

DR. ALVIN'S PUBLICATIONS

PLAYING WITH THE TEACHABLE MACHINE BY GOOGLE

DR. ALVIN ANG



1 | PAGE

COPYRIGHTED BY DR ALVIN ANG
WWW.ALVINANG.SG

CONTENTS

I. Getting Started	3
II. Training the Output & Predicting.....	5
III. Further Options	6
A. Downloading the Model in .h5 Format.....	6
B. Advanced – Under the Hood.....	7
1. Accuracy Per Class.....	8
2. Confusion Matrix.....	9
3. Accuracy Per Epoch.....	10
4. Loss Per Epoch	11
C. Use Webcam and Add More Classes	12
About Dr. Alvin Ang	13

I. GETTING STARTED

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

Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

Marshmallow 100%

Not Marshmallow

Teachable Machine

New Project

Open an existing project from Drive.

Open an existing project from a file.

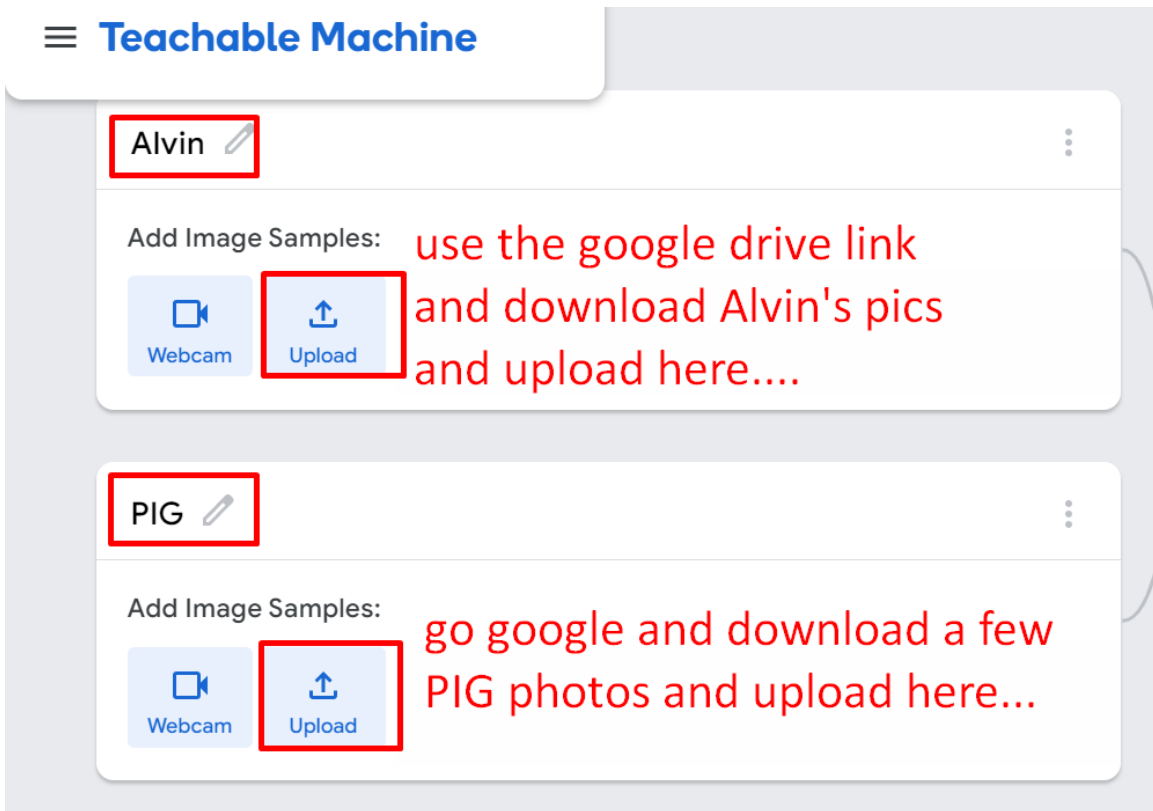
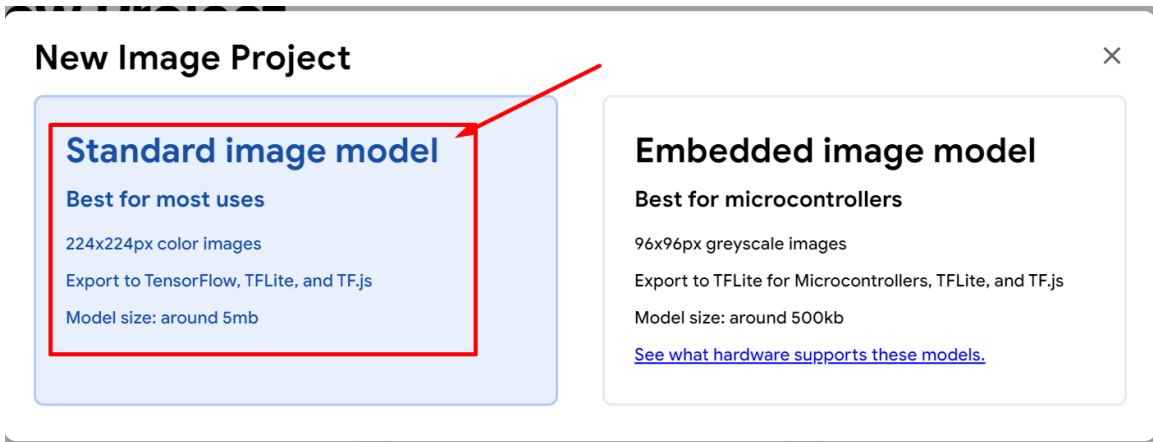
Image Project
Teach based on images, from files or your webcam.

