**基于MFCC-GMM声纹识别的树莓派声控小车**

**摘 要**

在现代社会，随着智能硬件的发展以及互联网的普及，越来越多的智能设备进入到我们的生活。与此同时，隐私保护、身份认证和加密也成为人们越来越关心的问题。目前指纹识别和人脸识别已经大范围应用在智能手机，智能门锁等领域中。声纹特征作为与指纹特征和面相特征相似的生物信息，目前还没有得到大规模的使用。声纹作为每个人特有的生物信息，经过相关实验，可以作为区别人与人身份的特征。声纹识别目前主要分为说话人识别和说话人认证两大领域，说话人识别是用来判断当前的说话人是已知训练模型的中的哪一位，而说话人认证则是用来判断当前说话人是否为训练模型中已知的人。目前对人声纹识别的研究还并不成熟，识别的准确率还达不到商用的标准。但声纹识别拥有广泛的应用前景，它具有很多方面的优势，如可扩展及移植性好，所需设备成本低，以及相较于指纹识别和人脸识别，声纹识别可把空气作为介质进行远距离传播。目前在某些安全部门已开始使用声纹识别作为身份认证，并有相关产品问世。本文主要探究声纹识别领域的相关算法和研究，并在分析之后选用合适的算法及模型设计一个基于树莓派开发板的声纹识别小车系统，主要工作如下：

学习了解说话人识别的相关算法技术。将传统机器学习模型与深度学习理论和说话人识别相结合。学习了解基于MFCC的声音特征提取、基于GMM和HMM的说话人识别、基于深度学习的说话人识别以及基于谱减法的语音降噪。

对各个模型和算法进行对比，进而分析各个模型和算法的优缺点。并针对已有的硬件设施设计出合理的系统。提出合理的功能和性能需求分析。

针对系统设计和需求分析，实现一个完整的基于说话人识别的树莓派小车系统，并对系统进行测试，检验系统是否达到设计要求。

**关键词**：声纹识别；梅尔频率倒谱系数；混合高斯模型；背景音降噪

**ABSTRACT**

In modern society, with the development of intelligent hardware and the popularity of the Internet, more and more smart devices are entering our lives. At the same time, privacy protection, identity authentication and encryption have become a growing concern. At present, fingerprint recognition and face recognition have been widely applied in the fields of smart phones and smart door locks. The voiceprint feature, as a biological information which is similar to fingerprint and facial features, has not yet been used on a large scale. As a unique biological information for each person, voiceprint can be used as a distinguishing feature between the person and the person through relevant experiments. Voiceprint recognition is mainly divided into two areas: speaker recognition and speaker authentication. Speaker recognition is used to determine who is speaking. Speaker authentication is used to determine if the current speaker has the right to open the device. At present, the research on voiceprint recognition is still not mature, and the accuracy of recognition is still not up to commercial standards. However, voiceprint recognition has broad application prospects. It has many advantages, such as scalability and portability, low equipment cost, and voiceprint recognition can use air as a medium for long distances transmission compared to fingerprint recognition and face recognition. Now voiceprint recognition has been used as an identity authentication in some security departments, and related products have been introduced. This paper mainly explores related algorithms and research in the field of voiceprint recognition. And to design a reasonable voiceprint recognition system, main tasks as follows:

(1) Learn about the relevant algorithmic techniques for speaker recognition and combine traditional machine learning models and deep learning theory with speaker recognition techniques. Learn about MFCC-based sound feature extraction, speaker recognition based on GMM and HMM, speaker recognition based on deep learning, and speech noise reduction based on spectral subtraction.

(2) Compare the different models and algorithms and analyze the advantages and disadvantages of each model and algorithm. Design a reasonable system for existing hardware facilities and develop a reasonable demand analysis.

(3) According to the system design and requirements analysis, implemented a complete speaker-recognition-based Raspberry Pi car system and verify that if the system meets the design requirements.

KEY WORDS: speaker recognition, MFCC, GMM, noise reduction

**目录**

[1. 绪论 1](#_Toc11396858)

[**1.1** **研究背景与意义** 1](#_Toc11396859)

[**1.2** **研究现状与分析** 2](#_Toc11396860)

[**1.3** **本文研究内容** 3](#_Toc11396861)

[**1.4** **本文结构安排** 4](#_Toc11396862)

[2. 相关理论与技术 6](#_Toc11396863)

[**2.1** **声音特征提取** 6](#_Toc11396864)

[**2.1.1** **梅尔频率倒谱系数（MFCC）** 6](#_Toc11396865)

[**2.2** **基于机器学习的声纹识别算法** 9](#_Toc11396866)

[**2.2.1** **混合高斯模型（GMM）** 9](#_Toc11396867)

[**2.2.2** **隐马尔可夫模型（HMM）** 14](#_Toc11396868)

