硕 士 学 位 论 文

基于深度学习的说话人识别方法研究

Study on Speaker Recognition Based on Deep Learning

作 者 姓 名：徐启鹏

学科、 专业：电子与通信工程

学　　号：31909065

指 导 教 师：殷福亮 教授

陈　喆 教授

完 成 日 期：2022 年 6 月 5 日

大连理工大学

Dalian University of Technology

摘　要

说话人识别是应用语音信号处理方法，根据给定的语音确定说话人的身份相关信息，它在网络通信、消费电子、智能终端、人机交互、安全支付等领域有着广阔的应用前景。说话人识别按照任务类型主要分为说话人辨识、说话人确认和说话人日志，依据文本内容则分为文本无关和文本相关说话人识别。近年来，随着深度学习理论的发展，基于深度神经网络的说话人识别技术取得了新进展，但其计算量大、参数多，限制了其在嵌入式系统中的应用，此外其识别性能仍然有待提升。

本文应用深度学习理论，研究了基于深度学习的文本无关说话人识别方法，所做的主要工作如下：

(1) 针对现有方法参数量大、效率较低的问题，提出了基于 TSCA-ResMBConv 结构的说话人识别模型。在该模型中，引入反向瓶颈层(Inverted Mobile Bottleneck Convolution, MBConv) 和混合反向瓶颈层(Fused Inverted Mobile Bottleneck Convolution, Fused-MBConv) 结构，大幅度减少参数量；考虑到说话人可能会在特定的时间段发出比其它时间段更有个性特征的声音，提出了时间段通道注意力机制(Time Segment Channel Attention, TSCA)模块，相较于压缩激励(Squeeze-and-Excitation，SE)模块，TSCA 能够建立特征图通道信息和时间段的关联，从而提升模型的识别性能。实验结果表明，本文提出的 TSCA-ResMBConv 结构能够以较少的参数量实现优于基准方法的识别性能。

(2) 考虑到说话人确认和说话人辨识两者间的强关联性以及说话人识别网络的通用结构时间池化层的特点，提出了一种结合多任务学习方案的时间池化方法。该方法应用多任务学习策略，使得时间池化层采用的自注意力机制的查询向量具有更多的有效信息，并在模型训练过程中将三元组损失函数(Triplet loss)考虑样本对与 AAM-Softmax 考虑类别信息的特点相结合，构建包含说话人确认任务和说话人辨识任务的网络模型。实验结果表明，该方法能够有效提高说话人确认任务的识别性能。

关键词：说话人识别；说话人确认；注意力机制；深度可分离卷积；多任务学习；度量学习

Study on Speaker Recognition Method Based on Deep Learning

Abstract

Speaker recognition is the task of identifying persons from their voices by virtual of speech signal processing method, and it has a broad application prospect in network communication, consumer electronics, intelligent terminal, human-computer interaction, secure payment and other fields. There are three major branches of speaker recognition: speaker identification, speaker verification and speaker diarization. Speaker recognition can also be divided into text-independent and text-dependent according to text information. In recent years, with the development of deep learning theory, speaker recognition technology based on deep neural network has made new progress, but its large amount of parameters and computation cost limit its application in embedded systems. In addition, its recognition performance still needs to be improved.

In this thesis, deep learning theory is applied to study the text independent speaker recognition method based on deep learning theory. The main work is as follows:

(1) To reduce the requirement of high computational resources for existing methods, an efficient speaker recognition model based on TSCA-ResMBConv structure is proposed. In this model, the number of parameters is greatly reduced by introducing Fused MBConv and MBConv structures. The TSCA module is proposed in consideration of the fact that speakers may produce more characteristic sounds during certain periods of time than other periods of time. Compared with SE module, TSCA can establish the association between channel information and time segment, thus improving the recognition performance of the model. Experimental results show that the TSCA-ResMBConv structure proposed in this thesis can achieve better recognition performance than the benchmark method with fewer parameters.

(2) Considering the strong correlation between speaker verification and speaker identification, as well as the characteristics of time pooling layer in the general structure of speaker recognition network, a time pooling method combined with multi-task learning scheme is proposed to improve recognition performance. In this method, multi-task learning strategy is applied to make the query vector of self-attention mechanism adopted by time pooling layer have more effective information. In the process of model training, the Triplet considering sample pairs and AAM-Softmax considering category information are combined to construct a network model containing speaker verification task and speaker identification task. Experiments show that this scheme can improve the recognition performance of speaker verification task effectively.

Key Words: Speaker Recognition; Speaker Verification; Attention Mechanism; Depthwise Separable Convolution; Multi-task Learning; Metric Learning

目录

1 绪论

本章主要介绍说话人识别的研究背景意义、发展历史和研究现状，并概述本文的主要研究工作。

1.1 研究背景及意义

同指纹、人脸信息类似，声纹是每个人均具有的个性特征。说话人识别，也被称为声纹识别。说话人识别就是根据给定的语音信息，应用语音处理方法，识别出其中的说话人身份信息。说话人识别技术在人机交互、消费电子、智能终端、支付安全等领域具有广泛应用前景。

说话人识别是语音信号处理[1]领域的重要研究方向。依据识别对象的不同，说话人识别分为文本相关、文本无关以及文本提示型三种类型。文本相关说话人识别任务要求说话人测试时录入指定文本的语音，文本无关说话人识别任务则不对录入文本作规定，文本提示型说话人识别任务则要求说话人按提示文本进行发音，提示文本通常来源于一个较大的确定集合。相对而言，文本无关的说话人识别任务应用场景更为广泛，同时用户使用方式更为简单便捷。

说话人识别从任务上可以分为说话人确认(speaker verification)、说话人辨识(speaker identification)和说话人日志(speaker diarization)[2]。说话人确认是指将待测语音与某个说话人的已知语音进行匹配，来确认待测语音是否出自于该说话人。

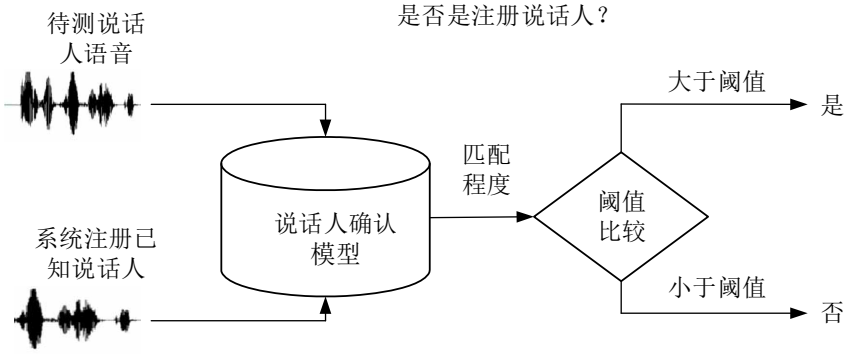


图 1.1 说话人确认系统框图

Fig. 1.1 Block diagram of speaker recognition system

说话人辨识则将待测输入音频与已知的说话人语音集合中语音进行匹配，从而判断待测音频来源于已知说话人集合中的哪一个。说话人辨识依据是否涉及集外数据分为闭

集说话人辨识与开集说话人辨识。闭集说话人识别即待检测说话人预先确定在已知的说话人集合中，将待测语音的特征与已知说话人集合中全部说话人特征进行匹配计算，匹配度最高的即为待测语音所属的说话人；而对于开集说话人辨识任务，即待测说话人可能不在已知的说话人集合中，此时在进行说话人辨识任务时，既需要对来源于集内的已知说话人语音做出正确匹配，也需要当待测说话人来源于集外时能够辨认出是集外的说话人。说话人辨识任务与说话人确认任务有很大的相关性，说话人确认任务可以视为系统注册说话人数量为 1 时的开集说话人辨识任务。

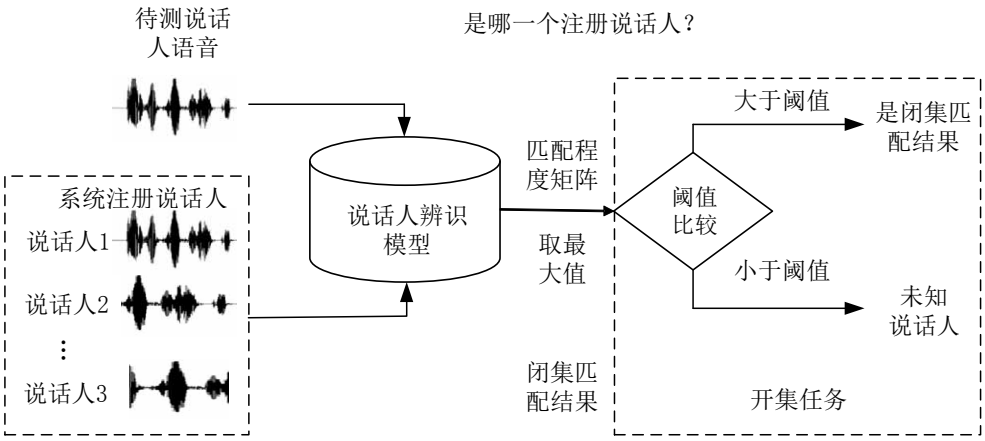


图 1.2 说话人辨识系统框图

Fig. 1.2 Block diagram of speaker classification system

说话人日志也称为说话人分割聚类，如图 1.3 所示：

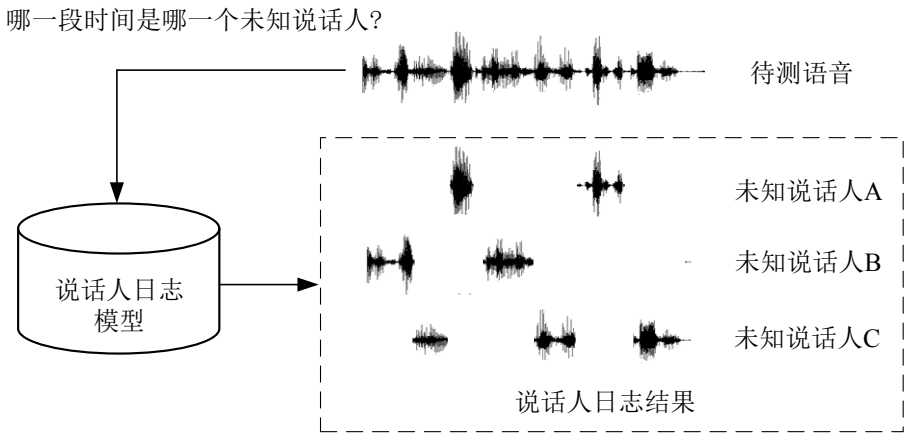


图 1.3 说话人日志系统框图

Fig. 1.3 Block diagram of speaker diarisation system

说话人日志区别于前两种任务并没有已知的注册说话人信息，其任务目标则是对一段语音进行处理，判断每一段语音来源于哪一个未知说话人。在视频会议、法庭录音等领域均有应用价值，比如对于会议录音用说话人日志的方式可以分离出各个说话人对应的语音便于后续的分析处理。说话人日志系统的示意图如图 1.3 所示。

多年来，传统说话人识别技术的研究已取得较大进展，并在实际中得到应用，涌现出一些软硬件产品。近年来，随着深度学习理论的快速发展，基于深度神经网络的说话人识别技术取得了新突破，但是深度学习方法计算量和参数量均很大，限制了其在嵌入式系统中的应用，其识别性能也有待提高。为此，本文应用深度学习理论，研究基于深度神经网络的文本无关说话人识别方法。

1.2 说话人识别发展历史与研究现状

1946 年美国贝尔实验室首次提出语谱图的概念。1960 年代，Gunnar Fant 研究发展了对人类语音产生机制的生理模型[3]，为心理声学和语音信号处理的发展奠定了重要基础。1962 年美国贝尔实验室的 Kersta[4]首次以实验形式验证了使用语谱图即声纹信息实现身份识别的有效性和可行性，并报告了通过肉眼观察的方式在特定说话人集合中实现 99%的识别准确性的实验结果。此后声纹这一生物信息逐渐得到认识并成为重要的研究方向。随着计算机以及语音数字信号处理技术的发展，产生了一些基于数字信号处理的说话人识别方法，其中研究适用于说话人识别的语音特征以及分类模型是两种重要的研究方向。

语音特征参数随着语音信号处理技术的发展得到大量研究，并在说话人识别领域得到关注。1969 年，Luck 研究了基于倒谱语音特征的说话人识别方法[5]，有效提升了说话人识别系统的识别性能。1976 年，Atal 探讨了不同语音测量方法，并研究了基于线性预测倒谱系数的说话人识别系统[6]，取得了良好的效果。1980 年，Davis 等人基于说话人识别系统研究了多种特征的有效性，通过实验得出梅尔频率倒谱系数[7]在说话人识别任务上有更好的表现。

说话人识别模型是重要的研究方向并逐渐成为说话人识别研究领域的主流。1963 年，Pruzansky 提出了基于模板匹配的说话人识别方法[8]。1981 年，Furui 等人提出了基于梅尔倒谱系数与动态时间规整的说话人识别方法[9]。1987 年，Burton 等人提出了基于矢量量化的文本相关说话人识别方法[10]，矢量量化技术在说话人识别任务中开始得到广泛关注与研究[11,12]。1989 年，Naik 提出了基于隐马尔可夫模型的说话人识别系统[13]并取得了一定的实验结果。1990 年代，Reynolds 提出了基于高斯混合模型的说话人识别方法[14]，

在 TIMIT 语音数据库实现了约 99%的闭集说话人识别准确率。随着机器学习领域支持向量机(support machine learning, SVM)[15]理论的发展，Schmidt[16]将支持向量机模型引入说话人识别取得了较好的结果。2000 年，Reynolds 提出了基于极大似然的通用背景模型[17]并应用于说话人确认系统，因其简单有效且鲁棒性强的优点而迅速成为说话人识别的主要技术。此后基于高斯混合模型以及通用背景模型的众多方法得到研究。2008 年以后，超矢量技术被提出并应用于说话人识别系统，Kenny 进一步提出联合因子分析技术[18,19]。2011 年，受联合因子分析启发 Dehak 进一步提出了 i 向量(i-vector)[20]，并在说话人识别领域得到广泛采用。

随着深度学习理论的发展，基于深度学习的说话人识别方法逐渐得到广泛的研究。常用的语音特征参数往往包含了较多的冗余信息，使用深度神经网络结构被认为能够从梅尔倒谱系数等语音特征中进一步提取出说话人特征(speaker embedding)。2014 年，Variani 将神经网络应用在说话人识别系统中，提出了通过神经网络结构提取的基于帧级别的特征 d 向量(d-vector)[21]，对后续基于深度学习的说话人识别模型有着重要的启发作用。2016 年，Snyder 提出了使用神经网络提取语段级特征的 x 向量(x vector)系统[22]，此后提取语段级别特征的方法在说话人识别领域被广泛采用。2018 年，Snyder 在原方法的基础上进一步提出了基于 TDNN 的 x-vector 方法[23]，得益于更大量的训练数据，实现了优于传统方法 i-vector 的表现。2020 年，Brecht 等人对 x-vector 的 TDNN 结构进行改进，加入了 Res2Net 结构[24]和通道注意力机制，提出了 ECAPA-TDNN 模型[25]，在说话人确认任务中取得了较好的性能表现。同年，Tawara[26]等人研究了基于极短语音的说话人识别方法，通过实验发现对于低于 1.4s 的极短语音，提取帧级别的特征优于语段级别特征，这对于说话人识别系统的实际应用有着一定的指导意义。2021 年， Kim 等人提出了基于自适应卷积神经网络的说话人识别方法[27]，该方法能够有效地利用说话人语音特性随着时间和音素改变的特点。

目前说话人识别仍有一些尚待解决的问题如下所示。

(1) 尽管语音的特征参数得到了大量的研究，目前仍然没有找到仅包含说话人个性特征的理想语音特征参数，大部分说话人识别系统直接采用和语音识别任务相同的特征参数。

(2) 同一说话人受到年龄增长或是健康状况等因素影响，语音信号会有较大的变化和差异[28]，这对于说话人识别系统有着很大的影响。说话人所处的不同的声学环境和场景，对于说话人识别系统的鲁棒性也是较大的挑战。很多实际场景收集到的语音往往经

过了一定的语音编码处理（比如电话音），说话人识别系统对各种编码语音的鲁棒性也是需要实际考虑的。

(3) 目前，基于深度学习的说话人识别方法可以取得良好的识别性能，但是其计算量和参数量大，对于实时系统的硬件性能有着较高的要求。因此，研究参数量和计算量相对较小的说话人识别系统，是实际应用中需要考虑的问题。

1.3 主要研究工作及章节安排

本文主要研究基于深度学习的文本无关说话人识别方法。本文主要工作如下：首先针对当前基于深度学习的说话人识别方法参数量大的问题，提出了低参数量的 TSCA-ResMBConv 结构。该方法使用反向瓶颈层结构大幅度减少参数量，并采用本文提出的 TSCA 模块提升识别性能；其次提出了一种基于自注意力机制和多任务学习的说话人识别方法，该方法可以替代说话人识别模型中常用的时间池化方法，有效提高说话人识别模型的识别性能。

本文分为四个章节，各章节的内容安排如下：

第一章为绪论，首先阐述了说话人识别的研究背景和研究意义，然后介绍了说话人识别的研究历史和研究现状，分析了近几年主流的说话人识别方法，最后介绍了本文的主要研究内容和章节安排。

