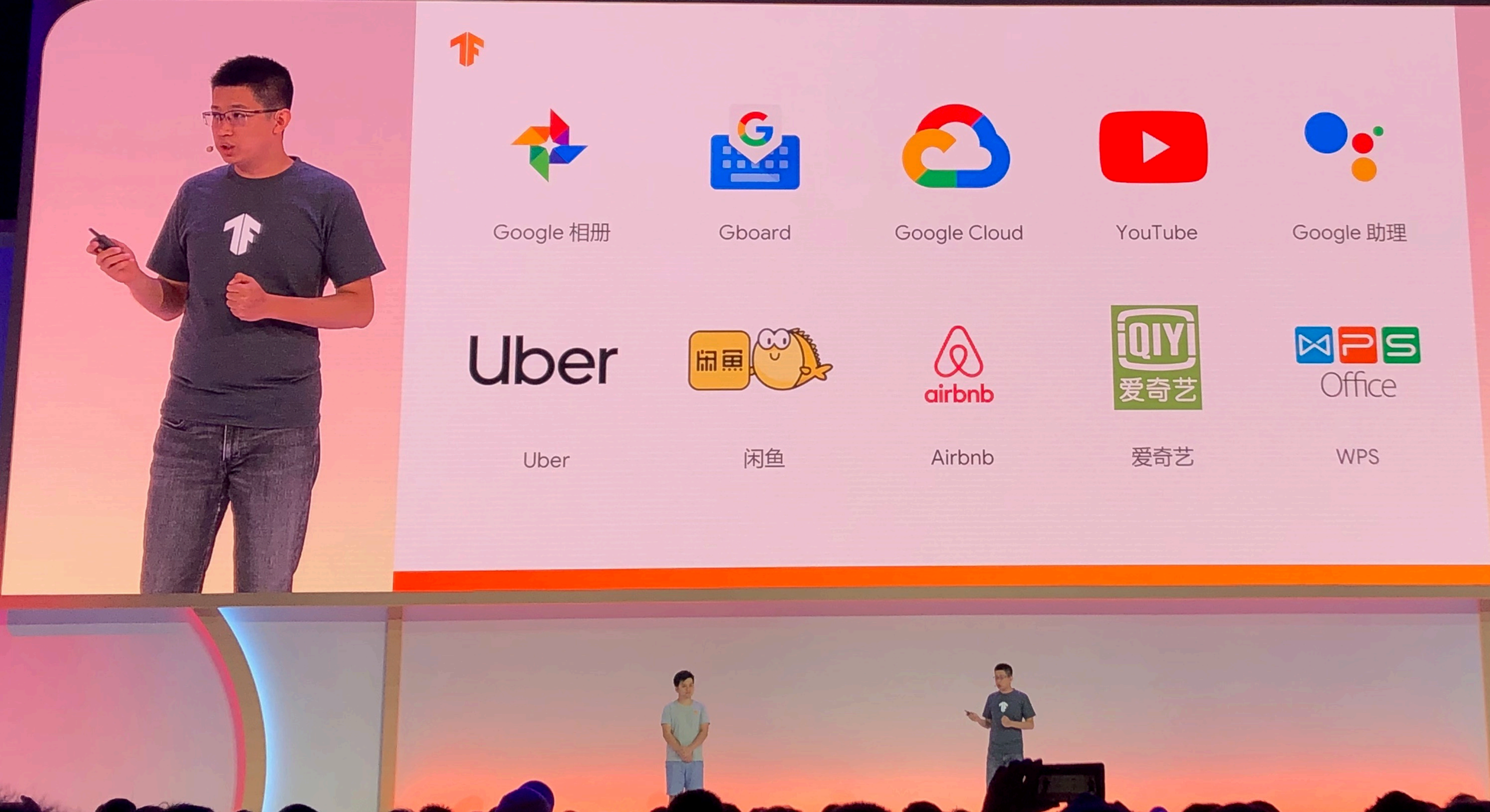
端上机器学习的需求

- 低延迟
- 无需连接网络
- 隐私保护

DanceLike

在不牺牲性能的情况下
并行运行五个任务

- 两个身体部位的图像分割模型
- 匹配分割模型结果并评分
- 播放录制的视频
- 编码视频
- 动态时间矫正 (DTW)

