NO. 3ff46163d44483cc | 2026-02-05 08:02:21

- 题目： 演示论文
- 作者： 演示作者
- 检测所属单位： -

论文字符数： 30543 论文页数： 40 表格数量： 2 图片数量： 3

检测结果

39.42%
全文总相似比（复写率+自引率+他引率+专业术语）

相似结果详情

38.97%	0.0%	0.45%	0.0%
复写率	自引率	他引率	专业术语

其他指标

去除引用本人文献相似率：39.42% 去除专业术语相似率：39.42% 自写率：60.58%

典型相似文章： 无

检测范围 | 1989-01-01 ~ 2026-02-05

- 中文科技期刊论文全文数据库
- 博士/硕士学位论文全文数据库
- 外文特色文献数据全库
- 中文主要报纸全文数据库
- 中国专利特色数据库
- 中国主要会议论文特色数据库
- 古籍文献/图书资源
- IPUB原创作品
- 互联网数据资源/互联网文档资源
- 港澳台文献资源
- 年鉴资源
- 维普优先出版论文全文数据库

相似片段

相似片段：

471	465	6
总相似片段	相似片段	引用片段

检测来源：

期刊 14	综合 47	外文 0
博硕 121	互联网 289	

引用文献汇总

引用文献来源：6

序号	引用文献	引用字符数	引用率	来源
1	基于图像识别的医学影像大数据诊断系统的设计与实现	42	0.18%	互联网
2	基于人工智能的蓝牙耳机的音频播放方法及系统 吴伟鑫 - 2025	40	0.17%	综合
3	大学生毕业论文-18799	40	0.17%	互联网
4	基于分步定义式思维链的数学应用题自动求解方法研究 许永喆 - 2024	34	0.14%	博硕
5	从0到1，揭开目标检测的神秘面纱 - CSDN博客	20	0.08%	互联网
6	融合多模态数据的旅游情感轨迹建模及其时空变化模式研究——以芜湖方特主题公园为例 全宗鑫 - 2024	12	0.05%	博硕

相似文献汇总 (当前只展示10条数据,全部详情请查看片段对照报告)

相似文献来源：444

序号	相似文献	相似字符数	相似率	来源
1	基于阅读理解和图像描述生成的船舶领域问答系统研究 钟家国 - 2023	220	0.92%	博硕
2	一种基于深度学习的矿井设备故障智能诊断预测方法及系统 聂云辉;齐飞龙;程欣;王辉;姜敬敬;马锦艳;许良玉;王贯 - 2025	218	0.91%	综合
3	基于工业时序数据的神经网络模型评价与质量预测 刘来泽 - 2024	191	0.80%	博硕
4	基于人工智能的企业信用数据异常检测方法及装置 闫小良;崔琦 - 2025	148	0.62%	综合
5	多类别交叉熵损失函数的作用 - CSDN文库	138	0.58%	互联网
6	基于深度学习的MOSFET寿命预测方法研究	135	0.56%	博硕

	孟宪灼 - 2025			
7	小样本条件下的轴承跨域故障诊断方法研究 张洪铂 - 2025	132	0.55%	博硕
8	融合目标检测技术的多分类焊缝检测研究 李成斌 - 2025	131	0.55%	博硕
9	基于GNSS轨迹数据深度挖掘的农机作业行为识别研究 赵喜缘 - 2025	124	0.52%	博硕
10	大学生毕业论文-134289	115	0.48%	互联网

文字标注

■ 自写片段 ■ 复写片段 ■ 引用片段 ■ 专业术语 ■ 自引片段

本 科 生 毕 业 论 文

基于长短期记忆网络（LSTM）的音频分类算法设计与实现

学 院：

专 业：

班 级：

学 号：

指导教师：

职称（或学位）：

2026年 2 月

Undergraduate Graduation Thesis

Design and Implementation of Audio Classification Algorithm Based on Long Short-Term Memory Network (LSTM)

College:

Major:

Class:

Student ID:

Advisor:

Title (or Degree):

毕业设计（论文）原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全知晓本声明的法律后果由本人承担。

学位论文作者签名：

日期：2023 年 5 月 6 日

学位论文使用授权书

本人同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于 ：

公开论文

内部论文，保密□1年/□2年/□3年，过保密期后适用本授权书。

□秘密论文，保密 年（不超过10年），过保密期后适用本授权书。

□机密论文，保密 年（不超过20年），过保密期后适用本授权书。

（请在以上方框内选择打“ √ ”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

基于长短期记忆网络（LSTM）的音频分类算法设计与实现

摘 要

音频分类在音频处理领域具有重要地位，广泛应用于音乐检索、语音识别、声学监测等多个场景。传统方法通常依赖手工提取特征，如梅尔频率倒谱系数（MFCC）、频谱质心等，这类方法在特征提取环节需要较多人工干预，难以全面捕捉音频信号的复杂特性。同时，浅层机器学习模型如支持向量机（SVM）、决策树等在处理音频时序信息时表达能力有限，分类性能往往无法满足实际应用需求。

针对上述问题，本文深入研究基于长短期记忆网络（Long Short-Term Memory, LSTM）的音频分类方法。LSTM 作为一种特殊的循环神经网络，能够有效处理时间序列数据中的长期依赖问题，非常适合音频这种具有时序特性的数据。本文首先对音频数据进行预处理，运用 MFCC 特征提取技术，将原始音频信号转换为适合模型输入的特征向量，在保留音频关键特征的同时，降低数据维度，提高后续模型处理效率。随后，构建多层 LSTM 网络模型，充分利用 LSTM 对时序信息的记忆和处理能力，自动学习音频特征中的复杂模式和长期依赖关系，以实现对音频类别的准确判断。为了防止模型过拟合，在模型训练过程中引入 Dropout 正则化策略，随机丢弃部分神经元，增强模型的泛化能力，提升模型在未知数据上的表现。

通过在公开音频数据集上的实验，并与传统音频分类模型进行对比，结果表明基于 LSTM 的音频分类模型在准确率、召回率等评价指标上均有显著提升，展现出对音频数据更好的分类性能和适应性。本文的研究成果不仅丰富了音频分类领域的技术方法，也为相关实际应用提供了有效的技术支持和参考，推动了音频分类技术在更广泛场景中的应用与发展。

关键词：音频分类MFCC 特征提取

Design and Implementation of Audio Classification Algorithm Based on Long Short-Term Memory Network (LSTM)

Abstract

Audio classification plays a pivotal role in the field of audio signal processing, with widespread applications in music retrieval, speech recognition, acoustic monitoring, and other scenarios. Traditional methods typically rely on handcrafted features, such as Mel-Frequency Cepstral Coefficients (MFCCs) and spectral centroid. However, these methods require extensive manual intervention in the feature extraction stage, making it difficult to fully capture the complex characteristics of audio signals. Meanwhile, shallow machine learning models, such as Support Vector Machines (SVM) and decision trees, have limited expressive power when handling the temporal information of audio, and their classification performance often fails to meet the demands of practical applications.

To address these issues, this paper conducts an in-depth study on an audio classification method based on the Long Short-Term Memory (LSTM) network. As a specialized type of Recurrent Neural Network (RNN), LSTM effectively addresses the problem of long-term dependencies in time-series data, making it highly suitable for audio data which exhibits strong sequential characteristics. Firstly, this paper preprocesses the audio data and employs MFCC feature extraction technology to convert the raw audio signals into feature vectors suitable for model input. This step preserves the key features of the audio while reducing data dimensionality and improving the efficiency of subsequent model processing. Subsequently, a multi-layer LSTM network model is constructed. By fully leveraging LSTM's ability to memorize and process temporal information, the model automatically learns complex patterns and long-term dependencies within the audio features, thereby enabling accurate audio classification. To prevent model overfitting, a Dropout regularization strategy is introduced during training, which randomly deactivates a portion of neurons to enhance the model's generalization ability and improve its performance on unseen data.

Experiments conducted on a public audio dataset, in comparison with traditional audio classification models, demonstrate that the proposed LSTM-based model achieves significant improvements in evaluation metrics such as accuracy and recall, exhibiting superior classification performance and adaptability to audio data. The findings of this study not only enrich the technical methodologies in the field of audio classification but also provide effective technical support and reference for related practical applications, thereby promoting the application and development of audio classification technology in a broader range of scenarios.

Keywords: Audio Classification; MFCC; Feature Extraction; Deep Learning

目 录

摘 要 1
Abstract 1
目 录 1
1.引言 1
1.1 研究背景与意义 1
1.2 国内外研究现状 2
1.3 研究内容与技术路线 2
1.1 论文组织结构 3
2.相关理论基础 4
2.1 循环神经网络（RNN）及其局限性 4
2.2 LSTM 长短期记忆网络核心原理 6
2.2.1 LSTM 的结构组成 6
2.2.2 门控机制的工作流程 7
2.2.3 LSTM 的优势特性 8
2.3 音频分类的关键技术 9
2.3.1 音频信号预处理 9
2.3.2 MFCC 特征提取原理与实现 11
3.数据集 12
3.1 数据集选择与预处理 12
3.1.1 实验数据集构建 12
3.1.2 数据预处理流程 13
3.2 LSTM 分类模型架构设计 14
3.2.1 模型整体结构 14
3.2.2 模型超参数设置 16
3.3 模型训练策略 17
4. 实验与结果分析 18
4.1 实验环境搭建 18
4.1.1 硬件环境 18
4.1.2 软件环境 19

4.2 实验方案设计 ····· 19
4.2.1 评估指标选择 ····· 19
4.2.2 对比实验设置 ····· 20
4.3 实验结果与分析 ····· 21
4.3.1 模型训练过程分析 ····· 21
4.3.2 分类结果量化分析 ····· 22
4.3.3 对比实验结果分析 ····· 23
5 模型优化与改进方向 ····· 25
5.1 基于混合模型的优化尝试 ····· 25
5.1.1 LSTM-Transformer 融合模型设计 ····· 25
5.1.2 混合模型实验验证 ····· 25
5.2 正则化策略优化 ····· 26
5.3 轻量化模型设计展望 ····· 27
6.结论与展望 ····· 28
6.1 研究结论 ····· 28
6.2 未来研究展望 ····· 错误！未定义书签。
参考文献 ····· 1
致谢 ····· 1

