开源深度赋能小米AIoT

小米集团副总裁 | 崔宝秋

小米集团首席语音科学家 | Daniel Povey

小米的核心战略

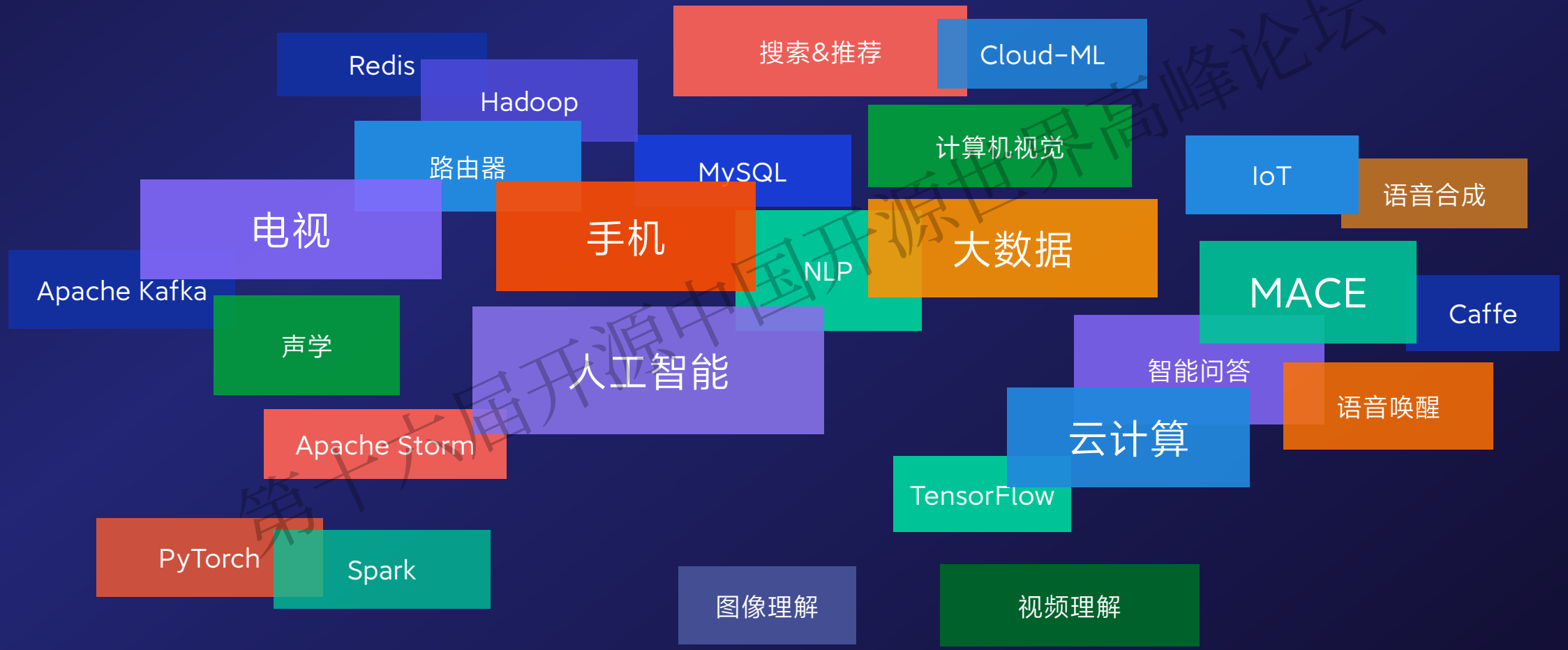
手机 × AIoT


第十六届开源中国开源世界高峰论坛

全球领先的消费级AIoT平台

IoT平台已连接设备数超过3.51亿台

拥有5个及以上小米IoT设备的用户680万



小米移动端深度学习框架 MACE

2018年6月28日在“第13届开源中国开源世界”大会上正式开源

MACE 重要的里程碑

边缘侧推理框架 MACE Micro

支持框架	Tensorflow	Caffe	MegEngine	ONNX(Pytorch / Kaldi)		
适配工具	Tensorflow / Caffe / ONNX模型转换 / 量化 / 压缩工具					
AI引擎	MACE AI Engine				Micro AI Engine	
AI算子	Neon Kernel	OpenCL Kernel	Hexagon NN	MTK NN	MCU Kernel	
异构芯片	CPU	GPU	DSP/HTA	APU/VPU	MCU	DSP

