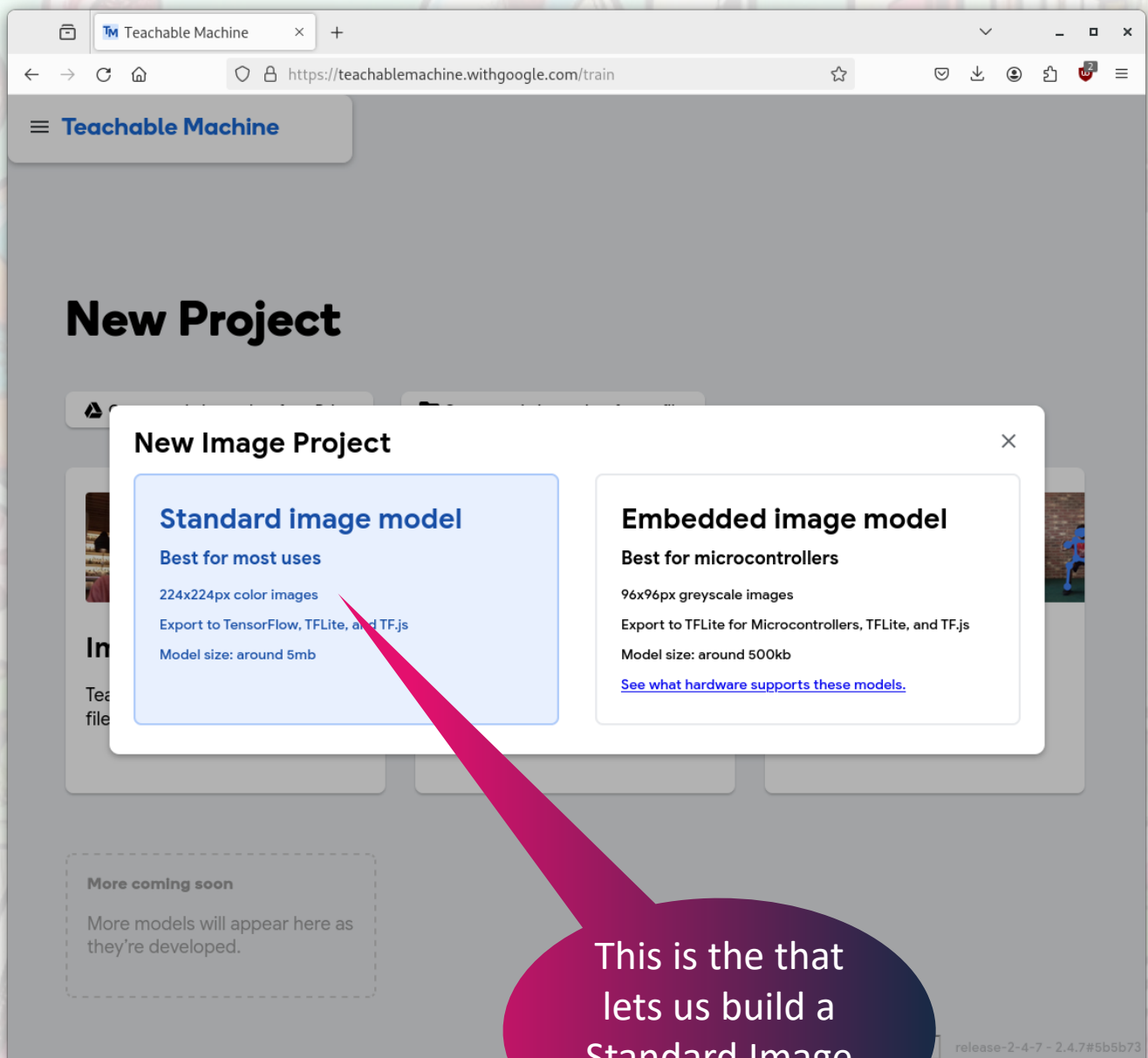
Build the model!



University of
Sunderland

Let's get classifying

Hopefully you are now seeing a screen like this!



