DOI:10.13878/j.cnki.jnuist.2021.05.003

郭新[1]　罗程方[2]　邓爱文[2]

基于深度学习的开放场景下声纹识别系统的设计与实现

摘要

针对现实应用场景中短时语音和混叠有噪声情况下声纹识别准确性低的问题,本文设计了一种改进的基于深度学习的声纹识别算法,提高了声纹识别模型在短时语音和带噪环境下的鲁棒性,并将该模型部署到了嵌入式设备中.本文主要对声纹识别算法的编码层和损失函数进行改进.对于编码层,本文使用了基于差分编码的 NeXtVLAD 技术,同时对帧级特征中的静态声纹特征和动态声纹特征进行建模.对于损失函数,本文将基于小样本学习框架的余弦-原型损失函数 cosine-Prototypical 与附加间隔分类损失函数 AM-Softmax 进行融合来训练声纹识别模型,使得模型在特征空间中的同类特征尽可能集聚,异类特征尽可能分离.此外,本文还将声纹识别算法部署在 Raspberry Pi 平台上,实现了能快速推理的声纹识别系统.实验结果表明:这种改进的声纹识别系统在多种开放场景下,能够实时、准确地完成声纹识别任务,可以达到实际应用的要求.

关键词

深度学习;开放场景;短时语音;声纹识别;差分化编码;NeXtVLAD;树莓派

中图分类号 TN912.3;TP18

文献标志码 A

收稿日期 2021-08-08

资助项目 广东省青年创新人才项目(2018GkQNCX005)

作者简介

郭新,女,博士,讲师,研究方向为说话人识别.guoxin@gdcp.edu.cn

1 广东交通职业技术学院 机电工程学院,广州,510520

2 华南理工大学 自动化科学与工程学院,广州,510641

0 引言

声纹作为生物特征识别中的一种行为特征,因其蕴含丰富的个人信息,比如性别、年龄、情绪、语种等,被作为身份识别的一种方式,且因其伪造难度高、隐私性弱、辨识性强等优势,近些年来引起了业界广泛的关注.2018 年被称为“声纹元年”,是声纹技术迈向产业化的重要转折点,这一年,中国人民银行发布了《移动金融基于声纹识别的安全应用技术规范》(JRT0164—2018)[1].声纹识别技术在国内正逐渐被应用于金融、社保、公安、智能家居等重要领域.

在学术界,声纹识别又称说话人识别,它是提取语音信号中能够表征说话人个性特征的信息,利用传统机器学习或深度学习,自动地实现说话人身份识别的一种生物特征识别技术.声纹识别(Speaker Recognition,SR)分为声纹辨认(Speaker Identification,SI)和声纹确认(Speaker Verification,SV)[2],其中声纹辨认是 1:n 任务,声纹确认是 1:1任务,本文算法主要是基于 SV 的,整个系统的实现是两者兼有.目前,说话人识别应用的痛点是短时语音和基于开放环境的噪声影响,这就是本文的主要研究内容之一.

20 世纪 90 年代,声纹识别最经典的模型是 GMM-UBM 模型及 i-vector,在深度学习之前,它们一直是占据业界的主流技术.近些年来深度学习下的声纹研究已呈现愈来愈繁荣的局面,从深度卷积网络下的 d-vector 技术到 x-vector 技术[3],尤其 x-vector 是当前声纹领域中主流的模型框架.x-vector 框架下的主要研究内容包括声音预处理、特征提取层、特征编码层和损失函数的优化.对于声音预处理,常用的是 MFCC (Mel-scale Frequency Cepstral Coefficients) 和 Fbank (FilterBank),当然也可以使用原始语音,但原始语音的训练需要强大的内存和算力,比较耗费成本;特征提取层网络主要是 TDNN[4]、DCNN[5] 和 RNN[6];特征编码层的作用是将不等长帧级别的语音特征转换成等长的语句级特征,以方便特征的比对,目前主流的编码层融合技术有统计池化 SP[7]、ASP[8]、NetVLAD[9] 及 GhostVLAD[9] 等;损失函数方面除了应用分类损失函数外,还根据声纹模型的本质结合了度量损失函数[10],也取得了优异的性能.

