浙江大學
ZHEJIANG UNIVERSITY

信息、控制与计算

实验报告

实验名称 远程声控系统

姓　名 卢泽熙

学　号 3220102478

实验日期 January 10, 2025

指导老师 刘雷

1 远程声控制系统概述

远程声控制系统实现了以下功能：首先采集不同的语音指示信号，进行适当压缩；然后通过噪声信道实现远程传输，远端接收后再通过适当计算识别出是何指示，最后送入一个能控/能观的水下机器人一维运动控制系统，完成不同的速度控制。该系统可分为三个模块：

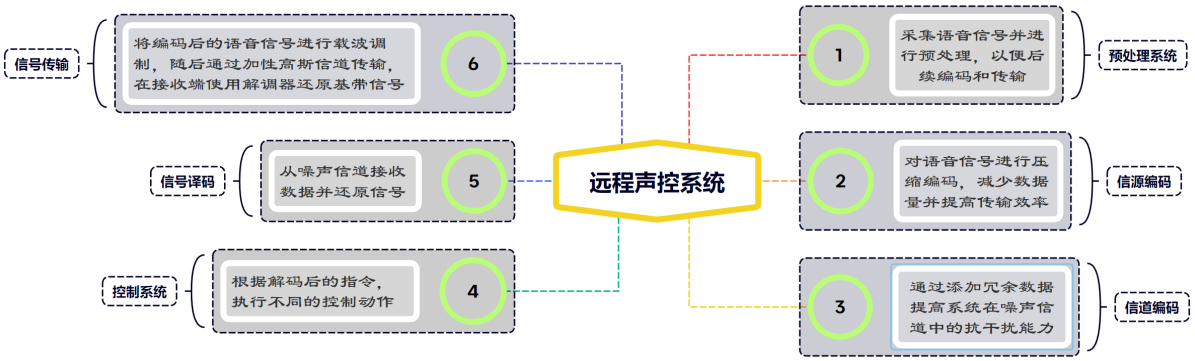
(1). 语音信号采集模块；

(2). 语音信号传输模块；

(3). 语音信号识别模块；

(4). 语音信号控制模块。

以下是该远程声控制系统的结构框图：



1.1 语音信号采集模块

语音采集模块主要完成语音信号的采集和量化操作。使用者输入的语音是模拟信号，必须经数字化处理后才能在通信系统中传输。本实验的语音信号共有三条：**加速**、**刹车**、**停止**。

1.2 语音信号传输、识别模块

远程声控系统的主体部分，完成信号的传输、识别。首先，在发送端，为了使信源减少冗余、有效传输，需要对数据进行压缩（**采用霍夫曼编码**）；与此同时，为了提高系统的抗干扰能力和纠错能力，可以增加校验码冗余（**采用重复编码**）；通过调制，我们可以提高信号的抗噪能力，从而通过 AWGN 信道传输；在接收端收到的信号，进行解调、信道解码、信源解码操作，将其还原为原始的语音信号，并送入识别系统识别控制指令（**采用 CNN 模型**）。

1.3 语音信号控制模块

控制系统的被控对象是一个机器人，传入的不同语音信号可以控制机器人的速度：

- **加速**：使机器人下潜速度逐渐增加；
- **刹车**：使机器人下潜速度逐渐降低；
- **停止**：使机器人在极短时间内下潜速度变为零。

2 语音信号采集系统

2.1 语音信号生成与采样

2.1.1 系统原理

在将模拟信号转换为数字信号时，需要先对其进行采样，即按照一定时间间隔采样获得时间上离散的信号。人耳可听的频率范围最高为 20kHz，为了验证后续信号的重建效果，根据奈奎斯特采样定理，可以使用 44.1kHz 的采样频率进行采样，从而将输入的语音信号在时间上离散化。

2.1.2 代码展示

```
Fs = 44100;       % Sampling frequency (Hz)
nBits = 16;       % Bit depth
nChannels = 1;    % Number of channels (1 for mono)
recDuration = 1; % Recording duration (seconds)

% Create an audiorecorder object
recObj = audiorecorder(Fs, nBits, nChannels);
% Get the recorded audio data
audio_signal = getaudiodata(recObj);
% Save the recorded audio as a WAV file
audiowrite(filename, audio_signal, Fs);
```