MACE Micro 研发目标

好移植

在芯片架构、操作系统/文件系统、编译工具链/C++库等方面，不同IoT芯片差异较大

不使用堆内存分配
不依赖任何OS
不依赖C++库
不依赖文件系统
除Math库之外不依赖第三方库
C++98标准

高性能

IoT芯片大多算力低，而深度学习模型层数较多，消耗算力较多

模型优化
访存优化
初始化前置
指令优化

低功耗

IoT设备大多数需24小时不间断运行，电源为电池供电。

访存优化
BFloat16支持
不依赖文件系统

易使用

框架会被公司内部使用并开源，提高开发者的开发效率。

单元测试
基准测试
自动化工具
模型保护
Bazel编译支持

MACE Micro 的潜在落地场景

健康监测

耳机降噪

语音唤醒

行为识别

儿童玩具

MACE应用案例：NLP

KALDI

ONNX

MACE应用案例：场景识别

MACE应用案例:
人脸关键点检测/相机美颜

MACE应用案例：单摄背景虚化

MACE应用案例：魔法换天

MACE应用案例：超级夜景

MACE应用案例：图像超分辨率

MACE应用案例：文档检测/矫正

MACE应用案例：文档增强

长达一生的青春

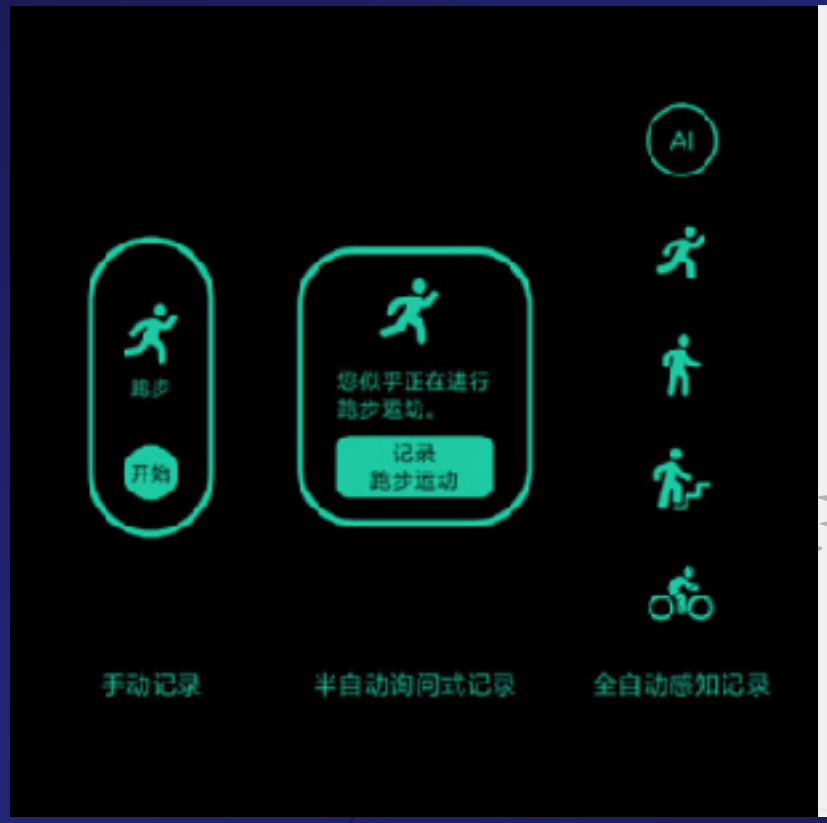
长达一生的青春

MACE应用案例：AI大片

AI大片算法的功能在于通过AI模型自适应地为不同场景的图片生成高级大片视感的效果。

如上图所示，左图为原始图像，中图和右图为大片算法处理后得到的两种不同的效果，经算法处理后，整体增加了图片画质。

业务落地：手机AI运动行为感知

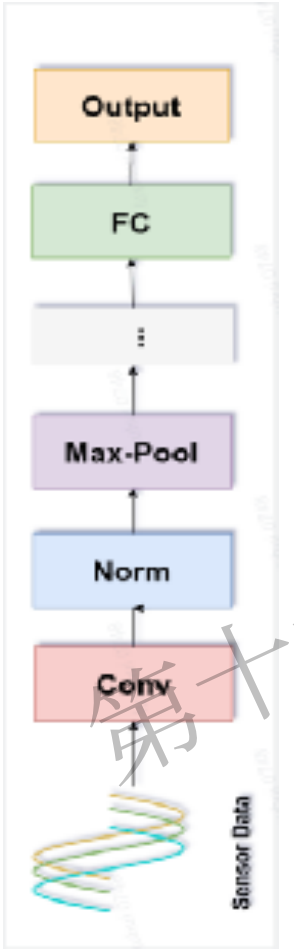


灵弦算法特点

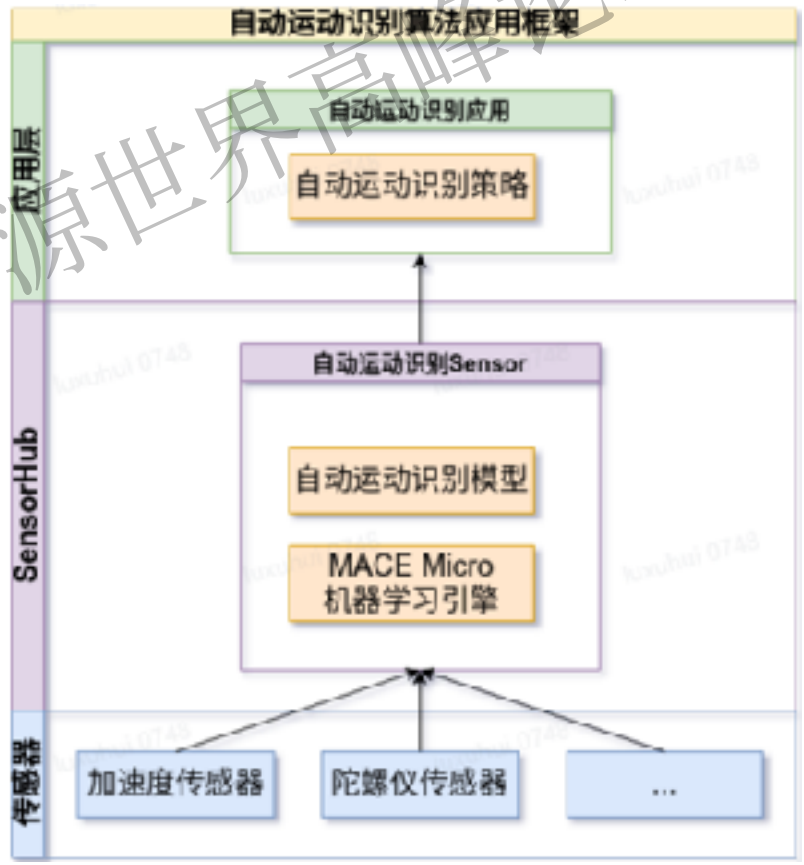
- 国内第一家支持多种运动行为感知技术的手机厂商
- 准确率为94.3%
- 召回率80.9%
- 准确率与召回率超越Google的表现

