电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目 **基于神经网络的声音事件检测系统研究**

专业学位类别	工 程 硕 士
学　　号	**201952011734**
作者姓名	王雨江
指导教师	周军　　教　授
学　　院	信息与通信工程学院

分类号 _____ 密级 _____

UDC [注1] _____

学 位 论 文

基于神经网络的声音事件检测系统研究

（题名和副题名）

王雨江

（作者姓名）

指导教师 周军 教 授

电子科技大学 成 都

申请学位级别 硕士 专业学位类别 工程硕士

提交论文日期 2022 年 4 月 25 日 论文答辩日期 2022 年 5 月 27 日

学位授予单位和日期 电子科技大学 2022 年 6 月

答辩委员会主席 阎波

评阅人 熊金涛、杨海芬

注 1：注明《国际十进分类法 UDC》的类号。

Research on sound event detection system based on neural network

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline **Master of Engineering**

Student ID **201952011734**

Author **Wang Yu jiang**

Supervisor **Prof. Zhou Jun**

School **School of Information and Communication Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：王甬江　　日期：2022 年 6 月 4 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：王甬江　　导师签名：周军

日期：2022 年 6 月 4 日

摘 要

声音事件检测（Sound Event Detection, SED）是利用声音信号的特征去预测其声音事件种类的技术，它在智能家居、公共安全等领域具有较为广阔的应用前景。

传统的声音事件检测技术一般基于 GMM-HMM 模型，其识别准确率较低，且编解码计算复杂度较大，难以在实际生活中得到应用。与传统的机器学习方法相比，近年来国内外研究人员提出了基于神经网络（Neural Network, NN）的检测方法，显著提高了识别准确率。然而，基于 NN 的 SED 算法的一个主要问题是它们通常涉及大量参数和浮点运算数(floating point operations, FLOPs)，从而导致较高的处理延迟与硬件开销，使得基于 NN 的方法一般难以适用于要求低延迟和低存储的物联网设备。因此，构建网络复杂度低且识别准确率较高的声音事件检测算法成为本文的研究重点。论文设计了一种低复杂度高准确率的轻量级声音事件检测算法，并在该算法基础上实现了基于 FPGA-DPU 的声音事件检测系统。以下为本文的主要工作：

首先，由于目前声音事件检测算法存在参数量与 FLOPs 较高的问题，本文使用了一种选择性可分离卷积机制。该机制能够有效降低算法的参数量与 FLOPs，同时达到了较高的识别准确率。

然后，为了在维持低算法复杂度的基础上提高声音事件检测算法的识别准确率，论文使用了一种协调注意力机制。该机制基本不增加算法的复杂度，可同时作用于通道域、时域和频域，让检测算法重点关注与声音事件检测有关的特征和区域，减少对检测任务影响小的区域的关注。

随后，将本文中的轻量级声音事件检测算法通过 FPGA 的深度学习处理单元(DPU)进行了实现，从而构建了一个基于 FPGA-DPU 的声音事件检测系统。该系统基于 ZCU104 平台来开发设计的，通过使用 Vivado2020、DNNDK 与 PetaLinux 开发平台完成 DPU 的部署。

最后，在常用的声音事件检测数据集（ESC-50、ESC-10 和 UrbanSound8K）上进行了测试与分析。本文设计的轻量级声音事件检测算法的总参数量仅为 0.246M，算法的 FLOPs 仅为 203M，在 ESC-50 数据集准确率为 87.3%。在基于 FPGA-DPU 的声音事件检测系统中，ESC50 与 ESC10 数据集中单个音频平均识别时间为 8.24ms（UrbanSound8K 为 6.6ms），完全满足实时性的需求。

关键词：声音事件检测，神经网络，低复杂度

ABSTRACT

Sound event detection (SED) is a technology that uses the characteristics of sound signals to predict the types of sound events. It has a broad application prospect in smart home, public security and other fields.

The traditional voice event detection technology is generally based on GMM-HMM model. Its recognition accuracy is low, and the coding and decoding computational complexity is large, so it is difficult to be applied in real life. Compared with traditional machine learning methods, researchers at home and abroad have proposed a detection method based on neural network (NN) in recent years, which significantly improves the recognition accuracy. However, a major problem with NN based sed algorithms is that they usually involve a large number of parameters and floating point operations (FLOPs), resulting in high processing delay and hardware overhead, making NN based methods generally difficult to apply to IOT devices requiring low latency and low storage. Therefore, building a sound event detection algorithm with low network complexity and high recognition accuracy has become the focus of this thesis. This thesis designs a low complexity and high accuracy lightweight sound event detection algorithm, and implements a sound event detection system based on FPGA-DPU. The main work of this thesis is as follows:

Firstly, due to the high parameter and FLOPs in the current audio event detection algorithm, a selective separable convolution mechanism is used in this thesis. This mechanism can effectively reduce the parameters and FLOPs of the algorithm, and achieve a high recognition accuracy.

Then, in order to improve the recognition accuracy of voice event detection algorithm while maintaining low algorithm complexity, a coordinated attention mechanism is used. This mechanism basically does not increase the complexity of the algorithm, and can act on the channel domain, time domain and frequency domain at the same time, so that the detection algorithm can focus on the features and regions related to sound event detection, and reduce the attention to the regions that have little impact on the detection task.

Then, the lightweight sound event detection algorithm in this thesis is implemented by the deep learning processing unit (DPU) of FPGA, and a sound event detection system

based on FPGA-DPU is constructed. The system is developed and designed based on ZCU104 platform, and the DPU deployment is completed by using Vivado2020, DNNDK and PetaLinux development platforms.

Finally, it is tested and analyzed on the commonly used sound event detection data sets (ESC-50, ESC-10 and UrbanSound8K). The total parameters of the lightweight sound event detection algorithm designed in this thesis is only 0.246M, and the FLOPs of the algorithm is only 203M. The accuracy of the ESC-50 data set is 87.3%. In the audio event detection system based on FPGA-DPU, the average recognition time of a single audio in ESC-50 and ESC-10 data sets is 8.24ms (UrbanSound8K is 6.6ms), which fully meets the real-time requirements.

Keywords: Sound event detection, neural network, low complexity

目录

第一章 绪论

1.1 研究背景及意义

在各类型的信号中，声音包含了非常丰富的信息。声音事件检测（Sound Event Detection, SED）是利用声音信号的特征去预测其声音事件种类的技术。这里的声音事件不仅仅指人类产生的声音事件，还有自然界中的声音事件[1]。这些声音事件可能有不同的来源（例如人类、动物、植物）；这些声音事件可能有不同的长度与频率；这些声音事件可能有带有不同噪声的和不带有噪声的[2]。

声音事件检测可以在真实的环境中对特定的声音进行识别，如救护车鸣笛、枪响等需要被重点关注的声音[2,3]。在一些特定的情况下，声音检测相较于视觉检测会有更好的效果[4,5]。例如在监控系统中，多数情况下是使用视觉检测的方式，但是当出现逆光、暗光、光线不均匀等情况时，视觉检测的效果便会大幅度降低，此时便需要使用声音检测来提高检测效果。比如在智能家居场景中，可以使用声音监控设备监控厨房的声音，等到食物将要煮熟时进行提醒，这样便不需要在厨房等待食物煮熟；在家居监控时将视觉监控与声音监控结合起来，更能识别家电的工作情况，判断家电是否需要进行维修，同时还能对入室盗窃等突发情况进行预防。同时，声音的采集与存储相较于视频信号而言更加简单方便，因此在智能家居、智慧城市、智能交通等领域有着较为广阔的应用前景。再比如，当家长不方面实时就近照看婴儿与小孩时，可以使用有关音频检测设备，当婴儿或者小孩啼哭或发生意外时，家长能够实时收到信息从而快速赶到孩子身边。

该论文研究的方向是声音事件检测中单声音事件识别，即对只包含一种声音事件的音频进行识别分类。不同于传统的声音识别任务，例如，在音乐和语音识别方面，SED 在频率范围和时间相关性方面的已知知识非常有限。传统的声音事件检测技术一般是基于 GMM-HMM 模型，其识别准确率较低，且编解码计算复杂度较大，难以在实际生活中得到应用。与传统的机器学习方法相比，近年来国内外研究人员提出了基于神经网络（Neural Network, NN）的方法，显著提高了识别准确率。然而，基于 NN 的 SED 算法的一个主要问题是它们通常涉及大量参数和浮点运算数(FLOPs)。为了解决上述问题，提出了一个基于神经网络的轻量级的 SED 算法，和其他研究声音事件检测的算法相比，提出的算法拥有较少的参数量与 FLOPs，同时，该算法还拥有较高的识别准确率。

1.2 国内外研究现状

本文的研究的声音事件检测从类别上是属于单声音事件。常用到的公共数据集有 ESC-10、ESC-50、UrbanSound8K。

声音事件检测的相关研究发展时间较短，很多特征提取方法和识别器类型都是从语音识别借鉴过来的[6]。早期的声音事件检测主要是基于信号处理与传统的机器学习的方法，随着深度学习技术的发展，一些基于神经网络的技术也逐渐应用于声音事件检测中。

在声音事件检测的研究初期，多数对于声音事件检测的研究是使用的传统机器学习方法。在传统的机器学习中，首先是对声音信号进行初步的特征提取，然后再以这些特征作为输入实现分类。例如， Y. Wang 等人从声音信号中提取了很多的低级的特征，然后从低级的特征中获取高维的特征[7,8]。但传统的使用人工提取高维特征的方法并不可取，提取的高维特征往往会有较大的冗余。因此特征提取后一般会进行降维操作，比如可以采用独立成分分析方法对声音特征进行降维[9]。1970-1980 年代，声音事件检测方法主要是模板匹配方法，比如高斯混合模型(GMM)[10]和隐马尔可夫模型(HMM)[11]以及两者的结合。该方法虽然有效，但编码解码时计算量巨大。

在 20 世纪 80 年代的后期，随着计算技术和存储技术的发展，人工神经网络等技术逐渐兴起，也在声音事件检测领域得到应用。随机森林[12]以及带有的支持向量机[13]等机器学习方法在早期声音事件检测领域也取得了不错的效果。

进入 21 世纪，深度学习技术开始得到发展，并且在声音事件检测等领域得到应用，较大地促进了声音事件检测的技术的发展。最先是有研究人员使用深度神经网络(DNN)去进行声音信号的特征的提取，与运行效率低且难以准确提取声音信号特征的人工提取方式相比，DNN 可以更准确快速地提取声音信号的特征，它提高了复杂声音事件检测的准确率，因此在声音识别方面获得了广泛的应用。Deng 等人在文献[14]中使用 DNN 改进了由 GMM 和 HMM 组成的 GMM-HMM 模型，提出了 DNN-HMM 模型并取得了更好的检测效果。

近年来，卷积神经网络(CNN)在声音事件检测领域取得了显著的成功[15-17]。根据声音事件检测系统的要素，有关研究大多围绕声音特征的选择与提取、神经网络的设计、数据的预处理三个部分展开。在特征的选择与提取上，大量文献对其进行了有关讨论，其中：

文献[18]提出了一个声音事件识别(SED)算法，该算法由多个特征通道组成，作为具有注意力机制的深度卷积神经网络(CNN)的输入。该论文的新颖之处在于使用了多个特征通道，包括梅尔频率倒谱系数（MFCC）、伽马调频率倒谱系数（GFCC）、

恒定 Q 变换（CQT）和色度图。作者认为，各个不同的特征能关注到声音信号的不同频率信息，例如 GFCC 对比 MFCC 能够更好地描述瞬态声音；CQT 在音频场景分类方面比 MFCC 更容易捕获中低频率的信息；色度图则是能够更好分析音高信息。

文章[19]提出的算法也拥有多个特征，但是与文章[18]不同的是，作者是将多个特征以线性组合的方式形成特征集。作者认为由于 log-mel 谱图和梅尔倒数系数 MFCC 是声音识别中应用最广泛的听觉特征，因此首先提取这两个特征集。然后，通过 Librosa 库提取色度、光谱对比度和 tonnetz。Log-mel 频谱图、色度、光谱对比度和 tonnetz 聚合形成 LMC 特征集，MFCC 与色度、光谱对比度和 tonnetz 组合形成 MC 特征集。

文章[20]提出了一个可学习的听觉滤波器组，它基于一维(1D)卷积神经网络，具有 gammatone 滤波器组形式。作者认为，过去，已经提出了许多基于可学习听觉特征的 ESC 方法，这些方法是通过对原始输入波形执行简单的一维卷积获得的，以优于传统的手工特征，例如梅尔频率滤波器组。这里提出了一个可学习的 gammatone 滤波器组层，该层由带通 gammatone 滤波器的参数形式表示的 1D 内核组成，用于获取原始波形的时频表示。

文献[21]中，提出了一种端到端的环境声音分类方法。这个方法将声音信号作为卷积神经网络的输入，可以直接从声音信号中进行学习。几个卷积层用于捕获信号的精细时间结构并学习与分类任务相关的各种过滤器。所提出的方法可以处理任何长度的音频信号，因为它使用滑动窗口将信号分成重叠的帧。评估了考虑多种输入大小的不同架构，包括使用 Gammatone 滤波器组对第一个卷积层进行初始化，该滤波器组对人类耳蜗中的听觉滤波器响应进行建模。在 UrbanSound8K 数据集上评估了所提出的端到端方法在环境声音分类方面的性能，实验结果表明它达到 89% 的平均识别准确率。

在文献[22]中，同样提出了学习的声音模型直接从原始语音波形中获取特征的方法。但作者认为：当前的基于原始语音波形的模型通常使用时域卷积层来实现提取特征，单个大小提取的特征过滤器不足以建立鉴别音频类别的特征。因此，作者提出了一种多尺度卷积的方法，通过提升相应的频率分辨率与学习滤波器的所有频率区域大小，最终可以得到更好的音频表示。同时，在基于波形的特征和基于频谱的特征，作者还使用了两种不同的相位融合方法，最终形成了一个端到端的网络。在环境声音分类数据集 ESC-10 和 ESC-50 上进行了测试，均取得了不错的识别效果。

在网络模型方面，有网络集成、迁移学习、注意力机制和可分离卷积等有关方法。例如：

在文献[23]中，作者提出了将两个模型集成起来，构建一种新模型。具有 log-mel 音频表示的卷积神经网络(CNN)和基于 CNN 的端到端学习都已用于环境事件声音识别(ESC)。但是，log-mel 特征可以通过有效的融合方法从原始音频波形中学习到的特征来补充。文章提出了一种新颖的堆叠 CNN 模型，该模型具有多个递减滤波器大小的卷积层，以提高具有 log-mel 特征输入或原始波形输入的 CNN 模型的性能。然后使用 Dempster-Shafer (DS) 证据理论将这两个模型相互结合起来，为 ESC 构建集成 DS-CNN 模型。最后在场景的三个公共数据集上的实验结果表明，该识别方法可以实现比其他相同类型输入特征 CNN 模型更高的性能。从而表明基于 log-mel 特征输入的模型和基于直接从原始波形学习特征的模型是有一定互补性的。

文献[24]提出了一个具有时间保留的多流卷积神经网络。环境声音分类系统在不同的声音分类任务和不同时间结构的音频信号中往往不稳定。该网络依赖于由原始音频和频谱特征组成的三个输入流，并利用一个由能量随时间变化计算出的时间注意函数。结果表明，在三个常用到的环境声音和音频场景分类数据集上具有良好的准确率。但是，该神经网络存在网络参数量与运算量过大的问题。

文献[25]提出了基于简单的对数幂短时傅里叶变换(STFT)频谱图的算法，并将它们与图像中的几种知名方法（ResNet 与注意力机制）相结合。采用迁移学习的方法，利用图像数据集进行预训练以解决环境声音检测数据集中数据量不足的问题。调查跨域预训练、架构更改的影响，最后在 ESC 数据集与 UrbanSound8K 数据集上进行测试，取得较好的识别准确率。

文章[26]提出了将时间注意力和通道注意力结合起来的 CNN 算法，该算法可以通过生成互补信息来增强 CNN 的代表性。同时，该文章还提出了一种数据增强的方法，将两个音频按线性混合后组成一个新的音频作为网络的输入。从而避免了由于有限的训练数据可能导致的过度拟合。

文献[18]还使用了可分离卷积，分别在时间域和特征域上工作，以降低网络的复杂度，从而大幅减少模型的参数量与运算量。

在数据增强方面，有基于原始数据集进行时域或频域变换的，也有进行声音混合的，同时，还有利用 GAN 网络生成新的数据的方式。例如：

文献[27]探讨了对原始数据集作时间偏移、音调偏移、动态范围压缩和背景噪声的数据增强后的对网络识别性能的影响。结合数据增强，所提出的模型产生了更先进到的结果。最后，检查了每个增强对每个类的模型分类准确度的影响，并观察到

每个增强对每个类的准确度的影响不同，这表明通过应用类，可以进一步提高模型的性能。

文献[28]提出了基于卷积神经网络的环境声音分类算法，并使用生成对抗网络（GAN）来增强音频数据。结果表明，相比较于传统的时间偏移等数据增强方法，利用 GAN 进行数据增强，所提升的准确率更高。

此外，还有将类间学习、多任务学习和联邦学习等有关方法运用到声音事件检测领域，并取得了不错的效果。

文献[29]提出了在声音事件检测上使用类间的方法。作者提出了一种新的深度声音识别的学习方法：班间学习(BC 学习)。该方法是通过识别类间的声音为类间的声音来学习一个具有区分性的特征空间。作者首先通过以随机比例混合两个属于不同类的声音来生成类间的声音。然后，将混合的声音输入到模型中，并训练模型输出混合比。BC 学习的优点不仅仅局限于训练数据变化的增加；BC 学习导致了特征空间中 Fisher 准则的扩大，并使类的特征分布之间的位置关系得到了正则化。实验结果表明，BC 学习提高了在各种声音识别网络、数据集和数据增强方案上的性能，其中 BC 学习始终是有益的。此外，作者构建了一个新的深度声音识别网络(EnvNet-v2)，并通过 BC 学习对其进行训练。实验结果表明，该方法取得了超过人类的水平。

文献[30]中提出了使用两个长短期记忆网络（LSTM）的新型人工智能模型来进行声音识别的检测任务。该模型在输入时同时使用了原始的语音数据和从原始语音中提出的特征向量，并使用迁移学习进行增强的方法。在最终的输出部分，使用到了融合决策的方法，从两个长短时记忆网络中选择输出结果最好的进行输出，最终取得了不错的实验结果。

文献[31]中提出了使用多任务学习的方法，将声音事件检测(SED)与声学场景分类(ASC)进行起来。一些作品基于多任务学习（MTL）对声音事件和声学场景进行了联合分析，其中声音事件和场景的知识可以帮助它们相互估计。传统的基于 MTL 的方法利用单热场景标签来训练声音事件和场景之间的关系;因此，传统方法无法模拟声音事件和场景的相关程度。但是，在真实环境中，某些声学场景中可能会发生常见的声音事件;另一方面，一些声音事件只发生在有限的声学场景中。在文章中，作者提出了一种基于 SED 和 ASC 的 MTL 的 SED 新方法，该方法利用声学场景的软标签，使能够模拟声音事件和场景的相关程度。

文献[32]中利用了 ESC-50 数据集的特点，针对该声音事件数据集提出了一个新的识别网络。文章的新颖之处在于将分层声音分类嵌入到分类过程中。由于 ESC-50 数据集包含 50 个精细的声音类别，5 个粗略的声音类别。在最终输出分支上产

生 5 分类与 50 分类两个分支，最终将各种不同的分支情况的模型结合起来，取得了不错的识别效果。

在文献[33]中，作者考虑了所需的音频数据集的采集会涉及到大量的隐私问题。因此作者使用了分布式特性使联邦学习（FL）方法去解决利用大规模数据同时缓解隐私问题。由于之前没有关于 SED 的 FL 的研究，为了解决这一差距并促进该领域的进一步研究，作者在家庭和城市环境中为 SED 创建并发布了新颖的 FL 数据集。此外，文章在三个深度神经网络架构的 FL 上下文中对数据集进行基线结果。结果表明，FL 是一种很有前途的 SED 方法，但面临着分布式客户端边缘设备固有的不同数据分布的挑战。

在文献[34]中，作者考虑到声音事件检测中研究缺乏与任何建议方法进行比较的基准结果的困难。作者采用了基于高斯混合模型 GMM 的分量描述，并使用了线性与非线性的支持向量机的描述方式。