代码使用了 audiorecorder 记录音频信号，audiowrite 保存音频信号；**录制音频为“加速”**。

2.2 语音信号量化

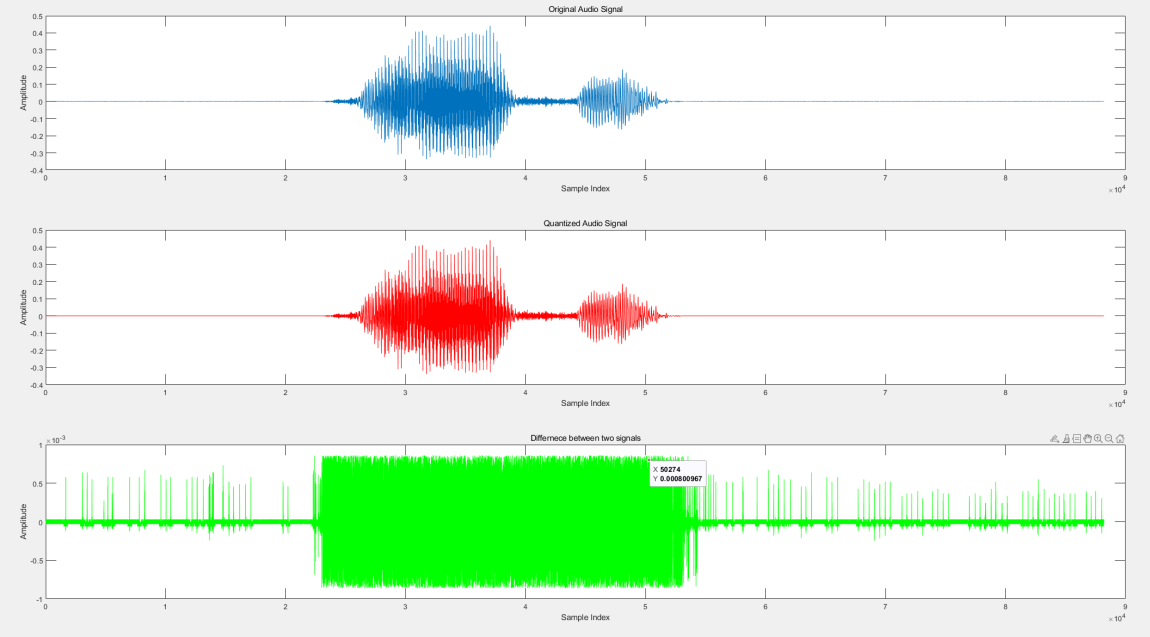
2.2.1 系统原理

离散化后的信号进一步进行均匀量化操作，才能进行后续编码。均匀量化器会将信号幅值范围划分为若干个等宽区间，每个区间对应一个量化等级。信号幅值落入某个区间时，取该区间的中心值作为量化后的幅值，从而完成幅值离散化。在本系统中，使用 8 位均匀量化，将信号幅值映射为 256 个离散值，既保留了语音的主要特征，又减少了存储和处理的复杂性。

2.2.2 代码展示

```
% Find the maximum absolute value of the audio signal.
MAX = max(abs(audio_signal));
MIN = min(abs(audio_signal));
% Calculate the quantization step size.
step_size = (MAX - MIN) / (2^bits);
% Quantize the signal.
quantized_signal = round(audio_signal / step_size) * step_size;
```

2.2.3 处理结果



可以看到量化后的信号还是比较准确的，相比原信号差异在 10^{-3} 量级。

3 语音信号传输系统

3.1 信源编码

3.1.1 系统原理

考虑到系统的准确性和有效性需求，我们将对信源消息进行无损不等长编码，这里采用二元 Huffman 编码，以在准确传输信号的同时获得最短的平均码长，更好地减少冗余。在对信源符号进行 Huffman 编码时，首先根据每个符号的概率大小进行排序，将概率最小的两个符号排在最后两位，标以 0 和 1；重复过程，最后得到仅有两个符号的辅助源，其中一个标以 0，另一个标以 1。从反方向开始进行码字分配，最终得到各个符号对应的码字。

3.1.2 代码展示

```
% Calculate the probability of each unique value in the signal
[unique_values, ~, idx] = unique(quantized_signal);
freq = accumarray(idx, 1);
probabilities = freq / numel(quantized_signal);
% Create a Huffman dictionary using the unique values and their probabilities
dict = huffmandict(unique_values, probabilities);
% Encode the signal using the generated dictionary
encoded_signal = huffmanenco(quantized_signal, dict);
```

为了方便仿真流程，代码使用了 MATLAB 内置函数 unique、huffmandict、huffmanenco，其中 huffmanenco 可以根据符号与概率分布直接完成 huffman 编码。