24小时耗电量不超过手机电量的1%

业务落地：可穿戴设备AI运动行为感知



小米手表
SensorHub芯片：高通
QCC1110



Xiaomi Vela

Xiaomi Vela

基于开源实时操作系统NuttX打造的嵌入式物联网软件平台

智能手环	运动手表	智能音箱	智能家居	智能传感器	相机ISP
					

Xiaomi Vela：打通碎片化的IoT应用，支持高性价比的MCU设备，为IoT的繁荣构建基础设施

Xiaomi Vela满足小米自身的业务需求

小米IoT模组

小米每年数千万的IoT模组出货量，需要适配不同vendor、不同芯片和不同RTOS，“碎片化”严重

智能音箱

小爱音箱采用Vela可以运行在低主频低成本的MCU上

手环和运动手表

手环、运动手表和TWS耳机都对Vela有强需求

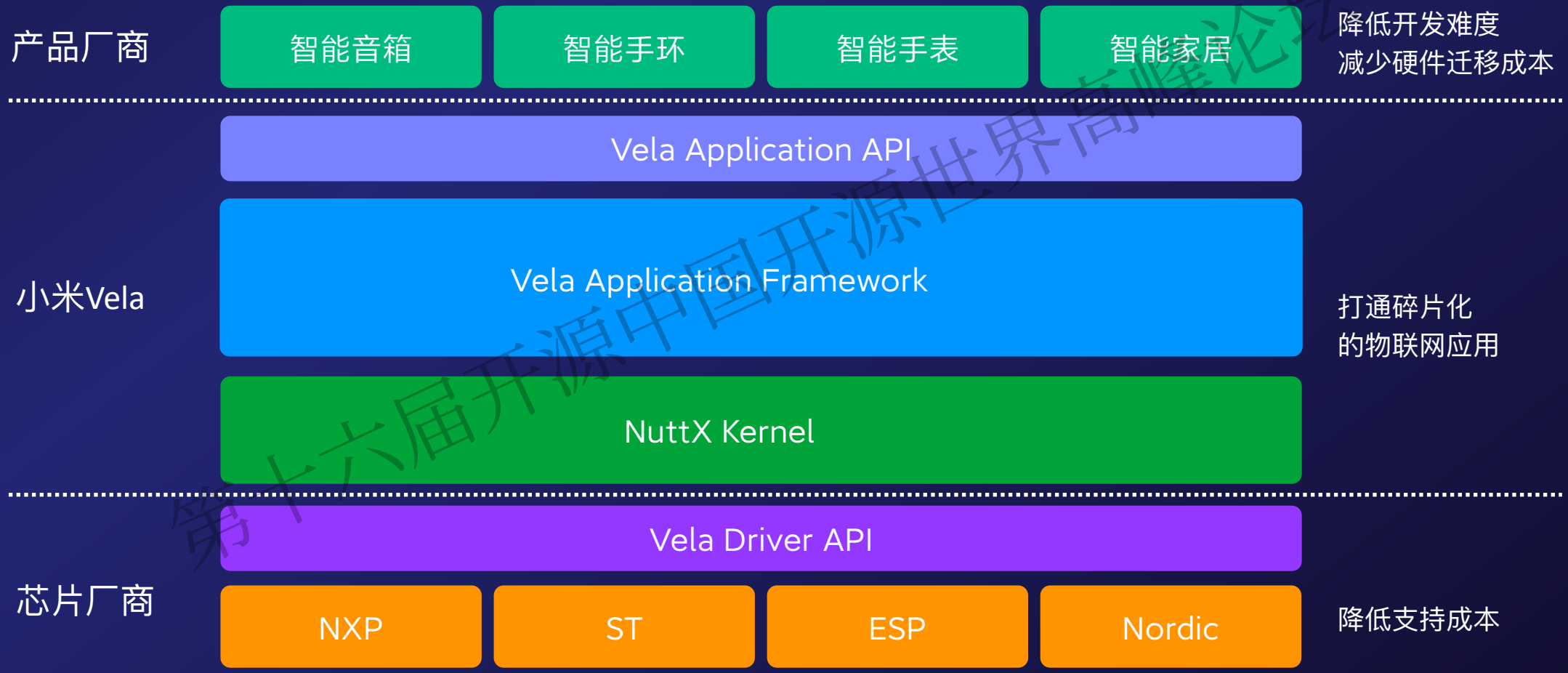
统一的软件平台带来更好的互联互通体验

