TensorFlow

Criando Androids mais inteligentes

Luiz GUStavo

TensorFlow Developer Advocate

@gusthema

Perguntas:

www.slido.com

Code: #88857

TensorFlow

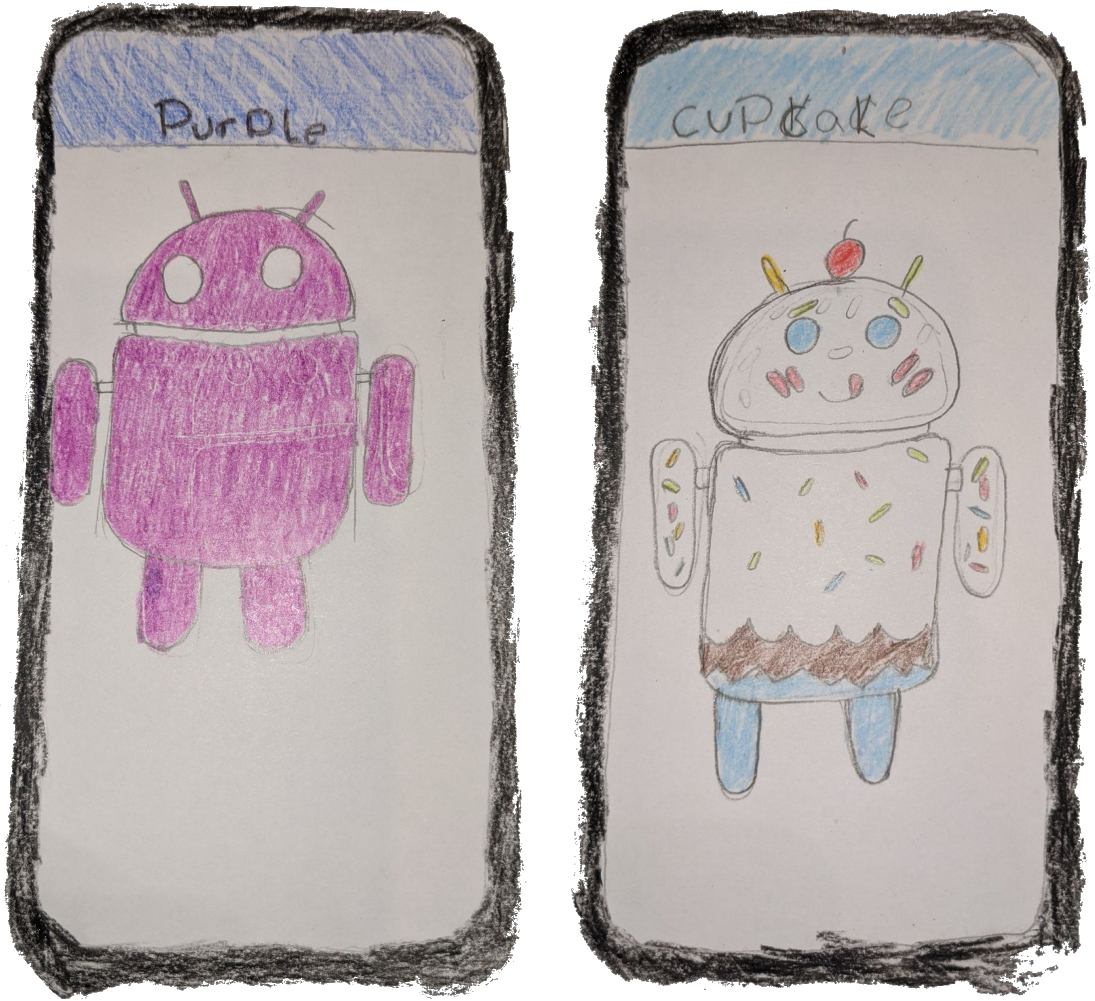
...for Science!

an Android™ mini collectible special edition series

...for Science²!

an Android™ mini collectible special edition series

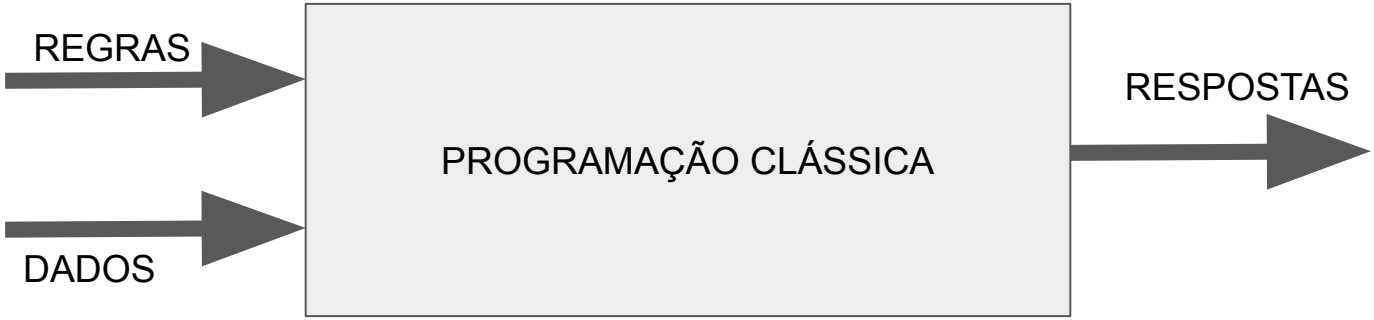
Meu Sonho:

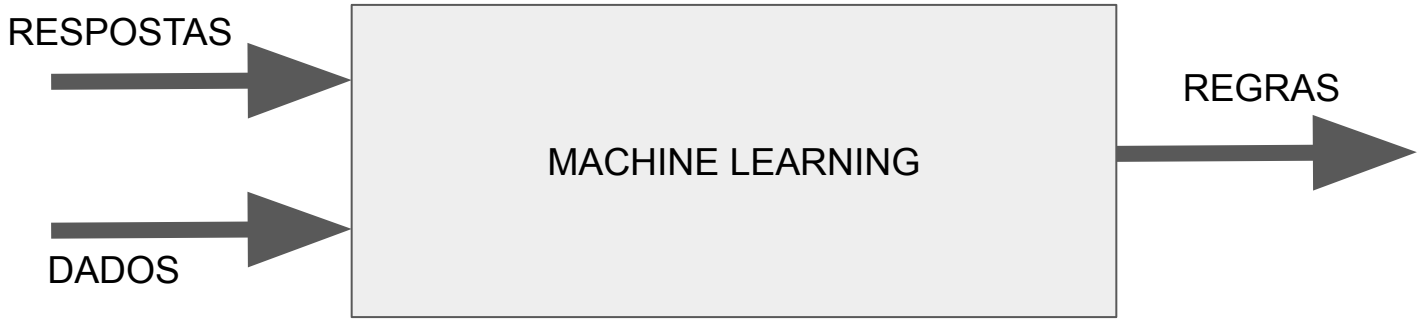


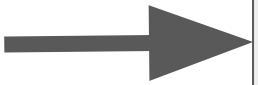
by n7

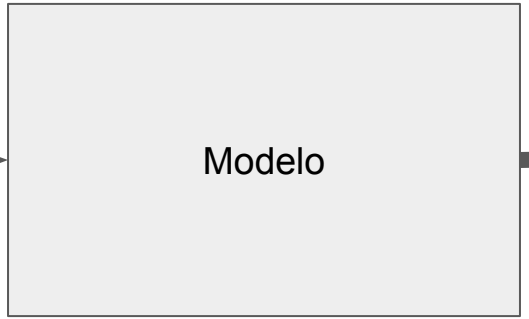
Machine Learning

1. O que é?
2. Quanto eu preciso aprender para conseguir resolver meu problema?
3. Quais recursos estão disponíveis?


REGRAS
DADOS
PROGRAMAÇÃO CLÁSSICA
RESPOSTAS


RESPOSTAS
DADOS
MACHINE LEARNING
REGRAS





Queen

Usuário

Pesquisador

Especialista

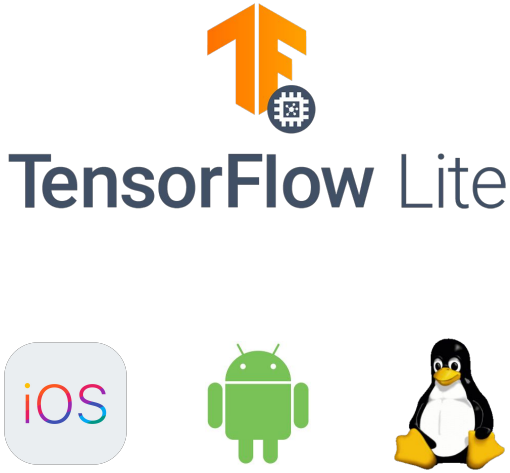
O que é TensorFlow

Uma plataforma de aprendizado de máquina **para todos** usarem e resolver problemas reais.

- Open source
- Free
- end-to-end

O que é TensorFlow Lite?

