智能管道漏气检测系统

万宇杰；孙晨；朱培志

第一部分　设计概述

1.1 设计目的

本智能管道漏气检测系统的设计目的是通过采用机器学习算法和嵌入式系统技术，实现对管道漏气的自动检测和警报。本系统旨在利用高灵敏度麦克风传感器实时采集管道漏气时产生的声音信号，并通过嵌入式计算模块进行实时处理和分析，准确判断漏气位置和漏气速率，并通过摄像头将管道漏气位置可视化。通过报警模块实时警报和无线通信模块发送报警信息，实现对漏气事件的实时监控和管理。该设计旨在提高漏气检测的准确性、实时性和灵敏度，为管道安全运行提供可靠保障。同时，嵌入式系统结构和低功耗设计使该系统适用于多个应用场景，具有一定的市场潜力。

1.2 应用领域

本智能管道漏气检测项目应用领域广泛。主要应用领域包括石油和天然气工业、市政燃气、工业生产和农业等领域。在石油和天然气工业中，该项目可以及时检测管道漏气，避免安全事故和经济损失。在市政燃气领域，该项目能够确保城市燃气供应的安全性和稳定性。在工业生产中，该项目可应用于各种管道系统，确保生产过程的安全运行。在农业领域，该项目能够及时检测燃气或气体泄漏，确保农业生产环境的安全。该项目的应用领域广泛，能够为不同行业和领域提供管道漏气检测的智能可视化解决方案，提高安全性和效率。

1.3 主要技术特点

（1）声音信号采集与分析：采用高灵敏度麦克风传感器实时采集管道漏气时产生的声音信号。提取声音信号的 MFCC 系数，其具有模拟人类听觉特性、压缩数据维度、去除不相关信息、鲁棒性强等优点可以高效准确判断漏气情况。这种非接触式的检测方法具有高灵敏度和高准确性的优势。

（2）机器学习算法应用：项目部署了基于机器学习的算法卷积神经网络，对传感器数据进行实时处理和分析。卷积神经网络能够自动提取数据特征，检测漏气情况，提高漏气检测的精度和鲁棒性。

（3）嵌入式系统技术：采用嵌入式计算模块作为运算单元高性能 STM32H723 板卡。嵌入式系统结构具有较强的计算能力和实时性，适合进行实时数据处理和分析，其低功耗设计也使系统能够长时间稳定运行。同时，MFCC 压缩数据维度从而减少内存占用并提高运算速率。

（4）实时监测与报警：通过实时监测管道漏气，能够及时发现漏气情况并通过蓝牙技术远程报警，报警模块通过声音报警和 LED 灯方式进行实时警报，提醒相关人员处理漏气事件。实时监测和报警功能提高了漏气事件的处理效率和安全性。

（5）物联网远程监控功能：本设计可视化展示功能，将漏气位置数据通过蓝牙传输到手机 APP 和监控系统，手机 APP 发出警报声，监控系统实时显示漏气位置。这种可视化展示帮助运维人员直观地了解漏气位置的具体位置，加快问题定位和修复的速度，提高工作效率。

1.4 关键性能指标

（1）系统的实时性能：统从漏气事件发生到发出报警的时间差。

（2）系统稳定性能：系统能否长时间稳定运行，保持持续的漏气检测和报警功能。

（3）漏气判断准确性，系统判断漏气位置的误报率和漏报率。

（4）系统功耗：系统运行所需的能量消耗。

（5）抗干扰能力：系统对于复杂环境中的干扰源（噪声、震动等）的抵抗能力以及漏气声音提取和分析效率。

（6）可视化效果：系统提供的漏气位置可视化效果的质量和准确性。

1.5 主要创新点

（1）集成 MFCC 与卷积神经网络：项目引入了 MFCC 声音预处理和卷积神经网络（CNN）机器学习算法，用于对传感器采集到的声音信号进行分析和处理。这一创新点使系统能够自动学习和提取声音信号中的特征，相较于传统方法，具备更高的精度和鲁棒性。