文献[35]中，作者针对基于深度学习的声音事件检测系统的模型参数与消耗资源逐渐增大的问题，提出了自己的解决方法。为了解决模型参数与消耗资源逐渐增大这个问题，作者首先建立了一个自定义数据库，这其中包含智能家居或建筑物内外发生的事件。声音事件：如雨，风，人类步态和车辆的通过。在建立数据库之后，作者使用了顺序特征选择技术来减少用 MFCC 提取的特征的尺寸。所选特征用于在上述声音事件上训练递归神经网络 （RNN）。最后将所提出方法的结果与使用 MFCC 特征训练的相同 RNN 和用 mel 频谱（MFB）特征训练的卷积神经网络（CNN）进行了比较。最终表明提出的系统在前一种情况下表现准确，但与 CNN 相比，在使用所提出的系统进行训练期间实现了更高的分类精度和显着减少的参数方面略好。

在声音事件检测应用到物联网领域中，文献[36]提出了自己的端到端基于物联网的声音事件分类系统。作者首先使用音频设备收集的有关环境声音的元数据，然后收集到的环境声音数据传输到基于云的平台并聚合在一起，最终得以产生详细的描述性分析和可视化。最后在云平台中使用了神经网络进行声音事件分类，从而实现了声音事件检测在物联网领域的应用。

针对于嵌入式设备的应用，文献[37]自己的优化策略。因为当前的嵌入式设备存储资源有限且处理器性能较低，因此需要针对于嵌入式设备开发适合其部署的神经网络。在本文中，作者提出了基于 PhiNets 的新型神经架构，用于微控制器单元上的实时声学事件检测。本文所提出的模型易于扩展，以满足硬件要求，并且可以在声音特征提取后的频谱图和原始语音波形上运行。其中，在频谱图输入时，作者使用了最大池化的方法而非跨步卷积的方法对特征图进行下采样，同时将原始

语音波形的输入块替换为 2D 卷积。而针对于原始波形输入的方式，提出了一种适用于波形的架构变体，从而利用了一维卷积。最终，作者在 UrbanSound8K 数据集上进行了测试，测试结果表明系统在 UrbanSound8K 上的识别准确率达到了 77%，但是其参数量只有 27K，其运算量仅有 43M，能够满足嵌入式设备对于存储资源与计算资源的要求。

综上所述，现有的基于神经网络的声音事件检测算法已经可以达到较高的识别准确度，但是它们涉及大量的参数和 FLOPs[38-41]，导致算法需要大量的存储资源，并且算法的延迟会比较高。例如，基于 CRNN 的 SED 算法[42]可以实现较高的识别准确率，但其 FLOPs 大于 4G，参数量大于 15M。在物联网应用场景中，通常都需要要求低功耗、低存储、高准确率。为解决上述问题，论文提出了一个基于神经网络的轻量级的声音事件检测算法，该算法采用了选择性可分离卷积和协调注意力机制的设计方法，使得其能在拥有较少参数量与 FLOPs 的基础上还拥有比较高的准确率。

1.3 本文主要内容

本文针对声音事件检测系统的特征提取、架构设计、性能优化与实现进行了相关研究。讨论了一种基于神经网络的声音事件检测算法，该算法选择卷积神经网络 CNN 作为分类网络架构，同时利用了协调注意力机制强化特征提取，提升识别准确率；在此基础上，还使用选择性可分离卷积来降低系统的复杂度。并在该算法的基础上实现了基于 FPGA-DPU 声音事件检测系统。

本文一共包括六个章节，每章节的大致内容如下所述：

第一章首先介绍了声音事件检测的研究背景与意义，然后在此基础上阐述了声音事件检测的发展历史与研究现状，最后说明了本文的研究工作和章节安排。

第二章主要介绍了声音事件检测的基本原理。首先介绍了声音事件检测的系统架构，然后就各种声音信号的特征提取方法进行说明，最后阐述了常见的声音事件检测模型所涉及到的技术原理。

第三章为基于神经网络的声音事件检测算法设计。本章节详细论述了提出的轻量级声音事件检测算法。其中，对提出的选择性可分离卷积机制与协调注意力机制进行了重点阐述。

第四章为基于 FPGA-DPU 的声音检测系统设计。将提出的轻量级声音事件检测算法通过 FPGA 的深度学习处理单元(DPU)进行了实现，从而构建了一个基于 FPGA-DPU 的声音事件检测系统。主要介绍了 DPU 的有关知识与实现过程。

第五章为实验结果与分析。首先对实验数据及开发环境进行简单介绍，然后简单介绍数据增强和模型训练的设置。最后重点阐述所提出的声音事件检测算法的实验结果与相应的分析。

第六章为总结与展望。简单总结本文的有关研究内容，并对接下来的声音事件检测领域的研究工作提出了展望。

第二章 声音事件检测基本原理

声音事件检测（Sound Event Detection, SED）是利用声音信号的特征去预测其声音事件种类的技术，它是计算机听觉领域中的一个重要的研究方向。声音事件检测主要就是给定一个待检测的声音信号，经过特征提取后使用神经网络等方法去预测它所对应的声音事件类别。

本章首先介绍基于神经网络的声音事件检测算法的基本框架，然后对各种常见的声音特征提取的方法进行了介绍。最后，为了方便后序章节的论述，对常见的几种声音事件检测算法进行了介绍。

2.1 声音事件检测算法介绍

声音事件检测是针对真实环境声音，利用声音信号的特征去预测其声音事件种类。基于神经网络技术的声音事件检测算法框架如图 2-1 所示。算法中一般包括音频输入、声音特征提取、神经网络、系统输出等。

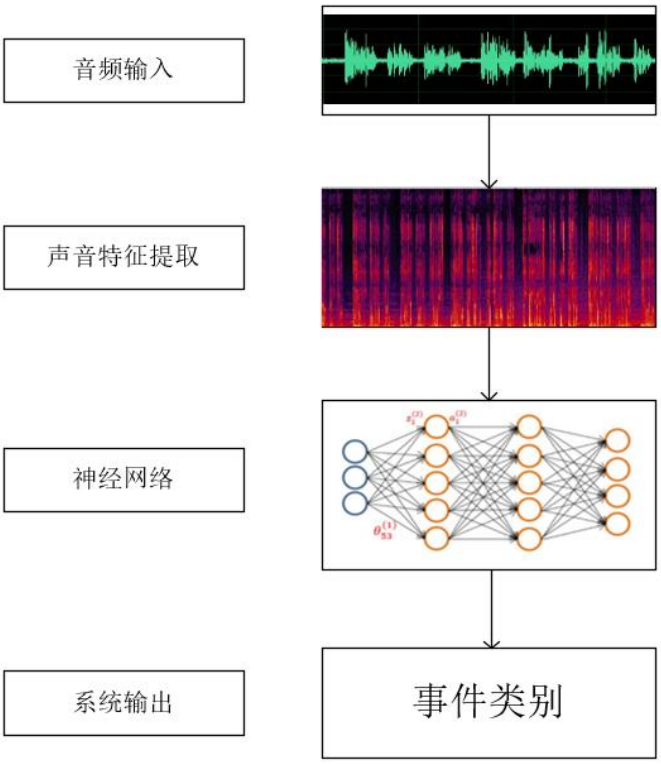

图 2-1 声音事件检测算法框架

声音事件检测算法一般会包括两个阶段，第一个阶段就是训练的阶段，完成声音事件检测算法的设计和训练,第一个阶段的输入为训练集与对应的事件标签，输出为音频对应的事件类别；第二个阶段就是测试阶段，利用第一个阶段的算法对声音信号的类别进行预测，第二个阶段的输入为测试集，输出为神经网络预测的音频的声音事件类别。图 2-2 显示了声音事件检测算法的两个阶段。具体过程总结如下：

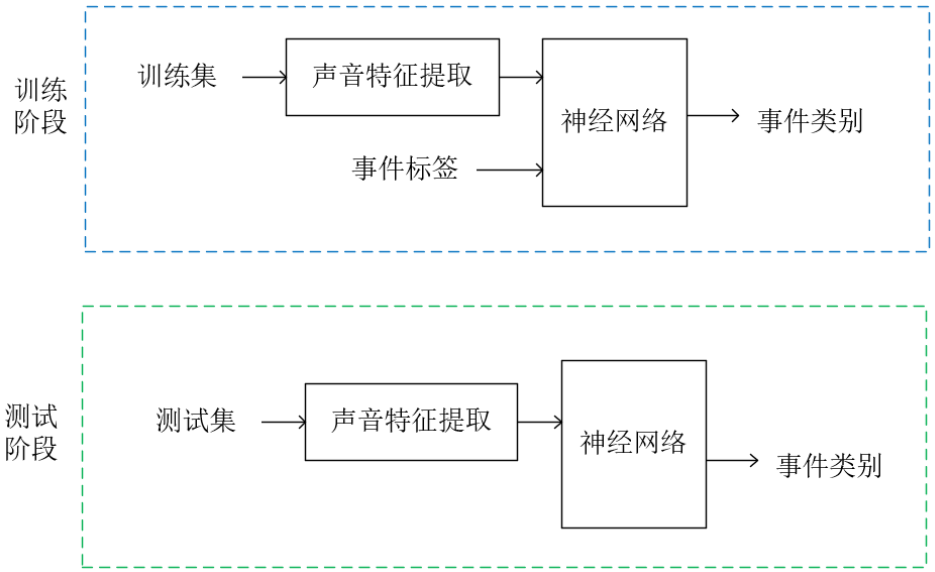

图 2-2 声音事件检测算法的两个阶段框架图

（1）训练阶段

训练阶段主要有两个部分，即设计声音事件检测算法和对算法进行训练。设计声音事件检测算法就是设计一个接收声音信号，输出该声音事件对应类别的算法。具体来说，声音信号经过分段和声音特征提取可以得到一个频谱图，该频谱图便是神经网络的输入，它是一个二维矩阵结构，其中横轴代表时间，纵轴代表着频率。然后把这频谱图送入到神经网络中，使用 CNN、RNN、CRNN、LSTM 等深度学习模型进行分类，最终得到该音频的声音事件类别。

在设计声音事件检测算法之后，想要得到分类效果更好的算法，便需要对设计的算法进行训练以得到分类效果更好的算法，一般是通过对参数进行更新而实现的。在对声音事件检测算法的训练中，输入的是声音信号数据集及所对应的事件标签，神经网络接收到的是声音信号经过初级特征提取而得到的频谱图与声音信号所对应的事件标签。通过反向传播算法可以进行模型参数更新[43]，然后使用一些优化算法，便可以缩小神经网络的预测值与真实值之间的差距。当预测值和真实值之间的差距缩短到一定范围内，即认为得到一个比较可靠的声音事件检测算法。

（2）测试阶段

在声音事件检测算法的测试阶段，使用到的是没有对应事件便签的声音信号，将它输入到声音事件检测算法中，最终得到检测算法的预测的声音事件类别。在这个阶段，声音信号经过和前面的训练阶段相同的分段和声音特征提取后得到一个频谱图，然后将该频谱图送入到训练好的声音事件检测算法中，经过对应神经网络的处理，最终得到声音事件检测算法所预测的声音事件类别。

2.2 声音特征提取

目前，大多数声音事件检测算法不采用将声音信号作为神经网络的输入，而是选择将声音信号经过声音特征提取后得到的频谱图作为神经网络的输入。声音信号直接作为神经网络的输入会有两大缺点。第一，一些使用神经网络直接接收原始声音信号比较复杂，且准确率不高；第二，目前音频的采样率都较大，一般都是32kHz、44.1kHz 等，音频数据量较大，不适合作为神经网络的输入。因此，目前声音事件检测算法会先将声音信号经过声音特征提取得到一个二维频谱图，进而降低神经网络的输入数据量并得到更高的准确率。本小节将总结常见的声音特征提取：STFT 声谱图、小波变换特征、梅尔频谱特征与梅尔倒谱系数。

2.2.1 STFT 声谱图

与图片数据不同，声音传递的信号是一种一维的时域上的信号，但是如果仅仅只在时间域上进行观察，难以发现声音信号特征。而在频域上，能够比较好地观察声音信号的特征，所以一般是在频域上对声音信号进行分析。其中傅里叶变换分析方法可以完成音频信号在时域和频域之间的互相转换[44]。但傅里叶变换也有一定的局限性，它对于在分析的时间范围上频率稳定的信号，它对声音信号的分析效果良好。因此，有必要对信号进行时频分析，即不仅要考虑信号在时域的特性，还要考虑信号在频域的特性。

短时傅里叶变换(STFT) 是一个用来处理声音信号的方法，它即可以在时间域上对声音信号进行分析，也可以在频域上对其进行分析。它内部利用了傅里叶变换，使得它可以在频域上对信号进行分析，同时它也保留了时间域上的有关信息。实际上，STFT 就是将一段长时间的声音信号经过分帧处理，变成若干个短时间的声音信号，然后进行傅里叶变换。STFT 的有关过程大致如下：

（1）分帧加窗

一般来说，时间较长的声音信号的各个时间段上所对应的声音波形通常是不同的，即认为该声音信号是长时不平稳信号。根据经验，认为比较短的时间的信号

是平稳的信号，这个时间一般为 15-80ms。所以将时间比较长的声音信号进行分帧，帧的长度一般为 15-80ms。为了使相邻两帧的信号之间有一定的连续性，一般相邻两帧数据设定一定的重叠。为了对信号进行截取，通常会对每一帧数据进行加窗，一般使用汉明窗，避免了频谱泄露以及吉布斯现象。

（2）短时傅里叶变换

在分帧加窗后，还需要对数据进行处理，即对各个帧的数据进行傅里叶变换。离散场景中，公式(2-1)为其计算表达式。

$$STFT[x[n]](m,k)=\sum_{n=0}^{L-1}x[n]w[n-m]e^{-j\frac{2\pi kn}{L}},k=0,1,2,\dots,L-1 \tag{2-1}$$

其中，x[n]代表信号在时域上的第 n 个点所对应的值，w[m]代表一个长为 m 的窗，L 代表一帧数据的点数。

（3）STFT 声谱图生成

声谱图有很多种，常见的有以下几种，即幅度谱、功率谱和 Log 功率谱。幅度谱指的是幅度，一般是将短时傅里叶变换的结果取绝对值。对于功率谱而言，就是对经过短时傅里叶变换后的特征进行平方处理和取平均处理。对于 Log 功率谱而言，就是对功率谱取对数。最后，把经过处理得到的特征按它对应的帧的时间顺序进行排列便得到了最终的声谱图。

一般而言，得到的声谱图的水平方向轴便是指它的时间，垂直方向轴便是指它的频率，颜色的深浅则代表其能量的大小。不同声音信号对应的声谱图有所不同，一个声音信号的 STFT 声谱图如图 2-3 所示。

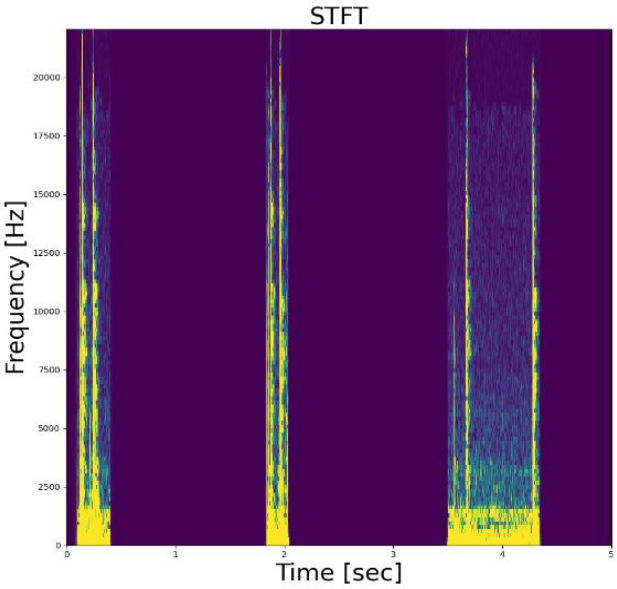

图 2-3 一个声音信号的 STFT 声谱图

从图中可以看出，该声音信号为非连续的短时信号，而不是连续的长时信号。在 STFT 中，每一帧的大小一般都是固定的。如果每一帧的长度太短，会导致频率的分辨率较差；若每一帧的长度太长，会导致时间域上的分辨率变差。对于低频信号，有可能连一个周期都不能覆盖；对于高频信号，可能覆盖了多个周期，不能很好的反映信号变化。

2.2.2 小波变换特征

短时傅里叶变换可进行时频分析，小波变换（wavelet transform，WT）也常作为时频分析的工具[45]。小波变换是一种新的对声音信号进行分析的方法，它与 STFT 类似，都是将时间较长的信号分割为时间较短的信号，同时它还可以根据频率的高低对分割的信号的长度进行改变。它可以重点分析信息特征丰富的区域，通过一些伸缩与平移的计算对信号进行多个不同尺度的分析，最后可以得到在高的频率的信号部分在时间上会细分，在低的频率的信号部分在频率上回细分。能够根据信号的特征，对不同频率信号有不同的聚焦。

小波变换的函数与傅里叶变换的函数有所不同，小波变换采用的函数是有限长的具有衰减性质的小波基而非无限长的正弦函数力[45]。公式(2-2)为其所对应的计算公式。

$$WT(\alpha,\tau)=\frac{1}{\sqrt{\alpha}}\int_{-\infty}^{+\infty}f(t)*\phi\left(\frac{t-\tau}{\alpha}\right)dt \tag{2-2}$$

其中，α是和尺寸有关的变量，它反应小波变换的伸缩量。τ为平移量，它反应了小波变换在时间上的平移量。使用小波变换，最后可以得到在高的频率的信号部分在时间上会细分，在低的频率的信号部分在频率上回细分。

2.2.3 梅尔频谱特征

梅尔频谱特征与梅尔频率倒谱系数 MFCC 特征是语音识别方向最常见的一类特征，它们在声纹识别[46]以及声音事件检测[47,48]等领域有着广泛的应用。本节重点介绍梅尔频谱特征，MFCC 特征将会在下一个章节进行介绍。

由对人耳的听觉特征的研究，发现使用 Mel 频率更加符合实际的人耳听觉，即在 1000Hz 之下呈现一种线性关系，在 1000Hz 之上呈现一种对数关系。对于频率比较接近的声音，人耳一般是不能分辨出来的，这就是临界带宽的概念[49]，当两个声音的频率差距到一定范围时，人耳才可以分辨出来。公式（2-3）为 Mel 频率与 Hz 频率（即实际频率）的关系表达式：

$$f_{mel}=2595*\lg\left(1+\frac{f}{700}\right) \tag{2-3}$$

Mel 频率与 Hz 频率对应关系图如图 2-4 所示。其中，横轴代表的是实际频率，即 Hz 频率，竖轴代表的是 Mel 频率。

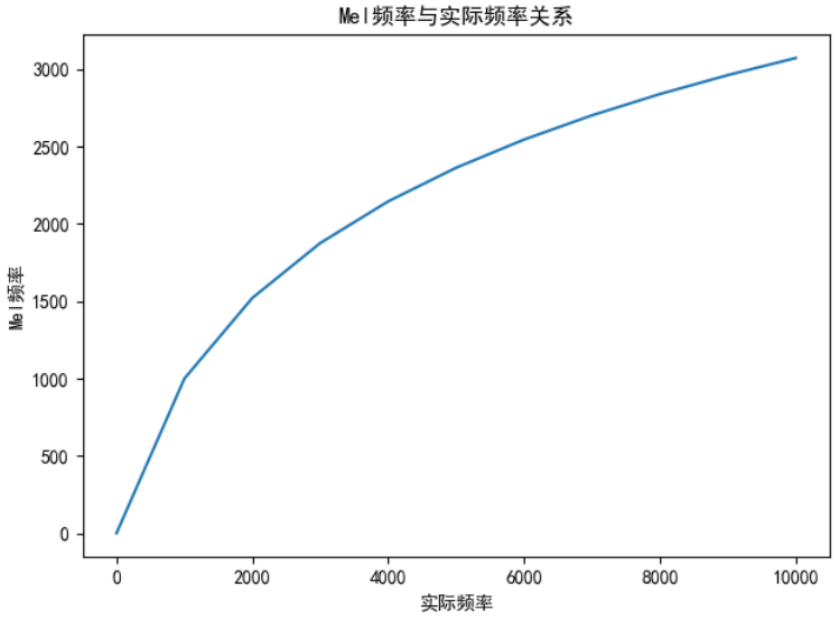

图 2-4 Mel 频率与 Hz 频率对应关系图

从上图中可以看出，当实际频率较低的时候，即大概在 1000Hz 以下的范围，Mel 频率与实际频率基本上表现为线性相关。而当实际频率较高的时候，即大概在 1000Hz 以上的范围，Mel 频率与实际频率之间的关系转为由之前的线性相关对数相关关系。