TensorFlow Lite é um framework para distribuir ML em **dispositivos móves ou sistemas embarcaddos**



Explosão ML on-device

- Menor Latencia
- Sem necessidade de conexão de rede
- Privacidade preservada

On device ML permite uma nova geração de produtos

1000s de apps em producão

Photos

GBoard

Cloud

YouTube

Assistant

Uber

Hike

Mercari

Viber

Airbnb

Text

Classification
Prediction

Speech

Recognition
Text to Speech
Speech to Text

Image

Object detection
Object location
OCR
Gesture recognition
Facial modelling
Segmentation
Clustering
Compression
Super resolution

Audio

Translation
Voice synthesis

Content

Video generation
Text generation
Audio generation

Modelos pre treinados

TensorFlow Lite

tensorflow.org/lite/models

TensorFlow Hub

tfhub.dev

TensorFlow Hub

Search for models, collections & publishers

Send feedback

Filters

Clear all

Problem domain

Model format

TF.js | TFLite | Coral

TF Version

TF1 | TF2

Fine tunable

Architecture

Publisher

Dataset

Language

Audio pitch extraction .JS TFLite

spice

Published by: Google Updated: 08/14/2020

A mobile-compatible pitch extraction model to recognize the dominant pitch in sung audio. Trained (in a self-supervised way) on the MIR-1k dataset.

Image classification TFLite

aiy/vision/classifier/plants_V1

Published by: Google Updated: 08/14/2020

Mobile plants classification model.

MobileNet V2 | iNaturalist

Image object detection TFLite

object_detection...

Published by: Google Updated: 08/14/2020

A class-agnostic mobile object detector.

MobileNet V2

Image classification TFLite

aiy/vision/classifier/birds_V1

Published by: Google Updated: 08/14/2020

Mobile bird classification model.

MobileNet V2 | iNaturalist

Text embedding TFLite

albert_lite_base

Published by: TensorFlow Updated: 08/14/2020

ALBERT is "A Lite" version of BERT with greatly reduced number of parameters. This is a TFLite compatible version. It can be fine-tuned for downstream tasks and converted to TFLite models

Transformer | SQuAD

Image classification TFLite

disease-classification

Published by: AgriPredict Updated: 06/24/2020

Fine-grained image classification model for crop disease classification.

MobileNet | Internal AgriPredict Dataset

Image classification TFLite

aiy/vision/classifier/insects_V1

Published by: Google Updated: 08/14/2020

Mobile insect classification model.

MobileNet V2 | iNaturalist

Image classification TFLite

efficientnet/lite4/classification

Published by: TensorFlow Updated: 08/14/2020

EfficientNet-Lite models, trained on Imagenet (ILSVRC-2012-CLS), optimized for TFLite, designed for performance on mobile CPU, GPU, and EdgeTPU.

EfficientNet-B4 | ImageNet (ILSVRC-2012-CLS)

Image classification TFLite

efficientnet/lite0/classification

Published by: TensorFlow Updated: 08/14/2020

EfficientNet-Lite models, trained on Imagenet (ILSVRC-2012-CLS), optimized for TFLite, designed for performance on mobile CPU, GPU, and EdgeTPU.

EfficientNet-B0 | ImageNet (ILSVRC-2012-CLS)

Image classification TFLite

aiy/vision/classifier/food_V1

Published by: Google Updated: 08/14/2020

Food classification model.

MobileNet V1

Image classification TFLite

cropnet/classifier/cassava_disease_V1

Published by: Google Updated: 08/14/2020

Cassava disease classification model

MobileNet V3

Image classification TFLite

efficientnet/lite2/classification

Published by: TensorFlow Updated: 08/14/2020

EfficientNet-Lite models, trained on Imagenet (ILSVRC-2012-CLS), optimized for TFLite, designed for performance on mobile CPU, GPU, and EdgeTPU.

EfficientNet-B2 | ImageNet (ILSVRC-2012-CLS)

Image classification TFLite

on_device_vision/classifier...

Published by: Google Updated: 08/14/2020

Mobile classification model for popular US products.

Image classification TFLite

object_detection...

Published by: Google Updated: 08/14/2020

A mobile object labeler.

MobileNet V2

Image classification TFLite

on_device_vision/classifier...

Published by: Google Updated: 08/14/2020

Landmark recognition model. Tuned for Asia.

Image classification TFLite

vision/classifier/fungi_mobile_V1

Published by: Danish Mycological Society Updated: 08/14/2020

Mobile-compatible image classification model trained on Fungi dataset provided by the Danish Mycological Society.

MobileNet V2

TensorFlow Lite

developers

TensorFlow Hub

Search

Send feedback

Collection ml-model-binding

Published by: **Android Studio**

Last update: 2020/06/11

Collection of TFLite models that can be used with Android Studio ML Model Binding.

Overview

Use TensorFlow Lite models from TF-Hub with ML Model Binding in Android Studio version 4.1 or later. ML Model Binding makes it easy for you to directly import `.tflite` model files and use them in your projects. Learn more about how to import TFLite models in Android Studio.