本文的工作主要是设计并实现一套基于深度学习模型下声纹识别系统.该系统的算法是基于端到端的深度声纹模型,能实现特征类间距离增大、类内距离减小的目标,由此改善短时语音和混叠噪声影

响下声纹识别模型的可辨别性,且参数量较少,更适合部署在嵌入式硬件上,同时,性能依然能够与目前最优秀的性能比肩.硬件方面是利用 Raspberry Pi 实现实时、准确地操作,以得到良好的用户体验,达到实际场景的应用需求.

1 声纹识别系统

本文的声纹识别是基于深度学习的,深度学习声纹的框架可以分为声纹录入和识别两个阶段.深度学习声纹识别系统的框架如图 1 所示.

如图 1 所显示,声纹模型的建立需要大量的训练实验,才能构建合适的模型,声纹建模视为训练阶段,声纹录入和识别阶段就是测试阶段.

本文目标是设计并制作一个能实际应用的声纹识别系统,可以实现以下功能:

1) 实现用户注册和身份识别功能;

2) 实现开放场景下声纹识别的实时性,即短时内完成声音采集、注册和识别;

3) 实现可视化识别结果;

4) 实现脱机离线运行功能.

要实现以上功能需求,需设计声纹识别系统的软件部分和硬件部分.

2 声纹识别系统的软件设计

2.1 软件框架结构

该软件系统主要包含声音采集模块、声音预处理模块、特征提取模块、注册与识别模块和结果输出模块这五部分.为了充分发挥系统性能和方便代码维护和更新,本软件使用了多线程的编写方式.声纹识别主要分为三个线程:声音采集子线程、主线程、注册和识别子线程.声音采集子线程负责调用麦克风阵列采集声音,然后触发子函数将声音回传到主线程中.主线程负责整个软件程序的工作逻辑和各种工具函数调用,同时负责 GUI 的绘图和刷新、声音采样和预处理.注册和识别子线程负责提取声音特征、特征匹配和识别判断.软件系统总体流程如图 2 所示.

2.2 算法框架结构

图 2 中声纹识别模型的好坏直接由声纹识别的算法决定,如何设计一个优秀的算法是本工作的关键.

该声纹识别任务的算法框架主要包括四个部分:输入特征帧、主干网络(特征提取器)、编码层和损失函数,如图 3 所示.其中输入音频为变长的帧级特征,经过预处理后,成为特征为 40 维的对数梅尔频谱特征,即 Fbank.

2.2.1 主干网络

主干网络,也称帧级别特征提取器.该网络由 Squeeze-and-Excitation(SE)模块[11]和快速 resnet34 网络[12]组成.主干网络可接受任意长度的语音作为输入,并产生任意长度的帧级别特征.Fast-ResNet34 结构与原始 34 层 ResNet 网络结构相同.但是,为了降低计算成本,Fast-ResNet34 中的每个残差模块仅使用原始通道数量的 1/4,以此作为特征提取器,可以有效提取局部多帧声学特征.