Xiaomi Vela的主要技术优势

- 开源的基因
- 高度可裁剪
- 和Linux兼容性好，代码易复用
- 完整度高
- 背靠小米强大的IoT生态

过去20年，虽然很多公司都尝试过推物联网操作系统，但目前市场并没有出现一个绝对的统治者。

Xiaomi Vela成为上下游厂商间的桥梁

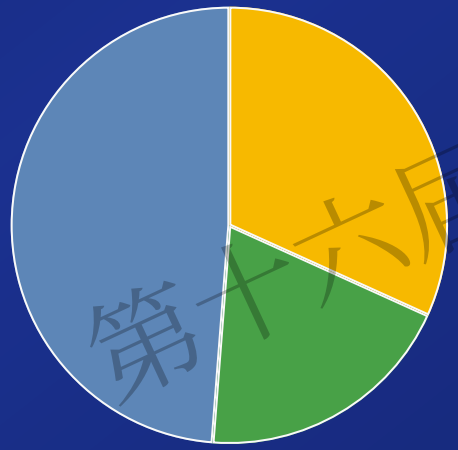


小米积极推动NuttX开源社区发展

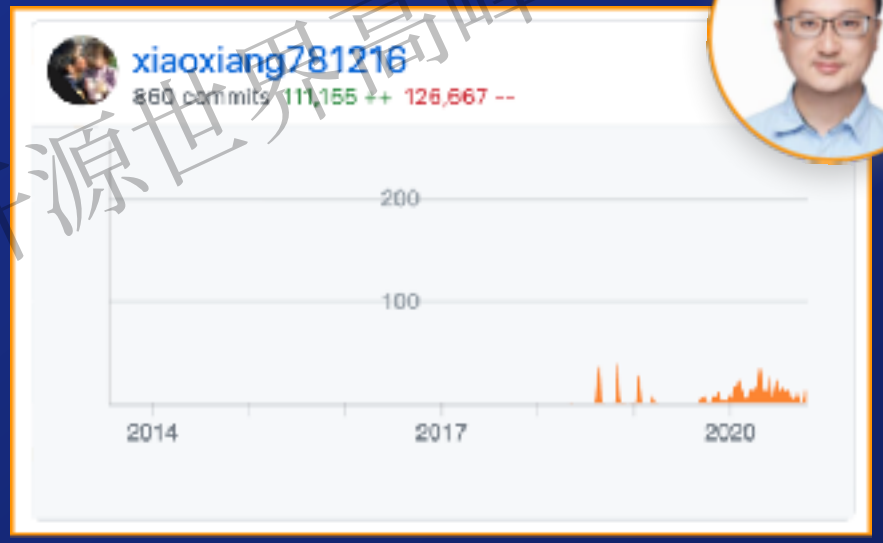
https://nuttx.apache.org

小米Vela团队今年贡献了
约1/3的patch到NuttX社区

各公司在NuttX项目patch数量（2019/12 ~ 2020/11）



- 小米(1138 patches)
- 索尼(697 patches)
- 其他：NXP、乐鑫、PX4社区…(1747 patches)



