TensorFlow.js

Machine Learning For The Web : An Introduction to Tensorflow.JS

Shivay Lamba

TensorFlow.js SIG & Working Group Member
TFUG New Delhi Mentor, Google Code In Mentor for TensorFlow

@howdevelop on Twitter!

Machine Learning in JavaScript

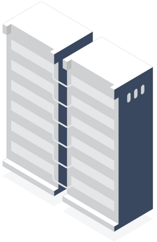
Why?

What is Tensorflow :

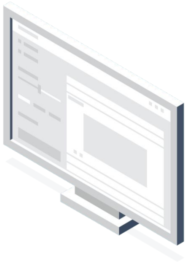
TensorFlow is a free and open-source software math library, and is used for machine learning applications such as neural networks.

Use ML anywhere JavaScript can run

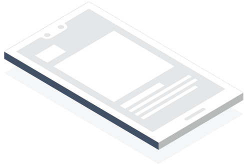
Browser



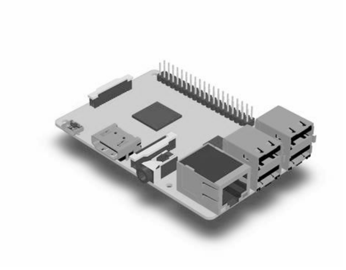
Server



Desktop



Mobile



IoT

Browser

Server

node
JS

Mobile

React Native

WeChat

PWA

Desktop

Electron

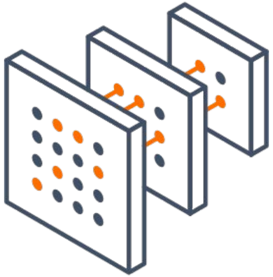
IoT

RaspberryPi

(via Node)

Run, Retrain, Write

Reuse existing models, or create your own

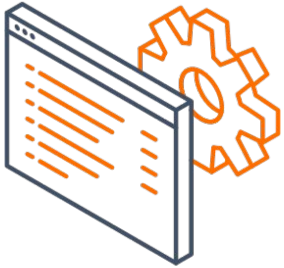


Run existing models

Pre-packaged JavaScript **or** Converted from Python

Retrain existing models

With transfer learning



Write models in JS

Train from scratch

For anything you may dream up

Augmented Reality

Gesture-based interaction

Sound recognition

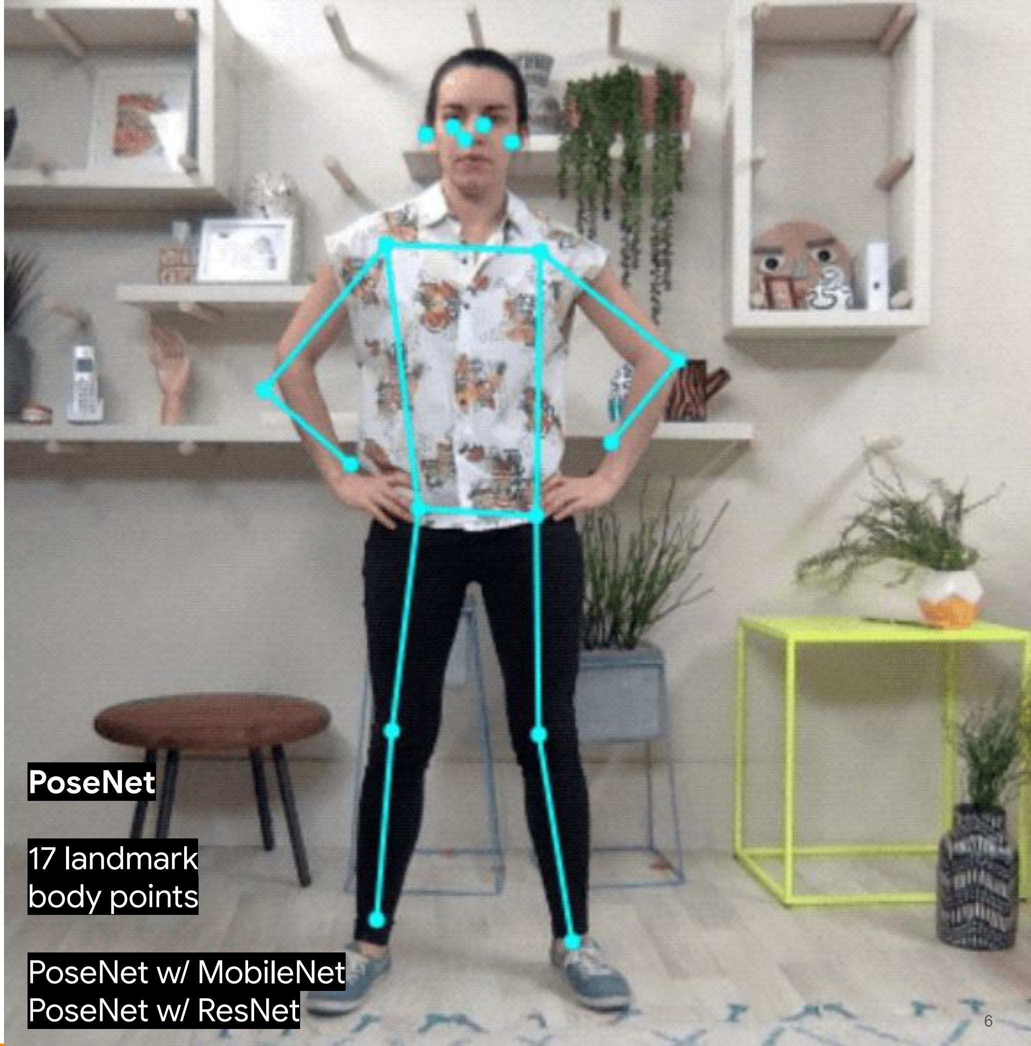
Accessible web apps

Sentiment analysis, abuse detection, NLP

Conversational AI

Web-page optimization

And much more...



PoseNet

17 landmark body points

PoseNet w/ MobileNet
PoseNet w/ ResNet

Pre trained models

Easy to use JavaScript classes for common use cases

We have several...

And continually expanding our collection.

+ Image classification

+ Object detection

+ Body Segmentation

+ Pose Estimation

+ Text Toxicity

+ Sentence encoding

+ Speech Commands

+ KNN Classifier

New models added:

- Face Mesh
- Hand Pose
- BERT Q&A

Let’s check some out...