第二章主要介绍了说话人识别领域的基础知识，包括说话人识别系统所涉及的语音处理基础，深度学习方法，本文所用的基准方法以及评价指标和方法。

第三章主要介绍了本文提出的基于 TSCA-ResMBConv 结构的说话人识别系统。首先详细介绍了坐标注意力机制的基本原理以及本文提出的 TSCA 模块的方法。其次介绍了反向瓶颈层的基本原理以及本文提出的 TSCA-ResMBConv 基本结构。然后对 TSCA-MBConv 的整体结构设置进行详细介绍，最后通过实验测试 TSCA 模块以及 TSCA-ResMBConv 结构的有效性。

第四章主要介绍了本文提出的基于自注意力机制以及多任务学习的说话人识别系统。首先对经典时间池化方法进行分析，并对本章采用的时间池化方法进行简要介绍。其次介绍了多任务学习的基本原理，进一步介绍了本文采用的多任务学习方法的损失函数设置以及三元组构建策略。接着详细介绍了本文提出的结合基于自注意力机制的时间池化层以及多任务学习的方法，最后通过实验对其有效性进行测试。

2 说话人识别相关知识

2.1 语音处理基础

2.1.1 语音的线性产生模型

语音信号的产生机理是理解说话人识别的知识基础。咽腔、口腔和鼻腔三个空气腔体共同组成人的声道系统，对发音有着决定性的影响。语音信号产生期间，肺部以及相连的肌肉起到声道系统的激励源作用，若声道处于收紧状态，则肺部产生的气流使得声带振动，此时产生的声音为浊音，而声带开闭一次的时间为基音周期，对应倒数被称为基音频率。若声带未振动则此时发出的声音为清音。当声带处于放松状态时，可以利用舌头和嘴唇发出摩擦音和爆破音两种声音。声道通过调音运动的方式，调整声道的形状从而发出各种不同的声音。调音所涉及的声道各部分器官即为调音器官，包括舌、颚、嘴和唇等部分。

依据前述声音的产生机制，可得语音信号生成的线性模型，如图 2.1 所示。

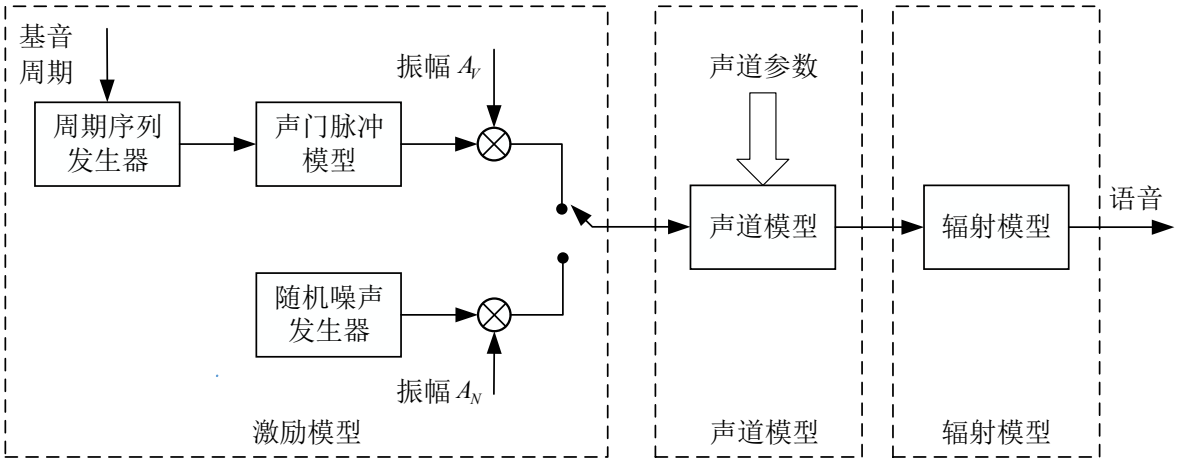


图 2.1 语音产生的线性模型

Fig. 2.1 Linear model of speech production

语音的线性产生模型将语音的产生过程分为激励模型，声道模型和辐射模型。研究表明在产生浊音、清音、摩擦音或爆破音时激励有所不同。当说话人发出浊音时，气流通过绷紧的声带产生振动，使声门处形成准周期脉冲串，该周期即为基音周期受到声带的绷紧程度影响。浊音对应的激励源为以基音周期为周期的斜三角形的脉冲串，其单个周期的信号可表示为：

$$g(n)=\begin{cases}\dfrac{1}{2}\left[1-\cos(n\pi / N_1)\right], & 0\le n\le N_1\\ \cos\left[\pi(n-N_1)/2N_2\right], & N_1\le n\le N_1+N_2\\ 0, & \text{其他}\end{cases} \tag{2.1}$$

式中N_1为斜三角波上升沿时间，N_2为下降沿时间。该函数的频域形式通常用 Z 变换的全极点模型形式表示：

$$G(z)=\frac{1}{(1-g_1z^{-1})(1-g_2z^{-1})} \tag{2.2}$$

单位脉冲串的 Z 变换形式为：

$$E(z)=\left(\frac{A_v}{1-z^{-1}}\right) \tag{2.3}$$

其中 A_v 控制浊音的能量：

于是，激励模型可以表示为：

$$\mathrm{U}(z)=E(z)G(z)=\frac{A_v}{1-z^{-1}}\times\frac{1}{(1-g_1z^{-1})(1-g_2z^{-1})} \tag{2.4}$$

当产生清音、摩擦音或爆破音时，气流直接经声门进入声道，此时的激励信号为零均值，方差为 1 的白噪声，仅需通过A_N调节其能量。对于声道模型，常用的一种数学模型是共振峰模型，即将声道看作谐振频率为共振峰的谐振腔，通常采用全极点模型来刻画共振峰特性，即：

$$V(z)=\frac{1}{\sum_{i=0}^{p}a_iz^{-i}} \tag{2.5}$$

口唇的辐射效应可以用一个高通滤波器表示，即：

$$R(z)=(1-rz^{-1}) \tag{2.6}$$

综上可得完整的语音信号产生模型$H(z)=U(z)V(z)R(z)$。

2.1.2 语音前端处理

说话人识别系统同语音识别系统等语音任务相似，在后续的模型处理之前，均包含分帧，加窗，特征提取等处理部分，这些部分也被称作语音前端处理。

(1)分帧

语音信号具有短时平稳性，即语音信号在较短的时间段内有着基本保持不变的特性。利用语音信号的短时平稳性对语音分段处理从而获得语音时变的特性。分的每一段可以称为一帧，根据语音的特性通常取 10~30 毫秒对应的序列长度即帧长，逐帧提取特征即可得到连续的特征序列，如图 2.2 所示。通常采用重叠分帧的方式处理语音，即帧与帧之间存在重叠部分，从而保证获取语音信息的连续性，避免因为后续加窗等处理而损失边界信息。对于输入 N 点语音信号，经过分帧后即可得到$x_i(n)$，其中$i = 1,2,\dots,n_s$，$n = 1,2,\dots,L$。式中n_s为帧数，L为帧长。

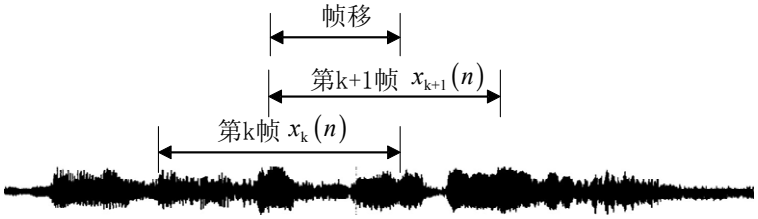


图 2.2 语音的分帧过程

Fig. 2.2 Process of speech signal framing

(2) 加窗

对语音信号分帧相当于对信号进行非周期截断，会导致频谱发生频带内拖尾现象，即出现频谱泄露。对分帧后的语音信号进行加窗操作可使每一帧信号具有部分周期函数的特性，因此能够有效减少频谱泄露。语音信号处理中常用的窗函数有汉宁(Hanning)窗和汉明(Hamming)窗等，汉宁窗和汉明窗的表达式为：

$$w(n)=0.5-0.5\cos\left(\frac{2\pi n}{N}\right),\quad n=0,1,\dots,N-1 \tag{2.7}$$

$$w(n)=0.54-0.46\cos\left(\frac{2\pi n}{N}\right),\quad n=0,1,\dots,N-1 \tag{2.8}$$

其中，$w(n)$是窗的系数，N 是窗的长度。

(3) 特征提取

无论是语音的时域表示还是频域表示，对于说话人识别任务区分性均不明显，而语谱图对于肉眼观察的方式而言有着较好的区分性但是对于数字信号处理方法而言仍然需要进一步处理。语音特征提取可以提取出相较于原始形式更有效的信息，从而便于后续处理。如前文所述声道形状对于发音有着决定性的作用，能够有效反应声道特性的特征在说话人识别任务得到了广泛应用，这些描述声道特性的参数特征通常有线性预测倒谱系数、感知线性预测、梅尔倒谱系数等。近些年来，对数梅尔频谱也在语音和音频相关任务中广泛使用。对数梅尔频谱较多保留了音频中的原始信息，适用于后端模型较为复杂的方法，比如深度学习方法。在本文中主要考虑的特征是梅尔倒谱系数和对数梅尔频谱，而对数梅尔频谱与梅尔倒谱系数有着极大的相似性，实际上仅比梅尔倒谱系数少一步 DCT 操作，其它相关的计算步骤完全相同。

对于人耳听觉机制的模仿是语音特征设计重要的方向。人耳对于声音频率的听觉范围是 20~20000Hz，在此范围内对于频率的感知特性并非线性。研究表明对于频率小于 1kHz 的声音，其物理频率与感知频率近似呈现线性特性，而对于频率大于 1kHz,的声音，两者关系则呈对数特性。人耳的听觉具有选择性，这是由于人耳基底膜分为许多不同部分分别对应于不同频率群，而对于同一频率群的语音，大脑叠加在一起进行判断。符合这种听觉机制的频率尺度即为梅尔频率。梅尔频率与实际频率之间的转换关系为：

$$f_{\text{Mel}} = 2595 \times \lg(1 + \frac{f}{700}) \tag{2.9}$$

对分帧加窗后得到的语音时域信号进行傅里叶变换，即可得到分帧的频域表示。将实际频率映射到梅尔频率后，用一组三角形带通滤波器模拟人耳听觉特性的滤波器组，将频域信号通过梅尔滤波器组即可得到梅尔频谱。梅尔滤波器组的频域形式为：

$$H_m(i) = \begin{cases} 0, & i < f(m-1) \text{或} i > f(m+1) \\ \dfrac{i - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le i \le f(m) \\ \dfrac{f(m+1) - i}{f(m+1) - f(m)}, & f(m) \le i \le f(m+1) \end{cases} \tag{2.10}$$

式中$H_m(i)$表示第 m 个梅尔滤波器的频域形式，*f(m-1)*、*f(m)*、*f(m+1)*分别表示该三角滤波器的下限频率、中心频率、上限频率。

人耳对声音响度的感知也是呈现对数特性的，对梅尔频谱进一步取对数后，则可得到对数梅尔频谱特征(Log Mel Spectrum)。进一步进行离散余弦变换，可得到梅尔倒谱系数(Mel Frequency Cepstral Coefficients，MFCC)[7]特征。

2.2 深度学习基础

2.2.1 基本结构

2014 年以后，随着神经网络方法的发展，深度学习方法逐渐成为说话人识别领域的研究重点。在说话人识别领域，多层神经网络的结构能够从对数梅尔频谱或梅尔倒谱系数等语音特征参数中进一步提取说话人的个性特征，从而便于进一步进行相关的说话人识别任务。神经网络的基本操作有全连接层、卷积层、池化层、激活函数、批归一化层(batch normalization,BN)[29]等。此外说话人识别领域常使用沿时间维度计算的池化层即时间池化层(temporal pooling, TP)。以下对神经网络的基础操作进行介绍。

(1) 卷积层

卷积(convolution)层是卷积神经网络结构的基本模块。图 2.3 展示了通过一个卷积核进行卷积操作生成一个特征图的过程。卷积操作通过多个卷积核大小的三维滑动窗口在特征图上进行滑动取值并与卷积核参数计算从而得到多个相应的特征图，该步骤通常认为提取了原特征图的多种特征。

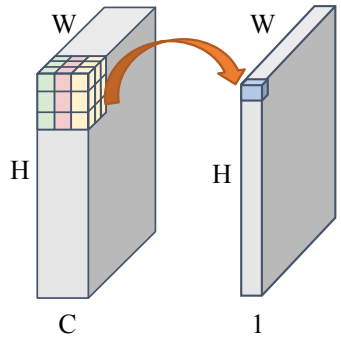


图 2.3 卷积运算

Fig. 2.3 Convolution operation

(2) 激活函数

激活函数(activation function)是非线性映射函数，是神经网络的重要模块。激活函数对于神经网络模型学习理解较为复杂的函数具有重要的作用，它们将非线性特性引入到神经网络模型中。常用的激活函数有 ReLU 函数和 Sigmoid 函数，其表达式为：

$$\mathrm{ReLU}(x)=\begin{cases}x, & x>=0\\ 0, & x<0\end{cases} \tag{2.11}$$

$$\mathrm{Sigmoid}(x)=\frac{1}{1+e^{-x}} \tag{2.12}$$

(3) 池化层

池化(pooling)层的作用是缩减模型的大小，提高计算速度，同时提高所提取特征的鲁棒性，图 2.4 展示了池化层处理的基本方式。常用的池化操作有最大池化层，平均池化层，最大池化层对每一个小区域选最最大值作为池化结果，平均池化层则选取平均值作为池化结果。

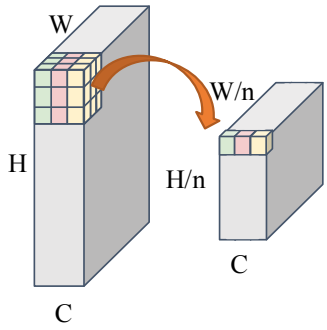


图 2.4 池化运算

Fig. 2.4 Process of pooling

(4) 全连接层

全连接(fully connected layer, FC)层可以在神经网络模型中实现分类作用。卷积层、池化层和激活函数等操作可以将原始数据映射到隐层特征空间，全连接层则往往起到将前几层学到的分布式特征表示映射到样本标记空间的作用。对于输入向量x_{in}，输出为$x_{out}=W\cdot x_{in}+b$。其中$x_{in}\in R^{C_{in}\times 1}$，$x_{out}\in R^{C_{out}\times 1}$，$W\in R^{C_{out}\times C_{in}}$，$b\in R^{C_{out}\times 1}$。

(5) 批归一化层[29]

批归一化(Batch Normalization, BN)层是神经网络常用的加速模型收敛的方法。对于一个小批次的输入$B=\{x_{1\ldots m}\}$，BN 层通过计算输入的均值和方差，并将其高斯归一化后进一步重构变换，使得其满足特定的高斯分布，从而使模型训练更加容易进行。BN 层由输入$B=\{x_{1\ldots m}\}$到计算输出$\{y_i=\mathrm{BN}_{\gamma,\beta}(x_i)\}$的计算过程如下：

$$\mu_{\mathrm{B}}=\frac{1}{m}\sum_{i=1}^{m}x_i \tag{2.13}$$

$$\sigma_{\mathrm{B}}^{2}=\frac{1}{m}\sum_{i=1}^{m}\left(x_{i}-\mu_{\mathrm{B}}\right)^{2} \tag{2.14}$$

$$\hat{x}_{i}=\frac{x_{i}-\mu_{\mathrm{B}}}{\sqrt{\sigma_{\mathrm{B}}^{2}+\dot{\mathrm{o}}}} \tag{2.15}$$

$$y_{i}=\gamma\hat{x}_{i}+\beta\equiv\mathrm{BN}_{\gamma,\beta}\left(x_{i}\right) \tag{2.16}$$

其中γ和β为可学习参数。

(6) Softmax 层

Softmax 层即归一化指数函数，常用于多分类问题例如说话人辨识任务。Softmax 能够将全连接层的输出向量 x 转换为向量 s。s 的每个元素被视为归属于某个类的概率，其值均在 0 到 1 之间且和为 1。Softmax 的表达式为：

$$s_{i}=\frac{\exp(x_{i})}{\sum_{j=1}^{n}\exp(x_{j})} \tag{2.17}$$

其中，s_i表示对应第 i 类的概率。

(7) 时间池化层

时间池化(temporal pooling)层是一种处理时间序列的池化层，是说话人识别领域常用的一种沿着时间维度进行池化的方法。对于变长的输入时间序列$x_1, x_2, \ldots, x_T$，经过运算后得到对其压缩的信息y。基本的时间池化层是对输入时间序列进行取平均运算，即：

$$y=\sum_{i=1}^{T}x_{i} \tag{2.18}$$

2.2.2 神经网络模型

说话人识别任务常用的神经网络模型有时延神经网络(time delay neural network, TDNN)[30]，卷积神经网络(convolutional network, CNN)等。下面对其简要介绍。

