AN0148
应用笔记

AT32上实现关键词语音识别（KWS）

前言

关键词识别（Keyword Spotting，KWS）属于语音识别领域的一个子领域，用户在智能设备上进行语音交互时比较常用到该技术。
2018 年 ARM 和斯坦福大学进行了合作，并开源了预训练 TensorFlow 模型及其语音关键词识别代码。本文基于此开源模型和代码，在 AT32 MCU 上对 KWS 效果进行展示。

识别效果视频链接　　https://b23.tv/3UNwWEH

参考资料：

- *ML-KWS-for-MCU*
  https://github.com/ARM-software/ML-KWS-for-MCU
- *Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition*
  https://arxiv.org/pdf/1804.03209.pdf
- *Hello Edge: Keyword Spotting on Microcontroller*
  https://arxiv.org/pdf/1711.07128.pdf

支持型号列表：

支持型号	AT32F403A

目录

表目录

图目录

1 KWS 概述

关键字定位（Keyword Spotting，KWS）技术，已成为可穿戴设备、物联网设备和其他智能终端的关键。诸如“Alexa”，“Hey Siri”或“Ok Google”等短语唤醒智能手机和家用电器上的语音激活功能，已经是语音交互设计产品的广泛需求。

对于 KWS，实时响应和高精度才能获得良好的用户体验。最近，神经网络已成为 KWS 架构的一个有吸引力的选择，因为与传统的语音处理算法相比，它们具有更高的准确性。
由于需要实时在线识别的要求，导致 KWS 应用在内存和计算能力有限的微型微控制器上运行会受到一定限制。KWS 的神经网络架构设计必须考虑这些限制。于是，研究人员设计出由于传统 CNN 的深度可分离卷积神经网络（DS-CNN）架构技术。

为了进一步介绍了 DS-CNN 架构，并展示了开发人员如何在 MCU 上实现 DS-CNN KWS。2018 年 ARM 和斯坦福大学进行了合作，并开源了预训练 TensorFlow 模型及其语音关键词识别代码，并将结果发表在论文 *Hello Edge: Keyword Spotting on Microcontrollers* 中。

本文基于此开源模型和代码，在 AT32 MCU 上对 KWS 效果进行展示。

2 KWS 实现原理

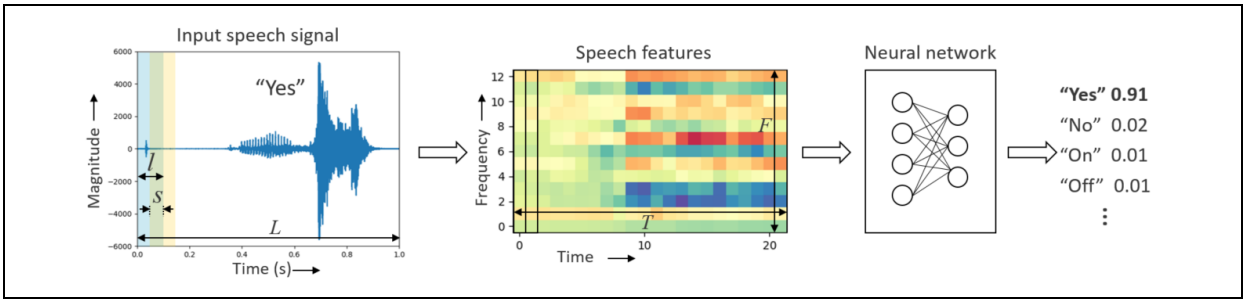
2.1 关键词识别 KWS

一个典型的 KWS 系统由一个特征提取器和一个基于神经网络的分类器组成，如下图所示。首先，长度为 L 的输入语音信号被分成长度为 l 且步幅为 s 的重叠帧，总共有帧 T 帧。

$$T = \frac{L - l}{S} + 1$$

从每一帧中提取 F 个语音特征，则长度为 L 的整个输入语音信号总共生成 T × F 个特征。Log-mel filter bank energies (LFBE) 和 Mel-frequency cepstral coefficients (MFCC) 常用于基于深度学习的语音识别，特别适用于传统语音处理技术。使用 LFBE 或 MFCC 进行特征提取涉及将时域语音信号转换为一组频域频谱信号，从而实现输入信号的维度压缩。提取的语音特征矩阵服务于输入分类器模块，该模块导出所输出分类的概率。在需要从连续音频流中识别关键字的实际场景中，利用后端处理模块可以在一段时间内平均每个输出类的输出概率，从而提高预测的整体置信度。