Follow these steps to use a model in your Android Studio project:

1. Install **Android Studio 4.1 Beta 1 or later**
2. Download the `.tflite` model file from the model details page. Pick a model with metadata if one is available.
3. In Android Studio, open the TensorFlow Lite model import dialog in the File menu at **File > New > Other > TensorFlow Lite Model**. Select the `.tflite` model file that you downloaded.
4. Click Finish.

This imports the model file into your project and places it in the **ml/** folder. Clicking on the model file in your project will open the model viewer, which provides shows the following:

- **Model:** High-level description of the model
- **Tensors:** Description of input and output tensors
- **Sample code:** Example of how to interface with the model in your app

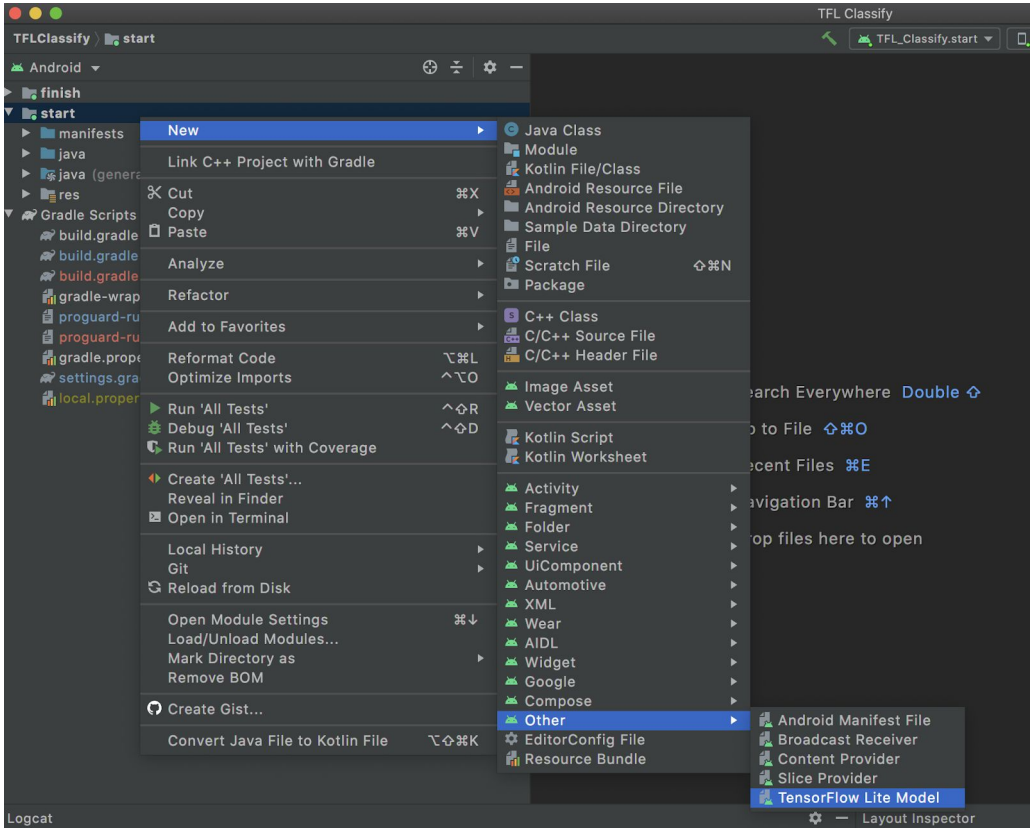
https://tfhub.dev/android-studio/collections/ml-model-binding/1