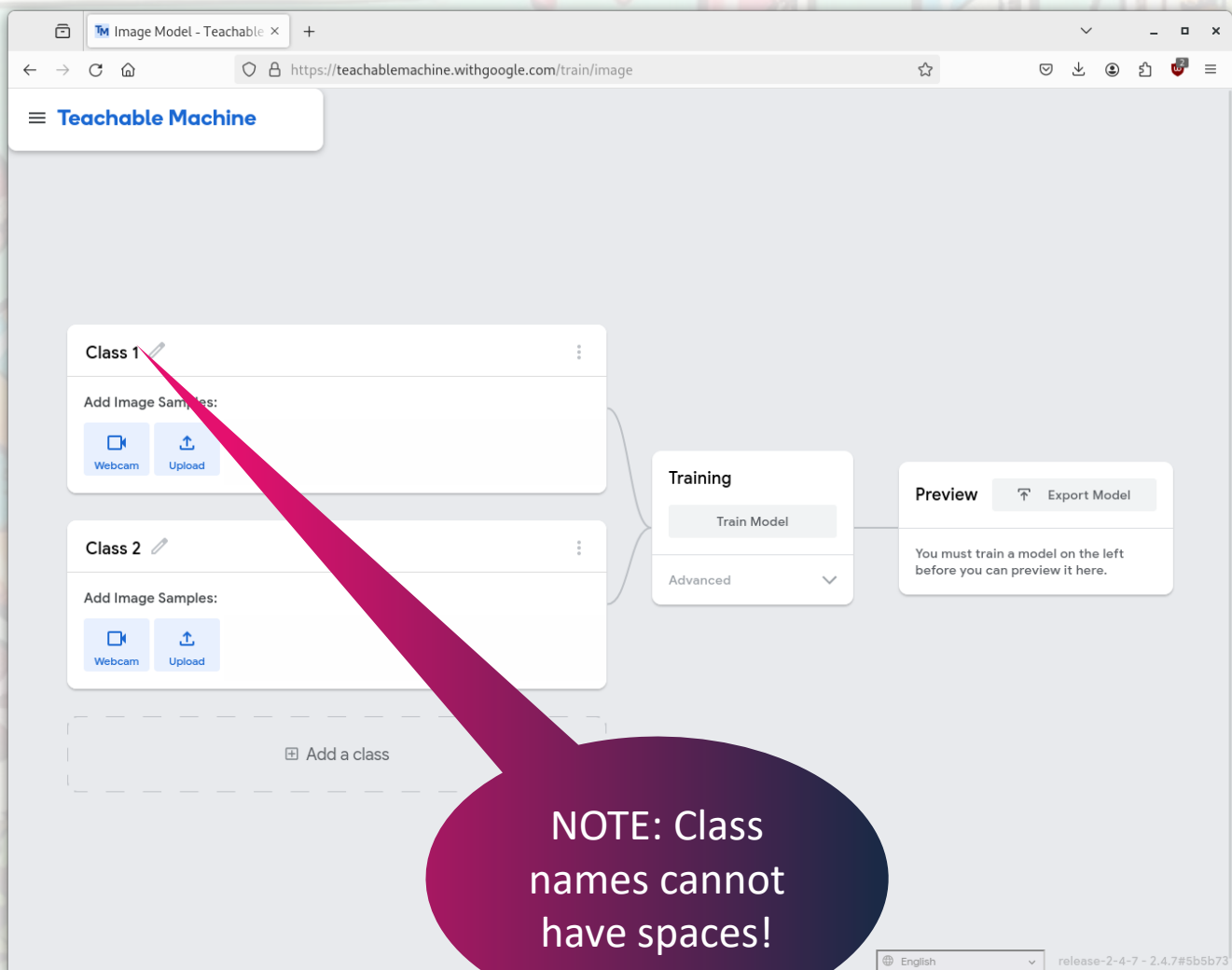
Build the model!



University of
Sunderland

Let's get classifying

Your screen should look something like below:



Rename **Class 1** by clicking the edit icon and name it to your first sweet. In our example its the Quality Street Green Triangle.

Build the model!



University of
Sunderland

Let's get classifying

The screenshot shows a web interface for training a model. On the left, there are two class configuration panels. The first panel is for 'GreenTriangle' and the second is for 'Class 2'. Each panel has a title bar with an edit icon and a three-dot menu, followed by a section 'Add Image Samples:' with 'Webcam' and 'Upload' buttons. Below these panels is a dashed box with a '+ Add a class' button. On the right, there is a 'Training' panel with a 'Train Model' button and a dropdown menu currently set to 'Advanced'. To the right of the training panel is a 'Preview' panel with an 'Export Model' button and a message: 'You must train a model on the left before you can preview it here.'

Do the same for **Class 2** and rename it to your next sweet. In our example it's the classic toffee penny (ToffeePenny)

This screenshot is similar to the previous one, but the second class is now named 'ToffeePenny'. The 'GreenTriangle' panel remains at the top, and the 'ToffeePenny' panel is below it, both with 'Add Image Samples:' sections and 'Webcam'/'Upload' buttons. The 'Add a class' button is still at the bottom. The 'Training' and 'Preview' panels on the right are identical to the previous screenshot, with the 'Train Model' button and the message about training requirements.

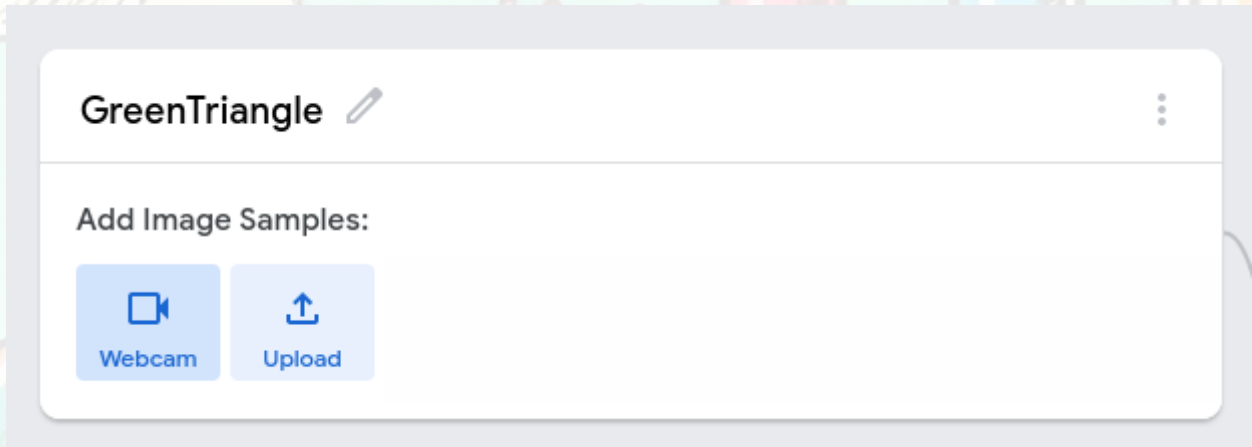
Build the model!