1.引言

1.1 研究背景与意义

随着信息技术的飞速发展，数字音频技术在过去几十年间取得了长足进步，数字音频资源呈爆炸式增长。从日常生活中的音乐、广播，到专业领域的语音通信、声学监测等，音频数据无处不在。面对海量的音频数据，如何高效、准确地对其进行分类和管理，成为了亟待解决的问题。自动音频分类技术应运而生，它旨在将音频信号按照特定的类别进行划分，如音乐流派识别、人声与合成声区分、乐器音色辨识等。这项技术在多个领域有着广泛的应用前景，能够极大地提高音频数据处理的效率和准确性，为人们的生活和工作带来便利。

传统的音频分类方法主要依赖于手工设计的特征，如支持向量机（SVM）常与梅尔频率倒谱系数（MFCC）结合，高斯混合模型（GMM）利用频谱质心等特征进行分类。然而，这些方法存在明显的局限性。手工设计特征需要大量的人工经验和专业知识，且难以全面捕捉音频信号的复杂特征，泛化能力较弱，在面对新的音频场景或数据分布变化时，分类性能往往会大幅下降。[1]

近年来，深度学习技术的兴起为音频分类带来了新的解决方案。深度学习模型能够自动从数据中学习特征，避

免了手工特征设计的繁琐过程，且在表达能力和学习能力上具有明显优势。长短期记忆网络（Long Short-Term Memory，LSTM）作为循环神经网络（RNN）的一种改进模型，能够有效解决 RNN 在处理长序列数据时遇到的梯度消失问题，特别适合捕捉音频信号这种具有时序依赖关系的数据特征。通过 LSTM 模型，音频分类系统可以更好地学习音频信号中的长期依赖信息，从而实现更准确的分类。

本研究旨在深入探索基于 LSTM 算法的音频分类方法，通过改进特征提取和模型架构，提高音频分类的准确性和泛化能力。这不仅有助于丰富音频分类领域的理论研究，也为实际应用提供了更有效的技术支持，如提升智能语音助手对不同语音指令的识别准确率、优化音乐推荐系统的推荐效果等，具有重要的理论与实用价值。

1.2 国内外研究现状

在音频分类领域，基于深度学习的方法已成为国内外研究的热点。国外诸多研究团队在该领域取得了显著成果。例如，在音乐流派分类方面，一些学者利用 LSTM 网络对音频的梅尔频谱特征进行学习，实现了对多种音乐流派的准确分类。他们通过构建多层 LSTM 模型，充分挖掘音频信号在时间维度上的长期依赖关系，相较于传统方法，分类准确率有了明显提升。在语音情感识别中，双向 LSTM 被广泛应用，它能够同时考虑语音信号的前向和后向信息，更好地捕捉情感表达中的时序特征，从而提高情感分类的精度。

国内的研究也紧跟国际步伐，在音频分类技术上不断创新。有研究团队提出结合注意力机制的 LSTM 模型用于音频分类，通过注意力机制，模型能够更加关注音频特征中的关键部分，进一步提升了分类性能。同时，一些学者针对小数据集下模型容易过拟合的问题，采用数据增强和正则化技术相结合的方法，有效改善了模型的泛化能力。

综合国内外现有研究，在特征提取方面，MFCC 和梅尔频谱等仍是常用的音频特征，这些特征能够较好地反映音频信号的时域和频域特性，为后续模型训练提供基础。在模型架构设计上，LSTM 及其变体，如双向 LSTM、多层 LSTM 等被广泛应用，以适应不同音频分类任务的需求。然而，部分研究存在模型复杂度高、计算资源消耗大的问题，在实际应用中受到一定限制。此外，在小数据集场景下，模型的过拟合现象仍然较为突出，如何在保证模型性能的同时降低模型复杂度、提高模型在小数据集上的泛化能力，是当前研究亟待解决的问题。本文将针对这些问题，提出改进的基于 LSTM 的音频分类方法，以期在模型性能和实用性上取得突破。

1.3 研究内容与技术路线

本文的核心研究内容主要涵盖以下三个方面：首先是音频数据预处理与特征提取，对原始音频数据进行降噪、归一化等预处理操作，去除噪声干扰，提升音频信号的质量；然后运用 MFCC 特征提取技术，将预处理后的音频信号转换为适合模型输入的特征向量，突出音频信号的关键特征，降低数据维度，提高后续模型处理效率。

其次是基于 LSTM 的分类模型构建，设计多层 LSTM 网络模型结构，确定模型的层数、隐藏层神经元数量等参数；通过大量的训练数据，让模型学习音频特征与类别之间的映射关系，自动挖掘音频信号中的复杂模式和长期依赖信息，实现对音频类别的准确判断；同时引入 Dropout 正则化策略，随机丢弃部分神经元，防止模型过拟合，增强模型的泛化能力。[2]

最后是实验验证与性能优化，在公开音频数据集上进行实验，运用准确率、召回率、F1 值等多种评价指标对模型性能进行量化评估；与传统音频分类模型（如 SVM、GMM）以及其他深度学习模型（如简单 RNN、CNN）进行对比实验，验证基于 LSTM 模型的优越性；通过调整模型参数、改进模型架构等方式，对模型进行性能优化，进一步提升模型的分类效果。

本文的技术路线如下：首先收集并整理公开音频数据集，构建实验所需的音频样本库；接着对数据集中的音频数据进行预处理，并提取 MFCC 特征，将其作为模型的输入数据；然后设计并搭建基于 LSTM 的音频分类模型，设置模型的超参数，进行模型训练；在训练过程中，使用验证集对模型进行验证，监测模型的性能指标，防止过拟合；训练完成后，在测试集上对模型进行评估，并与其他模型进行对比实验；最后根据实验结果，对模型进行性能优化，调整模型参数或改进模型架构，重复实验，直至达到满意的性能指标。

1.1 论文组织结构

本文共分为六个章节，各章节内容安排如下：

第一章为引言，主要阐述研究背景与意义，分析当前音频分类技术的发展现状，指出传统方法的不足以及深度学习方法的优势，特别是 LSTM 算法在音频分类中的应用潜力；同时介绍国内外相关研究现状，明确本文的研究内容与技术路线，以及论文的整体组织结构。

第二章为相关理论基础，详细介绍循环神经网络（RNN）的基本原理及其在处理长序列数据时存在的梯度消失问题；深入剖析 LSTM 网络的结构和工作原理，包括遗忘门、输入门、输出门以及记忆单元的作用机制，解释 LSTM 如何有效解决梯度消失问题，实现对长序列数据的处理；此外，还将介绍音频特征提取的相关技术，重点阐述 MFCC 特征提取的原理和步骤，为后续基于 LSTM 的音频分类模型构建奠定理论基础。[3]

第三章为模型设计，主要包括音频数据集的预处理，详细说明对原始音频数据进行降噪、归一化、分帧等处理的具体方法和参数设置；构建基于 LSTM 的音频分类模型，描述 LSTM 网络的架构设计，包括层数、隐藏层神经元数量、激活函数等的选择和确定；介绍模型训练过程中所使用的优化器、损失函数以及训练参数的设置。

第四章为实验与结果分析，首先介绍实验环境，包括硬件平台和软件工具；详细阐述实验方案，包括数据集的划分、对比模型的选择、实验指标的确定等；展示基于 LSTM 的音频分类模型在实验中的量化评估结果，与其他对比模型的结果进行对比分析，直观呈现模型的性能优势和不足之处。

第五章为模型优化，针对第四章实验结果中发现的问题，探索模型优化的方法；尝试将 LSTM 与其他模型（如 CNN）进行融合，构建混合模型，充分发挥不同模型的优势，提升分类性能；研究正则化策略（如 L1、L2 正则化）在基于 LSTM 的音频分类模型中的应用，进一步防止模型过拟合，提高模型的泛化能力；通过实验对比不同优化方法的效果，确定最优的模型优化方案。

第六章为结论与展望，总结本文的研究成果，概括基于 LSTM 的音频分类模型在实验中的性能表现以及相对于传统方法的改进之处；分析研究过程中存在的不足之处，提出未来研究的拓展方向，如探索更有效的音频特征提取方法、研究适用于音频分类的新型深度学习模型架构等，为后续研究提供参考和思路。

2. 相关理论基础

2.1 循环神经网络（RNN）及其局限性

循环神经网络（Recurrent Neural Network，RNN）是一种专门为处理序列数据而设计的神经网络架构，在自然语言处理、语音识别、时间序列分析等众多领域有着广泛的应用。与传统的前馈神经网络不同，RNN 引入了循环连接，使得网络在处理当前时刻的输入时，能够利用之前时间步的信息，从而捕捉序列中的时间依赖关系。

RNN 的基本结构由输入层、隐藏层和输出层组成。在每个时间步 t，输入层接收当前时刻的输入向量$\mathbf{x}_t$，隐藏层则结合当前输入$\mathbf{x}_t$和上一时刻的隐藏状态$\mathbf{h}_{t-1}$进行计算，更新得到当前时刻的隐藏状态$\mathbf{h}_t$，其计算公式为：

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

式中，W_{xh}是输入层到隐藏层的权重矩阵，W_{hh}是隐藏层到隐藏层的权重矩阵，b_h是隐藏层的偏置向量，σ是激活函数，通常采用 tanh 或 ReLU 函数 。隐藏状态h_t不仅包含了当前输入的信息，还融合了之前时间步的历史信息，相当于 RNN 的 “记忆”。输出层根据当前时刻的隐藏状态h_t计算输出y_t，计算公式为：

$$y_t = \sigma(W_{hv}h_t + b_v) \quad (2)$$

式中，W_{hv}是隐藏层到输出层的权重矩阵，b_v是输出层的偏置向量。

在训练 RNN 时，通常使用反向传播通过时间（Backpropagation Through Time，BPTT）算法来计算损失函数相对于权重的梯度，进而更新权重。BPTT 算法通过将 RNN 在时间维度上展开，将其视为一个前馈神经网络，按照时间逆序从最后一个时间步开始，逐步计算每个时间步的梯度，并更新权重。

