



Detectando uso de máscara em tempo real com TensorflowJS



TRANSFORMATION

COMPETÊNCIAS PARA EVOLUÇÃO
DAS PESSOAS E NEGÓCIOS

Bio

Machine Learning Engineer

Mestre pela FEEC (Unicamp)

Aprendiz frustrado de surfista



Rafael Ribeiro

Data Scientist | Data Engineer | Machine Learning Engineer



Agenda

- **Objetivo**
- **Contexto**
 - Como olhar imagens
 - Diferentes aplicações com imagens
 - Redes neurais convolucionais
 - Transferência de aprendizado
- **TensorflowJS**
 - Iniciando
 - Carregando modelo
 - Treinamento
 - Inferência/Predição
 - Demonstração
- **Conclusão**
- **Referências**

Objetivo



Com Máscara



Sem Máscara

Diferente aplicações



Diferentes aplicações

Classificação



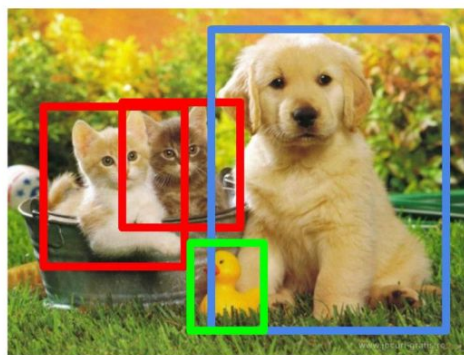
Gato

**Classificação
+ Localização**



Gato

**Detecção de
Objetos**



Gato, Cachorro, Pato

**Segmentação de
Instância**



Gato, Cachorro, Pato

Único Objeto

Múltiplos Objetos

Como computador vê



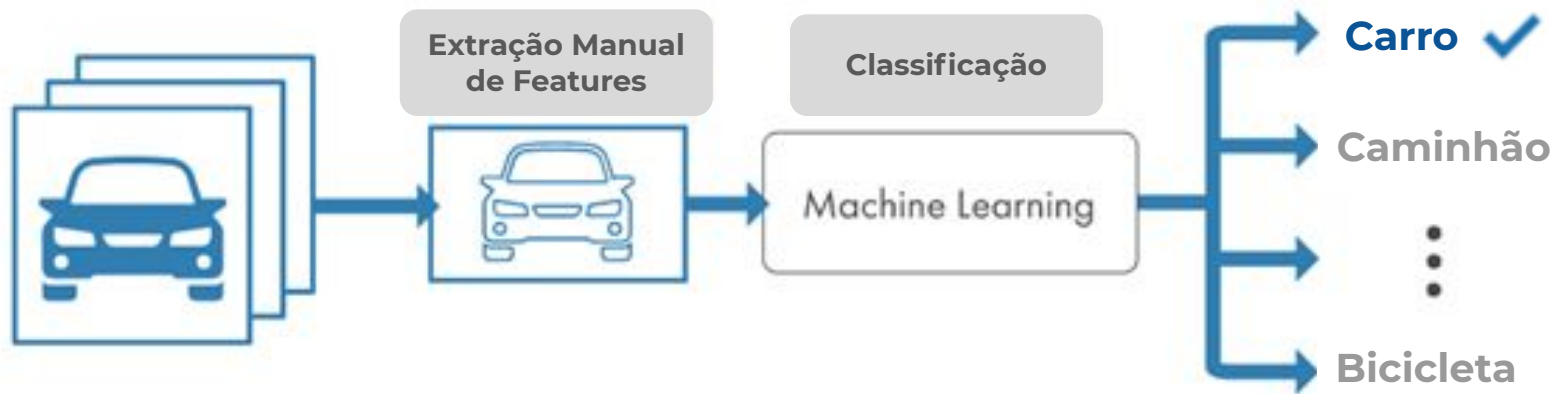
54	42	48	36	7	78	42	21	44	35	15	28	7	80
97	33	60	38	96	15	2	90	12	7	93	45	87	85
81	48	67	66	88	22	75	99	87	83	73	40	66	96
31	49	58	85	80	31	51	99	36	5	57	81	57	75
21	55	65	17	59	15	20	19	88	74	0	27	26	35
55	75	37	13	46	70	42	35	13	98	35	78	92	27
52	60	81	38	56	56	79	89	6	43	71	67	24	66
33	22	71	12	56	15	0	79	46	17	87	17	15	88
11	31	33	78	54	78	70	43	55	24	84	49	89	76
52	66	93	53	9	33	23	51	23	90	27	98	74	82
17	7	24	25	96	31	3	67	78	61	96	86	99	12
86	55	81	70	7	61	48	39	13	64	38	37	40	93
84	24	70	29	21	34	41	82	9	43	77	74	58	91
69	17	38	15	32	46	9	60	60	21	7	58	25	97

