

深圳技术大学本科毕业论文（设计）

中期检查表

学生姓名	李乐雅	学院	大数据与互联网学院
专业	物联网工程	班级	物联网 1 班
指导教师	吕羽	中期检查时间	2024 年 3 月 17 日
论文题目	基于深度学习的胎儿心率分析增强及基线自动确定		

毕业论文（设计）的目录和主要研究内容：			
毕业设计目录：			
第一章绪论			
1.1 研究背景及意义			
1.2 国内外研究现状			
1.3 主要研究内容			
1.4 章节安排			
第二章胎心宫缩信号及深度学习相关基础和理论			
2.1 胎心宫缩监理论			
2.1.1 胎心宫缩信号采集方式			
2.1.2 胎心宫缩信号波形特征			
2.1.3 胎心宫缩信号状态分类			
2.2 深度神经网络模型介绍			
2.2.1 卷积层			
2.2.2 池化层			
2.2.3 全连接层			
2.2.4 编码器与解码器			
2.3 本章小结			
第三章基于深度神经网络的基线确定算法研究			
3.1 研究流程设计			
3.2 实验数据预处理			
3.2.1 数据集介绍			
3.2.2 数据异常值预处理			
3.2.3 数据扩充算法			
3.3 算法模型设计			
3.4 模型训练方案			
3.5 本章小结			
第四章实验与分析			
4.1 性能评估指标			
4.2 与其他方法对比			
4.2.1 其他方法介绍			

4.2.2	实验设置
4.2.3	实验结果对比分析
4.3	消融实验
4.3.1	实验设计
4.3.2	结果分析
4.4	本章小结
第五章总结与展望	
5.1	总结
5.2	展望
参考文献	

### 主要研究内容：

使用深度学习方法对胎儿缺氧进行评估，并通过分析基线和减速类型来确定胎儿的健康状况。该研究提出了一种改进的 U-Net 模型，用于将胎心率（FHR）信号分割为三个状态。在进行分割后，通过滤波计算完整的基线，考虑到 FHR 信号的形态和长期与短期的相关性。研究还引入了多尺度特征提取模块和长短期滤波，以满足 FHR 信号形态分析的临床需求。

### 毕业论文（设计）工作进展情况（详述）：

#### 当前进展：

##### 1. 数据预处理与数据增强

```
def segment(data, signal_len, l, n, s, num):
    r = signal_len % s
    k = int((signal_len - r) / s)
    head = 0

    for i in range(0, k - n + 1):
        #print(head + l, head)
        s_block = data.iloc[head:head+l, :]
        d = pd.DataFrame(s_block)
        #filepath = './DB2/dataset/train/train' + num + "_" + str(head) + '.csv'
        filepath = './DB2/dataset/train/train' + num + "_" + str(head) + "-1_.csv"
        d.to_csv(filepath, index=False, sep=',')
        head = head + s
    head = signal_len - l
    d = pd.DataFrame(data.iloc[head:head + l, :])
    filepath = './DB2/dataset/train/train' + num + "_" + str(head) + "-1_.csv"
    d.to_csv(filepath, index=False, sep=',')
```

```
信号放大
for i in range(1, 44):
    # 读取CSV文件
    num = "%02d" % i
    filepath = './original_data/train' + num + ".csv"
    df = pd.read_csv(filepath)
    signal_len = len(df)
    # 放大
    df.loc[df['label'] == 1, 'FHR'] += 1
    df.loc[df['label'] == 2, 'FHR'] -= 1

    segment(df, signal_len, l, n, s, num)
```

```

# 信号缩小
for i in range(1,44):
    # 读取CSV文件
    num = "%02d" % i
    filepath = './original_data/train' + num + ".csv"
    df = pd.read_csv(filepath)
    signal_len = len(df)
    # 放大
    df.loc[df['label'] == 1, 'FHR'] -= 1
    df.loc[df['label'] == 2, 'FHR'] += 1

    #segment(df, signal_len, l, n, s, num)
    reversed_df = df.iloc[::-1]
    segment(reversed_df, signal_len, l, n, s, num)