(1) TDNN

时延神经网络的基本结构如图 2.5 所示：

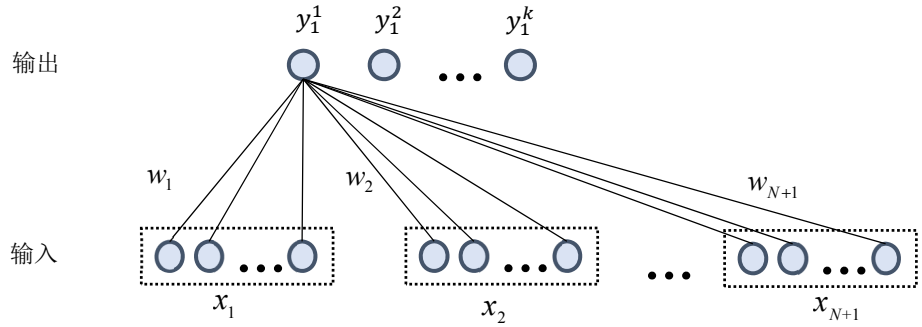


图 2.5 时延神经网络基本结构

Fig. 2.5 Basic structure of TDNN

若时延的时间为 N，则每次取连续 N+1 帧输入$x_1, x_2, \ldots, x_{N+1}$，用一组权重$w_1, w_2, \ldots, w_{N+1}$进行乘加操作即可得到对应时刻的一个隐层节点$y_1^i$，即：

$$y_1^i = w_1^T x_1 + w_2^T x_2 + \cdots + w_{N+1}^T x_{N+1} + b \tag{2.19}$$

式中x_i，y_i，w_i，$b \in R^{d\times 1}$。

(2) CNN

基本的卷积神经网络是由卷积层，池化层，激活函数以及 Softmax 层等基本结构组成的神经网络结构。卷积神经网络通过堆叠卷积层实现提取输入图片高阶语义的效果，在图像处理、自然语言处理以及语音领域得到了广泛的应用。卷积神经网络的基本结构如图 2.6 所示。

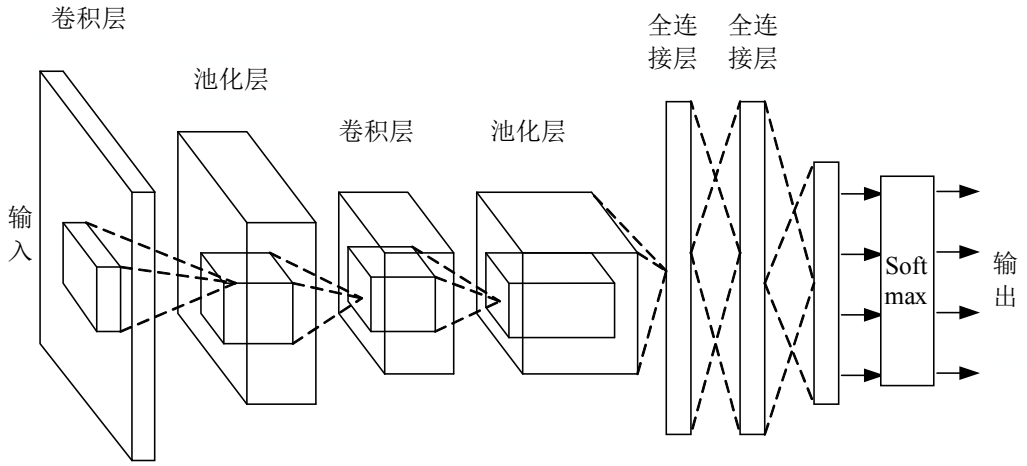


图 2.6 卷积神经网络基本结构

Fig. 2.6 Basic structure of convolutional neural network

ResNet[31]是常用的卷积神经网络结构，其主要特点是在卷积层之间加入残差连接结构，即短路连接(shortcut connection)。ResNet 通过构建多个具有残差结构的残差块结构，改善了训练深层网络出现的梯度消失问题，使得其能够训练相较于经典卷积神经网络模型更深的层数。

2.2.3 度量学习

度量学习(metric learning)[32,33]是说话人识别领域的常用方法。在实际应用中，经常需要判断数据之间的相似度。对于评价特定简单任务中数据间的相似度，可以直接通过距离函数来衡量。常用的几种距离有欧氏距离、内积或余弦相似度等。两个向量 a 和 b 的欧式距离表达式为：

$$d=\sqrt{(a-b)^T(a-b)} \tag{2.20}$$

余弦距离表达式为：

$$d=1-\cos(\theta)=1-\frac{a^T\cdot b}{\|a\|_2\cdot\|b\|_2} \tag{2.21}$$

对于更多较为抽象和复杂的任务，则不适合直接对原始形式的数据用前述的距离函数计算，比如计算两张人脸图像之间的匹配度或是两个说话人语音之间的相似度。这是由于图像像素或是语音表示组成复杂，很难直接进行相似度比较。度量学习则可以学习一个变换函数，将数据映射到一个向量空间，在新的向量空间中相似点之间的相似度较大，而非相似点则较小。基于深度学习的深度度量学习 (deep metric learning, DML)主要有以下两种学习方式：基于样本对的学习方式以及基于分类标签的学习方式。基于样本对的学习方式以样本对为输入，通过直接学习样本对之间的相似度来获得有效的特征表示。基于分类标签的度量学习方法则通过将每个样本分类到正确的类别来学习有效的特征表示，每个样本的标签在学习过程中参与损失计算。下面对基于样本对的学习方式三元组损失函数(Triplet loss)[34]以及基于分类标签的学习方式 AM-Softmax[35]进行介绍。

(1) Triplet loss

Triplet loss 的主要方法是构建三元组，其中包含了一个锚向量(Anchor) x_a，一个正样本x_p和一个负样本x_n。Triplet loss 的主要目标是通过训练使得对应三元组内锚向量与正样本的距离$d(x_a,x_p)$越来越小，锚向量与负样本的距离$d(x_a,x_n)$越来越大。其形式为：

$$L_{\text{Triplet}}=\max\left(0,m+d\left(x_a,x_p\right)-d\left(x_a,x_n\right)\right) \tag{2.22}$$

根据$d(x_a,x_n)-d(x_a,x_p)$与 m 的关系，可以将三元组分为简单三元组、困难三元组、半困难(semi-hard)三元组。当$d(x_a,x_n)-d(x_a,x_p)>m$时L_{Triplet}为 0, 此时其为简单三元组。当$d(x_a,x_n)-d(x_a,x_p)<0$时其为困难三元组，当$0\le d(x_a,x_n)-d(x_a,x_p)\le m$时其为半困难三元组。简单三元组说明度量学习结构对该三元组对应的样本已经有很好的区分能力此时其对应L_{Triplet}为 0 故对网络训练不产生影响，而困难三元组和半困难三元组则对网络训练均有影响，若构建的三元组包含太多的简单三元组则会使训练的效率较低，因此在训练神经网络时采用合理的策略构建三元组是重要的问题。目前来说训练时构建三元组的策略分为离线构建和在线构建的方式。离线构建方法将所有的训练数据输入到神经网络中，得到每一个训练样本的特征并挑选出困难三元组以及半困难三元组参与到对应 epoch 进行训练。在线构建则在训练的每个 batch 中选取困难三元组以及半困难三元组进行对应 batch 的训练。

(2) AM-Softmax

AM-Softmax 相较于 Softmax 加入了度量学习思想，能够使相同类别的特征间的类内距离减少，不同类别的特征间的类间距离增大，如图 2.7 所示。

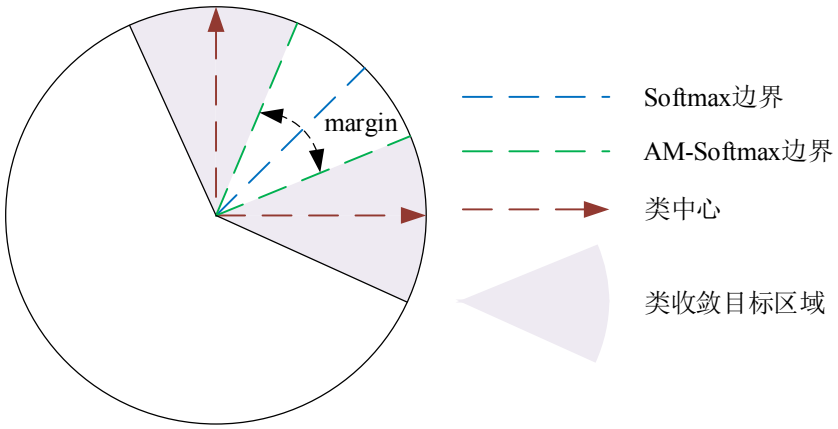


图 2.7 AM-Softmax 与 Softmax 对比

Fig. 2.7 Comparison between AM-Softmax and Softmax

Softmax 损失函数可以表示为：

$$L_{\text{Softmax}}=\frac{1}{N}\sum_{i=1}^{N}-\log p_i=\frac{1}{N}\sum_{i=1}^{N}-\log\frac{e^{f_{y_i}}}{\sum_{j=1}^{C}e^{f_j}} \tag{2.23}$$

其中$f_j = W_j^T x = \|W_j\|\|x\| cos\,\theta_j$，$x$表示最后一层全连接层的输入，$W_j^T x$即可得到第$j$个类别对应的值。不同于 Softmax，在 AM-Softmax 中先将W_j和x归一化，则$f_j = W_j^T x = \|W_j\|\|x\| cos\,\theta_j = cos(\theta_j)$，进而加入 margin 减少类内距离放大类间距离，即令$f_{y_i} = cos\left(\theta_{f_{y_i}}\right) - m$，最后通过系数$s$放大$f_{y_i}$，以增强表达能力。AM-Softmax 损失函数为：

$$L_{\text{AM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{y_i}\right)-m\right)}+\sum_{j\neq y_i}e^{s\cos\left(\theta_j\right)}} \tag{2.24}$$

2.2.4 注意力机制

人类往往能够关注到整体信息中更重要的局部信息从而形成判断，比如看到一张含有人像的画像会重点关注到人脸。模仿人类注意力的想法最早出现在计算机视觉领域，随着深度学习理论的发展，注意力机制得到了广泛研究[36,37]。

注意力机制的基本结构如式(2.25)。对于注意力机制的输入 F，得到相应的键值(Key, K)以及值(Value, V)，同时生成查询向量(Query, Q)，根据 Q 以及 K 的运算得到注意力系数，进而对 V 加权平均即可得到 F 经过注意力机制的输出结果。

$$\text{Attention}(Q, F) = \sum_{i=1}^{L}\text{Similarity}(Q, K_i)\times V_i \tag{2.25}$$

压缩与激励(Squeeze-and-Excitation, SE)模块[38]是受到广泛关注的通道注意力机制方法，其具有计算量小，使用灵活的特点，其基本结构如图 2.8 所示。

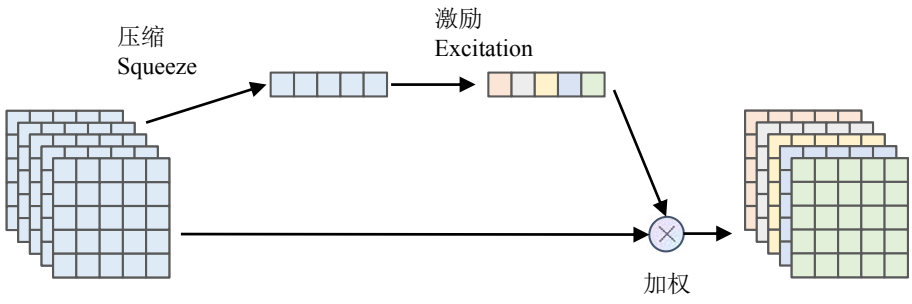


图 2.8 SE 模块基本结构

Fig. 2.8 Basic structure of SE module

SE 模块的基本操作是对于输入 H×W×C 的特征图，通过压缩和激励操作得到 1×1×C 的缩放因子，并与原特征图对应通道相乘即可得到 SE 的输出结果。其中，压缩操作通

过全局平均池化实现，得到 1×1×C 的压缩向量；激励操作则通过对压缩向量进行两次全连接层和激活函数的组合计算实现。

2.3 三种基准方法结构

本部分分别对三种基准方法 x-vector[23]、ResNetSE34L[39]、VGG-M[39]进行介绍。

2.3.1 x-vector 结构

基准方法 x-vector 的基本结构由 TDNN、统计池化层(statistics pooling)[23]组成。其基本结构如表 2.1 所示，其中 TP 为时间池化层，对于 x-vector 方法其为统计池化层。

表 2.1 x-vector 基本结构

Tab. 2.1 Basic structure of x-vector

层	层上下文	总上下文	输出
TDNN1	[t-2, t+2]	5	512×T
TDNN2	{t-2,t, t + 2}	9	512×T
TDNN3	{t-3,t, t + 3}	15	512×T
TDNN4	{t}	15	512×T
TDNN5	{t}	15	1500×T
TP	[0,T)	T	3000
FC1	{0}	T	512

2.3.2 ResNetSE34L 结构

基准方法 ResNetSE34L 是对标准 ResNet34 结构在说话人识别任务上调整得到的结构，如表 2.2 所示。

表 2.2 ResNetSE34L 基本结构

Tab. 2.2 Basic structure of ResNet34L

层	输出通道	块数	步长	输出 H×W
Conv1	16	1	[2,1]	D/2×T
Res1	16	3	[1,1]	D/2×T
Res2	32	4	[2,2]	D/4×T/2
Res3	64	6	[2,2]	D/8×T/4
Res4	128	3	[1,1]	D/8×T/4
Mean	128	1	-	1×T/4
TP	-	1	-	128
FC1	-	1	-	512

其中 conv1 为 7×7 卷积，Mean 沿着特征图 H 方向取均值，Res1 至 Res4 均为加入了 SE 模块的残差块。TP 为时间池化层。

2.3.3 VGG-M 结构

基准方法 VGG-M[39]对原始的 VGG 网络结构调整得到的结构，其基本结构如表 2.3 所示。其中 TAP 为时间平均池化，即直接沿时间维度取平均。Conv 为卷积层，Maxpool 为最大池化层。

表 2.3 VGG-M 基本结构

Tab. 2.3 Basic structure of VGG-M

层	核大小	输出通道	Stride	输出 H×W
Conv1	5×7	96	1×2	D×T/2
Maxpool	1×3	96	1×2	D×T/4
Conv2	5×5	256	2×2	D/2×T/8
Maxpool2	3×3	256	2×2	D/4×T/16
Conv3	3×3	384	1×1	D/4×T/16
Conv4	3×3	256	1×1	D/4×T/16
Conv5	3×3	256	1×1	D/4×T/16
Maxpool5	3×3	512	2×2	D/8×T/32
Conv6	4×1	512	1×1	D/8×T/32
TAP				512

2.4 评价指标和方法

对于说话人辨识任务，其为多分分类任务即系统得到待测样例所属的已知注册说话人分类结果，常使用准确率(accuracy)衡量系统的性能。准确率指分类正确的样本占总测试样本的比例。对于说话人确认任务，其为二分类任务即系统得到测试样例对是匹配还是冒认的判决结果，常用的评价指标是等错误率(equal error rate, EER)和最小检测代价(min detection cost function, minDCF)。

2.4.1 等错误率(EER)

等错误率是说话人确认任务的重要指标。介绍等错误率之前还需要介绍更为基本的错误接受率(false acception rate, FAR)，错误拒绝率(false rejection rate, FRR)以及 DET(detection error trade-off curve)曲线。

对于说话人确认任务来说，系统发生错误有两种情况：一种是匹配的说话人被认定为冒认者即错误拒绝，一种是冒认者被认定为匹配的说话人即错误接受。两种错误的实际含义大不相同。定义 FAR 为冒认者被认定为匹配的说话人的检测样例占总测试匹配样例的比例，FRR 为匹配的说话人被认定为冒认者的检测样例占总测试冒认样例的比例。如式(2.26)和式(2.27)。

$$P_{FAR} = \frac{N_{FA}}{N_{neg}} \tag{2.26}$$

$$P_{FRR} = \frac{N_{FR}}{N_{pos}} \tag{2.27}$$

式中P_{FAR}为错误接受率，P_{FRR}为错误拒绝率，N_{FA}为冒认者被认定为匹配的说话人的检测样例数，N_{FR}为匹配的说话人被认定为冒认者的检测样例数。N_{neg}为实际冒认测试样例数，N_{pos}为实际匹配测试样例数，实际总测试样例数为$N_{neg} + N_{pos}$。

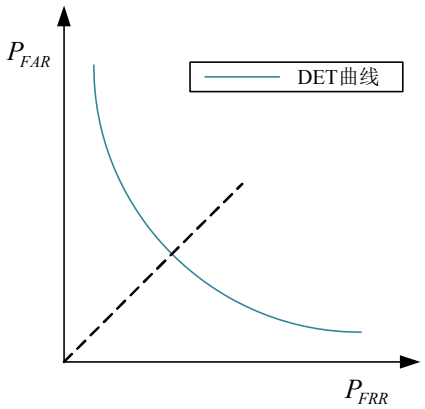


图 2.9 DET 曲线

Fig. 2.9 Detection error trade-off curve

P_{FAR}与P_{FRR}的取值范围均为 0 至 1 之间。实际测试中由于系统不同的阈值设置，两者往往有着不同的变化趋势，DET 曲线是以错误拒绝率为横轴，错误接受率为纵轴的曲线，能够反应不同阈值设置下的错误拒绝率和错误接受率变化，如图 2.9 所示。其中错误接受率与错误拒绝率相等时的比率即为等错误率，即对应图 2.9 中虚线与 DET 曲线的交点。

2.4.2 最小检测代价(minDCF)

除了等错误率以外，最小检测代价也是一种有效的评价指标。对于实际说话人确认系统而言，错误接受和错误拒绝的代价往往是不同的，最小检测代价则考虑到了这种情况。检测代价函数(detection cost function)定义为:

$$\mathrm{DCF} = C_{FRR} \cdot P_{FRR} \cdot P_{tar} + C_{FAR} \cdot P_{FAR} \cdot (1 - P_{tar}) \tag{2.28}$$

式中C_{FRR}为错误拒绝代价，C_{FAR}为错误接受代价。P_{tar}为实际匹配测试样例出现的先验概率。通过调整说话人确认系统的阈值，可得 minDCF 即 DCF 的最小值，即为最小检测代价。

2.5 本章小结

本章主要介绍了说话人识别的相关知识。首先介绍了语音处理的基础知识，包括语音的线性产生模型和语音的前端处理方法；其次介绍了深度学习的基本结构与模型、度量学习与注意力机制的基本方法；然后，对本文所用的三种说话人识别基准方法进行介绍；最后，介绍了说话人识别的评价指标和方法。