2.2.2 编码层

SV 中编码层的作用就是编码声纹特征为特定的语句长度,即将变长的帧级特征转化成定长的语句级特征,以最大程度保留整体特征,这里使用差分

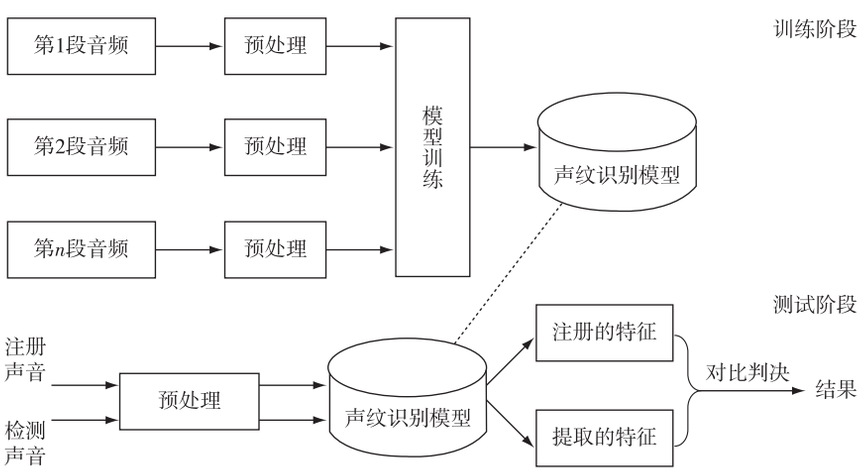

图 1 深度学习声纹识别系统框架

Fig. 1 Framework of deep learning based SR system

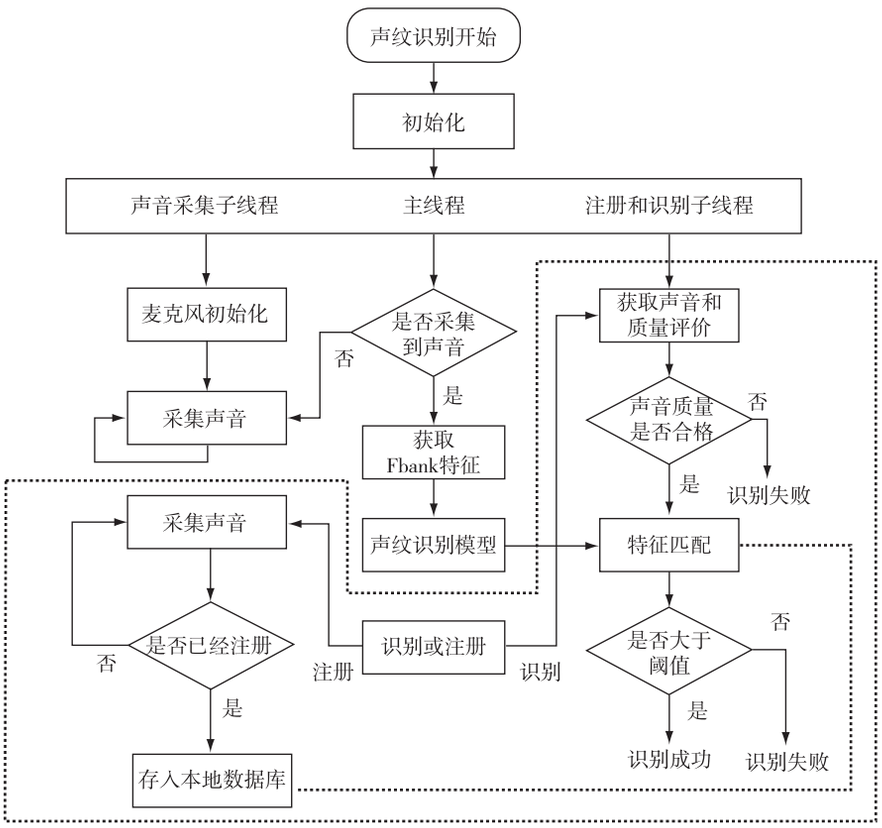

图 2　声纹识别软件总体流程

Fig. 2　Overall flow chart of the SR software