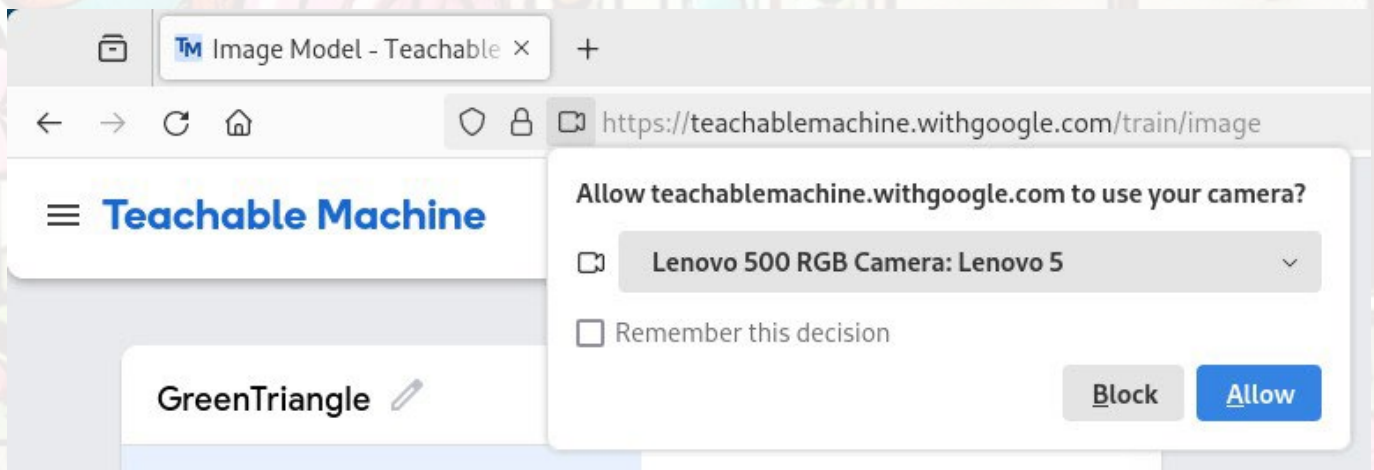
University of
Sunderland

Time to add the photos (webcam)

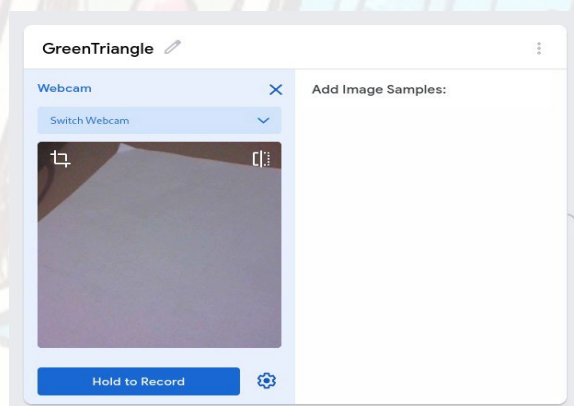
Click the Webcam for the first Class (GreenTriangle)



Your browser may ask for permission to use the Webcam. Check the **Remember this decision** checkbox and click **Allow**.



If all goes well, the webcam will start and you'll see a preview



Build the model!