KALDI

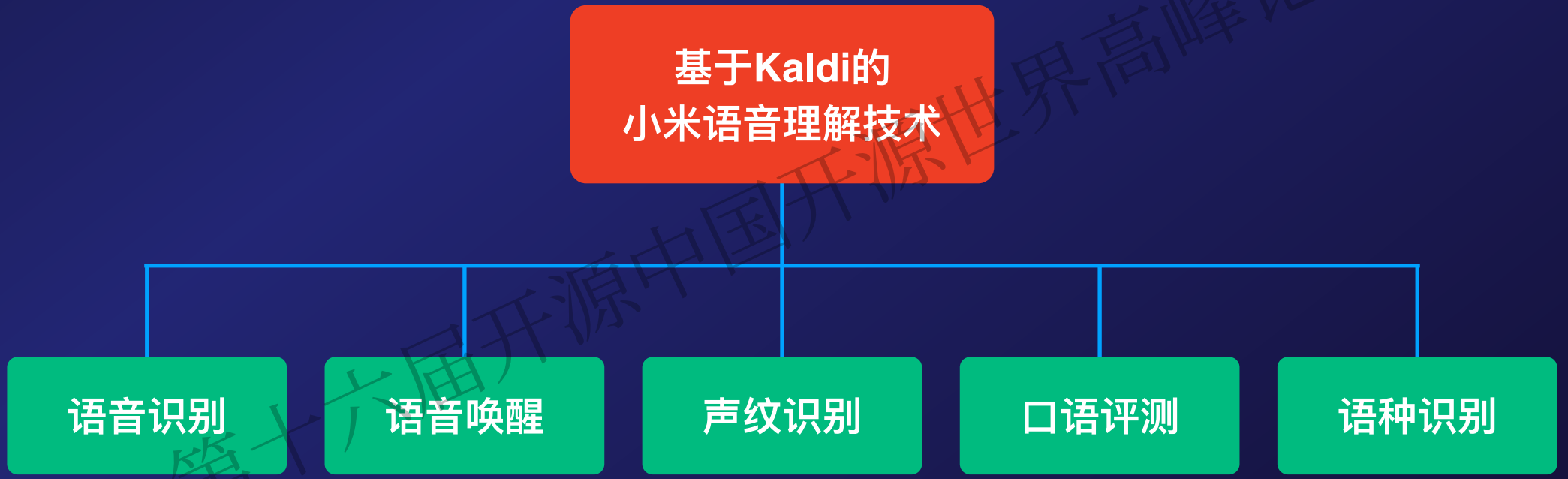
语音识别

Kaldi是目前全球最流行的语音识别开源工具集。

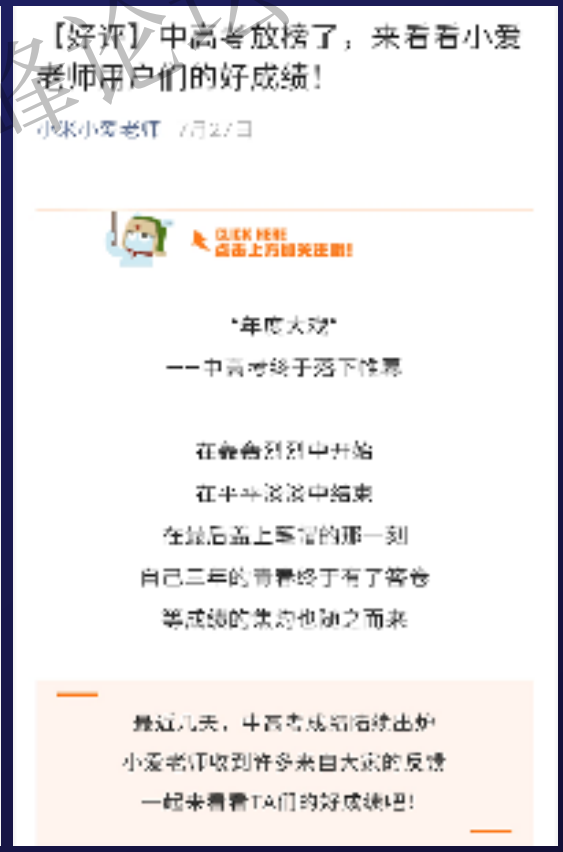
Kaldi在学术界降低了语音人才的入门门槛，为各大学术研究和挑战赛提供基线系统。初创公司和团队纷纷使用Kaldi结合自己的数据迅速验证业务并为用户服务。