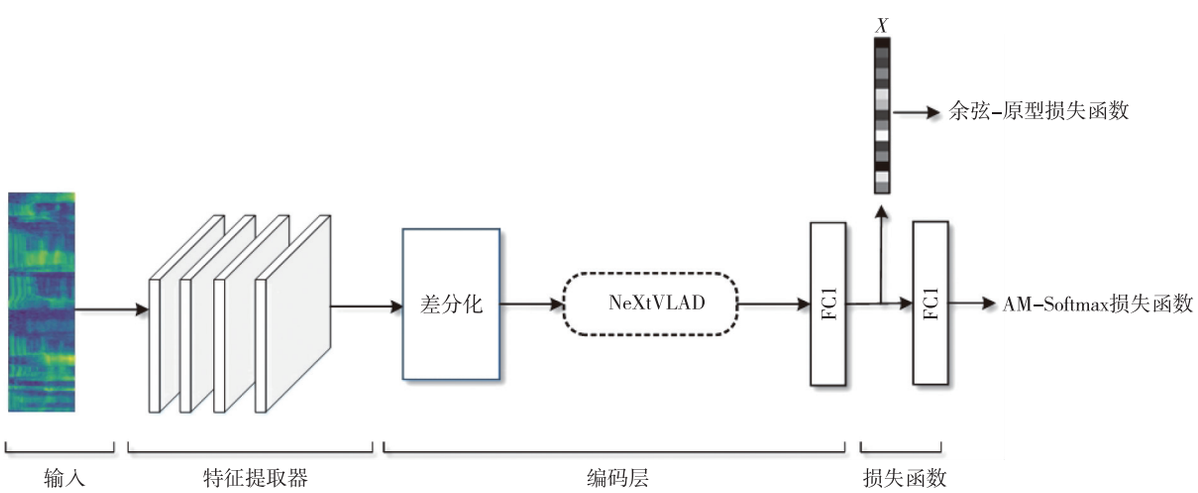

图 3　声纹系统算法框架

Fig. 3　Algorithm framework of the SR system

化编码方式,以捕捉特征帧之间的静态特征和动态特征,差分化编码层结构如图 4 所示.

图 4 中的 delta 是差分化编码,差分化公式可表示为

$$\Delta^t = \frac{\sum_{a=1}^{A} a(f_{t+a} - f_{t-a})}{2\sum_{a=1}^{A} a^2}, \tag{1}$$