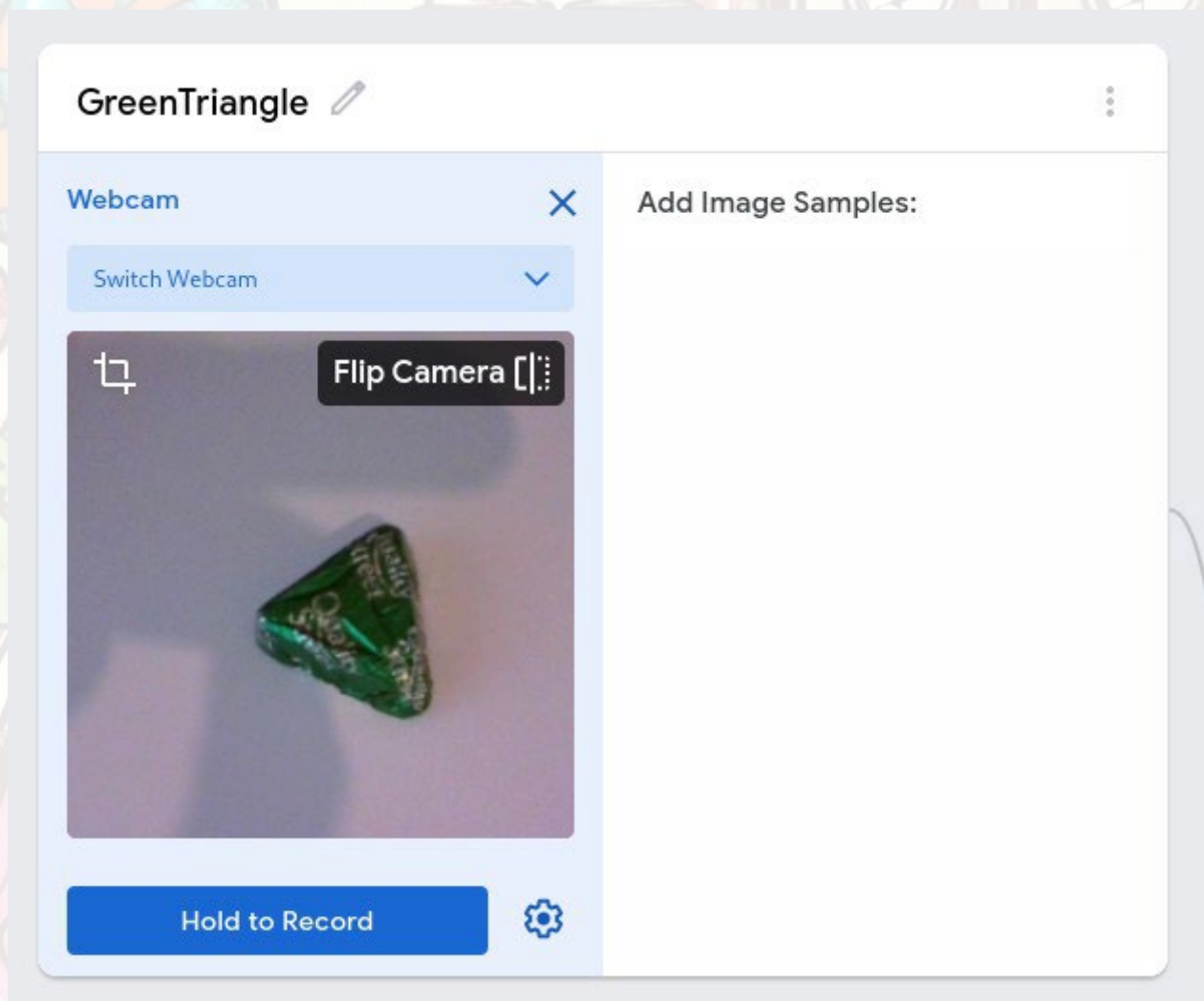


University of
Sunderland

Time to add the photos (webcam)

Before you begin the next part works best if you have a plain background. Also, you can hold the webcam to get a better photo of the sweets.

To record images, press the **Hold to Record** button



Build the model!

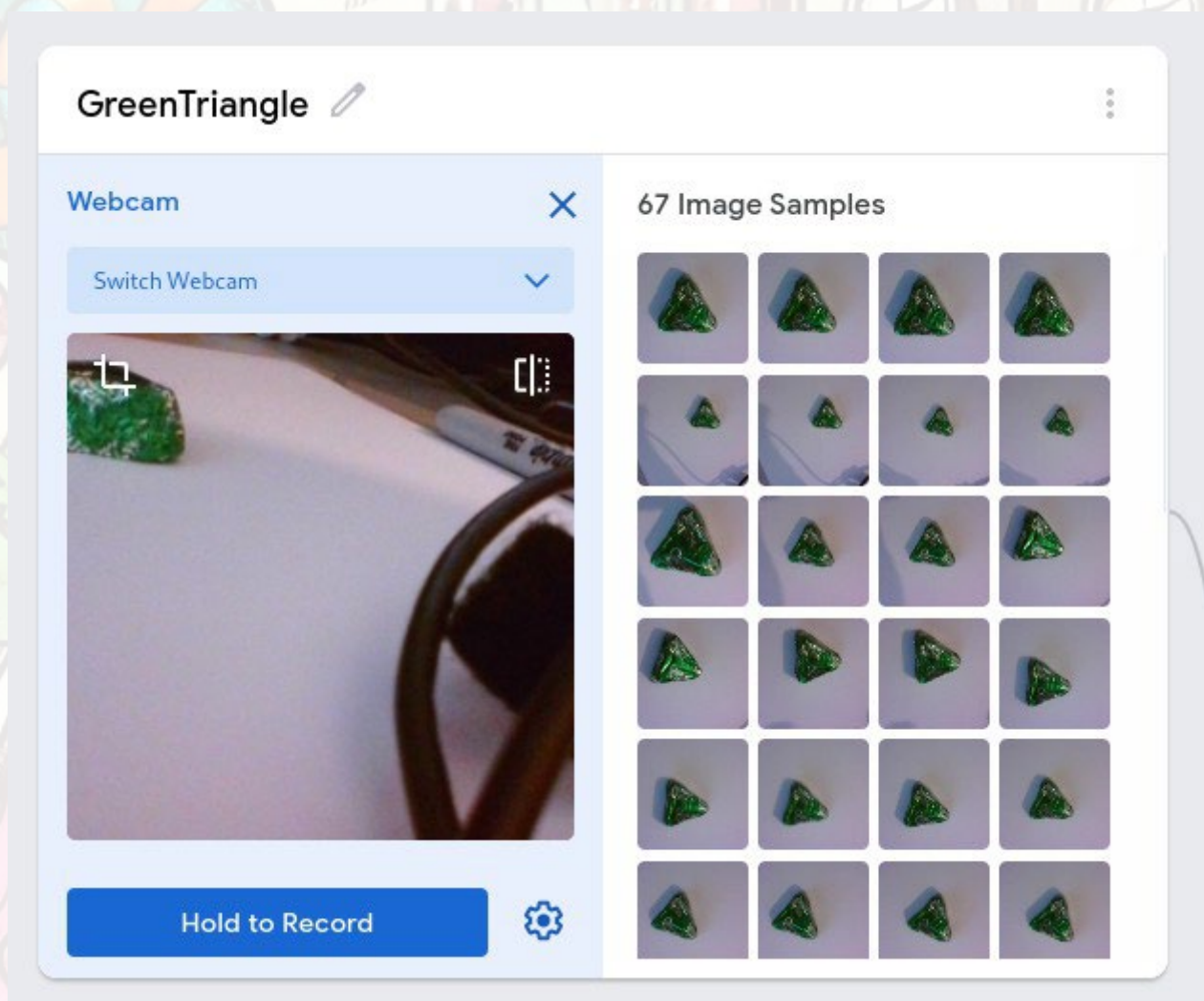


University of
Sunderland

Time to add the photos (webcam)

The webcam will start taking photos at regular intervals. Make sure you move the webcam around a little to get the sweet in different positions. This will improve the effectiveness of the model.

NOTE: If you make a mistake, you can always remove an image.



Build the model!