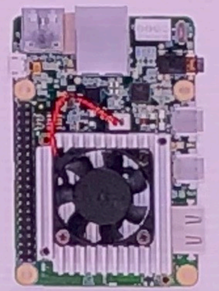

Google 相册
Gboard
Google Cloud
YouTube
Google 助理
Uber
闲鱼
airbnb
爱奇艺
WPS Office
Uber
闲鱼
Airbnb
爱奇艺
WPS

技术挑战

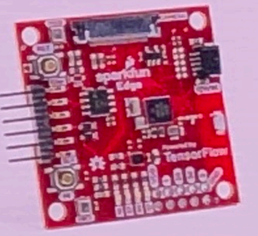
- 算力少
- 内存小
- 少耗电

安卓和 iOS

嵌入式系统
(比如 Raspberry Pi)



硬件加速器
(比如 Edge TPU)



微控制器 (MCU)

TensorFlow Lite

tensorflow.google.cn/lite/models

TensorFlow Lite 示例应用

使用 TensorFlow Lite 的一系列 Android 和 iOS 应用。

图像分类

使用预训练模型测试图像分类解决方案。该模型可识别移动设备摄像头输入帧中 1000 个不同类型的对象。

探索 Android 应用
探索 iOS 应用

对象检测

探索使用预训练模型的应用。该模型会在移动设备摄像头输入帧中 1000 个不同的可识别对象周围绘制边界框并添加标签。

探索 Android 应用
探索 iOS 应用

姿态估计

Explore an app that estimates poses of people in an image.

探索 Android 应用

语音识别

探索使用麦克风检测关键字并返回讲话人说出该关键字的概率得分的应用。

探索 Android 应用
探索 iOS 应用

手势识别

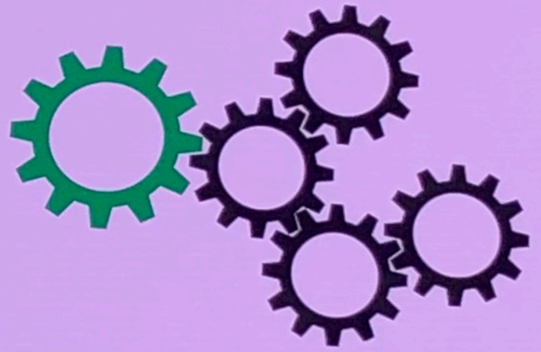
使用 TensorFlow.js 训练神经网络识别摄像头拍摄到的手势，然后使用 TensorFlow Lite 将模型转化为在设备上运行推断。

探索 Android 应用
探索 iOS 应用

敬请期待更多示例应用，如 BERT，风格迁移，语音唤醒等等

三部曲

1. 训练 TensorFlow 模型
2. 转换到 TensorFlow Lite 格式
3. 部署模型到端侧设备



创建并转换你的模型

转换

TensorFlow → Saved Model

```
x, y = np.ones((10, 224, 224, 3,)), np.ones((10, 1,))  # Training
data

model = tf.keras.Sequential([
    tf.keras.layers.InputLayer((224, 224, 3,),
name="input_image"),
    tf.keras.layers.Conv2D(128, 1, name="conv_2d"),
    tf.keras.layers.GlobalMaxPooling2D(name="max_pool"),
    tf.keras.layers.Dense(1, activation="sigmoid", name="prob"),
])
model.compile("adam", "binary_crossentropy")

model.fit(x, y, batch_size=2, epochs=1)
tf.keras.models.save_model(model, saved_model_dir)
```

部署以及使用

部署

载入模型 → 数据预处理 → 模型推理 → 使用结果

```
/** Initializes an {@code ImageClassifier}. */
ImageClassifier(Activity activity) throws IOException {
  tfliteModel = loadModelFile(activity);
  tflite = new Interpreter(tfliteModel, tfliteOptions);

  imgData =
      ByteBuffer.allocateDirect(
          DIM_BATCH_SIZE
              * getImageSizeX()
              * getImageSizeY()
              * DIM_PIXEL_SIZE
              * getNumBytesPerChannel());
  imgData.order(ByteOrder.nativeOrder());
}
```