式中:a 是差分阶数,$a = \frac{L-1}{2}$,L 是窗长.例如:L =5

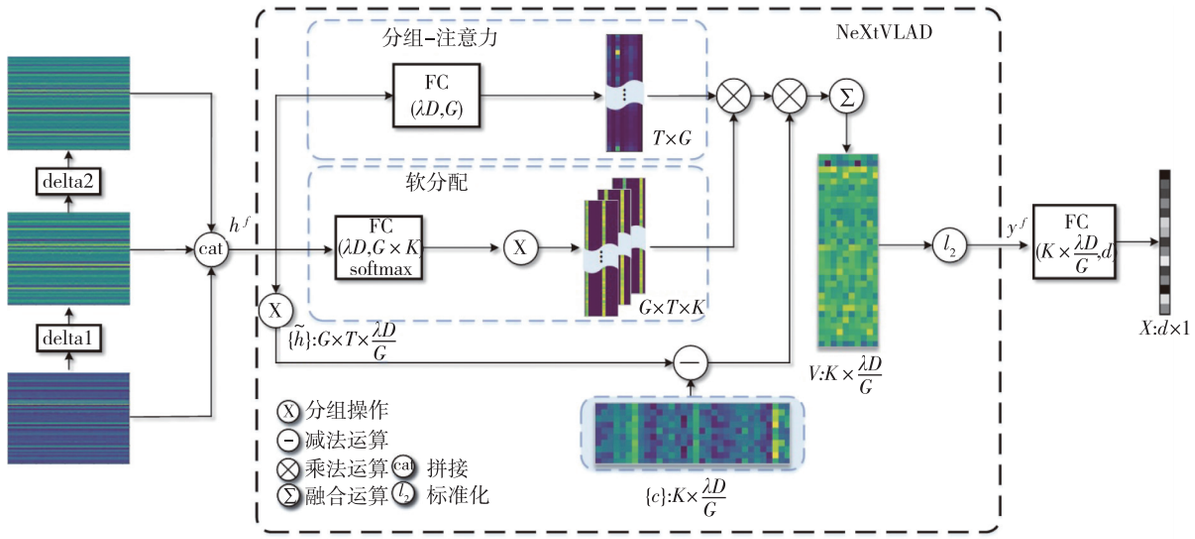

图 4　差分化输入的编码结构

Fig. 4　Encoding structure of differential input

时,$a=2$,就代表进行二阶差分的计算.

NeXtVLAD 原本是用在视频压缩上的一种编码方法,本文将其引入到 SV 任务中,实验结果表明 NeXtVLAD 在声纹识别中也具有优秀的表现[5]. NeXtVLAD[13]与 NetVLAD[9]性能相似,但模型参数大大减少,提升了训练的收敛速度.

如图 4 所示,假设特征提取器输出的隐藏层特征为 f^t,那么输入编码层的特征经过一阶差分和二阶差分的拼接,输入编码层的隐藏特征可以表示为

$$h^f = f^t \oplus f^t_{\Delta^1} \oplus f^t_{\Delta^2}\ , \tag{2}$$

式中 Δ^1代表一阶差分,Δ^2 代表二阶差分.

那么经过编码层的特征向量可表示为

$$\boldsymbol{y}^f = f_{\mathrm{NeXtVLAD}}(h^f)\ . \tag{3}$$

经过一个全连接后,得到声纹特征 embedding 的表示为

$$x = \mathrm{FC}(\boldsymbol{y}^f)\ . \tag{4}$$

2.2.3　损失函数

声纹识别在实际应用场景中是开集任务,兼具分类和小样本学习两种属性,本文使用组合函数[5],即融合了基于间隔的分类损失函数 L_{AMS}[14] 和基于小样本的原型余弦损失函数 L_{CP}[15],如式(5) 所示:

$$L = L_{\mathrm{CP}} + \beta L_{\mathrm{AMS}}\ , \tag{5}$$

式中

$$L_{\mathrm{AMS}} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{\boldsymbol{e}^{S_{n,n}}}{\sum_{j=1}^{N}\boldsymbol{e}^{S_{n,j}}}\ , \tag{6}$$

$$L_{\mathrm{CP}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\boldsymbol{e}^{s(\cos\theta_{y_i}-m)}}{\boldsymbol{e}^{s(\cos\theta_{y_i}-m)}+\sum_{j\neq y_i}\boldsymbol{e}^{s\cos\theta_j}}\ , \tag{7}$$

N 代表一个 minibatch 中说话人的个数,$\boldsymbol{e}^{S_{n,n}}$ 和 $\boldsymbol{e}^{S_{n,j}}$ 代表相似矩阵,θ_{y_i}代表目标角度,权重 $\beta=1$ 时可以达到最好的效果,s 是一个缩放因子,设为 30.

损失函数 L_{AMS}是明确鼓励声纹类间距离拉大,L_{CP}可以在声纹特征中找到与目标样本接近的原型来实现度量空间的优化,使得同类之间距离拉小,同时原型网络也可以很好地处理训练集中未曾出现的样本归属问题,大大提升了模型的鲁棒性.

3　声纹识别系统的硬件实现

3.1　硬件结构

声纹识别系统的硬件主要包括树莓派(Raspberry Pi,RPi)核心板、电源模块、麦克风和显示屏.该系统算法是基于 Ubuntu 系统环境进行开发、调试和部署进行的.其结构示意和实物图分别如图 5 和图 6 所示.