**University of
Sunderland**

Time to add another class

Once you have the images from the webcam for the sweet you can move on.

When you started the process teachable machine added two class, there's no point in just having one! So, add your next sweet.

GreenTriangle 



67 Image Samples



Webcam



Upload



ToffeePenny 



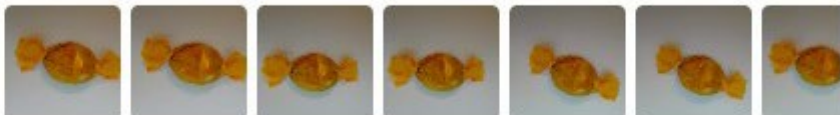
90 Image Samples




Webcam



Upload



 Add a class

Build the model!



University of
Sunderland

Add the rest of the classes

There are 11 sweet varieties in the Quality Street box. To add the rest, select the “Add a class” and repeat the process.

When you have finished there is one more class to add.

What about when there’s no sweet. Add an additional class called “**None**” and using the webcam make a class based on the “background” of where you took the sweet photos.

+ Add a class

Class 3 



Add Image Samples:



Webcam



Upload

Class 4 



Build the model!

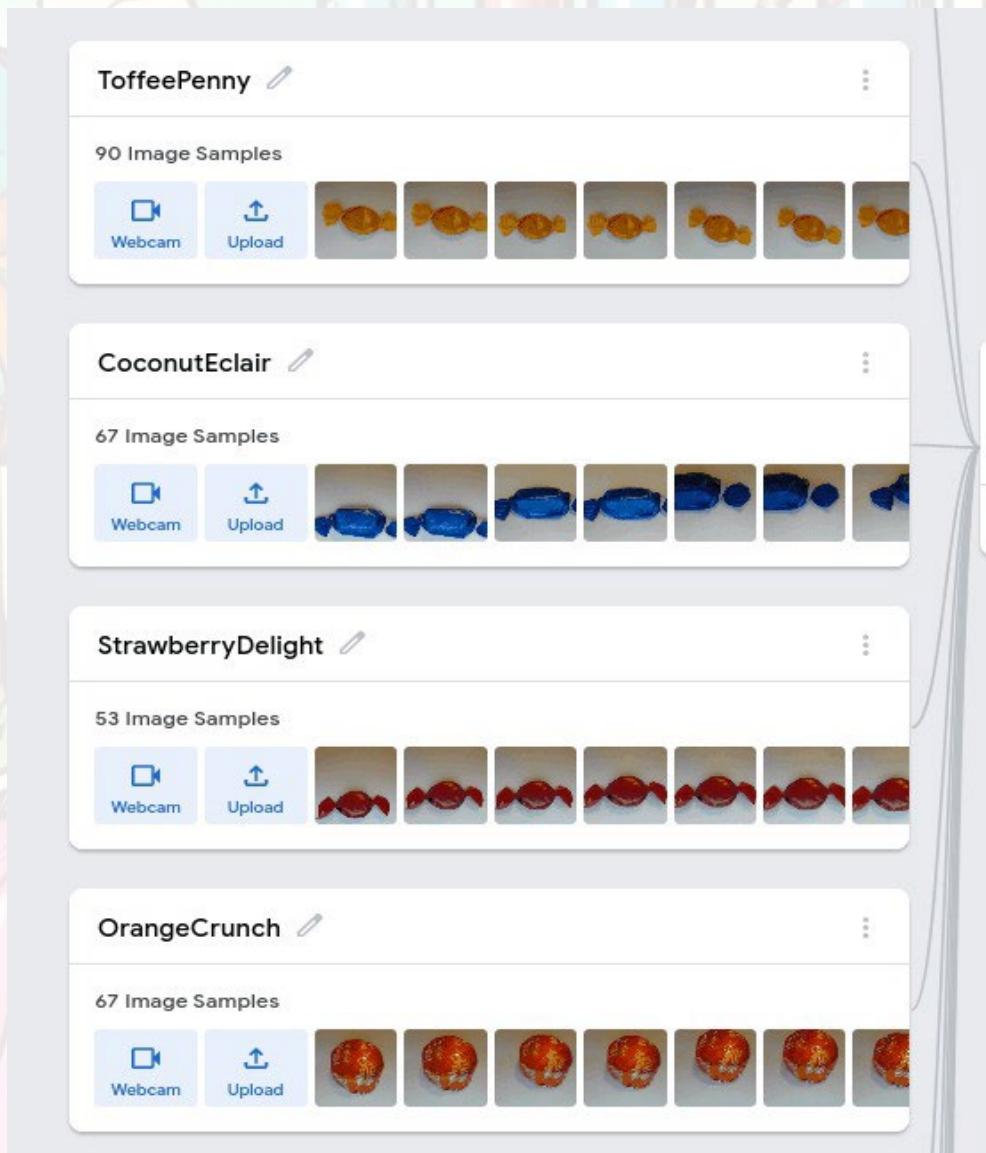


**University of
Sunderland**

All classes added

If all went well, you'll have something that looks something like the picture below.

That's your dataset created for the sweet sorter. Now we need to convert the dataset into an AI model.



Build the model!