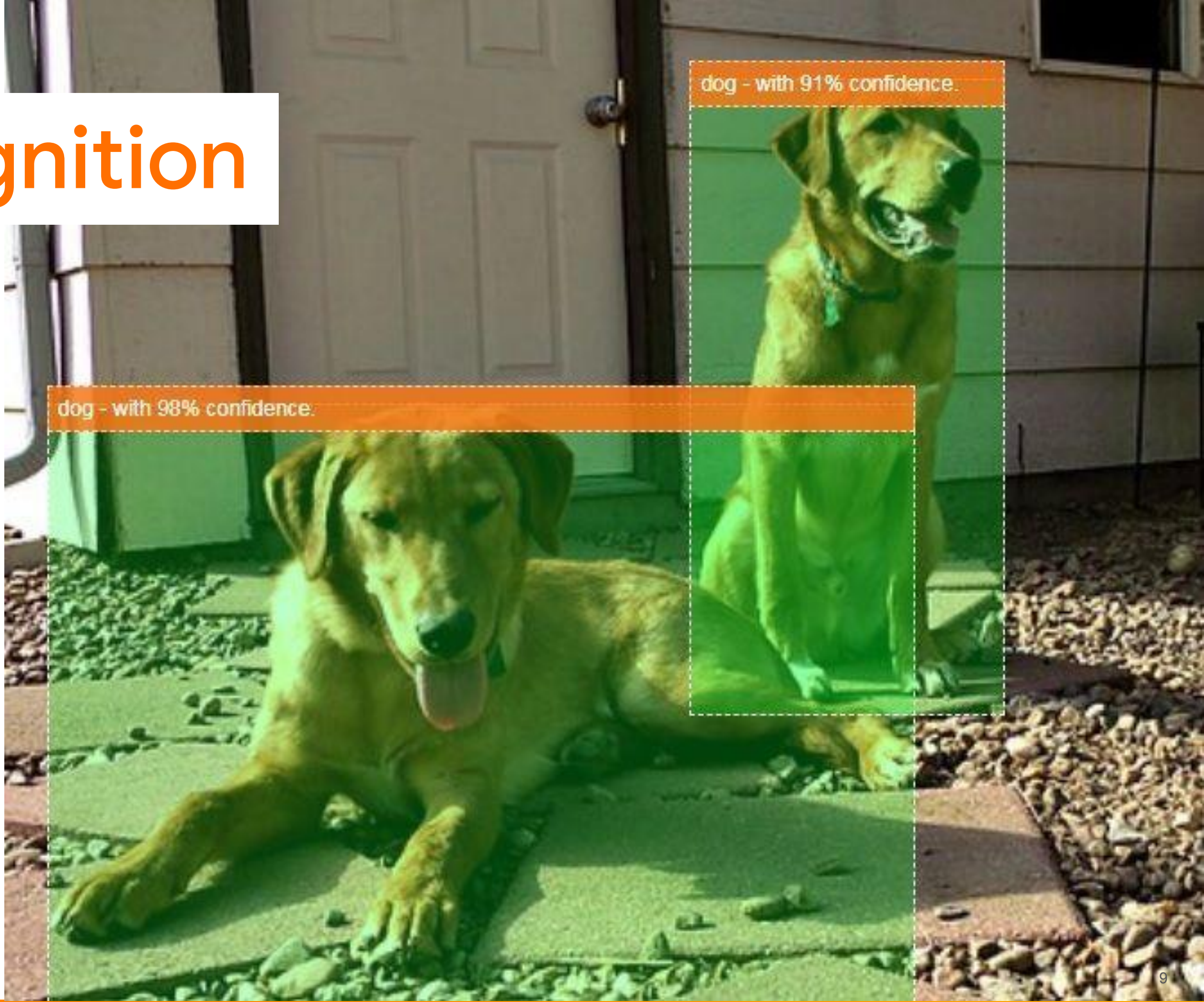
tensorflow.org/js/models

Object Recognition

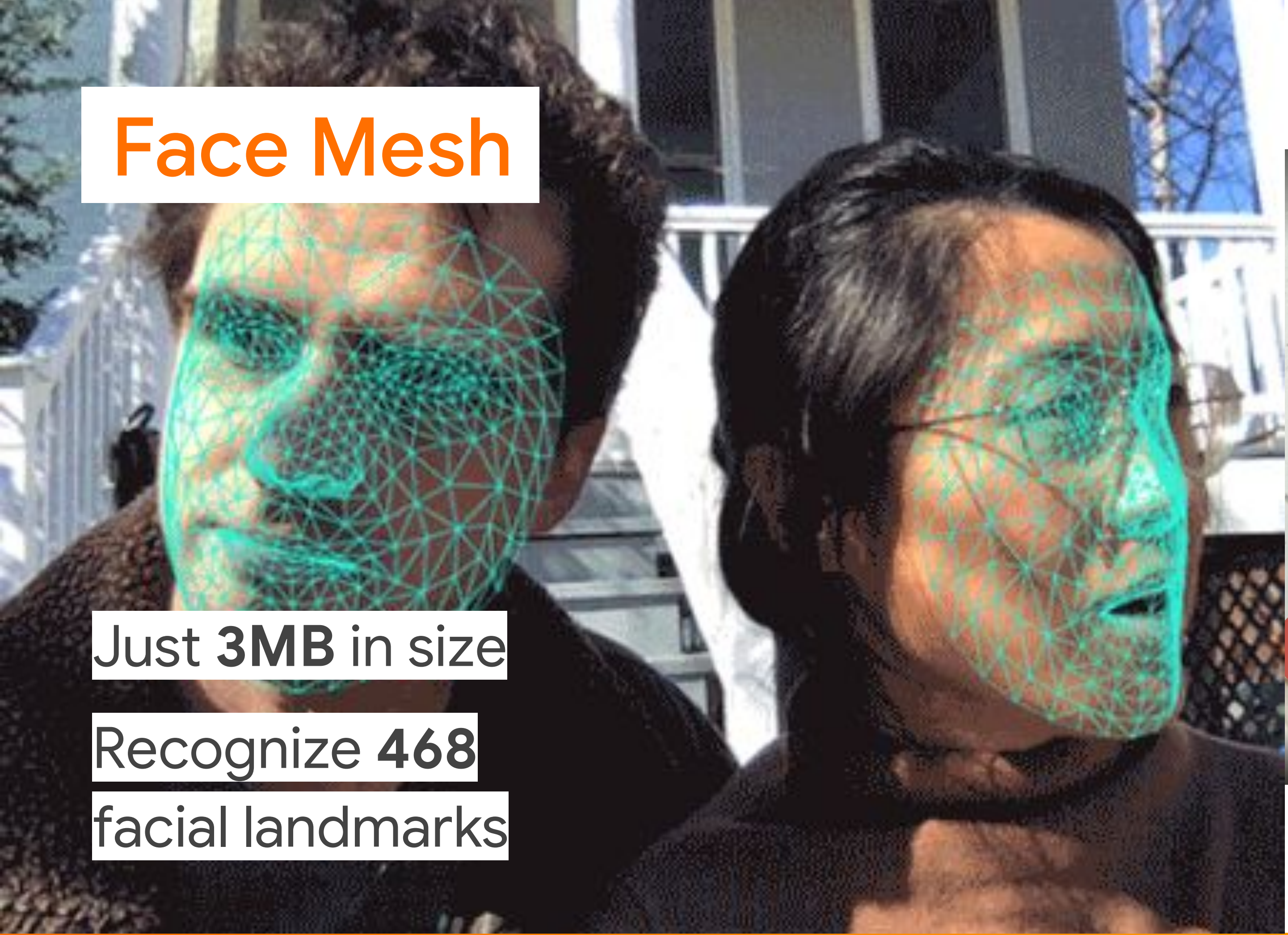
Using COCO-SSD

Trained on 90 object classes

Demo



Face Mesh

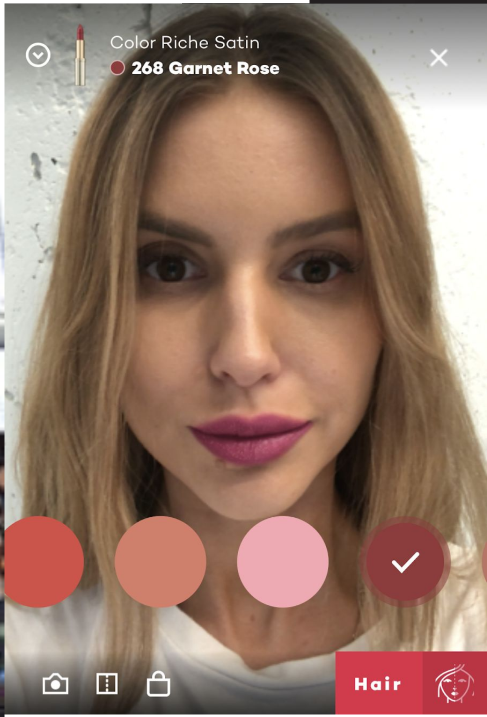


Just **3MB** in size

Recognize **468** facial landmarks

L'ORÉAL

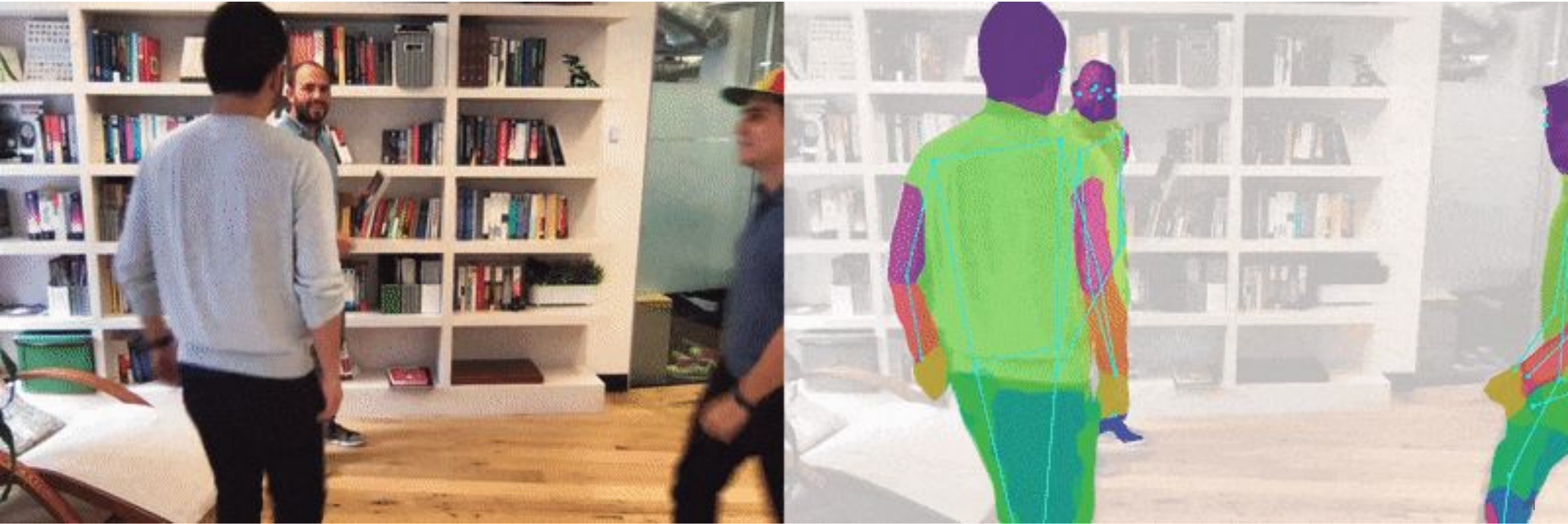
MODI FACE



Demo

Body Segmentation

Distinguish **24** body areas, across multiple bodies, real time.



With a little bit of creativity...