图 1. KWS 数据管道



2.2 卷积神经网络 CNN

基于 DNN 的 KWS 的一个主要缺点是它无法有效地对语音特征中的局部时间和频谱相关性进行建模。 CNN 是通过将输入时域和谱域特征视作图像，并对其进行二维卷积处理。卷积层之后通常是批量归一化 、基于 ReLU 的激活函数和可选的最大/平均池化层，这些处理可以降低特征的维数。在推理过程中，批量归一化的参数可以折叠到卷积层的权重中。在某些情况下，为了减少参数和加速训练，在卷积层和密集层之间添加了一个线性低秩层，这是一个没有非线性激活的全连接层

2.3 深度可分离卷积神经网络 DS-CNN

深度可分离卷积神经网络（DS-CNN）。最近，深度可分离卷积已被提出作为标准 3-D 卷积操作的有效替代方案，并已用于在计算机视觉领域实现紧凑的网络架构。 DS-CNN 首先将输入特征图中的每个通道与一个单独的 2-D 滤波器进行卷积，然后使用逐点卷积（即 1x1）在深度维度上组合输出。通过将标准的 3-D 卷积分解为 2-D 卷积，然后是 1-D 卷积，深度可分离卷积在参数数量和操作方面都更加高效，这使得即使在资源受限的微控制器设备中也可以实现更深、更宽的架构。

3 例 KWS 实作

3.1 KWS 测试平台

KWS 系统需要使用到两个平台，即 PC 端和 AT32 MCU 端。

PC 端：

利用 TensorFlow 与 Python 撰写完整的深度学习程序代码并训练模型，因本文件使用的学习模式为监督式的学习，需给系统大量的训练数据和 Labels，接着将提取到的特征用以训练 CNN 模型，并反复修正训练的模型，直到模型为此系统优化的状态。

AT32 MCU 端：

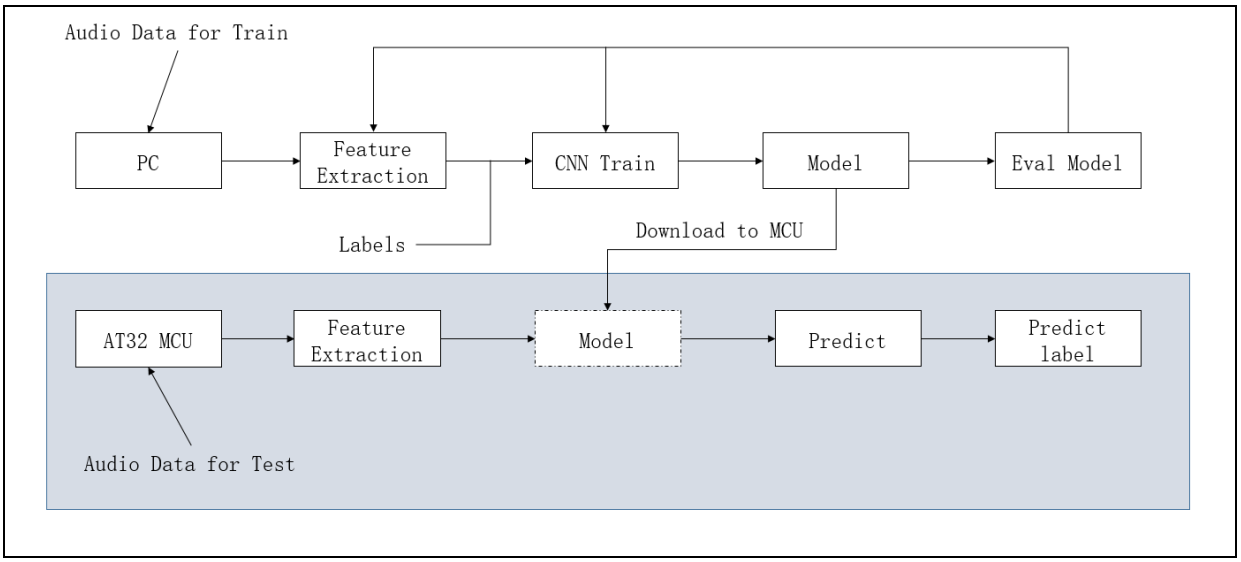
利用 ARM 提供的 CMSIS-NN 的函式库、DSP 函数库和 CNN 函数库，结合 PC 端训练好的模型（该模型已下载待 MCU）。对输入到 MCU 端的音频数据进行识别，实现对该语音数据可能的标签进行分类和预测。

