嵌入式系统项目设计开题报告

学　生 1：　　闫文杰 21121947

学　生 2：　　陈鲲鹏 21121404

学　生 3：　　张博阳 21122177

学　生 4：　　晋远帆 21121911

学　生 5：　　贺 健 21122174

组　　长：　　闫文杰 21121947

指导老师：　　刘凯

完成时间：　　2024 年 5 月

一、 课题内容概述

随着无线通信技术的迅速发展，通信系统中采用了多种复杂的调制方式。调制识别作为无线通信领域的一个重要问题，对于信号分析、频谱管理以及安全监测具有重要意义。传统的调制识别方法往往依赖于专业领域知识和手工设计的特征提取器，而基于深度学习的调制识别方法则能够自动地学习信号的特征，具有更好的泛化能力和适应性。

本课题使用 HackRF One 无线电接收模块进行信号接收，通过 PC 模型训练，并且使用 NVDIA 嵌入式开发板 jetson nano 进行模型部署与进一步的加速优化，从而实现本项目的调制识别的功能。本项目旨在完成产品化，并且加深对于信号调制、信号接收、模型构建部署等流程的了解。

该课题的核心问题在于如何利用深度学习技术结合 HackRF One 无线电接收模块实现对无线通信中的调制方式进行准确识别。包括如何获取真实的无线信号数据，并对其进行预处理（信号采样、去噪、归一化等），以便于深度学习模型的训练和识别；如何设计适用于调制识别的深度学习模型与针对调制识别任务的损失函数和评价指标；如何将训练好的模型导出并且部署到嵌入式平台上以及如何利用 NVDIA 的优势进行推理加速等附加功能。

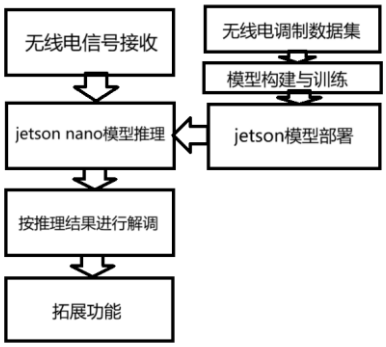
二、 课题相关市场调研

调制识别技术是无线通信领域的重要研究方向之一。随着物联网、5G 等技术的快速发展，对于调制方式的准确识别需求日益增加。在国内外，有许多研究机构和企业致力于调制识别技术的研究与开发；主要集中在通信领域的大学、研究机构以及通信设备制造商。一些国外公司和研究机构已经开发出了基于深度学习的调制识别系统，并在通信设备、智能无线电等领域得到应用。在过去的 20 多年里，国内外围绕模拟和数字调制的识别开展了大量的工作，并取得不少研究成果。比较经典的两类自动调制识别算法：基于似然估计和基于特征的方法，得到了广泛应用。

产品方面，针对调制识别的产品也在不断涌现，涵盖了从数据处理到模型训练的各个环节。这些软件产品为调制识别技术的研究和应用提供了便利；除此之外智能通信设备作为调制识别技术的应用领域之一，市场潜力巨大。随着 5G、物联网等技术的发展，智能通信设备的需求将会不断增加，从而推动调制识别技术的应用与发展。

三、 课题实施方案

3.1 叙述课题设计的基本技术路线



1.数据采集与准备：使用 HackRF One 无线电接收模块进行无线信号的采集与预处理。

2.深度学习模型设计与训练：设计适用于调制识别的深度学习模型。

3.模型部署与应用：将训练好的深度学习模型部署到嵌入式平台 Jetson Nano 上。

4.优化与改进：使用 tensorRT 对模型进行转换以实现加速效果。

3.2 课题总体设计方案

3.2.1.硬件平台选型：

本次课题的研究对象为短波无线电信号，其频率范围在 3MHz-30MHz。经过对几款主流的软件无线电平台的调研，有如下对比：

表 1 不同硬件平台对比

硬件平台	HackRF One	BladeRF x40	USRP
RF 范围	10MHz - 6GHz	300MHz - 3.8GHz	50MHz - 6GHz
带宽	20MHz	28MHz	61.44MHz
双工	半双工	全双工	全双工
ADC 采样位数	8bit(正交)	12bit(正交)	12bit(正交)