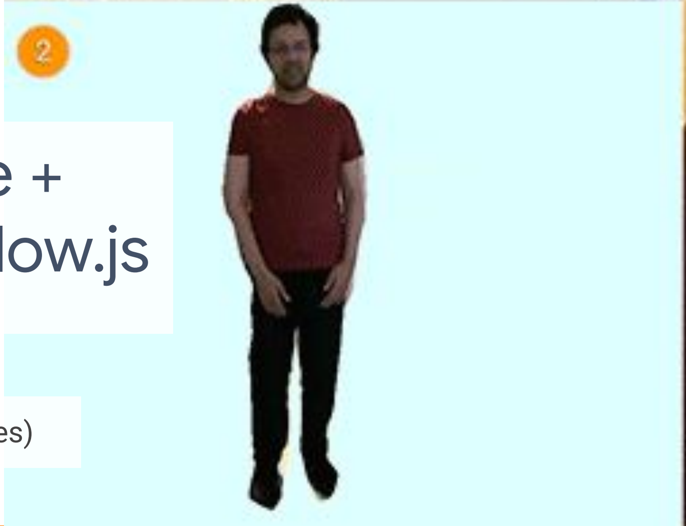
We can emulate the superpowers we were promised in the movies which we should have by now!

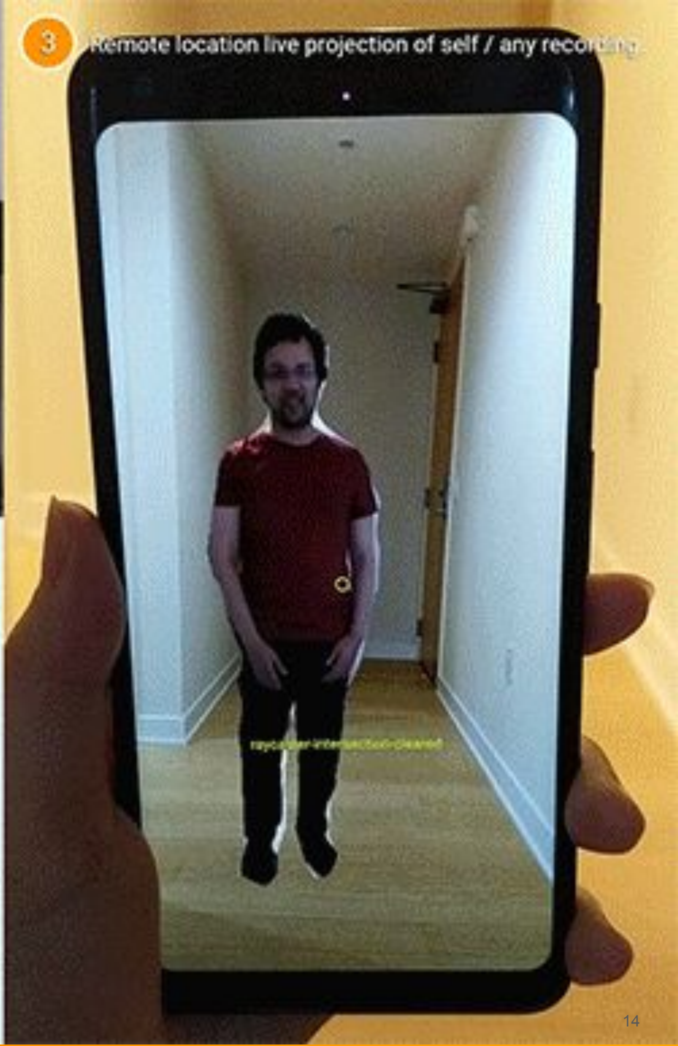
Lasers

Created by Ricky Reusser (@rickyreusser) + Max Bittker (@maxbittker), USA

WebGL Shaders
+ TensorFlow.js

Teleportation



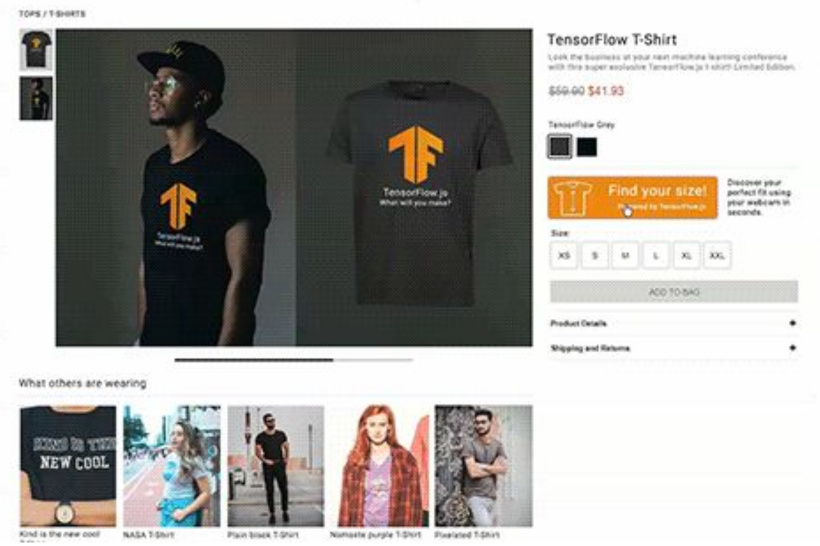

3
Remote location live projection of self / any recording
raycaster-intersection-cleared

WebRTC + AFrame + Three.js + TensorFlow.js

Created by Jason Mayes (@jason_mayes)