其中 Raspberry Pi 核心板是 4 GB、4B 的配置套件,具有高计算能力和价格低廉等特点.其 CPU 使用的是 ARM 的 Cortex-A 系列种的 A72,是 ARM 首次采用 64 位 ARM v8-A 的四核处理器,时钟频率为 1.5 GHz,树莓派 4B 的 GPIO 引脚的速度 50.8 kHz,比树莓派 3B 提升了 2 倍,性能提升的同时价格并没有明显上涨;电源使用可以支持树莓派的 UPS 板+100 00 mA 电池的扩展板;麦克风利用支持树莓派

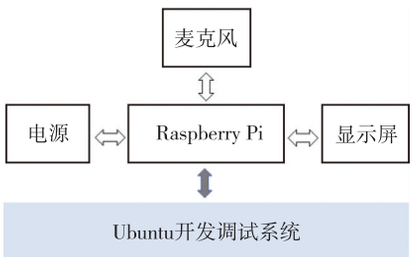

图 5　声纹识别硬件结构示意图

Fig. 5　Schematic of the SR hardware

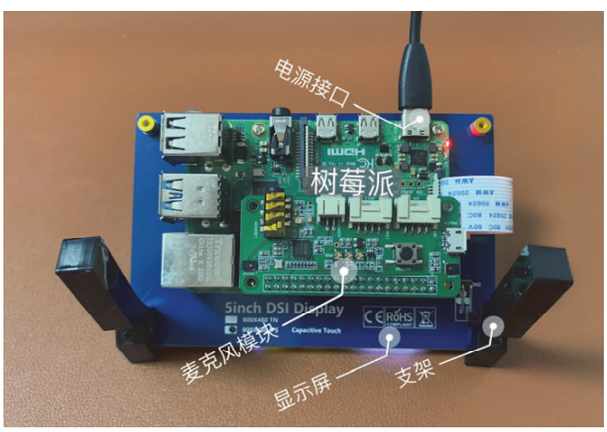

图 6　声纹识别系统实物图

Fig. 6　Physical picture of the SR system

的 ReSpeaker Pi 的麦克风阵列,可实现 3 m 拾音,支持双麦克风,声音更加清晰;显示屏使用 5 寸的 IPS 电容触摸屏,分辨率 800×480,可以实现 PWM 亮度调节.

3.2　GUI 设计

声纹识别系统的人机交互界面 GUI 是基于 PyQt5 开发的,PyQt5 是基于图形程式框架 Qt5 的 python 接口.该识别系统的人机交互界面是基于实际需求而设计的,力求简洁大方、易于操作.如图 7a 和 7b 所示.

该交互界面结构框架分为系统登录模块、用户信息管理模块、声纹信息管理模块、声纹参数配置模块和识别结果输出模块.通过对声纹交互系统的设计与部署,极大程度地提高了声纹管理的效率,更好地促进了声纹技术的发展.

4　实验结果分析

下面通过实验验证所提算法的先进性以及系统的有效性和实用性.

4.1　算法先进性

在说话人识别中,常以 FAR 为横坐标、FRR 为纵坐标绘制成一条曲线(ROC 曲线),当此曲线同 45°角,即 $y=x$ 直线相交时候交点对应的值,称为 EER,此时 FRR=FAR,此评价指标最常用于说话人识别中,其中 EER 是错误拒绝率 FRR(False Rejection Rate)和错误接收率 FAR(False Acceptance Rate)相等时的错误率,通常 EER 值越低代表性能越好.

表 1　各系统的性能和模型参数对比

Table 1　Comparison of performance and size of model parameters between different systems

系统	EER/%	参数量/MB
基线(fast-ResNet34,Vox1)	3.50	2.0
基线(TDNN,Vox2)[16]	2.47	8.9
本文算法(Vox1)	2.73	1.5
本文算法(Vox2)	1.83	1.5

表 1 显示了本文算法在 VoxCeleb-1 和 VoxCeleb-2 上性能和模型参数的优越性.从表 1 可知,本文的基于差分编码的声纹识别算法在公开数

a.声纹识别系统初始化界面　　b.识别结果界面

图 7　声纹识别系统人机交互界面

Fig. 7　GUI of the SR system

据集 VoxCeleb-1 和 VoxCeleb-2 都取得优异的性能,在数据集 VoxCeleb-1 上,精准度比基准线(系统为 NeXtVLAD+AAMS)下降 22%,参数量下降 25%;在数据集 VoxCeleb-2 上,精准度比基准线[16] 下降 25.9%,参数量下降 83.1%,该系统在算法上基本达到了如今声纹系统单尺度结构下的最优水平,且模型参数小,更合适部署在嵌入式设备上.