3 基于 TSCA-ResMBConv 的说话人识别方法

3.1 引言

基于深度学习的说话人识别方法，往往具有参数量大、复杂度高的特点，这限制了其应用场景和部署方式。减小模型复杂度是深度学习领域的重要研究方向。常见的减少模型复杂度的方式有使用轻量级卷积结构[40-42]、知识蒸馏[43,44]、模型量化[45]等。轻量级卷积即使用参数量和计算量较标准卷积更低的卷积方法，如 Xception[40]、ShuffleNet[41]、EfficientNet V2[42]等。知识蒸馏方法通过令小模型学习大模型使得小模型效果得到提升。模型量化则考虑在计算机的硬件设备中整型数据运算较浮点数更快的特点，将神经网络参数量化为特定范围内的整数。

提升说话人识别系统的识别性能是相关研究领域的重点。通过轻量型的通道注意力机制提升模型性能是受到广泛关注的方法，常用的结构有 SE 模块、CBAM 模块[46]、ECA 模块[47]、SA 模块[48]等。

本章应用轻量级卷积结构，提出了基于 TSCA-ResMBConv 结构的说话人识别模型。在该模型中，将反向瓶颈卷积结构引入说话人识别任务，大幅减小了模型的参数量；对说话人的语音特性进行分析，提出了时间分段通道注意力机制(time segment channel attention, TSCA)，通过建立特征图通道信息和时间段的关联，有效提升了模型的识别性能。实验结果表明，该方法能够以较小的参数量，实现较好的说话人识别性能。

3.2 时间分段通道注意力机制

不同说话人的个性特征可能和某段时间内一些有辨识性的频带上的能量分布有着较大的相关性，这主要体现在：

(1) 在某些时间段，说话人可能发出比其它时间段更有个性特点的语音。相关研究认为，在一段持续的语音中，某些音素可能体现出更多的个性信息[26]。

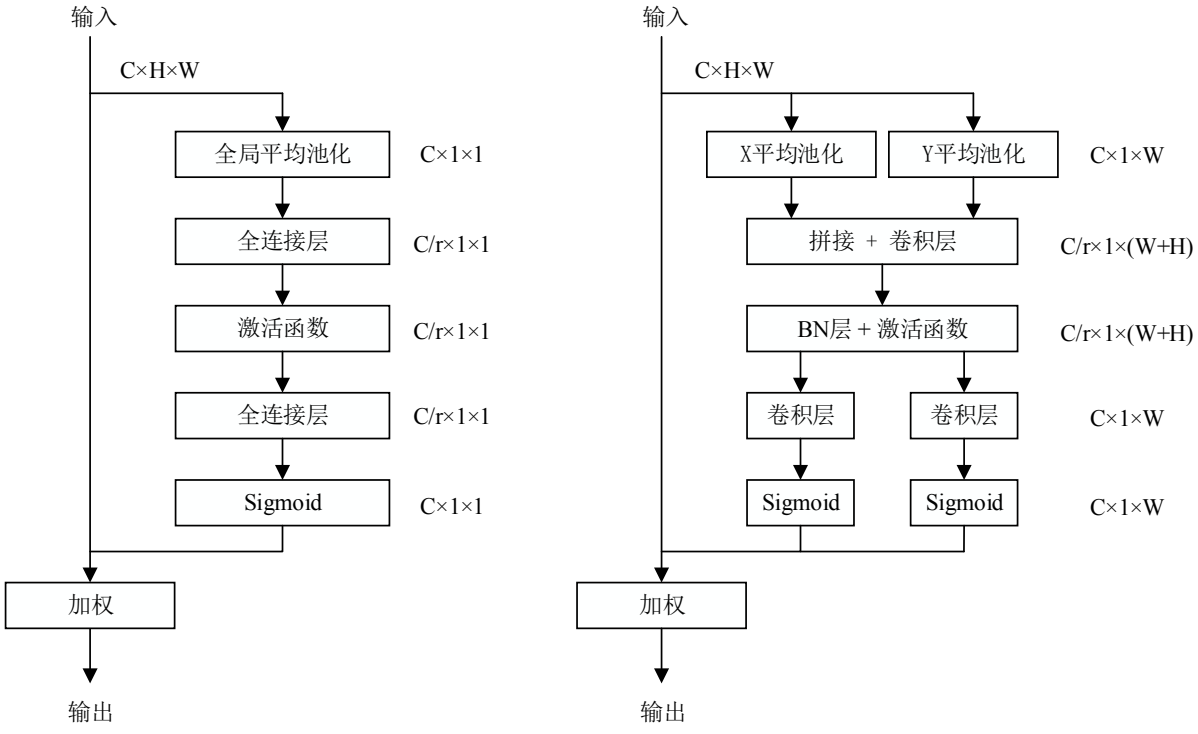
(2) 有的说话人声音较为嘶哑，则其声音在共振峰附近的子带上的能量相对较弱，在其它频点上的能量相对较多。而有的人声音穿透力强则意味着在共振峰附近的子带上的能量较大，其它频点上的能量较小。因此说话人在特定频带上的分布可能体现出相较于其它频带更多的个性信息。

因此对于说话人识别任务，实现考虑前述时间段和频带信息的注意力机制可能是有效的改进。尽管说话人识别任务上使用的特征并非原始的频带能量，但这种频带能量分布的不同也会直接影响到语音特征并使提取的语音特征在特定维度上的分布有所不同。

对于基于卷积的神经网络结构，尽管深层的特征图是更高级的语义信息，其仍包含了输入的原始特征的位置信息，例如全卷积网络[49]能够得到输入图片包含位置信息的分类标签实现语义分割任务。因此，对于说话人识别任务，特征图的位置信息可以反映出对应时间和对应频带的能量分布情况，通过引入对特征图位置信息的注意力机制实现使模型关注特定时间段特定频带的效果。本章基于 SE 模块和坐标注意力机制(coordinate attention, CA)[50]，提出了时间分段通道注意力机制(time segment channel attention, TSCA)从而有效利用了时间段和频带信息，提升了模型的识别性能。

3.2.1 坐标注意力机制基本原理

坐标注意力机制(coordinate attention, CA)[50]是一种加入了位置信息的通道注意力机制，是对 SE 模块的改进。图 3.1(a)表示 SE 模块的基本步骤，图 3.1(b)表示坐标注意力机制模块的基本步骤。



(a) SE 模块　　(b) CA 模块

图 3.1 SE 模块和 CA 模块结构对比

Fig. 3.1 Structure of SE module and CA module. (a) SE module. (b) CA module

对于输入特征图$X=\{x_1,x_2,\ldots,x_C\}\in R^{C\times H\times W}$，SE 模块在输入特征图各个通道上采用了全局平均池化得到$Z=\{z_1,z_2,\ldots,z_C\}\in R^{C\times 1}$，第 c 个通道的计算如式(3.1)所示。对于 SE 模块，该步骤丢失了特征图中的位置信息。

$$z_c=\frac{1}{H\times W}\sum_i^H\sum_j^W x_c(i,j) \tag{3.1}$$

相比较于 SE 模块，CA 模块并没有直接进行全局平均池化，而是对输入特征图 X 分别沿着 H 和 W 方向进行平均池化，从而得到两个分别保留了 H 和 W 方向信息的特征图$Z^H=\{z_1^H,z_2^H,\ldots,z_C^H\}\in R^{C\times H}$和$Z^W=\{z_1^W,z_2^W,\ldots,z_C^W\}\in R^{C\times W}$，其中第 c 个通道的输出z_c^H和z_c^W各个分量由式(3.2)和式(3.3)计算得到。这使得其相比较于 SE 模块能够建立不同的空间方向上的远距离依赖关系。

$$z_c^H(h)=\frac{1}{W}\sum_i^W x_c(h,i) \tag{3.2}$$

$$z_c^W(w)=\frac{1}{H}\sum_j^H x_c(j,w) \tag{3.3}$$

进一步，对于z_c^H和z_c^W由式(3.4)得到f:

$$f=\delta\left(F_1\left(\left[Z^H,Z^W\right]\right)\right) \tag{3.4}$$

式中[·,·]为拼接操作，F_1为 1×1 卷积函数，δ为激活函数，$f\in R^{C/r\times(H+W)}$。

将f沿空间方向拆分为f^H和f^W，并经过式(3.5)和式(3.6)可得加权向量$K^H=\{k_1^H,k_2^H,\ldots,k_C^H\}\in R^{C\times H}$和加权向量$K^W=\{k_1^W,k_2^W,\ldots,k_C^W\}\in R^{C\times W}$。

$$K^H=\delta\left(F_h(f^H)\right) \tag{3.5}$$

$$K^W=\delta\left(F_w(f^W)\right) \tag{3.6}$$

最后对X进行加权得到 CA 模块输出$Y=\{y_1,y_2,\ldots,y_C\}\in R^{C\times H\times W}$，计算$Y$第 c 个通道的计算方法如下：

$$y_c(i,j)=x_c(i,j)\times k_c^H(i)\times k_c^W(j) \tag{3.7}$$

3.2.2 时间分段通道注意力机制

本节分析了在说话人识别任务上直接使用坐标注意力机制的缺点，并在坐标注意力机制的基础上提出了在说话人识别任务上效率更高的时间分段通道注意力机制(time segment channel attention, TSCA)。在说话人识别任务上，直接使用坐标注意力机制有以下缺点：

(1)与 SE 模块相比，坐标注意力机制运算量有着显著的增加。SE 模块的主要计算量在于两层全连接层，可以理解为对 C×1×1 大小的特征图计算一次 1×1 卷积后得到 C/r×1×1 的特征图并进一步使用 1×1 卷积得到 C×1×1 大小的加权向量。而对于 CA 模块，主要计算量等同于对 C×1×(H+W)的特征图使用两次 1×1 卷积，尺寸先变为 C/r×1×(H+W)后变为 C×1×(H+W)。因此，若使用了同样的压缩系数 r，则 CA 模块运算量大约是 SE 模块运算量的(H+W)倍。

(2)坐标注意力机制在将Z^H和Z^W拼接后使用相同的 1×1 卷积核操作，而在将f拆分为f^H和f^W后，使用不同的 1×1 卷积核操作但参数量是相当的。而对于说话人识别任务来说，往往时间对应的维数 W 显著大于特征对应的维数 H。例如取 40 维特征，采用 10ms 的帧移分帧的 2s 语音，输入到卷积神经网络中在不改变尺寸的情况下，对应的 H 为 40，W 为 200，W 为 H 的 5 倍，同时为了尽可能保留时间序列的信息，较深层的网络中这一比值往往显著大于 5，最大能够达到几十。因此用同样的参数量处理 W 和 H 方向的信息，可能意味着对于 W 方向有效性较低，对于 H 方向有效性相对较高。

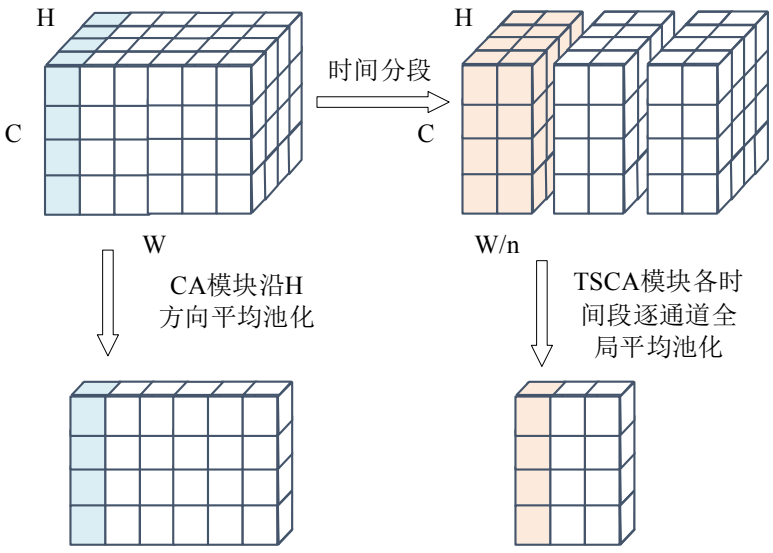


图 3.2 TSCA 模块与 CA 模块对比

Fig. 3.2 Difference between TSCA and CA

本节结合语音信号的特点，提出了时间分段通道注意力机制。该方法首先将原特征图按时间即 W 方向分段后，在每一段上进行逐通道全局平均池化得到 C×1×n 的特征图从而建立通道与时间段的关联，此外由于 H 方向信息较为重要且维数较小，对原特征图沿 W 方向平均池化，得到 C×H×1 的特征图，从而建立通道与 H 方向的关联。此时拼接后可以得到 C×1×(H+n)尺寸的特征图；此后与 CA 模块相似，计算式(3.4)并拆分后得到f^H和f^n。进一步经过式(3.5)和式(3.6)，可得加权向量K^H和K^n。为了仍然对原始的特征图加权，还需要对得到的K^n进行上采样，得到和原特征图尺寸对应的K^W，最后进行加权计算。TSCA 通道与时间段关联的方式，以及与 CA 的对比如图 3.2 所示。其中对时间分段后取各时间段特征图，逐通道的平均池化步骤等效于将原特征图沿 H 方向平均池化，得到 C×1×W 的特征图后，进一步用平均池化的方式沿时间方向将其压缩为 C×1×n 大小，得到压缩的时间信息，这意味着在同样参数量下，时间分段通道注意力机制在捕捉时间方向上的信息更加高效。

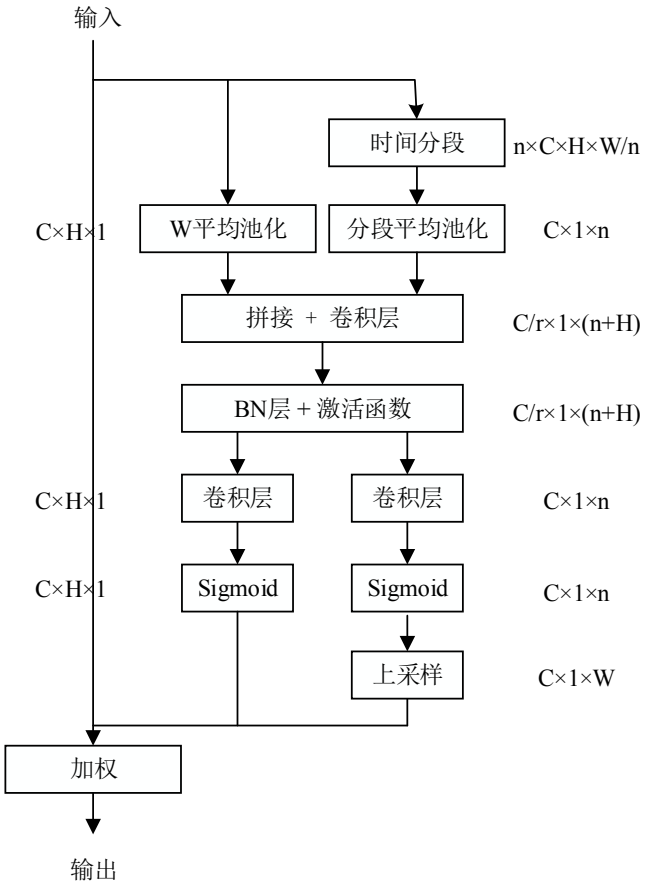


图 3.3 TSCA 模块

Fig. 3.3 basic pipeline of TSCA module

TSCA 模块总流程如图 3.3 所示。

由于卷积运算部分占据了主要的运算量，仅考虑卷积运算部分则 TSCA 与 CA 的运算量比值为:

$$\alpha = \frac{(H+n)}{H+W} \tag{3.8}$$

当 W>>H 时，$\alpha \approx n/W$。若 n 取 10，W 取 100 则 TSCA 计算量约为 CA 的 1/10。由此可见在说话人识别任务中 TSCA 计算量相较于 CA 有着明显的下降。

3.3 反向瓶颈结构

3.3.1 深度可分离卷积

深度可分离卷积(depthwise separable convolution)是一种降低常规卷积运算参数量的方法。深度可分离卷积分为深度卷积(depthwise convolution)和逐点卷积(pointwise convolution)两个过程。

深度卷积运算得到一个通道输出的基本过程如图 3.4 所示。对于输入$H \times W \times C_{in}$，标准卷积对应的卷积核尺寸为$k \times k \times C$，深度卷积对应的卷积核尺寸为$k \times k \times 1$。若输出为C_{out}个通道，则深度卷积步骤对应的参数量为$k \times k \times C_{out}$。由于深度卷积所得的特征图缺少了通道之间的信息交互，即不同通道之间互不影响，需要进一步加入通道信息的交互。而在深度卷积之后使用逐点卷积则可以弥补深度卷积的缺点。

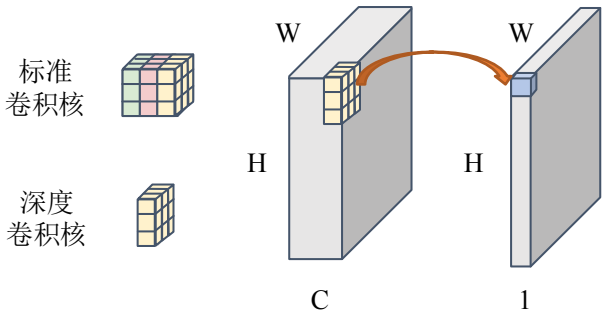


图 3.4 深度卷积基本过程

Fig. 3.4 Basic process of depthwise convolution

逐点卷积运算得到一个通道输出的基本过程如图 3.5 所示。逐点卷积是采用 1×1 大小卷积核的标准卷积操作，通过逐点卷积可以弥补深度卷积缺少通道间联系的缺点。若输出为C_{out}个通道，则逐点卷积步骤对应的参数量为$1 \times 1 \times C_{in} \times C_{out}$。