TensorFlow Lite

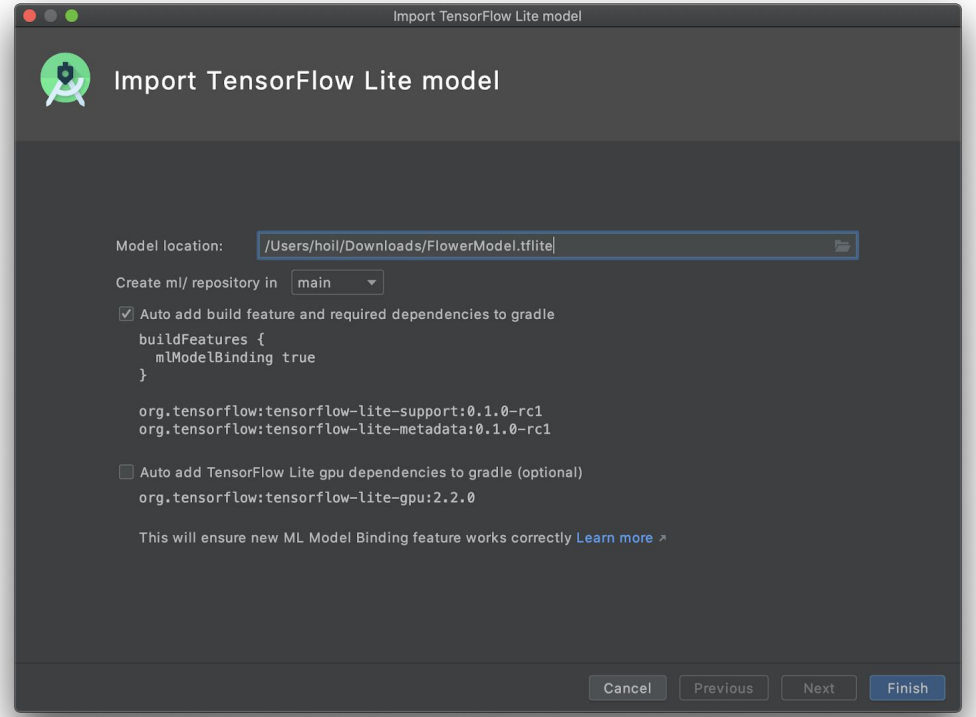

developers

Vamos começar com algo que conhecemos

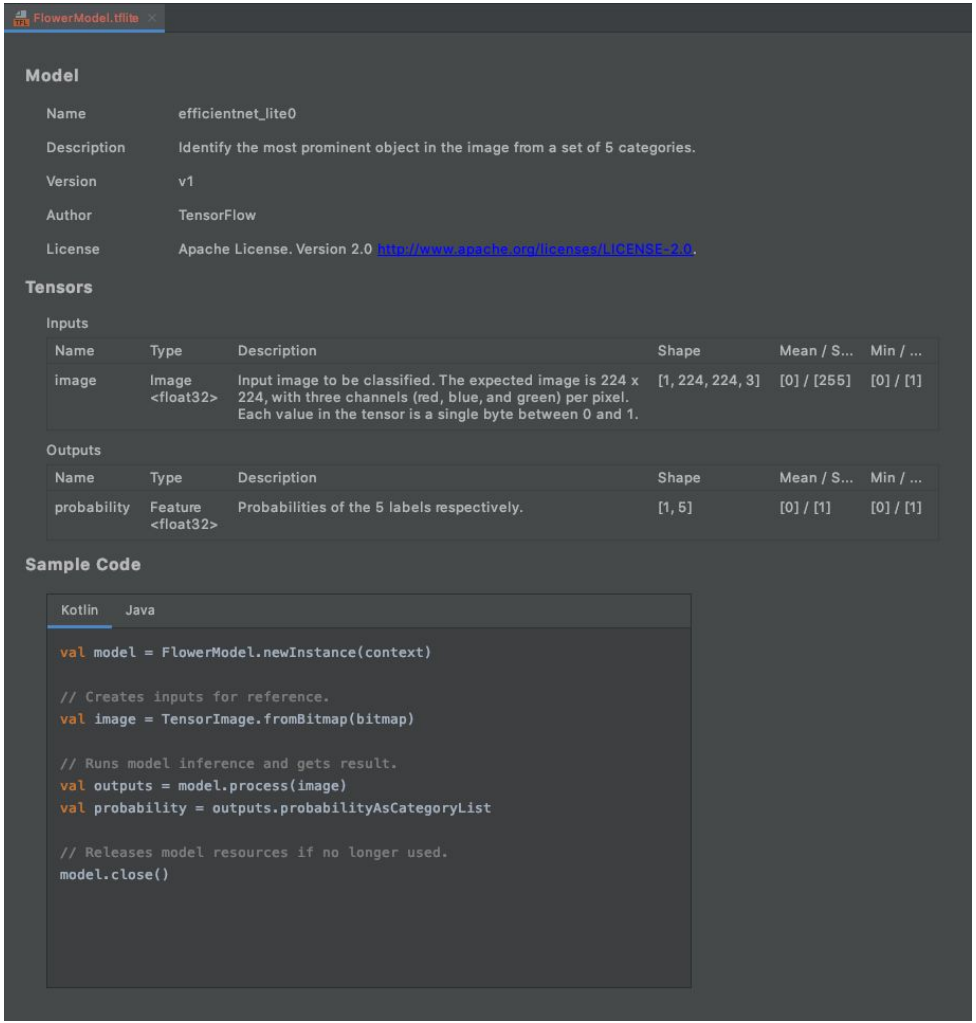
Adicionando um modelo



Adicionando um modelo



Adicionando um modelo



```
val model = MobilenetV210224Quantized1Metadata1.newInstance(context)

// Creates inputs for reference.
val image = TensorImage.fromBitmap(bitmap)

// Runs model inference and gets result.
val outputs = model.process(image)
val probability = outputs.probabilityAsCategoryList

// Releases model resources if no longer used.
model.close()
```

Próximo passo

- Infelizmente o modelo escolhido não sabe o que é um Android :(
- Será que existe algum modelo já pronto?
- Mas se não tem, vou precisar criar um modelo do zero?