图 2-5 为梅尔频谱特征的获取流程图。主要包括了预加重、分帧、加窗、快速傅里叶变换、求能量谱、梅尔滤波和取对数等有关部分，最终可以获得梅尔频谱特征。

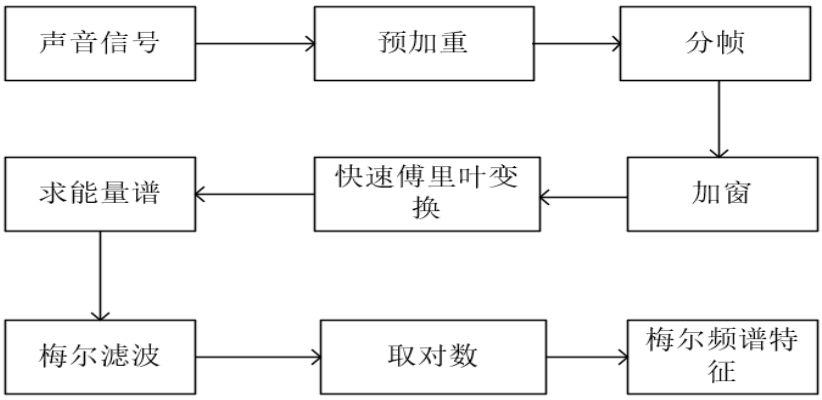

图 2-5 梅尔频谱特征的获取流程图

梅尔频谱特征的具体计算如下：

（1）预加重

目前，发声系统在发出声音信号的时候，高频率容易被抑制，因为需要使用预加重进行补偿。一般来说，就是乘以一个在高频率处大而低频率处小的系数，这样频率比较高的部分就可以得到补偿。预加重就是过滤得到较高频率的部分，即高通滤波器，公式(2-4)为预加重的计算公式

$$H(z) = 1 - \alpha z^{-1} \tag{2-4}$$

（2）分帧

类似于 STFT，将长的信号利用分帧的方法变成一个个短的信号，然后使用例如快速傅里叶变换的方法去处理音频信号。每一帧的长度一般为 20~40ms，相邻两帧之间会有一定的重叠。

（3）加窗

是为了平滑信号，防止频谱泄漏，故需要进行加窗，一般采用的是汉明窗。公式(2-5)显示了它进行计算的过程。

$$w(n,\alpha) = (1-\alpha) - \alpha\cos(2\pi\frac{n}{N-1}) \tag{2-5}$$

在表达式中，N 代表的是所使用的汉明窗的长度，$0 \le n \le N-1$，α为汉明窗的参数。一般来说，α取 0.46。

（4）快速傅里叶变换

快速傅里叶变换是一种有效地获取声音信号的频谱信息的方法，它可以很好得将声音信号从时间域上转化到频域上。公式(2-6)显示了它具体的计算流程。

$$X_i(k) = \sum_{n=0}^{N-1} x_i(n) e^{-j\frac{2\pi nk}{N}}, 0 \le \mathrm{k} \le \mathrm{N}-1 \tag{2-6}$$

$x_i(n)$：信号在时间域上的第 i 帧第 n 个时间点，$X_i(k)$：信号在频域上的第 i 帧的第 k 个频率点。

（5）求能量谱

根据上述结果，计算信号的每个帧的每个频率点的能量。一般来说，每个帧信号的能量谱是通过将快速傅里叶变换的输出结果平方得到的。公式(2-7)显示了其计算过程。

$$S_i(k) = |X_i(k)|^2, 0 \le k \le N-1 \tag{2-7}$$

其中，S_i代表第 i 帧信号的能量谱

（6）梅尔滤波，取对数

经过上述处理的频域信号会有很多，难以直接用来表示声音特征，因此需要用滤波器组对频域信号进行简化，一般将一段的频域用一个值来表示。具体计算为：使用上述 FFT 得到的幅度谱，然后和梅尔滤波器组里的滤波器一个个对应相乘相加，这样便得到了对应滤波器在一个频域的能量值。如果滤波器有 32 个，那么就会得到 32 个能量值。图 2-6 为一种梅尔滤波器组图。

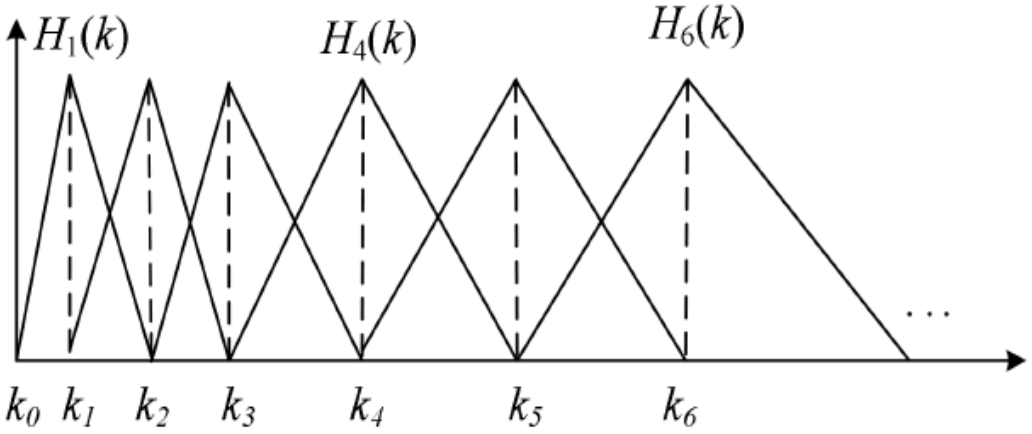

图 2-6 一种梅尔滤波器组图

得到能量值后，由于人耳对声音的频率(即 Hz 频率)并不是线性感知的，故需要经过公式(2-3)进行频率转换，然后才可以进行倒谱分析。

从图 2-6 中可以看出，在频率比较低的部分，梅尔滤波器的数量较多，分布相对密集；在频率比较高的部分，滤波器组的数量较少，分布相对稀疏。这也是比较符合人耳的实际听觉特征，即对低频率的声音更加敏感。

2.2.4 梅尔倒谱系数

计算梅尔倒谱系数 MFCC 特征的方法和计算梅尔频谱特征的方法差不多，也包括了分帧加窗、快速傅里叶变换、求能量谱、梅尔滤波和取对数等部分，只是在最后加了 DCT 的计算。图 2-7 为 MFCC 获取流程图。

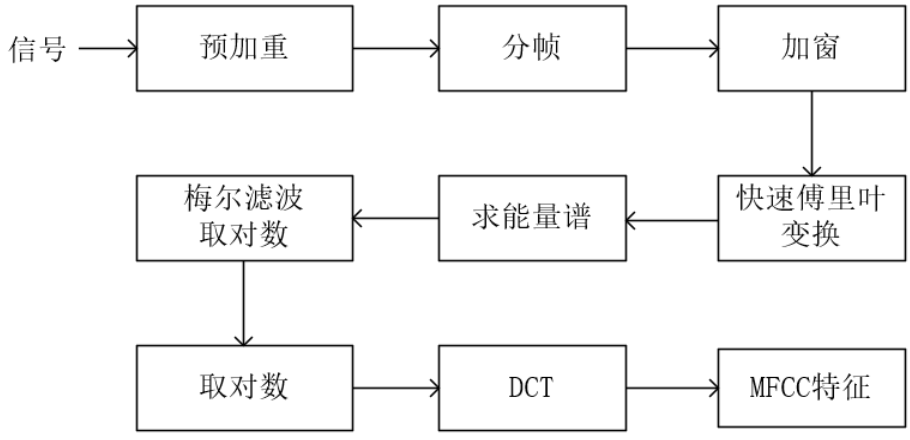

图 2-7 MFCC 获取流程图

未来获取低频信号的信息，这里使用比较简单的 DCT 变换来进行操作。从上一小节可以知道，相邻的滤波器之间的覆盖的频率范围有一定的交叉，所以滤波器

输出的值之间会有相关性，故要使用 DCT 对数据进行进一步的压缩，最后便可以得到需要的特征参数。DCT 的变换公式如公式(2-8)所示。

$$C(n)=\sum_{m=0}^{M-1}s(m)\cos(\frac{\pi n(m-0.5)}{M}),n=1,2,\dots,L \tag{2-8}$$

上述公式中，L： MFCC 特征个数，$C(n)$：DCT 变换的第 n 个系数，M：梅尔滤波器的个数，$s(m)$：第 m 个滤波器的能量。

2.3 声音事件检测算法

声音事件检测算法主要有传统模型与深度学习模型两大类。本节主要阐述传统模型中的 GMM-HMM 模型，深度学习模型中的 CNN、RNN 模型。

2.3.1 传统模型

声音识别框架中的经典的模型是：GMM-HMM 模型。在 GMM-HMM 模型中，一般以音素为单位进行建模。假设一个词句识别任务，那么它的一般运算流程如下：首先输入需要进行识别的音频信号，然后将连续的音频信号拆分为若干个语音帧，然后对每一帧进行特征计算，然后将每一个特征经过 GMM 进行运算得到对应的观察序列概率，将观察序列概率与对应的状态转移图结合起来，使用维特比搜索的方法，最后将音素组合成单词，单词连接起来成为一个句子。图 2-8 为 GMM-HMM 进行语音识别框架图：

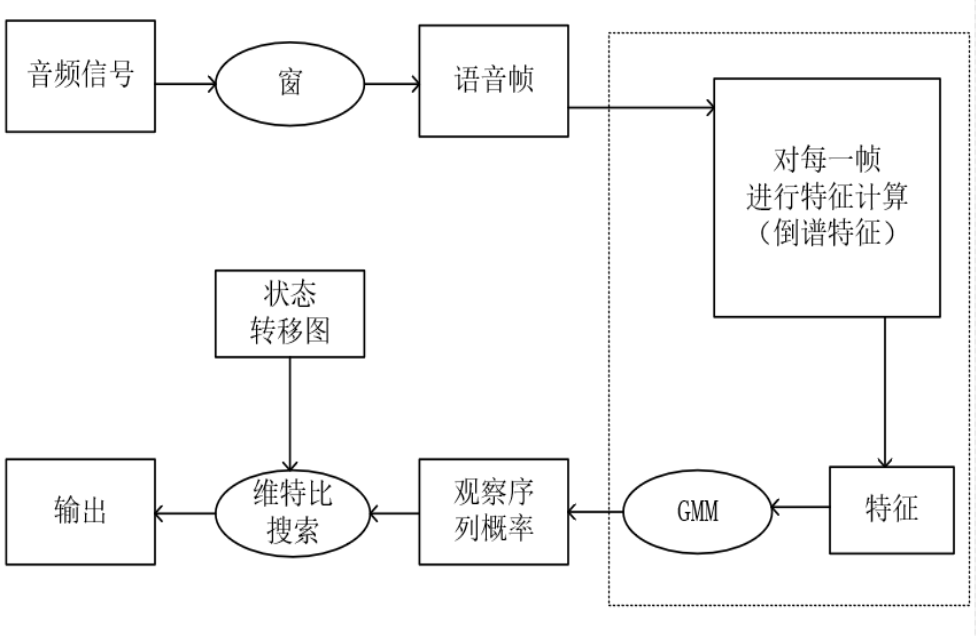

图 2-8 GMM-HMM 语音识别框架图

高斯混合模型（GMM）是指利用多个高斯分布的组合来拟合数据的方式，并采用期望最大（EM）算法进行训练，常被用来解决多分类问题。

假设是现有一个分类任务，分类数是 P，那么 GMM 模型需要 P 个高斯模型，公式 2-9 便是它的概率密度函数

$$p(\mathrm{x}) = \sum_{j=1}^{P} P(j)p(\mathrm{x} \mid j) = \sum_{j=1}^{P} P(j)N_j(\mathrm{x};\mu_j,\sigma_j^2) \tag{2-9}$$

$P(j)$是一个先验概率，指的是选择第 j 个模型的概率；$p(\mathrm{x} \mid j) = N_j(\mathrm{x};\mu_j,\sigma_j^2)$可以看作是一个条件概率，也就是在 j 个模型中输出 x 的概率。可以发现模型的数目为 P，因此它们的概率之和一定为 1，即$\sum_{j=1}^{P} P(j) = 1$。

关于 GMM 的训练，一般是使用期望最大化 (EM) 算法[45]。EM 是一种使用迭代的思想的优化算法，常用来极大化似然估计参数[45]。其计算过程大致如下：

（1）、E-step：根据当前参数估计$p(\mathrm{j} \mid x)$，其计算过程如公式(2-10)所示：

$$P(j \mid x) = \frac{p(x|j)P(j)}{p(x)} = \frac{p(x|j)P(j)}{\sum_{j=1}^{P} P(j)p(x|j)} \tag{2-10}$$

（2）、M-step：对以下参数进行更新 ：GMM 的均值、方差以及单个高斯模型的权重，公式(2-11)——公式(2-13)显示了均值、方差和单个高斯模型的权重的计算。

$$\hat{\mu}_j = \frac{\sum_n P(j|\mathrm{x}^n)\mathrm{x}^n}{\sum_n P(j|\mathrm{x}^n)}, n = 1,2,\cdots,P \tag{2-11}$$

$$\hat{\sigma}_j^2 = \frac{\sum_n P(j|\mathrm{x}^n)\|\mathrm{x}^n-\mu_k\|^2}{\sum_n P(j|\mathrm{x}^n)}, n = 1,2,\cdots,P \tag{2-12}$$

$$\hat{P}(j) = \frac{1}{N}\sum_n P(j \mid \mathbf{x}^n), n = 1,2,\cdots,P \tag{2-13}$$

其中，$\hat{\mu}_j$表示的是单个高斯模型中的第 j 个的数学期望的估计值，$\hat{\sigma}_j^2$表示的是单个高斯模型中的第 j 个的方差的估计值，$\hat{P}(j)$ 表示的是单个高斯模型中的第 j 个所对应的权重值

（3）、反复进行 E-step 和 M-step，当各个参数达到一定范围，就表明训练完成。

隐马尔可夫模型（HMM）是一个统计模型，用来描述一个含有隐含未知参数的马尔可夫过程。它的模型解释为：在已知当前状态的情景下，其未来状态与当前状态有关，而与过去状态无关。其有关计算如公式(2-14)所示。

$$P(i_t \mid i_{t-1}, i_{t-2}, \cdots, i_1, i_1) = P(i_t \mid i_{t-1}), t = 1,2,\cdots,T \tag{2-14}$$

此外，还有两个重要假设：

1、齐次马尔科夫假设——对于序列中任何时刻而言，前一时刻的隐藏状态决定当前的隐藏状态；

2、观察独立性假设——对于观察序列和状态序列而言，在任何时刻，观察序列状态仅取决于它所对应的隐藏状态。

符合两个假设的状态序列与观察序列如图 2-9 所示。

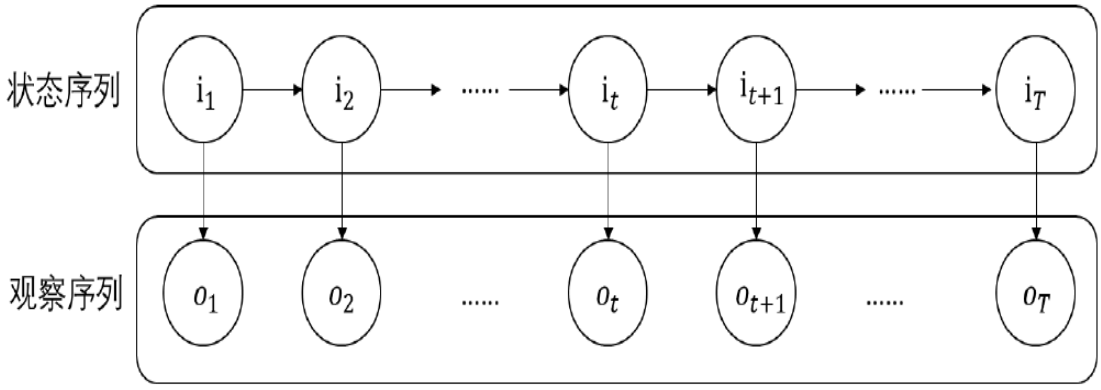

图 2-9 符合两个假设的状态序列与观察序列图

2.3.2 深度学习模型

随着人工智能的发展，卷积神经网络（CNN）也得到发展。最初 CNN 广泛运用在图像领域，近年来在语音领域也得到广泛运用。CNN 是一种具有前馈性质的神经网络，它可以进行卷积的操作，也是比较有代表性的算法[50]。

CNN 它包括了多个部分，一般有：卷积层、池化层和全连接层。CNN 它的输入一般是二维的数据，它的输出则是神经网络的预测结果。一般来说，卷积层是 CNN 里面最重要的部分，卷积层里面会有一个或者多个卷积核，每个卷积核的参数都可以在训练中通过训练得到。卷积层从输入中提取特征，单独的一个卷积层只能提取简单特征，多个卷积层便组合在一起便可以得到复杂的特征。

现在的卷积层一般也包括卷积计算部分和激活函数部分。卷积计算实际就是卷积核进行滑动操作，与输入特征的对应部分相乘相加，最终得到输出的特征。一般有填充（补零）和无填充两种卷积方式，无填充卷积方式计算如图 2-10 所示。

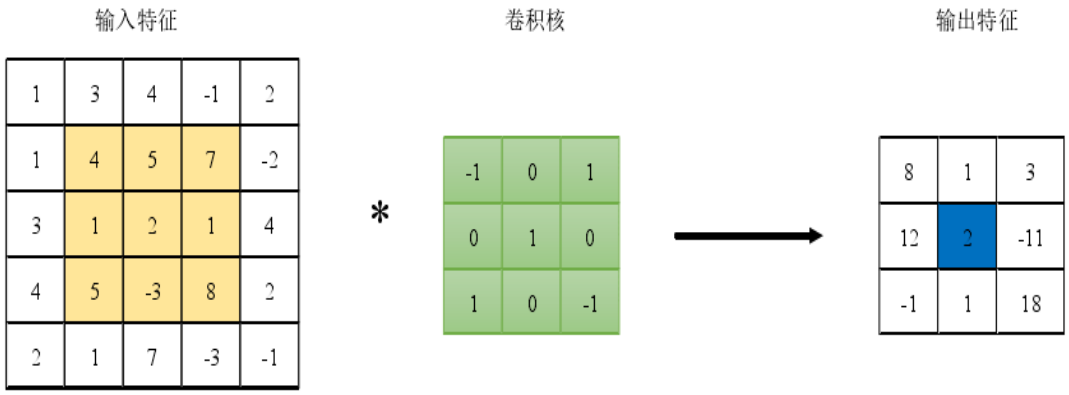

图 2-10 无填充卷积计算图

对于有填充卷积方式，会在输入特征图的四周先进行补 0 填充，然后与卷积核进行卷积操作。有填充卷积方式计算如图 2-11 所示。

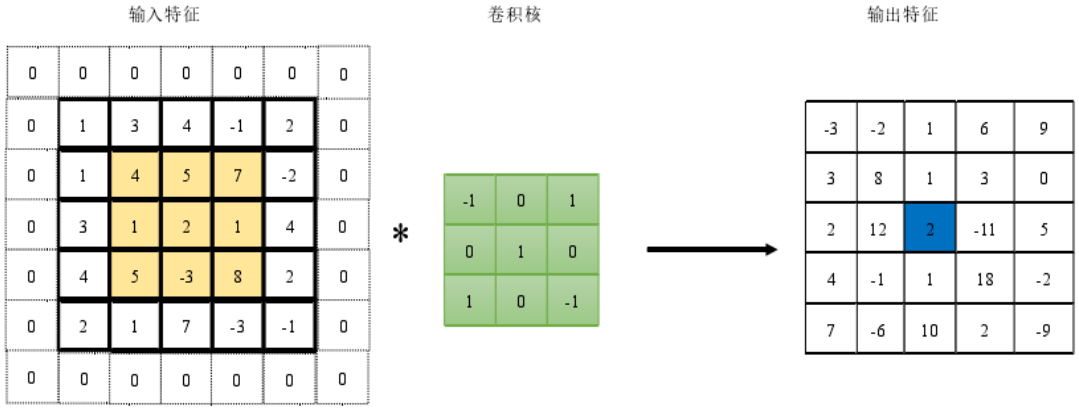

图 2-11 有填充卷积计算图

输出特征的大小如公式(2-15)所示。

$$S_{out} = \frac{S_{in} - S_{kernel} + 2 \times S_{pad}}{S} + 1 \tag{2-15}$$

其中，S_{in}：输入数据大小，S_{kernel}：卷积核的大小，S_{out}：输出数据的大小，S_{pad}表示补 0 的行数或者列数，S表示卷积计算时，卷积核每次移动的大小，一般为 1。

一般来说，在卷积计算后，都会有激活函数用以提取特征。激活函数形式多样，譬如 sigmoid、tanh、relu 等[51]。公式(2-16)——公式(2-18)为三种激活函数的数学计算公式，其函数图如图 2-12 所示。