3.1.3 处理结果

该离散无记忆信源 U 产生的输出序列长度为 $L = 88200$，信源字符表由 256 个符号组成。统计各符号出现概率与码长，可以得到信源熵与码率：

$$H(U) = -\sum_{i=1}^{256} p_i \log(p_i) = 3.2383bit, \quad R = 3.344bit$$

因此可以计算得到编码效率

$$\eta = \frac{H(U)}{R} = 96.9\%$$

以下是 MATLAB 部分输出结果：

```
Symbol: 0.39 -> Code: 1  0  0  1  1  1  1  0  0  1  1  0  1  0  0
Symbol: 0.39 -> Code: 1  0  0  1  1  1  1  0  1  1  1  0  1  1
Symbol: 0.40 -> Code: 1  0  0  1  1  1  1  0  1  1  1  0  1  0
Symbol: 0.40 -> Code: 1  1  1  1  0  1  0  1  0  0  1  0  0  1
Symbol: 0.40 -> Code: 1  0  0  1  1  1  1  0  0  1  1  0  1  1  1
Symbol: 0.40 -> Code: 1  1  1  1  0  1  0  1  0  0  1  0  0  0
Symbol: 0.40 -> Code: 1  0  0  1  1  0  0  0  1  1  1  0  0  1
Symbol: 0.41 -> Code: 1  0  0  1  1  1  1  0  0  0  1  1  0  0  0  1
Symbol: 0.41 -> Code: 1  0  0  1  1  1  1  0  0  1  1  0  1  1  0
Symbol: 0.41 -> Code: 1  0  0  1  1  0  0  0  1  1  1  0  0  0  1
Symbol: 0.41 -> Code: 1  0  0  1  1  1  1  0  0  0  1  1  0  0  0  0
Symbol: 0.42 -> Code: 1  0  0  1  1  1  1  0  0  0  1  1  0  0  1  1
Symbol: 0.42 -> Code: 1  0  0  1  1  1  1  0  0  0  1  1  0  0  1  0
Symbol: 0.43 -> Code: 1  0  0  1  1  0  0  0  1  1  1  0  0  0  0  0  1
Symbol: 0.44 -> Code: 1  0  0  1  1  0  0  0  1  1  1  0  0  0  0  0  0
Symbol: 0.44 -> Code: 1  0  0  1  1  0  0  0  1  1  1  0  0  0  0  1
```

3.2 信道编码

3.2.1 系统原理

在通信系统中，由于信道中噪声的存在，信号从发送端传输到接收端时，往往会出现偏差甚至错误。为了解决这一问题，在信源编码之后，我们需要进行信道编码，即在码字序列中添加冗余信息，以提高系统的抗噪声能力，从而实现更可靠的信息传输。

本系统采用了一种简单的信道编码方式——**重复码**，其原理是将每个比特数据重复 n 次发送，从而增加接收端进行纠错的能力。在接收端，通过统计接收到的 n 个比特中的“多数票”来恢复原始比特。具体而言，当 $n = 3$ 时，对于比特 1，发送端会发送 111，对于比特 0，发送端会发送 000。接收端若收到 101，则会通过“多数表决”将其判断为 1。这种信道编码实现简单，在较低噪声环境下能有效纠正单比特错误。然而，它也存在以下不足之处：

- **冗余度高**：每个比特需要 n 倍的传输资源，降低了信道利用率。
- **纠错能力有限**：仅适用于低噪声环境，且无法纠正多比特错误。
- **不适用于高速率传输**：对于需要传输大量数据的系统，会显著增加传输时间和带宽需求。

实际通信系统中会使用更加复杂高效的信道编码方法，如**汉明码**、**卷积码**、**低密度奇偶校验码**。

3.2.2 代码展示

```
function channol_encoded_signal = channol_encoding(encoded_signal, n)
    channol_encoded_signal = zeros(1, length(encoded_signal) * n);
    % Perform repetition encoding
    for i = 1:length(encoded_signal)
        for j = 1:n
            channol_encoded_signal((i-1)*n+j) = encoded_signal(i);
        end
    end
end
```

3.3 数字通信传输

3.3.1 系统原理

在数字通信系统中，信号在传输前需要经过调制，以便更适合在物理信道上传输。本系统采用了**二进制相移键控（Binary Phase Shift Keying, BPSK）**调制方式。BPSK 是一种最简单的数字调制方式，广泛用于无线通信和有线通信系统中。BPSK 通过对二进制数据进行相位变化来表示 0 和 1。其基本原理是将二进制数据映射到两个相位相差 180° 的信号上：

