

c) Web app and image training on device: TensorFlow.js Transfer Learning Image Classifier

The screenshot shows a web browser window for a TensorFlow.js transfer learning codelab. On the left is a vertical sidebar with numbered steps from 1 to 12. Step 1 is titled "Before you begin". The main content area has a header "transfer learning with TensorFlow.js" and a sub-section "About this codelab" with a last update date of Mar 31, 2022, and a note that it was written by a Googler. Below this is a detailed description of transfer learning advantages and a summary of the codelab's purpose. A "Prerequisites" section is at the bottom of the main content area, and a "Next" button is visible on the right.

The screenshot shows a code editor interface for a "Tensorflow.js Boilerplate" project. It features three tabs: HTML, CSS, and JS. The HTML tab contains a basic HTML structure with a title, meta tags, and a script tag for "TensorFlow.js Hello World". The CSS tab contains a simple CSS file with a single rule for the body element. The JS tab contains the Apache License 2.0 text, which is also present in the CSS tab. The code editor includes a toolbar with various icons and a status bar at the bottom.

TensorFlow.js Hello World

Loaded TensorFlow.js - version: 3.11.0

← → C 🔍 codepen.io/jasonmayes/pen/BaNjLyo#anon-signup

Glitch

style.css PRETTIER

citrine-unique-tote

Settings Assets Files LICENSE.md README.md index.html script.js style.css

style.css

```

5  * you may not use this file except in compliance with
6  * the License. You may obtain a copy of the License at
7  * http://www.apache.org/licenses/LICENSE-2.0
8  *
9  * Unless required by applicable law or agreed to in
10 * writing, software distributed under the License is distributed on an
11 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
12 * express or implied. See the License for the specific language governing
13 * limitations under the License.
14 */
15 // CSS files add styling rules to your content
16
17 body {
18   font-family: helvetica, arial, sans-serif;
19   margin: 2em;
20 }
21
22 h1 {
23   font-style: italic;
24   color: #FF6F00;
25 }
26
27 video {
28   clear: both;
29   display: block;
30   margin: 10px;
31   background-color: #000000;
32   width: 640px;
33   height: 480px;
34 }
35
36 button {
37   padding: 10px;
38   float: left;
39   margin: 5px 3px 5px 10px;
40 }
41
42 .removed {
43   display: none;
44 }
45
46 #status {
47   font-size: 150%;
48 }
49
50
51
52
53
54
55
56
57
58
59
5

```

citrine-unique-tote.glitch.me/

Remix

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

Loaded TensorFlow.js - version: 3.11.0

Enable Webcam Gather Class 1 Data Gather Class 2 Data Train & Predict! Reset

codepen.io/jasonmayes/pen/BaNjLyo

Tensorflow.js Boilerplate

Jason Mayes [PRO](#) [Follow](#)

HTML

```
<button id="reset">Reset</button>
<!-- Import TensorFlow.js library -->
<script>
```

CSS

JS

```
once.
tf.tidy(function () {
  let answer = mobilenet.predict(tf.zeros([1,
    MOBILE_NET_INPUT_HEIGHT, MOBILE_NET_INPUT_WIDTH, 3]));
});
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

MobileNet v3 loaded successfully!

Enable Webcam Gather Class 1 Data Gather Class 2 Data Train & Predict! Reset

Add to Collection Fork Embed Export Share

codepen.io/jasonmayes/pen/BaNjLyo?editors=1111

Tensorflow.js Boilerplate

Jason Mayes [PRO](#) [Follow](#)

HTML

```
<button id="enableCam">Enable Webcam</button>
<button class="dataCollector" data-ihot="0" data-
```

CSS

```
margin: 10px;
background: #000000;
width: 640px;
```

JS

```
// Compile the model with the defined optimizer and
// specify a loss function to use.
model.compile({
  // Adam changes the learning rate over time which is
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

MobileNet v3 loaded successfully!

Console

```
"=====
dense_Dense1 (Dense)      [null,128]          131200
"
"=====
dense_Dense2 (Dense)      [null,2]            258
"
"=====
Total params: 131458
Trainable params: 131458
Non-trainable params: 0
"
>
```

Add to Collection Fork Embed Export Share

codepen.io/jasonmayes/pen/BaNjLyo?editors=1111

Tensor Jason Mayes

codepen.io wants to
Use your camera

Block Allow

HTML

```
<video id="webcam" autoplay muted></video>
```

CSS

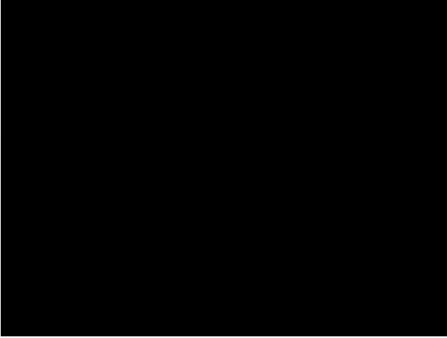
```
video {  
  clear: both;  
  display: block;  
  margin: 10px;
```