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2-16}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2-17}$$

$$\mathrm{relu}(x) = max(0, x) \tag{2-18}$$

上述公式中，x 为卷积计算后的输出值。

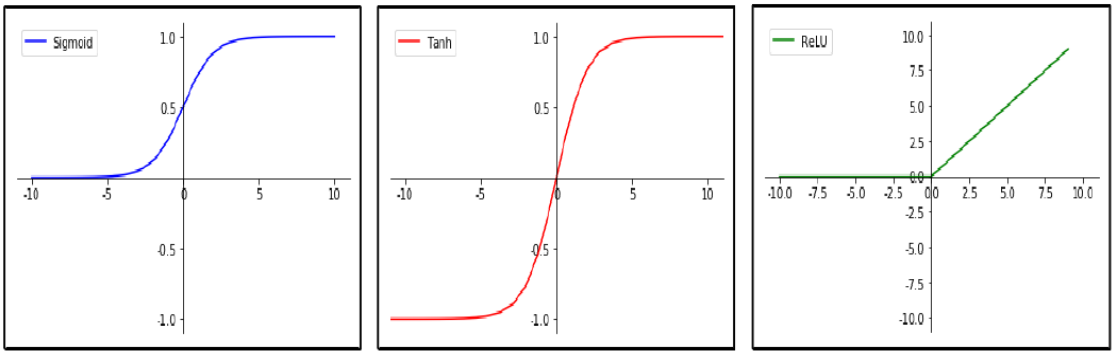

图 2-12 三种常见激活函数图

在 sigmoid 函数中，当 x 比较大或者比较小的时候，函数斜率越来越接近于 0，这样梯队下降算法将会计算得更加缓慢。对于 tanh 函数，相较于 sigmoid，它最终输出的均值接近于 0 而不是 0.5，但与 sigmoid 类似，当当 x 比较大或者比较小的时候，函数斜率越来越接近于 0，这样梯队下降算法将会计算得更加缓慢。对于 relu 函数，在输入大于 0 时，函数斜率为 1，可以始终维持高学习速度，因此目前被广泛运用。

考虑到经过卷积层后，数据的大小变化不大，为更好的降低数据量，因此在 CNN 中还引入了池化层。对于卷积层的输出，池化层不仅可以保留其中的重要部分，还可以大幅度减少不重要部分的信息，这样可以对 CNN 起到简化结构的作用。常见的池化方式有两种：最大池化(max pooling)、平均池化(mean pooling)。最大池化输出的是作用区域的最大值；而平均池化输出的是作用区域的平均值。最大池化示意图如图 2-13 所示。

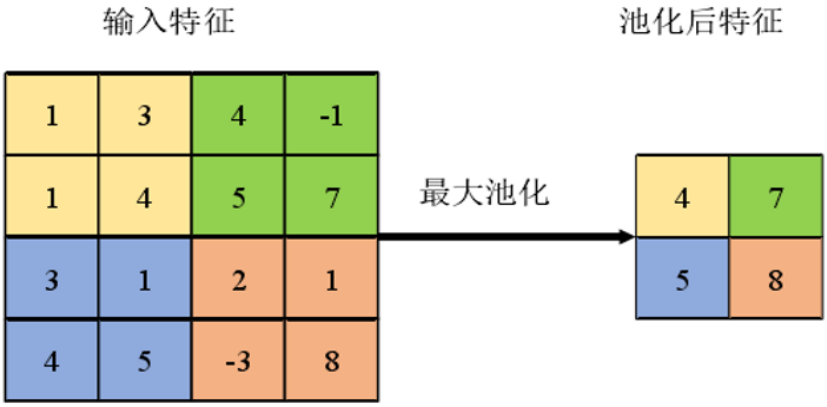

图 2-13 最大池化示意图

对于平均池化，输出的是作用区域的平均值。平均池化示意图如图 2-14 所示。

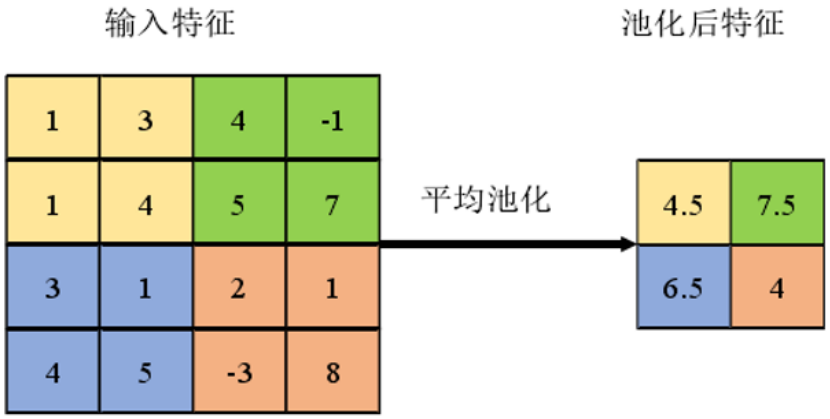

图 2-14 平均池化示意图

全连接层主要的就是乘法操作，它一般位于 CNN 的最后面的部分，也就是一般在卷积层和池化层的后面，这样便可以把它们输出的特征做进一步的处理。处理后的结果便在输出层进行输出。图 2-15 为全连接层示意图。

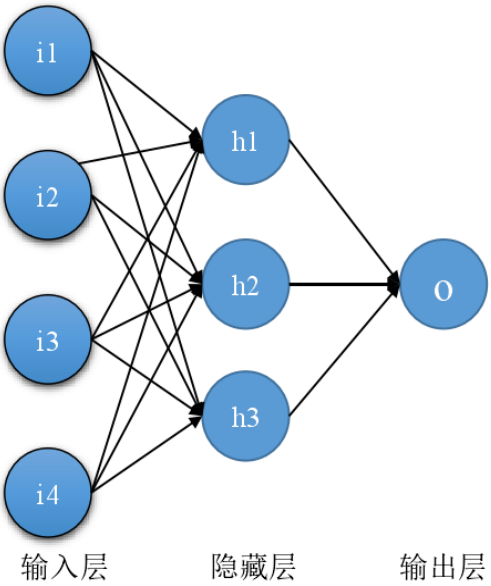

图 2-15 全连接层示意图

图 2-16 为一个简单的 CNN 示意图。

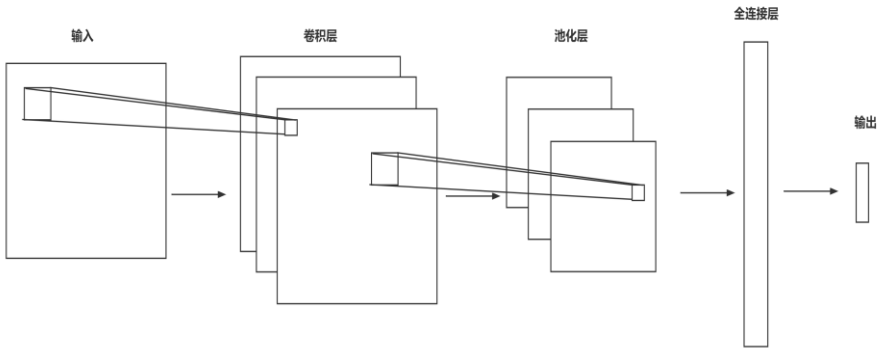

图 2-16 一个简单的 CNN 示意图

在深度学习领域，除了 CNN 以外，RNN 也是比较常见的。循环神经网络（RNN）是一种特殊的神经网络。它有一定的记忆力。其输出不仅与当前时间的输入有关，还受前一时间的输出影响。

由于 RNN 的输出与前一时间有关，更适用于与时间有关的处理任务，因此与 CNN 相比较，RNN 对与时间有关信号处理显得更加高效。所以，在自然语言处理、语音识别、声音事件检测等领域中，RNN 得到越来越多的应用，并且也取得了不

错的成果。图 2-17 为一个简单的 RNN 示意图。其中左侧图为其简化图，右侧为 RNN 的展开图。

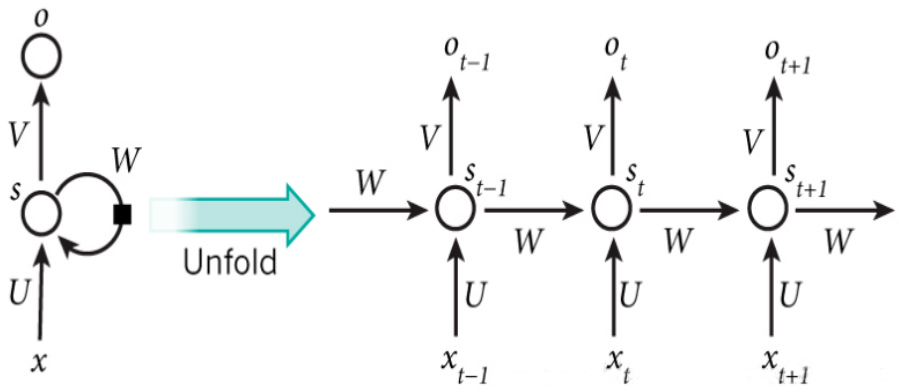

图 2-17 RNN 示意图

在图中，有 U、W、V 三个表示参数的矩阵。x_t:RNN 在 t 时刻的输入；s_t： RNN 在 t 时刻的隐藏状态；o_t: RNN 在 t 时刻的输出。RNN 有关的数学计算如公式(2-19)和(2-20)示。

$$s_t = f(Ws_{t-1} + Ux_t + a) \tag{2-19}$$

$$o_t = g(V_t + b) \tag{2-20}$$

其中，$f(x)$和$g(x)$表示隐藏层和输出层的有关激活函数，a、b 表示的是函数偏置。

RNN 虽然能够很好地处理时序问题，但在序列的最后，会出现问题。为了克服这种情况，产生了长短时记忆(LSTM)[52]网络和门控循环单元(GRU)[53]。

LSTM 中有各种不同的门以调节信息流。输入门控制当前时刻的数据输入；遗忘门决定应该丢弃或保留哪些信息；输出门决定下一个隐藏状态。GRU 思想和 LSTM 类似，GRU 只有重置门以及更新门这两个门[51]。重置门决定着前一时刻的隐藏状态是否成为当前候选的隐藏状态[51]。

2.4 本章小结

本章节主要介绍声音事件检测的基本原理。首先，介绍了目前基于神经网络的声音事件检测算法，其中重点对声音事件检测算法的训练和测试过程进行介绍。然后介绍了目前常用的几种声音特征提取方式；最后，对声音事件检测算法的传统模型中的 GMM-HMM 进行介绍，并介绍深度学习模型如 CNN、RNN 等原理。

第三章 声音事件检测算法设计

3.1 轻量级声音事件检测算法设计

一般而言，基于神经网络的声音事件检测算法包括声音信号输入、特征提取、神经网络模型和系统输出等几个部分。论文提出的基于 CNN 的声音事件检测算法也有系统输入、特征提取、神经网络模型、系统输出这几个部分。

论文提出了一个轻量级的声音事件检测算法 LSED。在该算法中，首先使用了选择性可分离卷积方案，以降低计算网络参数量与 FLOPs，同时实现高识别准确率。其次，使用协调注意力方案以进一步提高识别准确率。

图 3-1 为所提出的 LSED 算法的整体架构。它主要包括音频输入、声音特征提取、神经网络和输出 4 个部分。

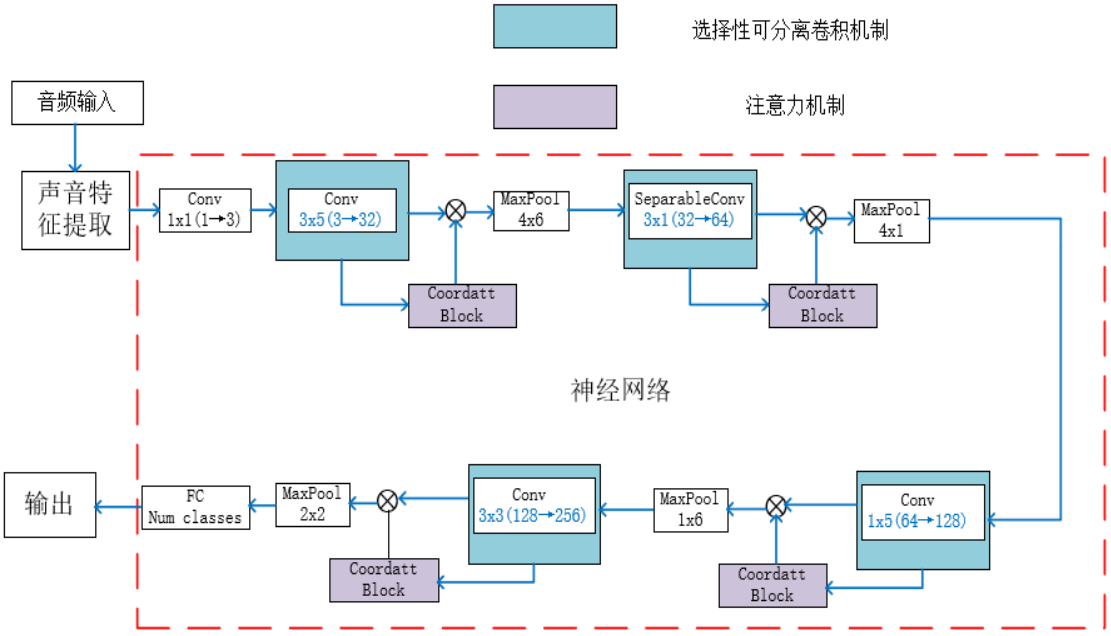

图 3-1 LSED 算法整体架构

其中，音频输入部分调用相关函数，完成对输入音频的读取工作。输出部分对神经网络的输出向量进行处理，从而得到音频信号所对应的声音事件类别。

在 LSED 算法中，声音特征提取部分，负责对输入声音信号进行声音特征提取，将声音信号由线性时域变换到梅尔频域，和直接输入相比，降低了数据的输入量并提高了识别准确率。在该模块中，输入的是声音信号，最后得到的声音信号所对应的梅尔频谱，该梅尔频谱将会成为后面神经网络部分的输入。

然后，将提取得到的梅尔频谱特征输入到神经网络模块提取更深层次的特征并进行声音事件分类。在该神经网络模块中，使用到了选择性可分离卷积机制与协调注意力机制。其中，选择性可分离卷积机制将普通卷积与深度可分离卷积相结合，主要用来降低 LSED 算法的参数量与 FLOPs，从而构建轻量级的 SED 算法；而协调注意力机制在基本不增加 LSED 算法复杂度的基础上，重点关注与声音事件分类相关的特征，从而提高 LSED 算法的识别准确率。这两个模块将在 3.3 和 3.4 小节中进行阐述。

最后，选择将预测概率最大所对应的声音事件类别进行输出，作为 LSED 算法的最终输出。

论文提出的 LSED 算法有关参数配置如表 3-1 所示，包括卷积核大小、输入大小与输出大小。

表 3-1 LSED 算法有关参数配置

网络层类别	卷积核	输入大小	输出大小
卷积层 0	3×1×1	(1,128,256)	(3,128,256)
块 1_卷积层	32×3×5	(3,128,256)	(32,128,256)
注意力块 1	-	(32,128,256)	(32,128,256)
块 1_池化层	4×6	(32,128,256)	(32,32,42)
块 2_卷积层	64×3×1	(32,32,42)	(64,32,42)
注意力块 2	-	(64,32,42)	(64,32,42)
块 2_池化层	4×1	(64,32,42)	(64,8,42)
块 3_卷积层	128×1×5	(64,8,42)	(128,8,42)
注意力块 3	-	(128,8,42)	(128,8,42)
块 3_池化层	1×6	(128,8,42)	(128,8,7)
块 4_卷积层	256×3×3	(128,8,7)	(256,8,7)
注意力块 4	-	(256,8,7)	(256,8,7)
块 4_池化层	2×2	(256,8,7)	(256,4,3)
全连接层	3072×50	(3072,1)	50

在表 3-1 中，“输入大小”和“输出大小”有三维张量类型与二维张量类型。其中，三维张量类型中，对应的维度依次为通道维度、时间维度和频率维度；二维张量类型中，对应的维度依次为时间维度和通道维度。其中，卷积层 0 为声音特征提取部分的 1×1 内核的卷积，用于通道的转换。全连接层最终的输出大小为语音数据集所对应的事件类别数，ESC-10 与 UrbanSound8K 为 10 个类别数，ESC-50

为 50 个类别数，表 3-1 选取 ESC-50 的类别数作为全连接层的输出大小。同时，在神经网络部分，将 1 个卷积层及其相邻的 1 个注意力块及池化层视为 1 个模块，由此将神经网络部分为了 4 块及最终全连接层。

由于语音数据集的数据量相对较少，且 logmel 频谱图有与图像一样的二维结构，因此可以使用图像数据集用迁移学习的方式获得更高的准确率。如图 3-2 为预训练与微调阶段框架图。

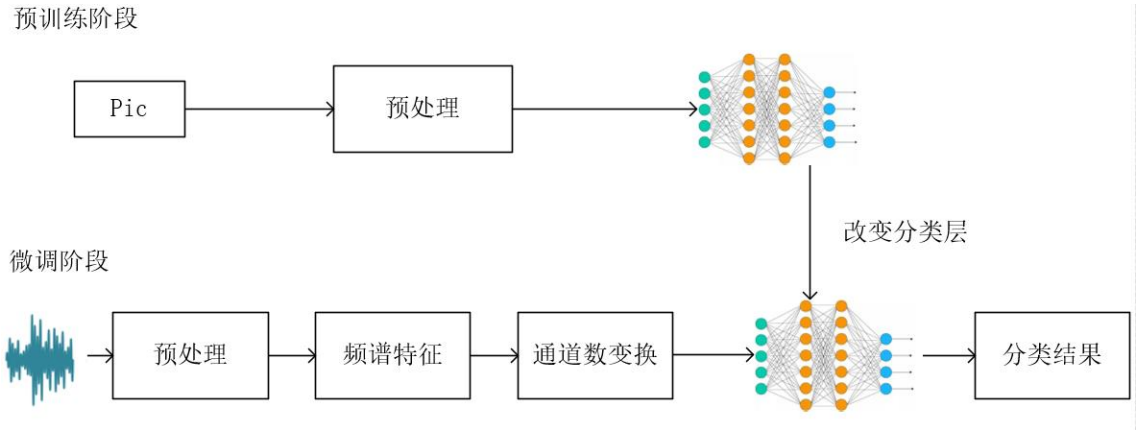

图 3-2 预训练阶段与微调阶段框架图

算法预训练阶段使用的数据集为图像数据集 ILSVRC，该数据集总共 1000 个不同的图像类别，用样本数大的图像数据集进行迁移学习来训练神经网络便可以获得比较好的神经网络的权重。然而，由于 ILSVRC 数据集包含 3 个输入通道（即 RGB 通道），而 logmel 频谱图仅有 1 个通道，在将其送入神经网络之前，因此使用可学习的 1×1 内核通过逐点卷积将单通道的 logmel 频谱图转换为类似于包含 RGB 三个通道的图像。在预训练阶段结束后，使用预训练得到的最优模型的参数，通过对频谱特征进行通道数变换和改变最后的分类层，应用声音事件检测数据集进行微调。

3.2 声音特征提取设计

在 LSED 算法中，声音特征模块，负责对输入声音信号进行声音特征提取，将声音信号由线性时域变换到梅尔频域。

与大多数现有的 SED 方法一样，声音特征提取模块进行初步的特征提取以提高检测精度。首先，修剪音频，剪辑音频的静音部分，并将音频调整为原始长度。然后，将剪辑处理后的音频进行处理以获得 logmel 频谱图。在此过程中，输入音频信号首先使用短时间傅立叶变换 (STFT) 进行处理，汉明窗口大小为 23.2 ms（44.1kHz 下的 1024 个样本）和 50% 重叠。然后处理后的音频信号通过一个带

有 128 个三角带通滤波器的梅尔滤波器组和一个对数计算模块，得到 128×429（用于 ESC-10 和 ESC-50）和 128×343（US8K）的特征图。

由于神经网络的输入为 128*256，因此还需要对得到的特征图进行拆分处理，以得到符合神经网络输入的特征。将得到 128×429（ESC-10 和 ESC-50）和 128×343（UrbanSound8K）的特征图拆分成多个新的频谱图，其中每一个新的频谱图包含 256 帧，50% 比例的重叠。对于最后一个频谱图，可以应用与倒数第二个频谱图的不同重叠比例以确保它的宽度为 256。最终，128×429（ESC-10 和 ESC-50）的特征图将被拆分为 3 个 128×256 的特征图作为神经网络的输入，128×343（US8K）的特征图被拆分为 2 个 128×256 的特征图作为神经网络的输入。声音特征提取流程如图 3-3 所示。