- 比特 0 映射为 $+1$（相位 0°）
- 比特 1 映射为 -1（相位 180°）

信号在传输过程中会受到各种噪声的干扰。本实验采用**加性高斯白噪声（AWGN, Additive White Gaussian Noise）信道模型**进行信道仿真。AWGN 信道具有以下特点：

- 噪声为高斯分布，均值为 0；
- 噪声能量通过信噪比（SNR）进行控制。

在接收端，需要对噪声污染后的信号进行解调。BPSK 解调的原理是基于接收到的信号幅度进行决策，该过程可以通过比较信号符号的正负性实现：

- 若接收到的信号大于 0，则判定为比特 0；
- 若接收到的信号小于 0，则判定为比特 1。

3.3.2 代码展示

```
% BPSK Modulation: Map bits to -1 and +1
modulated_signal = 2*encoded_channel_signal - 1;
% Transmit over AWGN channel
noisy_signal = awgn(modulated_signal, snr, 'measured');
% BPSK Demodulation: Map received values back to binary (1 or 0)
received_signal = noisy_signal > 0;
```

MATLAB 中使用 awgn 函数模拟噪声信道，该函数会根据设定的信噪比对输入信号添加噪声。

3.4 语音信号译码

3.4.1 系统原理

语音信号译码的过程可以分为信道译码和信源译码两个关键步骤：

1. 信道译码：在信道传输过程中，信号可能会受到噪声干扰，导致比特错误。信道译码的目的是尽可能纠正这些错误，从而恢复原始比特流。在本系统中，采用了多数表决的方法进行信道译码。具体实现方式如下：

- 将接收到的信号按照 n 位一组进行重塑。
- 对每一组 n 位信号，统计其中 1 和 0 的数量。
- 如果 1 的数量大于 0 的数量，则将该组信号解码为 1，否则解码为 0。

2. 信源译码：信源译码的目的是从经过信道译码后的比特流中恢复原始语音信号。在本系统中，使用了 Huffman 编码进行信源译码。解码过程如下：

- 利用已知的 Huffman 字典（dict）对信道译码后的比特流进行解码。
- 通过 MATLAB 内置函数 `huffmandeco` 完成解码，将比特流还原为原始符号序列。

这种分层的译码方式能够有效降低信道噪声带来的影响，同时保留信源的完整性，从而在接收端尽可能还原高质量的语音信号。

3.4.2 代码展示

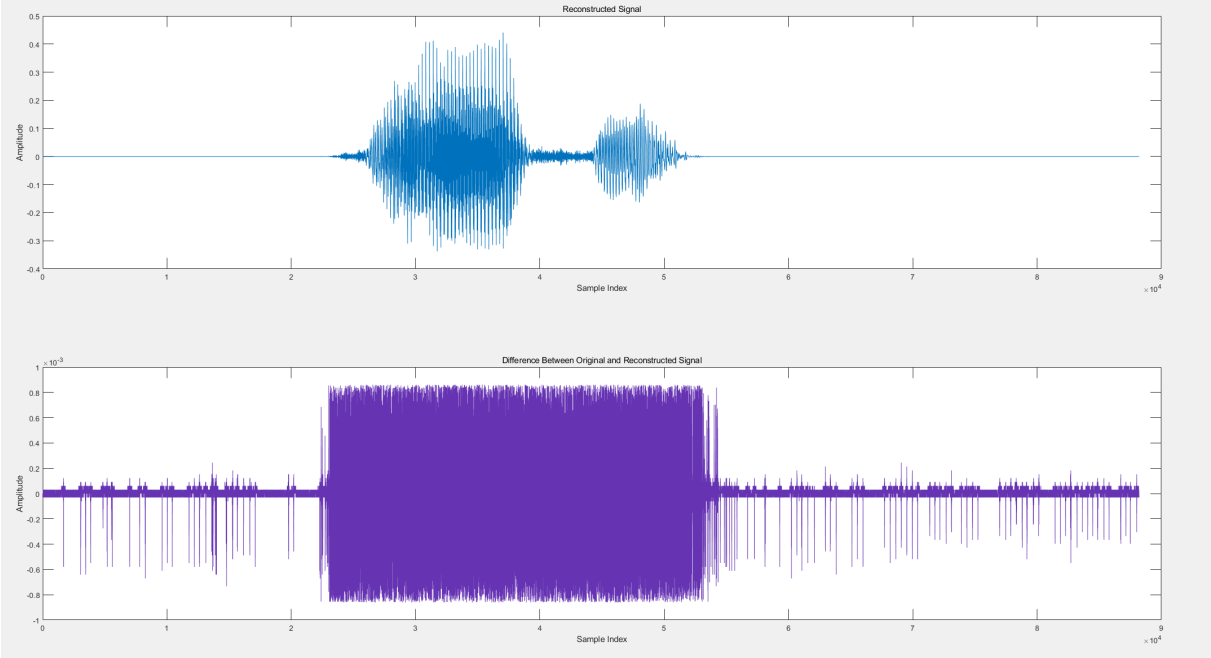
```
% channol_decoding
function channol_decoded_signal = channol_decoding(received_signal, n)
    % Reshape the received signal into groups of n for majority voting
    reshaped_signal = reshape(received_signal, n, []);

    % Majority voting to decode each bit without using mode
    channol_decoded_signal = zeros(1, size(reshaped_signal, 2));
    for col = 1:size(reshaped_signal, 2)
        ones_count = sum(reshaped_signal(:, col) == 1);
        zeros_count = sum(reshaped_signal(:, col) == 0);
        if ones_count >= zeros_count
            channol_decoded_signal(col) = 1;
        else
            channol_decoded_signal(col) = 0;
        end
    end
    channol_decoded_signal = channol_decoded_signal(:);
end

% huffman decoding
function reconstructed_signal = huffman_decoding(decoded_signal, dict)
    reconstructed_signal = huffmandeco(decoded_signal, dict);
end
```