Transfer Learning

- Vamos especializar nosso modelo!
- Aprender outra linguagem é mais fácil quando já sabemos uma (Java -> Kotlin)
- Em muitos e muitos casos, não é necessários criar um modelo novo, apenas re-ensinar nosso modelo algumas classes novas.
- Isso vale pra Imagem, Texto, Audio e diversas outras categorias

TFLite Model Maker

Customize os modelos com seu próprio dataset,
Sem muito expertise em ML

TFLite Model Maker é a lib de transfer learning para TFLite

```
# 1. Load data.
data = ImageClassifierDataLoader.from_folder('flower_photos/')

# 2. Customize the model.
model = image_classifier.create(data) # Default model is EfficientNet-Lite0

# 3. Evaluate the model.
loss, accuracy = model.evaluate()

# 4. Export to tflite.
model.export('flower_classifier.tflite')
```

ML com seus dados

Use seus dados

Facil para trocar a arquitetura base

```
# ResNet 50.
model = image_classifier.create(data, resnet_50_spec)

# EfficinetNet-Lite4.
model = image_classifier.create(data, efficientnet_lite4_spec)
```

Model Maker funciona com

Imagem

Classification
(MobileNet,
EfficientNet-Lite,
ResNet...**)**

Object detection*

Texto

Classification
(BERT
ALBERT-Lite*
MobileBERT*)

QA

https://www.tensorflow.org/lite/tutorials/model_maker_image_classification

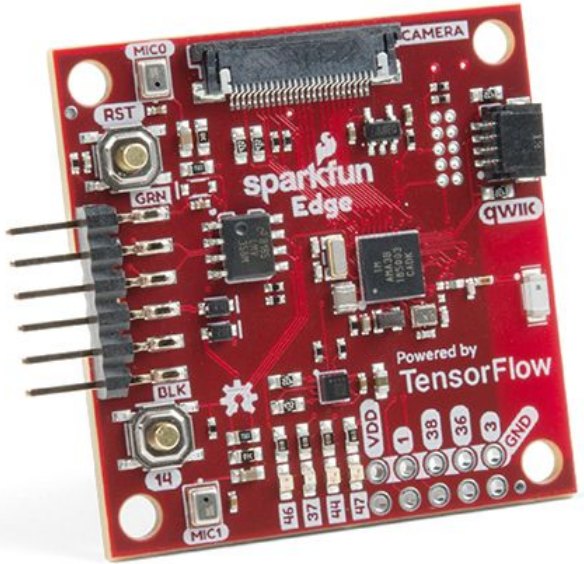
TensorFlow

Otimizações

Fazendo modelos menores e mais rápidos

Considerações

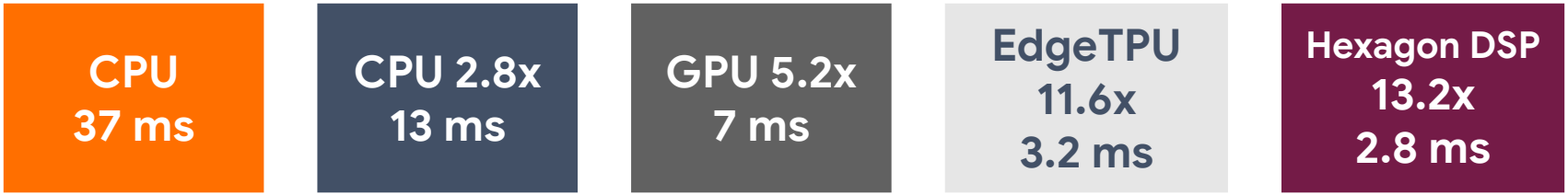
- Processamento limitado
- Memória limitada
- Consumo de bateria
- Tamanho do App



Métodos

1. Utilizar hardware especifico
2. Utilizar arquiteturas optimizadas para mobile
3. Quantização (quantization)
4. Aplicar Pruning ("poda")

Hardware accelerator



Floating point	Quantized Fixed-point	OpenCL Float16	Quantized Fixed-point	Quantized Fixed-point