Or other delightful creations

Such as clothing size estimation



TOPS / T-SHIRTS

TensorFlow T-Shirt

Look the business at your next machine learning conference with this super exclusive TensorFlow.js t-shirt! Limited Edition.

~~$59.90~~ $41.93

TensorFlow Grey

■ ■

Find your size!
Powered by TensorFlow.js

Discover your perfect fit using your webcam in seconds.

Size

XS S M L XL XXL

ADD TO BAG

Product Details ✦

Shipping and Returns ✦

What others are wearing

Kind is the new cool

NASA T-Shirt

Plain black T-Shirt

Namaste purple T-Shirt

Pixelated T-Shirt

Created by Jason Mayes (@jason_mayes)

Combine with other web tech

Web XR + WebGL

+ TensorFlow.js

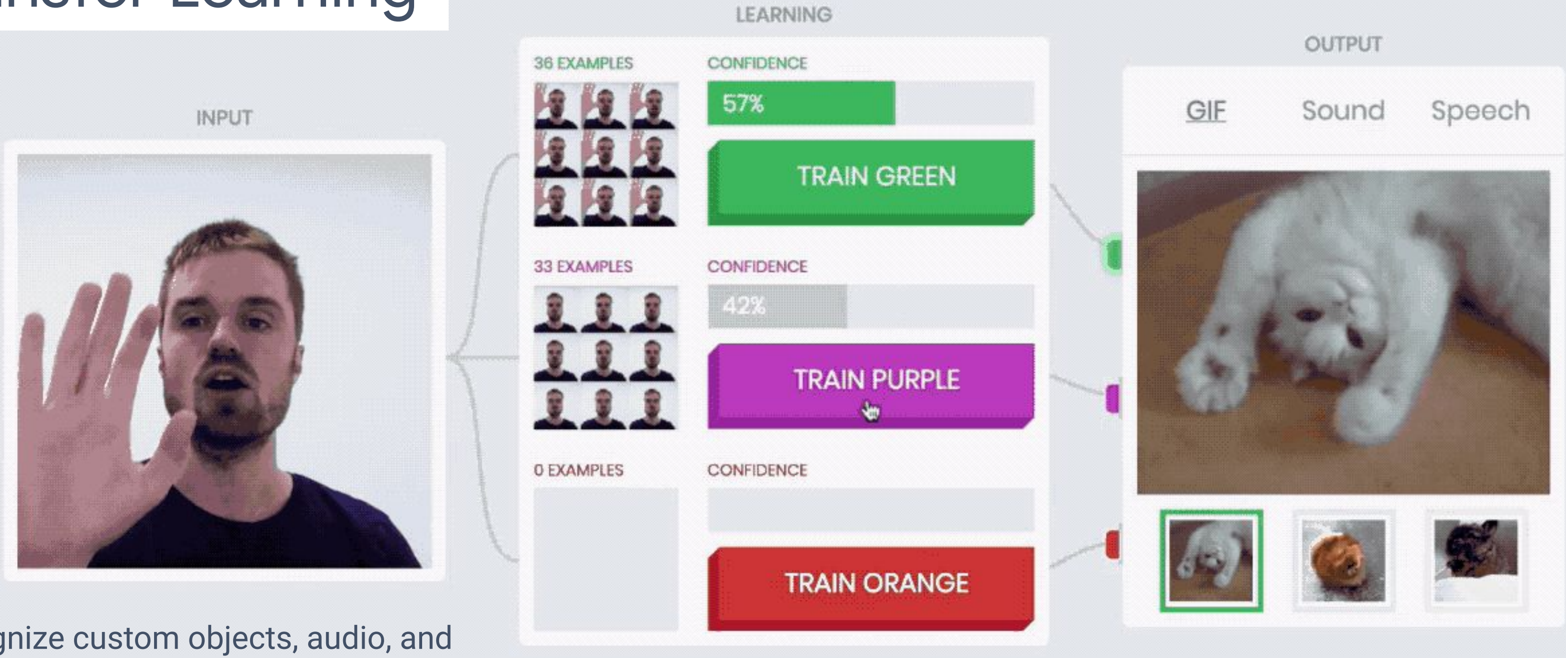
Created by Alexandre Devaux (@AlexandreDevaux), France

Transfer Learning

Retrain existing models to work with your own data

Teachable Machine

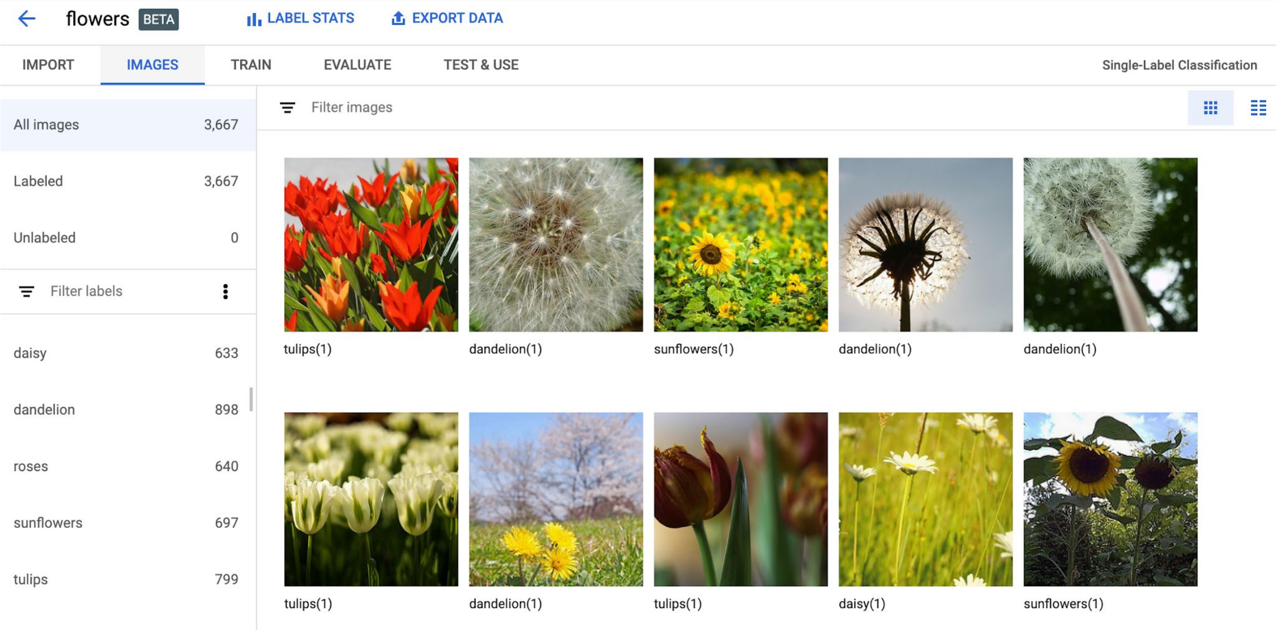
Transfer Learning



Recognize custom objects, audio, and poses in the browser in minutes. Demo.