3.4.3 处理结果

将译码后重建的信号与原信号做差对比（如下图所示）可以发现，信号的误差数量级约为 10^{-3}，因此该信号通过编码、信道传输、译码后准确性基本能够保持，可以经过计算识别后送入控制系统。至此，语音信号传输模块成功设计。



4 语音信号识别模块

语音信号识别基于 CNN 模型：首先创建语音数据集，训练一个能够完成对 “加速、刹车、停止” 进行分类的 CNN 模型，随后将完成解码的语音信号通过神经网络识别，得到对应的控制指令：

- 加速识别为 1；
- 刹车识别为 2；
- 停止识别为 3。

4.1 信号特征提取

语音信号经过特征提取过程可以大幅提高训练的效率和准确率，本实验使用了 MFCC（Mel 频率倒谱系数）特征。MFCC 是一种常用于语音信号处理的特征提取方法。它通过模拟人耳对不同频率的感知，将音频信号从时域转换到频域，并进一步提取低维特征，用于语音识别和音频分类等任务。其核心步骤包括：对音频信号进行分帧和加窗处理，计算每帧的傅里叶变换以获得频谱，然后通过 Mel 滤波器组映射到 Mel 频率尺度上，接着取对数幅度谱，再进行离散余弦变换，最终得到一组较低维且能够较好表示语音信号特性的 MFCC 特征系数。

```
1    features = mfcc(audio_signal, Fs, 'NumCoeffs', numCoeffs)';
```

(1). **音频文件导入与标准化：**预先录制好包含三个指令信号的音频，将音频文件以采样频率 F_s 进行读取，并对其进行标准化处理，以确保音频信号的幅值在 $[-1, 1]$ 范围内。

(2). **MFCC 特征提取：**对每个音频信号计算 MFCC 特征，其中使用了 13 个倒谱系数。该特征能够有效表示音频信号的时频特性，常用于语音识别任务。

(3). **特征存储与标签分配：**将每个音频信号提取的 MFCC 特征向量存储于特征矩阵中，同时为每个信号分配一个对应的标签，用于后续的训练和测试。

4.2 CNN 模型训练

将经过特征提取、标签化的数据集送入 CNN 网络训练，CNN 网络定义如下：

```
layers = [                          % CNN Architecture
    imageInputLayer([numCoeffs+1 cols 1])
    convolution2dLayer([3 3],16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer([3 3],32,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    fullyConnectedLayer(3) % 3 Classes
    softmaxLayer
    classificationLayer
];
% Set Training Options
options = trainingOptions('adam', ...
    'MaxEpochs',50, ...
    'MiniBatchSize',1, ...
    'InitialLearnRate',1e-3, ...
    'Verbose',true, ...
    'Plots','training-progress');
```