[**2.3** **基于深度学习的说话人识别** 17](#_Toc11396869)

[**2.3.1** **深度神经网（DNN）** 18](#_Toc11396870)

[**2.4** **谱减法语音降噪** 19](#_Toc11396871)

[**2.5** **对比分析** 20](#_Toc11396872)

[3. 需求分析 22](#_Toc11396873)

[**3.1** **功能需求** 22](#_Toc11396874)

[**3.1.1** **树莓派小车基础功能** 22](#_Toc11396875)

[**3.1.2** **说话人识别功能** 22](#_Toc11396876)

[**3.1.3** **语音识别功能** 22](#_Toc11396877)

[**3.1.4** **数据抓取功能** 22](#_Toc11396878)

[**3.2** **性能需求** 23](#_Toc11396879)

[**3.2.1** **准确性** 23](#_Toc11396880)

[**3.2.2** **高效性** 23](#_Toc11396881)

[**3.2.3** **稳定性** 23](#_Toc11396882)

[4. 系统设计与功能 25](#_Toc11396883)

[**4.1** **设计思路** 25](#_Toc11396884)

[**4.2** **系统功能模块设计思路介绍** 27](#_Toc11396885)

[**4.2.1** **小车基本功能模块** 27](#_Toc11396886)

[**4.2.2** **语音唤醒模块** 27](#_Toc11396887)

[**4.2.3** **声纹识别模块** 29](#_Toc11396888)

[**4.2.4** **语音识别模块** 30](#_Toc11396889)

[**4.2.5** **数据爬取模块** 31](#_Toc11396890)

[**4.3** **系统流程** 31](#_Toc11396891)

[**4.3.1** **身份识别及信息录入** 31](#_Toc11396892)

[**4.3.2** **语音唤醒** 32](#_Toc11396893)

[**4.3.3** **输入指令及身份验证** 32](#_Toc11396894)

[**4.3.4** **语义分析及应答** 32](#_Toc11396895)

[5. 系统的具体实现与性能测试 34](#_Toc11396896)

[**5.1** **小车基本功能模块的实现** 34](#_Toc11396897)

[**5.2** **语音唤醒功能的实现** 37](#_Toc11396898)

[**5.3** **声纹识别模块的实现** 38](#_Toc11396899)

[**5.4** **语音识别模块的实现** 40](#_Toc11396900)

[**5.5** **数据抓取模块** 41](#_Toc11396901)

[**5.6** **性能测试** 42](#_Toc11396902)

[6. 总结与展望 44](#_Toc11396903)

[**6.1** **总结** 44](#_Toc11396904)

[**6.2** **展望** 44](#_Toc11396905)

[致谢 46](#_Toc11396906)

[附录一 英文原文 47](#_Toc11396907)

[附录二 中文翻译 55](#_Toc11396908)

参考文献

致谢

附录

1. 绪论
   1. **研究背景与意义**

声纹识别是通过分析声音的特征来识别说话人身份的技术，用来解决“谁在说话”的问题。对声音的识别包括语音识别和声纹识别，也就是通常所说的说话人识别。声纹识别与指纹识别和虹膜识别类似，通过分析人体的生物特征和行为来自动识别人身份。该项技术是由贝尔实验室在上个世纪四十年代末开发，主要用于军事情报领域，六十年代后期美国的法医鉴定，法庭证据等领域都使用了这项技术。如今声纹识别已经应用到各行各业，尤其在法庭审判和案件侦破中提供强有力的证据。

以指纹特征，人类面部特征，人类语音特征为代表的生物识别技术，在今后数年内将成为IT行业重要的技术革命。随着智能设备进入人们的日常生活，安全和隐私成为人们越来越关注的问题，由于面相，指纹等生物特征的独一无二，可以作为身份识别最好的ID。所以近些年来，指纹识别，人脸识别大范围的应用到包括智能手机在内的智能设备当中，作为一种有安全保障的识别方式为用户提供身份验证。由于每个人的声腔尺寸和发生器被操纵的方式不同，声音也可以作为确定一个人身份的工具。联邦调查局曾对2000多起与声纹识别相关的案件进行的统计，其中利用声纹识别作为证据的案件只有0.31%的错误率。所以声纹是一种与指纹，面相特征一样的，可靠的身份特征。而且相比较指纹认证需要用户与设备进行直接接触，声音可以在空气中传输，因此可以在很远的距离进行身份的验证。目前，声纹识别已经应用到证券交易、银行交易、个人设备声控锁、汽车声控锁、公安取证等各行各业。

