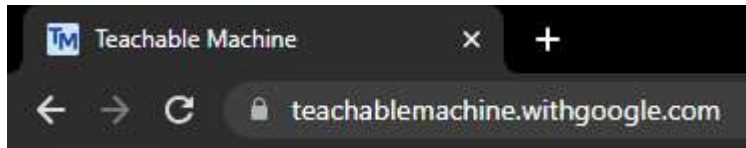
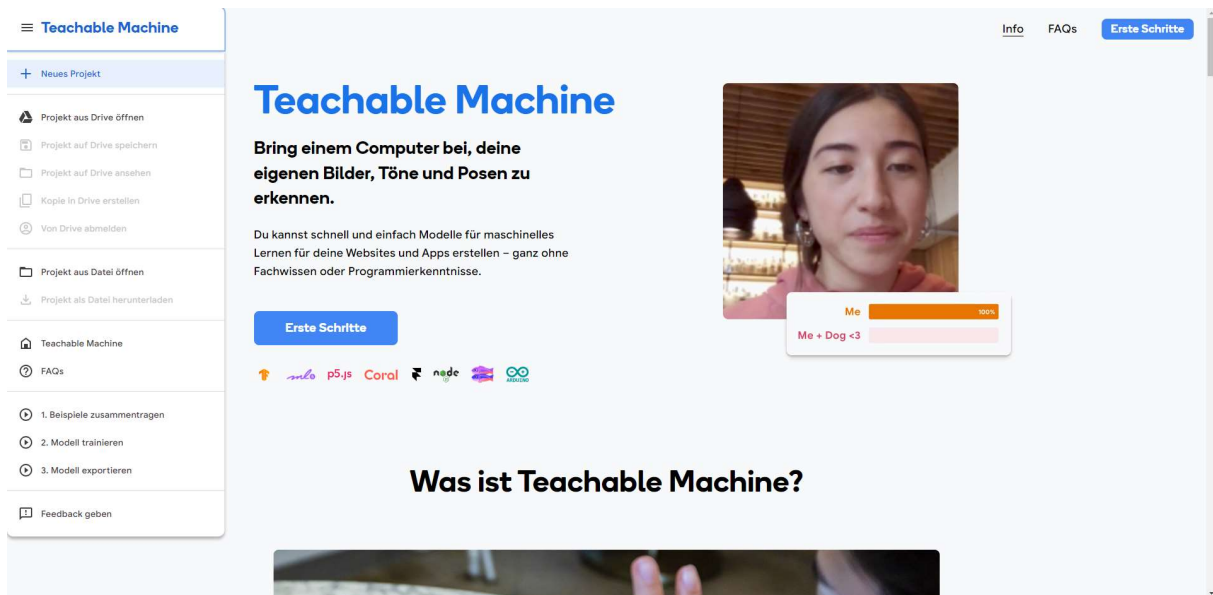


# Google Teachable Machine Protokoll

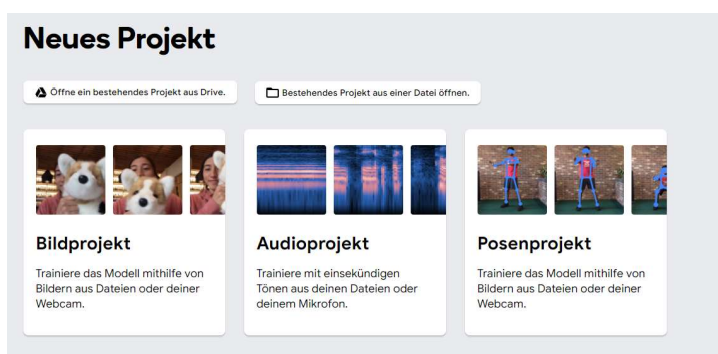


(<https://teachablemachine.withgoogle.com>)