因此，对于既定模型的 KWS 识别，AT32 MCU 端可实现完全离线识别，无需实时与 PC 通信或联网通信。本示例，AT32 MCU 端智能识别的关键词列表如下
"yes","no","up","down","left","right","on","off","stop","go";

没有输入信号时，输出标签为"Silence";输入信号不在关键词列表时，输出标签为"Unknown"。

注意：由于篇幅限制，本文只介绍 AT32 MCU 的实现流程，如下图阴影部分。

图 2. KWS 实现流程



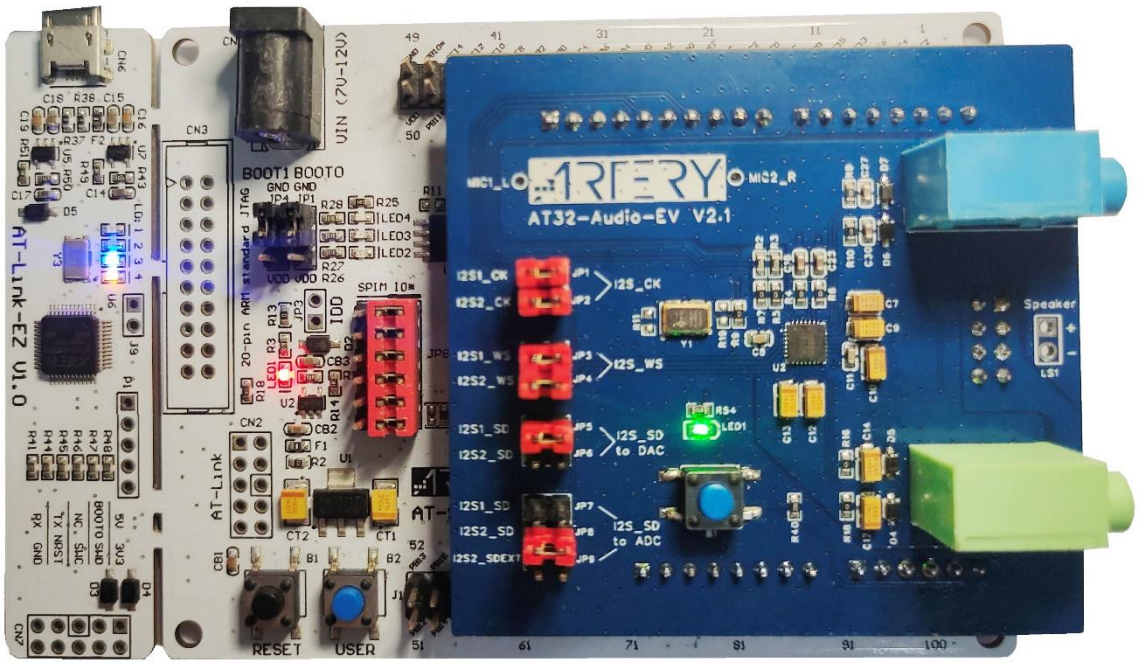
3.2 资源准备

1) 硬件环境:

AT-START-F403A BOARD V1.x

AT32-Audio-EV V2.x

图 3. KWS 测试的硬件环境



2) 软件环境

MDK V5.31 或更新版本，使用 ARM Compiler V6 进行编译

…\PACK\ArteryTek.AT32F403A_407_DFP.2.1.2.pack 或更新版本

…\PACK\ARM.CMSIS-DSP.1.11.0.pack 或更新版本

ML-KWS-for-MCU-master\Project\mdk_v5

3.3 软件设计

3.4 实验效果

在 AT32-Audio-EV V2.x 端 LINE_IN 输入语音信号后，AT Link 虚拟串口会打印输出 KWS 识别的标签和概率。

图 4. 串口打印识别信息

识别效果视频链接　　https://b23.tv/3UNwWEH

4 文档版本历史

表 1. 文档版本历史

日期	版本	变更
2022.9.7	2.0.0	最初版本

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。