

image detection through teachable machine

Submitted by UMEHABIBA



May 21, 2020

subject data science

ROLL-NUM BSEM-S17-016

**Acknowledgment:**

It is extraordinary joy for me to attempt this undertaking. I feel exceptionally pleased doing the venture. I am appreciative to my guide Dr. Arffan Jaffar. This task would not have finished without his gigantic assistance and commendable experience. At whatever point I was out of luck; he was there behind me.

**Dedication:**

Firstly, I want to say thanks to our parents for giving support to me. Secondly, it dedicated to my teacher from nursery to primary level ma’am Shahzadi who taught me how I can write and how I can speak. Also thanks to my friends Rabia and Kainat who always support me and never leave me in any condition either happy or sad. I want to thank you to my Professor M.Imran and Professor Majid Rafique who taught me how we can handle the situation of troubles. At the end, we dedicated all of this to Dr. Arffan Jaffar who teaching us.

# 

# **INTRODUCTION**

An image is a picture that has been created and store in electronic form. It can be described in the terms of vector graphics and raster graphics. We can detect image and store them in the various manners there are many different ways to detect the images and storing them. In this project I will use the teachable machine to detect the image and export that to tensorflow.js. The image will be classified in different classes then the models formed are then trained and, in the end, code is formed.

## **Teachable** **Machine**

“Teachable machine is a browser application that you can train with your webcam to recognize objects or expressions”

It is an effort by Google to make machine learning and AI accessible to the wider public without requiring any specialized training, knowledge in computer science or coding. It is basically a web base tool.

## **Working of Teachable Machine**

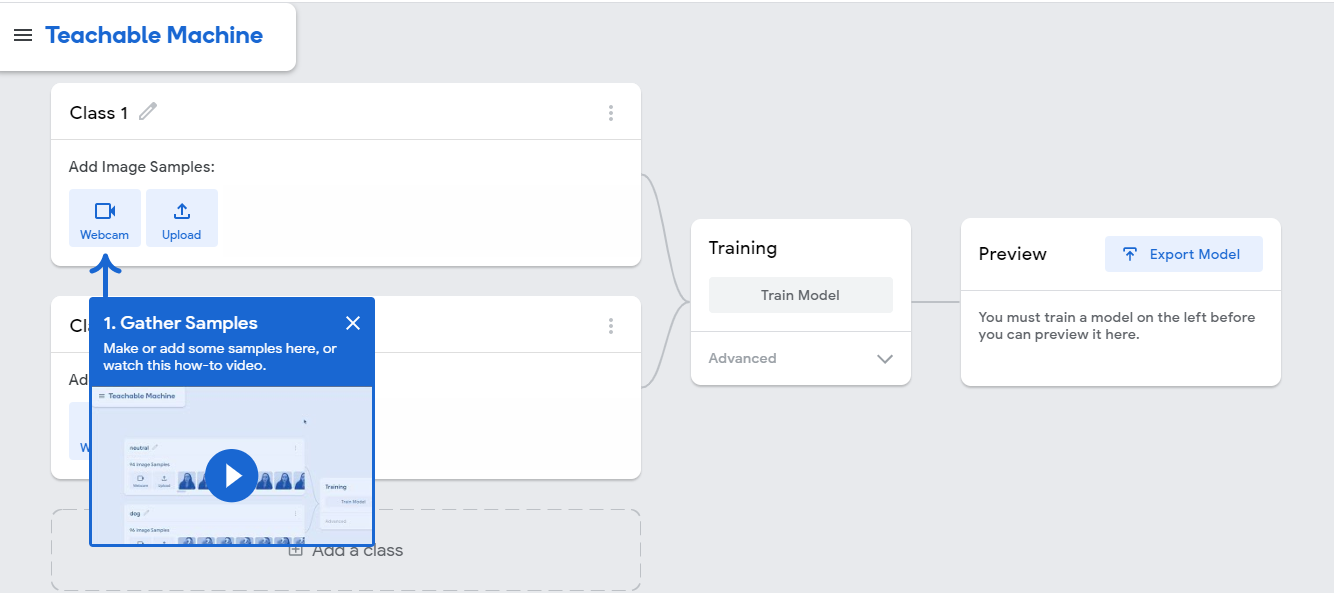
Teachable Machine is a web tool that makes it fast and easy to create machine learning models for your projects, no coding required. Train a computer to recognize your images, sounds & poses then export your model for your sites, apps, and more