* 1. **研究现状与分析**

本节介绍一下声纹识别的发展和研究现状。声纹识别技术分为两个大类：即说话人辨别和说话人确认技术。说话人辨别技术是从多个人中判断出当人说话的是其中的哪个一个人，说话人确认是用来判断未知说话人是否为某一个已知的人。声纹识别的常用方法有模板匹配法，VQ聚类法，最近邻近法。伴随着深度学习概念的问世，深度神经网络（DNN）[[1]](#endnote-1)，循环神经网（RNN）[[2]](#endnote-2)等方法的出现，对声纹识别技术的研究提供了越来越多的思路和方法。

这些方法虽然处理手段各不相同，但是基本原理是类似的。都是利用语谱图，通过提取说话人的基音频谱和包络，共振峰及其出现的频率和轨迹等参数特征，然后再与其他方法结合进行声纹识别。但是这些方法还存在着很大的局限和弊端。在对声音处理之前，需要对其进行很多处理，例如语音检测（VAD）、去噪，语音增强等。去除环境噪音和混响不仅对声纹的检测很重要，同时对语音含义的识别也很重要。

VAD通常有两种方式，Long-Term Spectral Divergence（LTSD）和能量检测，此外声音的特征主要通过梅尔频率倒谱系数(MFCC)[[3]](#endnote-3)提取，传统的机器学习模型主要有隐马尔可夫模型（HMM）[[4]](#endnote-4)和混合高斯模型（GMM）[[5]](#endnote-5)[[6]](#endnote-6)。虽然声纹特征有着很好的唯一性，但目前声纹识别技术还不能提供极其精确的分辨，尤其是在混合说话人的环境下和噪音很大的情况下。

随着近几年深度学习的发展带来了模式识别的提升，但对于声纹识别的研究进展仍不大。目前声纹识别技术存在的主要问题有：

1. 声纹的采集和特征的建立
2. 噪音问题
3. 混合人说话
4. 说话人身体状况
5. 说话人情绪

声纹识别还面临着诸多的问题，关于发展趋势，声纹识别和其他生物特征识别一样，向着深度学习的方向发展，但是由于深度学习是基于数据驱动的模型，要有庞大的真实场景的数据集，以及对数据的精确标注，目前还比较难实现。所以传统机器学习的算法和模型还会在目前的声纹识别研究中占有很大比重。

* 1. **本文研究内容**

本文的研究任务是打造一个基于声纹识别用户身份的声控智能小车。系统选用树莓派[[7]](#endnote-7)开发板，主要开发语言为python[[8]](#endnote-8)。基于MFCC声音特征提取，及传统机器学习中GMM模型的声纹识别小车。系统主要包含以下功能：

1. 系统语音唤醒
2. 声纹特征提取及训练
3. 噪声处理
4. 说话人识别和认证
5. 语音转文字（STT）
6. 语音控制小车行为
7. 文字转语音（TTS）
8. 简单聊天功能
9. 小车的基本运动，视频传输及超声测距

其中主要实现的功能为基于声纹识别的人物识别和人物认证以及背景环境音去噪。

* 1. **本文结构安排**

本文的主要结构安排如下：

1. 绪论。 主要介绍了声纹识别的研究背景和研究意义，然后介绍了声纹识别相关的研究现状，并且分析了声纹识别目前的难点以及发展趋势。最后介绍了本文的主要研究内容和结构安排。
2. 相关理论与技术。 本章介绍说话人识别和噪声处理领域的主要原理和算法。 首先介绍了基于梅尔频率倒谱系数（MFCC）的声纹特征提取，随后介绍了用于声纹识别的传统机器学习模型，以高斯混合模型（GMM）和隐马尔可夫（HMM）模型为主。第三小节介绍了深度学习在声纹识别领域的研究和现状，主要以深度神经网络（DNN）为主。第四小节介绍了近期比较流行的背景音去噪算法，分别介绍了基于主成分分析（PCA）[[9]](#endnote-9)[[10]](#endnote-10)的降噪算法和基于谱减法的降噪算法。最后本章对各小节提到的算法进行了比较和分析，指出各个算法的优缺点。
3. 需求分析。 本章主要对声纹识别小车系统做需求分析，分别从功能和性能上对该系统进行需求分析。功能需求分析了系统要实现怎样的功能，性能需求分析了系统要达到怎样的要求，如准确性、稳定性和高效性。
4. 系统的设计与功能。本章介绍了针对于第三章的需求分析，小车的设计思路，系统流程和功能模块。
5. 系统的具体实现与性能测试。本章按照需求分析和系统的设计流程来设计系统，介绍了每一个功能模块是如何设计实现的，每一个性能要求是怎样满足的。最后对系统的性能进行了测试，运行并展示得到的测试结果。
6. 总结与展望。 对本系统涉及的方法和研究及实现的效果进行总结，提出不足，并对未来的研究方向提出展望。
7. **相关理论与技术**