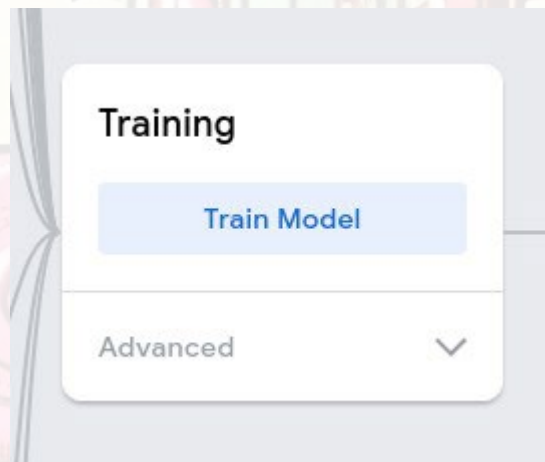


**University of
Sunderland**

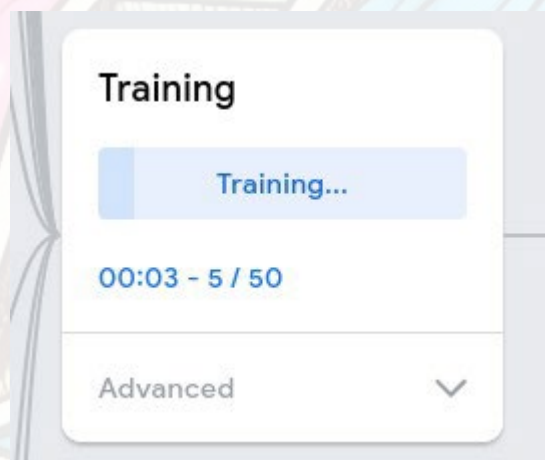
Let's build the model!

To train the model each photo from each class is converted into a set of numbers that are then passed through a neural network to produce a model.

To start the training process, select the “Train Model” option. Be patient it does take a while; there's a lot of photos!



You can see the progress of the model training. Once you select the “Train Model” button it becomes a progress bar.



Build the model!



University of
Sunderland

Check the model!

Once the training is complete “Teachable Machine” provides you with a very basic testing facility. You can point the webcam at the sweets, and it will try and identify them.

The bars show you how confident the model is when predicting the match.

The screenshot displays the Teachable Machine web interface. On the left, a list of training categories is shown, each with a 'Webcam' button, an 'Upload' button, and a row of image samples:

- OrangeCrunch**: 67 Image Samples
- MilkChocBlock**: 88 Image Samples
- OrangeCream**: 91 Image Samples
- ToffeeFinger**: 71 Image Samples
- PurpleOne**: 105 Image Samples
- Fudge**: (no sample count visible)

In the center, a 'Training' panel indicates 'Model Trained' and has an 'Advanced' dropdown menu.

On the right, the 'Preview' panel shows the model's output for a live webcam feed. The input is 'Webcam' and the 'Switch Webcam' button is visible. The output section displays confidence bars for various categories:

Category	Confidence
Green...	100%
Toffe...	
Coco...	
Straw...	
Oran...	
MilkC...	
Oran...	
Toffe...	
Purpl...	
Fudge	
Cara...	

Build the model!



**University of
Sunderland**

Before you move on

- Did it work as expected?
- What happens when you present a sweet at an angle, or only show part of the sweet to the webcam?
- What happens when you don't present a sweet to the webcam? Has your “None” class worked?
- What's happens if you present a different sweet that wasn't in your training set (put something in front of the webcam)?
- How do you think you could improve the training data?

**Have a chat with other students to see
what they think.**

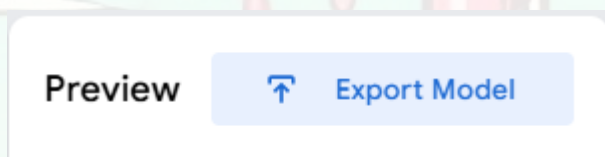
Use the model!



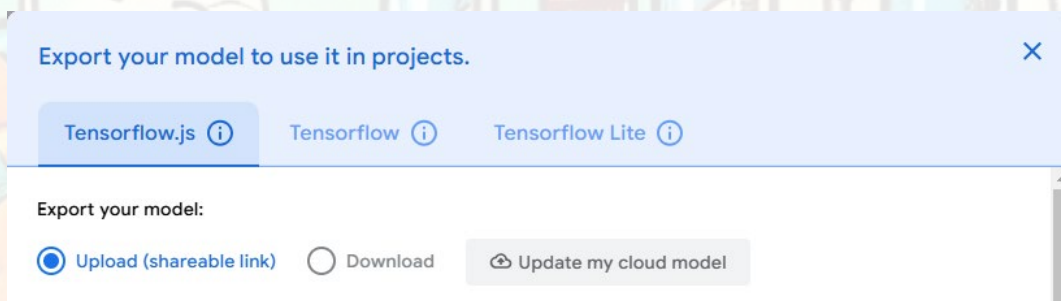
University of
Sunderland

Let's upload the project so we can use it.

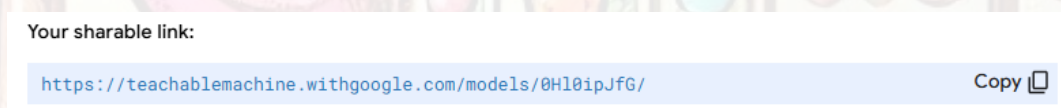
Click **Export Model**



Click **Upload (shareable link)** radio button



You'll also see two important things now. The Sharable link.



And the sample code which will already have the shared link in the code for you.

Code snippets to use your model:

Javascript

p5.js

Contribute on Github

Learn more about how to use the code snippet on [github](#).