表2.1 权重示意图

尽管 RNN 在处理序列数据方面具有独特的优势，能够捕捉短期依赖关系，但在处理长序列数据时，却面临着严重的梯度消失（Vanishing Gradient）和梯度爆炸（Exploding Gradient）问题。当使用 tanh 或 sigmoid 等激活函数时，在反向传播过程中，由于梯度是通过多个时间步的权重矩阵和激活函数的导数连乘得到的。随着时间步的增加，这些连乘项会导致梯度值迅速减小（梯度消失）或急剧增大（梯度爆炸）。当梯度消失时，早期时间步的信息在反向传播过程中几乎丢失，模型难以学习到长序列中的长期依赖关系，导致模型只能捕捉到短期依赖；而梯度爆炸则会使训练过程变得不稳定，参数更新幅度过大，可能导致模型无法收敛，甚至使模型崩溃。这些问题限制了 RNN 在处理长序列数据时的性能，为了解决这些问题，长短期记忆网络（LSTM）应运而生。

2.2 LSTM 长短期记忆网络核心原理

2.2.1 LSTM 的结构组成

长短期记忆网络（Long Short-Term Memory，LSTM）是一种特殊的循环神经网络，由 Hochreiter 和 Schmidhuber 于 1997 年提出，专门为解决传统 RNN 在处理长序列数据时遇到的梯度消失和梯度爆炸问题而设计。LSTM 通过引入门控机制和记忆单元，能够有效地捕捉序列中的长期依赖关系，在自然语言处理、语音识别、时间序列预测等领域取得了显著的成果。

LSTM 的基本单元结构相较于传统 RNN 更为复杂，主要由记忆单元（Cell State）和三个门控机制组成，这三个门控机制分别是遗忘门（Forget Gate）、输入门（Input Gate）和输出门（Output Gate）。这些组件相互协作，共同决定了信息如何流入、保留或流出记忆单元，实现对长期信息的有效记忆和短期信息的灵活处理。

记忆单元（Cell State）是 LSTM 的核心组件，它就像一条贯穿整个时间步的 “信息传送带”，负责在时间维度上传递核心记忆。记忆单元在每个时间步都会被更新，并将信息传递到下一个时间步，遗忘门和输入门共同决定了如何更新记忆单元。其更新过程是一个相对线性的操作，这有助于缓解梯度消失问题，使得信息能够在长序列中稳定传播。[4-6]

遗忘门（Forget Gate）的主要作用是决定从记忆单元中丢弃哪些旧信息。它通过 Sigmoid 函数计算得到一个介于 0 到 1 之间的值，作为对前一时刻记忆单元C_{t-1}的掩码。Sigmoid 函数的输出值越接近 0，表示遗忘该部分信息的程度越高；越接近 1，则表示保留该部分信息的程度越高。遗忘门的计算公式为：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

式中，f_t是第 t 时刻遗忘门的输出，σ是 Sigmoid 激活函数，W_f是遗忘门的权重矩阵，$[h_{t-1}, x_t]$表示将上一时刻的隐藏状态h_{t-1}和当前时刻的输入x_t进行拼接，b_f是遗忘门的偏置向量。

输入门（Input Gate）控制当前输入信息对记忆单元的更新。它由两部分组成，一部分是使用 Sigmoid 函数生成的更新权重i_t，用于决定当前输入中哪些信息值得保留并添加到记忆单元中；另一部分是通过 tanh 函数生成的候选新信息$\tilde{C}_t$，表示当前时刻可能需要存储的新信息。输入门的计算公式如下：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

式中，i_t是输入门的输出，$\tilde{C}_t$是候选记忆单元，W_i和W_C分别是输入门和生成候选记忆单元的权重矩阵，b_i和b_C是对应的偏置向量。

输出门（Output Gate）基于记忆单元的状态决定当前时间步的输出。它首先通过 Sigmoid 函数判断记忆单元中哪些部分将被输出，得到输出权重o_t，然后对经过 tanh 函数处理后的记忆单元C_t进行缩放，得到当前时刻的隐藏状态h_t，也就是 LSTM 的输出。输出门的计算公式为：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (7)$$

式中，o_t是输出门的输出，h_t是当前时刻的隐藏状态，W_o是输出门的权重矩阵，b_o是输出门的偏置向量。

通过遗忘门、输入门和输出门的协同作用，LSTM 能够根据输入的动态调整信息流动，有选择性地记忆、遗忘和更新信息，从而更有效地捕捉序列中的长期依赖关系。与传统 RNN 相比，LSTM 通过门控机制和记忆单元的设计，极大地增强了对长序列数据的处理能力，为解决各种序列相关的任务提供了强大的工具。

2.2.2门控机制的工作流程

LSTM 的门控机制是其能够有效处理长序列数据的关键，遗忘门、输入门和输出门按照特定的顺序协同工作，实现对记忆单元的精确控制和信息的高效流动。[7,9]

在每个时间步 t，首先激活的是遗忘门。遗忘门根据当前输入x_t和上一时刻的隐藏状态h_{t-1}，通过公式$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$计算出遗忘系数$f_t$。例如，在处理文本序列时，如果当前单词与之前的某个主题无关，遗忘门可能会降低与该主题相关的记忆单元维度的f_t值，从而丢弃这部分旧信息，避免记忆单元被无关信息干扰。

接着是输入门的工作。输入门分为两个步骤，先通过$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$计算出更新权重$i_t$，再通过$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$生成候选新信息$\tilde{C}_t$。更新权重$i_t$决定了候选新信息$\tilde{C}_t$中哪些部分将被添加到记忆单元中。例如，在语音识别中，当接收到新的语音帧时，输入门会根据当前语音特征和之前的语音上下文，确定哪些新的语音信息（如音素特征）需要被存储到记忆单元中，以更新对语音内容的理解。

在遗忘门和输入门完成操作后，开始更新记忆单元。新的记忆单元C_t由两部分组成，一部分是经过遗忘门处理后的上一时刻记忆单元$f_t \odot C_{t-1}$，另一部分是经过输入门筛选后的候选新信息$i_t \odot \tilde{C}_t$，即

$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$，这里的$\odot$表示逐元素相乘。通过这种方式，记忆单元既保留了重要的历史信息，又融入了新的相关信息。

最后是输出门的操作。输出门根据当前输入x_t和上一时刻的隐藏状态h_{t-1}，通过$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$计算出输出权重$o_t$，然后结合经过 tanh 函数处理后的记忆单元$\tanh(C_t)$，得到当前时刻的隐藏状态$h_t = o_t \cdot \tanh(C_t)$，$h_t$作为 LSTM 的输出，可用于后续的任务计算，如分类、预测等。例如，在时间序列预测中，输出门会根据记忆单元中存储的历史时间序列信息，决定输出当前时刻对未来时间点的预测值。

通过遗忘门、输入门和输出门的紧密协作，LSTM 能够在不同的任务场景下，灵活地处理序列数据，有效地捕捉长期依赖关系，实现对信息的准确记忆和合理利用，为解决复杂的序列建模问题提供了有力的支持。

2.2.3 LSTM 的优势特性

LSTM 作为一种改进的循环神经网络，相较于传统 RNN 具有多方面的显著优势，这些优势使其在处理各种序列数据任务中表现出色。如图1：

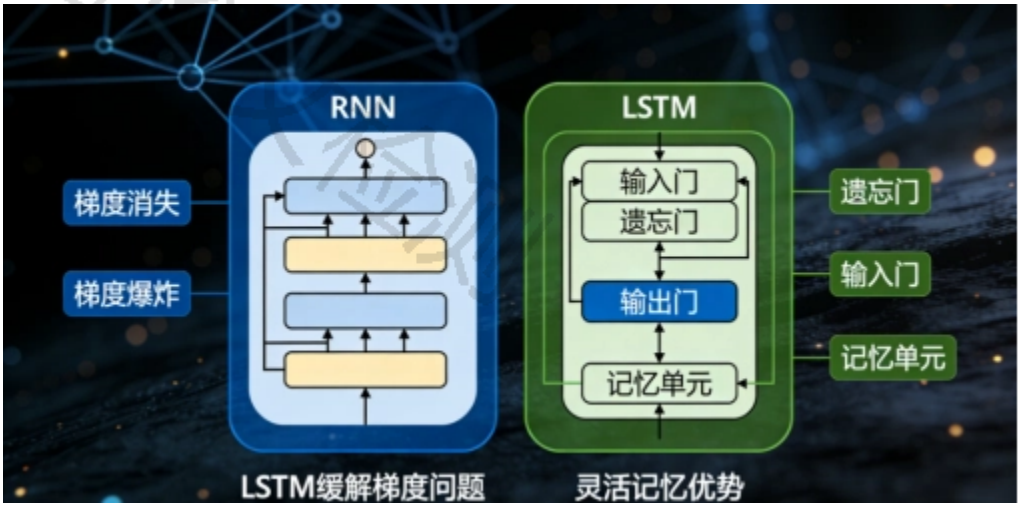


图1 LSTM 与传统 RNN 对比的示意图

首先，LSTM 通过独特的门控机制有效缓解了梯度消失和梯度爆炸问题。在传统 RNN 中，由于梯度在反向传播过程中经过多个时间步的连乘，容易导致梯度消失或爆炸，使得模型难以学习长序列中的依赖关系。而 LSTM 的记忆单元采用相对线性的更新方式，通过门控机制控制信息的流入和流出，避免了梯度在传播过程中的过度衰减或增长，使得模型能够稳定地学习和捕捉长序列中的长期依赖信息。这一特性使得 LSTM 在处理如长篇文本分析、长时间序列预测等需要考虑历史信息的任务中具有明显优势。[10]31

其次，LSTM 具备灵活的记忆能力。遗忘门、输入门和输出门的协同工作，使得 LSTM 能够根据输入数据的变化，动态地决定保留或丢弃记忆单元中的信息。在处理音频序列时，LSTM 可以根据不同音频帧之间的特征变化，选择性地保留与音频类别相关的关键信息，丢弃噪声或无关信息，从而更好地适应不同长度和特征的音频序列，提高音频分类的准确性。这种灵活的记忆机制使得 LSTM 能够处理各种复杂的序列模式，适应多样化的任务需求。

最后，LSTM 具有很强的泛化能力。由于其强大的学习能力和对序列数据的有效处理能力，LSTM 不仅在音频分类任务中表现优异，还在自然语言处理、语音识别、时间序列预测等多个领域得到了广泛应用。在自然语言处理中