Besuchen Sie folgende Website, um mit dem Training der Maschine zu beginnen



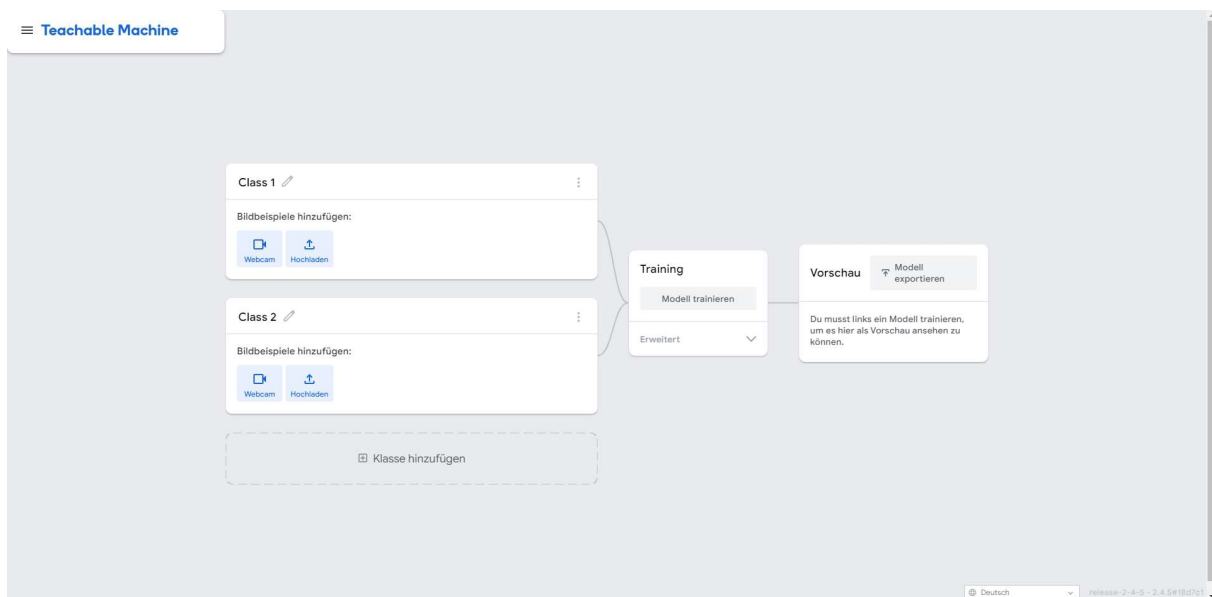
Klicken Sie auf Neues Projekt erstellen bzw. Projekt aus Drive/Datei öffnen, falls Sie ein bestehendes Projekt öffnen möchten.



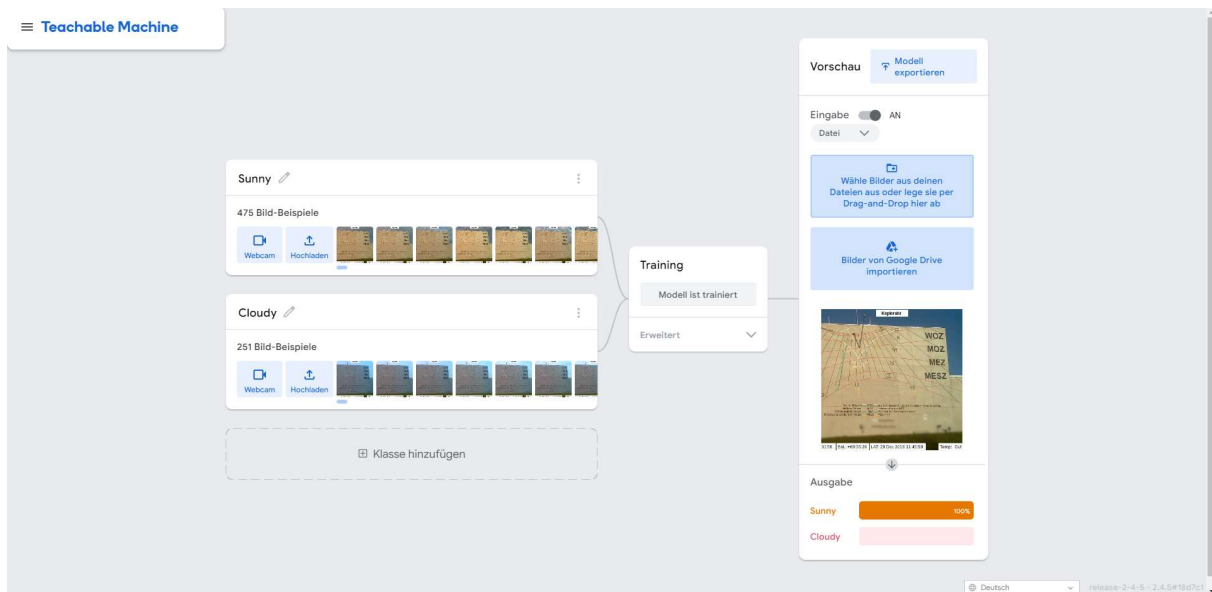
Wählen Sie nun zwischen Bild-, Audio- oder Posen-Erkennung aus.