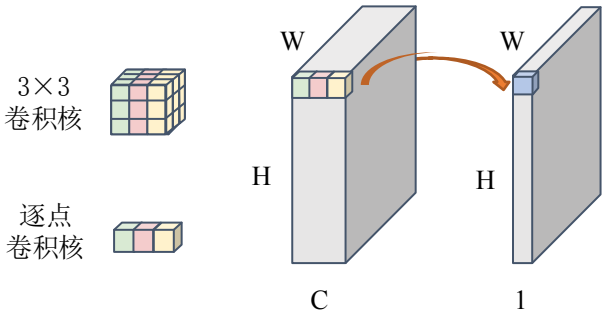


图 3.5 逐点卷积基本过程

Fig. 3.5 Basic process of pointwise convolution

深度可分离卷积的总参数量与标准卷积的比值为：

$$\frac{k\times k\times C_{out}+C_{in}\times C_{out}}{k\times k\times C_{in}\times C_{out}}\approx\frac{1}{k^2} \tag{3.9}$$

式中 k 为卷积核尺寸，对于常用的 3×3 卷积，深度可分离卷积的参数量仅为标准卷积的 1/9，因此采用深度可分离卷积能够显著减少参数量。

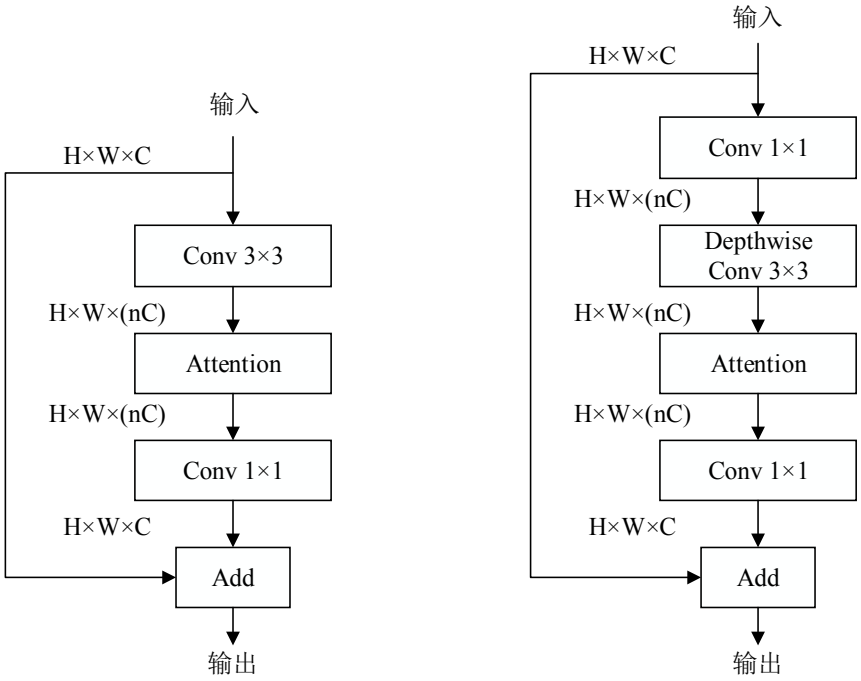
3.3.2 反向瓶颈层结构

反向瓶颈层的主要方法是构建先提升通道数后减少通道数的残差块，并采用深度可分离卷积替代一部分常规卷积以减少参数量和计算量。本部分使用了 EfficientNet V2[42] 所采用的两种反向瓶颈层结构，如图 3.6 所示。

在神经网络的浅层，此时 1×1 卷积虽然理论计算量仅为 3×3 卷积的 1/9，但由于特征图的通道数并不多，且神经网络的浅层的运算量本身占比较小因此实际的运算量差距并不大，因此在神经网络的浅层可以直接采用 3×3 常规卷积和 1×1 卷积实现反向瓶颈层结构，这使得模型相较于对应位置使用深度可分离卷积，小幅度增加计算量的同时提升一定的识别性能。此时的结构即为混合反向瓶颈层，如图 3.6(a)所示。而在更深层的神经网络，此时特征图的通道数较大，使用 1×1 卷积以及 3×3 深度可分离卷积和 1×1 卷积的组合可以相较于 3×3 常规卷积和 1×1 卷积的组合取得明显的减少参数量和计算量的效果，此时的模型结构如图 3.6(b)所示。此外，在 3×3 卷积和 1×1 卷积之间使用了高斯误差线性单元(gaussian error linear units, GELU)，其余位置不加入激活函数，每一个卷积层后均使用了 BN 层。GELU 函数的表达式为：

$$\begin{aligned}\mathrm{GELU}(x) &= xP(X \le x) = x\mathrm{F}\ (x) \\ &\approx 0.5x\left(1+\tanh\left[\sqrt{2/\pi}\left(x+0.044715x^3\right)\right]\right)\end{aligned} \tag{3.10}$$

式中$\Phi(x)$为标准高斯分布函数，$X\sim\mathcal{N}(0,1)$服从标准分布。相较于 ReLU 采用固定的门限 0 决定输入是否被激活，GELU 则更加平滑。



(a) Fused-MBConv　　(b) MBConv

图 3.6 反向瓶颈层基本结构

Fig. 3.6 Basic structures of inverted bottleneck convolution. (a) Structure of Fused-MBConv. (b) Structure of MBConv.

反向瓶颈层往往加入 SE 模块对不同通道的特征图按照重要程度进行相应的加权，即对应图 3.5 中的 Attention 操作，本章的主要结构则在对应位置使用了提出的 TSCA 模块。此外，经典的反向瓶颈层仅当卷积的步进设为(1,1)时加入残差连接结构，本章在使用的所有 Fused-MBConv 和 MBConv 结构中均加入了残差连接，若输入输出通道数不同，则对输入特征图经过 1×1 卷积层调整到输出的通道数并进行相加。

3.4 整体模型结构

本节对本章提出的 TSCA-ResMBConv 整体结构进行介绍。本章提出的 TSCA-ResMBConv 结构的整体设置如表 3.1 所示。

表 3.1 整体模型结构

Tab. 3.1 Structure of proposed model

层名	数量	步长	输出通道	输出尺寸
Conv1	1	[2,1]	16	D/2×T
F-MBConv1	3	[1,1]	16	D/2×T
F-MBConv2	4	[2,2]	32	D/4×T/2
MBConv1	6	[2,2]	64	D/8×T/8
MBConv2	3	[1,1]	128	D/8×T/4
Mean	1	-	128	1×T/4
Reshape	1			128×T/4
SP	1	-		256
FC	1			512

本章提出的 TSCA-ResMBConv 结构参考了 ResNetSE34L 的设置，在网络中对输入的时间维度进行了缩小，最终缩小到了原始时间长度的 1/4，这虽然损失了一定的时间信息但提高了模型的运算效率。网络输入的原始特征为 200 帧 40 维的对数梅尔频谱，即 40×200 的输入特征，通过表 3.1 所示的神经网络结构最终可以得到 512 维的说话人特征。在训练时基于表 3.1 的结构设置使用相应的损失函数。测试时首先对每一个说话人提取对应的说话人特征，其次对于提取得到的说话人特征使用余弦距离作为相似性的度量函数，最后选取合适的阈值计算相应的 EER 和 minDCF 指标。表中的步长设置为：对于 F-MBConv2， MBConv1 块均为其结构中第一个卷积层的步长，其余均为[1,1]。

模型主要部分如下：

(1) Conv1 对输入的特征处理并将特征图调整到 F-MBConv1 层输入的通道数，其为 7×7 的卷积操作。

(2) F-MBConv1 以及 F-MBConv2 为 3.3.2 节中的 Fuse-MBConv 结构构成的块，每个块的第一个结构输出通道数为输入通道数的两倍，其余块输入输出通道数相同。其中的 Attention 部分使用了本章提出的 TSCA 模块，从而高效地利用输入特征图的空间信息。

(3) MBConv1 和 MBConv2 即为 3.3.2 节的对应结构构成的块，每个块的第一个结构输出通道数为输入通道数的两倍，此外其余块输入输出通道数相同。该部分用于网络的较深层，能够有效减少参数量，提高模型的效率。

(4) Mean 和 Reshape 步骤认为特征图每个通道均提取出了对应时间的一种特征，直接对 H 方向平均池化获得该通道对应特征的表示，并通过 Reshape 操作将尺寸为 C×H×W 的三维的特征图调整为 D×T 尺寸的时间序列形式。

(5) SP 即统计池化层(statistic pooling)，模型使用统计池化层，将提取到的时间序列压缩成为语段级别的特征。

(6) FC 即全连接层，用于将前一层得到的向量维度转换为实验部分设定的说话人特征的维数。

3.5 实验部分

3.5.1 实验设置

实验部分选用了 VoxCeleb1[51]数据集。VoxCeleb1 数据集是牛津大学发布的大规模说话人识别数据集，共包含 1251 个说话人共计 153516 段语音。该数据集分为开发集和测试集，对于说话人确认任务，数据集被划分为包含 1211 个说话人共计 148642 段语音的开发集和包含 40 个说话人共计 4874 段语音的测试集。本实验采用了 VoxCeleb1 数据集的官方划分方式，使用对应开发集作为深度学习模型的训练集，使用测试集来衡量模型的效果。训练和测试样本均使用 2s 时长的语音段，分帧设置为窗长 25ms 帧移 10ms，基于 Torchaudio 工具包提取 40 维对数梅尔频谱作为输入特征。实验的软硬件环境为：64 位 Ubuntu16.04 系统、Pytorch 开源框架、NVIDIA GTX1080Ti 显卡×2、Intel i7-8700k 处理器。

为了便于比较，实验中除了 VGG-M 之外所有模型的时间池化层均使用统计池化，VGG-M 则使用原始的时间平均池化(temporal average pooling, TAP)方法[39]。所有实验均未使用数据增强方案。若对应实验有关设置未具体说明，且对应部分为本章提出方法，未说明部分均按照表 4.1 对应部分设置。实验中对应图 3.1 和图 3.3 中所有注意力机制中的 C/r 值均设为 8。除了第一部分损失函数实验，其余实验部分均使用 Triplet 损失函数，对应式(2.22)中的超参数 m 设为 0.1。实验设置均使用 Adam 优化器，初始学习率为 0.001，epoch 为 500。 不同模型训练时长在 12h 至 48h 之间。

本章 Triplet 损失函数的三元组构建策略如表 3.2 所示。记 batch size 为 N，每个 batch 均包含了 N/2 个说话人，每个说话人包含 2 个不同的语音作为锚向量和正样本。对第 i 个锚向量，计算其与其它类别对应的所有正样本的距离，找到距离最大的 10 个样本随机选取其中之一构成第 i 个三元组的负样本。该挑选负样本策略即表 3.2 中的 *Select*()，重复 N/2 次即可得到 N/2 组三元组。

表 3.2 三元组构建方法

Tab. 3.2 The method for triplet mining

输入: 一个批次B，对应共计 N 个样本，分为 N/2 组，每组对应同一个说话人的两个不同样本，即(a_i, p_i)。$B = \{(a_1, p_1), (a_2, p_2), \ldots, (a_{N/2}, p_{N/2})\}$

for $i = 1$ to $N/2$ **do**
  for $j = 1$ to $N/2$ **do**
    if $j \neq i$
      $d_j = D(a_i, p_j)$ # D 为计算距离函数
  end for
  $j = Select(d)$
  $n_i = b_j$ # n 为负样本
  $T_i = (a_i, p_i, n_i)$ # a 为锚样本，p 为正样本
end for

输出: $N/2$ 个三元组$\{T_1, T_2, ..., T_{N/2}\}$，$T_i = (a_i, p_i, n_i)$

实验均选取等错误率(EER)以及最小检测代价(minDCF)作为评价指标，其中 EER 以及 minDCF 越小，说明模型的性能越好，有效性越高。在如式(2.28)minDCF 的计算中，取C_{FRR}和C_{FAR}值为 1，P_{tar}设为 0.05，这表明$1 - P_{tar}$对应冒认测试样例的先验概率为 0.95 远大于P_{tar}即匹配测试样例的先验概率，这使得该指标相较于 EER 更优先考虑降低 FAR，对错误接受的情况更加敏感。

本节首先测试了不同损失函数对模型指标的影响，然后比较测试了不同组合设置的 Fused-MBConv 和 MBConv 结构的参数量和效果，接着分析了不同 TSCA 模块设置对模型的影响。最后，将本章提出的 TSCA-MBConv 结构与 x-vector、VGG-M、ResNetSE34L 三种基准方法进行比较。

3.5.2 实验结果

(1) 不同损失函数对模型性能的影响

本部分测试了不同损失函数在不同 batch size 下对模型性能的影响，实验模型的设置如 3.4 节所示。分别测试了基于 Triplet，GE2E[52]，Angleproto[39]，Prototype[53]，AAM-Softmax[54]以及 AM-Softmax[35]损失函数的方法，结果如表 3.3 所示。部分损失函数的超参数设置参考了文献[39]中的设置，主要设置如下：Triplet 的超参数 *m* 设为 0.1；GE2E、Prototype、Angleproto 的超参数 *M* 设为 3；AM-Softmax 的超参数 *m* 设为 0.1，*s* 设为 30；AAM-Softmax 的超参数 *m* 设为 0.1，*s* 设为 30。

表 3.3 不同损失函数对模型性能的影响

Tab. 3.3 Effect of different loss on model performance

损失函数	Batch Size	EER	minDCF
Triplet	320	4.43	0.317
Triplet	160	4.78	0.381
Angleproto	320	4.57	0.329
Angleproto	160	4.61	0.330
AM-Softmax	320	4.76	0.335
AM-Softmax	160	4.79	0.343
AAM-Softmax	320	4.59	0.229
AAM-Softmax	160	4.47	0.329
GE2E	320	4.65	0.344
GE2E	160	4.74	0.353
prototype	320	4.61	0.339
prototype	160	4.69	0.341

实验表明，在较大的 batch size 和足够多的 epoch 设置下，TSCA-ResMBConv 结构使用 Triplet 损失函数可以取得稍好的 EER 测试结果。而在较大 batch size 下 AAM-Softmax 损失函数对应的 minDCF 最小，结合本实验中P_{tar}较小，这意味者在优先减小 FAR 的情况时使用 AAM-Softmax 效果最好，而不考虑错误接受和错误拒绝的不同代价时，基于 Triplet 方法的 EER 更小效果最好。

此外，batch size 的大小对实验结果有着一定的影响，对 Triplet 损失函数影响最大，对 AM-Softmax 和 AAM-Softmax 影响则较小。较小的 batch size 不利于构建有效的三元组因而不利于基于样本对的 Triplet 损失函数。对于基于分类损失 AM-Softmax 和 AAM-Softmax，batch size 的设置对其影响相对较小，这可能是由于其能够通过标签信息使得相同说话人的不同样本产生全局的关联，此外相同的 epoch 设置下较小的 batch size 意味着更多的反向传播次数，这些因素一定程度减小了较小 batch size 信息的相对不足带来的影响。

(2) Fused-MBConv 和 MBConv 不同组合设置对模型性能和参数量的影响

本部分在 Fused-MBConv 和 MBConv 总的块数设为 4 的基础上，测试了两者不同块数设置对参数量和性能指标的影响。Fused-MBConv 块数越多则意味着参数量和计算量

越大。本部分实验旨在选取参数量和识别性能适中的组合设置。其实验结果如表 3.4 所示。

表 3.4 不同组合设置对模型性能的影响

Tab. 3.4 The effect of different settings on model performance

F-MBConv	MBConv	参数量(M)	EER	minDCF
[3,4,6,3]	[]	1.577	4.53	0.341
[3,4,6]	[3]	1.058	4.51	0.345
[3,4]	[6,3]	0.609	4.43	0.339
[3]	[4,6,3]	0.531	4.69	0.356
[]	[3,4,6,3]	0.512	5.18	0.391

实验结果表明，当 F-MBConv 块和 MBConv 块的组合分别设置为[3,4]和[6,3]时可以取得性能和参数量较好的平衡。该方案相较于 MBConv 块设为[3,4,6,3]即完全使用 MBConv 的设置，能够在仅增加 19%参数量的情况下减少 0.75 的 EER。尽管 F-MBConv 块设置为[3,4,6]以及[3,4,6,3]时参数量更大，但并未取得明显优于其设置为[3,4]时的性能表现。Fused-MBConv 和 MBConv 的方法兼具参数量和性能上的优点，

(3) 反向瓶颈层的放大倍数对模型性能的影响

本部分对如图 3.6 所示的反向瓶颈层通道放大倍数的影响进行实验测试，结果如表 3.5 所示：

表 3.5 不同放大设置对模型性能的影响

Tab. 3.5 The effect of different multiply settings on model performance

F-MBConv 放大设置	MBConv 放大设置	参数量(M)	EER	minDCF
[1,1]	[1,1]	0.378	5.18	0.384
[1,1]	[2,2]	0.607	4.69	0.358
[2,2]	[2,2]	0.609	4.43	0.339
[2,2]	[4,4]	0.989	4.39	0.329
[4,4]	[4,4]	1.13	4.31	0.325

表 3.5 表示反向瓶颈层放大倍数不同取值对模型识别性能的影响，其中使用含有两个元素的数组表示两个块中放大倍数的设置，如 MBConv 放大设置为[2,2]表明 MBConv1 和 MBConv2 中的反向瓶颈层放大倍数分别为 2 和 2。

实验结果表明，反向瓶颈层的放大倍数越大，每个反向瓶颈层的表达能力越强，因而模型的识别性能也越好，但也意味着参数量和内存资源占用的增加。当所有反向瓶颈层中的放大倍数均设为 2 时可以取得识别性能和参数量较好的平衡。