**Classificação
da
Imagem**

85% COM Máscara
15% SEM Máscara

Extração Manual de Features

Machine Learning



Desafios encontrados

Ponto de vista



Escala



Deformação



Oclusão



Iluminação



Confusão

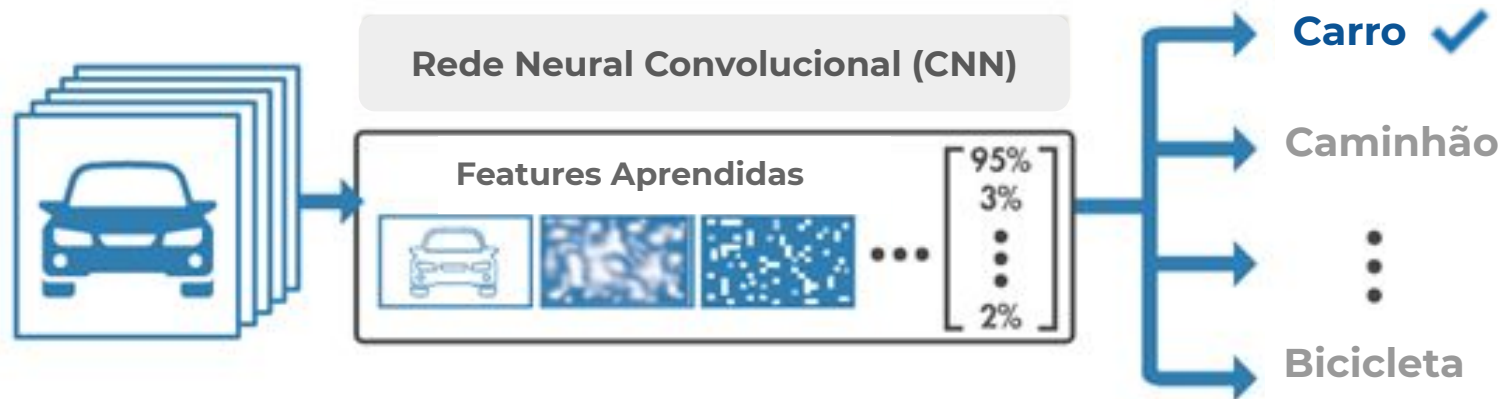