ADC 最高采样率	20Mbps	40Mbps	61.44Mbps
接口	高速 USB2.0	USB3.0	USB3.0
预算	$300	$420	$675

经比较，HackRF One 的频率范围、带宽、采样分辨率等均能满足本次项目要求，不足的是 USB2.0 的接口最高传输速率为 32Mbytes/s，限制了一些情形下对高带宽信号处理的应用。且 HackRF One 整体的运行代码和架构都是开源的，可以兼容 GNU Radio 等软件平台，对于 HackRF One 和 GNU Radio 结合进行信号的收集以及处理，网上有较多的开源模块，方便进行调用。最终结合对预算控制的需求，选择 HackRF One 作为本次项目的软件无线电平台。

本次课题的无线电信号接收和处理部分在 Ubuntu 操作系统下基于 HackRF One 硬件平台结合 GNU Radio 软件平台实现。

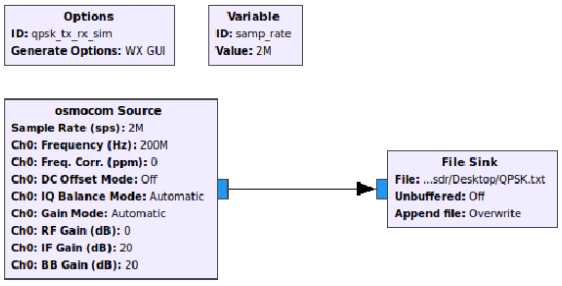


图 4.8 GNU Radio 接收端流图

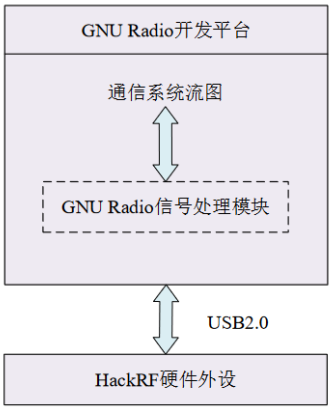


图 2（左）GNURadio 接收流图

图 3（右）系统结构框图

GNU Radio 内部提供了大量的数字信号处理工具、无线通信协议和调制解调器等功能。例如 osmocom Source 模块，该数据流块能实现对 HackRF One 进行驱动，使其处于接收信号的状态，并将收到的信号采样后输出，通过配置该模块的参数可调整接收信号的频率范围、采样率及数据格式等。osmocom Source 模块输出的信号可通过 File Sink 模块保存在指定的路径下，简单原理图如图 2 所示。

需要注意的是，HackRF 可以处理的最大带宽为 20Mhz，如果要对大于 20M 带宽的信号进行扫频，可以对采集的信号做分数倍采样率转换后从而达到采样目的。

3.2.2 模型与框架方案

我们希望设计一种神经网络结构，能够有效地检测 I/Q 信号的调制方案。该网络比处理相同或类似任务的其他架构的数量级更轻，此外，训练参数的数量不取决于信号持续时间。

通过论文研读我们决定采用：LModCNNResNet Relu 模型，一种使用残差网络结构和 ReLU 激活函数的调制识别模型。理由如下：

1. 数量级：

通过比较以下五个模型，我们可以发现 Mod-LRCNN 和 Mod-LCNN 网络所需的参数较少，并且对于这两个网络，参数的数量并不取决于信号的持续时间

Number of signal samples	RML-ConvNet	RML-CNN/VGG	RML-ResNet	Mod-LCNN (ours)	Mod-LRCNN (ours)
128	2,829,399	199,111	179,303	37,487	97,663
1024	21,179,479	256,455	236,647	37,487	97,663

表 3.2.2.1 五个不同网络参数的数目

2. 准确性：下面这张表给出了五个不同神经网络的数据集的测试集的准确率，以百分比表示。所有网络在 RadioML2016.04c 上的表现都相对相同。但 RMLConvNet 和 RML-CNN/VGG 在其他数据集上无法达到不如其他网络的良好性能。这是由于第一个网络的参数数量太多，导致了训练集的过拟合。对于第二个网络，这可以用网络的深度来解释。本研究中开发的 Mod-LCNN 在 AugMod 数据集上进行测试时优于或等于 RML-ResNet，但在 RadioML2018 上未能给出良好的结果。这可以用参数的数量太少来解释。添加更多的层会产生太深的架构，从而降低性能。

dataset	RML-ConvNet	RML-CNN/VGG	RML-ResNet	Mod-LCNN (ours)	Mod-LRCNN (ours)
128 samples					
RadioML2016.04C	93	92	94	93	**94**
RadioML2016.10A	84	83	90	89	**91**
RadioML2016.10B	88	90	92	92	**92**
RadioML2018.01A	50	69	76	68	**76**
AugMod (ours)	61	56	65	**75**	72
1024 samples					
RadioML2018.01A	61	87	88	85	**89**
AugMod (ours)	68	14	78	**83**	82