（2）嵌入式系统结构的应用：项目采用高性能的嵌入式计算模块 STM32H723 板卡作为运算单元，具备强大的计算能力和实时性，适应快速的漏气检测需求。同时，低功耗设计使得系统能够长时间稳定运行，适用于多种应用场景。

（3）高灵敏度传感器的应用：项目采用高灵敏度麦克风作为传感器，实时采集管道漏气时产生的声音信号。这种高灵敏度传感器具备出色的信号捕捉能力，提供准确的漏气检测数据，增强了系统的灵敏度和可靠性。

（4）实时监测与报警能力：项目具备实时监测漏气情况并进行报警的能力。通过持续监测管道漏气并及时发出报警，系统能够快速响应漏气事件，降低事故风险和经济损失，增强了系统的安全性和应急响应能力。

（5）物联网远程监控功能：项目引入了将漏气位置实时显示在管道画面上的可视化展示功能。这一创新点使运维人员能够直观地了解漏气位置的具体位置，快速定位和修复漏点，提高工作效率。这种可视化展示功能对于地面管道漏气检查工程具有重要帮助。

第二部分　系统组成及功能说明

2.1 整体介绍

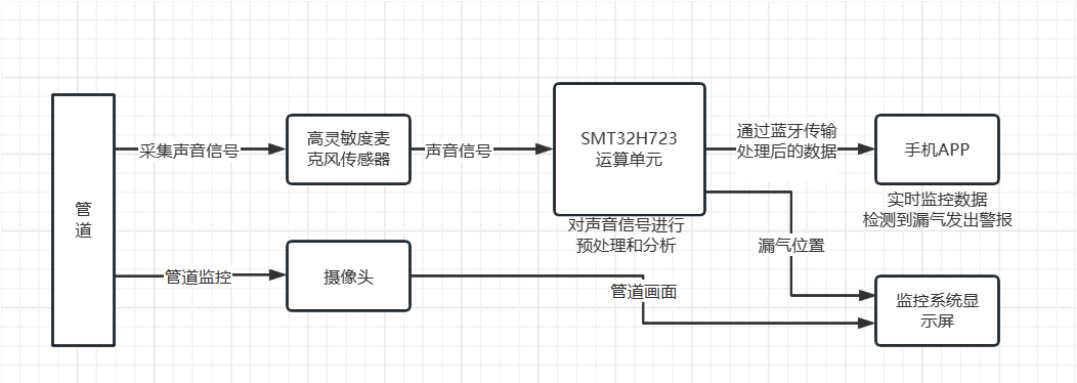

图 1 系统框图

整体说明：

高灵敏度麦克风传感器实时采集管道漏气声音数据，将声音数据传给 STM32H723 进行 MFCC 声音信号预处理和卷积神经网络学习，STM32H723 将处理好的数据通过蓝牙传输给手机 APP，手机 APP 检测到漏气立即发出警报。同时监控系统将漏气位置实时显示在显示屏上。

2.2 各模块介绍

高灵敏度麦克风传感器模块：采用两个麦克风传感器模块，每个模块板载 7 个高灵敏度硅麦克风。两个麦克风传感器模块模块便于定位漏气位置。

STM32H723：通过对声音数据进行 MFCC 实时处理和卷积神经网络学习并分析，准确地判断漏气情况，并生成相应的报警信号和数据。

监控系统、手机 APP：手机 APP 作为系统的用户界面，通过蓝牙技术实时接收来自 STM32H723 的漏气报警数据，并进行相应的警报并提供漏气处理的用户界面交互，漏气位置被框选显示在监控系统屏幕上。

第三部分　完成情况及性能参数

（1）系统的实时性能检测，共实验 10 次，每次记录 10 组漏气发生到报警时间差 t 的数据并取平均值

实时性能	1	2	3	4	5	6	7	8	9	10
t	2.67s	2.10s	2.31s	2.58s	2.89s	2.92s	2.28s	2.94s	2.24s	2.12s