Variação classe

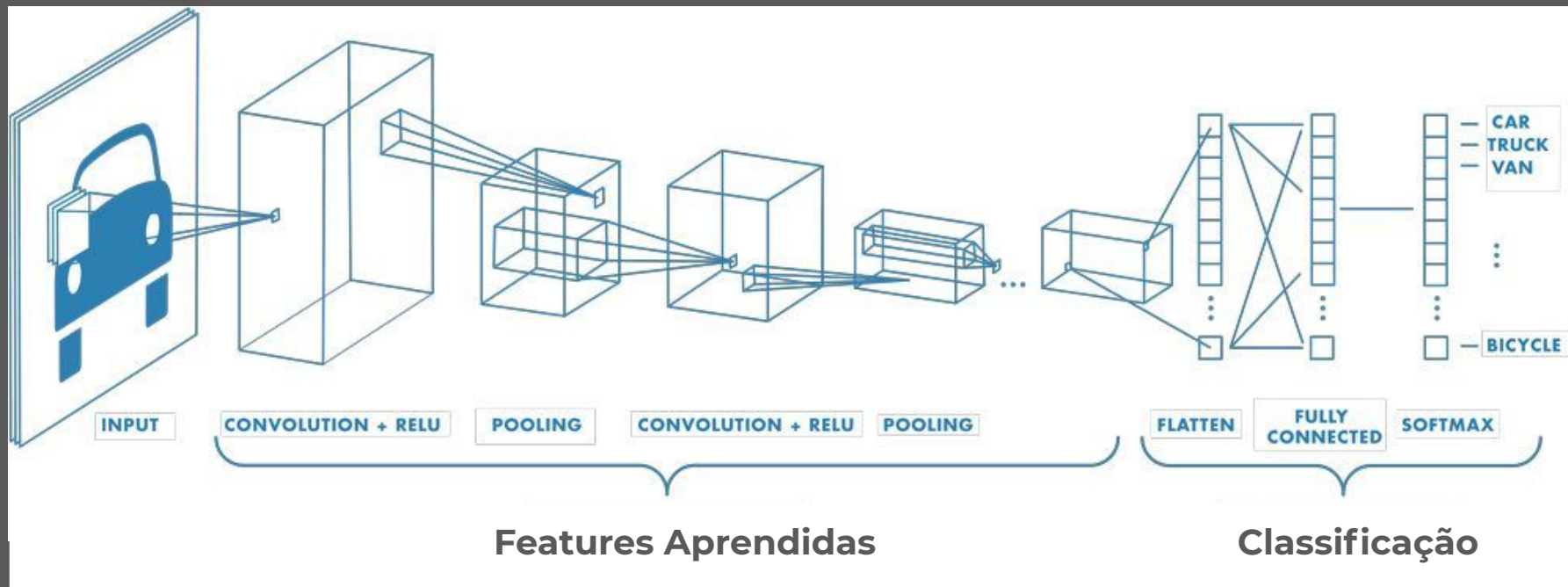


Redes Neurais Convolucionais

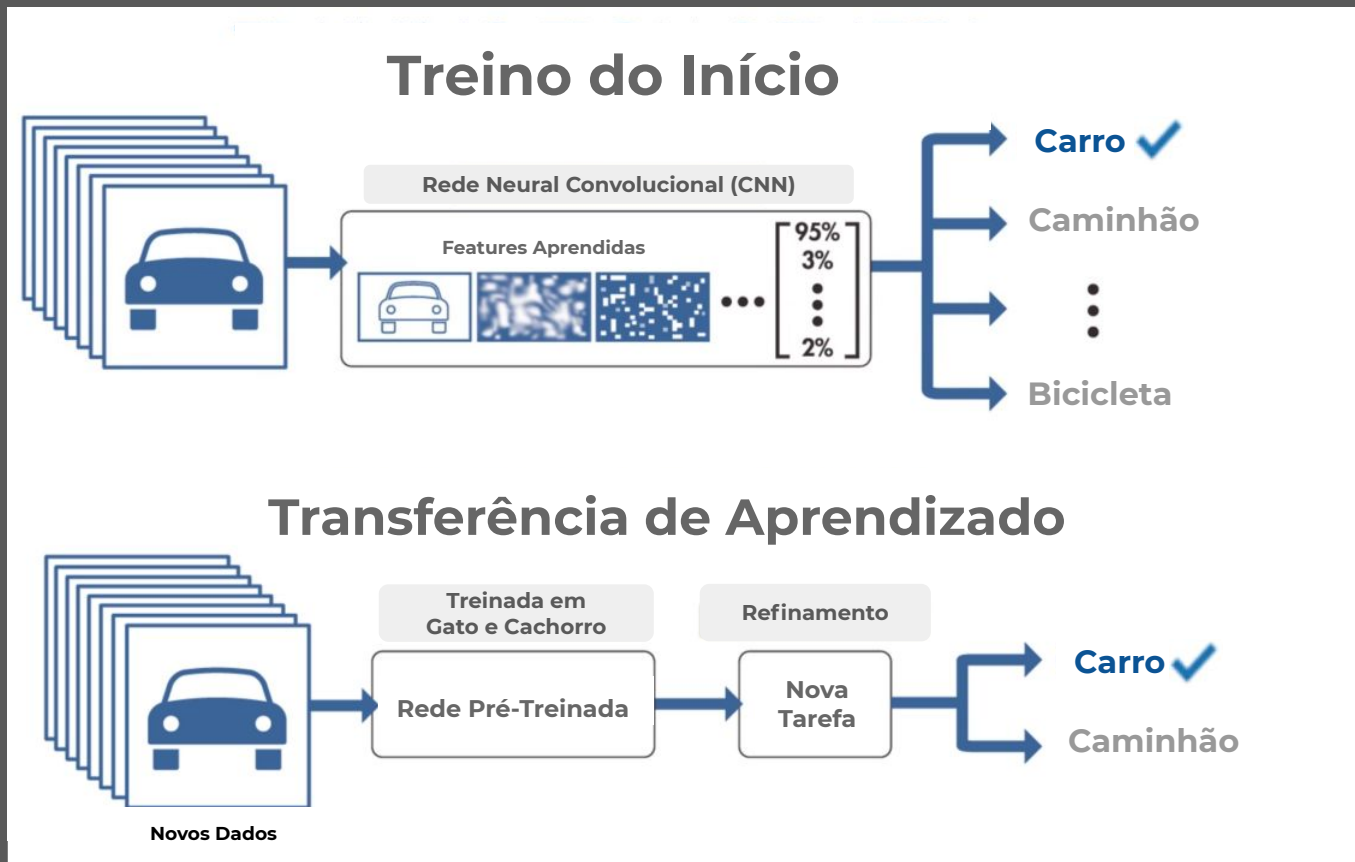
Deep Learning



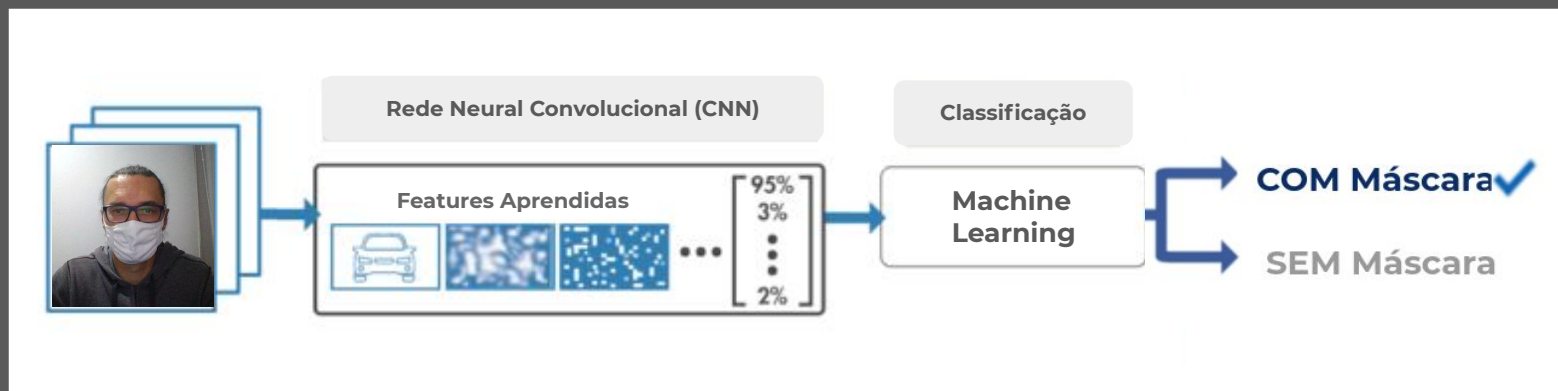
Detalhes de uma CNN



Transferência de Aprendizado