Audio Project
Teach based on one-second-long sounds, from files or your microphone.

Pose Project
Teach based on images, from files or your webcam.

English

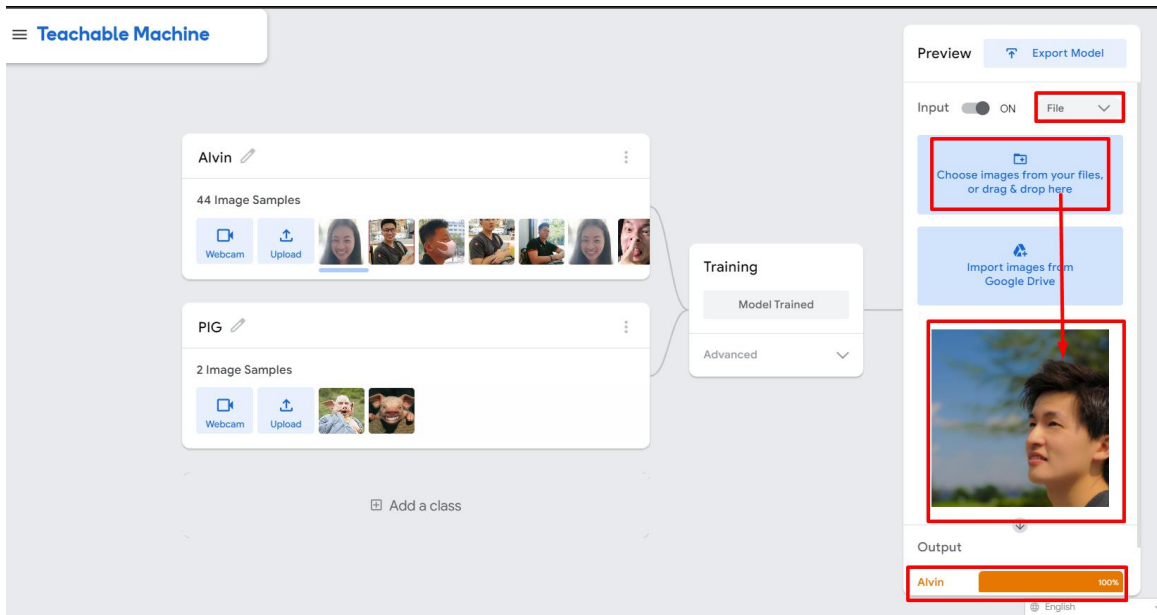
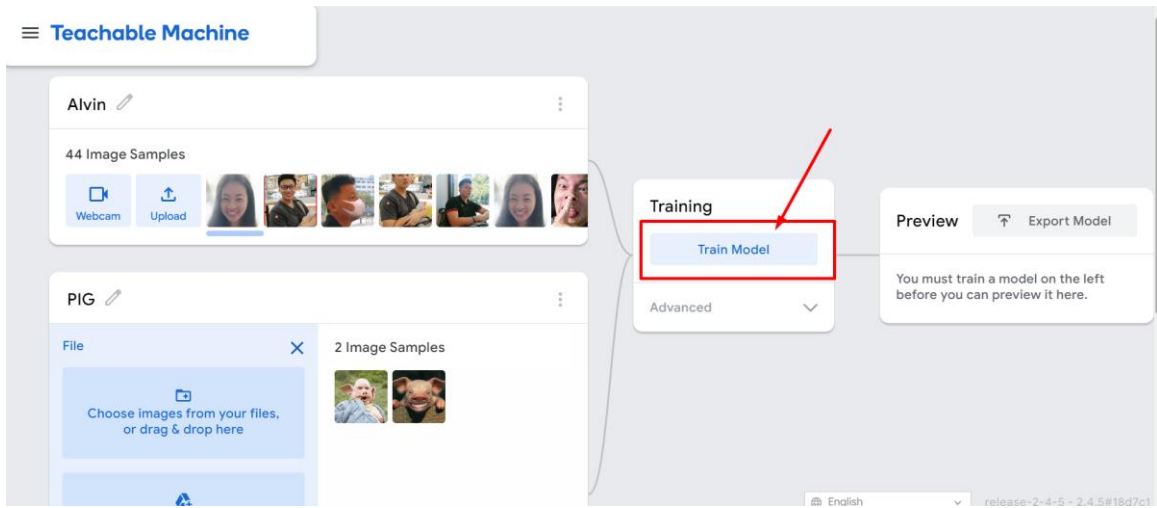
release-2-4-5 - 2.4.5



My photos are here....

https://drive.google.com/drive/folders/11BXyN2pn_qZYt1WIqcV8gQLL9Q4y5Dvc?usp=sharing

II. TRAINING THE OUTPUT & PREDICTING



- Instead of uploading photos, you may also use the webcam function to test the output...

III. FURTHER OPTIONS

A. DOWNLOADING THE MODEL IN .H5 FORMAT

Export your model to use it in projects.

Tensorflow.js ⓘ **Tensorflow ⓘ** Tensorflow Lite ⓘ

Model conversion type:
 Keras Savedmodel [Download my model](#)

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:
Keras [Contribute on Github](#)

```
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np

# Load the model
model = load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
# Replace this with the path to your image
image = Image.open('<IMAGE_PATH>')
#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
```

[Copy](#)

Export Model

you can download the model as .h5 format

you can copy the code and run this model back in Google Colab (to save time on training)

B. ADVANCED – UNDER THE HOOD

The screenshot displays a machine learning interface with several key components:


- Training Panel:** Shows a 'Model Trained' button and a list of training parameters: Epochs (50), Batch Size (16), and Learning Rate (0.001). The 'Advanced' section is expanded, and the 'Under the hood' section is visible at the bottom.
- Preview Panel:** Contains a 'Choose image or drag' button, an 'Import from Google' button, and a photo of a man with glasses.
- Vocab:** A dropdown menu showing the current vocabulary.
- Accuracy per class:** A table showing accuracy and sample counts for each class.
- Confusion Matrix:** A heatmap showing the relationship between actual and predicted classes.
- Accuracy per epoch:** A line graph showing training and testing accuracy over time.

CLASS	ACCURACY	# SAMPLES
Alvin	1.00	7
PIG	0.00	1

Class	Alvin	PIG
Alvin	7	0
PIG	1	0

Legend for Accuracy per epoch: acc (blue line), tes (orange line).