部署

TensorFlow Lite

转换器
(转换成TF Lite格式)

解释器

算子内核

delegate

为 ARM Neon 指令集高度优化

集成硬件加速器如 GPU, DSP 和 Edge TPU

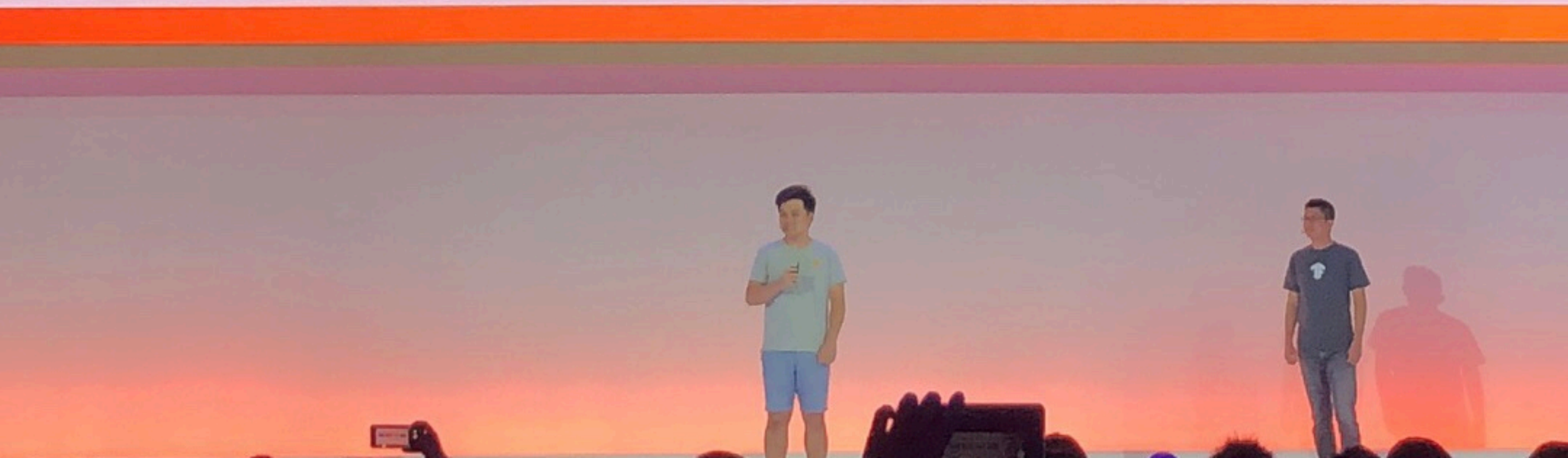
和 Android Neural Network API 集成

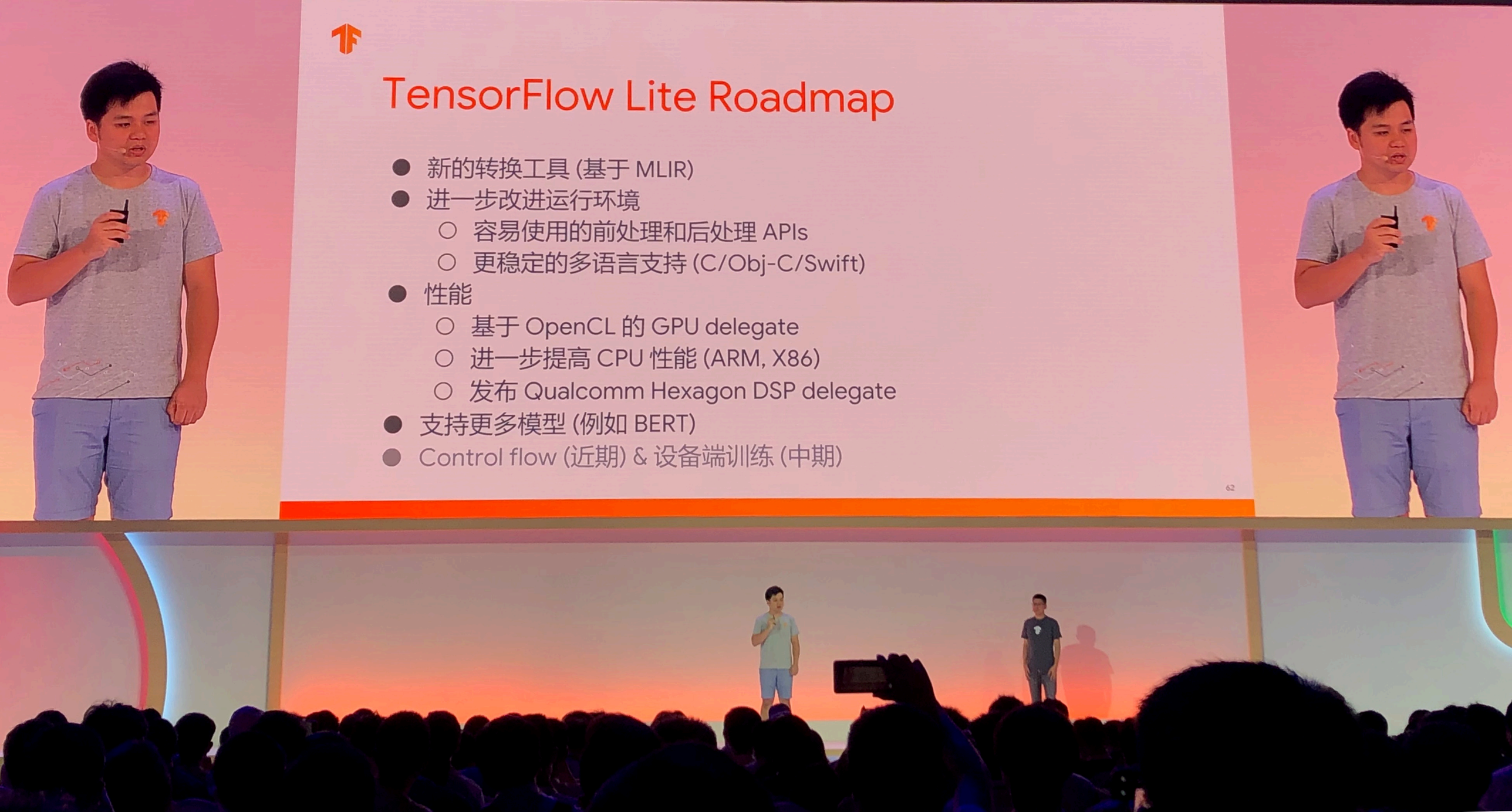
使用硬件加速器

CPU	GPU 5x	DSP 15.2x	EdgeTPU 38x
76 ms	15 ms	5 ms	2 ms
CPU on MobileNet V1	GPU OpenGL Float16	Quantized Fixed-point	Quantized Fixed-point

MobileNet V1

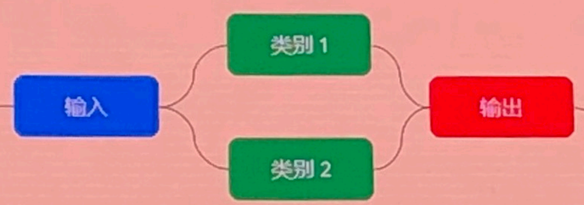
***el 3** - Single Threaded Kyro CPU, Adreno GPU & Hexagon DSP*




TensorFlow Lite Roadmap
新的转换工具 (基于 MLIR)
进一步改进运行环境
容易使用的前处理和后处理 APIs
更稳定的多语言支持 (C/Obj-C/Swift)
性能
基于 OpenCL 的 GPU delegate
进一步提高 CPU 性能 (ARM, X86)
发布 Qualcomm Hexagon DSP delegate
支持更多模型 (例如 BERT)
Control flow (近期) & 设备端训练 (中期)

MCU

有声音吗?



MCU

是人类的声音吗?

复杂网络

应用处理器 (AP)

tensorflow.google.cn/lite/microcontrollers/overview

应用场景

- 语音唤醒
- 图像分类
- 运动检测
- 智能传感器
- 智能门锁