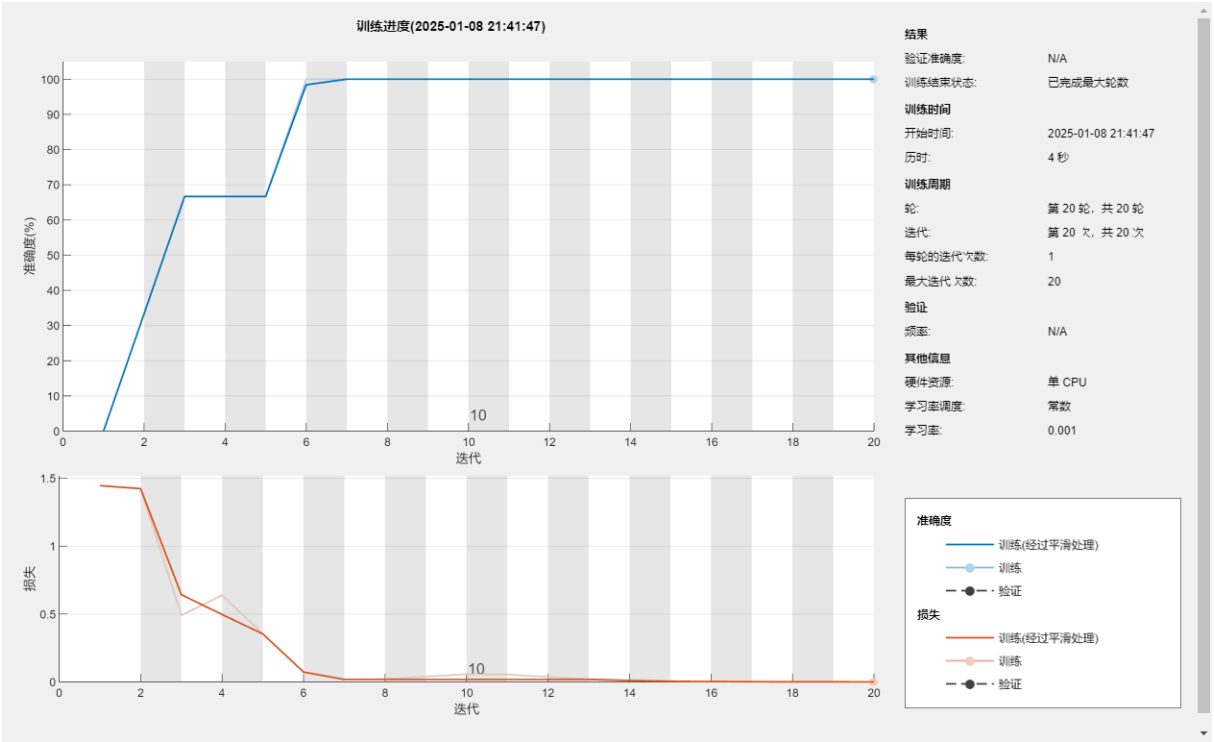
该代码定义了一个用于图像分类的卷积神经网络 (CNN)。网络输入为大小为 `[numCoeffs+1, cols, 1]` 的单通道语音图像，经过以下结构处理：

- **两层卷积块**: 每个包含 3×3 卷积层、批标准化层、ReLU 激活函数及 2×2 最大池化层。第一层有 16 个通道，第二层有 32 个通道。
- **全连接层与分类**: 使用一个输出为 3 的全连接层（对应 3 类分类），接 softmax 层和分类层完成最终分类。

训练设置：训练采用 Adam 优化器，最大训练轮数为 50，每轮使用单样本小批量训练，初始学习率设为 1×10^{-3}，并可视化训练过程。该网络适用于小型分类任务。

4.3 效果检验

以下是可视化训练过程，可以看到神经网络已经收敛。



录制”加速“音频，送入该神经网络进行识别，可以得到以下结果，网络识别成功。

```
Passing the reconstructed signal through the CNN...
Predicted Class: 1
```