4.2 多环境下声纹系统识别结果

在本系统上对 12 个年龄段在 18~60 岁的人,在 5 种环境下,进行验证性实验,每种环境每个时长测试 3 次,进行了共计 540 次的实验,那么每种情况就是进行了 36 次实验,表 2 中百分数表示准确率,即准确的次数除以 36.实验结果如表 2 所示.

表 2 多种环境下声纹识别效果

Table 2 Performances of speaker recognition under various noisy scenarios %

环境	时长		
	3 s	6 s	10 s
室内静声(约 25 dB)	95.0	99.2	99.5
室内轻微噪声(约 15 dB)	90.5	95.2	97.2
室内较大噪声(约 0 dB)	80.8	88.2	92.1
室外轻微噪声(约 15 dB)	90.1	94.4	98.4
室外较大噪声(约 0 dB)	82.5	89.7	92.4

由表 2 可看出,在轻微噪声情况下,室内室外的识别效果均达到 90.1%以上,尤其是在时长大于 6 s 的情况下,本声纹识别的准确率都在 94.4%以上,可以达到实际应用的需求.室内较大噪声的效果比室外噪声差,这是因为在室内测试语音受混响影响比较大,短时语音(3 s)的准确率 82.5%,这是比较极端的情况,实际场景应用中,还是比较少见的.

5 结束语

本文针对现实应用场景下短时语音和混叠有噪声情况下声纹识别准确性低问题,提出了一种改进的深度学习下声纹识别算法并将该模型部署到了嵌入式设备中,为实现声纹技术的落地提供了助力.该系统算法方面,主要改进了编码层和损失函数,结合差分化方式处理特征提取器输出的帧级特征,它对帧级特征中的静态声纹特征和动态声纹特征进行建模,将差分处理后的特征输入到编码层,并在编码层使用 NeXtVLAD 技术;此外,本文还融合了基于小样本的余弦-原型度量损失函数和 AM-Softmax 分类损失函数,使得模型在特征空间中同类特征距离尽可能小,异类特征之间的距离尽可能远,即增强了声纹特征的可辨识性,这使得算法性能达到当前最优水平.硬件方面,本文使用性价比超高的 Raspberry Pi 作为控制核心,进行系统部署,且能够快速实现推理.实验结果表明,在多种开放场景下,这种改进的声纹识别系统能够实时、准确地完成声纹识别任务,可以达到实际应用的要求.

参考文献

References

[1] 刘丽敏,荆继武.快速发展的我国生物特征识别标准规范[J].中国信息安全,2019(2):68-72
LIU Limin, JING Jiwu. Rapidly developing biometric identification standard specification in China [J]. China Information Security,2019(2):68-72

[2] Reynolds D A, Rose R C. Robust text-independent speaker identification using Gaussian mixture speaker models[J].IEEE Transactions on Speech and Audio Processing,1995,3(1):72-83

[3] Snyder D,Garcia-Romero D,Povey D,et al.Deep neural network embeddings for text-independent speaker verification [C] // Interspeech 2017. ISCA: ISCA, 2017: 999-1003

[4] Desplanques B, Thienpondt J, Demuynck K. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification[J].arXiv e-print,2020,arXiv:2005.07143