表 3.2.2.2 五种不同的神经网络结构在不同的数据集上的准确性

（一次输入给模型的样本数量为 512，执行 200 次迭代）

3. 信噪比：在 AugMod 数据集上，我们比较了每个神经网络在分类调制时的性能，作为信噪比的函数。结果给出了错误率作为信噪比的函数。为本研究开发的 Mod-LRCNN 在信噪比=0 上比以往研究的最佳体系结构 RML-ResNet 的性能好 40%以上。在[0,30]的 SNR 范围内，Mod-LRCNN 有效地提高了 5 dB 的性能

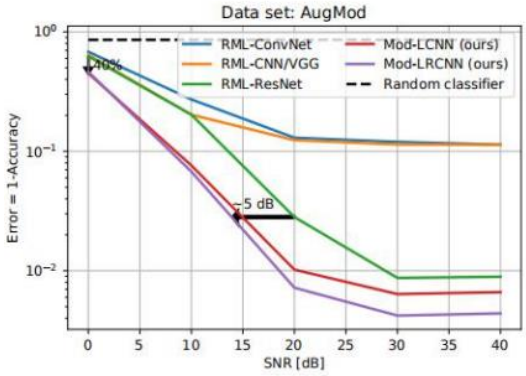


表 3.2.2.4 五种不同的神经网络架构的错误率（横轴：输入信号信噪比 SNR）

4. 训练速度：我们通过观察在 AugMod 数据集上运行时的每个纪元的时间来评估训练时间，其中采样点个数：1024。其他数据集也给出了类似的结果。Mod-LRCNN 每个示例运行 3.1 ms，每个历元 27 秒（一轮训练所需的时间），每批 512 个样本，总训练时间为 1.5 小时，200 个时代。 Mod-LCNN 和 RML-CNN/VGG 的速度是前者的两倍，而 RML-ResNet 的速度是后者的 1.25 倍。最后，由于数量大，RML-ConvNet 运行时间是两倍与后者。与 Mod-LCNN 相比，Mod-LRCNN 运行每个历元的时间是 LCNN 的两倍，这一事实被其更高的精度和需要更少的历元来收敛相平衡。

3.3 课题相关软件结构和主要模块流程设计

3.3.1 训练推理架构

1.训练推理：

训练推理的具体流程如下：代码的运行流程如下：首先，根据所选择的数据集名称，加载相应的数据集。数据集可以是 AugMod、RadioML2016.04c、RadioML2016.10a、RadioML2016.10b 或 RadioML2018.01a。加载数据集后，将信号数据、标签、信噪比等信息存储在相应的变量中。接下来，对数据进行预处理和准备工作。可以选择对信号进行切割，设置信号的持续时间，并进行信噪比的筛选。然后，将数据集划分为训练集和测试集，并进行相应的数据转换和处理。接着，定义了一个训练模型的函数。在该函数中，首先清除之前的模型，然后使用指定的深度学习模型架构进行模型的实例化。接着，使用训练集进行模型的训练，设置了训练的轮数和批次大小，并在训练过程中记录了训练时间。训练完成后，将模型保存到指定的路径中。最后，根据选择的深度学习模型架构，调用训练模型的函数进行模型训练。

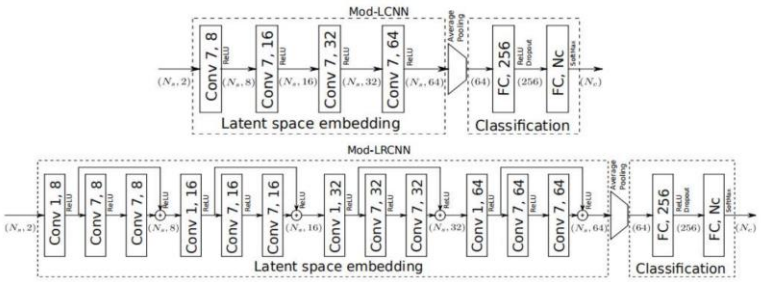


表 3.4.1.1 具体模型架构示例

2. 识别与展示：主要思路是通过多线程实现信号识别的功能。首先，在初始化函数中加载预训练的深度学习模型，并设置相关参数。然后，通过数据获取函数获取信号数据，并进行预处理后放入队列中。接着，在预测结果函数中从队列中获取数据，使用加载的模型进行预测，并输出预测结果。最后，在绘制预测结果函数中，通过绘制信号的波形图展示预测结果。通过启动主线程，同时运行数据获取、预测结果和绘制预测结果的线程，实现了信号识别的功能。

```
此次随机提取的信号类型为： BPSK
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step
此信号最终识别结果为：BPSK
具体各种类信号比例如下：
Class BPSK: 99.65%
Class PSK8: 0.04%
Class QAM16: 0.04%
Class QAM32: 0.05%
Class QAM64: 0.03%
Class QAM8: 0.09%
Class QPSK: 0.11%
```