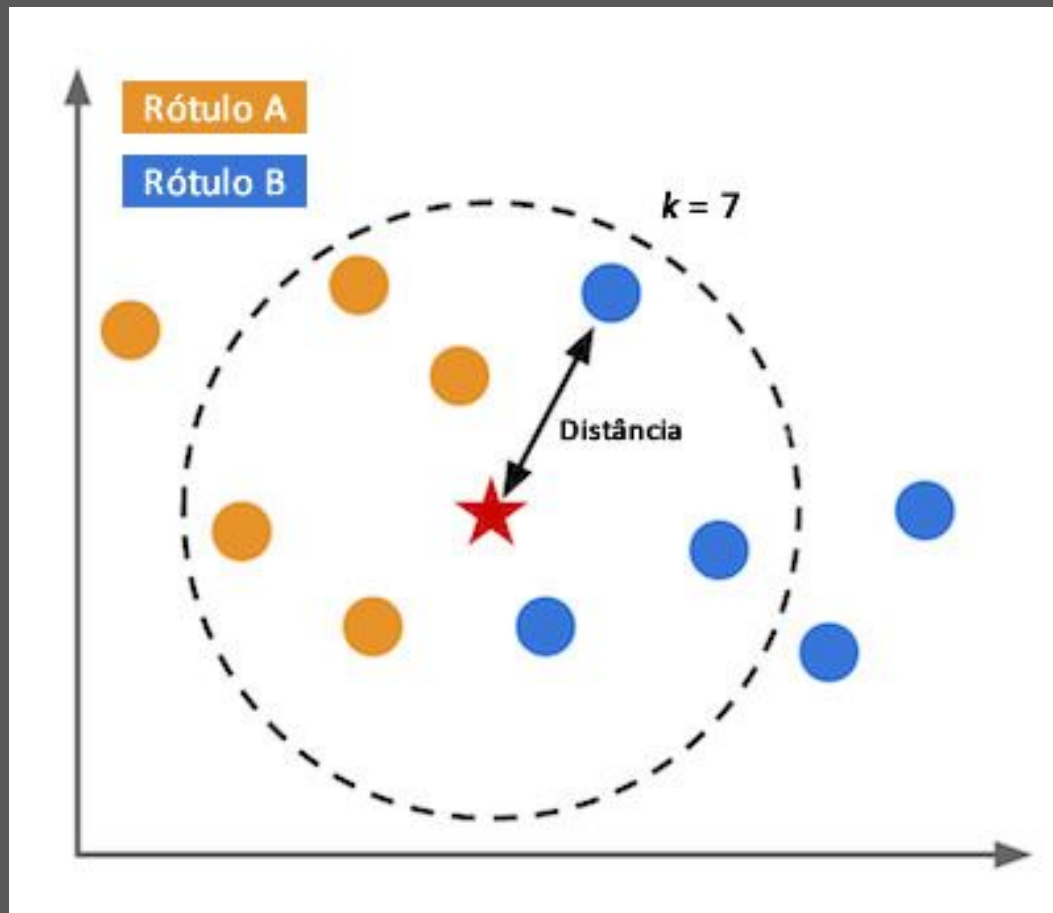


Extrator de Features



K-Nearest Neighbor

- Distância entre vetores
- A maioria entre os K mais próximos será o resultado



TensorflowJS

É uma biblioteca para machine learning em JavaScript

Como funciona



Executar modelos existentes

Use modelos do JavaScript prontos ou converta modelos do TensorFlow em Python para que sejam executados no navegador ou pelo Node.js.



Treinar novamente modelos existentes

Treine modelos de ML pré-existentes com seus próprios dados.