JS

```
if (navigator.mediaDevices.getUserMedia) {  
  // getUserMedia parameters.  
  const constraints = {  
    video: true,  
    ...other constraints...  
  };
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

MobileNet v3 loaded successfully!



Enable Webcam Gather Class 1 Data Gather Class 2 Data Train & Predict! Reset

Console

Add to Collection Fork Embed Export Share

Assignment 5: Advanced Keras TensorFlow.js: Make your own TensorFlow.js Boilerplate

codepen.io/jasonmayes/pen/BaNjLyo?editors=1111

Tensorflow.js Boilerplate Jason Mayes PRO + Follow

Awaiting TensorFlow.js load

HTML

```
<p id="status">Awaiting TensorFlow.js load</p>  
<video id="webcam" autoplay muted></video>
```

CSS

```
video {  
  clear: both;  
  display: block;  
  margin: 10px;
```

JS

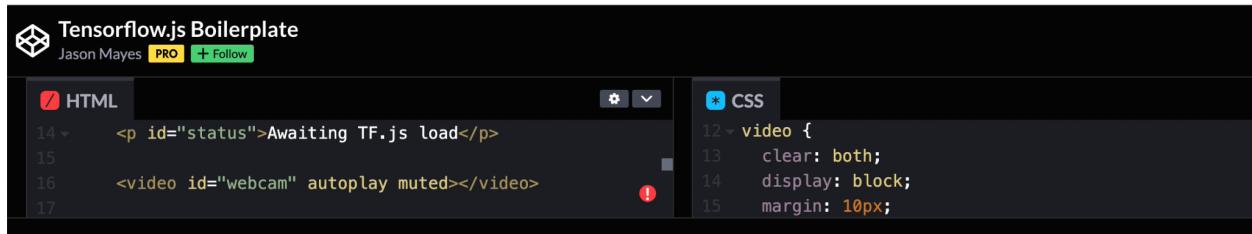
```
if (navigator.mediaDevices.getUserMedia) {  
  // getUserMedia parameters.  
  const constraints = {  
    video: true,  
    ...other constraints...  
  };
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

MobileNet v3 loaded successfully!



Gather Class 1 Data Gather Class 2 Data Train & Predict! Reset

Tensorflow.js Boilerplate
Jason Mayes PRO + Follow

HTML

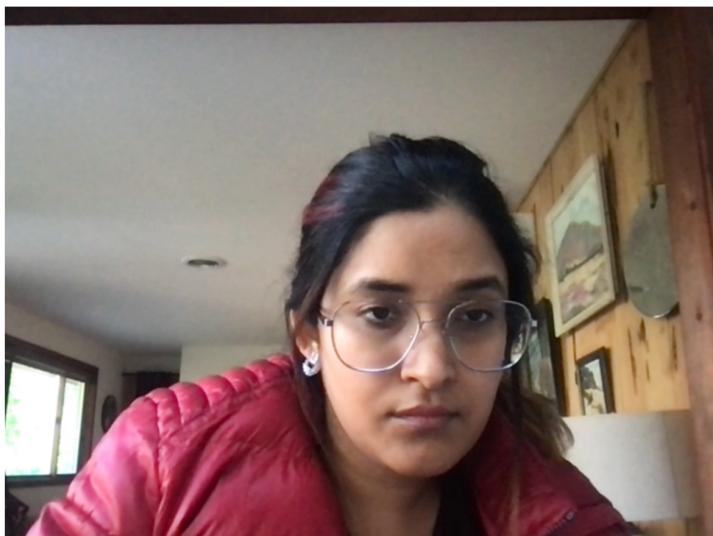
```
14 <p id="status">Awaiting TF.js load</p>
15
16 <video id="webcam" autoplay muted></video>
```

CSS

```
12 * video {
13   clear: both;
14   display: block;
15   margin: 10px;
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

Class 1 data count: 914. Class 2 data count: 1063.



Gather Class 1 Data

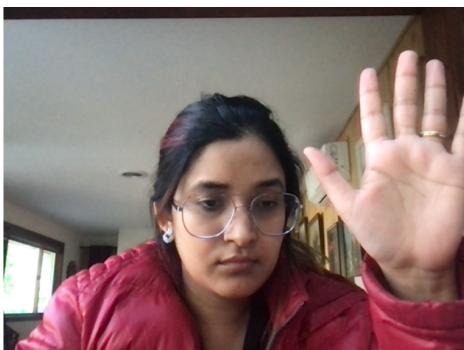
Gather Class 2 Data

Train & Predict!

Reset

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

Prediction: Class 1 with 95% confidence



Gather Class 1 Data

Gather Class 2 Data

Train & Predict!

Reset