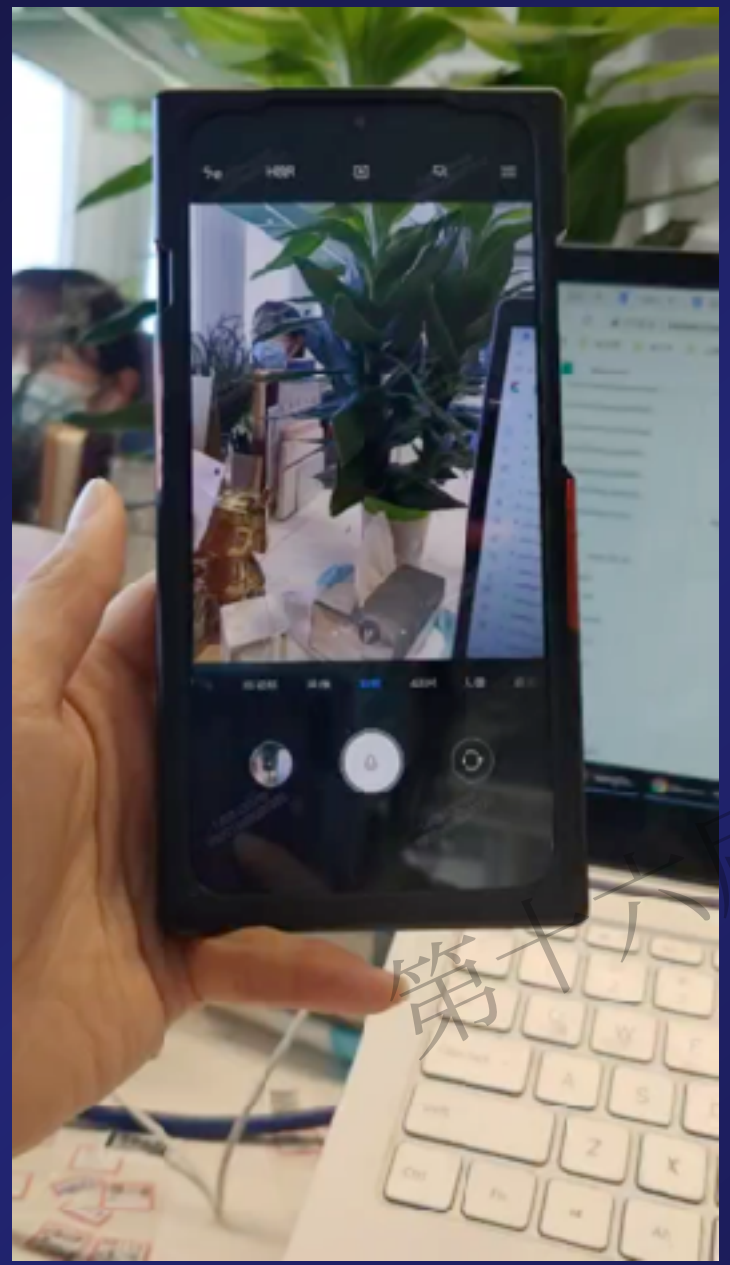
几乎所有做语音识别的机构和企业都在使用Kaldi。

Kaldi助力小米AIoT语音理解



离在线语音识别





语音唤醒

长语音理解 =

语音识别

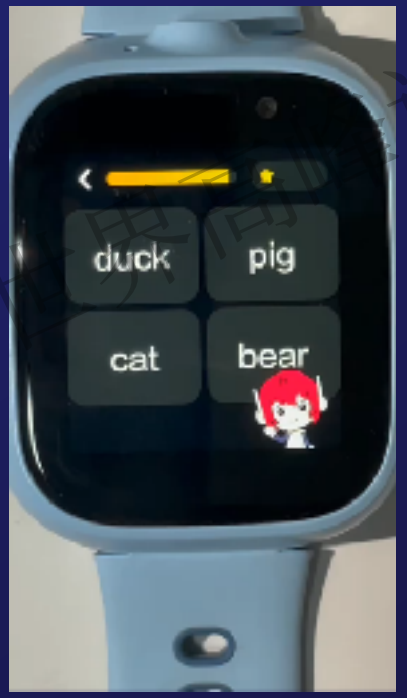
+ 说话人分割

+ 中英语种识别

“聆听”