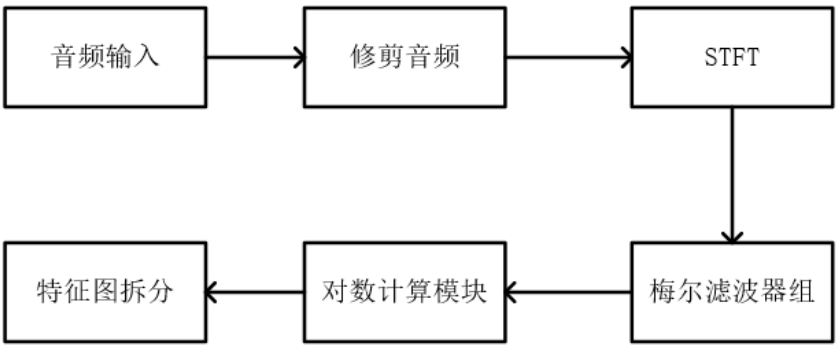

图 3-3 声音特征提取流程图

3.3 选择性可分离卷积机制设计

对于基于 CNN 的 SED 模型，计算复杂度主要来自卷积部分。一般来说，可以使用深度分离卷积去替代标准卷积从而降低计算的复杂度。然而，它和标准卷积相比，对特征学习的能力较差，会导致识别准确度有所下降。因此，对于声音事件检测任务，需要在计算复杂度和识别准确率之间取得平衡，选择性地采用标准卷积和深度可分离卷积。

下面章节首先会介绍标准卷积，然后介绍其复杂度的具体计算过程；接着会对深度可分离卷积进行介绍，介绍其复杂度的计算并与标准卷积的复杂度进行对比。最后介绍轻量级的声音事件检测算法 LSED 的选择性可分离卷积机制的设计思想与具体内容。

3.3.1 标准卷积

正常的卷积运算便是标准卷积，也称为普通卷积。对于标准卷积而言，卷积操作所需要的参数量和 FLOPs 较大。图 3-4 为标准卷积操作示意图。在该例子中，输入数据的大小是 3×9×9，卷积层的部分有 128 个 3×3×3 的卷积核，得到是 7×7×128 的输出。

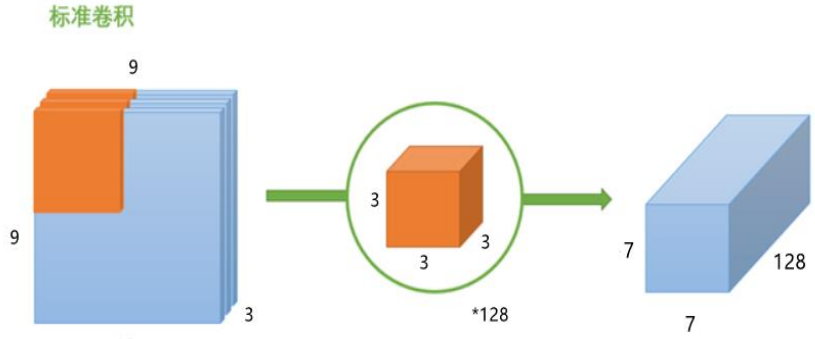

图 3-4 标准卷积操作示意图

在这过程中，所需要的参数量（即卷积核的参数量）为$3 \times 3 \times 3 \times 128$，即 3456 个参数。在这过程中，所需要的 FLOPs 为：$3 \times 3 \times 3 \times 128 \times 7 \times 7$，即 169344。

一般情况下，假定输入的长度、宽度和通道数分别是：Win、Hin、Cin，卷积核大小为$k * k$，输出的长度、宽度和通道数Wout、Hout、Cout。则标准卷积的参数量计算如公式(3-1)所示，FLOPs 计算如公式(3-2)所示。其中 params 表示标准卷积操作中所涉及到的参数量。

$$params = k * k * Cin * Cout \tag{3-1}$$

$$FLOPs = k * k * Cin * Cout * Wout * Hout \tag{3-2}$$

3.3.2 深度可分离卷积

深度可分离卷积可以看作是将标准卷积分成了两步进行操作，分别为深度卷积和逐点卷积[54]。逐点卷积是一种卷积核大小为 1×1 的标准卷积，它可以将输入的维度降低或者升高，从而得到输出。与标准卷积相比，深度可分离卷积可以大幅度地降低标准卷积操作带来的参数量和 FLOPs。图 3-5 为一个深度可分离卷积的例子。在该例子中，输入数据的大小是 3×9×9，卷积的部分有 3 个 3×3 的卷积核和 128 个 1×1 的卷积核，得到是 7×7×128 大小的输出。

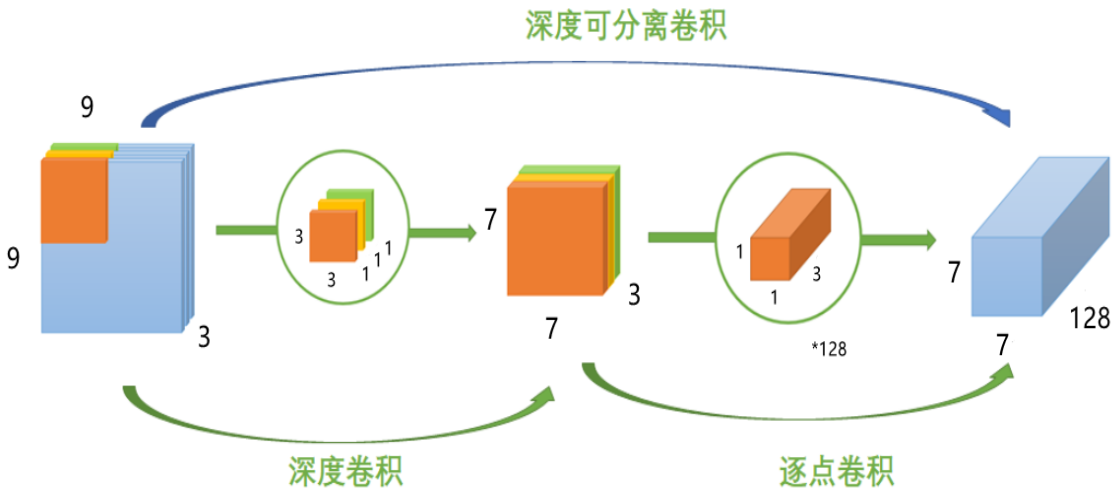

图 3-5 深度可分离卷积示意图

在深度卷积部分，参数量为$3*3*3$，即 27 个参数；在逐点卷积部分，参数量为$1*1*3*128$，即 384 个参数。那么可以得到整个过程需要使用 411 个参数。对于 FLOPs，在深度卷积部分，FLOPs 为$3*3*3*7*7$，即 1323；在逐点卷积部分，FLOPs 为$1*1*3*7*7*128$，即 18816。则整个深度可分离卷积过程的 FLOPs 为 20139。对比上小节标准卷积的参数量与 FLOPs，可以发现，深度可分离卷积参数量不到标准卷积的 12%，FLOPs 不到标准卷积的 20%，从而表明了使用深度可分离卷积能够较大层度降低卷积操作过程中的参数量与 FLOPs。

一般情况下，假定输入的长度、宽度和通道数分别是：Win、Hin、Cin，卷积核大小为$k*k$，输出的长度、宽度和通道数Wout、Hout、Cout。使用深度可分离卷积需要的参数量和 FLOPs 分别如公式(3-3)和公式(3-4)所示。其中 params 表示深度可分离卷积中的所涉及到的参数量。

$$params = k*k*Cin + 1*1*Cin*Cout \tag{3-3}$$

$$FLOPs = k*k*Cin*Wout*Hout + 1*1*Cin*Wout*Hout*Cout \tag{3-4}$$

3.3.3 选择性可分离卷积

论文提出选择性可分离卷积方案：在一部分阶段使用标准卷积，在一部分卷积使用深度可分离卷积。在第一、第三阶段使用标准卷积，在第二、第四阶段使用深度可分离卷积。其选择性可分离卷积结构如图 3-6 所示。

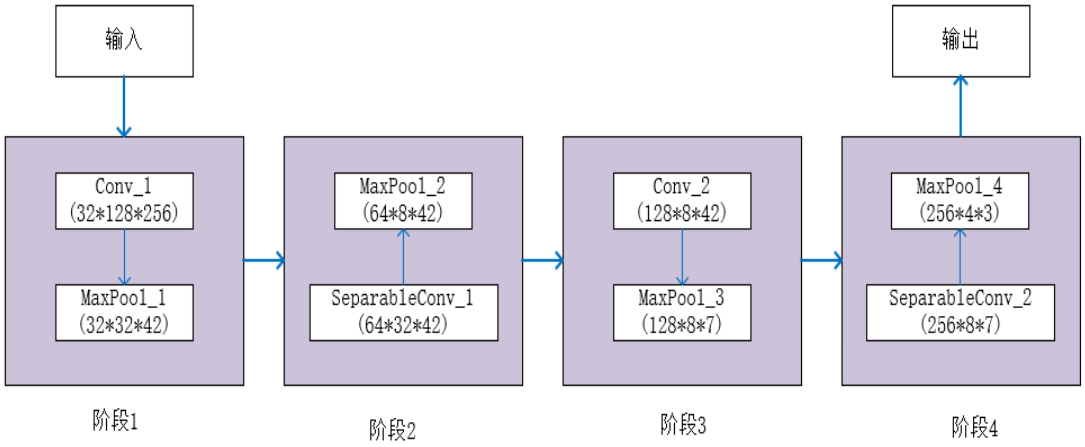

图 3-6 选择性可分离卷积结构图

标准卷积与深度可分离卷积有各自的优点与缺点。深度可分离卷积能够大幅度降低卷积操作的参数量与 FLOPs，但学习能力比标准卷积弱，导致精度下降。通常而言，对于 CNN 网络，较低阶段的卷积层更加靠近输入，能够提取更多特征，因此阶段一采用标准卷积。而在较高阶段的卷积层中，由于通道数较大，所以为了能够降低网络复杂度，则尽可能多地使用深度可分离卷积。当连续采用过多的深度可分离卷积时，学习能力会出现下降，因此在第三阶段采用标准卷积，而在第二、第四阶段采用深度可分离卷积。

通过这样的设计，第一阶段与第三阶段采用标准卷积，第二和第四阶段采用深度可分离卷积。这样的设计保证了模型的复杂度与识别准确率。

3.4 协调注意力机制设计

上一小节提出的选择性可分离卷积机制能够有效的降低模型的复杂度，并能够拥有一定的检测精度。为了尽可能提高声音事件识别准确度，在轻量级的声音事件检测算法 LSED 中，还采用了协调注意力机制。通过让网络层专注于重要的权重，注意力机制已被广泛用于提高准确性。文献[55]提出的注意力方法，在通道域和空间域上使用了注意力机制，提高了其在计算机视觉上的准确率。文献[56]提出协调注意力方法，通过将通道注意分解为两个一维特征编码过程，分别沿两个空间方向聚合特征。这样，一个方向可以捕获长距离依赖关系，另一个方向可以很好保留位置信息。该方法能够有效提高在计算机视觉的准确率。

在提出的轻量级模型中，考虑到神经网络的输入是对频域和时域信息进行编码的梅尔谱图特征，论文采用协调注意方案来提取两个空间方向的位置注意信息，

声学特征中表示时间和频率的方向是感兴趣的主要信息。图 3-7 为论文使用的协调注意力机制示意图。

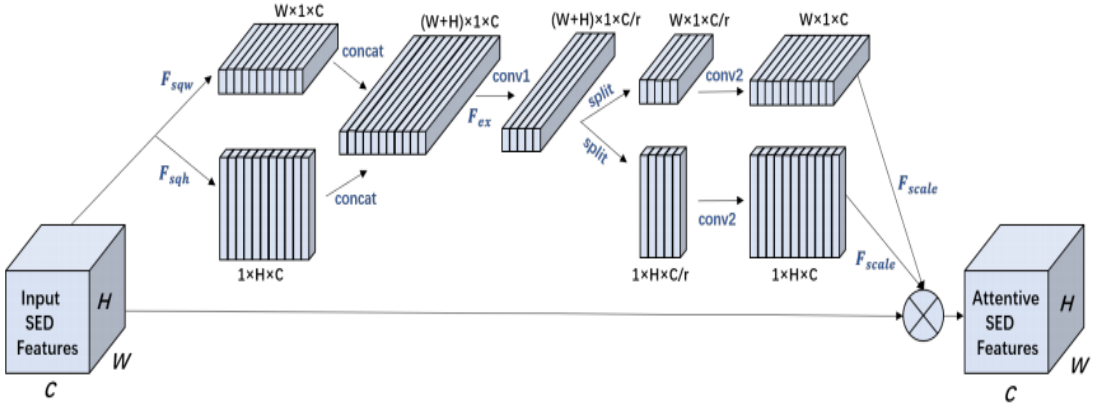

图 3-7 协调注意力机制示意图

输入 SED 特征首先通过具有两个空间范围的池化内核(H,1)或(1,W)进行挤压操作（F_{sqh}和F_{sqw}），分别沿垂直坐标（频域）和水平坐标（频域）以生成的聚合特征图。通过将上述两个变换的特征图连接起来(concat)，然后通过卷积核大小为 1×1 的卷积变换(conv1)和激励函数(F_{ex})，产生中间特征图$\mathrm{f} \in \mathrm{U}^{(W+H)\times 1\times C/r}$。然后将中间特征图f沿垂直和水平方向分成两个单独的张量$\mathrm{f}^{\mathrm{h}} \in \mathrm{U}^{1\times H\times C/r}$和$\mathrm{f}^{\mathrm{w}} \in \mathrm{U}^{W\times 1\times C/r}$。 之后，期望张量$\mathrm{f}^{\mathrm{h}}$和张量$\mathrm{f}^{\mathrm{w}}$的通道数能和输入特征的通道数相同，所以使用了两个 1×1 卷积(conv2)进行张量的变换。最后，通过将输入的 SED 特征和两个经过重新缩放(F_{scale})的张量与 sigmoid 激活函数相乘，获得输出注意力的 SED 特征。

上述内容便是对论文提出的协调注意力机制的简单介绍。下面章节将对嵌入协调信息与产生协调注意力部分进行具体介绍。

3.4.1 嵌入协调信息

经过特征提取后得到的梅尔频谱是一个二维的特征，它相应的位置信息对于与空间结构有关的声音检测任务至关重要。为了使注意力块获得精确的位置信息并捕获长距离依赖，因此将全局池化分解为成对的 1D 特征编码操作。因此，高度为 h 的第 c 个通道的输出可以表示为：

$$z_c^h(h) = \frac{1}{W}\sum_{0\le i<W} x_c(h,i) \tag{3-5}$$

类似地，宽度为 w 的第 c 个通道的输出可以写为

$$z_c^w(w) = \frac{1}{H}\sum_{0\le j<H} x_c(j,w) \tag{3-6}$$

上面的两个变换就是在两个方向上进行特征编码，产生一对方向感知的特征图。这样，一个方向可以捕获长距离依赖关系，另一个方向可以很好保留位置信息。这可以帮助神经网络获取准确的特征。

因此，产生了两个输出特征图，一个为$\mathbf{z}^h \in \mathrm{U}^{1\times H\times C}$，它的高度是 H，通道数是 C；另一个为$\mathbf{z}^w \in \mathrm{U}^{W\times 1\times C}$，它的宽度是 W，通道数是 C。

3.4.2 产生协调注意力

如上所述，等式(3-5)和等式(3-6) 完成了两个方向上的特征编码。在得到特征后，需要有第二个转换，称为产生协调注意力。设计有以下三个标准：首先，对于网络模型复杂度增加不能过大。其次，它可以保留位置关系，获取对识别任务重要的区域特征。最后，它还应该能够从各个不同的通道之间获取与识别任务有关的特征。

具体来说，给定由等式(3-5)和等式(3-6)生成的聚合特征图，首先将它们连接起来，然后发送到一个共享的 1×1 卷积变换函数 F1，产生中间特征图f。其有关数学计算如公式(3-7)所示

$$\mathbf{f} = \delta(F_1([\mathbf{z}^h, \mathbf{z}^w])) \tag{3-7}$$

上述公式中，[·,·]表示沿空间维度的链接操作，δ为非线性激活函数，$\mathrm{f} \in \mathrm{U}^{(W+H)\times 1\times C/r}$为中间特征图，它可以反应水平方向和垂直方向的空间信息。此外，还使用了r 来控制块大小的缩小率。然后将中间特征图f沿垂直和水平方向分成两个单独的张量$\mathrm{f^h} \in \mathrm{U}^{1\times H\times C/r}$和$\mathrm{f^w} \in \mathrm{U}^{W\times 1\times C/r}$。之后，期望张量$\mathrm{f^h}$和张量$\mathrm{f^w}$的通道数能和输入特征的通道数相同，所以使用了两个 1×1 卷积变换F_h和F_w。其有关数学计算如公式(3-8)和公式(3-9)所示。

$$\mathbf{g}^h = \sigma(F_h(\mathbf{f}^h)) \tag{3-8}$$

$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w)) \tag{3-9}$$

其中，σ为 sigmoid 激活函数。由于$\mathbf{f}$的通道数一般较大，所以期望能够减少它的通道数，这里使用到了缩减率r。然后，输出得到$\mathbf{g}^h$和$\mathbf{g}^w$可以作为最后注意力权重的一部分，和输入的数据进行结合便可以得到输出 Y，有关数学计算如公式(3-10)所示。

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{3-10}$$

如上所述，通过采用协调注意力机制来提取声学特征的两个维度上的注意力信息。这有助于神经网络关注时域和频域中感兴趣的区域的信息，并提高检测精度。

3.5 本章小结

本章详细介绍了基于 CNN 的轻量级声音事件检测算法。首先，介绍了轻量级的声音事件检测算法 LSED 的整体框架；然后，详细阐述了论文提出的两种机制，即选择性可分离卷积机制与协调注意力机制。选择性可分离卷积是指在卷积神经网络的不同阶段采用不同类型的卷积操作，协调注意力机制是指在时域、频域和通道域中对于声音事件检测有关的特征和区域进行重点关注。主要介绍了两种机制提出的原因和有关的算法原理。

第四章 基于 FPGA-DPU 的声音事件检测系统实现

本章将重点介绍基于 DPU(FPGA 的深度学习处理单元)的声音事件检测系统的设计。即将提出的轻量级算法通过 FPGA 的深度学习处理单元(DPU)进行了实现，从而构建了一个基于 FPGA-DPU 的声音事件检测系统。该系统基于 ZCU104 平台来开发设计的，通过使用 Vivado2020、DNNDK 与 PetaLinux 开发平台完成 DPU 的部署。

本章将依次从 FPGA-DPU、系统设计实现、实现效果部分分别进行介绍。其中，将详细介绍系统设计实现部分。

4.1 FPGA-DPU 介绍

本文采用的开发板为 ZCU104 开发板，其实物图如图 4-1 所示。

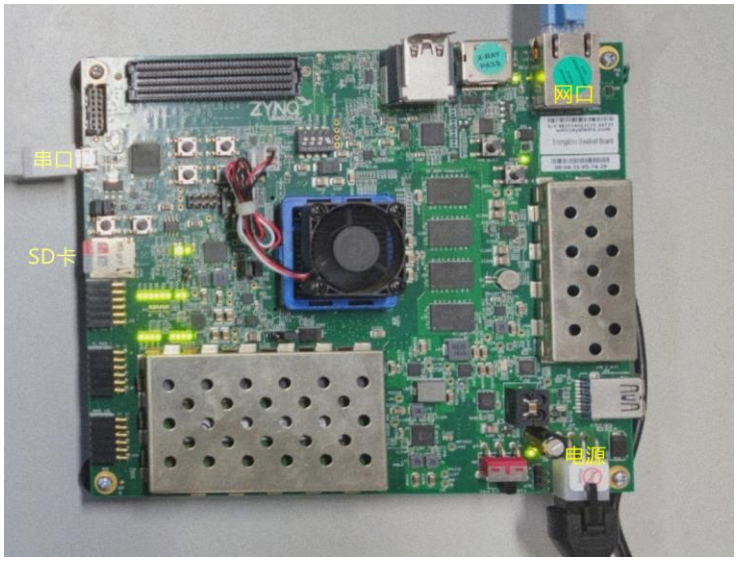

图 4-1 ZCU104 实物图

该开发板属于 Zynq® UltraScale+™ MPSoC 系列。在该开发板的 PS 端配备四核 ARM® Cortex™-A53 应用处理器、双核 Cortex-R5 实时处理器，能够较快地处理相应任务。同时，开发板配备网口，串口与 sd 卡槽，能够较为方便地与 PC 进行通信。

Xilinx®深度学习处理单元 (DPU) 是赛灵思公司针对卷积神经网络进行优化的可编程引擎。它由高性能调度器模块、混合计算阵列模块、取指令单元模块和全局内存池模块组成。通过在 FPGA 开发板中调用 DPU 能够对卷积神经网络进行实现与加速。

DPU 使用有其特定的指令集，可以有效应用于深度学习的网络部署中。目前，已经在 DPU 上部署成功一些神经网络，例如 VGG、ResNet、GoogLeNet、YOLO、SSD、MobileNet 和 FPN 等。