(4) TSCA 时间分段数 n 的影响

本部分测试了如图 3.3 所示的 TSCA 结构中的时间分段数 n 对于整体方法的性能影响。实验中各层的时间维度的长度均为 50 的倍数，因此将时间分段数分别设置为 1、2、5、10、50 测试其对应识别性能。其测试结果如表 3.6 所示。

表 3.6 TSCA 模块 n 的取值影响

Tab. 3.6 Effect of n in TSCA module

n 取值	参数量(M)	EER	minDCF
1	0.609	4.59	0.353
2	0.609	4.51	0.349
5	0.609	4.47	0.345
10	0.609	4.43	0.339
50	0.609	4.49	0.341

实验表明在分段数小于等于 10 时，时间分段数越大会使结果稍好，当分段数大于 10 后分段数的设置对性能影响并不显著。时间分段使得对应部分以时间段的形式理解特征图，而当分段数为 1 时则等效于未进行时间分段，因此损失了特征图在不同时间段的分布信息。

(5)TSCA 不同加入方式的影响

表 3.7 TSCA 不同位置对模型的影响

Tab. 3.7 Effect of different TSCA position settings

TSCA 加入方式	参数量(M)	EER	minDCF
[0,0,0,0]	0.608	4.81	0.362
[0,0,0,1]	0.608	4.61	0.353
[0,0,1,1]	0.609	4.52	0.341
[0,1,1,1]	0.609	4.50	0.343
[1,1,1,1]	0.609	4.43	0.339

表中以包含 4 个布尔元素的数组的形式表示 TSCA 在对应 4 个块中的加入情况，例如[0,0,1,1]表示 F-MBConv1 和 F-MBConv2 没有加入 TSCA 模块，MBConv1 和 MBConv2 加入了 TSCA 模块，其它设置以此类推。实验表明在所有块中均加入 TSCA 模块可以有效提升模型在测试集上的表现。

(6) TSCA 有效性测试

本部分对 TSCA 模块的有效性进行测试，比较了基准方法 ResNetSE34L 和本章方法使用不同注意力机制模块的性能指标。其中如图 3.3 所示的 TSCA 结构中的时间分段数 n 设为 10。实验结果如表 3.8 所示。

表 3.8 TSCA 模块有效性实验

Tab. 3.8 Experiment on effectiveness of TSCA module

基础结构	注意力机制	参数量(M)	EER	minDCF
ResNetSE34L	SE	1.48	5.22	0.389
ResNetSE34L	SA	1.47	5.05	0.364
ResNetSE34L	CBAM	1.50	4.92	0.358
ResNetSE34L	ECA	1.47	5.19	0.380
ResNetSE34L	CA	1.49	4.86	0.357
ResNetSE34L	**TSCA**	**1.49**	**4.89**	**0.351**
ResMBConv	-	0.560	4.83	0.369
ResMBConv	SA	0.561	4.53	0.349
ResMBConv	SE	0.592	4.67	0.351
ResMBConv	CBAM	0.592	4.47	0.343
ResMBConv	ECA	0.561	4.75	0.360
ResMBConv	CA	0.609	4.50	0.348
ResMBConv	**TSCA**	**0.609**	**4.43**	**0.339**

为了测试不同注意力模块的影响，本部分将 ResNetSE34L 和 TSCA-ResMBConv 结构中的原注意力机制模块分别替换为 SE 模块、SA 模块、CBAM 模块、ECA 模块、CA 模块、TSCA 模块中的不同模块。

实验表明 TSCA 模块相较于 SE 模块、SA 模块、ECA 模块有着更好的效果，与 CA 模块和 CBAM 模块相当。本部分 TSCA 模块的时间分段数为 10，对应部分计算量约为 CA 模块的 1/10，这表明 TSCA 模块具有一定的实用性和有效性。实验结果中，TSCA、CA、CBAM、SA 均取得相较于 SE、ECA 更好的识别性能，考虑到 TSCA、CA、CBAM、

SA 均在一定程度上结合了特征图的空间信息，这表明空间信息对于说话人识别具有重要的作用，因此考虑了空间信息的注意力机制优于未充分结合空间信息的注意力机制。本章使用的 Fuse-MBConv 和 MBConv 结构有效减少了参数量同时具有一定的有效性，TSCA-ResMBConv 结构在不加入 TSCA 模块时，即可取得相较于 ResNetSE34L 更好的测试结果；加入 TSCA 模块，可以进一步降低 0.4 的 EER 指标。

(7) 与基准方法的性能比较

表 3.9 汇总了本章提出的方法与基准方法以及 ECAPA-TDNN[25]的对比，其中所有方法均未使用数据增强。

表 3.9 本章方法与基准方法比较

Tab. 3.9 Comparison between different baseline models and models proposed

方法	参数量(M)	EER	minDCF
x-vector	4.25	5.15	0.393
VGG-M	4.09	6.91	0.501
ResNetSE34L	1.49	5.22	0.389
ECAPA-TDNN	7.08	4.41	0.315
Ours	**0.609**	**4.43**	**0.339**

实验结果表明，TSCA-ResMBConv 相较于基准方法，具有参数量更小，在测试集上识别性能更好的特点。相较于 x-vector 方法，TSCA-ResMBConv 降低了 86%的参数量，取得了降低 0.67 等错误率的更好测试表现。相较于 ResNetSE34L，降低了 59%的参数量，降低了 0.87 等错误率。相较于 VGG-M 方法，降低了 85%的参数量，降低了 2.47 的等错误率。此外本章提出方法在参数量远小于 ECAPA-TDNN 的情况下，取得了相近的识别性能。

3.6 本章小结

本章主要关注构建有一定识别性能的轻量级高效说话人识别模型，提出了 TSCA 模块，结合设计轻量化卷积神经网络的思路，设计了 ResMBConv 主体结构，进一步提出了 TSCA-ResMBConv 结构。本章首先介绍了坐标注意力机制的基本原理和 TSCA 模块的基本步骤，其次介绍了本章采用的反向瓶颈卷积的结构，然后对整体方法和设置进行介绍，最后通过实验对 TSCA-ResMBConv 结构的有效性，以及 TSCA 模块的有效性进行实验测试。实验表明本章提出的 TSCA-ResMBConv 结构在显著降低了基准方法的模型参数量的同时，在测试集上仍有较好的性能表现。此外经过测试本章提出的 TSCA 模

块相较于 SE 模块、ECA 模块、SA 模块有着更好的识别效果，相较于 CA 模块、CBAM 模块有着相当的识别效果。

4 基于自注意力机制与多任务学习的说话人识别方法

4.1 引言

提升模型识别性能是说话人识别领域重要的研究方向。使用多任务学习策略是提升模型性能的有效方法。Liu 等人提出了结合说话人识别和语音识别的多任务学习方法[55]，使得基于帧级别说话人特征的说话人识别方法有一定的识别性能提升。Jung 等人提出了基于关键词识别和说话人识别的多任务学习方法[56]，有效提升了关键词识别任务和说话人识别任务的识别性能。

时间池化层是提取语段级说话人特征的重要方法和改进方向。基于多任务学习以及时间池化层的特点，本章提出了一种结合基于自注意力机制的时间池化层以及多任务学习的改进方法，该整体方法能够替代基于语段级说话人特征的说话人识别方法中的时间池化方法，进一步提升对应模型的识别性能。本章首先对说话人识别领域的常用时间池化方法进行分析，提出了一种基于自注意力机制的时间池化方法，使其自注意力机制的查询向量由输入生成而非固定不变；接着提出了基于多任务学习的改进方法，该方法使得查询向量的构建更为有效，同时结合了 Triplet 和 AAM-Softmax 损失函数的特点，能够进一步提升模型的识别性能。

4.2 基于自注意力机制的时间池化方法

4.2.1 自注意力机制基础

自注意力机制[57]是在自然语言处理等领域受到关注的注意力机制。对于输入 $F=[f_1,f_2,...,f_{n_f}] \in R^{d_f \times n_f}$，自注意力机制直接从输入 F 计算得到对应的 Q。自注意力机制计算过程如下：

$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.1}$$

式中 $Q=[q_1,q_2,...,q_{n_f}] \in R^{d_q \times n_f}$，$K=[k_1,k_2,...,k_{n_f}] \in R^{d_k \times n_f}$，$V=[v_1,v_2,...,v_{n_f}] \in R^{d_v \times n_f}$，且均由 F 计算得到，即：

$$Q=W_Q \times F \tag{4.2}$$

$$K=W_K \times F \tag{4.3}$$

$$V = W_V \times F \tag{4.4}$$

4.2.2 经典时间池化方法分析

时间池化层(temporal pooling)是说话人识别领域的重要方法。记从原始音频经过前端处理提取得到的特征序列为$X = \{x_1, x_2, \ldots, x_{T'}\} \in R^{D' \times T'}$，经过说话人识别模型，可得说话人特征序列$H = \{□_1, □_2, \cdots, □_T\} \in R^{D \times T}$。为了进一步提取语段级别的说话人特征，需要进一步从说话人特征序列中得到压缩的信息 y。下面对统计池化(statistics pooling, SP)[23]、自注意池化 (self-attentive pooling, SAP)[58]和注意统计池化 (attentive statistics pooling, ASP)[59]三种时间池化的方法进行分析。

(1) SP 方法

统计池化的计算步骤为：

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x_t, \quad t = 1, \cdots, T \tag{4.5}$$

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \mu)^2}, \quad t = 1, \cdots, T \tag{4.6}$$

$$y_{SP} = [\mu, \sigma] \tag{4.7}$$

SP 直接对输入的时间序列进行求取均值和方差，并进行拼接从而得到池化结果。这使得 SP 具有计算量小的特点。然而无论输入如何变化，每一个时间序列的权重都是相同的，这使得不同时间的特征之间缺少关联。

(2) SAP 方法

SAP 是加入了自注意力思想的时间池化方法。其计算步骤为：

$$e_t = q^T \cdot f(W \cdot h_t + b_1) + b_2 \tag{4.8}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T}\exp(e_t)} \tag{4.9}$$

$$y_{SAP} = \sum_{t=1}^{T} \alpha_t h_t \tag{4.10}$$

式中f为激活函数。

相较于 SP，SAP 加入了注意力机制的思想。可能有一些时刻对应的特征向量包含的说话人个性信息更多因而重要程度更大，相应有一些时刻特征向量含有的个性信息可

能较少，SAP 对不同时刻对应的□$_t$加入自注意力机制,使得重要的帧被加强，相对信息量较少的帧则权重较小。SAP 方法对□$_t$直接进行加权求和，并没有包含输入序列的二阶统计信息即标准差。

(3) ASP 方法

ASP 相较于 SAP，加入了对输入序列二阶统计信息的考虑。ASP 同样包含了式(4.8)和式(4.9)的计算步骤，其余步骤为：

$$\tilde{\mu} = \sum_{t=1}^{T} \alpha_t h_t \tag{4.11}$$

$$\tilde{\sigma} = \sqrt{\sum_{t=1}^{T} \alpha_t h_t^T \cdot h_t - \tilde{\mu}^T \cdot \tilde{\mu}} \tag{4.12}$$

$$y_{ASP} = [\tilde{\mu}, \tilde{\sigma}] \tag{4.13}$$

ASP 相较于 SAP 加入了输入二阶统计量即标准差信息。ASP 同 SAP 的查询向量 q 均是固定而与输入无关的，这意味着其有效性有进一步改进的空间。

4.2.3 基于自注意力机制的时间池化方法

本章提出的自注意力机制方法，在 ASP 的基础上进一步使得查询向量随输入改变。对于输入$H = \{□_1, □_2, \cdots, □_T\} \in R^{D \times T}$，分别计算 q 和 K：

$$q = W_q \cdot \frac{1}{n} \sum_{t=1}^{T} h_t \tag{4.15}$$

$$k_t = W_k \cdot h_t \tag{4.16}$$

其中$q \in R^{D_q \times 1}$，K=$[k_1, k_2, \ldots, k_T] \in R^{D_k \times T}$。

通过可训练参数W_q，进一步提取出时间序列 H 的均值向量中的有效信息，并作为自注意力机制的查询向量 q。这使得其相比固定的查询向量，能够根据输入而改变。

令自注意力机制的 V 直接等于 H,通过式(4.17)至(4.21)可得自注意力机制最终输出。

$$e_t = \frac{q^T \cdot k_t}{\sqrt{d_q}} \tag{4.17}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_t)} \tag{4.18}$$

$$\hat{\mu} = \sum_{t=1}^{T} \alpha_t h_t \tag{4.19}$$

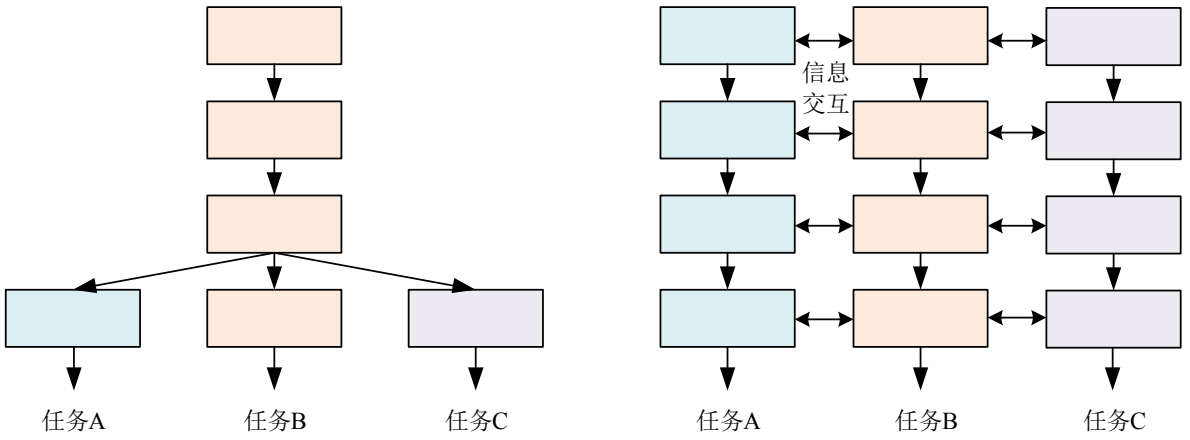
$$\hat{\sigma} = \sqrt{\sum_{t=1}^{T} \alpha_t h_t^T \cdot h_t - \hat{\mu}^T \cdot \hat{\mu}} \tag{4.20}$$

$$y = [\hat{\mu}, \hat{\sigma}] \tag{4.21}$$

4.3 多任务学习方法

4.3.1 多任务学习基础

多任务学习(Multi-Task Learning, MTL)[60,61]是一种机器学习方法，其目的是利用多个相关任务中包含的有效信息来帮助提高所有任务的泛化性能。基于深度学习的多任务学习方法分为硬参数共享(hard parameter sharing)和软参数共享(soft parameter sharing)方法，如图 4.1 所示。



(a) 硬参数共享　　(b) 软参数共享

图 4.1 多任务学习基本类型

Fig. 4.1 Different types of multi-task learning. (a) Basic structure of hard parameter sharing. (b) Basic structure of soft parameter sharing.