```

## 2. 设计网络模型

```

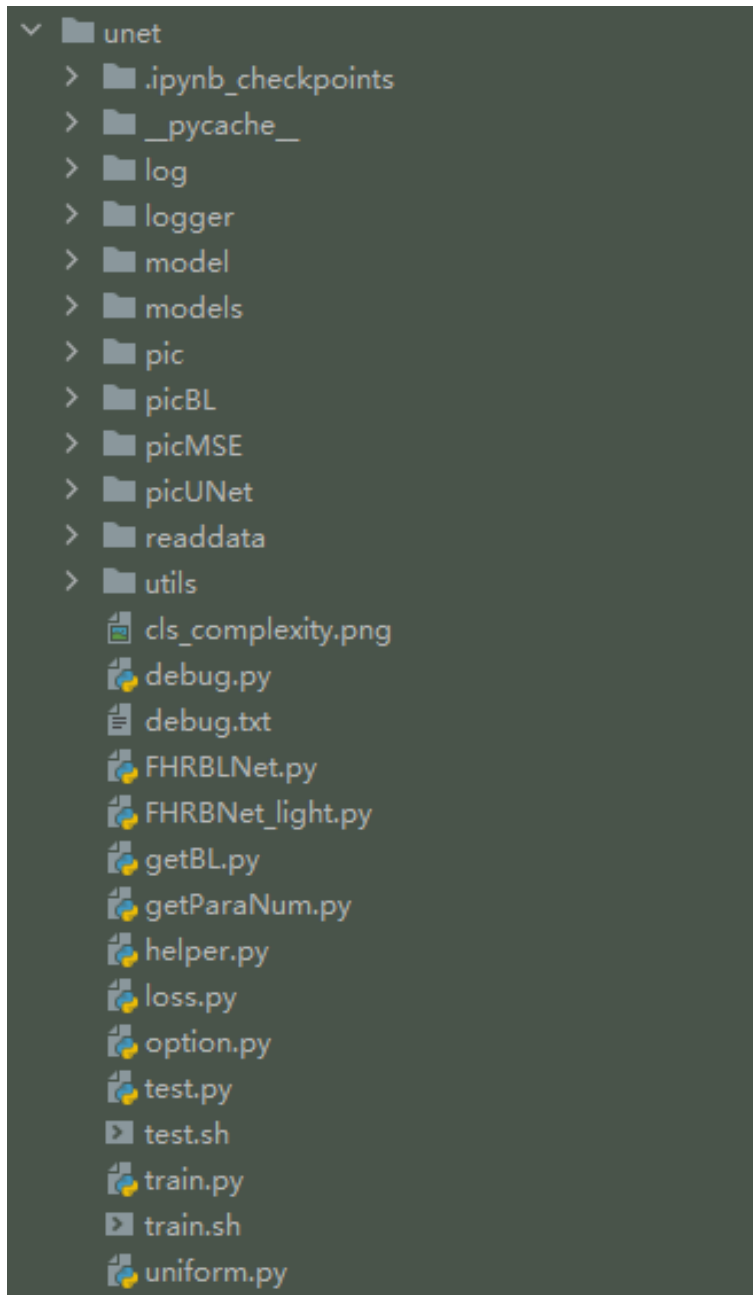
class MseUNet(nn.Module):
    def __init__(self, ch=64):
        super(MseUNet, self).__init__()
        self.c1 = Conv_Block(1, ch)
        self.s1 = MultiScaleEncoder(21, 31, 81, ch, ch) # instead of skip connection
        self.d1 = DownSample(ch)
        self.c2 = Conv_Block(ch, ch*2)
        self.s2 = MultiScaleEncoder(21, 31, 81, ch*2, ch*2)
        self.d2 = DownSample(ch*2)
        self.c3 = Conv_Block(ch*2, ch*4)
        self.s3 = MultiScaleEncoder(21, 31, 81, ch*4, ch*4)
        self.d3 = DownSample(ch*4)
        self.c4 = Conv_Block(ch*4, ch*8)
        self.s4 = MultiScaleEncoder(21, 31, 81, ch*8, ch*8)
        self.d4 = DownSample(ch*8)
        self.c5 = Conv_Block(ch*8, ch*16)
        self.u1 = UpSample(ch*16)
        self.c6 = Conv_Block(ch*16, ch*8)
        self.u2 = UpSample(ch*8)