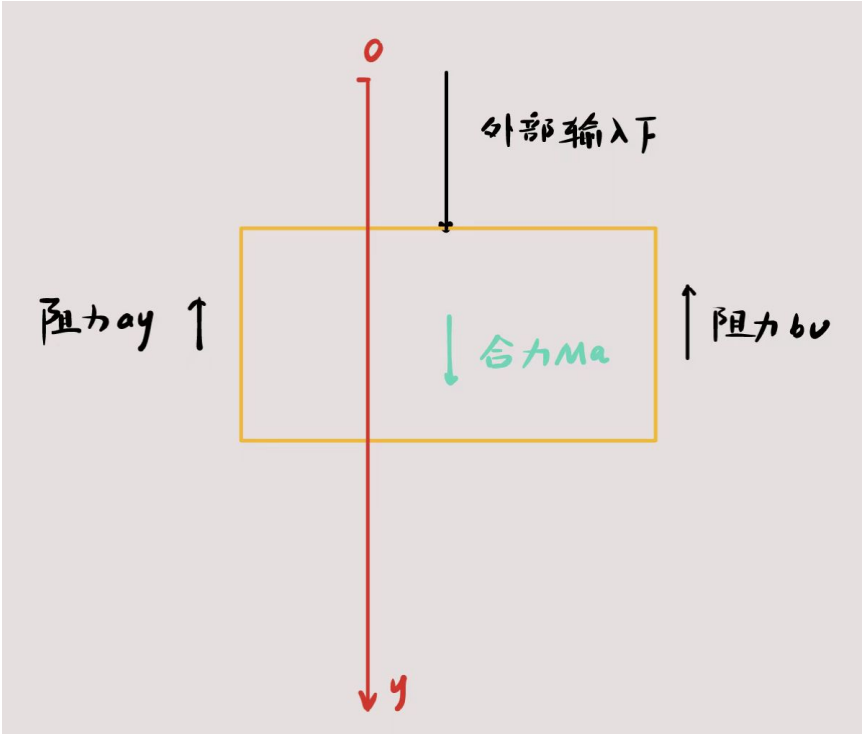
缺点说明：由于时间和数据集限制，经过分析识别模块存在以下不足之处：数据集样本数量较少，数据未经过常规的数据增强处理，导致模型的泛化能力不足，语音的语调、语速、音量变化可能会对指令识别造成影响，从而降低了模型在不同输入变化下的鲁棒性。因此该模型还存在优化空间；但目前正常朗读 3 条语音信号模型还是能准确识别的。

5 语音信号控制系统

输入控制系统的信号完成如下转化：

```
% Control Force Based on Predicted Class
if YPred_num == 1
    F = 5000;
elseif YPred_num == 2
    F = -5000;
else
    F = -50000;
end
```

经过转化，“加速、刹车、停止”信号能转变为力的信号，从而控制水下机器人速度。



5.1 控制方程

控制系统根据输入控制机器人的速度，其控制方程为：

$$M\frac{d^2y}{dt^2} = F - b\frac{dy}{dt} - ay$$

在上式中，F 为系统输入的外力，b 为水的阻力系数，ay 代表下潜深度对机器人产生的影响，M 为机器人质量。令 $x_1 = y, x_2 = \dot{x_1}$，可以得到如下状态方程与输出方程：

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & 1 \\ -\frac{a}{M} & -\frac{b}{M} \end{pmatrix}\mathbf{x} + \begin{pmatrix} 0 \\ \frac{1}{M} \end{pmatrix}F \quad \dot{\mathbf{y}} = \begin{pmatrix} 0 & 1 \end{pmatrix}\mathbf{x}$$

5.2 能控性判断

系统能控性的判定基于可控性矩阵 $\mathcal{C}$ 的秩：

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} \end{bmatrix}$$

若矩阵 $\mathcal{C}$ 的秩等于系统的状态向量维度（即 n），则系统是可控的。对于给定的状态方程：

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\frac{a}{M} & -\frac{b}{M} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ \frac{1}{M} \end{pmatrix}$$

计算可控性矩阵：

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{M} \\ \frac{1}{M} & -\frac{b}{M^2} \end{bmatrix}$$

计算矩阵的行列式：

$$\det(\mathcal{C}) = -\frac{1}{M^2}$$

因为 $b \neq 0$ 且 $M \neq 0$，则 $\det(\mathcal{C}) \neq 0$，所以系统是能控的。

5.3 能观性判断

系统可观性的判定基于可观性矩阵 $\mathcal{O}$ 的秩：

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}$$

若矩阵 $\mathcal{O}$ 的秩等于状态维度 n，则系统是能观的。对于给定的输出方程：

$$\mathbf{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

计算能观性矩阵：

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{a}{M} & -\dfrac{b}{M} \end{bmatrix}$$