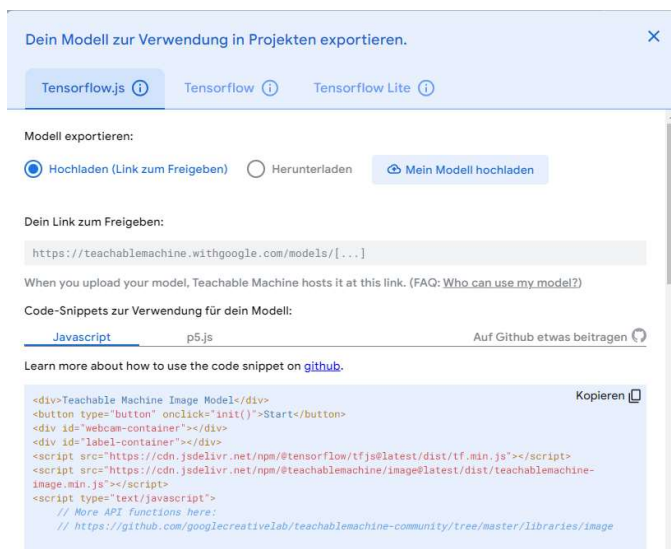
Da in unserem Fall größere und farbige Bilder aufgenommen werden, habe ich ein Modell mit Standardbildern verwendet.



Im Training-Fenster angekommen, kann man Klassen erzeugen, diese benennen, gespeicherte Bilder hochladen oder mit der Webcam aufnehmen, das Modell trainieren und in der Vorschau mit Bildern oder Webcam-Aufnahmen testen bevor man es exportiert.



So würde ein fertiges Projekt aussehen. Nun kann wie oben erwähnt mit gespeicherten Bildern oder Webcam-Aufnahmen getestet werden, wobei unten ausgegeben wird mit welcher Wahrscheinlichkeit das Bild zu welcher Klasse gehört. Selbstverständlich können noch weitere Beispielbilder zu den Klassen hinzugefügt werden, um die Genauigkeit zu erhöhen oder gänzlich neue Klassen erstellt werden.



Nun bleibt nur noch das Modell zu exportieren. Dies ist in verschiedenen Sprachen und Formaten möglich.

Dein Modell zur Verwendung in Projekten exportieren.

Tensorflow.js

Tensorflow

Tensorflow Lite

Konvertierungs-Typ des Modells:

☒ Keras

☐ Savedmodel

↓ Mein Modell herunterladen

Macht aus deinem Modell ein keras .h5-Modell. Hinweis: Die Konvertierung geschieht in der Cloud. Deine Trainingsdaten werden jedoch nicht hochgeladen, sondern nur das trainierte Modell.

Code-Snippets zur Verwendung für dein Modell:

Keras

OpenCV Keras

Auf Github etwas beitragen

```
from keras.models import load_model # TensorFlow is required for Keras to work
import cv2 # Install opencv-python
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# CAMERA can be 0 or 1 based on default camera of your computer
camera = cv2.VideoCapture(0)
```

Kopieren

In unserem Fall würde sich OpenCV am besten eignen.