本章主要对系统所涉及到的相关技术进行论述。本系统的目的是设计一个可以识别出说话人身份的声控小车。目前对声纹识别的研究主要分为两大类，一类是基于机器学习的传统算法模型，另一类是基于深度学习神经网络算法模型。下面分别介绍声音特征的提取技术相关算法，声纹识别相关算法（包括传统机器学习模型和深度学习模型）以及语音降噪的相关算法。

* 1. **声音特征提取**

对于一段语音信息的研究，首先需要对声音进行预处理和声纹特征的提取。目前比较流行的声音特征的提取方法是梅尔频率倒谱系数（MFCC），线性预测系数（LPCS）和线性预测倒谱系数（LPCCs）。但随着梅尔频率倒谱系数（MFCC）的提出，线性预测系数（LPCS）和线性预测倒谱系数（LPCCs）逐渐被人们遗忘，目前在声音特征提取领域，以MFCC特征提取技术为主。通过梅尔频率倒谱系数（MFCC）得到声音特征后对其进行抽象表示，然后对抽象表示进行泛化处理，使用机器学习进行学习并得到相应的模型，最后用测试语音与该模型进行对比，对说话人进行预测。所以特征提取是模型训练的前提。下面详细介绍MFCC的原理和数学表达。

* + 1. **梅尔频率倒谱系数（MFCC）**

梅尔频率倒谱系数被广泛的应用于语音识别领域。由Davis和Mermelstein在1980年代提出，并逐渐成为最先进的技术之一。

梅尔频率倒谱是基于声音频率的非线性梅尔刻度的对数能量频谱的线性变换。相比正常对数倒频谱，能更好的近似人类的听觉系统。MFCC特征提取有两个关键的步骤：梅尔频率分析和倒谱分析。

梅尔频率分析：

实验研究表明人耳就像一个滤波器组，只关注一些特定的频率。也就是说，它只让某些频率的信号通过。并且在低频区有很多滤波器，高频区的滤波器就变得比较少。

梅尔刻度就是一种基于人耳对等距高音变化感官判断而定的非线性频率刻度。梅尔刻度滤波器组在低频部分分辨率高，与人耳的听觉特性相符。

倒谱分析：

对于一个语音频谱图，峰值表示语音的主要频率成分，称之为共振峰，而这些共振峰就携带了声音的特征属性，所以在语音识别中，我们主要把共振峰的位置和他们转变的过程提取出来，这个变化的过程是一个连接这些峰的平滑曲线，称之为频谱包络。

MFCC通常有以下几个过程：

1. 将一个语音信号分为多个讯框。
2. 将语音信号预加强。
3. 进行傅里叶变换，将信号转变至频域。
4. 把每个讯框得到的频谱通过梅尔滤波器，得到梅尔刻度。
5. 在每一个梅尔刻度上提取对数能量。
6. 对上述结果进行反离散傅里叶变换，得到倒频谱域。
7. 梅尔频率倒谱系数就是这个倒频谱图的幅度。[[11]](#endnote-11)[[12]](#endnote-12)

MFCC的原理：

声音信号是连续变化的，为了将该信号简化，假设在一个极短的时间间隔中，音频信号不发生改变。将这种采样点集合成一个单位，称它为讯框。每一个讯框大多为20-40毫秒，若讯框太短，那么每个讯框的采样点将不够做出可靠的频谱计算，若太长，则每个讯框信号会变化太大。

预强化是为了消除发声过程中，声带和嘴唇造成的影响，来补偿信号受到发声系统所压抑的高频部分。从而突出高频的共振峰。

由于在时域上很难看出信号的特征，所以通过傅里叶变换将它们转换成频域上的能量来观察，不同的能量分布代表不同的语音特性。

同时能量谱中存在的大量无用信息，其中包含了一些人耳无法辨别的高频变化，因此通过梅尔滤波器来消除这些讯息。

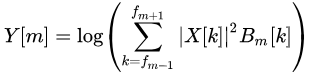
最后对得到的对数滤波器的能量进行离散傅里叶反变换，得到倒谱频域。

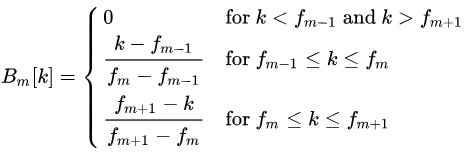
详细的推导过程：

1. 对信号进行傅里叶变换

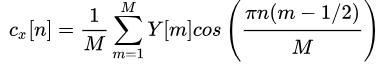
X[k]=FT{x[n]} (2.1)

1. 根据以下公式算出Y[m]，其中Bm[k]是梅尔频率倒频谱的遮罩。

 (2.2)

 (2.3)

1. 对Y[m]做IDCT得C:\Users\saluton\AppData\Local\Temp\1558663818(1).png

 (2.4)

MFCC局限性：

MFCC在加性噪声的情况下并不稳定，所以一般在使用MFCC时，通常要对其进行归一化处理，从而降低噪声的影响。[[13]](#endnote-13)