（2）系统的判断准确性检测，共实验 10 次，每次记录 10 组漏气发生时报警次数和未报警次数和 10 组漏气未发生时报警次数和未报警次数。

误报率=未漏气时报警次数/10

漏报率=漏气发生时未报警次数/10

准确性	1	2	3	4	5	6	7	8	9	10
误报率	0%	0%	0%	0%	10%	0%	0%	0%	0%	0%
漏报率	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

（3）抗干扰能力检测：共实验 30 次，每次记录在噪声干扰源环境下手机 APP 接受到的准确漏气声音数据的抗干扰的准确率（准确数据次数/30）

抗干扰能力	1	2	3	4	5	6	7	8	9	10
准确率	93%	100%	100%	93%	90%	100%	97%	100%	93%	97%

第四部分　总结

4.1 可扩展之处

（1）连接 5G 网络，提供高速实时异地监控：通过将系统连接到 5G 网络，可以实现高速、稳定的数据传输和远程监控功能，监测数据可以实时传输到远程服务器或云端，运维人员实时监控和管理。

（2）麦克风阵列确定位置，便于扩展多管道并排、多段长管道漏气检测：通过使用麦克风阵列，可以进一步提高漏气位置的准确性和精度。麦克风阵列可以利用声音波在阵列中传播的时间差来确定漏气的具体位置，这对于多管道、多段漏气检测情况下的定位非常有用。麦克风阵列的设计也为后续的系统扩展提供了便利。

（3）实时采集数据集上传服务器自学习优化模型：系统可以实时采集到的漏气数据集可以上传到服务器进行存储和分析。利用这些数据，可以进行机器学习和深度学习算法的自学习优化，以进一步提升系统的准确性和鲁棒性。通过对大量实时数据的分析和学习，可以不断优化漏气检测模型，使其适应不同管道和漏气情况，提高系统的性能和稳定性。

（4）智能预警：通过增加智能预警和远程控制功能，系统训练的声音模型可以检测管道环境状态，异常时自动触发预警机制，并通过远程控制系统执行相应的应急措施。

4.2 心得体会

参与本次嵌入式芯片与系统设计竞赛过程中虽然面临了许多挑战，但也获得了许多宝贵的经验和收获。

通过本次竞赛，我们深入了解了嵌入式系统的设计与开发流程。我们从需求分析开始，逐步进行硬件选型、软件开发和系统集成，最终实现一个完整的嵌入式系统。这个过程让我们对嵌入式系统的各个方面有了更深入的了解，并提高了

我们的技术能力。同时通过有效的沟通和团队协作，我们解决了遇到的各种问题，并最终成功地完成了项目。这个经历增强了我们的合作意识和协调能力，为我们未来的职业生涯打下了坚实的基础。

本次竞赛也培养了我们的问题解决和创新能力。在项目中，我们面对各种技术难题和挑战，需要不断地学习新的知识、寻找解决方案和创新思路。通过积极的探索和尝试，我们学会了解决问题的方法，并提出了一些创新的解决方案。这个过程激发了我们的创造力和创新精神，为我们未来的学习和工作打下了良好的基础。

参与本次嵌入式芯片与系统设计竞赛是一次非常宝贵的经历。我们不仅学到了专业知识和技能，还培养了团队合作、问题解决和创新能力。这个经历将对我们的学习和未来的职业发展产生积极的影响，我们将以此为契机，不断提升自己，在嵌入式芯片领域取得更大的成就。

第五部分　参考文献

[1]崔琳,崔晨露,刘政伟等.改进 MFCC 和并行混合模型的语音情感识别[J].计算机科学,2023,50(S1):166-172.

[2]崔佳嘉,马宏忠.基于改进 MFCC 和 3D-CNN 的变压器铁心松动故障声纹识别模型[J].电机与控制学报,2022,26(12):150-160.DOI:10.15938/j.emc.2022.12.015.

[3]韩晓良,陈佳昌,周伟松.基于 3D 注意力的 MobileNet 图像分类算法改进[J].重庆邮电大学学报(自然科学版),2023,35(03):513-519.