[5] Cai W C,Chen J K,Li M.Exploring the encoding layer and loss function in end-to-end speaker and language recognition system[C]//Odyssey 2018 The Speaker and Language Recognition Workshop.ISCA:ISCA,2018.DOI: 10.21437/odyssey.2018-11

[6] Senior A, Sak H, Shafran I. Context dependent phone models for LSTM RNN acoustic modelling[C] // 2015 IEEE International Conference on Acoustics,Speech and Signal Processing (ICASSP).April 19-24,2015,South Brisbane,QLD,Australia.IEEE,2015:4585-458

[7] Nagrani A,Chung J S,Zisserman A.VoxCeleb: a large-scale speaker identification dataset [C] // Interspeech 2017. ISCA: ISCA, 2017. DOI: 10.21437/interspeech.2017-950

[8] Okabe K, Koshinaka T, Shinoda K. Attentive statistics pooling for deep speaker embedding [C] // Interspeech 2018. ISCA: ISCA, 2018. DOI: 10.21437/interspeech.2018-993

[9] Xie W D,Nagrani A,Chung J S,et al.Utterance-level aggregation for speaker recognition in the wild [C] // ICASSP 2019: 2019 IEEE International Conference on Acoustics,Speech and Signal Processing (ICASSP).May 12-17,2019,Brighton,UK.IEEE,2019:5791-5795

[10] Kye S M,Kwon Y,Chung J S.Cross attentive pooling for speaker verification[C] //2021 IEEE Spoken Language Technology Workshop (SLT). January 19-22, 2021, Shenzhen,China.IEEE,2021:294-300

[11] Hu J, Shen L, Sun G. Squeeze-and-excitation networks [C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. June 18-23, 2018, Salt Lake City, UT, USA. IEEE, 2018: 7132-7141
[12] Chung J S, Huh J, Mun S, et al. In defence of metric learning for speaker recognition [C]// Interspeech 2020. ISCA: ISCA, 2020. DOI: 10.21437/interspeech.2020-1064
[13] Lin R C, Xiao J, Fan J P. NeXtVLAD: an efficient neural network to aggregate frame-level features for large-scale video classification [M]// Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 206-218
[14] Wang H, Wang Y T, Zhou Z, et al. CosFace: large margin cosine loss for deep face recognition [C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. June 18-23, 2018, Salt Lake City, UT, USA. IEEE, 2018: 5265-5274
[15] Snell J, Swersky K, Zemel R S. Prototypical networks for few-shot learning [J]. arXiv e-print, 2017, arXiv: 1703.05175
[16] Wu Y F, Guo C K, Gao H C, et al. Vector-based attentive pooling for text-independent speaker verification [C]// Interspeech 2020. ISCA: ISCA, 2020: 936-940

A deep learning-based speaker recognition system for open set scenarios

GUO Xin[1] LUO Chengfang[2] DENG Aiwen[2]
1 School of Mechanical and Electrical Engineering, Guangdong Communication Polytechnic, Guangzhou 510520
2 School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641

Abstract Due to the low accuracy of speaker recognition for short-term speech or under overlapping noisy situations, a new speaker recognition algorithm based on deep learning is proposed and then deployed on an embedded device. The encoding layer and loss function are the two aspects to improve the speaker recognition system in robustness. For the encoding layer, the NeXtVLAD technique based on differential encoding is used to model both static and dynamic speaker features at frame level. For the loss function, the cosine-prototypical loss function based on small-sample learning framework is fused with the additional margin classification loss function AM-Softmax to train the speaker recognition model, which enables the model to collect similar features and separate dissimilar features as much as possible in the feature space. Then the improved speaker recognition algorithm is deployed on the Raspberry Pi platform to realize speaker recognition with fast inference. The experimental results illustrate that the system can accomplish speaker recognition in real time and accurately under various open set scenarios, and meet the requirements of practical applications.

Key words deep learning; open set; short-term speech; speaker recognition; differential encoding; NeXtVLAD; Raspberry Pi(RPi)