Cloud Auto ML

Train custom vision models in the cloud + deploy to TF.js



All images	3,667
Labeled	3,667
Unlabeled	0
daisy	633
dandelion	898
roses	640
sunflowers	697
tulips	799

Code

```
<script src="//cdn.jsdelivr.net/npm/@tensorflow/tfjs/dist/tf.min.js"></script>
<script src="//cdn.jsdelivr.net/npm/@tensorflow/tfjs-automl/dist/tf-automl.min.js"></script>

<img id="daisy" crossorigin="anonymous"
src="//storage.googleapis.com/tfjs-testing/tfjs-automl/img_classification/daisy.jpg">

<script>
  async function run() {
   const model = await tf.automl.loadImageClassification('model.json');
   const image = document.getElementById('daisy');
   const predictions = await model.classify(image);
  }
  run();
</script>
```

TensorFlow.js: Write your own code

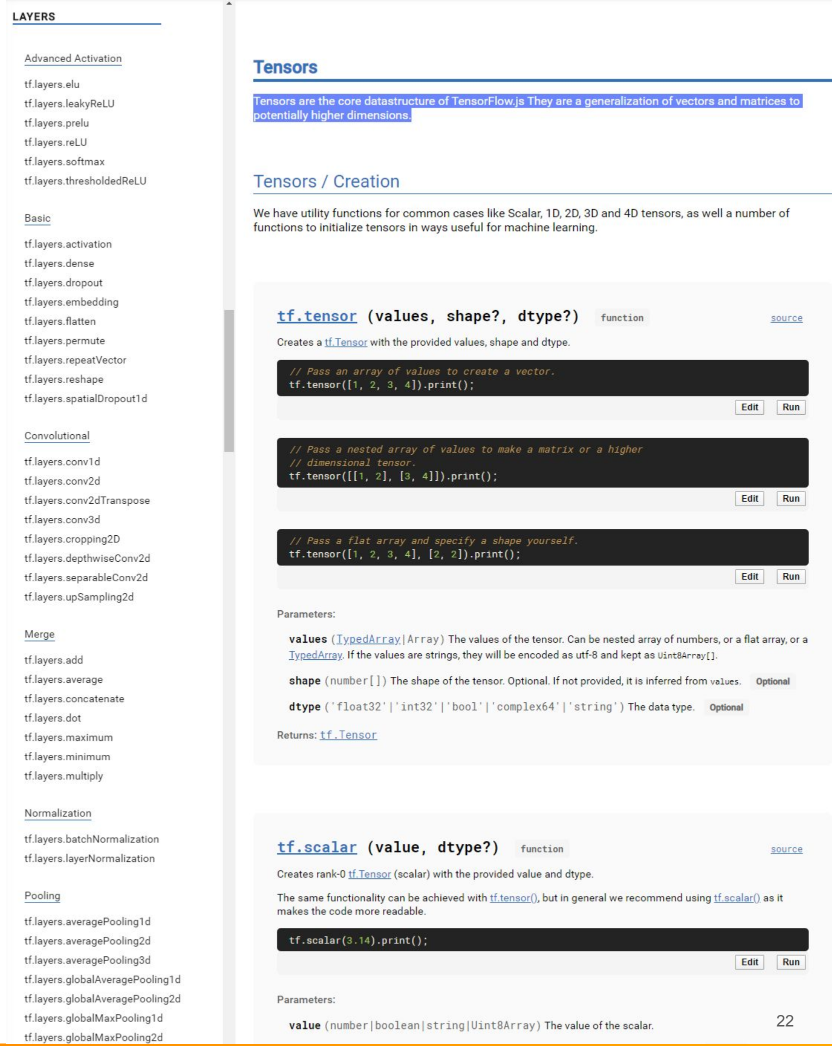
Super powers and performance

TensorFlow.js APIs

Create your own models with our APIs

Pick your flavour:

- High level **Layers API** (like Keras)
- Low level **Ops API** (mathematical)



LAYERS

Advanced Activation

- tf.layers.elu
- tf.layers.leakyReLU
- tf.layers.prelu
- tf.layers.reLU
- tf.layers.softmax
- tf.layers.thresholdedReLU

Basic

- tf.layers.activation
- tf.layers.dense
- tf.layers.dropout
- tf.layers.embedding
- tf.layers.flatten
- tf.layers.permute
- tf.layers.repeatVector
- tf.layers.reshape
- tf.layers.spatialDropout1d

Convolutional

- tf.layers.conv1d
- tf.layers.conv2d
- tf.layers.conv2dTranspose
- tf.layers.conv3d
- tf.layers.cropping2D
- tf.layers.depthwiseConv2d
- tf.layers.separableConv2d
- tf.layers.upSampling2d

Merge

- tf.layers.add
- tf.layers.average
- tf.layers.concatenate
- tf.layers.dot
- tf.layers.maximum
- tf.layers.minimum
- tf.layers.multiply

Normalization

- tf.layers.batchNormalization
- tf.layers.layerNormalization

Pooling

- tf.layers.averagePooling1d
- tf.layers.averagePooling2d
- tf.layers.averagePooling3d
- tf.layers.globalAveragePooling1d
- tf.layers.globalAveragePooling2d
- tf.layers.globalMaxPooling1d
- tf.layers.globalMaxPooling2d

Tensors

Tensors are the core datastructure of TensorFlow.js They are a generalization of vectors and matrices to potentially higher dimensions.

Tensors / Creation

We have utility functions for common cases like Scalar, 1D, 2D, 3D and 4D tensors, as well a number of functions to initialize tensors in ways useful for machine learning.

tf.tensor (values, shape?, dtype?) function source

Creates a tf.Tensor with the provided values, shape and dtype.

```
// Pass an array of values to create a vector.
tf.tensor([1, 2, 3, 4]).print();
```

Edit Run

```
// Pass a nested array of values to make a matrix or a higher
// dimensional tensor.
tf.tensor([[1, 2], [3, 4]]).print();
```

Edit Run

```
// Pass a flat array and specify a shape yourself.
tf.tensor([1, 2, 3, 4], [2, 2]).print();
```

Edit Run

Parameters:

- **values** (TypedArray|Array) The values of the tensor. Can be nested array of numbers, or a flat array, or a TypedArray. If the values are strings, they will be encoded as utf-8 and kept as `Uint8Array[]`.
- **shape** (number[]) The shape of the tensor. Optional. If not provided, it is inferred from `values`. Optional
- **dtype** ('float32'|'int32'|'bool'|'complex64'|'string') The data type. Optional

Returns: tf.Tensor

tf.scalar (value, dtype?) function source

Creates rank-0 tf.Tensor (scalar) with the provided value and dtype.