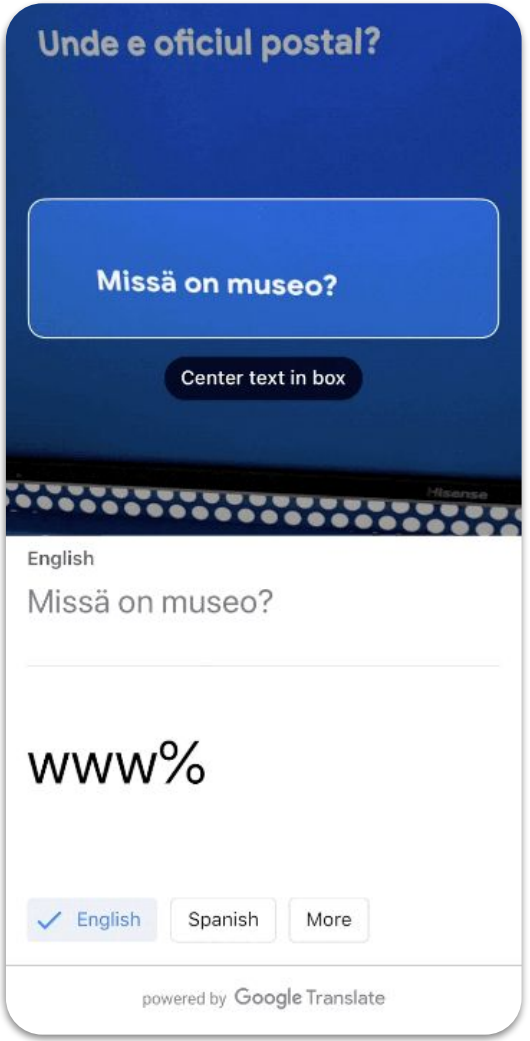
Mobilenet V1-1.0-224

Pixel 4 *- Single Threaded CPU, June 2020*

Outros recursos

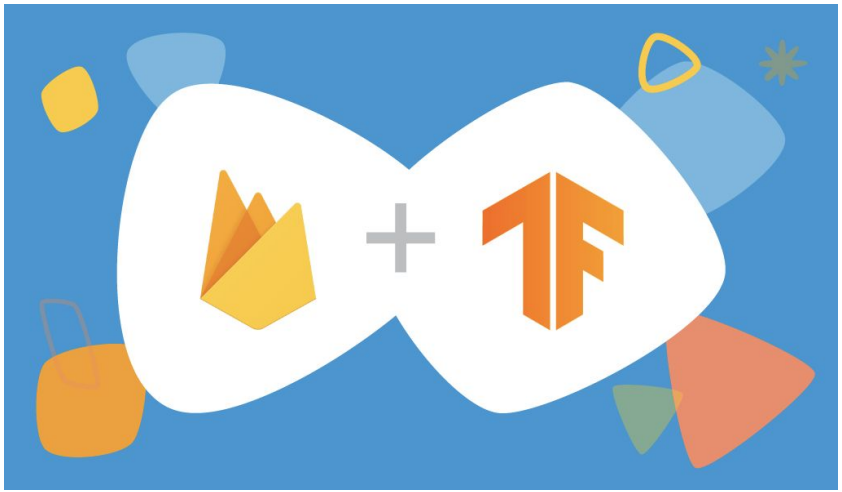
ML Kit

- API simples para casos de uso específicos
  - Image Labeling
  - Text Recognition
  - Object Detection and Tracking
  - Digital Ink
  - Language ID
  - Translation
  - Smart Reply
  - etc