表 3.4.1.2 识别结果示意图

3.3.2jetson nano 实现 tensorRT 加速

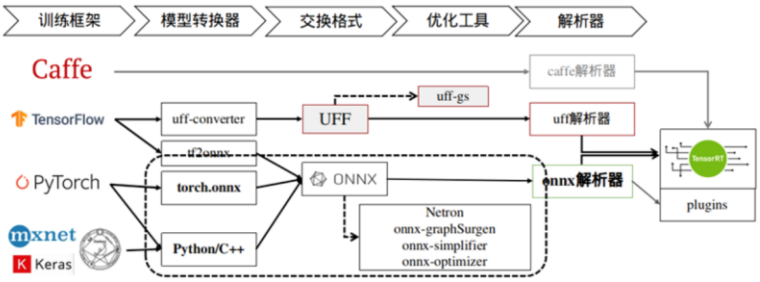
TensorRT 是 Nvdia 推出的一套专为深度学习推理打造的 SDK。在推理阶段，基于 TensorRT 的应用可以提供同比单一 CPU 平台高达 40 倍的加速效果。TensorRT 主要包括一个深度学习推理的优化器(optimizer)和运行时(runtime)，可以为深度学习推理应用提供低延迟和高吞吐的特性。它支持 C++和 Pytorch 两种代码方式。

TensorRT 是构建在 NVIDIA 的并行编程模型 CUDA 基础之上的，结合最新一代 Ampere 架构的 GPU，TensorRT 还可以利用 Tensor Core 实现稀疏性(Sparsity)加速的特点。对于深度学习推理应用的生产环境部署方面，TensorRT 提供了 INT8 和 FP16 的优化，可针对视频流，语音识别，推荐领域，欺诈检测，自然语言处理等。低精度推理能够极大的

降低应用的延迟，有益于实时服务，以及自动驾驶和嵌入式类型的应用。

在模型导出后进行模型转换的步骤：

1. 将通过原生框架(PyTorch, Paddle, TensorFlow 等)预训练模型转成 onnx 格式。
2. 将 onnx 模型转入 TensorRT
3. 应用优化器并生成一个 TensorRT engine
4. 在 GPU 上执行推理过程



四、工程实施计划

4.1 工程实施总体进度安排

时间	进度安排
第一周	选定课题，做相关的市场调研
第二周	基于市场调研初步制定出系统的整体框架，细化预期功能，做出十周的详细计划
第三周	学习使用 hackrf 模块，试着收到信号 模型结构设计 TensorRT 相关结构原理学习
第四周	使用对应数据集进行模型训练 尝试进行模型加速，运行 demo
第五周	无线电信号接收后结构分析，选择协议，进行变换
第六周	接口设计，将无线电信号进行处理并且在 PC 端进行推理
第七周	模型部署到 jetson nano 上进行调试
第八周	完善模型结构，增加拓展功能 使用 tensorRT 进行加速
第九周	完善所有模块功能 完成项目报告