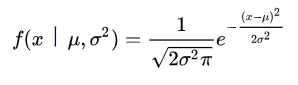
* 1. **基于机器学习的声纹识别算法**

目前比较流行的基于机器学习的声纹识别算法为混合高斯模型（GMM）和隐马尔可夫模型（HMM）。下面对这两种主流的算法进行详细介绍。

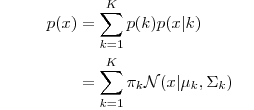
* + 1. **混合高斯模型（GMM）**

高斯混合模型是声纹识别中常用的训练模型。

高斯混合模型是对高斯模型的扩展，高斯混合模型采用多个高斯分布的组合来刻画分布图。高斯分布也就是常说的正态分布，是一种自然界中最为常见的分布形式。公式2.5为高斯分布的概率密度函数。

 (2 .5)

每个混合高斯模型（GMM）由K个高斯分布组成，这些高斯分布线性加在一起就组成了一个GMM概率密度函数见公式2.6。

 (2 .6)

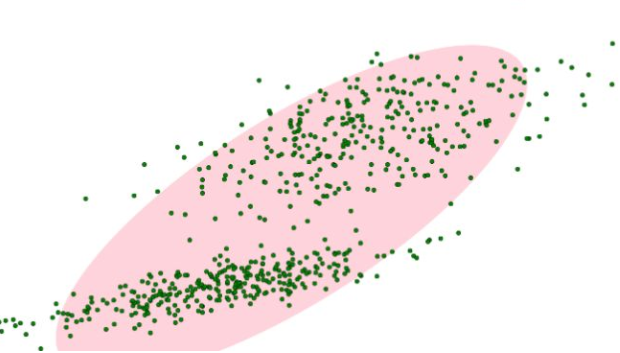


图 2.1 GMM数据采集点

如图2.1，绿色的点为观测信号，粉红色为拟合的模型，很明显一个一维的高斯分布式不能拟合这个信号的。类似这种情况，需要用多个高斯分布来拟合这个观测信号。



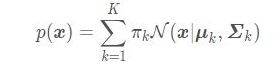
图 2.2 GMM数据采集点

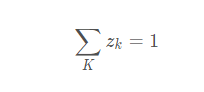
图2.2 表示了两个高斯分布对上述信号的拟合模型。

在语音识别中，观测到的信号就是带有噪声的信号，拿图2.1和图2.2来说，在这个语音当中的先验知识就是：这个语音里面有噪声，而噪声和人声的分布一般是不同的，所以就可以采用两个高斯分布来拟合这个信号。

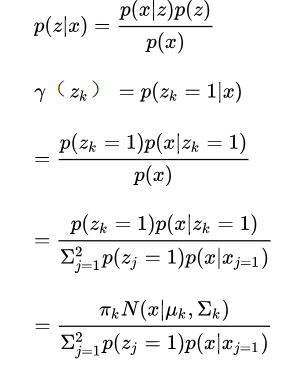
高斯模型采用了期望最大（EM）算法来进行训练。EM通过观察采样的概率值和模型的概率值的接近程度，判断训练的模型是否拟合良好。随后通过调整模型让新模型更加适合采样的概率值。不断迭代这个过程，直到两个值非常接近，停止更新并完成模型的训练，EM算法过程主要分为两步：E-step和M-step，推理和运算过程比价复杂，下面只进行简单的介绍。

EM算法过程：

 （2 .7）

公式2.7为GMM模型的表达式，在这里引入变量z，z表示选择第几类的概率 （2 .8）

我们需要在信号已知的情况下去进行分类，即：

 （2 .9）

公式2.9便为目标函数。根据当前的观测信号，估计出以及后验概率，然后再跟根据得到的后验概率重新估计，得出新的后验概率，一直迭代下去直至得出较好的拟合模型。

EM算法的核心思想是通过迭代的一个过程来找到一组最优的参数(μ∗,Σ∗,π∗)，使得这组参数表示的模型最有可能产生已有的采样数据，其实每次迭代过程就是一次矫正过程。图2.3是对EM算法优化的一个过程