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init">
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs/dist/teachablemachine-
image.min.js"></script>
```

Copy

Copy the code!

Use the model!



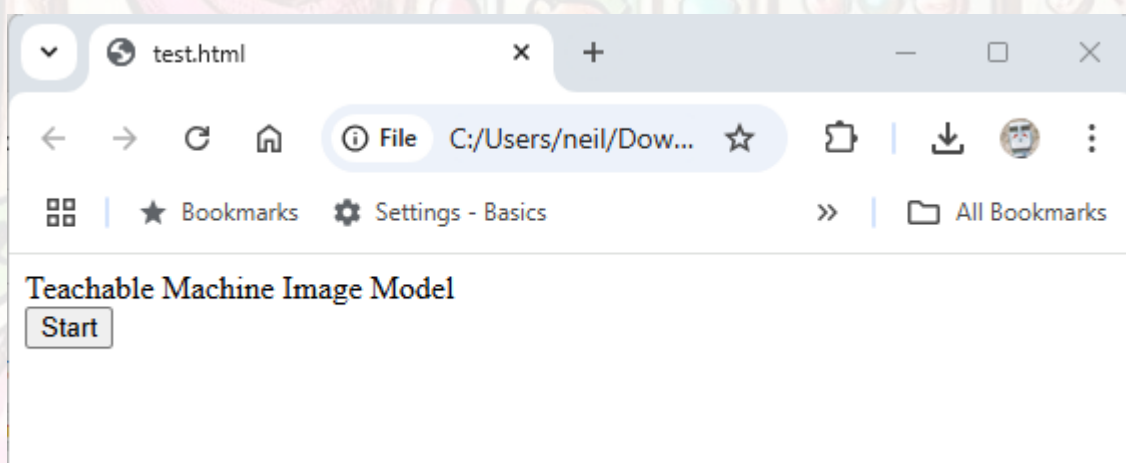
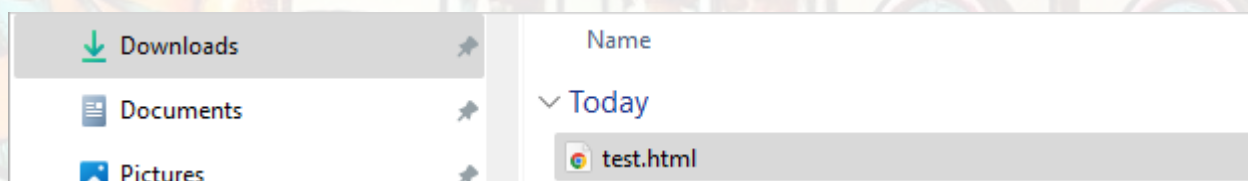
University of
Sunderland

Let's use the sample code.

Open a text editor (Notepad or VS Code) and paste the code into the file.

Save the code into a file as **test.html**. You can save it to the in your **Downloads** folder for ease.

Browse to the **Downloads** folder and double click the file.



Click the **Start** button and wave some sweets in front of the webcam!

Use the model!



**University of
Sunderland**

Let's use the sample code.

You are now running your own AI model in a browser!

Teachable Machine Image Model

Start



GreenTriangle: 0.00
ToffeePenny: 0.00
CoconutEclair: 0.00
StrawberryDelight: 0.00
OrangeCrunch: 0.00
MilkChocBlock: 0.00
OrangeCream: 0.00
ToffeeFinger: 0.00
PurpleOne: 1.00
Fudge: 0.00
CaramelSwirl: 0.00

It's all down to your imagination now. What could you use this technology for?

What about record faces with the class name as an ID to create a login facility? Is that better than a password?

Use the model!



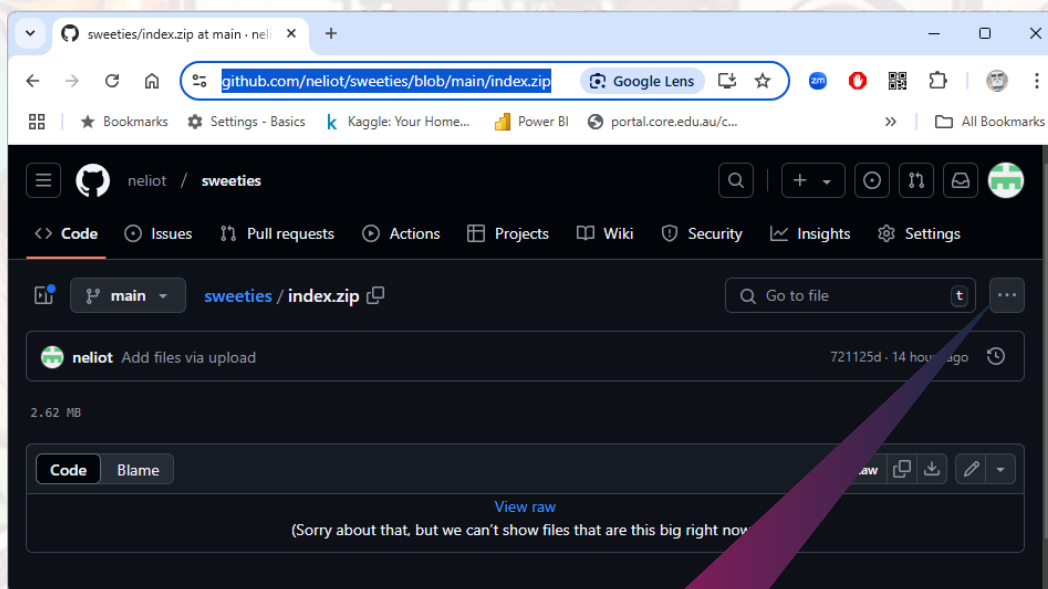
**University of
Sunderland**

What about building applications?

Now that you have the model in use using the sample code lets look at how a simple app can be built around the model.

Download the sample app (zip file) from the GitHub repository and unzip it into a folder.

<https://github.com/neliot/sweeties/blob/main/index.zip>



Select the
download
option from
here

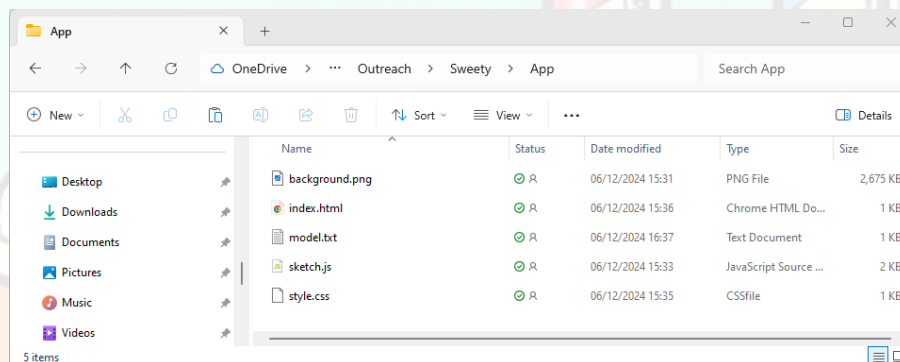
Use the model!



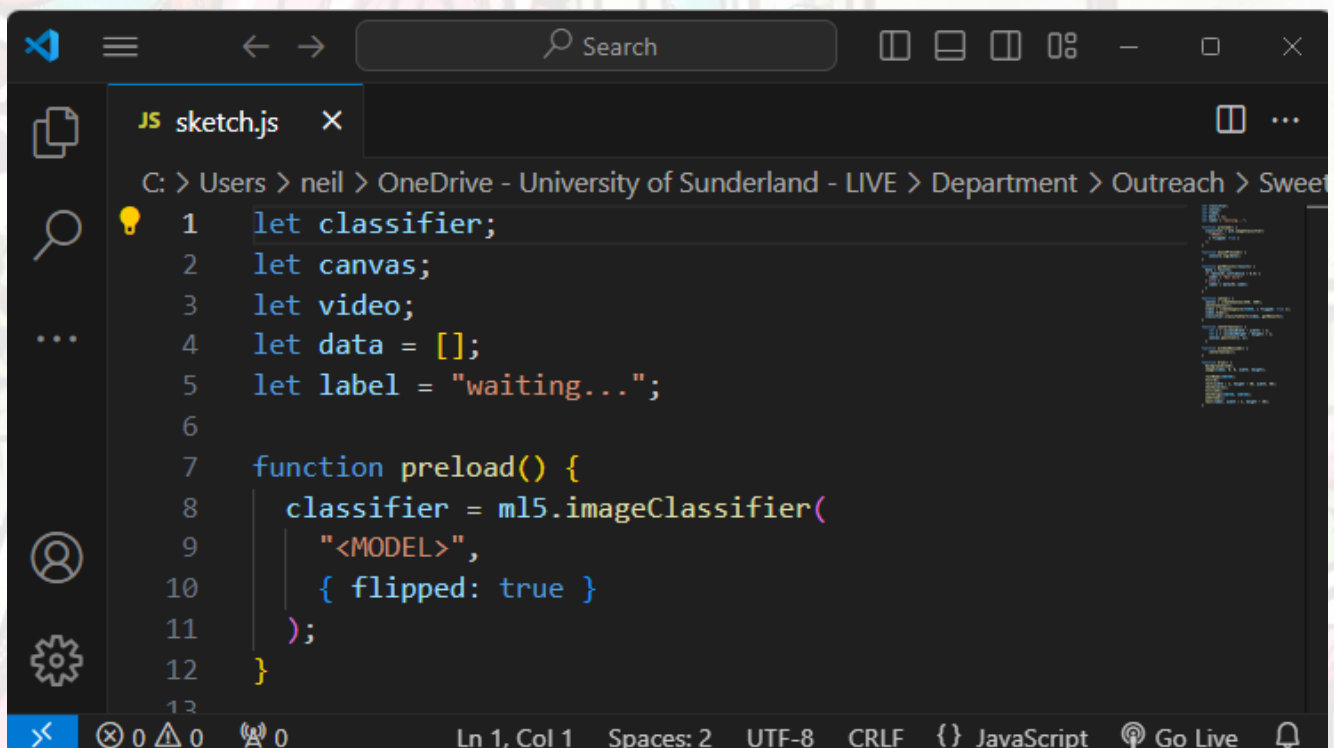
University of
Sunderland