小米第一届黑客马拉松冠军项目

口语评测

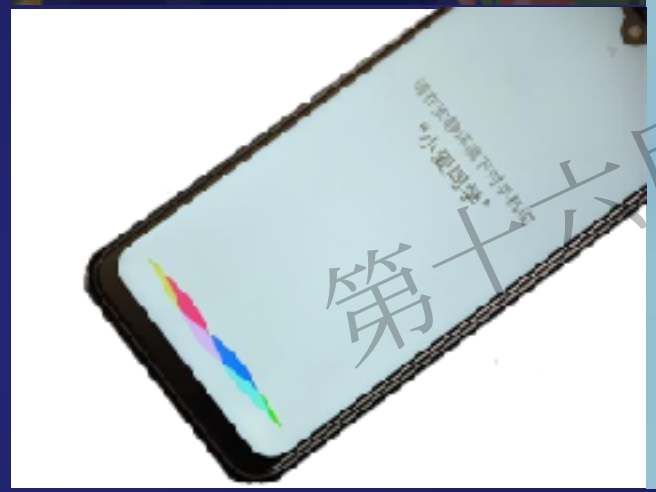


Kaldi开源脚本与口语评测数据库

Kaldi脚本链接： https://github.com/kaldi-asr/kaldi/tree/master/egs/gop_speechocean762

开源数据库： Zhang J, Zhang Z, Wang Y, Yan Z, Song Q, Huang Y, Li K, Povey D and Wang Y. speechocean762: An Open-Source Non-native English Speech Corpus For Pronunciation Assessment, INTERSPEECH 2021.

端到端声纹识别



- 语音人群划分
- 电视儿童锁
- 手机音箱声纹锁
- 语音画像追剧

依托Kaldi取得行业比赛好成绩

2021: 个性化语音唤醒挑战赛
双赛道冠军（1）

2019: 远场声纹识别挑战赛
双赛道冠军（1）

2021: 儿童语音识别挑战赛冠军

2021: 个性化语音唤醒挑战赛
双赛道冠军（2）

2019: 远场声纹识别挑战赛
双赛道冠军（2）

2020: 东方语种识别竞赛第三名


开源是人类技术进步的
最佳平台和模式

下一代Kaldi

Kaldi

- Created in 2011
- Quickly became the standard speech recognition toolkit
- Widely used in industry and academia
- Does speech recognition and other speech-related tasks
  - E.g. speaker identification

- Kaldi诞生于2011年，随后迅速成为语音识别领域的标准工具包
- 广泛用于工业界和学术界
- 适用于语音识别和说话人识别等语音相关任务

Kaldi to Next-gen Kaldi

Kaldi

- Convenient for production code
- C++-based
- Not easy to keep up with the latest machine learning trends
- Not as flexible as we would like.
- 适用于生产环境
- 基于C++代码
- 不方便同步最新的机器学习算法进展，缺乏灵活性

Next-gen Kaldi

- Compatible with PyTorch (easier to use recent model architectures like Transformer)
- Mostly Python-based
- More modular (separate codebases for different parts)
- 新一代Kaldi兼容PyTorch（从而可以十分便捷的使用最新的模型结构，比如Transformer）
- 大部分代码基于Python
- 更加模块化

Next-gen Kaldi structure

Recipes
（示例脚本集合部分）

Training data preparation
（数据准备部分）

Core algorithms
（核心算法部分）

- lhotse does data preparation; suitable for many speech processing tasks.
- k2 contains all core algorithms written in C++ (and CUDA code) , it’s the most interesting piece. It can even be used to do other sequence learning tasks, e.g. hand writing.
- icefall is the recipes (the example scripts), it is based on lhotse and k2.

k2: timeline

https://github.com/k2-fsa/k2

April 2020
k2 was created.

July 2020
CPU version was ready.