第六部分　附录

一、漏气声音信号的梅尔倒谱系数（MFCC）求解过程

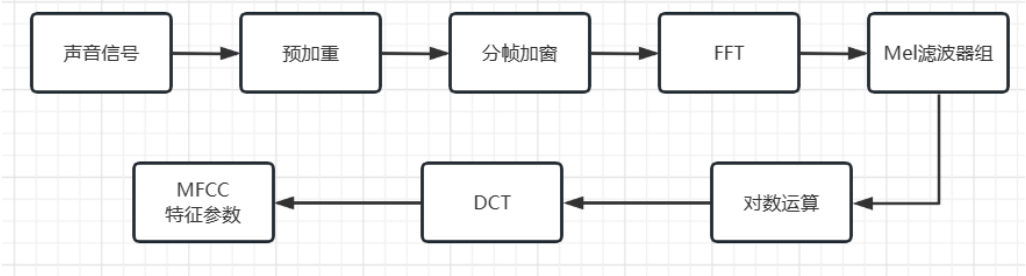

图 2 求 MFCC 的步骤

1. 分帧：将信号分成短时帧。

2. 声音信号归一化：对信号应用预加重滤波器以放大高频。

$$y(t) = x(t) / x(t)_{frame\_max}$$

3. 加窗：对每个帧乘以一个窗函数（汉宁窗）。

$$\mathbf{w}(t)=\frac{1}{2}(1-\cos\frac{2\pi t}{T})$$

4. FFT：做短时傅立叶变换。

$$S_i(k)=\sum_{n=1}^{N}s_i(n)e^{-j2\pi kn/N}1\le k\le K$$

5. 计算功率谱。

$$P=\frac{|FFT(x_i)|^2}{N}$$

6. 计算 Mel 滤波器：提取频带。

$$f_{mel}=2595*\log_{10}(1+\frac{f}{700})$$

7. MFCCs:应用离散余弦变换（DCT）对滤波器组系数去相关处理，产生滤波器组的压缩表示。

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

二、卷积神经网络

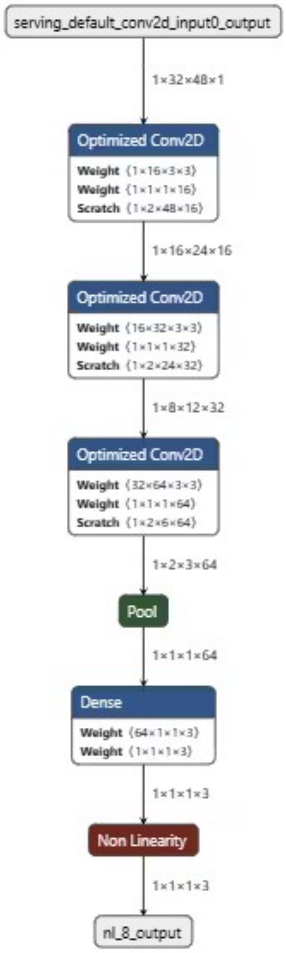

图 3 CNN 模型结构

三、声源定位算法

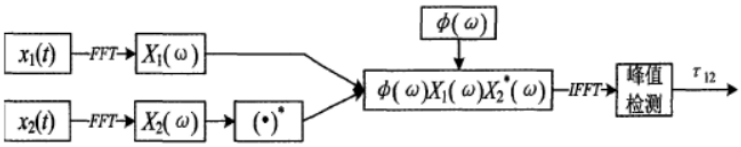

图 4 基于 GCC-PHAT 的时延估计

图 5 管道模型

假设两麦克风之间距离为 L 已知，即 a+b=L。漏气声到达 Mic1 和 Mic2 的距离分别为 a 和 b，通过 GCC-PHAT 算法得出声音到达两个麦克风的时间差τ12，即