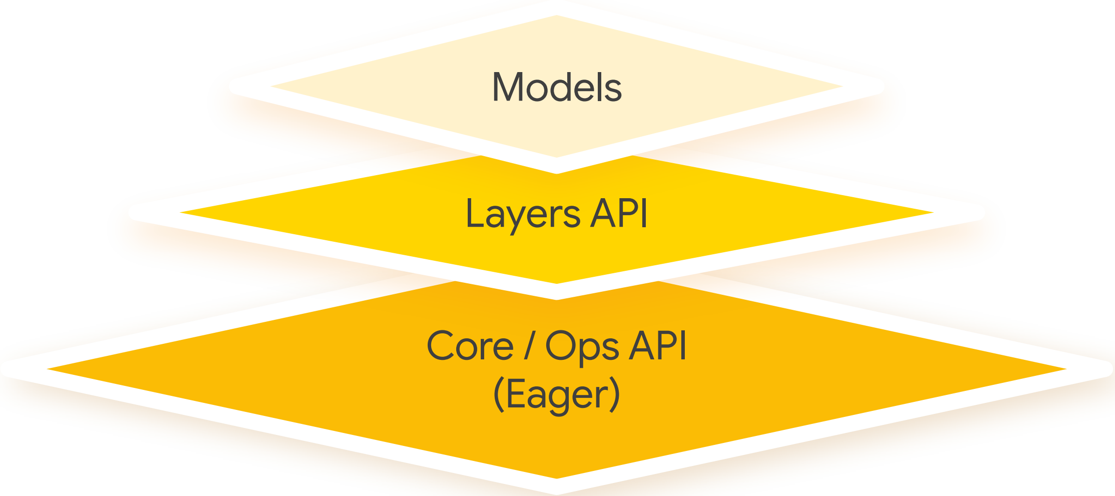
The same functionality can be achieved with tf.tensor(), but in general we recommend using tf.scalar() as it makes the code more readable.

```
tf.scalar(3.14).print();
```

Edit Run

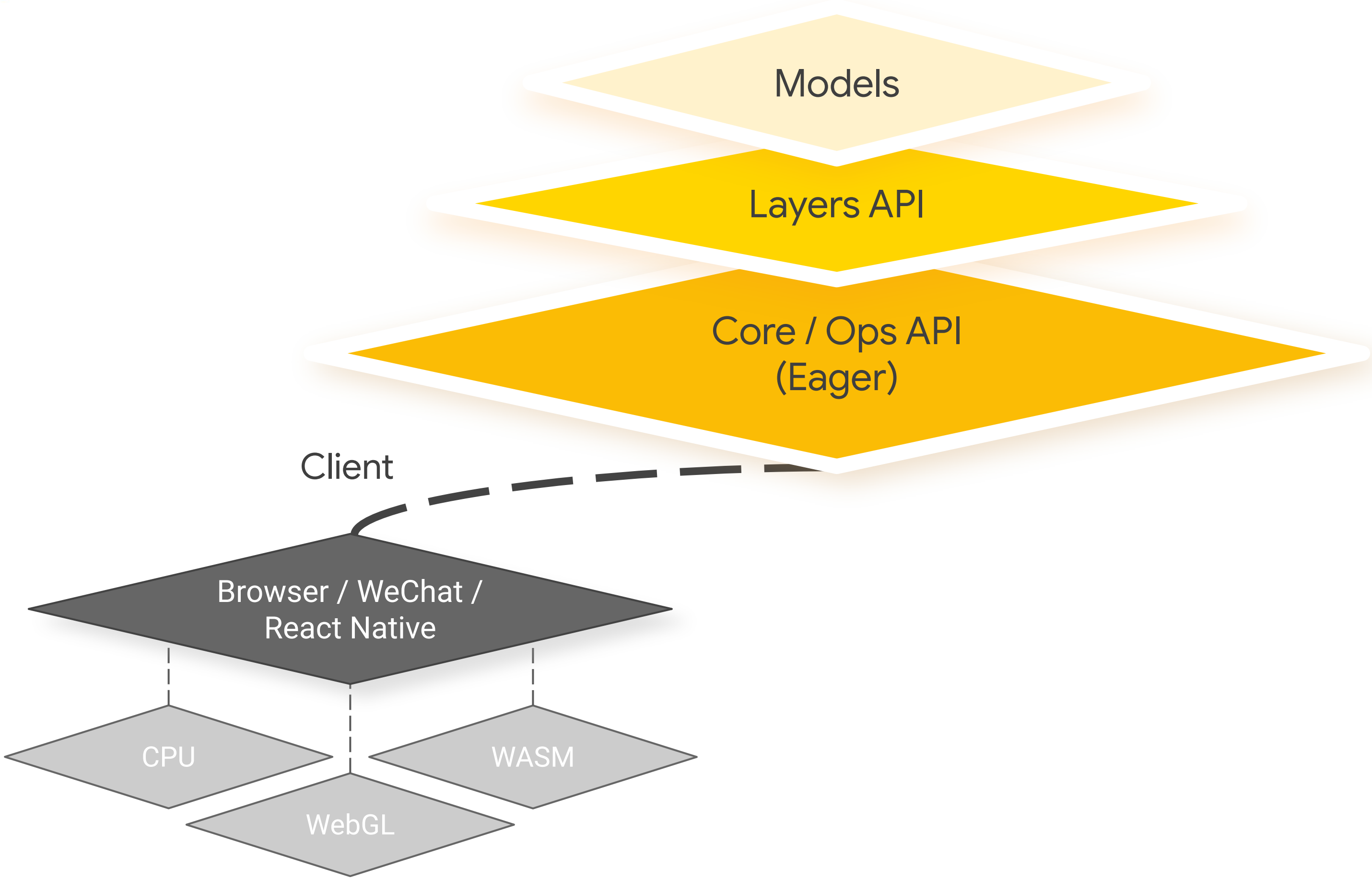
Parameters:

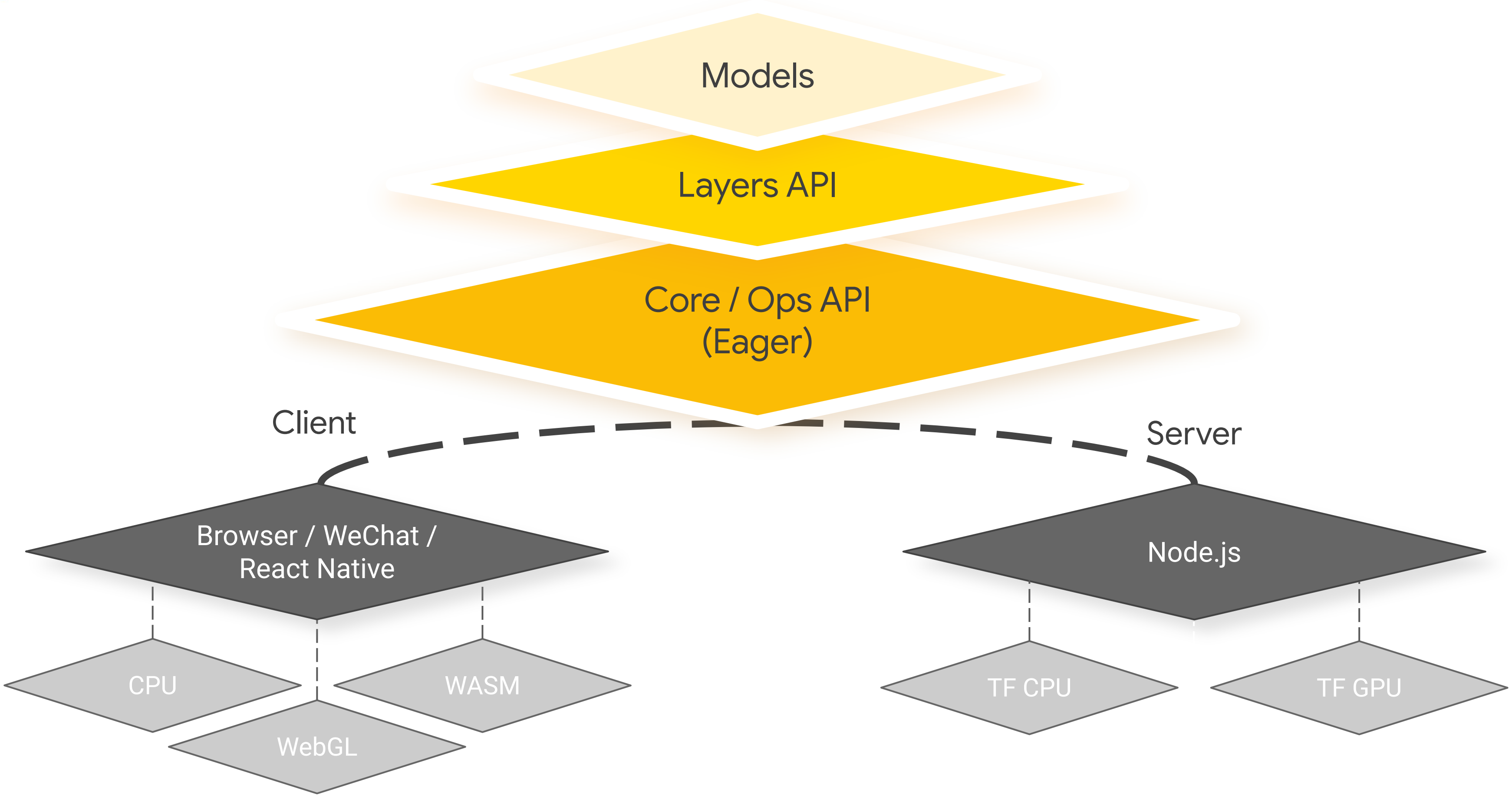
- **value** (number|boolean|string|Uint8Array) The value of the scalar.