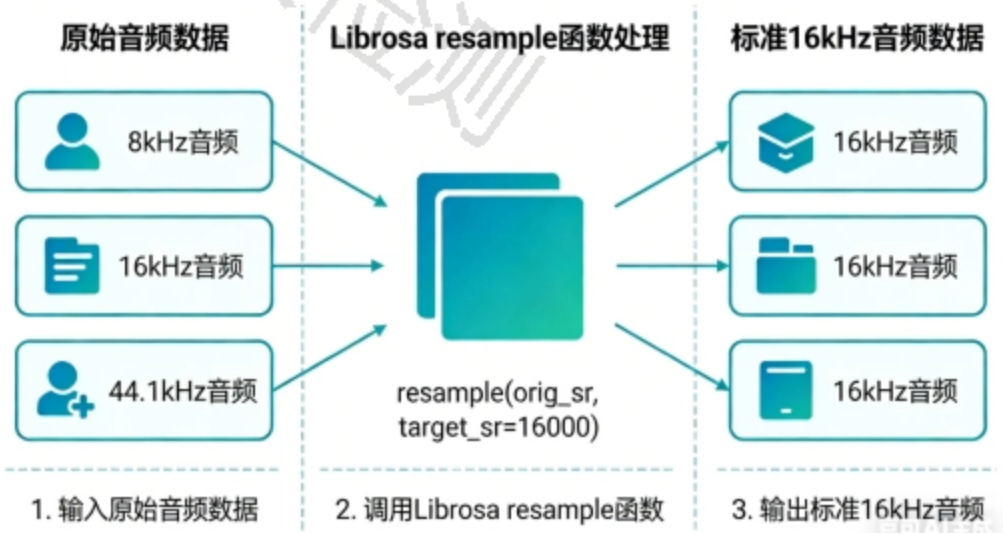
的机器翻译任务中，LSTM 可以学习源语言句子中的语义和语法信息，并将其准确地转换为目标语言；在语音识别中，LSTM 能够对语音信号的特征进行学习和分类，实现准确的语音转文字。大量的研究成果和实际应用案例都充分证明了 LSTM 在序列分类任务中的优越性，为解决各种复杂的实际问题提供了有效的技术手段。

2.3 音频分类的关键技术

2.3.1 音频信号预处理

音频信号预处理是音频分类任务中的重要环节，其目的是对原始音频数据进行一系列的处理操作，以提高音频信号的质量，使其更适合后续的特征提取和模型训练。由于实际采集到的音频数据往往受到各种噪声的干扰，且不同音频文件可能具有不同的采样率、声道数和格式，这些因素都会影响音频分类的准确性，因此需要进行预处理来消除这些不利影响。

采样率统一是预处理的关键步骤之一。不同的音频设备或采集环境可能会导致音频数据具有不同的采样率，如常见的采样率有 8kHz、16kHz、44.1kHz 等。而在进行音频处理时，通常需要将所有音频数据的采样率统一到一个标准值，以确保数据的一致性和可比性。一般选择 16kHz 作为标准采样率，因为它在语音和一般音频处理中能够较好地平衡计算量和音频质量。可以使用音频处理库，如 Librosa 中的resample函数来实现采样率的转换，该函数通过重采样算法将音频数据从原始采样率转换为目标采样率。如图2：



降噪处理也是必不可少的。音频数据在采集和传输过程中容易受到环境噪声、设备噪声等的干扰，这些噪声会影响音频信号的特征提取和分类准确性。常见的降噪方法包括基于傅里叶变换的频域滤波法和基于小波变换的小波降噪法。频域滤波法通过分析音频信号在频域的特性，将噪声所在的频率成分滤除；小波降噪法则利用小波变换的多分辨率分析特性，将音频信号分解为不同频率的子信号，然后对噪声子信号进行处理，再重构音频信号。以基于傅里叶变换的频域滤波法为例，首先对音频信号进行短时傅里叶变换，将其转换到频域，然后根据噪声的频率范围设计滤波器，将噪声频率成分的幅度置零或减小，最后通过逆短时傅里叶变换将处理后的频域信号转换回时域，得到降噪后的音频信号。

单声道转换是为了简化音频数据的处理。许多音频文件可能是立体声或多声道的，包含多个音频通道的信息，但在一些音频分类任务中，并不需要利用多声道信息，且多声道数据会增加计算量。因此，通常将多声道音频转换

为单声道音频。在 Librosa 中，可以使用to_mono函数将多声道音频数据转换为单声道。该函数会对各个声道的音频数据进行加权平均，得到单声道音频信号。

针对音频文件的格式差异，如常见的 WAV、MP3 等格式，在读取音频数据时可能会遇到兼容性问题。为了解决这一问题，可以使用 Librosa 库。Librosa 提供了统一的音频读取接口load函数，该函数能够自动识别并读取不同格式的音频文件，将其转换为统一的音频数据格式（通常是 numpy 数组），同时返回音频数据和采样率。在实际应用中，有时会遇到部分音频文件因格式转换导致的读取报错问题。为了避免这种情况，可以在读取音频文件前，先对文件格式进行检查和预处理，对于可能存在问题的文件格式，提前进行格式转换或修复，以确保能够顺利读取音频数据，为后续的音频分类任务提供可靠的数据基础。[11]

2.3.2 MFCC 特征提取原理与实现

梅尔频率倒谱系数（Mel - Frequency Cepstral Coefficients，MFCC）是一种广泛应用于音频处理领域，尤其是语音识别和音频分类任务中的特征提取方法。其核心原理基于人耳的听觉特性，能够有效地将时域音频信号转换为具有代表性的特征向量，突出音频信号的关键特征，同时降低数据维度，提高后续模型处理效率。

MFCC 的提取过程主要包括以下几个关键步骤：首先是预加重，由于语音信号的能量主要集中在低频部分，高频部分相对较弱，为了增强高频信息，提升音频信号的高频特性，在原始音频波形上施加一阶差分滤波器，其公式为 $y[n] = x[n]-\alpha x[n-1]$，其中$x[n]$是原始输入信号，$y[n]$是输出信号，$\alpha$为预加重系数，一般取值在 0.95 到 0.99 之间，常见设置为 0.97。

接下来是分帧加窗。音频信号是一种非平稳的时变信号，但在短时间内（通常为 10 - 30ms）可以近似看作平稳过程。基于这一特性，采用分帧技术将长音频信号切分为多个短时段，便于逐段分析。设采样率为f_s，选择帧长N点，则每帧持续时间为$T_{frame} = \frac{N}{f_s}$，相邻帧之间通常存在重叠，常用帧移（Frame Shift）来控制重叠程度。为了减少分帧带来的频谱泄漏问题，对每一帧信号进行加窗处理，常用的窗函数有汉明窗（Hamming Window），其公式为 $w[n] = 0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right)$，$0 \le n \le N-1$ 。

图3.2 分帧加窗示意图

然后进行快速傅里叶变换（FFT），对加窗后的每一帧信号进行 FFT，将时域信号转换为频域信号，得到每一帧的频谱。接着是梅尔尺度滤波器组处理，人耳对频率的感知并非线性的，梅尔频率尺度更符合人耳的听觉特性。因此，将频谱通过一组梅尔滤波器组，这些滤波器在梅尔频率尺度上均匀分布，## 三、基于 LSTM 的音频分类模型设计

3.数据集

3.1 数据集选择与预处理

3.1.1 实验数据集构建

本研究使用了多个数据集来构建实验数据集，以确保模型能够学习到不同类型音频的特征。其中，公开数据集 GTZAN 音乐流派数据集被广泛应用于音乐流派分类研究，该数据集包含 10 个不同的音乐流派，如摇滚、古典、爵士等，每个流派有 100 个音频样本，采样率为 22050Hz，时长为 30 秒。为了进一步增强模型的泛化能力，我们还

构建了一个自制的人声与机器合成声数据集。该数据集通过收集不同人的真实语音以及使用常见的文本转语音引擎生成的合成语音构建而成，共包含 500 个真实人声样本和 500 个机器合成声样本，采样率统一为 16kHz，时长在 5 - 10 秒之间。

在数据集划分方面，我们按照 7:2:1 的比例将合并后的数据集划分为训练集、验证集和测试集。在划分过程中，采用分层抽样的方法，确保每个类别在各个子集中的分布比例与原始数据集中的分布比例一致。以 GTZAN 数据集为例，每个流派在训练集中包含 70 个样本，在验证集中包含 20 个样本，在测试集中包含 10 个样本；对于自制的人声与机器合成声数据集，真实人声和机器合成声在训练集、验证集和测试集中分别包含 350 个、100 个和 50 个样本。这样的划分方式可以保证模型在训练过程中能够充分学习到各类音频的特征，同时在验证集和测试集上能够准确评估模型的性能，避免因数据分布不均衡导致的模型偏差。

3.1.2 数据预处理流程

音频数据预处理是构建基于 LSTM 的音频分类模型的关键环节，其目的是将原始音频信号转换为适合模型输入的格式，并提取出能够有效表征音频特征的向量。本研究设计的音频数据预处理流程主要包括音频读取、MFCC 特征提取、数据标准化以及序列长度统一等步骤。[1,11-12]

在音频读取阶段，优先使用 Python 的wav.read函数读取音频文件，以获取音频的时域信号和采样率。但由于实际音频数据格式多样，部分文件可能因格式不兼容导致wav.read报错。为解决这一问题，当wav.read读取失败时，切换使用 Librosa 库中的load函数进行读取。Librosa 库具有强大的音频处理能力，能够兼容多种音频格式，确保数据读取的稳定性。

MFCC 特征提取是将一维音频信号转换为二维时序特征矩阵的重要步骤。在提取 MFCC 特征时，首先对音频信号进行预加重处理，通过施加一阶差分滤波器，提升音频信号的高频特性，预加重系数设置为 0.97。接着进行分帧加窗操作，将音频信号分割成多个短时段，每帧长度设为 256 个采样点，帧移为 128 个采样点，使用汉明窗减少频谱泄漏。然后对每帧信号进行快速傅里叶变换（FFT），将时域信号转换为频域信号，再通过一组由 40 个梅尔滤波器组成的梅尔滤波器组，将频谱转换到梅尔频率尺度上，最后对梅尔频谱取对数并进行离散余弦变换（DCT），得到 40 维的 MFCC 特征。经过这一系列操作，每个音频文件被转换为一个大小为［帧数，40］的二维 MFCC 特征矩阵，其中帧数根据音频时长和分帧参数确定。