一般选择 Zynq®-7000 SoC 或 Zynq® UltraScale+™ MPSoC 器件作为硬件平台，使用硬件平台的可编程逻辑 (PL) 集成 DPU IP，并直接连接到处理系统 (PS)。DPU 可以通过特定的指令来实现图像的输入和数据的输出，以达到访问数据的目的。 还需要模块完成服务中断和协调数据传输的工作，这模块一般是在应用程序处理单元(APU)上运行的。

DPU 的顶层框图如图 4-2 所示。

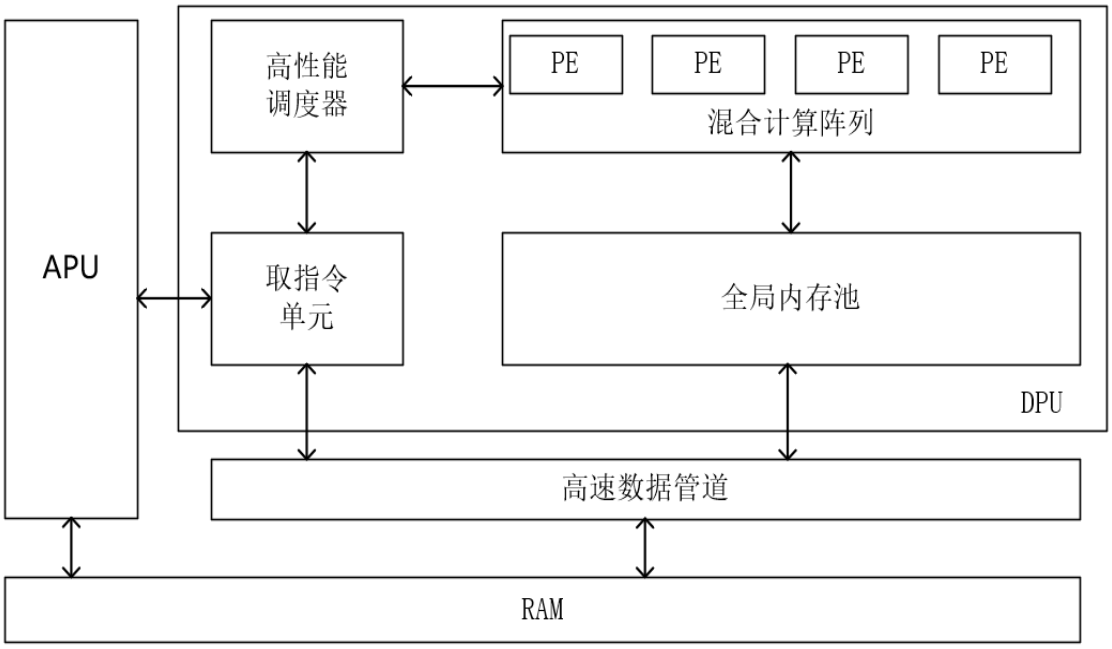

图 4-2 DPU 顶层框图

在上图中，APU 表示应用处理单元，PE 表示处理引擎，DPU 表示深度学习处理单元，RAM 表示随机存取存储器。

4.2 系统设计实现

本节将重点介绍基于 DPU 的声音事件检测系统的实现。首先介绍系统结构，在该部分将就系统的 PL 端与 PS 端的作用分别进行介绍。然后介绍 DPU 的部署，在该部分首先对 DPU 部署的有关硬件工程进行介绍，其次对 DPU 的相关参数配置进行介绍。接着介绍应用程序，该部分对 PS 端应用程序的流程进行介绍。最后介绍实现流程，该部分将对整个实现的流程步骤进行介绍。

4.2.1 系统结构

基于 FPGA-DPU 的声音事件检测系统的结构如图 4-3 所示，采用软硬件相结合的方法在 ZCU104 平台上进行设计。ZCU104 内部有 FPGA 可编程逻辑(PL)部分和处理系统(PS)部分，故可以采用软硬件分离的设计思路完成系统设计。其中，PL 部分通过使用 DPU IP 核完成神经网络的部署，PS 部分通过编写应用程序来完成对音频信号的预处理以及对神经网络识别结果的处理。

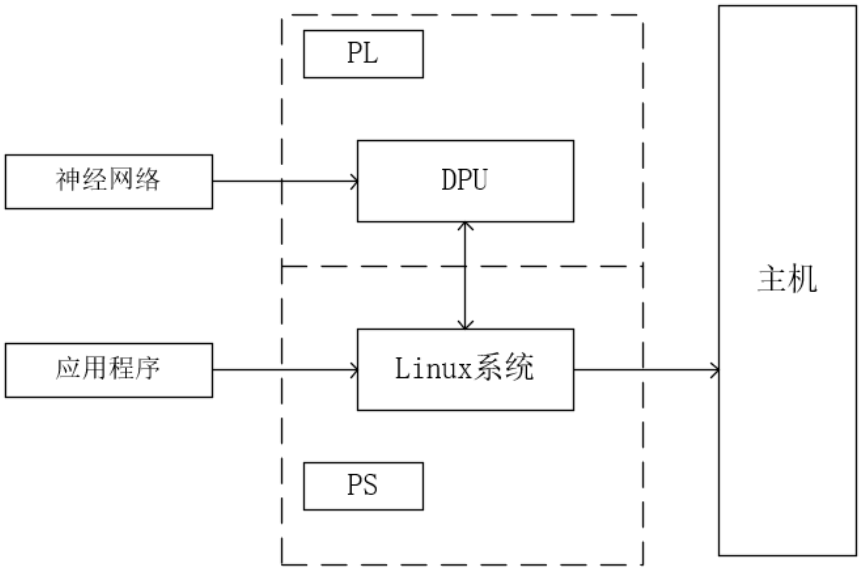

图 4-3 系统结构图

在系统中，通过使用 Vivado2020 创建工程，在工程中导入 DPU 的 IP 核，通过对有关参数进行配置，可以实现将 DPU 部署到 ZCU104 的 PL 部分。对上一章节设计好的算法进行神经网络的训练，然后使用 DNNDK 工具将训练好的神经网络模型经过量化、编译等操作后转成 DPU 能够识别的神经网络模型。PS 部分通过使用 PetaLinux 平台创建相应工程并进行配置，从而得到可以运行 DPU 的嵌入式 Linux 系统。应用程序可以调用相应的函数完成对 DPU 初始和创建等任务，同时还可以完成对音频信号的预处理和对神经网络结果的后处理，进而完成对声音事件的识别。

4.2.2 DPU 部署

Xilinx®深度学习处理单元 (DPU) 是针对卷积神经网络进行优化的可编程引擎。在 DPU 内部有大量的神经网络进行计算时所需要的加法器和乘法器等，进而可以使用 DPU 完成深度学习加速。

一般在 Vivado 工程中导入 DPU 的 IP 核，然后创建相关的硬件工程。DPU 部署的硬件工程如图 4-4 所示。

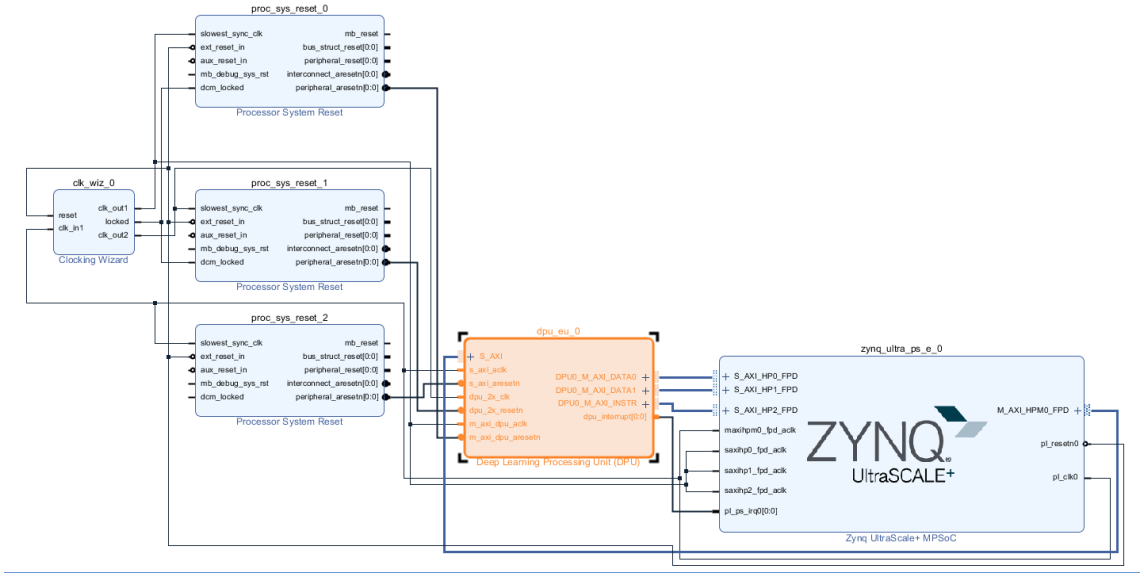

图 4-4 DPU 部署硬件工程图

在该硬件工程中，有 1 根中断线与 4 组 AXI 总线。中断线主要是 DPU 用来对处理器发出中断信号，告知处理器任务完成情况的信号线。AXI 总线中，有两组是用于数据传输的，分别为 DPU0_M_AXI_DATA0 与 DPU0_M_AXI_DATA1 的 AXI 总线。其余两组 AXI 总线分别用于传输指令和 DPU 访问处理器中的相关状态寄存器。

DPU 中有三个时钟域，分别为寄存器配置、数据控制器和计算单元部分的时钟域。而三个时钟均与 PL 端相连接。DPU 的时钟域如图 4-5 所示。

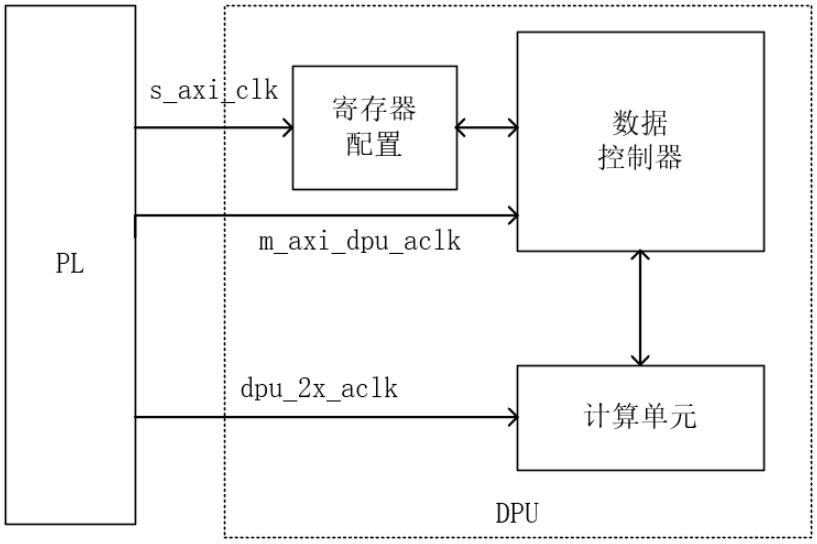

图 4-5 DPU 时钟域图

在 DPU 中，s_axi_clk 用于寄存器配置，通常将该时钟频率为主时钟频率，即 Clocking Wizard 中的 clk_in1 所对应的频率。m_axi_dpu_aclk 与控制寄存器相连接，时钟频率一般与主时钟频率相同。最后，计算单元所对应的时钟 dpu_2x_aclk 应该设置为主时钟频率的两倍。

然后根据所需要部署的神经网络的特点而进行相应的 DPU 的配置。本文 DPU 配置信息如表 4-1 所示。

表 4-1 DPU 配置表

参数	含义	配置
DPU Cores	DPU 核心数量	1
Arch of DPU	DPU 的结构	B1152
RAM Usage	RAM 利用率	低
Channel Augmentation	通道扩展	使能
Depth Wise Conv	深度卷积	使能
Average Pool	平均池化	使能
ReLU Type	ReLU 类型	ReLU + LeakyReLU + ReLU6

在一个 DPU IP 中最多可以选择四个内核。DPU 采用了流水线结构，多个内核可以带来更好的系统性能，本设计有关的卷积神经网络算法复杂度较低，使用 1 个内核即可满足其设计要求，所以选择使用 1 个 DPU 核心以节省系统功耗与系统资源。DPU 的结构有多种，如 B512、B800、B1024、B1152、B1600、B2304、B3136、B4096 等。DPU 结构的数字越大表明其卷积的并行度越高，相应的处理速度也越快，本设计选用 B1152 结构。

在 DPU 运行时片上 RAM 会缓存网络的权重、参数等信息，高 RAM 使用率意味着片上存储块将更大，从而使 DPU 在处理中间数据时具有更大的灵活性。本设计有关的算法复杂度较低，故本设计在 RAM 利用率上选择低利用率。

本设计的神经网络为多通道，故 DPU 配置上选择通道扩展。设计的神经网络算法采用了深度可分离卷积，故在 DPU 配置上选择了使能深度卷积。本设计的神经网络采用了池化操作，故在 DPU 配置上选择了使能平均池化。ReLU 类型可以设置卷积神经网络的激活函数，本设计在 DPU 配置上选择了 ReLU 和 ReLu6 作为激活函数。

4.2.3 应用程序

应用程序完成对音频信号的预处理、调用 DPU 部署神经网络和对神经网络识别结果的处理。应用程序创建相应的 DPU 任务，通过有关 API 函数调用神经网络完成对声音事件的检测。首先加载音频并对音频进行预处理，从而得到频谱特征作为神经网络的输入。然后是打开 DPU 并创建 DPU 内核与任务，为运行 DPU 任务做准备。接着加载编译后的卷积神经算法，并运行相应的 DPU 任务，将频谱特征送入 DPU 内核中进行处理。然后是依次销毁 DPU 任务与内核，并关闭 DPU。最后通过对 DPU 的输出结果进行后处理，得到输出结果，从而完成对声音事件的识别。应用程序相应的流程如图 4-6 所示。

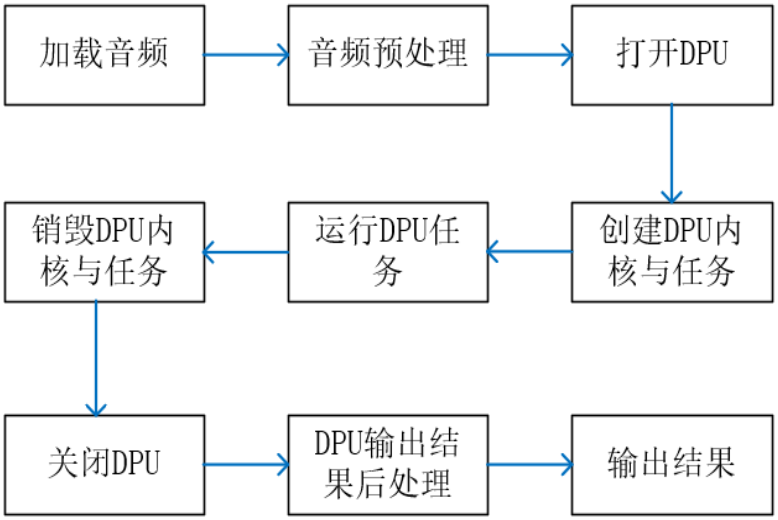

图 4-6 应用程序流程图

在音频信号的预处理部分，首先，修剪音频，剪辑音频的静音部分，并将音频调整为原始长度。然后，将剪辑处理后的音频进行处理以获得 logmel 频谱图。在此过程中，输入音频信号首先使用短时间傅立叶变换 (STFT) 进行处理，汉明窗口大小为 23.2 ms（44.1kHz 下的 1024 个样本）和 50% 重叠。然后处理后的音频信号通过一个带有 128 个三角带通滤波器的梅尔滤波器组和一个对数计算模块，得到 128×429（用于 ESC-10 和 ESC-50）和 128×343（US8K）的特征图。

由于神经网络的输入为 128*256，因此还需要对得到的特征图进行拆分处理，以得到符合神经网络输入的特征。将得到 128×429（ESC-10 和 ESC-50）和 128×343（US8K）的特征图拆分成多个新的频谱图，其中每一个新的频谱图包含 256 帧，50% 比例的重叠。对于最后一个频谱图，可以应用与倒数第二个频谱图的不同重叠比例以确保它的宽度为 256。最终，128×429（ESC-10 和 ESC-50）的特征

图将被拆分为 3 个 128×256 的特征图作为神经网络的输入，128×343（US8K）的特征图被拆分为 2 个 128×256 的特征图作为神经网络的输入。

在 DPU 输出结果后处理部分，对于 ESC-10/50 数据集，将 3 个 DPU 任务的输出的向量进行相加与取平均操作，使用处理后的结果来预测整段音频数据所对应的类别。对于 UrbanSound8K 则是通过平均 2 个频谱图的预测概率来评估整个音频数据。

由于电脑处理器为 X86 结构与使用的 ZCU104 开发板上的 PS 的 ARM 架构不同，所以需要编译链接。本设计使用 DNNC 工具对设计的卷积神经网络算法进行编译得到 elf 文件，然后使用 GCC 编译器将 elf 文件生成为.so 的动态链接库文件。当两种编译完成后，使用 Python 编程应用程序，在 python 文件中调用上述的.so 文件完成对 DPU 的调用，该.so 文件可以在 PS 端 ARM 结构上运行，同时也包含了带有神经网络算法的 DPU 信息。

4.2.4 实现流程

系统由 PS 与 PL 部分组成，其中 PL 部分负责 DPU 的部署，为神经网络的运行做好硬件准备。PS 部分为嵌入式 Linux 系统，应用程序在 PS 端运行，最后得到识别结果。系统的实现流程如图 4-7 所示。

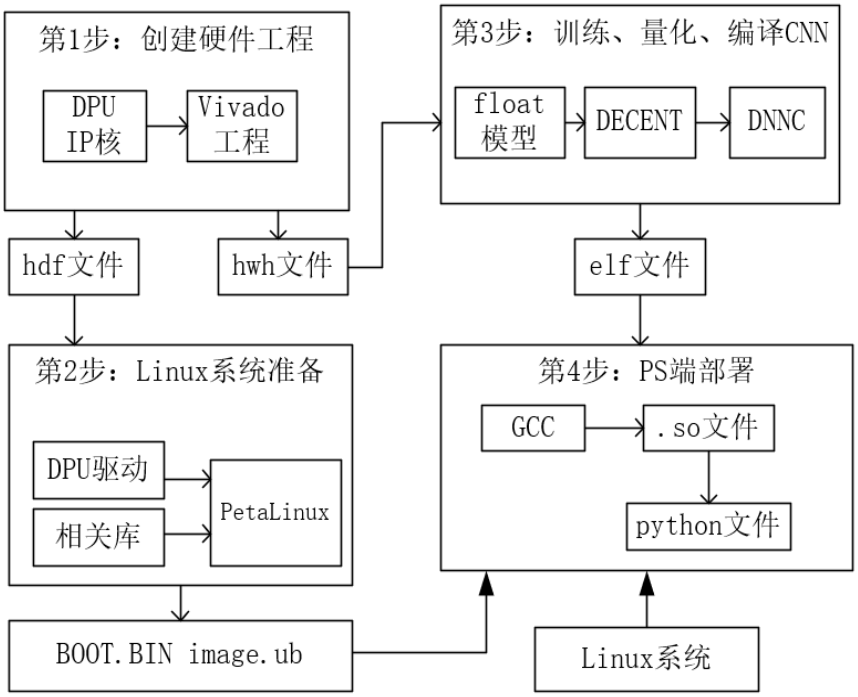

图 4-7 实现流程图

第一步为创建硬件工程，使用 Vivado2020 开发工具创建硬件工程，在工程中添加集成了 DPU 的 IP 核与 Zynq® UltraScale+，对相关信号线进行连接后生成包含硬件配置信息的 hdf 文件与 hwh 文件。第二步准备 Linux 系统，使用 PetaLinux 开发平台导入上一步生成的 hdf 文件，添加 DPU 有关驱动库和其他相关库，最后生成系统启动的 BOOT.BIN 文件和 image.ub 文件。第三步训量化和编译卷积神经网络模型，将训练后的浮点型模型使用 DECENT 工具进行量化，使用第一步生成的 hwh 文件与量化后模型通过 DNNC 工具进行编译生成 elf 文件。第四步为 PS 端部署，使用第二步生成的系统启动文件与 Xilinx 提供的 Linux 镜像文件生成 PS 端上运行的 Linux 系统，使用 GCC 将 elf 文件生成为动态链接库.so 文件，最后在 python 文件中进行调用。

4.3 实现结果

在处理完成板子上程序后，使用串口线和网线将开发板与 PC 进行连接，其中串口用于进入开发板的 Linux 系统，网线用于数据传输。开发板与 PC 连接如下图 4-8 所示。