对于硬参数共享，不同任务间共用了部分相同的网络结构，如图 4.1(a)所示。硬参数共享由于先将不同任务的输入通入到相同的网络结构中，因而处理的往往是具有较强关联性的任务。硬参数共享的多任务学习方法，经过训练得到的模型参数可以分为共享参数和任务相关参数。对于软参数共享，不同任务使用不同的模型来处理，每个任务的

模型之间可以获取其他任务对应模型中的梯度、特征图等信息如图 4.1(b)所示。软共享机制相对于硬参数共享使用更为灵活，也可以处理关联性不大的任务，然而软共享机制的多任务学习方法往往意味着相较于硬参数共享更多的参数。

4.3.2 基于 Triplet 和 AAM-Softmax 的多任务学习方法

在说话人识别任务中，对于说话人确认和说话人辨识任务，其训练样本所属的说话人标签均是已知的，因此训练说话人确认任务的同时也可以用同样的样本训练说话人辨识任务。以 Triplet 为代表的基于样本对的损失函数，能够具体考虑到样本对级别的信息，通过优化每个三元组的方式间接使同一类别样本在特征空间上接近，但其没有充分利用标签中包含的信息，因而可能意味着有一定的优化空间。而基于分类损失函数则能使模型充分考虑标签信息，相较于 Triplet 损失函数其更多以类别的方式理解样本。本部分认为，分类损失函数能够使得样本分类到对应的类别，这意味着其考虑了该样本对应于各个类别的概率，因而其使得模型获得的说话人特征包含了更有全局性的类信息。本章使用 AAM-Softmax[54]函数充分利用类别信息，AAM-Softmax 表达式如式(4.22)所示。

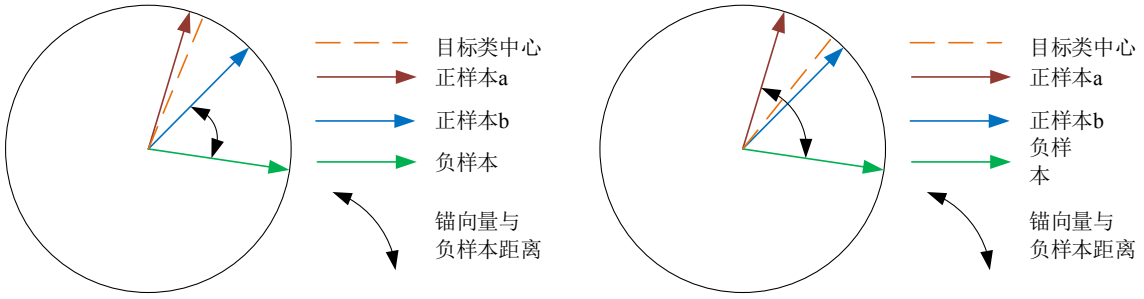
$$L_{\text{AAM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{4.22}$$

AAM-Softmax 直接将 AM-Softmax 表达式(2.24)中的$\left(cos\left(\theta_{y_i}\right)-m\right)$替换为$\left(cos\left(\theta_{y_i}+m\right)\right)$从而加快收敛速度和精度。本章通过自注意力机制的查询向量，在其后加入说话人辨识任务分支使用 AAM-Softmax 损失函数，而原说话人确认任务分支仍然使用 Triplet 损失函数，构成了本章多任务学习方案的损失函数策略。该方法将 Triplet 具体考虑样本对和 AAM-Softmax 考虑类别信息的特点结合起来，构建包含说话人确认任务以及基于分类损失函数的说话人辨识任务的损失函数L_{all}。该方法能够有效利用 AAM-Softmax 损失函数的特点，使得查询向量 q 中包含了一定程度的类别信息，从而提高查询向量 q 的有效性使得自注意力机制的有效性得到提高。本章采用硬参数共享的方式，将 Triplet 和 AAM-Softmax 结合计算最终的损失函数L_{all}：

$$L_{\text{all}} = L_{\text{triplet}} + \alpha L_{\text{AAM-Softmax}} \tag{4.23}$$

传统的基于 Triplet 损失函数的三元组构建方法，往往对于锚样本(Anchor)的选取没有明确的策略，用随机的方式从正样本中选取样本作为锚向量。本部分进一步对传统的三元组策略可能存在的不足进行探讨。在神经网络训练的过程中，对应每一个 batch，此时同一类别的不同正样本有效性可能有所不同。同一类别的大部分正样本可能倾向于靠

近某个向量，本部分假设其为目标类中心。若某个样本与目标类中心的距离更近，则意味着在该正样本有效性更高。若某个正样本距离目标类中心更远，则其可能有效性更低。因此在这一考虑下，传统的三元组构建策略可能会产生以下问题，如图 4.2 所示。



(a) 包含有效信息的三元组　　(b) 无有效信息的三元组

图 4.2 不同三元组示例

Fig. 4.2 Visualization of cases of triplet. (a) Visualization of effective triplet. (b) Visualization of ineffective triplet

图 4.2 表示同一组三元组以不同正样本为锚向量时的情形，其中若以正样本 b 作为锚向量，则此时$d(x_a,x_n)-d(x_a,x_p)<m$，这意味着该三元组没有被有效区分，具有一定的有效信息。若以正样本 a 为锚向量，则$d(x_a,x_n)-d(x_a,x_p)>m$，这表明该三元组没有被有效区分因而具有有效信息。若正样本 b 作为锚向量，此时正样本 a 距离目标类中心距离更近因而有效性更高，则构建该三元组使得无效信息参与反向传播。若正样本 a 作为锚向量，此时正样本 b 距离目标类中心距离更近因而有效性更高，则该三元组作为简单三元组使得有效信息被忽略。

考虑到前述传统三元组构建策略存在的问题，本部分构建了式(4.24)，以衡量说话人辨识任务支路所得的说话人特征的有效性，如下：

$$A(x)=\frac{W_{y_i}^{T}\cdot F(x)}{\left\|W_{y_i}\right\|\cdot\left\|F(x)\right\|} \tag{4.24}$$

式中$F(x)$为说话人辨识任务支路所得的说话人特征，其为计算 AAM-Softmax 损失函数前最后一层全连接层的输入向量。$W_{y_i}^{T}$为该全连接层参数矩阵的第y_i行，y_i为该样本在说话人辨识任务中所属的类别。$A(x)$即为$F(x)$与$W_{y_i}^{T}$的余弦相似度。

本部分进一步对第 3 章采用的传统构建三元组策略进行了微调，提出了基于多任务学习的三元组构建策略，其由输入 B 得到三元组 T 的步骤如表 4.1 表示。本部分由于采用了多任务学习策略，对同一说话人的两个不同正样本通过在说话人辨识任务支路的有

效性判断该正样本的有效性，选取有效性更高的正样本作为锚向量，具体步骤为计算式(4.24)取较大值对应的正样本作为锚向量，这是因为对应值越大，表明说话人辨识任务分支得到的说话人特征更接近对应类别的类中心因而更加有效。记 batch size 为 N，每个 batch 均包含了 N/2 个说话人，每个说话人包含 2 个不同的语音。具体步骤为首先基于前述的策略获得 N/2 组三元组对应的锚向量和正样本。其次对第 i 个锚向量，计算其与其它类别对应的所有样本的距离，找到距离最大的 10 个样本随机选取其中之一构成第 i 个三元组的负样本，该选择策略即表 4.1 中的$Select()$。重复前一步骤计算 N/2 次即可得到 N/2 组三元组。计算式(2.22)得到该三元组对应的$L_{triplet}$，以及计算每个三元组中三个样本的 AAM-Softmax 损失并取均值作为$L_{AAM-Softmax}$，即可计算式(4.23)的最终多任务学习的总体损失函数。

表 4.1 基于多任务学习的三元组构建方法

Tab. 4.1 The method for triplet mining based on multi-task learning

输入: N 个样本 $B=\{(a_1,b_1),(a_2,b_2),...,(a_{N/2},b_{N/2})\}$，分为 N/2 组，每组对应同一个说话人的两个不同样本，即 (a_i,b_i)。

for $i=1$ to $N/2$ **do**
  if $A(a_i)\geq A(b_i)$ #计算式(4.24)
    $p_i=b_i$
  else
    $p_i=a_i$
    $a_i=b_i$
  for $j=1$ to $N/2$ **do**
    if $j\neq i$
      $d_j=D(a_i,b_j)$ #D 为计算距离函数
  end for
  $j=Select(d)$
  $n_i=b_j$ #n 为负样本
  $T_i=(a_i,p_i,n_i)$ #a 为锚样本，p 为正样本
end for

输出: $N/2$ 个三元组 $\{T_1,T_2,...,T_{N/2}\}$，$T_i=(a_i,p_i,n_i)$

4.4 整体方法结构

记从原始音频经过前端处理提取得到的特征序列为$X = \{x_1, x_2, \ldots, x_{T'}\} \in R^{D' \times T'}$，经过说话人识别模型，可得说话人特征序列$H = \{□_1, □_2, \ldots, □_T\} \in R^{D \times T}$。本章提出方法的整体结构如图 4.3 所示。

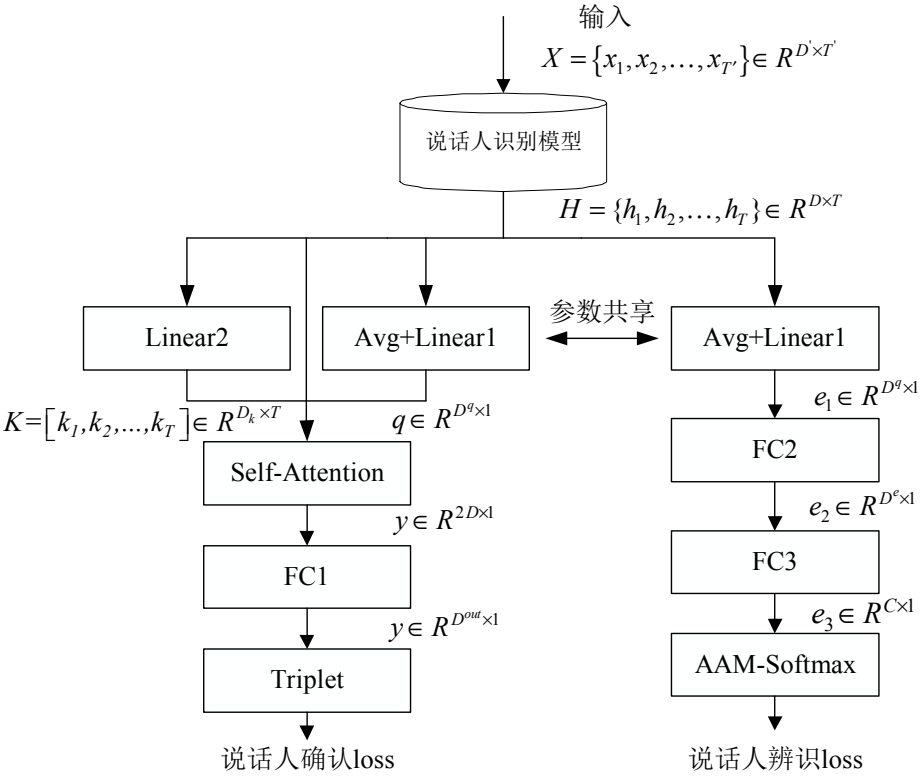


图 4.3 整体方法结构

Fig. 4.3 structure of proposed method

下面说明其主要步骤：

说话人识别模型指由输入X生成说话人特征序列H的结构，即说话人识别的主体神经网络结构。

Avg+Linear1 指进行式(4.15)的运算，通过该步骤可得自注意力机制的查询向量 q。其中说话人确认部分和说话人辨识部分均使用了步骤相同的 Avg+Linear1 操作，并共享参数。这使得 q 具有较多的类别信息，具有一定的有效性。

Linear2 指进行式(4.16)运算得到自注意力机制的键值 K。

Self-Attention 指进行式(4.17)至(4.21)的运算，得到自注意力机制的输出 y。

FC1，FC2，FC3 均为全连接层，在本章中 FC1 和 FC2 的输出均为 512 维。

方法总体损失函数如式(4.23)，三元组构建策略如表 4.1 所示。

4.5 实验部分

4.5.1 实验设置

本节的数据集选取以及软硬件环境与 3.5.1 节相同，本章提出方法如 4.4 节所示，测试时移除说话人辨识任务分支。本节首先实验测试了多任务损失函数的参数 α 的影响，其次探讨了本章提出方案采用不同设置对性能的影响，然后通过实验比较本章提出的方法与 SP、SAP、ASP 三种时间池化方法，分别在基准方法 x-vector，ResNetSE34L，以及第三章提出方法 TSCA-ResMBConv 上测试了使用经典时间池化方法以及本章方法的识别性能指标。最后对提出方法的有效性进行可视化分析。本部分训练和测试均使用两秒语音，分帧设置为窗长时间 25ms 帧移时间 10ms，基于 torchaudio 工具包提取 40 维对数梅尔频谱作为输入特征。本实验 batch size 设为 320，使用 Adam 优化器，初始学习率为 0.001 训练 500 个 epochs，训练时长为 12h 至 48h 之间。所有方法均未使用数据增强。

4.5.2 实验结果

(1) α 取值的影响测试

本部分测试如式(4.23)中 α 不同取值对模型性能的影响，如表 4.2 所示。

表 4.2 本章方法 α 取值的影响测试

Tab. 4.2 The impact of α in our method

模型	α	EER	minDCF
x-vector	0.1	4.70	0.365
x-vector	0.2	4.60	0.341
x-vector	0.3	4.67	0.343
x-vector	0.4	4.78	0.364
ResNetSE34L	0.1	4.77	0.362
ResNetSE34L	0.2	4.61	0.349
ResNetSE34L	0.3	4.62	0.345
ResNetSE34L	0.4	4.76	0.346
T-ResMBConv	0.1	3.87	0.271
T-ResMBConv	0.2	3.77	0.261
T-ResMBConv	0.3	3.82	0.282
T-ResMBConv	0.4	3.92	0.274

分别对基准方法 x-vector，ResNetSE34L 以及第三章提出的 TSCA-ResMBConv 模型，测试采用本章提出方法时，如式(4.23)中 α 不同取值对模型性能的影响。实验结果表明，当 α 取值为 0.2 时 EER 最小。α 较小时，AAM-Softmax 损失函数对应的说话人辨识任务对损失函数占比相对较小，此时说话人辨识任务更多起到辅助作用。

(2) 不同策略的性能比较

本部分探讨本章提出的方法在不同设置下的性能，主要包括:使用类似 SAP 的方式(此处记为方案 1)仅计算加权均值信息、同时使用加权均值和加权方差(即和图 4.3 完全相同，此处定为基准方案)的性能比较；使用第三章的三元组构建策略(此处记为方案 2)和使用本章三元组构建策略(即本部分基准方案)对性能的影响；使用多任务分支(即图 4.3 中的本部分基准方法)和不使用多任务分支仅保留说话人确认部分(即 α 取值为 0，此处记为方案 3)的影响。

表 4.3 不同方案性能比较

Tab. 4.3 Comparison between different methods

模型	方案	EER	minDCF
x-vector	基准	4.60	0.341
x-vector	方案 1	4.66	0.343
x-vector	方案 2	4.70	0.349
x-vector	方案 3	4.83	0.359
ResNetSE34L	基准	4.61	0.349
ResNetSE34L	方案 1	4.85	0.362
ResNetSE34L	方案 2	4.71	0.353
ResNetSE34L	方案 3	4.91	0.378
T-ResMBConv	基准	3.77	0.261
T-ResMBConv	方案 1	3.85	0.270
T-ResMBConv	方案 2	3.92	0.273
T-ResMBConv	方案 3	4.31	0.328

实验表明本章提出方法按照图 4.2 的设置时效果最好。分析认为方案 1 没有采用二阶信息因而相较于本部分的基准有一定的性能损失。方案 2 相较于本部分基准，没有充分利用说话人辨识任务的信息构建三元组，使得识别性能稍降。方案 3 没有使用多任务学习方法，因而缺少了多任务学习带来的增益。说话人辨识的多任务分支能够使得自注

意力机制更加有效，同时由于任务的关联性本身也会对硬参数共享的共用模型部分带来正面的增益。

(3) 基准方法使用不同方法的性能比较

本部分对比了基准方法 x-vector、ResNetSE34L、VGG-M 使用不同时间池化层以及本章方法的测试结果，如表 4.4 所示。

表 4.4 不同模型使用不同方案性能比较

Tab. 4.4 Comparison between models using different methods

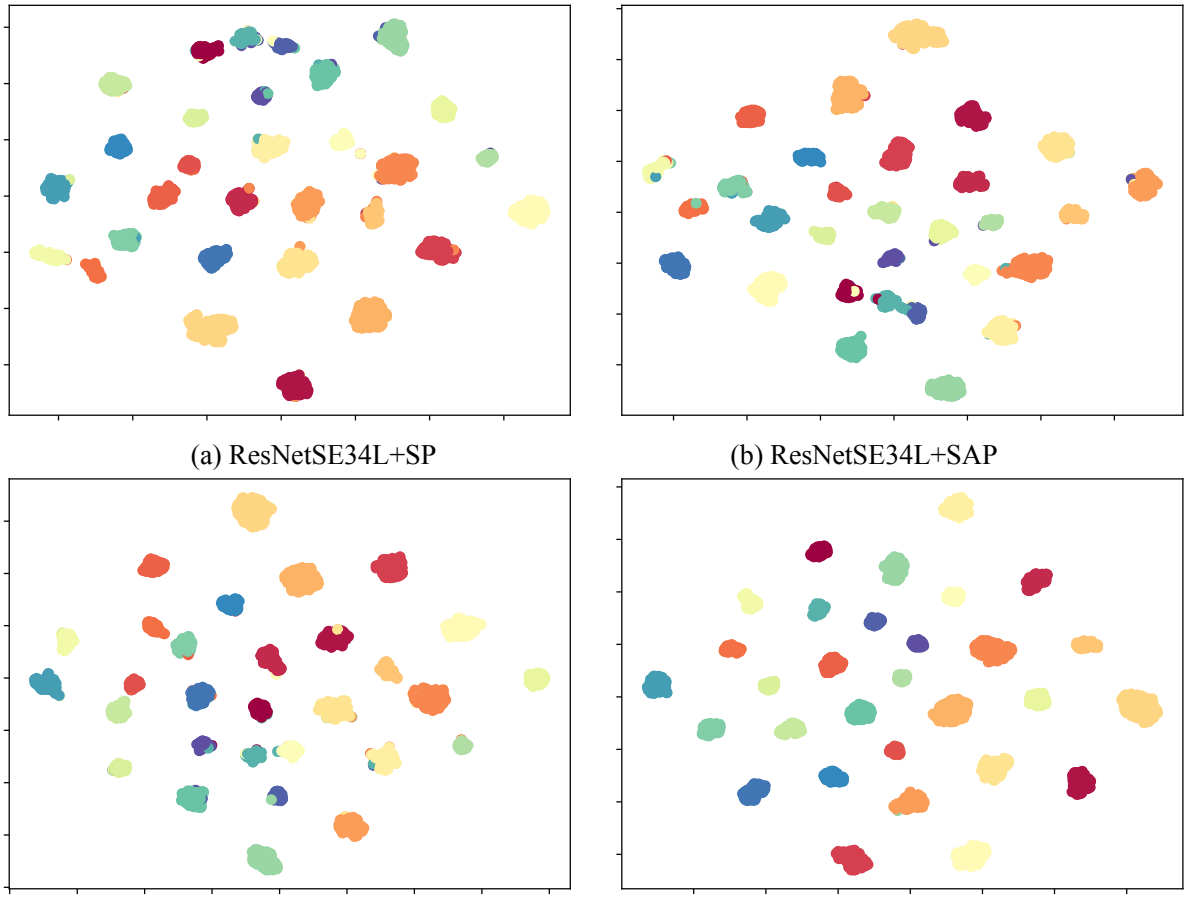
模型	方法	EER	minDCF
x-vector	SP	5.15	0.393
x-vector	SAP	5.07	0.375
x-vector	ASP	4.91	0.368
x-vector	**Ours**	**4.60**	**0.341**
ResNetSE34L	SP	5.22	0.389
ResNetSE34L	SAP	5.19	0.392
ResNetSE34L	ASP	4.95	0.369
ResNetSE34L	**Ours**	**4.61**	**0.349**
T-ResMBConv	SP	4.43	0.339
T-ResMBConv	SAP	4.39	0.328
T-ResMBConv	ASP	4.31	0.323
T-ResMBConv	**Ours**	**3.77**	**0.261**

实验结果表明，使用本章提出的时间池化方法与多任务学习方案，相较于仅使用 SP、SAP 或 ASP 的原方法有着显著的识别性能提升，对应的 EER 和 minDCF 均有较大幅度的降低。对于基准方法 x-vector，使用本章方法相较于 ASP 降低了 0.31 的 EER；对于基准方法 ResNetSE34L，使用本章方法相较于 ASP 降低了 0.34 的 EER；对于 TSCA-ResMBConv，使用本章方法相较于 ASP 降低了 0.54 的 EER。实际测试中 SAP 尽管加入了注意力机制，但由于其没有考虑输入时间序列的二阶信息，经过测试其对比 SP 提升并不显著。ASP 相较于 SAP，由于加入了输入序列的二阶信息，使得其相较于 SAP 有一定识别性能提升。

(4) 特征可视化

本部分基于基准方法 ResNetSE34L，对不同时间池化方法提取的特征进行可视化分析，选取 VoxCeleb1 训练集官方设置标签为 0 至 29 即前 30 个说话人的所有语音进行分

析，使用不同方法对应的模型提取 512 维特征后，基于 t-SNE 降维至 2 维进行可视化。可视化图中，不同说话人的语音用不同的颜色的点表示。可视化数据中共包含 30 个说话人共 3791 条语音，若数据被较好地区分，其可视化图中会形成 30 个簇，每一个簇表示对应的一个说话人，同时类内距离较小，类间距离较大。 可视化结果如图 4.4 所示：



(a) ResNetSE34L+SP　(b) ResNetSE34L+SAP

(c) ResNetSE34L+ASP　(d) ResNetSE34L+Ours

图 4.4 不同时间池化方法可视化分析

Fig. 4.4 Visualization of different temporal pooling methods. (a) Visualization of ResNetSE34+SP method. (b) Visualization of ResNetSE34+SAP method. (c) Visualization of ResNetSE34+ASP method. (d) Visualization of ResNetSE34+Ours method.