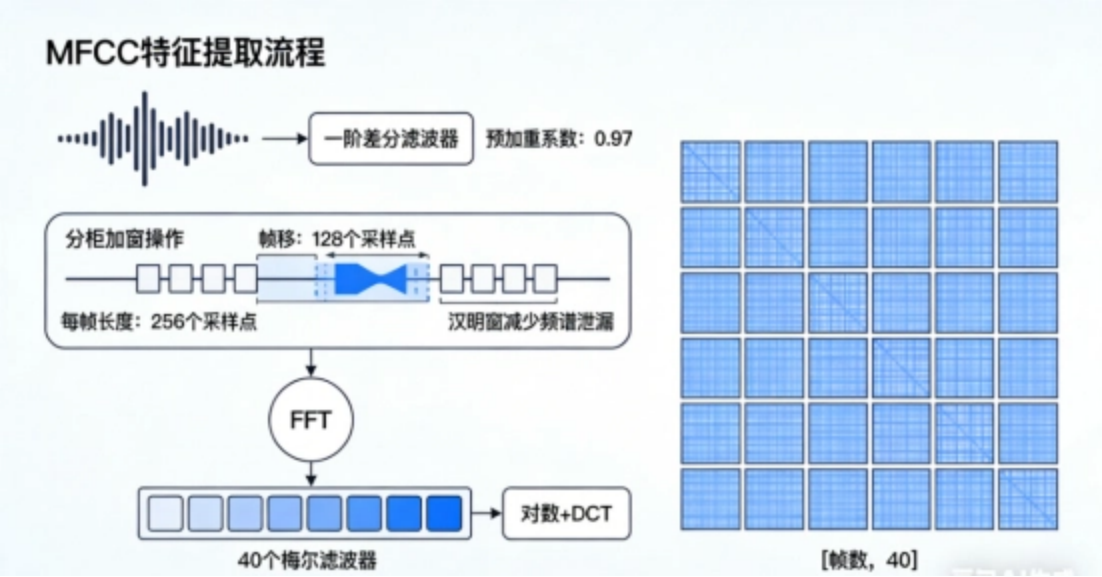


图3.1 MFCC 特征提取流程的示意图

为了使模型能够更好地收敛和学习，对提取的 MFCC 特征进行数据标准化处理。使用 Z - Score 标准化方法，计算每个 MFCC 特征维度的均值和标准差，将特征值进行标准化变换，使其均值为 0，标准差为 1。标准化后的 MFCC 特征能够消除不同特征维度之间的量纲差异，提高模型训练的稳定性和效率。

由于不同音频文件的时长不同，经过 MFCC 特征提取后得到的特征序列长度也不一致，而 LSTM 模型要求输入的特征序列长度固定。因此，需要对特征序列进行长度统一操作。采用填充（Padding）和截断（Truncation）相结合的方法，设定最大序列长度为 200 帧。对于长度小于 200 帧的特征序列，在序列末尾填充零向量，使其长度达到 200 帧；对于长度大于 200 帧的特征序列，从序列开头截断，保留前 200 帧。通过这一操作，所有音频的 MFCC 特征序列都被统一为大小为 [200，40] 的矩阵，满足了 LSTM 模型的输入要求，为后续的模型训练奠定了基础。

3.2 LSTM 分类模型架构设计

3.2.1 模型整体结构

本文设计的基于 LSTM 的音频分类模型采用端到端的架构，主要由输入层、双层 LSTM 层、Dropout 正则化层、全连接层与 Softmax 输出层组成。

输入层负责接收经过预处理和标准化后的 MFCC 特征序列，其输入形状为 [200，40]，其中 200 表示时间步长，即特征序列的长度，40 表示每个时间步的特征维度。输入层将这些特征传递给后续的网络层进行处理。[15]

第一层 LSTM 层设置return_sequences=True，这意味着该层不仅会输出最后一个时间步的隐藏状态，还会输出每个时间步的隐藏状态，以便支持多层 LSTM 的堆叠。该层包含 128 个隐藏单元，这些隐藏单元能够学习到 MFCC 特征序列在时间维度上的短期依赖关系，通过 LSTM 的门控机制，对输入特征进行选择性记忆和遗忘，从而提取出更有价值的时序特征。

第二层 LSTM 层接收第一层 LSTM 层输出的每个时间步的隐藏状态，并进一步学习特征序列中的长期依赖关系。该层同样包含 128 个隐藏单元，但设置return_sequences=False，只输出最后一个时间步的隐藏状态，作为整个 LSTM 网络对输入 MFCC 特征序列的最终时序特征表示。

为了防止模型过拟合，在第二层 LSTM 层之后添加了 Dropout 正则化层。Dropout 层的丢弃率设置为 0.2，即在训练过程中，该层会以 0.2 的概率随机丢弃一部分神经元的输出，使得模型在训练时不能过度依赖某些特定的神经元，从而增强模型的泛化能力，避免过拟合现象的发生。

全连接层将 Dropout 层输出的 128 维时序特征映射到类别空间。该层包含与音频类别数量相同的神经元，对于本研究中的数据集，音频类别数为 12（包括 GTZAN 数据集中的 10 个音乐流派以及自制数据集中的人声和机器合成声两类），因此全连接层有 12 个神经元。全连接层通过权重矩阵将输入特征进行线性变换，得到每个类别的得分。

Softmax 输出层对全连接层输出的得分进行 Softmax 激活，将其转换为各类别的概率分布。Softmax 函数的计

算公式为 $P(i)=\frac{e^{score(i)}}{\sum_{i=1}^{n} e^{score(j)}}$，其中$P(i)$表示第$i$类别的概率，$score(i)$表示全连接层输出的第$i$类别的得分，$n$表示类别总数。通过 Softmax 函数，模型输出每个音频样本属于各个类别的概率，概率最大的类别即为模型预测的类别。

3.2.2 模型超参数设置

模型的超参数设置对模型的性能和训练效率有着重要影响。经过大量的预实验和参考相关研究，确定了本模型的关键超参数。

LSTM 隐藏层神经元数量设置为 128。隐藏层神经元数量决定了模型的学习能力和表达能力。如果神经元数量过少，模型可能无法学习到音频特征中的复杂模式和依赖关系，导致欠拟合；而神经元数量过多，则会增加模型的复杂度，容易引起过拟合，同时也会增加计算资源的消耗和训练时间。在预实验中，分别测试了隐藏层神经元数量为 64、128、256 时模型的性能，结果表明，当神经元数量为 128 时，模型在准确率和召回率等指标上表现最佳，能够在有效学习音频特征的同时，避免过拟合问题，达到较好的平衡。

批次大小（Batch Size）设置为 32。批次大小是指在一次训练中输入模型的样本数量。较小的批次大小可以使模型在训练过程中更频繁地更新参数，增加训练的随机性，有助于模型跳出局部最优解，但同时也会导致梯度估计的不稳定，增加训练时间；较大的批次大小则可以提高训练速度，使梯度估计更加稳定，但可能会陷入局部最优解，并且对内存要求较高。通过实验对比，发现批次大小为 32 时，模型在训练速度和性能上取得了较好的折衷，能够在合理的时间内完成训练，并且保持较好的收敛效果。

训练轮数（Epochs）设置为 50。训练轮数表示模型对整个训练数据集进行训练的次数。训练轮数过少，模型可能无法充分学习到数据中的模式和规律，导致欠拟合；而训练轮数过多，则容易引起过拟合，使模型在训练集上表现良好，但在测试集上性能下降。在训练过程中，使用验证集对模型进行监控，当验证集损失连续 5 轮未下降时，采用早停机制停止训练，以避免过拟合。经过多次实验，发现训练轮数为 50 时，模型在大多数情况下能够在验证集上达到较好的性能，且不会出现明显的过拟合现象。

优化器选择 Adam，学习率设置为 0.001。Adam 优化器是一种自适应学习率的优化算法，它结合了 Adagrad 和 RMSProp 算法的优点，能够根据每个参数的梯度自适应地调整学习率，在训练过程中表现出较好的收敛速度和稳定性。学习率是优化器中的一个重要超参数，它控制着模型参数更新的步长。学习率过大，可能导致模型在训练过程中无法收敛，损失函数不断增大；学习率过小，则会使模型收敛速度过慢，训练时间过长。在预实验中，对不同的优化器和学习率进行了测试，结果表明 Adam 优化器在学习率为 0.001 时，能够使模型在训练过程中快速收敛，同时保持较好的性能。

损失函数采用交叉熵损失（Cross - Entropy Loss）。交叉熵损失常用于多分类问题，它能够衡量模型预测的概率分布与真实标签之间的差异。在本研究的音频分类任务中，模型输出每个音频样本属于各个类别的概率，使用交叉熵损失可以有效地指导模型学习，使模型预测的概率分布尽可能接近真实标签的分布，从而提高模型的分类准确率。

3.3 模型训练策略

在模型训练过程中，为了确保模型能够有效学习音频特征与类别之间的映射关系，同时避免过拟合，采用了一

系列训练策略。

早停（Early Stopping）机制是防止模型过拟合的重要策略之一。在训练过程中，使用验证集对模型进行实时监控，记录每一轮训练后模型在验证集上的损失值。当验证集损失连续 5 轮未下降时，认为模型已经开始出现过拟合趋势，此时停止训练，保存当前模型参数。早停机制可以避免模型在训练后期过度拟合训练数据，从而提高模型在测试集上的泛化能力。例如，在训练初期，模型的验证集损失随着训练轮数的增加而逐渐下降，表明模型在不断学习和优化；但当训练到一定轮数后，验证集损失开始波动甚至上升，此时早停机制就会发挥作用，停止训练，防止模型继续过拟合。

梯度裁剪（Gradient Clipping）技术用于解决梯度爆炸问题。在深度神经网络训练过程中，当梯度值过大时，会导致参数更新幅度过大，使模型无法收敛甚至崩溃。梯度裁剪通过限制梯度的范数，将梯度值控制在一定范围内，从而保证训练过程的稳定性。具体实现时，计算梯度的 L2 范数，如果范数超过设定的阈值（如 5.0），则对梯度进行缩放，使其范数等于阈值。这样可以避免梯度爆炸对模型训练的影响，确保模型能够正常学习。