        self.c7 = Conv_Block(ch*8, ch*4)
        self.u3 = UpSample(ch*4)
        self.c8 = Conv_Block(ch*4, ch*2)
        self.u4 = UpSample(ch*2)
        self.c9 = Conv_Block(ch*2, ch)
        self.out = nn.Conv1d(ch, 3, 1, 1)
        # self.out = nn.Sigmoid()
        # self.Th=nn.Sigmoid()

    def forward(self, x):
        R1 = self.c1(x)
        R2 = self.c2(self.d1(R1))
        R3 = self.c3(self.d2(R2))
        R4 = self.c4(self.d3(R3))
        R5 = self.c5(self.d4(R4))

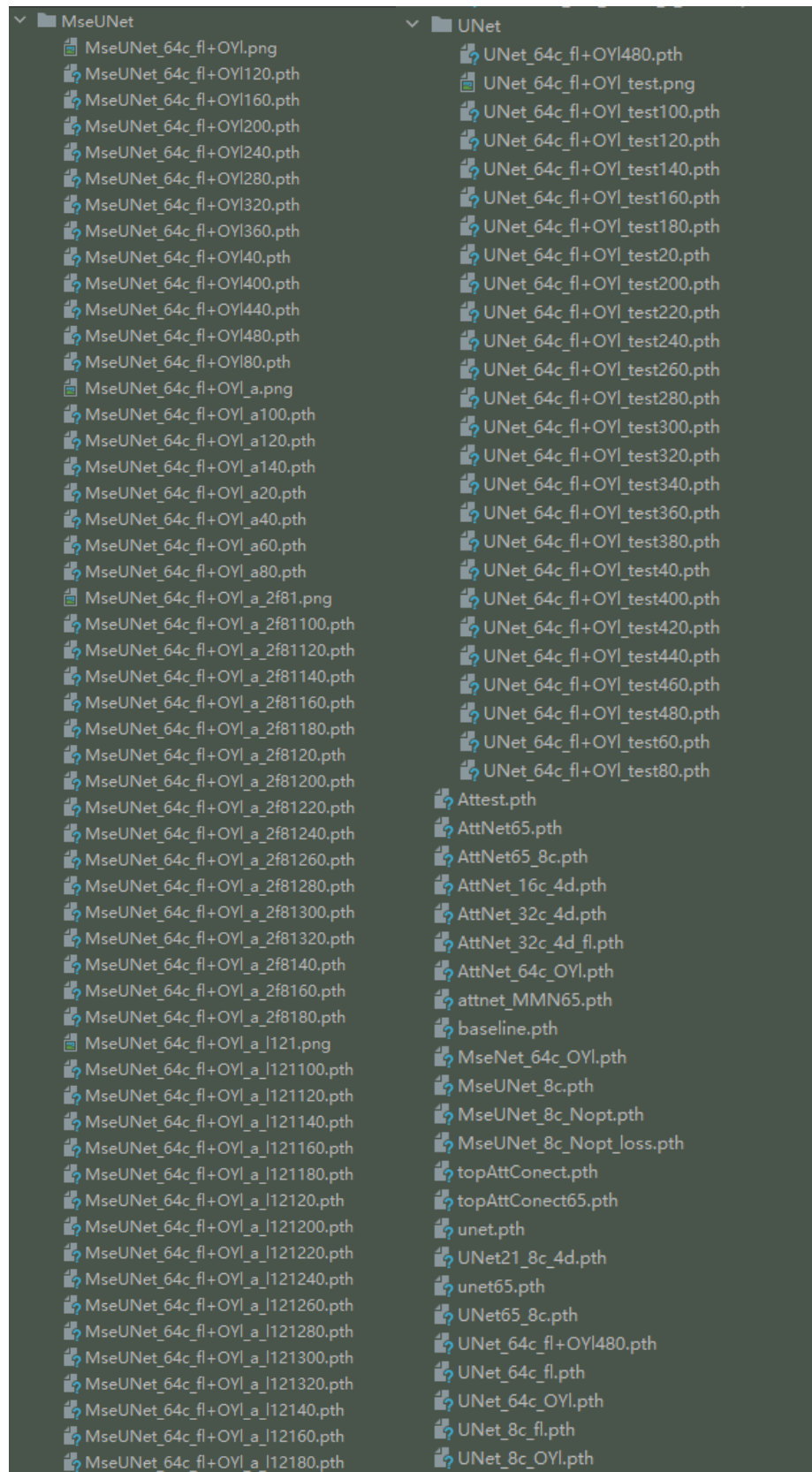
        S4 = self.s4(R4)
        S3 = self.s3(R3)
        S2 = self.s2(R2)
        S1 = self.s1(R1)
        # c6 c7 c8 c9 通道数等于对应R S的通道数和，init更改
        O1 = self.c6(self.u1(R5, S4))
        O2 = self.c7(self.u2(O1, S3))
        O3 = self.c8(self.u3(O2, S2))
        O4 = self.c9(self.u4(O3, S1))
        # return self.Th(self.out(O4))
        return self.out(O4)