Desenvolva ML com JavaScript

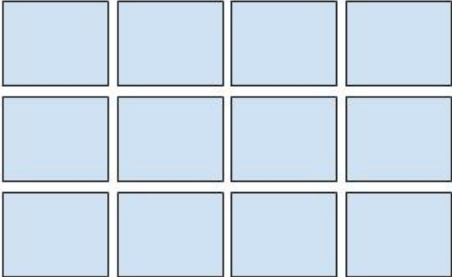
Use APIs intuitivas e flexíveis para criar e treinar modelos diretamente em JavaScript.

Tensoros

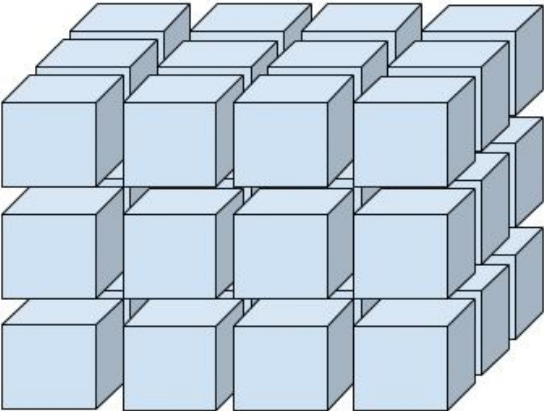
Rank 0: 
(scalar)

Rank 1: 
(vector)

Rank 2: (matrix)



Rank 3:



Carregando TensorFlowJS e dependências (HTML)

```
2 <head>
3   <!-- Load the latest version of TensorFlow.js -->
4   <script defer src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
5   <script defer src="https://cdn.jsdelivr.net/npm/@tensorflow-models/mobilenet"></script>
6   <script defer src="https://cdn.jsdelivr.net/npm/@tensorflow-models/knn-classifier"></script>
7
8   <meta name="viewport" content="width=device-width, initial-scale=1.0">
9   <link rel="stylesheet" href="styles.css">
10  <title>Mask detector</title>
11 </head>
```

Adicionando tag de Vídeo (HTML)

```
16 <div class="console" id="main">
17   <div class="container-video">
18     <video class="no-result" autoplay playsinline muted id="webcam"
19       width="300" height="300"></video>
20     <div class="text-class-result">Esperando</div>
21   </div>
22 > <div class="buttons">...
26   </div>
27 </div>
```

Inicializando (Javascript)

```
1  const webcamElement = document.getElementById('webcam');  
2  const videoContainerElement = document.querySelector('.container-video')  
3  const classResultElement = document.querySelector('.text-class-result')  
4  
5  let net;  
6  const classifier = knnClassifier.create();
```

Carregando Modelo e dados (Javascript)

```
43  async function app() {  
44  
45      // Load the model.  
46      net = await mobilenet.load();  
47      console.log('Successfully loaded model');  
48  
49      // Create an object from Tensorflow.js data API  
50      // which could capture image from the web camera as Tensor.  
51      const webcam = await tf.data.webcam(webcamElement);  
52  }
```