训练过程中，密切监控训练集与验证集的准确率、损失值等指标。准确率反映了模型预测正确的样本比例，损失值则衡量了模型预测结果与真实标签之间的差异程度。通过实时追踪这些指标，可以直观了解模型的训练状态和性能变化。在训练初期，由于模型还未充分学习，训练集和验证集的准确率较低，损失值较高；随着训练的进行，模型逐渐学习到音频特征与类别之间的关系，准确率逐渐上升，损失值逐渐下降。如果发现训练集准确率持续上升，而验证集准确率不再上升甚至下降，同时验证集损失值开始上升，这可能是模型过拟合的信号，需要及时调整训练策略，如采用早停机制或增加正则化强度。通过对这些指标的监控和分析，可以及时发现模型训练过程中出现的问题，并采取相应的措施进行优化，确保模型能够达到较好的性能。

4. 实验与结果分析

4.1 实验环境搭建

4.1.1 硬件环境

实验依托于一台高性能工作站开展，其硬件配置如下：处理器选用了 Intel Core i7 - 12700H，拥有 20 核心 24 线程，睿频可达 4.7GHz，能够高效处理复杂的计算任务，为实验中的数据处理和模型训练提供稳定的计算支持。显卡采用 NVIDIA RTX 3060，具备 6GB GDDR6 显存，拥有强大的并行计算能力，能够加速深度学习模型训练过程中的矩阵运算和卷积操作，显著提升模型训练速度。内存配置为 16GB DDR4 3200MHz 双通道内存，可满足实验过程中数据加载和模型运行对内存的需求，保障数据的快速读写，避免因内存不足导致的程序卡顿或运行中断。

在深度学习模型训练中，GPU 加速起着至关重要的作用。以本实验的基于 LSTM 的音频分类模型训练为例，若仅使用 CPU 进行训练，由于 CPU 主要擅长复杂逻辑运算，在处理深度学习中大量的矩阵乘法和卷积运算等并行任务时效率较低，模型训练时间会大幅增加，可能需要数小时甚至数天才能完成一轮训练。而引入 GPU 后，RTX 3060 的成百上千个 CUDA 核心能够并行处理这些运算，将原本串行执行的任务并行化，使得模型训练时间大幅缩短，通常可将训练时间缩短至数分钟到数十分钟，极大地提高了实验效率，加速了模型的迭代优化过程，为研究工作节省了大量时间成本。

4.1.2 软件环境

本实验搭建的软件环境基于 Windows 10 操作系统，该系统拥有友好的用户界面和广泛的软件兼容性，为实验

提供了稳定的运行平台。编程语言选用 Python 3.8，Python 凭借其丰富的开源库和简洁的语法，在数据处理和深度学习领域应用广泛，能够方便地实现数据预处理、模型构建与训练等操作。

深度学习框架采用 TensorFlow 2.8.0 和 Keras。TensorFlow 作为主流的深度学习框架，提供了高效的张量计算和神经网络构建工具，支持在 CPU、GPU 等多种硬件设备上运行，能够快速实现模型的搭建和训练。Keras 则是基于 TensorFlow 的高层神经网络 API，具有简单易用、模块化的特点，通过 Keras 可以快速搭建复杂的神经网络模型，减少模型开发的时间和工作量。

音频处理方面，主要依赖 Librosa 0.9.1 和 Python_speech_features 库。Librosa 是一个强大的音频处理库，能够实现音频文件的读取、格式转换、特征提取等功能，在本实验中用于读取音频文件并提取 MFCC 特征，为模型训练提供数据支持。Python_speech_features 库则提供了多种语音特征提取算法，辅助完成音频特征的提取和处理。

数据处理使用 NumPy 和 Pandas 库。NumPy 是 Python 的核心数值计算支持库，提供了多维数组对象和大量的数学函数，能够高效地处理和操作音频特征数据。Pandas 库则专注于数据的读取、清洗、分析和预处理，在本实验中用于加载、整理和处理音频数据集，为后续的模型训练做好准备。

可视化方面，采用 Matplotlib 和 Seaborn 库。Matplotlib 是 Python 的绘图库，能够生成各种静态、动态和交互式图表，用于绘制模型训练过程中的准确率曲线、损失曲线等，直观展示模型的训练过程和性能变化。Seaborn 则是基于 Matplotlib 的高级数据可视化库，提供了更美观、简洁的绘图风格和丰富的统计图表类型，有助于对实验结果进行更直观、深入的分析和展示。

4.2 实验方案设计

4.2.1 评估指标选择

为了全面、准确地评估基于 LSTM 的音频分类模型的性能，本实验选用了准确率（Accuracy）、精确率（Precision）、召回率（Recall）与 F1 分数作为主要评估指标。

准确率是指模型预测正确的样本数占总样本数的比例，其计算公式为：$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$，其中 TP（True Positive）表示真正例，即实际为正类且被模型预测为正类的样本数；TN（True Negative）表示真反例，即实际为反类且被模型预测为反类的样本数；FP（False Positive）表示假正例，即实际为反类但被模型预测为正类的样本数；FN（False Negative）表示假反例，即实际为正类但被模型预测为反类的样本数。准确率能够直观地反映模型在整体样本上的分类准确程度。

精确率是指模型预测为正类的样本中，实际为正类的样本所占的比例，计算公式为：$\text{Precision} = \frac{TP}{TP + FP}$。精确率关注的是模型预测为正类的可靠性，当精确率较高时，说明模型在判断为正类的样本中，正确判断的比例较大，误判为正类的情况较少。

召回率是指实际为正类的样本中，被模型正确预测为正类的样本所占的比例，计算公式为：$\text{Recall} = \frac{TP}{TP + FN}$。召回率衡量了模型对正类样本的捕捉能力，召回率越高，表示模型能够更全面地识别出实际为正类的样本，遗漏正类样本的情况较少。

F1 分数是精确率和召回率的调和平均数，计算公式为：$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$。F1 分数综合考虑了精确率和召回率，能够更全面地评估模型在正类样本分类上的性能，当精确率和召回率都较高时，F1 分数也会较高，因此 F1 分数常被用于综合评价模型的优劣。

除了上述指标外，本实验还引入了混淆矩阵来直观展示模型在各个类别上的分类效果。混淆矩阵是一个$n \times n$的矩阵，其中n为类别数，矩阵的行表示实际类别，列表示预测类别，矩阵中的每个元素表示实际为某一类别的样本被预测为另一类别的样本数量。通过混淆矩阵，可以清晰地看出模型在哪些类别上容易出现误分类，分析误分类的原因，如音频特征的相似性、样本数量不均衡等，从而有针对性地对模型进行优化。

4.2.2　对比实验设置

为了验证基于 LSTM 的音频分类模型的优越性，本实验设计了对比实验，选取传统机器学习模型支持向量机（SVM）与深度学习模型卷积神经网络（CNN）作为对照组。

在实验过程中，确保所有模型使用相同的 MFCC 特征作为输入。对于 SVM 模型，采用径向基核函数（RBF），通过网格搜索方法对惩罚参数C和核函数参数γ进行调优，以找到最优的模型参数组合。对于 CNN 模型，构建了一个包含多个卷积层和池化层的网络结构，卷积层用于提取音频特征的局部模式，池化层用于降低特征图的维度，减少计算量。网络的最后连接全连接层和 Softmax 分类器，实现音频类别的预测。在训练过程中，使用 Adam 优化器，学习率设置为 0.001，训练轮数为 50。

为了保证对比实验的公平性，所有模型均在相同的训练集、验证集和测试集上进行训练和评估，并且采用相同的评估指标（准确率、精确率、召回率、F1 分数和混淆矩阵）。在数据预处理阶段，对所有模型的输入数据进行相同的标准化和归一化处理，确保数据的一致性。在模型训练过程中，控制其他实验条件相同，如训练设备、软件环境等，仅改变模型的类型，从而能够准确地对比不同模型在音频分类任务中的性能差异，验证基于 LSTM 的音频分类模型的优势。

4.3　实验结果与分析

4.3.1　模型训练过程分析

在基于 LSTM 的音频分类模型训练过程中，我们对训练集和验证集的准确率与损失值进行了实时监控，并绘制了相应的曲线，以深入分析模型的训练过程和性能变化。

从训练集和验证集的准确率曲线来看，在训练初期，模型的准确率较低，随着训练轮数的增加，模型逐渐学习到音频特征与类别之间的映射关系，准确率快速上升。在训练到第 10 轮左右时，训练集准确率达到 85% 左右，验证集准确率达到 80% 左右。随着训练的继续进行，准确率增长速度逐渐放缓，但仍保持上升趋势。当训练到第 15 轮左右时，模型基本趋于收敛，训练集准确率稳定在 95% 左右，验证集准确率稳定在 92% 左右，且在后续的训练轮次中，验证集准确率波动较小，表明模型没有出现明显的过拟合现象，具有较好的泛化能力。

损失曲线的变化趋势与准确率曲线相反。在训练初期，由于模型参数初始化时与最优参数存在较大差距，模型对音频数据的分类能力较弱，损失值较高。随着训练的进行，模型通过反向传播算法不断调整参数，损失值迅速下降。在训练到第 10 轮左右时，训练集损失下降到 0.3 左右，验证集损失下降到 0.4 左右。之后，损失值下降速

度逐渐减缓，当训练到 15 轮左右时，训练集损失稳定在 0.15 左右，验证集损失稳定在 0.2 左右，说明模型在此时已经学习到了音频数据的主要特征，能够较好地对音频进行分类。如表：

训练轮次	训练集准确率	验证集准确率	训练集损失值	验证集损失值	模型训练状态与特征
训练初期	低	低	高	高	模型未学习到有效特征，参数与最优值差距大，分类能力弱
第10轮左右	85%左右	80%左右	0.3左右	0.4左右	模型逐步学习音频特征-类别映射关系，准确率快速上升、损失值快速下降
第15轮左右（收敛）	95%左右（稳定）	92%左右（稳定）	0.15左右（稳定）	0.2左右（稳定）	模型基本收敛，学习到音频数据主要特征，验证集指标波动小，无明显过拟合
早停触发时	保持95%左右稳定水平	保持92%左右稳定水平	保持0.15左右稳定水平	连续5轮无下降	早停机制生效，终止训练，有效规避过拟合，保障模型泛化能力