# IMPLEMENTATION

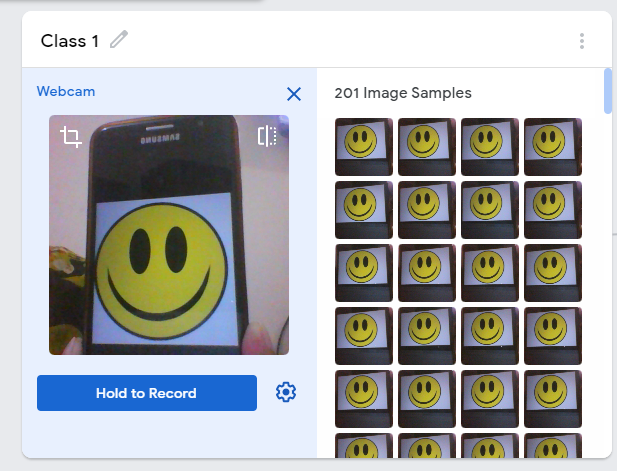
We choose to teachable method to identify and train the data gathered by the handmade sketches options we choose.

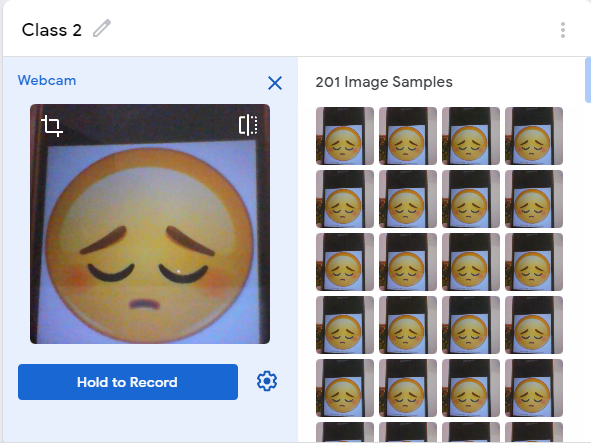
## Image detection through Teachable Machine

There are three steps through which we can process data

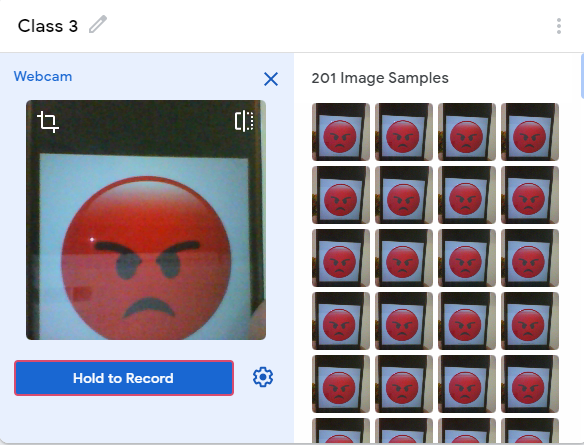
* Gather
* Train
* Export

1. **Add Samples:**

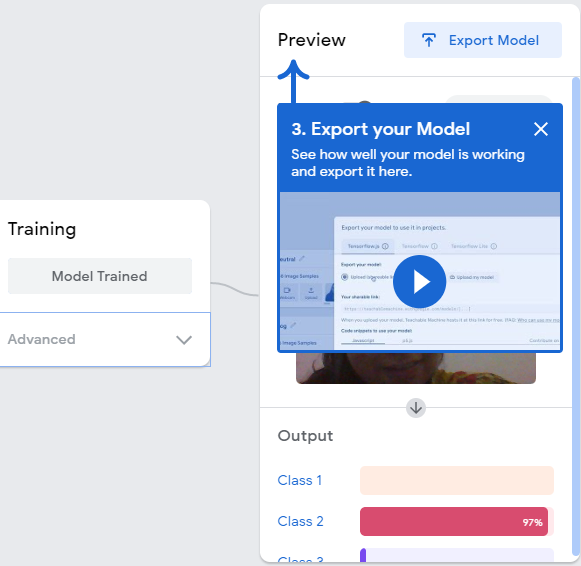
* Firstly, I selected my topic to implement on teachable machine. It is image detection and then I got this screen.
* Add class 1 through webcam. Hold a picture in front of webcam and record it 200 times. Then we get this.
* Add class 2 through different picture hold on front of webcam.