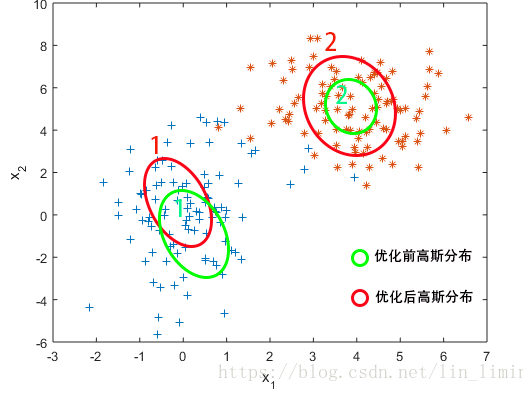


图2.3 GMM模型训练

图2.4梳理了混合高斯模型参数估计的逻辑流程

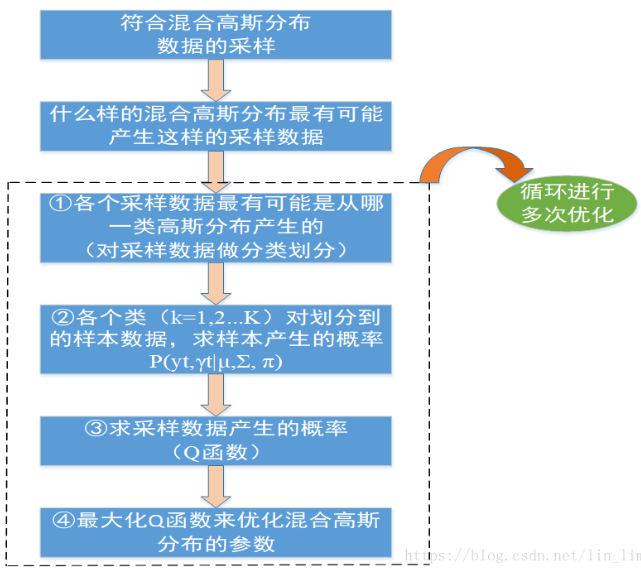
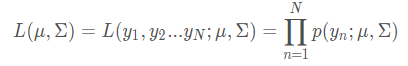


图2.4 GMM流程

 （2 .10）

公式2.10似然函数，所求出的参数就是最佳的高斯分布对应的参数。最大似然函数的意义就是通过使得样本集的联合概率最大对参数进行评估，从而选择最佳的模型。如下图2.5中所示，序号1对应的高斯分布式最佳模型。

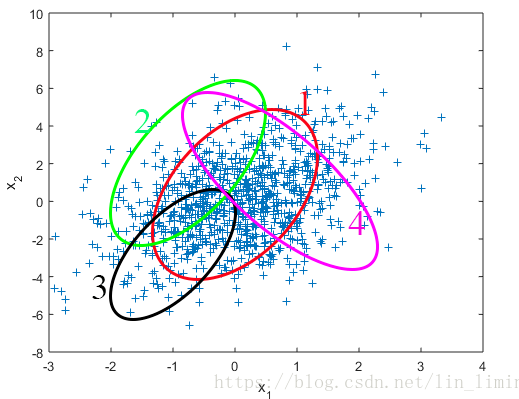


图2.5 GMM择优选择

* + 1. **隐马尔可夫模型（HMM）**

隐马尔可夫模型是一种比较经典的机器学习模型，在语音识别、模式识别和自然语言处理等领域都得到了广泛的应用。

隐马尔可夫模型是用来描述一个含有位置参数的马尔科夫过程。它可以从可观察的参数中确定出该过程的隐含参数，从而利用这些参数作进一步的分析。

在正常的马尔科夫模型中，状态是直接可见的。而在隐马尔可夫模型中状态并不是直接可见的，但是受状态影响的某些参量是可见的。一般来说，隐马尔可夫模型用来解决那些基于序列的（如时间序列和状态序列），并且在该序列当中一类数据是可以观测到的（观测序列），另一类序列是无法观察到的（状态序列）的问题。下图2.6强调了隐马尔可夫模型（HMM）状态变迁。其中x代表隐含状态，y代表可观察的输出，a代表转换概率，b代表输出概率。[[14]](#endnote-14)

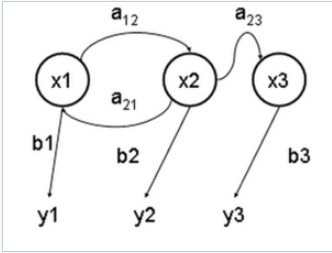


图2.6 HMM流程

HMM参数学习采用最大似然估计法

公式2.11为HMM的定义：

 （2 .11）

其中括号中的三个部分称为HMM三元素。

HMM研究的三个基本问题：

1.概率计算问题，给定和观测序列，计算在模型下，观测序列O出现的概率。

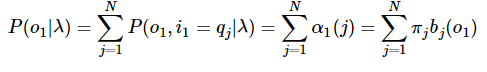
2. 学习问题。给定观测序列模型，估计模型使得最大。

3. 预测问题，给定和观测序列，求使最大的状态序列。

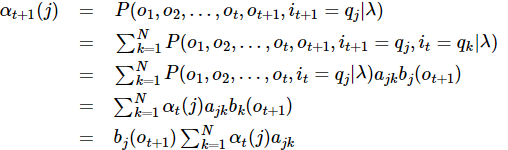
针对这三个问题，分别提出了前向算法，后向算法和Viterbi算法这里只给出简单的数学推导。