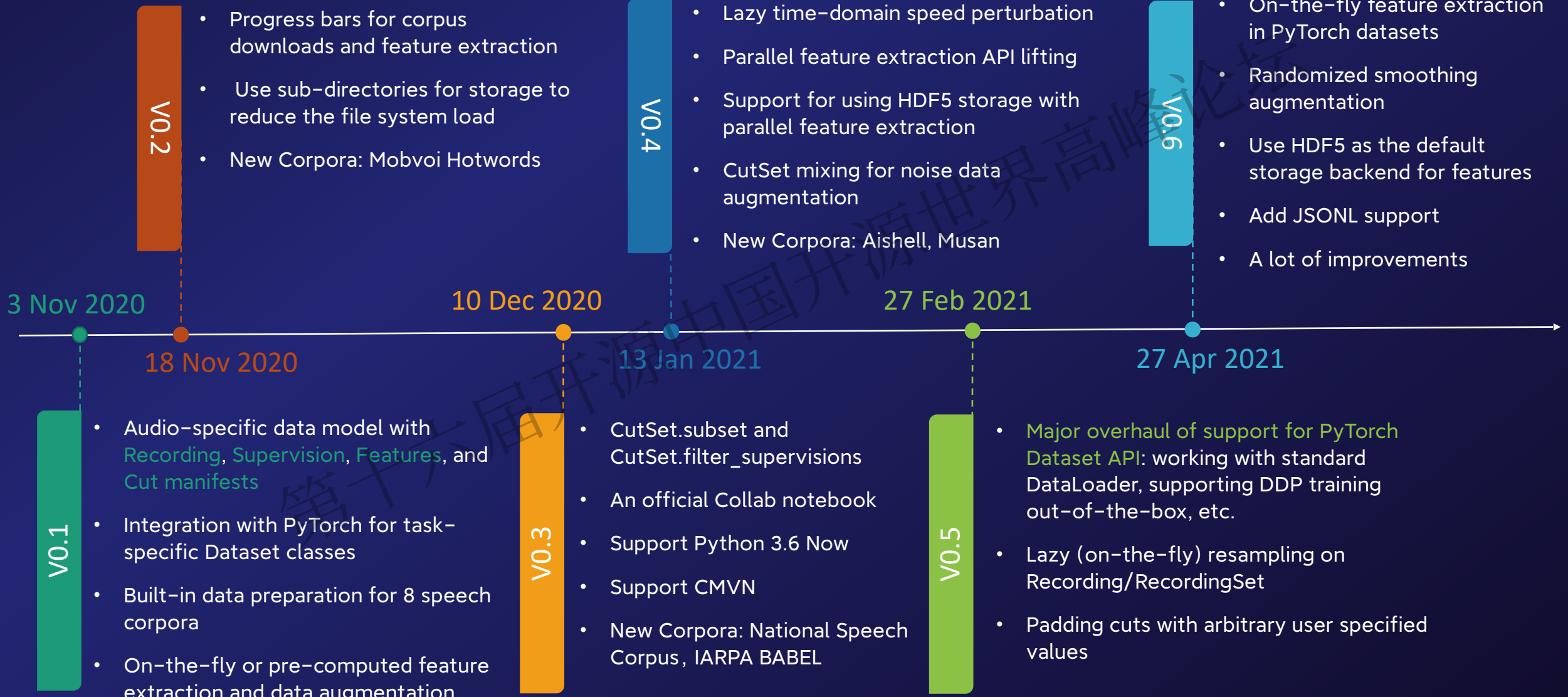
Nov 2020
CUDA version was ready.
k2 0.1 was released.

Dec 2020
The first ASR model was trained successfully with k2.

Until Now
- New algorithms
- Memory and speed optimization
- New platforms and installation methods (mac, Windows, Conda)
- k2 1.0 released today!

(k2 1.0于今日正式发布)

lhotse: timeline



icefall (snowfall) : timeline

- snowfall is draft version of icefall
  snowfall是icefall的草案项目

https://github.com/k2-fsa/snowfall

- Word Error Rate is measured on LibriSpeech test-clean.
- Noted we trained models on LibriSpeech 100h before May 2021 as we want to try different (new) ideas with k2 very fast that time.
- Still have catching up to do in WER.

Nov 2020
snowfall was created; trained the first model successfully based on lhotse and k2

Dec 2020
CTC(TDNN), WER ~15%

Jan 2021
MMI (TDNN) WER ~15%

Feb 2021
CTC (TDNN-LSTM) WER ~13%
MMI (TDNN-LSTM) WER ~11%
CTC (Transformer) WER ~8.5%

Mar 2021
MMI (Transformer) WER ~8%
MMI (Conformer) WER ~7%
CTC (Conformer) WER ~7%

Apr 2021
DDP training (Conformer, using pre-trained CTC alignment model) WER ~5.8%

May 2021
MMI (Conformer + VGG, model average) + 4-gram rescoring WER ~3.7% (trained on LibriSpeech 960h)

Ongoing work

- The most unfinished pieces is snowfall/icefall
- Since we started work, SpeechBrain was released
- Need to figure out whether it makes sense to integrate with pieces of it
- Targeting September for first version of icefall

- 预期在九月份正式发布icefall
- 有可能会与SpeechBrain（部分）集成

谢谢！