TF Lite + Firebase

1. Atualizar seu modelo on-demand
2. Medir a latência da inferência em produção
3. Teste A/B de versões do seu modelo

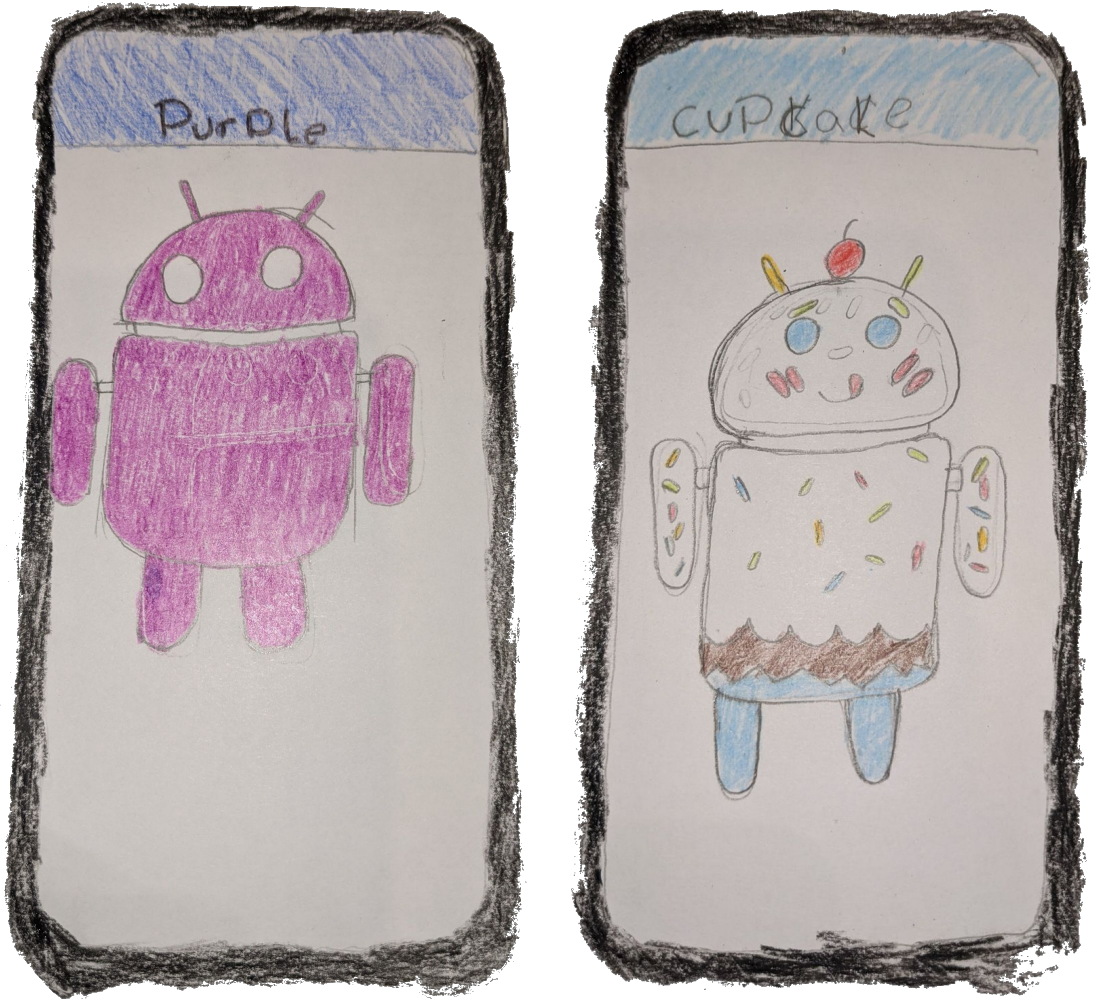


TensorFlow

Resumo

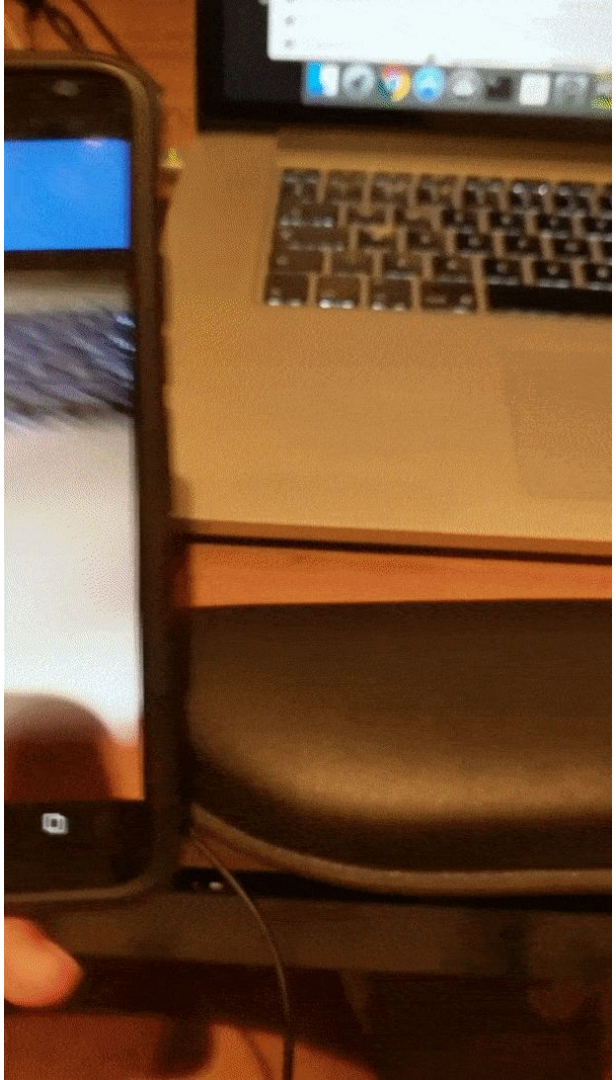
1. Por que on-device ML é importante?
2. Como usar TF Lite & Android Studio plugin
3. TF Hub é sua fonte para modelos
4. Customizando modelos com seus dados com Model Maker
5. Optimizando modelos para dispositivos móveis

Meu Sonho:



by n7

Realidade:



Muito Obrigado

Pra saber mais

https://www.tensorflow.org/lite

Luiz GUStavo

Developer Advocate

@gusthema

Achando os Androids