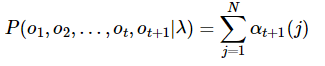
前向算法：

初始状态，考虑出现的概率：

 （2 .12）

在得到后，要求，可以先从出发：

（2 .13）

而，最终得到。

后向算法：

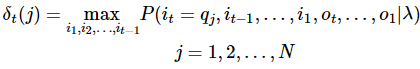
后向算法与前向算法过程类似，只不过后向算法是从后向前的因此最后的表达式为（2 .14）

其中 （2 .15）

Viterbi算法：

此算法用动态规划来解决HMM的预测问题，把隐藏的状态看出是最优路径上的一步，找最大的概率路径，即转化成寻找最优路径。

设t时刻最优路径为

 （2 .16）

于是可得

 （2 .17）

最初状态： （2 .18）

最后状态： （2 .19）

反向推导： （2 .20）

最终列 （2 .21）

其中。[[15]](#endnote-15)[[16]](#endnote-16)

* 1. **基**于深度学习的说话人识别

前面小节中介绍了基于机器学习的传统的说话人识别的模型。这些模型通过复杂的数学运算和精确的特征提取方法才能取得不错的效果。但随着深度学习概念的提出，说话人识别的研究逐渐在向深度学习的方向发展。目前比较流行的有基于深度神经网络（DNN）模型和身份验证矢量（i-vector）模型结合的方法。

* + 1. **深度神经网（DNN）**

神经网络是一组模仿人脸大脑构造而设计的算法，主要用于模式识别。对原始数据进行标记或者聚类。一般的传统机器学习系统主要由一个输入层和一个输出层组成的浅层网络。而在深度学习网络中，每一个节点层的学习是在前一层的基础之上。所以随着神经网络深度的加深，节点所能识别的特征就会越来越复杂。神经网络基于感知机的扩展，深度神经网络就是有很多隐藏层的神经网络。深度神经网络可以分为三层：输入层、隐藏层和输出层，如图2.7所示

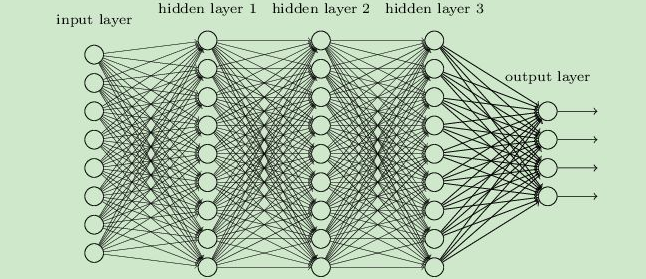


图2 .7 DNN网络示意图

第i层的任意一个神经元与第i+1层的任意一个神经元相连，从小的局部模型来说，模型满足一个线性关系加上激活函数。[[17]](#endnote-17)

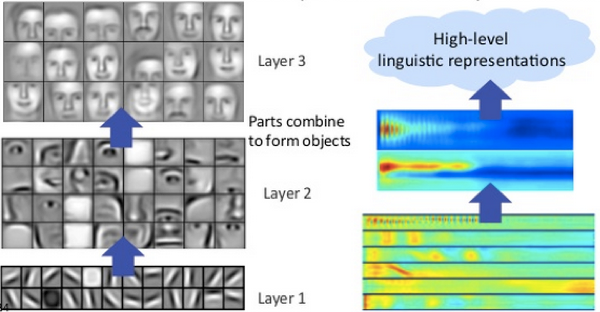


图2 .8 DNN网络示意图

除此之外，深度神经网络可以发现未标记的非结构化的数据中的潜在的结构，而且在真实的世界中，绝大多数数据属于这一类数据。与传统的机器学习相比，深度学习可以进行对特征的自动提取，不需要人为的干预。对于一些需要数据预处理和特征难提取的数据集，深度神经网络可以很好的解决这一类问题。

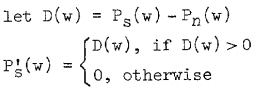
但由于深度神经网络需要大量的数据进行学习，花费大量时间进行数据标注，加之目前网络上行没有很好地开源数据集。目前基于深度学习的声纹识别还没有得到广泛的应用。在这里只做简单概述，本系统搭载在树莓派3B型号上，由于硬件条件有限，所以本系统还是采用了传统机器学习的方法来实现声纹识别功能。

* 1. **谱减法语音降噪**

谱减法原理：

谱减法语音降噪的原理很简单，就是用带噪声信号的频谱减去噪声信号的频谱，得到的就是一个没有噪音的频谱。谱减法基于一个基本假设，假设语音中的信号只有加性噪声，只要把带噪声的频谱减去，那么剩下的信号中肯定就不包含语音信号。谱减法去噪的前提是假设噪声信号是平稳或者缓慢变化的。[[18]](#endnote-18)