a-b=340*τ12，即可解出漏气具体位置。

重要代码：

```
  my_model = Sequential([
                            layers.Conv2D(16, (3, 3), strides=(1, 1), padding='same', use_bias=False),
                            layers.BatchNormalization(),
                            layers.Activation('relu'),
                            layers.MaxPooling2D(),#1
                            layers.Conv2D(32, (3, 3), strides=(1, 1), padding='same', use_bias=False),
                            layers.BatchNormalization(),
                            layers.Activation('relu'),
                            layers.MaxPooling2D(),#2
                            layers.Conv2D(64, (3, 3), strides=(2, 2), padding='same', use_bias=False),
                            layers.BatchNormalization(),
                            layers.Activation('relu'),
                            layers.MaxPooling2D(),#3
                            tf.keras.layers.GlobalAveragePooling2D(),
                            tf.keras.layers.Dense(3, activation="softmax")])
```

```
  //32*48
  void cal_mfcc(arm_mfcc_instance_f32 *S , float32_t *audio_input , float32_t *mfcc_output)
  {
    float32_t fft_temp[FFT_LENGTH+2]={0};
    float32_t mfcc_input_temp[512]={0};
    float32_t mfcc_output_temp[32]={0};
    uint32_t i,j;
    for(i=0;i<48;i++)
    {
      for(j=0;j<FFT_LENGTH;j++)
      {
        if(j<400)
         mfcc_input_temp[j]=audio_input[i*160+j];
        else
         mfcc_input_temp[j]=0;
//       printf("i=%u,j=%u,input:%f\r\n",i,j,mfcc_input_temp[j]);
```

```
      }

      arm_mfcc_f32(S,mfcc_input_temp,mfcc_output_temp,fft_temp);
      for(j=0;j<32;j++)
      {
        mfcc_output[j*48+i]=mfcc_output_temp[j];
//      printf("%f\r\n",mfcc_output_temp[j]);
      }
//    while(!KEY_Scan(0));
    }
  }
```

```
  void cal_tdoa(TDOA_Structure *TDOA_Str,uint8_t printf_option)
  {
    uint16_t i=0;
    uint32_t max_idx_rxy=0;
    float32_t max_val_rxy=0;
    arm_cfft_instance_f32 * S;

    arm_cfft_init_f32(S,FFT_LENTH);
    arm_cfft_f32(S,TDOA_Str->sound_data_x,0,1);//由于输入是顺序的,经过蝶形运算得到输出为错位的数,使能位翻转功能可将输出改为顺序
    arm_cfft_f32(S,TDOA_Str->sound_data_y,0,1);//输出没有利用对称性砍掉一半

    dot_prod_A_dot_conjB(TDOA_Str->sound_data_x,TDOA_Str->sound_data_y,FFT_LENTH,real_temp,imag_temp); //X(w) * conj(Y(w))  (a+bi)(c+di)=(ac-bd)+(ad+bc)i

    for(i=0;i<FFT_LENTH;i++)
    {
      TDOA_Str->rxy[i*2]=SCALE_rxy * real_temp[i] / sqrt(real_temp[i]*real_temp[i] + imag_temp[i]*imag_temp[i]);
      TDOA_Str->rxy[i*2+1]=SCALE_rxy * imag_temp[i] / sqrt(real_temp[i]*real_temp[i] + imag_temp[i]*imag_temp[i]);
    }//rxy = 1/phi * Rxy

    arm_cfft_f32(S,TDOA_Str->rxy,1,1);//ifft

    get_max_rxy(TDOA_Str,&max_val_rxy,&max_idx_rxy);//反傅里叶变换数组构成也是  实部 虚部 实部 虚部...

    if(max_idx_rxy>FFT_LENTH/2)
     TDOA_Str->t_differ=(float)(FFT_LENTH - max_idx_rxy - 1)/(float)Fs;//s
    else
```

```
27.      TDOA_Str->t_differ= - (float)max_idx_rxy/(float)Fs;//s
28.
29.     if(printf_option==1)
30.     {
31.   //    if(TDOA_Str->t_differ*34>50 || TDOA_Str->t_differ*34<-50)return;
32.      USART_printf(&huart3,"t_differ=%fms,dis=%fcm\r\n",TDOA_Str->t_differ*1000,TDOA_Str->
    t_differ*34000);
33.     }
34.
35.   }
```