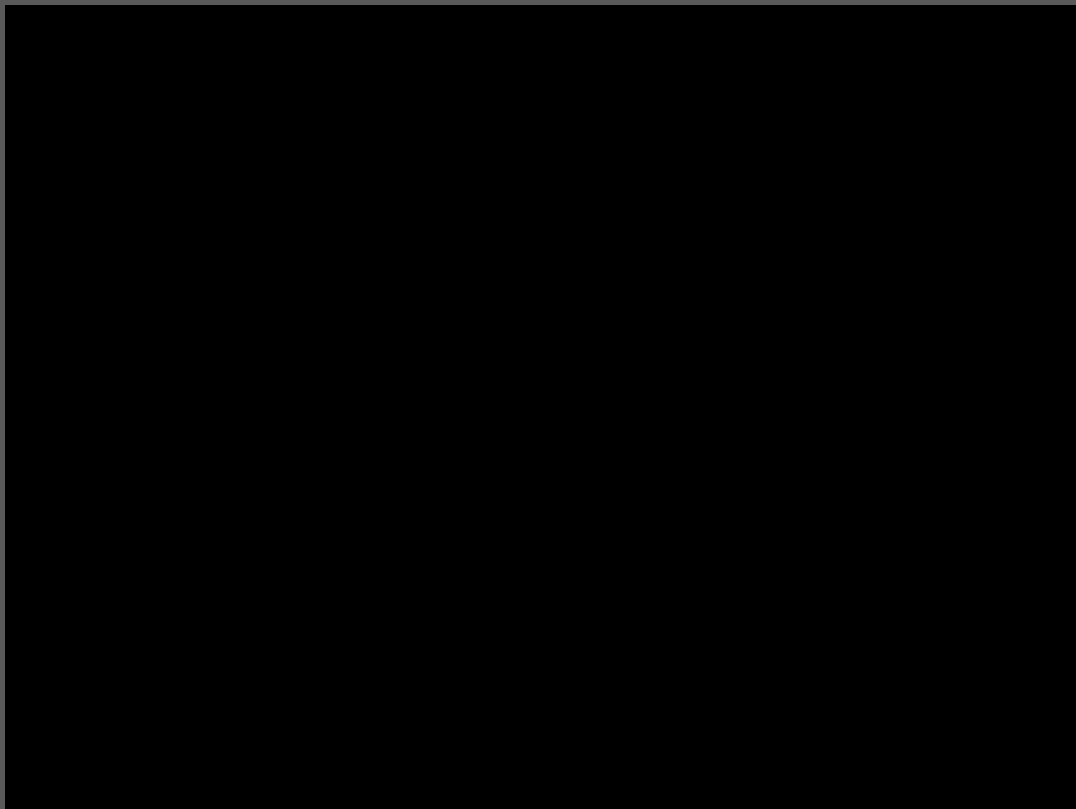
Realizando a classificação (Javascript)

```
78   while (true) {
79       if (classifier.getNumClasses() === 3) {
80           const img = await webcam.capture();
81           // Get the activation from mobilenet from the webcam.
82           const activation = net.infer(img, 'conv_preds');
83           // Get the most likely class and confidence from the classifier module.
84           const result = await classifier.predictClass(activation, k=3);
85           const classes = ['Sem Mascara', 'Neutro', 'Com Mascara'];
86           console.log(result.label, result.label === classes[0])
87 >   { /* ...
108       */}
109       // Dispose the tensor to release the memory.
110       img.dispose();
111   }
112   await tf.nextFrame();
113 }
```


Adicionando mais imagens (Javascript)

```
53 // Reads an image from the webcam and associates it with a specific class Index
54 const addExample = async classId => {
55     // Capture an image from the web camera.
56     const img = await webcam.capture();
57     takepicture(classId)
58     // Get the intermediate activation of MobileNet 'conv_preds' and pass that
59     // to the KNN classifier.
60     const activation = net.infer(img, true);
61     // Pass the intermediate activation to the classifier.
62     classifier.addExample(activation, classId);
63     // Dispose the tensor to release the memory.
64     img.dispose();
65 };
66 // When clicking a button, add an example for that class.
67 document.getElementById('class-a').addEventListener('click', () => addExample(0));
68 document.getElementById('class-b').addEventListener('click', () => addExample(1));
69 document.getElementById('class-c').addEventListener('click', () => addExample(2));
```

Versão Final (Javascript)



Conclusão

Prós

- Biblioteca poderosa para treinar e predizer usando Javascript
- Nível muito bom de maturidade
- Comunidade grande
- Rápido para visualizar o resultado

Contra

- Modelo fica disponível na máquina cliente

Perguntas



Referências

<https://www.tensorflow.org/>

<https://github.com/tensorflow/tfjs-models/>

<https://github.com/tensorflow/tfjs-models/tree/master/knn-classifier>

<https://codelabs.developers.google.com/codelabs/tensorflowjs-teachablemachine-codelab/index.html#0>

<https://medium.com/tensorflow/a-gentle-introduction-to-tensorflow-js-dba2e5257702>

<https://github.com/rfribeiro/ml-projects>