1. ACCURACY PER CLASS

Accuracy per class 

Accuracy per class is calculated using the test samples. Check out the vocab section to learn more about test samples.

Accuracy per class



CLASS	ACCURACY	# SAMPLES
Alvin	1.00	7
PIG	0.00	1

2. CONFUSION MATRIX

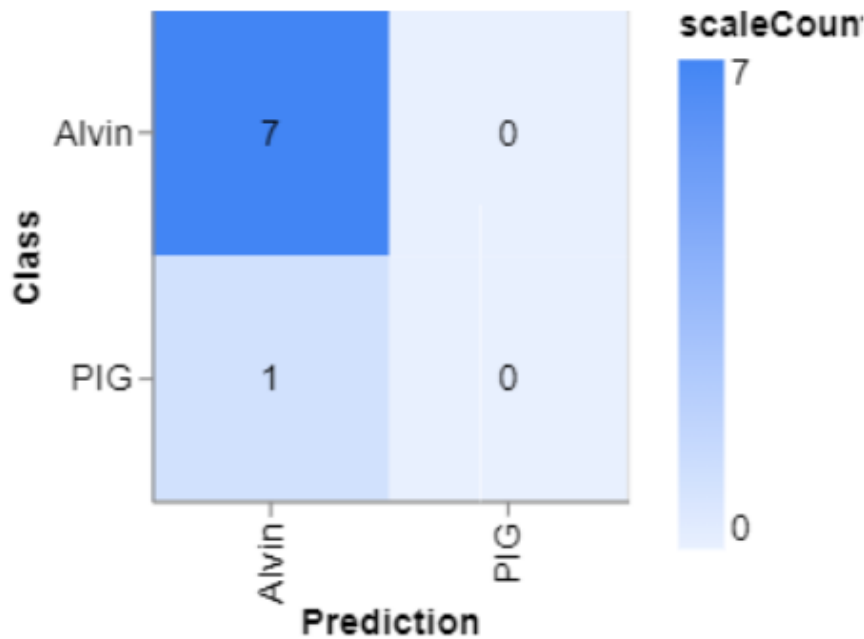
Confusion Matrix

A confusion matrix summarizes how accurate your model's predictions are. You can use this matrix to figure out which classes the model gets confused about.

The y axis (Class) represents the class of your samples. The x axis (Prediction) represents the class that the model, after learning,

guesses those samples belong to. So, if a sample's Class is "Muffin" but its Prediction is "Cupcake", that means that after learning from your data, the model misclassified that Muffin sample as a Cupcake. This usually means that those two classes share characteristics that the model picks up on, and that particular "Muffin" sample was more similar to the "Cupcake" samples.

Confusion Matrix



Out of total of 8 samples, 1 was predicted wrongly.

1 sample was a REAL PIG but Predicted as ALVIN!!!

(Alvin is a pig??? HAHAHA)