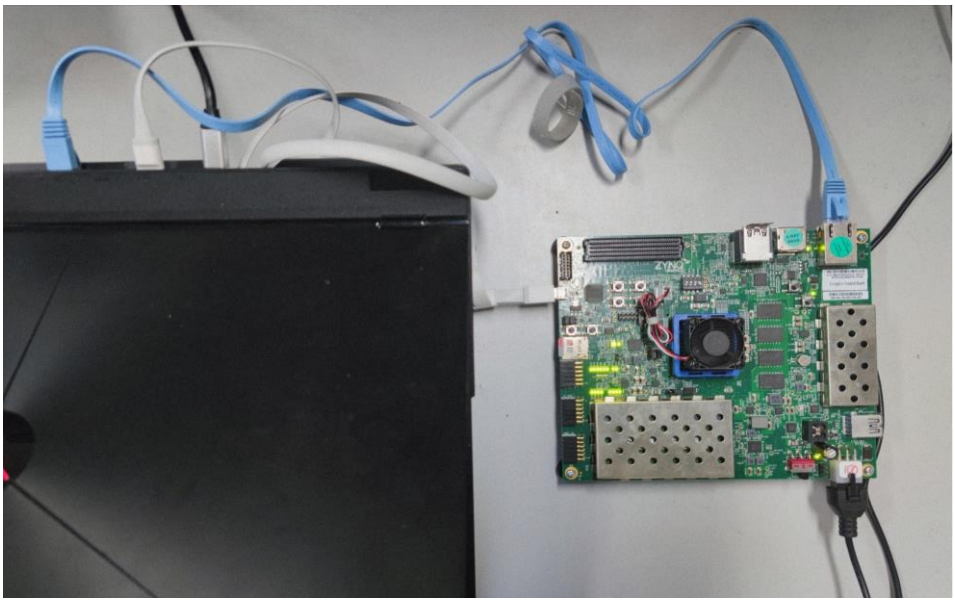

图 4-8 开发板与 PC 连接示意图

最后，依次从 ESC-10、ESC-50 和 UrbandSound8K 中随机选取 400 个音频进行测试。其中，由于 ESC-10 只有 400 个样本，因此选取了整个 ESC-10 数据集。在开发板上进行测试，在 PC 端的串口连接界面显示测试结果。ESC-10、ESC-50 和 UrbandSound8K 的测试结果分别如图 4-9、4-10 和 4-11 所示。

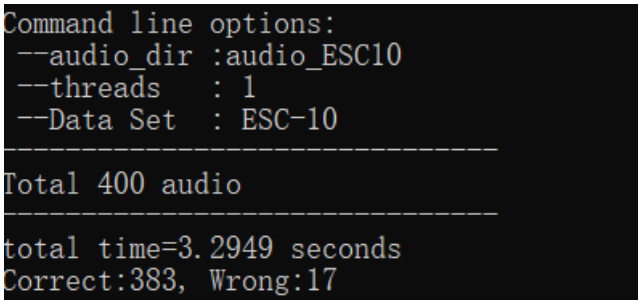

图 4-9 ESC-10 测试结果图

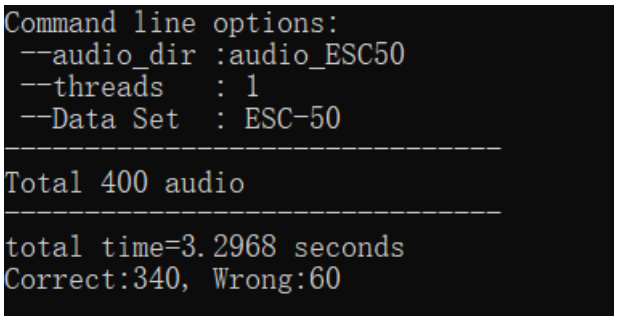

图 4-10 ESC-50 测试结果图

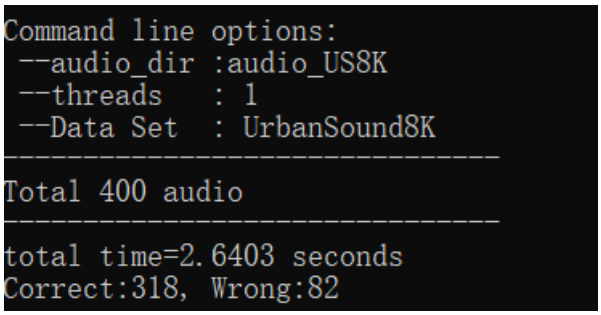

图 4-11 UrbanSound8K 测试结果图

从三个数据集的测试结果图中可以看出，ESC-10 中 400 个音频的总共识别时间为 3.2949s，ESC-50 中 400 个音频的总共识别时间为 3.2968s，UrbanSound8K 中 400 个音频的总共识别时间为 2.6403s。可以看出，ESC10 与 ESC-50 的识别时间接近，UrbanSound8K 的识别时间最短。这是因为，ESC10 为 ESC-50 的子集，它们每个音频的时间长度均为 5s，而 UrbanSound8K 的音频长度平均为 4s。在准确率上，ESC-10 准确率最高，UrbanSound8K 准确率最低。ESC50 与 ESC10 数据集中单个音频平均识别时间为 8.24ms（UrbanSound8K 为 6.6ms），完全满足实时性的需求。

从 ESC-50 的 fold4 中选择若干个音频进行测试，测试结果如图 4-12 所示。

```
Load audio: 4-172742-A-32.wav
Label: 32
Probability: 0.713    Result: 32

Load audio: 4-173865-A-9.wav
Label: 9
Probability: 0.7328    Result: 9

Load audio: 4-174797-A-15.wav
Label: 15
Probability: 0.7028    Result: 15

Load audio: 4-174860-A-3.wav
Label: 3
Probability: 0.8364    Result: 3
```

图 4-12 部分音频测试结果图

4.4 本章小结

本章节重点介绍了基于 FPGA-DPU 的声音事件检测系统的设计。首先对 FPGA-DPU 进行了简单介绍，然后就系统的设计实现中的各个模块进行了介绍，最后对 FPGA-DPU 的声音事件检测系统进行了测试。最终测试结果表明，在基于 FPGA-DPU 的声音事件检测系统中，ESC50 与 ESC10 数据集中单个音频平均识别时间为 8.24ms（UrbanSound8K 为 6.6ms），完全满足实时性的需求。

第五章 实验结果与分析

5.1 实验数据和开发环境

本小节将首先主要声音事件检测算法有关数据集和实验的开发环境。在数据集部分，将介绍声音事件检测中常用的三个数据集；在实验的开发环境部分，将介绍实验所使用的服务器信息与深度学习框架。

5.1.1 数据集介绍

在声音事件检测中常用的三个数据集分别为 ESC-50、ESC-10 和 UrbanSound8K。表 5-1 为三个声音事件数据集的对比。

表 5-1 常见的三个声音事件数据对比

数据集名称	样本量	平均时长	声音事件种类
ESC-50	2000	5s	50
ESC-10	400	5s	10
UrbanSound8K	8732	4s	10

ESC-50：

ESC-50 数据集是 2000 条环境录音的标记集合，适用于环境声音分类的基准方法。该数据集由 5 秒长的录音和 44.1 kHz 采样频率分为 50 个类别（每类 40 个示例），分为 5 个主要类别：“动物”、“自然音景和水声”、“人类/非语音”、“内部/家庭声音”和“外部 /城市噪音”。

ESC-10：

ESC-10 是从 ESC-50 数据集中选择的 10 个类（400 个样本,每类 40 个实例）的子集。10 个类别为：“狗吠”、“雨”、“海浪”、“婴儿哭泣”、“时钟滴答声”、“人打喷嚏”、“直升机”、“电锯”、“公鸡”和“火”。

UrbanSound8K ：

UrbanSound8K 数据集是来自 10 个不同类别的各种城市声音，数据集中有 8732 个短音频（即小于 4 秒的音频）。10 个类别分别为： “空调”、“汽车喇叭”、“儿童玩耍”、“狗吠”、“钻孔”、“发动机空转”、“枪声”、“手提钻”、“警报器”和“街头音乐”。剪辑具有不同的采样率、量化级别和通道。 所有波形都经过重新采样，采样率为 44.1 kHz、16 位量化和单声道声音。不同于 ESC-50 与 ESC-10 数据集，UrbanSound8K 数据集中的各个事件类别的数目有所不同，具体的事件类别分布见表 5-2。

表 5-2 UrbanSound8K 数据集声音事件类别分布表

事件类别	事件数目
空调	1000
汽车喇叭	429
儿童玩耍	1000
狗吠	1000
钻孔	1000
发动机空转	1000
枪声	374
手提钻	1000
警报器	929
街头音乐	1000
汇总	8732

对于声音事件检测，一般会使用 ESC-50、ESC-10 和 UrbanSound8K 数据集进行评测。

5.1.2 实验环境

本实验所使用的服务器的系统环境是 CentOS Linux release 8.4.2105，操作系统是 64 位，处理器是 Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz，内存是 256G，服务器总共有 7 块 Tesla V100 显卡，其单个显存为 32G，安装的显卡驱动版本为 460.73.01，安装的 CUDA 版本为 11.2。实验是在 PyTorch 的 Python 开源机器学习库中进行的。

实验所使用的深度学习框架为 PyTorch。PyTorch 是一个基于 Torch 的 Python 开源机器学习库，用于自然语言处理等应用程序。它主要由 Facebook 人工智能研究院(FAIR)在 2017 年推出，不仅能够实现强大的 GPU 加速，同时还支持动态神经网络，这一点是现在很多主流框架如 TensorFlow 都不支持的。目前，PyTorch 已兼容 Windows(CUDA,CPU)、MacOS(CPU)、Linux(CUDA,ROCm,CPU)。

5.2 数据增强与模型训练

为了有效利用有限的训练数据，首先将每个音频剪辑的原始 Logmel 频谱图（ESC-50/10 包含 429 帧，US8K 包含 343 帧）拆分成多个新的频谱图，其中包含 256 帧，50% 重叠。对于最后一个频谱图，可以应用与倒数第二个频谱图的不同重叠以确保它包含 256 帧。然后应用时间掩蔽[57]和混合[58]数据增强方法。掩蔽

操作是通过将频谱图中随机的 t 帧的值设置为 0 来屏蔽它们的。混合操作是通过混合两个掩蔽的频谱图来执行的，其公式如(5-1)中，

$$X = \alpha \times \{X_{masked}\}_i + (1-\alpha) \times \{X_{masked}\}_j \quad (5-1)$$

其中$\{X_{masked}\}_i$和$\{X_{masked}\}_j$是从先前的掩蔽频谱图中随机选择的两个样本。混合操作生成的新频谱图将在混合之前（掩蔽之后）添加到频谱图中以进行数据增强。

对于训练阶段，计算每个频谱图的概率以预测出 1 个声音类别，而在测试阶段，通过平均 K 个频谱图的预测概率来评估整个音频数据，ESC-10/50 的 K 为 3，US8K 的 K 为 2。在 LSED 的实验中实施了 N 折交叉验证方法，ESC-10/50 和 US8K 的 N 值分别为 5 和 10。ESC10、ESC50 和 UrbandSound8K 数据集的训练集、验证集和测试集的数量分布图分别如图 5-1、5-2 和 5-3 所示。

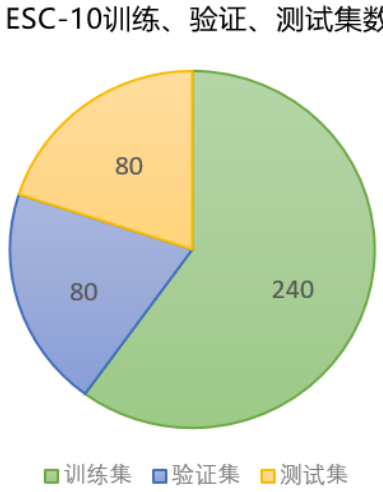

图 5-1 ESC-10 训练、验证与测试集分布图

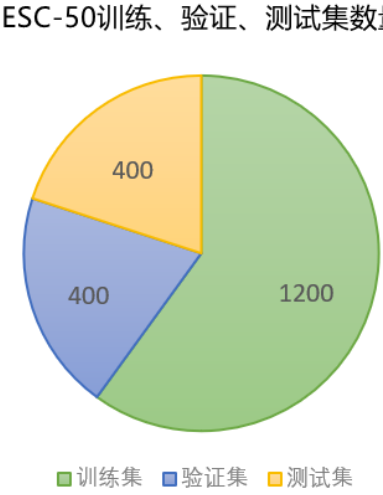

图 5-2 ESC-50 训练、验证与测试集分布图

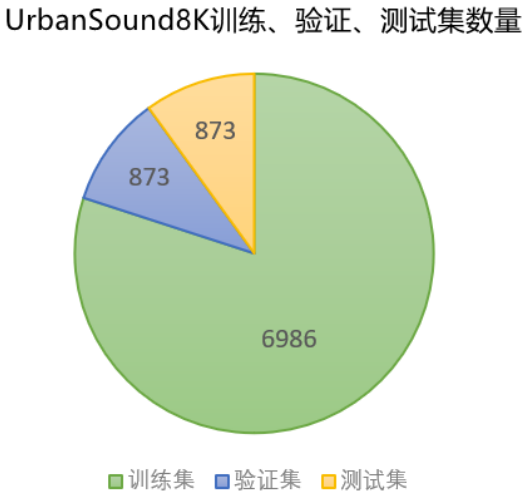

图 5-3 UrbanSound8K 训练、验证与测试集分布图

PyTorch 库用于训练所提出的模型，并通过使用 Nesterov 动量为 0.9 的随机梯度下降，minibatch 设置为 64，使用交叉熵损失函数来优化模型参数。该模型经过 80 个 epoch 的训练。 学习率初始化为 0.01，每 30 个 epoch 缩小 10 倍。

5.3 实验与结果分析

本小节将首先介绍声音事件检测的评价，然后就本文所提出的声音事件检测算法在常用三个数据集上的实验结果，并将其与其他有关研究进行对比，最后对实验进行分析。

5.3.1 评价指标

本小节主要介绍声音事件检测算法的相关评价指标。该指标主要分为两大类，即识别准确率与算法复杂度。下面将分类就两项评价指标进行介绍。

识别准确率：即在测试阶段中，预测正确的样本占所有测试样本的比例。一般认为，算法的识别准确率越高越好。

算法复杂度：对于神经网络而言，算法复杂度一般分类两项，即网络参数量与浮点运算数(FLOPs)。其中，浮点运算数(FLOPs)，可以理解为计算量。可以用来衡量算法/模型的复杂度。一般认为，算法的网络参数量和 FLOPs 越小该算法越好。

对于 CNN 而言，主要的 FLOPs 集中于卷积操作过程中。对于卷积层，标准卷积和深度可分离卷积过程中的参数量和 FLOPs 如本文 3.3.1 节所述和 3.2.2 所述。为了使神经网络的性能更加稳定，一般在卷积层之后添加归一化层进行数据的归一化处理。假设归一化层的输入为 $c * h * w$，那么该层的参数量为$2 * c$，其 FLOPs

为$c*h*w$。一般认为卷积层包括卷积操作和归一化操作，故将归一化层的参数量与 FLOPs 放入卷积层一同计算。对于池化层，池化层进行池化操作过程中不涉及到参数，同时一般不计算池化层的 FLOPs，所以一般在计算算法复杂度时不考虑池化层。对于全连接层，假设该全连接层输入的特征维度为a,输出的特征维度为b，则其参数量为$a*b$，其 FLOPs 为$a*b$。

以 ESC-50 数据集为测试数据集，提出的轻量级的声音事件检测算法 LSED 的详细配置如表 5-3 所示。在表 5-3 中，“输入大小”和“输出大小”有三维张量类型与二维张量类型。其中，三维张量类型中，对应的维度依次为通道维度、时间维度和频率维度；二维张量类型中，对应的维度依次为时间维度和通道维度。池化层中没有参数量与 FLOPs。

表 5-3 LSED 算法配置

网络层类别	卷积核	输入大小	输出大小	参数量	FLOPs
卷积层 0	3×1×1	(1,128,256)	(3,128,256)	3	98,304
块 1_卷积层	32×3×5	(3,128,256)	(32,128,256)	1,504	48,234,496
注意力块 1	-	(32,128,256)	(32,128,256)	856	199,680
块 1_池化层	4×6	(32,128,256)	(32,32,42)	-	-
块 2_卷积层	64×3×1	(32,32,42)	(64,32,42)	2272	2,967,552
注意力块 2	-	(64,32,42)	(64,32,42)	1,688	76,358
块 2_池化层	4×1	(64,32,42)	(64,8,42)	-	-
块 3_卷积层	128×1×5	(64,8,42)	(128,8,42)	41,216	13,805,568
注意力块 3	-	(128,8,42)	(128,8,42)	3,342	102,800
块 3_池化层	1×6	(128,8,42)	(128,8,7)	-	-
块 4_卷积层	256×3×3	(128,8,7)	(256,8,7)	34,432	1,913,856
注意力块 4	-	(256,8,7)	(256,8,7)	6,680	61,460
块 4_池化层	2×2	(256,8,7)	(256,4,3)	-	-
全连接层	3072×50	(3072,1)	50	153,600	153,600
合计				0.246M	67.5M

对于 ESC-50，是通过平均 3 个频谱图的预测概率来评估整个音频数据，故实际 FLOPs 应该是算法的 FLOPs 的 3 倍，故实际的 FLOPs 应该为 0.203G。

5.3.2 整体实验结果与分析

表 5-4 显示了本文提出的轻量级声音事件检测算法 LSED 的识别准确率与复杂度，以及与其他先进工作的比较。从表中可以看出，比较工作使用了 ESC-50、ESC-10 和 UrbanSound8K 数据集进行性能评估。

表 5-4 各算法性能对比

算法	识别准确率			算法复杂度	
	ESC-10	ESC-50	UrbanSound8K	参数量	FLOPs
LGTFB[20]	-	86.2%	83.4%	0.799M	0.8112G
Multi-StreamCNN[24]	94.2%	84.0%	-	137.58M	284.06G
ESResNet[25]	97.0%	91.5%	85.4%	23.61M	183.36G
ZhangCNN[26]	94.2%	86.5%	-	4.37M	0.485G
RethinkingCNN[59]	-	91.2%	85.1%	18.11M	1.62G
VGGishKD[60]	-	-	76.0%	1.88M	0.148G
SoundCLR[61]	99.6%	92.9%	85.8%	11.69M	258.72G
LSED	96.7%	87.3%	83.3%	0.246M	0.203G

从表 5-4 中可以看出，所有的算法均是 ESC-10 数据集识别准确度最高，UrbanSound8K 数据集的识别准确度最低。比较工作的参数个数为 0.246M ~ 137.58M，FLOPs 的数量从 0.148G 到 284.06G。

本文提出的 LSED 算法达到了最低的神经网络参数量，其参数量不足算法[60]的 14%，不足算法[61]的 3%。LSED 算法的 FLOPs 第二低，仅比算法[60]高一些，但它在 UrbanSound8K 数据集上比算法[60]识别准确率高出 7.3%。同时，本文提出的算法在三个数据集上均实现了较高的准确率（分别为 96.7%、87.3%和 83.3%），它的识别准确率也可以与其他算法相媲美。可以看出，所提出的算法具有较低的算法复杂度且识别准确率较高。

此外，在上一章节，使用了 400 个音频数据对基于 FPGA-DPU 的声音事件检测系统的识别速度进行了测试。最终测试结果显示：ESC50 与 ESC10 数据集中单个音频平均识别时间为 8.24ms（UrbanSound8K 为 6.6ms），完全满足实时性的需求。

从基于 FPGA-DPU 的声音事件检测系统的识别速度可以看出，提出的算法识别速度较快，延迟较低，能够满足对延迟和资源都有严格要求的物联网设备。

5.3.3 选择性可分离卷积实验

为了研究本文所提出的选择性可分离卷积机制对轻量级的声音事件检测算法LSED 的影响，本文进行了关于选择性可分离卷积机制的实验。

为此，在 LSED 算法的基础上，将所有卷积层全部设置为标准卷积，其余部分保持不变从而得到无选择性可分离卷积机制的 LSED 算法。使用了 ESC-50、ESC-10 和 UrbanSound8K 数据集进行性能评估，对无选择性可分离卷积机制的 LSED 算法测定其识别准确率与算法复杂度，并与 LSED 算法进行对比。选择性可分离卷积机制的有关实验结果如表 5-5 所示。

表 5-5 选择性可分离卷积机制实验结果

算法	识别准确率			算法复杂度	
	ESC-10	ESC-50	UrbanSound8K	参数量	FLOPs
无选择性可分离卷积的 LSED	97.1%	87.5%	83.5%	0.511M	0.276G
LSED	96.7%	87.3%	83.3%	0.246M	0.203G