计算其行列式：

$$\det(\mathcal{O}) = 0 \cdot \frac{b}{M} + 1 \cdot \frac{a}{M} = \frac{a}{M}$$

因为 $a \neq 0$ 且 $M \neq 0$，则 $\det(\mathcal{O}) \neq 0$，因此系统是能观的。

5.4 代码展示

控制系统部分代码实现如下，其中 F 为控制输入力，且只有在满足限制条件时才施加控制力：

- 速度最小为 0，最大不超过 40m/s。

代码使用 MATLAB 内置的 ‘ode45‘求解器对微分方程进行数值积分：

- ode45 基于 Runge-Kutta 方法，适用于求解非刚性常微分方程。
- velocity_control_ode 定义了动力学方程。

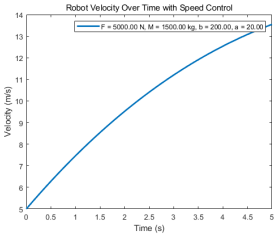
```
% Parameters
M = 1500;          % Mass of the robot (kg)
b = 200;           % Friction coefficient (N · s/m)
a = 20;            % Coefficient of depth impact (N/m)
v0 = 5;            % Initial velocity (m/s)
```

```
y0 = 0;          % Initial depth (m)
v_max = 40;      % Maximum velocity limit (m/s)
v_min = 0;       % Minimum velocity limit (m/s)
t_span = [0 5]; % Simulation time range (s)
% Define the state-space system considering velocity control
% x(1) = y (depth), x(2) = v (velocity)
velocity_control_ode = @(t, x) [x(2);
       (F * (x(2) < v_max && x(2) > v_min) - b * x(2) - a * x(1)) / M];
% Solve the ODE using ode45
[t, x] = ode45(velocity_control_ode, t_span, [y0 v0]);
% Plot the velocity curve
figure
plot(t, x(:,2), 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Velocity (m/s)');
title('Robot Velocity Over Time with Speed Control');
legend(sprintf('F = %.2f N, M = %.2f kg, b = %.2f, a = %.2f', F, M, b, a));
```

6 实验结果演示

分别录制“加速、刹车、停止”音频，MATLAB 输出结果如下：

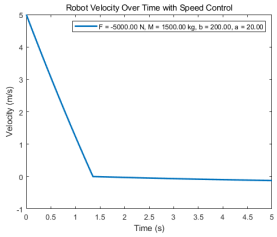
6.1 “加速”识别




```
Source Entropy H(X): 2.9286 bits/symbol
Code Rate R: 3.074 bits/symbol
Repetition encoding completed successfully.
BPSK modulation, AWGN channel transmission, and demodulation completed successfully.
BPSK modulation and transmission through AWGN channel complete.
Huffman decoding completed successfully.
Huffman decoding complete.
Passing the reconstructed signal through the CNN...
Predicted Class: 1
```

机器人逐渐加速，加速过程识别正确。

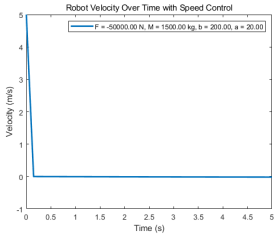
6.2 “刹车”识别




```
Source Entropy H(X): 3.7775 bits/symbol
Code Rate R: 3.8218 bits/symbol
Repetition encoding completed successfully.
BPSK modulation, AWGN channel transmission, and demodulation completed successfully.
BPSK modulation and transmission through AWGN channel complete.
Huffman decoding completed successfully.
Huffman decoding complete.
Passing the reconstructed signal through the CNN...
Predicted Class: 2
```

机器人逐渐减速，刹车减速过程识别正确。

6.3 “停止”识别




```
Source Entropy H(X): 4.2078 bits/symbol
Code Rate R: 4.238 bits/symbol
Repetition encoding completed successfully.
BPSK modulation, AWGN channel transmission, and demodulation completed successfully.
BPSK modulation and transmission through AWGN channel complete.
Huffman decoding completed successfully.
Huffman decoding complete.
Passing the reconstructed signal through the CNN...
Predicted Class: 3
```

机器人在短时间内立马停止，停止过程识别正确。

7 总结

本项目通过 MATLAB 仿真实现了一个完整的远程声控制系统，包括语音信号采集、量化、编码、传输与识别，并在水下机器人一维运动控制系统中完成了速度控制。通过引入霍夫曼编码和重复码，提升了信号传输的压缩效率与抗噪声能力；使用 BPSK 调制与 AWGN 信道模拟了真实通信环境下的信号传输过程；最终利用 CNN 模型成功识别了“加速、刹车、停止”三种语音指令，并实现了基于控制方程的速度调节，探讨了系统的能控性与能观性。从实验得到的结果来看，整个系统的功能得到了完整的实现，信号传输和识别的准确度符合预期，在一定程度上能够应对使用的需要。整个仿真过程中，我也遇到很多困难，比如无法把想法快速准确地用代码实现、实验流程繁琐但是时间有限。总之，经过一学期的学习，我掌握了信息理论、计算理论与控制理论的核心概念与方法，深刻体会了前人在领域开创过程中的创造性思维与严谨逻辑，也把学到的理论很好地应用了起来。