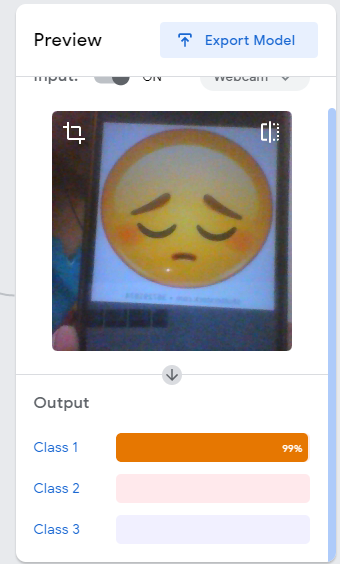


* Add class 3 and so on it is up to you how much you want to add classes.



1. **TRAIN:**





1. **EXPORT:**

<div>Teachable Machine Image Model</div>

<button type="button" onclick="init()">Start</button>

<div id="webcam-container"></div>

<div id="label-container"></div>

<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-image.min.js"></script>

<script type="text/javascript">

// More API functions here:

// https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

// the link to your model provided by Teachable Machine export panel

const URL = "./my\_model/";

let model, webcam, labelContainer, maxPredictions;

// Load the image model and setup the webcam

async function init() {

const modelURL = URL + "model.json";

const metadataURL = URL + "metadata.json";

// load the model and metadata

// Refer to tmImage.loadFromFiles() in the API to support files from a file picker

// or files from your local hard drive

// Note: the pose library adds "tmImage" object to your window (window.tmImage)

model = await tmImage.load(modelURL, metadataURL);

maxPredictions = model.getTotalClasses();

// Convenience function to setup a webcam

const flip = true; // whether to flip the webcam

webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip

await webcam.setup(); // request access to the webcam

await webcam.play();

window.requestAnimationFrame(loop);

// append elements to the DOM

document.getElementById("webcam-container").appendChild(webcam.canvas);

labelContainer = document.getElementById("label-container");

for (let i = 0; i < maxPredictions; i++) { // and class labels

labelContainer.appendChild(document.createElement("div"));

}

}

async function loop() {

webcam.update(); // update the webcam frame

await predict();

window.requestAnimationFrame(loop);

}

// run the webcam image through the image model

async function predict() {

// predict can take in an image, video or canvas html element

const prediction = await model.predict(webcam.canvas);

for (let i = 0; i < maxPredictions; i++) {

const classPrediction =

prediction[i].className + ": " + prediction[i].probability.toFixed(2);

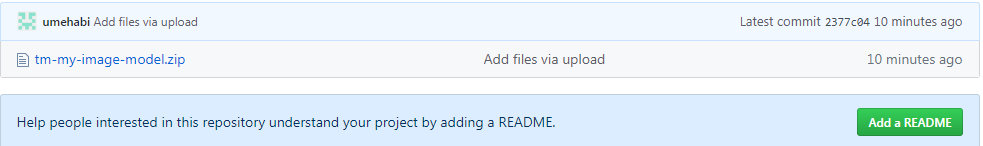
labelContainer.childNodes[i].innerHTML = classPrediction;

}

}

</script>

## Download file then upload on github:



# Conclusion:

The Teachable Machine is an amazing tool for machine learning. With it we can educate the device via specific form of fashions like Image Model, Audio Model and Pose Project we ought to train the device to vary between the numerous classes and at the quit constitute them in an awesome visible manner.

# References:

* <https://teachablemachine.withgoogle.com/train>
* <https://github.com/umehabi/MODEL>