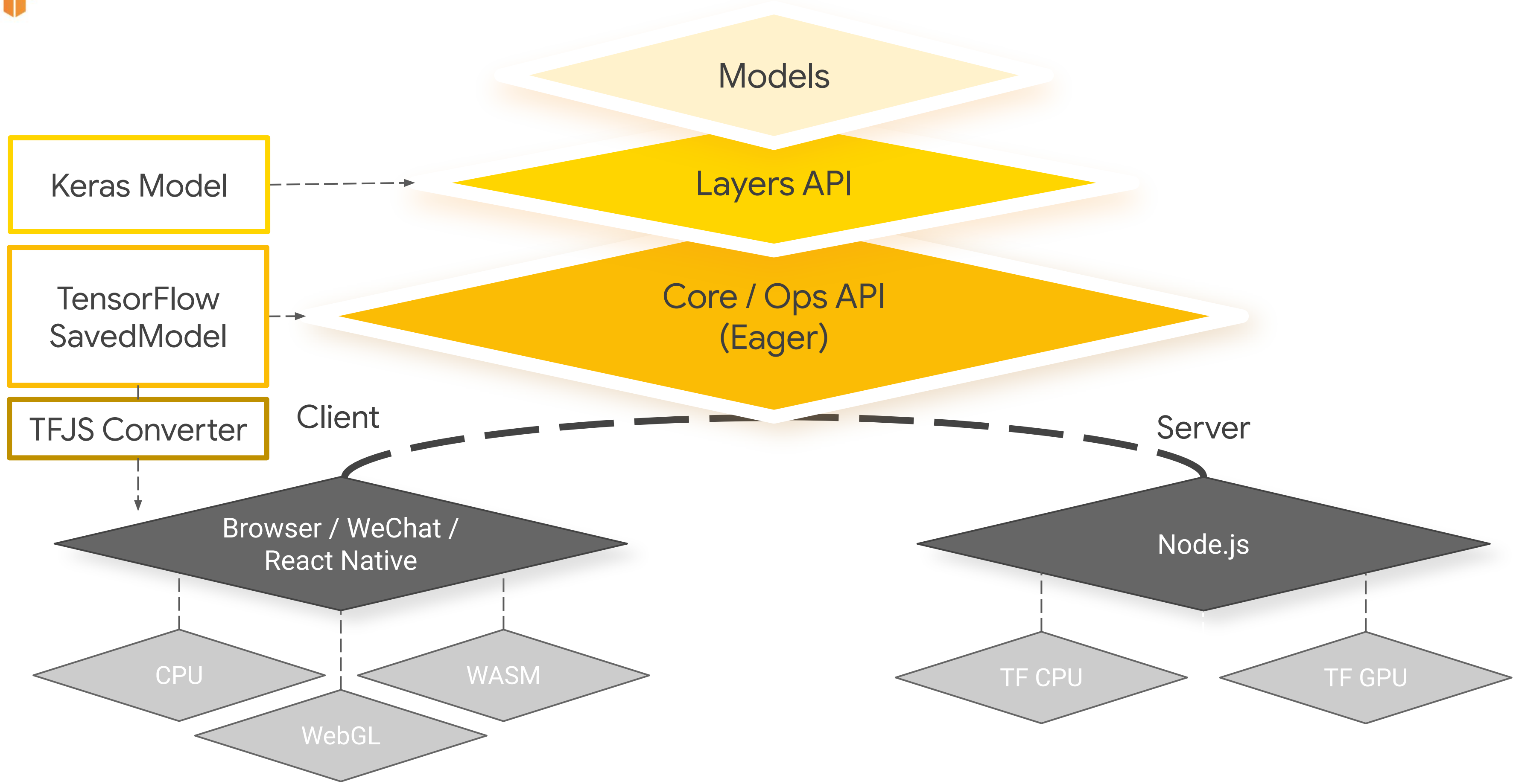

Models

Layers API

Core / Ops API
(Eager)

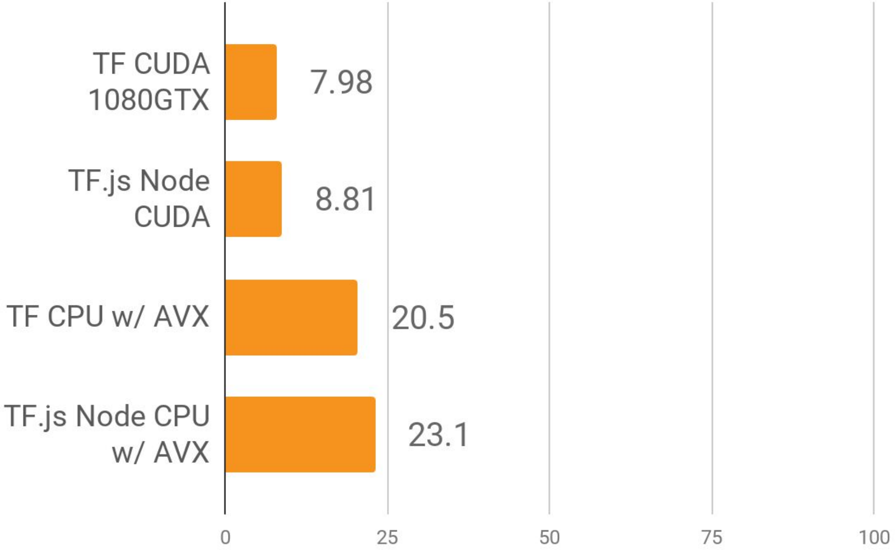

Models
Layers API
Core / Ops API
(Eager)
Client
Browser / WeChat /
React Native
CPU
WASM
WebGL


Models
Layers API
Core / Ops API
(Eager)
Client
Server
Browser / WeChat /
React Native
Node.js
CPU
WASM
WebGL
TF CPU
TF GPU