What about building applications?

Open a file explorer and navigate to the folder where you expanded the zip file.



Right Click on the **sketch.js** file and use the **open with** option to open the file in an editor such as **Visual Studio Code**.



Replace the **<MODEL>** placeholder with the URL for the model you create in Teachable Machine.

Use the model!



University of
Sunderland

What about building applications?

If you didn't manage to create your own model for this. Go back to the folder and there is a text file called "model.txt". It contains a URL to a pre-trained sweetie model for you.

Each sweetie has been scanned on a white background for the model.

Now you can launch the application. **Right Click** the `index.html` file and use the **open with** option to open the file in a web browser such as **Firefox**.



Now you can launch the application. **Right Click** the `index.html` file and use the **open with** option to open the file in a web browser such as **Firefox**.

And Finally!



**University of
Sunderland**

On the **BSc Computer Science** Programme at the **University of Sunderland** you'll study AI and Machine Learning in more detail, which means you'll be able to create some amazing things and have a good grasp on this important emerging technology.



Fancy a Challenge?



University of
Sunderland

Build a new model of the faces of all the people in your “team” (and other attendees if they are willing) and using the sample application check that the system can identify each of you.

Don’t forget to update the model **URL** in the **sketch.js** file.

Don’t forget to add the “**None**” class. The model will always try and match to at least one **Class**, so you need a fall back!

Remember no spaces in the **Class** names!