图 4.4(a)、图 4.4(b)、图 4.4(c)、图 4.4(d)分别表示在基准方法 ResNetSE34L 上使用 SP、SAP、ASP 以及本章方法的可视化结果。通过观察，不同方法均能一定程度上区分 30 个说话人的语音，基本形成相互分离的 30 个簇对应 30 个不同说话人。通过观察可视化图可以发现经典方法 SP、SAP、ASP 的可视化图中均存在一定对应簇中存在其它类别对应样本的状况，而对于图 4.4(d)对应的本章方法，该种情况则较少出现。通过观察图

4.4(d)对应的本章方法，其区分不同说话人的效果最好，主要表现在类间距离较大且较其它方法更加均匀，同时在一个簇中出现其它类别的说话人语音的情况最少。这表明本章方法能够有效地在训练集中区分不同的说话人对应的语音，能够更有效地学习训练集的数据从而获得表达能力更好的说话人个性特征。

4.6 本章小结

本章分析了三种说话人识别任务中的时间池化方法 SP、SAP 以及 ASP，对其潜在的特点和改进方向进行分析，提出了结合基于自注意力机制的时间池化层以及多任务学习策略的方法，该方法加入多任务学习机制，有效结合了 Triplet 和 AAM-Softmax 的特点，使得自注意力的查询向量含有更多全局类别信息因而相较于 ASP 方法更为有效。本章通过实验将本章方法应用于不同模型中，并与 SP，SAP 以及 ASP 进行比较，在数据集上取得了更好的测试结果，有效提升了说话人识别性能。

结 论

说话人识别即声纹识别是受到关注的一种生物信息识别方法，相较于人脸识别、虹膜识别和指纹识别等其它生物信息识别方法，具有成本较低，实现形式简单的特点，在智能家居、金融安全等领域有着广泛的应用前景。近年来，随着深度学习技术的发展，基于深度学习的说话人识别方法在相关识别性能指标上相较于传统方法得到了明显的提升。本文主要关注基于深度学习的文本无关说话人识别方法，提出了两点主要的结构和方法。本文主要研究成果总结如下：

(1) 提出了新颖高效的 TSCA-ResMBConv 模型用于说话人识别任务，该模型使用了低复杂度高效率的 Fused-MBConv 以及 MBConv 结构以实现降低模型参数量的效果，并使用本文提出的 TSCA 模块进一步增强模型的有效性。TSCA 模块是本文在 SE 模块和 CA 模块基础上提出的一种通道注意力机制，该方法能够有效建立特征图和时间段以及声学特征对应维度的关联，相比较于 SE 模块、ECA 模块、SA 模块具有更好的测试结果，相较于 CA 模块有着更高的运算效率和相当的测试结果。

(2) 提出了结合基于自注意力机制的时间池化层以及多任务学习策略的方法。该方法充分利用了说话人辨识任务和说话人确认任务的关联性,通过硬参数共享的思路，Triplet 加 AAM-Softmax 构成的多损失函数形式，以及自注意力机制的查询向量 q 的构造方式探索了多任务学习方法。相较于基于经典的时间池化层 ASP 的方案，本文的方法加入了多任务学习思想，使得自注意力机制查询向量更为有效，并构建了基于多任务学习的三元组，显著提升了原方案的识别性能，且具有一定的通用性。

本文对基于深度学习的识别方法进行了研究，取得了一定的成果。但是，受到时间和条件的限制，有些问题仍有优化改进的空间，如下所示:

(1) 本文提出的 TSCA-ResMBConv 方法参数量显著低于基准方法，其内存占用仍有优化改进的空间。基于 RepVGG 的思路，将所有残差连接重参数化可能对于进一步减少内存占用，提高模型效率和指标有着一定的参考价值。此外，使用神经架构搜索可能有助于找到效果更好的模型结构设计方法。本文提出的 TSCA 模块主要用于卷积神经网络结构中，而其对于基于 TDNN 的说话人识别方法的效果尚待研究。

(2) 本文没有探索基于更多语音任务的多任务学习方法。语音增强，语音合成，语音转换，语音分离等任务可能都和说话人识别有一定的关联性，探索更多的多任务组合可能也是说话人识别方法潜在的一种优化方向，具有一定的研究价值。

(3) 实际环境收集的语音往往经过编码，本文没有考虑不同编码算法对测试结果的影响。此外本文没有进行数据增强操作，没有探索数据增强对于说话人识别系统的潜在的增益。时域掩蔽，频域掩蔽，叠加噪声以及加入混响等数据增强方式可能对于考虑实际应用场景的说话人识别系统有着一定的正面效果。此外探索新的数据增强算法也是潜在的一种优化方向。

参考文献

[1] 韩纪庆. 语音信号处理 [M].北京:清华大学出版社,2019.
[2] BAI Z, ZHANG X. Speaker recognition based on deep learning: An overview[J]. Neural Networks,2021,140:65-99.
[3] FANT GUNNAR. Acoustic theory of speech production[M]. Walter de Gruyter, 1970.
[4] KERSTA L G. Voiceprint identification [J]. Nature, 1962, 196: 1253-1257.
[5] LUCK J E. Automatic speaker verification using cepstral measurements[J]. Journal of the Acoustical Society of America, 1969, 46 (4B) :1026-1032.
[6] ATAL B S. Automatic recognition of speakers from their voices [J]. Proceedings of the IEEE, 1976, 64 (4): 460-475.
[7] S. D, P. M. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1980, 28(4): 357-366.
[8] PRUZANSKY S. Pattern matching procedure for automatic talker recognition[J]. Journal of the Acoustical Society of America, 1963, 35 (3): 354-358.
[9] FURUI S. Cepstral analysis technique for automatic speaker verification[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1981, 29 (2): 254-272.
[10] BURTON D. Text-dependent speaker verification using vector quantization source coding [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1987, 35 (2): 133-143.
[11] SOONG F, ROSENBERG A, RABINER L, et al. Report: A vector quantization approach to speaker recognition [J]. AT & T technical Journal, 1987, 66 (2): 387-390.
[12] SOONG F, ROSENBERG A. On the use of instantaneous and transitional spectral information in speaker recognition [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1988, 36 (6): 871-879.
[13] NAIK J, NETSCH L, DODDINGTON G. Speaker verification over long distance telephone lines[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Glasgow, UK, 1989: 524-527.
[14] REYNOLDS D A. Speaker identification and verification using Gaussian mixture speaker models [J]. Speech Communication, 1995, 17 (1): 91-108.
[15] CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning,1995,20(3): 273-297.
[16] SCHMIDT M, GISH H. Speaker identification via support vector classifiers [C]. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, USA, 1996: 105-108.
[17] REYNOLDS D A, QUATIERI T F, DUNN R B. Speaker verification using adapted Gaussian mixture models [J]. Digital Signal Processing, 2000, 10(1): 19-41
[18] KENNY P, OUELLET P, DEHAK N, et al. A Study of Interspeaker Variability in Speaker Verification[J]. IEEE Transactions on Audio, Speech and Language Processing, 2008, 16(5): 980-988.
[19] KENNY P, BOULIANNE G, OUELLET P, et al. Speaker and session variability in GMM-based speaker verification[J]. IEEE Transactions on Audio Speech and Language Processing, 2007, 15(4):1448-1460.

[20] DEHAK N. Front-end factor analysis for speaker verification [J]. IEEE Transactions on Audio, Speech and Language Processing, 2011, 19 (4): 788-798.
[21] VARIANI E, VARIANI E, LEI X, MCDERMOTT E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 2014: 4052-4056.
[22] SNYDER D, GHAHREMANI P, POVEY D, et al. Deep neural network-based speaker embeddings for end-to-end speaker verification[C].2016 IEEE Spoken Language Technology Workshop, San Diego, CA, USA, 2016: 165-170.
[23] SNYDER D, GARCIA-ROMERO D, SELL G, et al. X-Vectors: Robust DNN embeddings for speaker recognition[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Alberta, Canada, 2018: 5329-5333.
[24] S. H G, M. M C, K. Z, et al. Res2Net: A new multi-scale backbone architecture[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021, 43(2): 652-662.
[25] DESPLANQUES B, THIENPONDT J, DEMUYNCK K. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification[J]. Interspeech, Shanghai, China, 2020:3830-3834.
[26] TAWARA N, OGAWA A, IWATA T, et al. Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, 2020: 6799-6803
[27] KIM S, PARK Y. Adaptive Convolutional Neural Network for Text-Independent Speaker Recognition[C]. Interspeech, 2021:66-70.
[28] 王琳琳. 说话人识别中的时变鲁棒性问题研究[D].北京:清华大学, 2013.
[29] IOFFE S, SZEGEDY C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning, Lille, France, 2015: 448-456.
[30] A. W, T. H, G. H, et al. Phoneme recognition using time-delay neural networks[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1989, 37(3): 328-339.
[31] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016: 770-778.
[32] MOUTAFIS P, LENG M, KAKADIARIS I A. An overview and empirical comparison of distance metric learning methods[J]. IEEE Transactions on Cybernetics. 2017, 47(3): 612-625.
[33] WANG H, LI Q, ZHANG D, et al. Key components of deep metric learning[C]. International Conference on Consumer Electronics and Computer Engineering, Guangzhou, China, 2022:648-651.
[34] F. SCHROFF, D. KALENICHENKO and J. PHILBIN. Facenet: A unified embedding for face recognition and clustering[C]. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015: 815-823.
[35] WANG H, WANG Y, ZHOU Z, et al. Cosface: Large margin cosine loss for deep face recognition[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018:5265-5274.
[36] XU K, BA J, KIROS R, et al. Show, attend and tell: Neural image caption generation with visual attention[J]. Computer Science, 2015:2048-2057.

[37] MNIH V, HEESS N, GRAVES A, et al. Recurrent models of visual attention[C]. Advances in neural information processing systems, Montreal, Canada, 2014: 2204-2212.
[38] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018: 7132-7141.
[39] CHUNG J S, HUH J, MUN S, et al. In defence of metric learning for speaker recognition[J]. 2020.
[40] CHOLLET F. Xception: deep learning with depthwise separable convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017: 1251-1258.
[41] MA N, ZHANG X, ZHENG H, et al. Shufflenet v2: Practical guidelines for efficient CNN architecture design[C]. European Conference on Computer Vision, Munich, Germany, 2018: 116-131.
[42] TAN M, LE Q. Efficientnetv2: Smaller models and faster training[C]. International Conference on Machine Learning. 2021:10096-10106.
[43] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. Computer Science, 2015, 14(7):38-39.
[44] GOU J, YU B, MAYBANK S J, et al. Knowledge distillation: A survey[J]. International Journal of Computer Vision. 2021, 129(6): 1789-1819.
[45] COURBARIAUX M, BENGIO Y, DAVID J. BinaryConnect: Training deep neural networks with binary weights during propagations[C]. International Conference on Neural Information Processing Systems, Montreal, Canada, 2015:3123-3131.
[46] WOO S, PARK J, LEE J, et al. CBAM: Convolutional block attention module [C]. European Conference on Computer Vision, Munich, Germany, 2018:3-19.
[47] WANG Q, WU B, ZHU P, et al. ECA-Net: efficient channel attention for deep convolutional neural networks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020:11531-11539.
[48] ZHANG Q, YANG Y. SA-Net: Shuffle attention for deep convolutional neural networks [C]. IEEE International Conference on Acoustics, Speech and Signal Processing, 2021:2235-2239.
[49] LONG J, SHELHAMER E, Darrell T. Fully convolutional networks for semantic segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(4):640-651.
[50] HOU Q, ZHOU D, FENG J. Coordinate attention for efficient mobile network design[C].IEEE Conference on Computer Vision and Pattern Recognition, 2021: 13713-13722.
[51] NAGRANI A, CHUNG J S, ZISSERMAN A. VoxCeleb: A large-scale speaker identification dataset[C]. Interspeech, Stockholm, Sweden, 2017:2616-2620.
[52] WAN L, WANG Q, PAPIR A, et al. Generalized end-to-end loss for speaker verification[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Alberta, Canada, 2018:4879-4883.
[53] SNELL J, SWERSKY K, ZEMEL R. Prototypical networks for few-shot learning[C]. Advances in Neural Information Processing Systems, Long Beach, California, USA, 2017: 4077-4087.
[54] DENG J, GUO J, XU E N, et al. ArcFace: Additive angular margin loss for deep face recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019: 4690-4699.
[55] LIU Y, HE L, LIU J, et al. Speaker Embedding Extraction with Phonetic Information[C]. Interspeech, Hyderabad, India, 2018:2247-2251.

[56] JUNG M, JUNG Y, GOO J, et al. Multi-task network for noise-robust keyword spotting and speaker verification using CTC-based soft VAD and global query attention[C]. Interspeech, Shanghai, China, 2020: 931-935
[57] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems, Red Hook, NY, USA, 2017: 6000-6010.
[58] CAI W, CHEN J, LI M. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system[C]. The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France, 2018: 74-81.
[59] OKABE K, KOSHINAKA T, SHINODA K. Attentive statistics pooling for deep speaker embedding[C]. Interspeech, Hyderabad, India, 2018: 2252-2256.
[60] CARUANA R. Multitask learning[J]. Machine Learning. 1997, 28(1): 41-75.
[61] VANDENHENDE S, GEORGOULIS S, Van GANSBEKE W, et al. Multi-task learning for dense prediction tasks: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021: 1.

攻读硕士学位期间发表学术论文情况

1　**徐启鹏**，殷福亮，陈喆. 基于神经网络的语音端点检测系统 V1.0.中国，计算机软件著作权，登记号：2022SR0283969.2022-2-28.

致 谢

时间过得太快，转瞬间研究生三年已过。也许没有来得及做更多的事情，但是经历的一切已弥足珍贵。

感谢我的研究生导师殷福亮老师。殷老师治学严谨，高屋建瓴，同时为人亲和而有社会责任感，是一位当之无愧的榜样人物。同时殷老师总是寓教于乐，激励大家学习之余，也传递乐观向上的生活态度，这不仅很接地气也使得我们实验室一直有很好的氛围。

感谢陈喆老师在科研和学业方面给与的指导。陈老师学术和工程能力极强，为人十分谦和。几年间不仅对我们专业知识有所教导，也“授人以渔”传授我们分析问题解决问题的方法，这些都是非常宝贵的收获。

感谢 A523 的梁羽贤、闫宇霆、邓翔宇、王国庆、闫钰、赵清颖同学，同窗勉励三年，相互学习，难得可贵。感谢崔行悦师姐、许宪法师兄给与的指导，你们给予的支持使我收获很大。感谢冯浩臻、孙博聪、吴为各位舍友，三年时光转瞬即逝感谢遇见你们。也感谢华为公司的合作人员张经理和殷经理，与华为合作的项目中我可能没有做到足够好，感谢诸位的宽容和耐心。

感谢大连理工大学，从大一到研三，在大工的七年是我人生中最为重要的时光，这里的学习氛围不时鼓励着我们勤学而上进。感谢家人背后的支持，如同涓涓细流。

最后，感谢提出宝贵意见的老师们及审阅本论文的专家学者们的辛苦付出。

大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：基于深度学习的说话人识别研究

作 者 签 名：徐志鹏 日期：2022 年 6 月 6 日

大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目：基于深度学习的说话人识别方法研究

作 者 签 名：徐志鹏 日期：2022 年 6 月 5 日

导 师 签 名：[illegible] 日期：2022 年 6 月 5 日