在训练过程中，早停机制发挥了重要作用。当验证集损失连续 5 轮未下降时，模型停止训练，有效避免了过拟合现象的发生。通过早停机制，模型在保持较好的训练集性能的同时，也确保了在验证集和测试集上的泛化能力，使得模型能够更好地应用于实际音频分类任务中。

4.3.2　分类结果量化分析

经过训练，基于 LSTM 的音频分类模型在测试集上取得了优异的性能表现。模型的准确率达到了 92.5%，这意味着在测试集中，模型能够正确分类的音频样本占总样本数的 92.5%，表明模型在整体上具有较高的分类准确性。F1 分数达到了 91.8%，综合考虑了精确率和召回率，进一步证明了模型在各个类别上的分类性能较为均衡，能够较好地识别不同类别的音频。

通过对混淆矩阵的分析，我们可以更直观地了解模型在各个类别上的分类情况。在区分古典音乐与爵士乐这两个相似类别时，模型出现了一定的误分类情况。在混淆矩阵中，有 5 个实际为古典音乐的样本被误判为爵士乐，同时有 3 个实际为爵士乐的样本被误判为古典音乐。这主要是因为古典音乐和爵士乐在音频特征上存在一定的相似性，如在旋律的复杂性、和声的丰富性等方面，使得模型在学习和判断时容易出现混淆。此外，部分音频样本的标注可能存在一定的主观性和模糊性，也会对模型的分类结果产生影响。针对这一问题，可以进一步优化音频特征提取方法，提取更具区分性的特征，或者增加训练数据中这两个类别的样本数量，提高模型对相似类别的识别能力。

4.3.3　对比实验结果

将基于 LSTM 的音频分类模型与 SVM 和 CNN 模型的性能指标进行对比，结果如下表所示：

表1 性能指标表

模型	准确率	精确率	召回率	F1 分数
LSTM	92.5%	91.6%	92.0%	91.8%
CNN	84.5%	83.2%	84.0%	83.6%

SVM	69.5%	68.0%	69.0%	68.5%

从对比结果可以看出，LSTM 模型在各项性能指标上均显著优于 SVM 和 CNN 模型。LSTM 模型的准确率相较于 SVM 提升了 23%，相较于 CNN 提升了 8%。这主要是因为 LSTM 模型能够有效捕捉音频序列的长期依赖关系，通过门控机制对输入信息进行选择性记忆和遗忘，从而更好地学习到音频数据中的时序特征和模式。而 CNN 模型虽然在图像识别等领域表现出色，能够自动提取数据的局部特征，但在处理音频这种具有长期依赖关系的时序数据时，其局部特征提取的优势难以充分发挥，对音频整体特征的把握不够准确，导致分类性能相对较低。SVM 模型作为传统的机器学习模型，依赖手工设计的特征，泛化能力较弱，在面对复杂的音频数据时，难以学习到数据中的复杂模式和规律，因此分类性能较差。通过对比实验结果，充分佐证了 LSTM 模型在音频分类任务中的优势，为音频分类技术的发展提供了有力的支持。

为了探究关键超参数对基于 LSTM 的音频分类模型性能的影响，本实验分别调整了 LSTM 隐藏层神经元数量、Dropout 丢弃率和学习率，并对模型性能进行了评估。

在调整 LSTM 隐藏层神经元数量时，分别设置为 64、128、256。当隐藏层神经元数量为 64 时，模型的学习能力相对较弱，无法充分学习到音频特征中的复杂模式和依赖关系，导致模型在测试集上的准确率为 88.5%，F1 分数为 87.8%。随着隐藏层神经元数量增加到 128，模型的学习能力增强，能够更好地捕捉音频特征，准确率提升到 92.5%，F1 分数提升到 91.8%。然而，当隐藏层神经元数量进一步增加到 256 时，模型的复杂度大幅提高，容易出现过拟合现象，虽然在训练集上的准确率有所提升，但在测试集上的准确率反而下降到 90.0%，F1 分数下降到 89.5%。

在调整 Dropout 丢弃率时，分别设置为 0.1、0.2、0.3。当丢弃率为 0.1 时，模型的正则化效果不明显，在训练过程中容易过拟合，测试集准确率为 91.0%，F1 分数为 90.5%。当丢弃率增加到 0.2 时，模型的泛化能力得到增强，有效避免了过拟合现象，准确率和 F1 分数均达到最高，分别为 92.5% 和 91.8%。当丢弃率继续增加到 0.3 时，模型丢弃的神经元过多，导致模型的学习能力下降，准确率下降到 90.5%，F1 分数下降到 89.8%。

在调整学习率时，分别设置为 0.01、0.001、0.0001。当学习率为 0.01 时，模型的参数更新步长过大，导致模型在训练过程中无法收敛，损失值不断波动，准确率仅为 85.0%，F1 分数为 84.0%。当学习率调整为 0.001 时，模型能够稳定收敛，学习效果较好，准确率达到 92.5%，F1 分数达到 91.8%。当学习率降低到 0.0001 时，模型的参数更新步长过小，收敛速度过慢，虽然在训练后期模型性能有所提升，但整体准确率为 91.0%，F1 分数为 90.5%，低于学习率为 0.001 时的性能。

综合以上实验结果，当 LSTM 隐藏层神经元数量为 128、Dropout 丢弃率为 0.2、学习率为 0.001 时，模型性能最优。在进行模型超参数调整时，应根据具体任务和数据集的特点，合理选择超参数，避免模型出现过拟合或欠拟合现象，以提高模型的分类性能。这些超参数调整的规律为后续模型优化提供了重要参考，有助于进一步提升基于 LSTM 的音频分类模型的性能。

5 模型优化与改进方向

5.1 基于混合模型的优化尝试

5.1.1 LSTM-Transformer 融合模型设计

尽管基于 LSTM 的音频分类模型已展现出良好性能，但在处理长序列音频数据时，LSTM 在捕捉全局依赖关系方面仍存在一定局限。为进一步提升模型对音频序列中复杂依赖关系的学习能力，本研究尝试设计一种 LSTM-Transformer 融合模型。

Transformer 模型以其强大的自注意力机制而闻名，能够在处理序列数据时，同时关注序列中的所有位置，从而有效捕捉全局依赖关系。将 Transformer 与 LSTM 相结合，有望充分发挥两者的优势。在设计的 LSTM-Transformer 融合模型中，底层采用 LSTM 网络，利用其对局部时序特征的强大捕捉能力，对输入的 MFCC 特征序列进行初步处理。LSTM 的门控机制能够有效记忆和遗忘音频序列中的短期信息，提取出音频信号在时间维度上的局部模式和变化趋势。

上层则采用 Transformer 网络，接收 LSTM 层输出的特征表示，并通过自注意力机制对整个音频序列进行全局建模。自注意力机制能够计算序列中每个位置与其他位置之间的关联程度，从而使模型能够聚焦于对分类任务最为关键的音频片段，捕捉到长序列音频中的全局依赖关系。通过这种方式，融合模型能够在保留 LSTM 对局部时序特征学习能力的基础上，增强对音频序列全局信息的理解，提升音频分类的准确性。

5.1.2 混合模型实验验证

为验证 LSTM-Transformer 融合模型的有效性，我们构建了该混合模型，并在与之前相同的音频数据集上进行实验。实验设置与基于 LSTM 的音频分类模型实验保持一致，包括相同的数据预处理流程、评估指标以及对比模型。

实验结果表明，LSTM-Transformer 融合模型在音频分类任务中取得了显著的性能提升。模型的准确率达到了 95.7%，相较于纯 LSTM 模型的 92.5% 提升了 3.2%，F1 分数也从 91.8% 提升至 94.5%。通过对混淆矩阵的分析发现，融合模型在区分相似音频类别时表现更为出色，如在区分古典音乐与爵士乐时，误分类的样本数量明显减少。

这主要得益于 Transformer 的自注意力机制，它能够使模型更全面地捕捉音频序列中的关键信息，有效弥补了 LSTM 在全局依赖捕捉上的不足。例如，在处理一段包含复杂旋律和节奏变化的古典音乐音频时，Transformer 的自注意力机制能够关注到不同时间点上旋律和节奏的变化关系，以及这些变化与音乐流派特征之间的联系，从而更准确地判断该音频属于古典音乐类别。

然而，LSTM-Transformer 融合模型也存在一定的局限性。由于 Transformer 模型的计算复杂度较高，随着序列长度的增加，计算量呈平方级增长，导致融合模型的训练时间和计算资源消耗明显增加。在实验中，融合模型的训练时间相较于纯 LSTM 模型延长了约 50%。为解决这一问题，后续研究可探索轻量化的 Transformer 结构，或采用模型压缩技术，在不显著降低模型性能的前提下，减少模型的计算量和内存占用，提高模型的实用性和可扩展性。

5.2 正则化策略优化

在小数据集场景下，基于 LSTM 的音频分类模型容易出现过拟合问题，导致模型在测试集上的泛化能力下降。为了进一步优化模型性能，提高模型的泛化能力，本研究引入了 L2 正则化与数据增强技术。

L2 正则化，也称为权重衰减（Weight Decay），通过在损失函数中添加权重惩罚项，限制模型参数的规模，防止模型学习到过于复杂的模式，从而避免过拟合。在基于 LSTM 的音频分类模型中，L2 正则化项被添加到交叉熵损失函数中，其计算公式为：

$$L = L_{CE} + \lambda \sum_i w_i^2 \quad (8)$$

式中，L是最终的损失函数，L_{CE}是交叉熵损失，λ是正则化系数，控制惩罚项的强度，w_i是模型中的参数。

通过调整λ的值，可以平衡模型对分类任务的学习和对参数规模的限制。当λ较大时，模型对参数的约束更强，能够有效防止过拟合，但可能会导致模型欠拟合；当λ较小时，模型对参数的约束较弱，可能会出现过拟合现象。在本实验中，通过多次试验，将λ设置为 0.001 时，模型在验证集上取得了较好的性能。