```

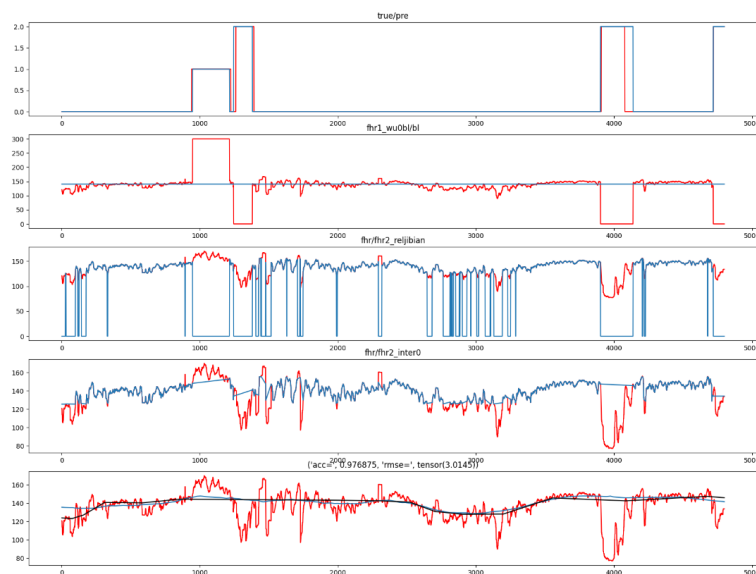
### 3. 编写代码，优化代码项目框架



#### 4. 调整模型参数，训练模型



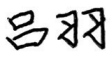

## 5. 模型效果可视化



实验在 Intel Xeon 4210 CPU, 128G 内存以及单片 NVIDIA RTX 4090 GPU 的 Linux 平台上进行，将公共 FHR 数据库 CLUF-DB 中 66 条有标签的 FHR 数据按 7：3 分成训练集和验证集训练模型。为了评估我们的自动确定方法和专家分析之间的一致性，我们使用了 RSMD 和加速或减速的 F-measure 来量化计算值与基线真实值之间的差异。目前模型的最优计算效果的平均 RSMD 低至 4.49dpm。

### 后续计划：

未来的工作包括在以前分类模型的卷积模块的基础上，开发一个多元时序模型和一个用于胎心宫缩信号质量评估的附加模块。重要的是，将两个网络整合到一个多任务框架中，允许任务指导参数共享的范围。这种方法寻求优化共享表示和特定任务表示的组合，从而增强 FHR 信号表示。

指导教师填写栏目（在正确项后方框内划√）
1、毕业论文（设计）进展情况： (1)提前完成□；      (2) 正常进行 <input checked="" type="checkbox"/> ；      (3) 延期滞后□
2、学生对毕业论文（设计）的认真程度： (1) 认真 <input checked="" type="checkbox"/> ；      (2) 较认真□；      (3) 不认真□
3、查阅文献资料的能力： (1) 强 <input checked="" type="checkbox"/> ；      (2) 一般□；      (3) 差□
4、已完成的毕业论文（设计）中期质量评价： (1) 好 <input checked="" type="checkbox"/> ；      (2) 中□；      (3) 差□
5、毕业论文（设计）方向有无更改： (1) 有□；      (2) 无 <input checked="" type="checkbox"/>
6. 对能否按期完成毕业论文（设计）的评估： (1) 能 <input checked="" type="checkbox"/> ；      (2) 否□
<p>存在的问题及后期指导工作意见：</p> <p>根据任务书中的进度安排，李乐雅同学按时完成了相应任务，其中算法部分已基本完成，并投稿录用了一篇论文。接下来应继续完善算法，进行不同方法的性能比较，并按时提交毕业论文。</p> <p style="text-align: right;">指导教师签名：  2024年 3 月 17 日</p>
<p>审查小组检查意见：</p> <p>李乐雅同学按时完成了相应任务，其中算法部分已基本完成，接下来应继续完善算法，并按时提交毕业论文。建议通过中期审查。</p> <p style="text-align: right;">审查小组负责人签名：  2024年 3 月 17 日</p>