	制作说明手册
第十周	联合调试，展示验收

4.2 工程实施小组成员分工

时间\姓名	闫文杰	张博阳	晋远帆	陈鲲鹏	贺健
第二周	根据安排调研和讨论研究方向	调研基于深度学习的调制识别模型，研读相关论文，学习相关基础知识	参与调研，参与选题。	根据安排调研和讨论研究方向	根据安排调研和讨论研究方向
第三周	学习模型部署 熟悉 tensorRT 流程 .熟悉 cuda 框架的基本操作和命令 配置好环境	准备与环境搭建： 1.安装配置深度学习框架（如 TensorFlow、PyTorch）和相关软件环境 2.确保通用 PC 机具备足够的计算资源和存储空间用于模型训练 3.熟悉深度学习框架的基本操作和命令	确定选题，参与完成开题报告；调研识别方案，配置项目环境复现方案；根据方案完成 demo 程序	调研 HackRF One 和 GNU Radio 平台的具体使用案例	调研 HackRF One 和 GNU Radio 平台的具体使用案例
第四周	尝试进行模型加速，运行 demo	数据收集与预处理： 1.收集（HarkRF）用于训练的调制识别数据集，包括各种调制方式的样本数据 2.进行数据预处理，包括数据清洗、标签处理、数据增强等	完成与无线电模块的对接；完善 demo 程序，增加功能；调研频段选择算法	无线电使用环境搭建；实际使用 HackRF One 和 GNU Radio 平台进行信号的收发测试	无线电使用环境搭建；实际使用 HackRF One 和 GNU Radio 平台进行信号的收发测试
第五周	继续测试，尝试转换不同格式的模型	模型设计与训练; 1.设计深度学习模型的架构和网络结构，选择合适的损失函数和优化器 2.开始模型的训练过程，监控训练过程中的指标并进行调整	确定 demo 程序基本设计，pyqt 编写设计 GUI 界面；尝试频段自动选择算法实现	继续测试接收多种不同调制方式的无线电信号的，并将数据进行发送； 同时研究数据如何发送	继续测试接收多种不同调制方式的无线电信号的，并将数据进行发送； 同时研究数据如何发送
	联调，部署其他同学训练好的模型	模型优化与调整： 1.对训练好的模型进行性能优化和调整，尝试不同的模型结构和	完成 pyqt 基本界面，界面美化，上位机环	继续测试接收多种	继续测试接收多种不同

第六周		参数设置 使用交叉验证等方法评估模型的泛化能力，并调整模型以提高泛化性能 （或者安装 CUDA 加速模型训练，方便测试与提升训练速度）	境下开发基本完成	不同调制方式的无线电信号的，并将数据进行发送； 同时研究数据如何发送	调制方式的无线电信号的，并将数据进行发送； 同时研究数据如何发送
第七周	模型加速与优化，寻找问题并且解决	模型评估与验证： 1.使用验证集对训练好的模型进行评估，计算准确率、召回率等指标 2.对于输入数据进行处理，主要研究信噪比提高的算法或者信号长短的选取，用于提高识别的准确率	调研 Jetson Nano CUDA 环境，将项目打包为 docker 镜像	尝试在 HackRF One 端发送不同调制的信号	尝试在 HackRF One 端发送不同调制的信号
第八周	完善模型结构，增加拓展功能 使用 tensorRT 进行加速	模型部署准备： 1.准备模型部署所需的相关工具和软件环境，包括模型转换工具、部署框架等 2.确保模型的兼容性和可移植性，以便于在 Jetson Nano 上部署和运行	Jetson Nano 镜像部署，测试，若部署失败，通过 minoconda 配置环境	继续尝试在 HackRF One 端发送不同调制的信号	继续尝试在 HackRF One 端发送不同调制的信号
第九周	完善所有模块功能 完成项目报告 制作说明手册	模型部署与测试： 1.将训练好的模型部署到 Jetson Nano 开发板上，并测试模型在开发板上的性能和准确率 2.调整部署过程中可能出现的问题，确保模型能够在开发板上正常运行	项目联调，代码清理优化；	利用 HackRF 进行信号的发送，并接收发送的信号，最后将信号数据进行上传； 联合调试	利用 HackRF 进行信号的发送，并接收发送的信号，最后将信号数据进行上传； 联合调试
第十周	验收展示	性能优化与总结： 1.对模型在 Jetson Nano 上的性能进行优化，提高推理速度和资源利用率 2.总结模型训练和部署过程中的经验教训，记录并分享相关的技	参与完成项目结题报告与答辩	利用 HackRF 进行信号的发送，并接收发送的信号，最后将	利用 HackRF 进行信号的发送，并接收发送的信号，最后将信号数据进行上

		术点和心得		信号数据进行上传； 联合调试	传； 联合调试

参考文献

[1] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in Engineering Applications of Neural Networks, C. Jayne and L. Iliadis, Eds. Cham: Springer International Publishing, 2016, pp. 213–226.

[2] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, April 2019, pp. 370–378.

[3] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," IEEE Wireless Communications Letters, vol. 8, no. 1, pp. 213–216, Feb 2019.

[4] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast Deep Learning for Automatic Modulation Classification," arXiv e-prints, p. arXiv:1901.05850, Jan 2019.

[5] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, "Deep learning for rf signal classification in unknown and dynamic spectrum environments," in 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). IEEE, 2019, pp. 1–10.

[6] K. Tekbıyık, A. Rıza Ekti, A. G¨orc¸in, G. Karabulut Kurt, and C. Kec¸eci,

"Robust and Fast Automatic Modulation Classification with CNN under Multipath Fading Channels," arXiv e-prints, p. arXiv:1911.04970, Nov 2019.