Models
Layers API
Keras Model
TensorFlow SavedModel
Core / Ops API (Eager)
TFJS Converter
Client
Server
Browser / WeChat / React Native
Node.js
CPU
WASM
WebGL
TF CPU
TF GPU

Model Inference Performance Only

Server

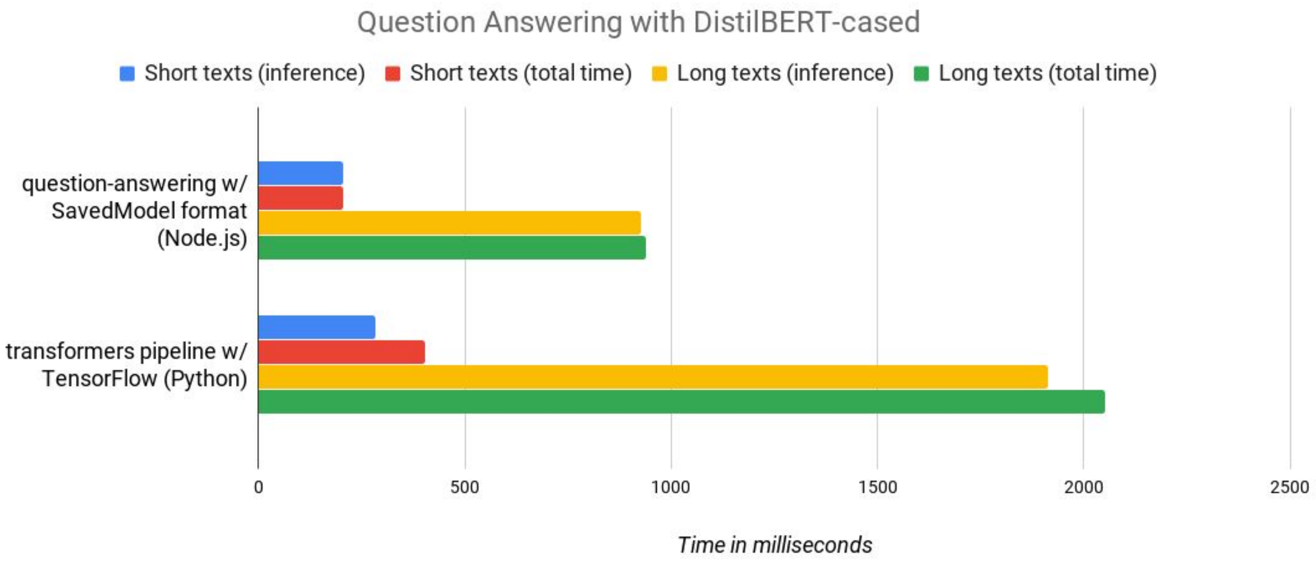


MobileNetV2 1.0_224 model inference time (ms)

Note: *If you have a lot of pre / post processing written in Node.js, you will get the benefits of the JIT at runtime, which can be significant outside of inference alone.*

Hugging Face DistilBERT

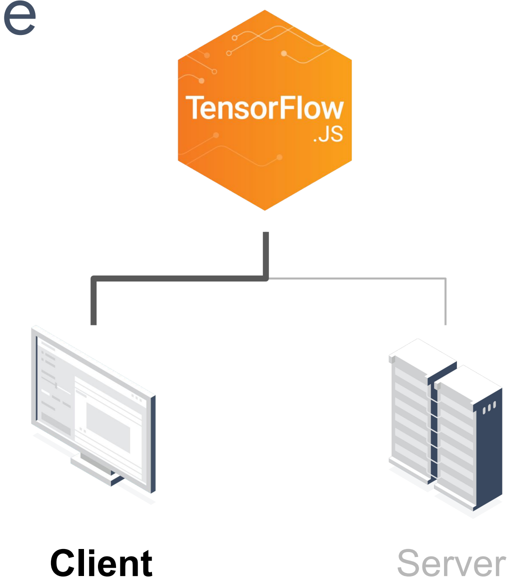
2x Perf boost using Node.js



5 client side super powers

Harder / impossible to achieve server side

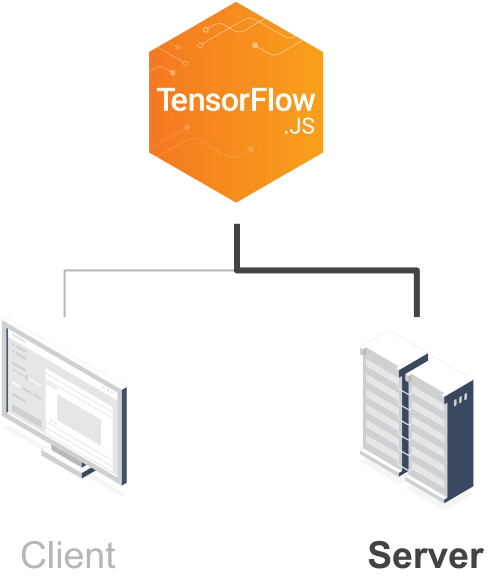
1. Privacy
2. Lower Latency
3. Lower Cost
4. Interactivity
5. Reach and Scale
   (GPU support on 84% of devices via WebGL)



Server side / Node.js benefits

That make TensorFlow.js easy to use

1. Use TensorFlow SavedModel without conversion
2. Run larger models than client side (GPU memory limits)
3. Code in 1 language - if you already use JS
   (67.8% of people use JS in development already)
4. Performance - C bindings / JIT boost for pre/post processing



Resources

Learn more and get inspired

Learn more

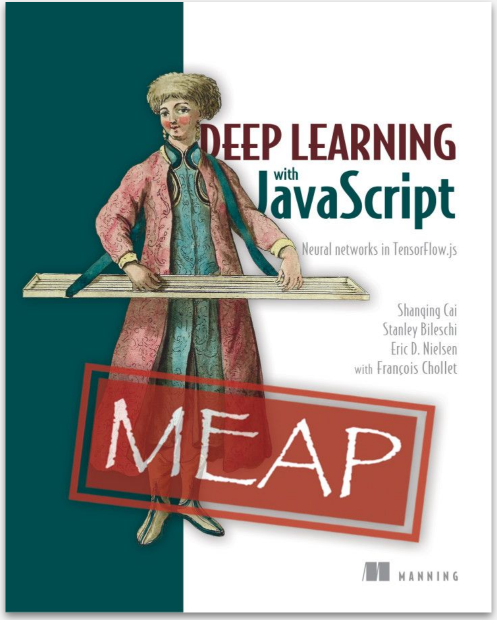
Get started fast!

Website / API:	*tensorflow.org/js*
Models:	*tensorflow.org/js/models*
Github Code:	*github.com/tensorflow/tfjs*
Google Group:	*tfjs@tensorflow.org*
Codepen:	*codepen.io/topic/tensorflow*
***Glitch**:*	*glitch.com/@TensorFlowJS*

Recommended Reading

Deep Learning with JavaScript

manning.com/books/deep-learning-with-javascript



Join the community

#MadeWithTFJS

Thanks to the global community for use of videos shown on this slide: *Jimmy (@MLBlock3), Manish Raj (@manishr41883690), Ben Farrell (@bfarrellforever), Wen (@yiwen_lin), Junya Ishihara (@jishiha), Rogerio Chaves (@_rchaves_), Amruta (@prenalys), FollowTheDarkside (@eatora22), and Alexandre Devaux (@AlexandreDevaux).*

What will you make?

Machine Learning
for everyone.

#MadeWithTFJS

Video by FollowTheDarkside, (@eatora22), Japan

Stay in touch

@howdevelop linkedin.com/in/shivaylamba

#MadeWithTFJS

TensorFlow