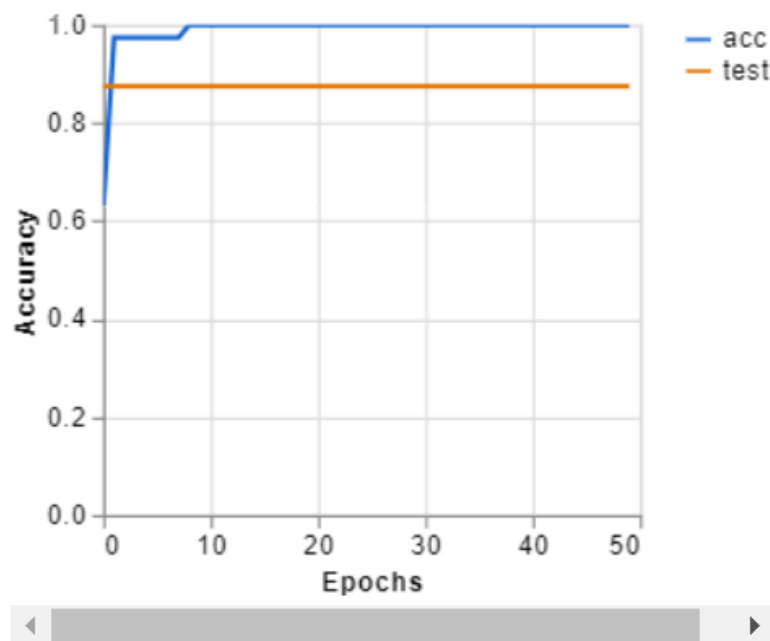
3. ACCURACY PER EPOCH

Accuracy ?

Accuracy is the percentage of classifications that a model gets right during training. If your model classifies 70 samples right out of 100, the accuracy is $70 / 100 = 0.7$.

If the model's prediction is perfect, the accuracy is one; otherwise, the accuracy is lower than one.

Accuracy per epoch



4. LOSS PER EPOCH

Loss

Loss is a measure for evaluating how well a model has learned to predict the right classifications for a given set of samples. If the model's predictions are perfect, the loss is zero; otherwise, the loss is greater than zero.

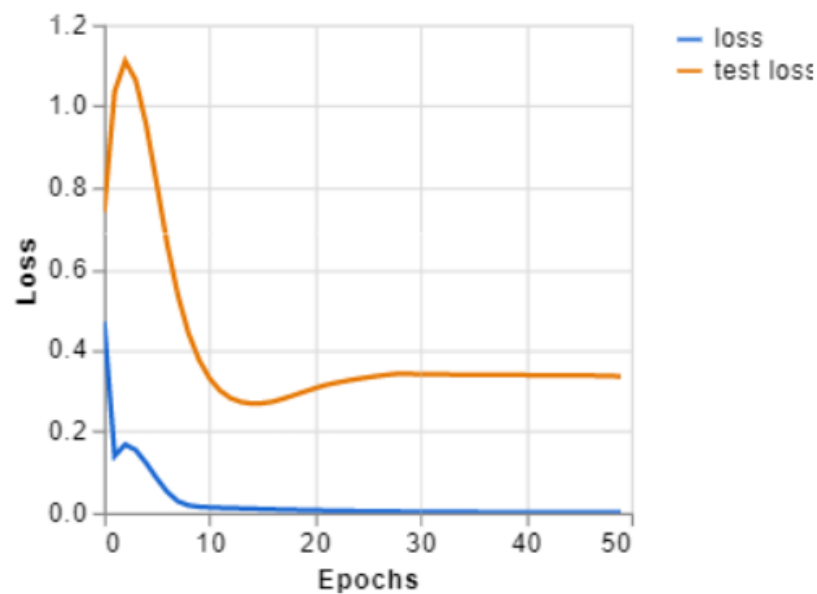
loss is greater than zero.
epoch

To get an intuitive sense of what this measures, imagine you have two models: A and B. Model A predicts the right classification for a sample but is only 60% confident of that prediction. Model B also predicts the right classification for the same sample but is 90% confident of that prediction. Both models have the same accuracy, but model B has a lower loss value.

loss
test loss

40 50
Epochs

Loss per epoch



C. USE WEBCAM AND ADD MORE CLASSES

The screenshot displays the Teachable Machine web interface. At the top left, there is a 'Teachable Machine' header with a menu icon. The main area is divided into two columns. The left column contains two class cards, 'Class 1' and 'Class 2', each with an 'Add Image Samples' section. In both cards, the 'Webcam' button is highlighted with a red box. Below these cards is an 'Add a class' button, also highlighted with a red box. The right column contains a 'Training' panel with a 'Train Model' button and a 'Preview' panel with an 'Export Model' button. Red text annotations are present: 'you may also use the webcam...' points to the 'Webcam' buttons in the class cards, and 'you may also 'Add More Classes!...' points to the 'Add a class' button. A message in the 'Preview' panel states: 'You must train a model on the left before you can preview it here.'

ABOUT DR. ALVIN ANG



Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.