从表 5-5 中可以看出，无选择性可分离卷积的 LSED 算法中，依旧是 ESC-10 的识别准确率最高，UrbanSound8K 的识别准确率最低。使用了选择性可分离卷积机制后的 LSED 算法在 ESC-10、ESC-50 和 UrbanSound8K 三个数据集上的识别准确率均略微低于无选择性可分离卷积机制的 LSED 算法，但却可以大幅度降低算法复杂度。以 ESC-50 数据集为例，使用了选择性可分离卷积机制后，识别准确率仅仅降低 0.2%，但是算法的参数量不到未使用选择性可分离卷积算法的 50%，FLOPs 不足其 75%。

故可以认为：使用选择性可分离卷积机制，能够在基本不降低声音事件识别准确率的基础上，较大程度地降低声音事件检测算法的复杂度。

5.3.4 协调注意力机制实验

为了研究本文所提出的协调注意力机制对轻量级的声音事件检测算法 LSED 的影响，本文进行了关于协调注意力机制的实验。为此，在 LSED 算法的基础上移除协调注意力机制，其余部分保持不变得到无协调注意力机制的 LSED 算法。使用了 ESC-50、ESC-10 和 UrbanSound8K 数据集进行性能评估，对无协调注意力机制的 LSED 算法测定其识别准确率与算法复杂度，并与 LSED 算法进行对比。协调注意力机制实验结果如表 5-6 所示。

表 5-6 协调注意力机制实验结果

算法	识别准确率			算法复杂度	
	ESC-10	ESC-50	UrbanSound8K	参数量	FLOPs
无协调注意力机制的 LSED	96.1%	85.8%	82.1%	0.233M	0.201G
LSED	96.7%	87.3%	83.3%	0.246M	0.203G

从表 5-6 中可以看来，使用了协调注意力机制后的 LSED 算法在 ESC-10、ESC-50 和 UrbanSound8K 三个数据集上的识别准确率均高出无协调注意力机制的 LSED 算法。以 ESC-50 数据集为例，使用了协调注意力机制后，识别准确率得以提升 1.5%；而算法的参数量仅增加 0.013M，其 FLOPs 仅增加 0.002G，使用协调注意力机制所带来的算法复杂度的增加几乎可以忽略不计。

故可以认为：使用协调注意力机制，能够在基本不增加声音事件检测算法复杂度的基础上，对声音事件识别准确率有一定的提升。

5.4 本章小结

本章主要是对提出的轻量级声音事件检测模型进行了测试。首先，介绍了声音事件检测常用的数据集合实验环境；接着，介绍了对音频数据的数据增强操作和实验有关的训练指标；最后，通过实验来分析轻量级声音事件检测算法 LSED 的性能，在该部分首先介绍了实验的有关评价指标，然后将 LSED 算法与其他前沿算法进行对比并测定了基于 FPGA-DPU 的声音事件检测系统的识别速度，最后就选择性可分离卷积与协调注意力机制单独做了实验。实验结果表明，提出的轻量级声音事件检测算法 LSED 拥有较少的参数量、FLOPs，同时还拥有较高的识别准确率，能够满足对延迟和资源都有严格要求的物联网设备。

第六章 总结与展望

6.1 研究内容总结

随着信息科技的飞速发展，现代科技在逐渐运用到人类生活的各个领域之中，为人类提供更加幸福的生活。声音事件检测（Sound Event Detection, SED）是利用声音信号的特征去预测当前声音事件种类的技术，它在智能家居、公共安全等领域具有较为广阔的应用前景。

传统的声音事件检测技术一般基于 GMM-HMM 模型，其识别准确率较低，且编解码计算复杂度较大，难以在实际生活中得到应用。与传统的机器学习方法相比，近年来国内外研究人员提出了基于神经网络（Neural Network, NN）的检测方法，显著提高了识别准确率。然而，基于 NN 的 SED 算法的一个主要问题是它们通常涉及大量参数和浮点运算数(FLOPs)，从而导致较高的处理延迟与硬件开销，使得该类方法一般难以适用于对资源和延迟都有严格要求的物联网设备。因此，构建网络复杂度低且识别性能较高的声音事件检测算法成为本文的研究重点。论文提高了一种低复杂度高准确率的轻量级声音事件检测算法，并基于此算法构建了声音事件检测系统。

本文首先介绍了声音事件检测的研究背景及意义。通过查阅大量与声音事件检测有关的文献，总结了声音事件检测的国内外研究历史与现状。重点就基于神经网络的声音事件检测技术做了阐述。

之后，本文就声音事件检测的基本原理进行了阐述。声音事件检测系统一般包括特征提取、神经网络模型、系统输出。简单介绍了常见的声音特征提取与检测模型有关理论知识。

随后，针对目前声音事件检测算法复杂度高的问题，论文提出了一种低复杂度高准确率的轻量级声音事件检测算法，该算法运用了选择性可分离卷积机制与协调注意力机制。

针对于声音事件检测算法复杂度较高的问题，本文使用了一种选择性可分离卷积机制。选择性可分离卷积是指针对卷积层所在位置的特点选择性采用不同的卷积方案。该机制能够有效保证拥有较少参数量和 FLOPs 的同时还拥有较高的声音事件检测准确率。

为了进一步所构建的声音事件检测模型的识别性能，论文使用了一种协调注意力机制。该机制可同时作用在通道域、时域和频域，让检测模型重点关注与声音事件检测有关的特征和区域，减少对无用的特征图通道域、时域和频域的关注。采

用协调注意力机制可以保证在少量增加模型复杂度的基础上大幅度提高模型的识别性能。

然后，将提出的轻量级声音事件检测算法通过 FPGA 的深度学习处理单元(DPU)进行了实现，从而实现一个基于 FPGA-DPU 的声音事件检测系统。该系统基于 ZCU104 平台来开发设计的，通过使用 Vivado2020 与 PetaLinux 开发平台完成 DPU 的部署，使用 DNNDK 编程有关应用程序的开发。

最后，在常用的声音事件检测数据集（ESC-50、ESC-10 和 UrbanSound8K）上进行了测试与分析。实验结果表明，本文提出的轻量级声音事件检测算法的总参数量仅为 0.246M，模型的 FLOPs 仅为 202M，ESC50 与 ESC10 数据集中单个音频平均识别时间为 8.24ms（UrbanSound8K 为 6.6ms），完全满足实时性要求。

6.2 未来研究展望

虽然论文提出的声音事件检测模型有较低的复杂度与较高的识别性能，但是论文的研究内容有限，依旧有很多值得进一步优化的地方。声音事件检测也是目前的研究热点，声音事件检测技术还存在诸多问题有待解决，可以在未来就以下问题进行研究。

首先，存在着声音事件检测的数据的数量与质量的问题。目前主流的声音事件检测模型都是基于神经网络的，而神经网络是基于数据驱动的，数据的数量与质量相当重要。目前，声音事件检测常用的 3 种数据集 ESC-10、ESC-50、UrbanSound8K 的数据量分别为 400、2000、8732。可以看出，这些数据集存在数据量较少的问题，且 UrbanSound8K 还存在着数据分布均衡的问题，不同类别的数据量不同。然后，声音事件检测网络放在现实环境中进行测试，准确率往往会比数据集上测试准确率低很多。表明，数据集与实际环境数据之间存在差异。

其次，需要设计可以接收任意长度的音频的数据的声音事件检测模型。目前，声音事件检测模型都是基于数据集来设计的，接收的数据长度即为数据集中数据的长度。一般是模型在训练和测试时接收长度为 5s 的 ESC-10 和 ESC-50 中的数据样本，接收长度为 4s 的 UrbanSound8K 中的数据样本。就实际生活场景而言，声音事件的数据长度往往是不同的，因此需要设计可以接收众多长短不一的音频数据的声音事件检测模型，增强系统的灵活性。

另外，本文所使用的数据集均为标准的数据集，与实际环境中的数据集有所不同，未来应该聚焦于收集真实环境中的声音事件数据集并设计相关算法。在真实环境中，存在大量多声音混合的情况，在未来需要对多声音事件进行研究。

然后，需要对音频信号的初级特征提取进行研究。目前，对声音事件检测系统的研究大多聚集于对声音事件检测算法的研究，而对于音频信号的声音特征提取的研究较少。当前主流的声音特征提取的方法为 Log-Mel 和 MFCC 特征。这些特征提取所使用的滤波器的参数一般是固定，不因音频的改变而改变，这显然不符合声音事件检测的特征。因此，需要设计可以根据音频自主学习的初级特征提取的模型来完成声音事件检测。

最后，需要将算法迁移到硬件设备上进行实际场景的应用。当前人工智能大多停留在实验室阶段，难以在实际场景中得到应用。声音事件识别在实际生活中有着广阔的应用场景。如果能在综合考虑设备的存储、性能和成本，考虑算法的复杂度与实现难度的基础上，将算法部署到便携性物联网设备中，将进一步增强该技术的实用性。

致　谢

岁月匆匆而过，不知不觉在电子科大待了三年。犹记得三年前过来电子科大复试时的场景，犹记得三年前在华师图书馆刷题备战考研的场景，犹记得三年前从本科毕业来到电子科大的场景。三年，身边出现了不少人和事，也在逐渐影响着我，在此，向三年经历中的同学和老师们表达内心的感激之情。

首先，感激自己的导师周军教授。在这三年时间里，周军老师教会了许多道理，有些学问上的，也有些生活上的。自己在学术科研方面并不是那么优秀，相反可能还有点愚笨。感谢周老师三年来不紧不慢地教导与学术指导，让我开始一点点养成独立思考的习惯，帮助我一点点提高自己的学术科研能力。同时，也感谢课题组的常亮老师，带领小组们参加比赛，在比赛期间给予了充分的指导与鼓励，也正是因此最终我们才得以进入决赛。

其次，感谢阙禄颖、贾丛含学长与语音课题组的小伙伴们，正是有你们，才使得研究生生活多了份欢声笑语。同时，感谢同届的陈泳吉、高昕毅、刘青松等同学们，感谢你们在学习生活中对我的帮助。

然后，感谢我的朋友，刘尚奇、赵进、黄彬等朋友，在我成功的时候为了鼓掌开心，在我失落低沉的时候为我分析和出谋划策。

最后，特别感谢自己的父母与家人！父母虽然文化水平不高，但对我的关爱、支持、鼓励却一直没少，感谢您们多年来的付出。

最后，感谢祖国。正是有你们的奋斗与努力，如我这等平凡的农村子弟才有了相对公平的读书机会，才可以从一个贫困农村的孩童逐步成为电子科大研究生，才有了追求美好生活的机会。也正是有你们的坚持与努力，一次次疫情才没能将我们的社会打垮，众多平凡的人民才得以在如此严重的疫情下生活下来。

感谢这三年来所走的路、所遇的人，所经历的事，这都会成为一段美好的回忆。最后，祝大家身体健康、生活愉快，愿疫情尽快过去。

参考文献

[1] Wang, Yun. "Polyphonic sound event detection with weak labeling." *PhD thesis* (2018).

[2] J. Liaw, W. Wang, H. Chu, M. Huang and C. Lu, "Recognition of the Ambulance Siren Sound in Taiwan by the Longest Common Subsequence," in IEEE International Conference on Systems, Man, and Cybernetics,Manchester, 2013, pp. 3825-3828.

[3] F. Vesperini, D. Droghini, E. Principi, L. Gabrielli and S. Squartini, "Hierarchic Conv Nets Framework for Rare Sound Event Detection," in 26th European Signal Processing Conference (EUSIPCO), Rome, 2018, pp. 1497-1501.

[4] Cristani M, Bicego M, Murino V. Audio-visual event recognition in surveillance video sequences[J]. IEEE Transactions on Multimedia, 2007, 9(2): 257-267.

[5] 张璐璐. 视频监控终端系统声音检测及告警功能软件设计[D]. 杭州: 浙江大学, 2013.

[6] Portelo J, Bugalho M, Trancoso I, et al. Non-speech audio event detection[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, China, 2009: 1973-1976.

[7] Y. Wang, F. Metze. A first attempt at polyphonic sound event detection using connectionist temporal classification[C]. 2017 ieee international conference on acoustics, speech and signal processing (icassp), 2017, 2986-2990

[8] Y. Wang, L. Neves, F. Metze. Audio-based multimedia event detection using deep recurrent neural networks[C]. 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2016, 2742-2746

[9] F. Kraft, R. Malkin, T. Schaaf, et al. Temporal ica for classification of acoustic events ia kitchen environment[C]. Ninth European Conference on Speech Communication and Technology, 2005,10-16

[10] Lafay G, Lagrange M, Rossignol M, et al. A morphological model for simulating acoustic scenes and its application to sound event detection[J]. IEEE/ACM Transactionson Audio, Speech, and Language, 2016, 24(10): 1854-1864.

[11] Heittola T, Mesaros A, Eronen A, et al. Context-dependent sound event detection[J]. EURASIP Journal on Audio, Speech, and Music Processing, 2013, 2013(1): 1-13.

[12] X. Xia, R. Togneri, F. Sohel, et al. Random forest classification based acoustic eventdetection[C]. 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, 163-168

[13] M.-W. Mak, S.-Y. Kung. Low-power svm classifiers for sound event classification onmobile devices[C]. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), 2012, 1985-1988

[14] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine,2012, 29(6): 82-97.

[15] Zhang H, McLoughlin I, Song Y. Robust sound event recognition using convolutional neural networks[C]. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). South Brisbane, QLD, Australia, 2015: 559-563.

[16] Phan H, Hertel L, Maass M, et al. Robust audio event recognition with 1-max pooling convolutional neural networks[C]. Interspeech, 2016, Beijing, China, 2016: 3653–3657

[17] Piczak K J. Environmental sound classification with convolutional neural networks[C]. 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). Boston,MA, USA, 2015: 1-6.

[18] Sharma J, Granmo O C, Goodwin M. Environment Sound Classification Using Multiple Feature Channels and Attention Based Deep Convolutional Neural Network[C]//Interspeech. 2020: 1186-1190.

[19] Su Y, Zhang K, Wang J, et al. Environment sound classification using a two-stream CNN based on decision-level fusion[J]. Sensors, 2019, 19(7): 1733.

[20] Park H, Yoo C D. CNN-based learnable gammatone filterbank and equal-loudness normalization for environmental sound classification[J]. IEEE Signal Processing Letters, 2020, 27: 411-415.

[21] Abdoli S, Cardinal P, Koerich A L. End-to-end environmental sound classification using a 1D convolutional neural network[J]. Expert Systems with Applications, 2019, 136: 252-263.

[22] Zhu B, Wang C, Liu F, et al. Learning environmental sounds with multi-scale convolutional neural network[C]//2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018: 1-8.

[23] Li S, Yao Y, Hu J, et al. An ensemble stacked convolutional neural network model for environmental event sound recognition[J]. Applied Sciences, 2018, 8(7): 1152.

[24] Li X, Chebiyyam V, Kirchhoff K. Multi-stream network with temporal attention for environmental sound classification[J]. arXiv preprint arXiv:1901.08608, 2019.

[25] Guzhov A, Raue F, Hees J, et al. Esresnet: Environmental sound classification based on visual domain models[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 4933-4940.

[26] Zhang Z, Xu S, Zhang S, et al. Learning attentive representations for environmental sound classification[J]. IEEE Access, 2019, 7: 130327-130339.

[27] Salamon J, Bello J P. Deep convolutional neural networks and data augmentation for environmental sound classification[J]. IEEE Signal processing letters, 2017, 24(3): 279-283.

[28] Madhu A, Kumaraswamy S. Data augmentation using generative adversarial network for environmental sound classification[C]//2019 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019: 1-5.

[29] Tokozume Y, Ushiku Y, Harada T. Learning from between-class examples for deep sound recognition[J]. arXiv preprint arXiv:1711.10282, 2017.

[30] H. Ankişhan, "An approach to the classification of environmental sounds by LSTM based transfer learning method," 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302398.

[31] K. Imoto, N. Tonami, Y. Koizumi, M. Yasuda, R. Yamanishi and Y. Yamashita, "Sound Event Detection by Multitask Learning of Sound Events and Scenes with Soft Scene Labels," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 621-625, doi: 10.1109/ICASSP40776.2020.9053912.

[32] J. Bajzik and R. Jarina, "Exploiting hierarchy in environmental sound classification," 2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA), 2022, pp. 1-4, doi: 10.1109/RADIOELEKTRONIKA54537.2022.9764900.

[33] D. S. Johnson et al., "DESED-FL and URBAN-FL: Federated Learning Datasets for Sound Event Detection," 2021 29th European Signal Processing Conference (EUSIPCO), 2021, pp. 556-560, doi: 10.23919/EUSIPCO54536.2021.9616102.

[34] U. Jithendra, U. Mittal and P. Chawla, "Audio Detection using Mel-frequency Cepstral Coefficients," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596443.

[35] A. Ahmed, Y. Serrestou, K. Raoof and J. -F. Diouris, "Sound event classification using neural networks and feature selection based methods," 2021 IEEE International Conference on Electro Information Technology (EIT), 2021, pp. 1-6, doi: 10.1109/EIT51626.2021.9491869.

[36] E. -L. Tan, F. A. Karnapi, L. J. Ng, K. Ooi and W. -S. Gan, "Extracting Urban Sound Information for Residential Areas in Smart Cities Using an End-to-End IoT System," in IEEE Internet of Things Journal, vol. 8, no. 18, pp. 14308-14321, 15 Sept.15, 2021, doi: 10.1109/JIOT.2021.3068755.

[37] F. Paissan, A. Ancilotto, A. Brutti and E. Farella, "Scalable Neural Architectures for End-to-End Environmental Sound Classification," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 641-645, doi: 10.1109/ICASSP43922.2022.9746093.

[38] Y. Xu, Q. Kong, W. Wang, et al. Large-scale weakly supervised audio classification using gated convolutional neural network[C]. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2018, 121-125

[39] Joshi, Pankaj, et al. "Time Aggregation Operators for Multi-label Audio Event Detection." INTERSPEECH. 2018.

[40] T. Iqbal, Y. Xu, Q. Kong, et al. Capsule routing for sound event detection[C]. 2018 26th European Signal Processing Conference (EUSIPCO), 2018, 2255-2259

[41] J. Yan, Y. Song, W. Guo, et al. A region based attention method for weakly supervised sound event detection and classification[C]. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, 755-759

[42] Y. Wang, J. Li, F. Metze. A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling[C]. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, 31-35

[43] Wang Y. D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533-536

[44] 杨毅明. 数字信号处理(第 2 版) [M]. 北京: 机械工业出版社,2017

[45] 韩纪元, 张磊, 郑铁然. 语音信号处理（第三版）[M]. 清华大学出版社, 2013

[46] 邱子璇. 基于神经网络的声纹识别研究[D]. 成都:电子科技大学, 2019.

[47] 刘亚明. 基于深层神经网络的多声音事件检测方法研究[D]. 合肥: 中国科学技术大学,2019.

[48] 汤保龙. 基于深度学习的音频事件检测方法研究[D]. 成都: 电子科技大学, 2020.

[49] 赵力. 语音信号处理[M]. 北京: 机械工业出版社, 2003.

[50] I. Goodfellow, Y. Bengio, A. Courville, et al. Deep learning[M]. MIT press Cambridge, 2016

[51] 阿斯顿·张, 李沐, 扎卡里·C. 立顿, et al. 动手学深度学习[M].人民邮电出版社, 2019

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

[53] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[54] A. G. Howard, M. Zhu, B. Chen, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[C]. CVPR, 2017

[55] S. Woo, J. Park, J.-Y. Lee, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European conference on computer vision (ECCV), 2018, 3-19

[56] Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13713-13722.

[57] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, Specaugment: A simple data augmentation method for automatic speech recognition, arXiv preprint arXiv:1904.08779.

[58] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk m-inimization, in: International Conference on Learning Representations, 2018

[59] K. Palanisamy, D. Singhania, A. Yao, Rethinking cnn models for audio classification, arXiv preprint arXiv:2007.11154.

[60] G. Cerutti, R. Prasad, A. Brutti, E. Farella, Neural network distillation on iot platforms for sound event detection, in: Interspeech, 2019, pp. 3609–3613.

[61] A. Nasiri, J. Hu, Soundclr: Contrastive learning of representations for improved environmental sound classification, arXiv preprint arXiv:2103.01929

攻读硕士期间取得的成果

科研成果：

Yang, M., Peng, L., Liu, L., **Wang, Y.**, Zhang, Z., Yuan, Z., & Zhou, J. (2022). LCSED: A low complexity CNN based SED model for IoT devices. *Neurocomputing*, *485*, 155-165. (SCI 二区期刊, 影响因子:4.438)

竞赛成果：

2020 年“强芯健魂、铸基智能”智能计算基础平台挑战赛决赛优胜奖

2020 年研究生人工智能创新大赛电子科技大学校内赛三等奖