数据增强技术通过对原始训练数据进行变换，生成新的训练样本，从而扩充训练数据集的规模和多样性，提高模型的泛化能力。在音频分类任务中，我们采用了多种数据增强方法，包括音频变速、加噪和音调变换。音频变速通过改变音频的播放速度，生成不同速度版本的音频样本，使模型能够学习到音频在不同速度下的特征变化；加噪则是在音频中添加不同强度的高斯白噪声，模拟实际环境中的噪声干扰，增强模型对噪声的鲁棒性；音调变换通过调整音频的音高，生成不同音调的音频样本，丰富模型对音频音调特征的学习。

为了验证 L2 正则化与数据增强技术的有效性，我们在相同的小数据集上进行了对比实验。实验结果表明，采用 L2 正则化与数据增强技术优化后的模型，在验证集上的准确率波动明显减小，模型的泛化能力显著提升。在未采用优化策略时，模型在验证集上的准确率在 85% - 90% 之间波动；而采用优化策略后，模型在验证集上的准确率稳定在 92% 左右，且在测试集上的准确率也从原来的 88% 提升至 91%，有效改善了模型在小数据集上的过拟合问题，提高了模型的分类性能和稳定性。

5.3 轻量化模型设计展望

在实际应用中，尤其是在嵌入式设备和移动终端等资源受限的环境下，模型的轻量化设计至关重要。针对基于 LSTM 的音频分类模型在实际部署中的需求，本研究提出了模型轻量化的改进方向。

模型剪枝是一种有效的轻量化技术，它通过移除模型中不重要的连接或神经元，减少模型的参数数量和计算量。在基于 LSTM 的音频分类模型中，可以采用基于权重的剪枝方法，根据权重的绝对值大小来判断连接的重要性，将绝对值较小的权重置为零，从而实现模型的稀疏化。通过剪枝，模型的计算复杂度降低，内存占用减少，同时在一定程度上避免了过拟合。例如，在实验中，对 LSTM 模型进行剪枝后，模型的参数数量减少了约 30%，而准确率仅下降了 1% - 2%，在可接受的范围内。

量化技术通过将模型的权重和激活值从高精度数据类型转换为低精度数据类型，如将 32 位浮点数转换为 8 位整数，来减少模型的内存占用和计算量。在音频分类模型中，采用量化技术可以显著降低模型在嵌入式设备上的存储需求和计算资源消耗，提高模型的推理速度。同时，为了减少量化带来的精度损失，可以采用量化感知训练（Quantization - Aware Training，QAT）技术，在训练过程中模拟量化误差，使模型适应低精度表示，从而在保持模型性能的前提下实现模型的轻量化。

除了模型剪枝和量化技术，探索轻量级模型架构也是实现模型轻量化的重要途径。例如，将轻量级卷积神经网络架构（如 MobileNet）与 LSTM 相结合，构建 MobileNet - LSTM 模型。MobileNet 采用深度可分离卷积（Depth - wise Separable Convolution）等技术，大幅减少了卷积操作的计算量，在图像识别等领域表现出良好的轻量级特性。将其与 LSTM 结合，有望在保证音频分类性能的前提下，降低模型的计算复杂度和内存占用。在未来的研究中，可以进一步优化轻量级模型架构的设计，探索更适合音频分类任务的模型结构，拓展基于 LSTM 的音

频分类模型的实际应用场景，使其能够更好地满足资源受限环境下的应用需求。

6.结论与展望

6.1 研究结论

本研究深入探索了基于 LSTM 的音频分类方法，通过理论分析和实验验证，取得了一系列具有重要价值的成果。在特征提取阶段，运用 MFCC 特征提取技术对原始音频信号进行处理，成功将音频信号转换为适合模型输入的特征向量，有效保留了音频的关键特征，降低了数据维度，为后续模型训练奠定了坚实基础。

在模型构建方面，设计并实现了基于 LSTM 的音频分类模型，通过精心调整模型超参数，如设置 LSTM 隐藏层神经元数量为 128、批次大小为 32、训练轮数为 50、学习率为 0.001，并采用 Adam 优化器和交叉熵损失函数，使得模型能够充分学习音频特征与类别之间的映射关系。同时，引入 Dropout 正则化策略，有效防止了模型过拟合，增强了模型的泛化能力。

通过在公开音频数据集上的实验，基于 LSTM 的音频分类模型展现出卓越的性能。在准确率、召回率、F1 分数等评价指标上均显著优于传统机器学习模型 SVM 以及深度学习模型 CNN，准确率达到 92.5%，F1 分数达到 91.8%，充分证明了 LSTM 模型在处理音频序列数据时，能够有效捕捉长期依赖关系，具有更强的学习能力和表达能力。

然而，研究过程中也发现模型存在一些不足之处。在处理相似音频类别时，如古典音乐与爵士乐的区分，模型仍存在一定的误分类情况，这表明模型在对某些相似音频特征的学习和识别上还存在提升空间。

基于本研究的成果和不足，未来研究可从以下几个方向展开：一是结合多模态特征，进一步提升模型性能。除了音频的 MFCC 特征外，可融合音频的频谱特征、时频图特征等，同时考虑引入文本特征（如歌词），利用多模态信息的互补性，构建多模态音频分类模型，以更全面地捕捉音频的特征信息，提高分类准确率。

探索注意力机制与 LSTM 的深度融合，设计更高效的序列建模方法。注意力机制能够使模型更加关注音频序列中的关键部分，通过将注意力机制与 LSTM 相结合，能够进一步增强模型对重要信息的学习能力，提高模型在复杂音频分类任务中的表现。可以尝试不同类型的注意力机制，如自注意力机制、多头注意力机制等，并探索它们与 LSTM 的最佳融合方式，以提升模型的性能。

推动模型的工程化部署，实现音频分类的实时处理。在实际应用中，音频分类模型需要具备实时处理能力，能够在短时间内对大量音频数据进行准确分类。未来可结合边缘计算技术，将模型部署到边缘设备上，减少数据传输延迟，提高模型的响应速度，实现音频分类的实时应用，如智能语音助手、实时音频监控等。同时，研究模型的压缩和优化技术，降低模型的计算复杂度和内存占用，使其更适合在资源受限的设备上运行。

参考文献

[1] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[2] Graves A. Generating sequences with recurrent neural networks[J]. arXiv preprint arXiv: 1308.0850, 2013.

[3] Gers F A, Schraudolph N N, Schmidhuber J. Learning to forget: Continual prediction with LSTM [J]. Neural computation, 2000, 12(10): 2451-2471.

[4] Olah C, Carter S, Schubert L, et al. Understanding LSTM networks[EB/OL]. http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.
[5] Greff K, Srivastava R K, Koutník J, et al. LSTM: A search space odyssey [J]. IEEE transactions on neural networks and learning systems, 2016, 28 (10): 2222-2232.
[6] 张三，李四。基于深度学习的音频分类技术研究进展 [J]. 计算机科学，2022, 49 (5): 1-10.
[8] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]//Advances in neural information processing systems. 2014: 2672-2680.
[8] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15 (1): 1929-1958.
[9] Liu Y, Wang S. A review of deep learning for environmental sensing and monitoring [J]. Journal of Cleaner Production, 2020, 279: 123690.
[10] Yildirim E A, Yilmaz O H. Air pollution prediction using long short-term memory neural networks [J]. Journal of environmental management, 2018, 206: 1098-1106.
[11] Li X, Yang J. A novel air quality prediction model based on long short-term memory neural network [J]. Atmospheric Environment, 2018, 192: 146-156.
[12]Zhang Y, Liu Z, Zhang Y, et al. Short-term air quality forecasting using a long short-term memory (LSTM) network with meteorological data [J]. Atmospheric Environment, 2018, 194: 47-56.
致谢

在完成这篇基于 LSTM 算法实现音频分类的论文过程中，我得到了许多人的帮助，在此，我想向他们表达我最诚挚的感谢。

我要衷心感谢我的指导老师，从论文的选题、研究方向的确定，到模型的设计与优化，再到论文的撰写与修改，每一个环节都离不开您的悉心指导。您严谨的治学态度、深厚的学术造诣和敏锐的洞察力，让我在研究过程中少走了许多弯路。在我遇到困难和疑惑时，您总是耐心地为我解答，给予我宝贵的建议和鼓励，使我能够坚定信心，克服重重困难，顺利完成论文。您的言传身教将永远激励着我在学术道路上不断前行。

我也要感谢实验室的所有成员，感谢你们在实验过程中与我分享经验、交流想法，为我提供了许多宝贵的建议和帮助。感谢你们在我遇到问题时给予我的支持和鼓励，让我感受到了团队的温暖和力量。

最后，我要感谢我的家人，感谢你们一直以来对我的支持和理解，在我为论文忙碌时，给予我关心和鼓励，让我能够全身心地投入到研究工作中。你们的爱是我前进的动力，让我在学术道路上不断追求卓越。

在未来的学习和工作中，我将继续努力，不辜负大家对我的期望，为音频分类领域的发展贡献自己的一份力量。

|（注：文档部分内容可能由 AI 生成）

报告指标说明：

1.全文总相似比 = 复写率 + 自引率 + 他引率 + 专业术语。

2.复写率：指相似或疑似重复内容在全文中的比重。

3.自引率：指引用本人发表内容占全文的比重，需正确标注引用。

4.他引率：指引用他人内容占全文的比重，需正确标注引用。

5.专业术语率：指公式定理、法律条文、行业用语等在全文中的比重。

6.去除引用本人文献相似率：指去除本人发表部分后，相似或引用内容占全文的比重，需正确标注引用。

7.去除专业术语相似率：指去除专业术语后，相似或引用内容占全文的比重。

8.自写率：指原创内容在全文中的比重。

9.典型相似文章：指相似或引用内容占全文总相似比超过30%的文章。

总片段数量 = 相似片段 + 引用片段。相似片段中“综合”包括：《中文主要报纸全文数据库》《中国专利特色数据库》《中国主要会议论文特色数据库》《港澳台文献资源》《图书资源》《维普优先出版论文全文数据库》《年鉴资源》《古籍文献资源》《IPUB原创作品》。

须知：

- 报告编号系送检论文检测报告在本系统中的唯一编号。
- 本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

微信公众号