公式2.24为谱减法的数学原理：

 （2 .24）

其中Ps（w）是带噪音信号的频谱，Pn（w）是估计出的噪声频谱，二者相减得到D（w）频谱之差，因为相减后可能出现负值，所以将负值全部设为0，这样得到的结果为去除噪音的结果。

谱减法的局限性很大，在噪声变化非常明显的情况下，谱减法的效果并不明显。

* 1. **对比分析**

这里主要对比用于声纹识别模型的相关算法。深度神经网络相对于传统机器学习算法GMM和HMM来说，在准确度上有着很大的优势同时不需要人为的指出要学习的特征，但对硬件的要求极高，对数据集的需求巨大，目前还没有很好的相关的数据集。因此在本系统中，基于深度学习的声纹识别只做理论上的学习，不做具体的实践。

针对机器学习的两种模型混合高斯模型（HMM）和隐马尔可夫模型（HMM），混合高斯模型可以看作是隐马尔可夫模型的一种状态，所以隐马尔可夫模型中的每一种状态都可以看作是一个高斯混合模型。在性能方面隐马尔可夫模型相较于高斯混合模型来说有着更高的准确性，但是同时需要消耗更多的系统资源和处理时间。而混合高斯模型的处理时间比隐马尔可夫模型快很多，占用的系统资源少，同时也有着可观的准确率。所以经过对比，本系统采用混合高斯模型来训练声纹模型。

1. **需求分析**

本系统意在设计一个基于声纹身份验证的树莓派声控小车，对本系统功能需求和性能需求的分析整理如下。

* 1. **功能需求**

本系统大体上要满足四大部分的功能：树莓派小车部分的基本功能、说话人识别功能、语音识别功能和数据抓取功能。

* + 1. **树莓派小车基础功能**

树莓派小车要实现的最基本的功能是小车可以根据指令前进、后退、左转、右转和停止。除此之外，为避免小车撞上障碍物，小车应可以实现简单的自动避障功能。小车同时会通过摄像头等硬件传输实时视频，所以要提供UI界面供用观察和使用。

* + 1. **说话人识别功能**

此功能要实现对说话人身份的认证，因此要实现对说话人声音的录入，声音特征的提取，说话人模型的建立和训练，以及声音的确认与匹配。

* + 1. **语音识别功能**

语音识别功能要分别实现语音转文字和文字转语音的功能，语音转文字功能用于将说话人发出的指令转化成程序可以处理的字符串，进而分析出说话人语音的含义，从而执行不同功能。文字转语音功能用于将抓取到的不同数据以语音的方式播放给用户，实现文字转语音的功能。

* + 1. **数据抓取功能**

数据抓取功能实现对用户发出的指令要求做出应答。如用户询问天气情况，系统会通过查询中国天气网等网站抓取天气数据；询问时间，系统会查询当前的时间和日期；询问百科，系统会查询百度百科的摘要；翻译词语，系统会将词语或句子送至谷歌等翻译API进行翻译，并储存翻译结果。此功能模块实现一个简单的交互聊天功能，针对用户其他不满足要求的输入，此模块也会提醒用户，还不具备此功能。

* 1. **性能需求**
     1. **准确性**

系统通过声纹识别来验证说话人的身份，从而获取设备的使用权。因此系统的识别功能要保持很高的准确性，既要确保陌生人无法打开设备，又要保证用户可以正常使用设备。系统应尽量避免错判和误判。要保证陌生人拒绝访问的识别率为百分之一百，用户声纹的识别率在百分之九十以上。

* + 1. **高效性**

系统为实时交互系统，要保证画面传输的清晰及流畅，同时为实现系统和用户直接的实时交互。系统应快速对数据进行处理，在用户发出指令后，应立刻对用户的指令做出应答。

* + 1. **稳定性**

系统应保持长期稳定的运行，一旦开机，在保持供电的情况下，系统应保持持续稳定的运行，避免突然关机，退出系统的情况。并且有很强的鲁棒性，在运行各个分模块发生错误时或用户输入未知指令时，不应导致系统崩溃，而应有相应的应急方案和错误处理，从而达到系统的稳定以及健壮性。

1. **系统设计与功能**
   1. **设计思路**

小车基于树莓派开发板，所以考虑硬件条件以及性能需求，开发板并不支持深度神经网络的大量数据采集及训练，所以采用传统的机器学习模型来搭建声纹识别系统，对于声纹特征的采集，采用目前流行的MFCC算法。树莓派对python有着很好的支持，并且通过python程序可以控制树莓派上的GPIO，进而可以控制大量的传感器，同时python有着大量机器学习的库支持，所以选用python作为本系统的开发语言。针对功能需求的分析，系统将会分为四大模块，树莓派小车基本功能模块，语音唤醒模块，语音识别模块，声纹识别模块和数据采集模块，并且各个模块之间直接相互独立。

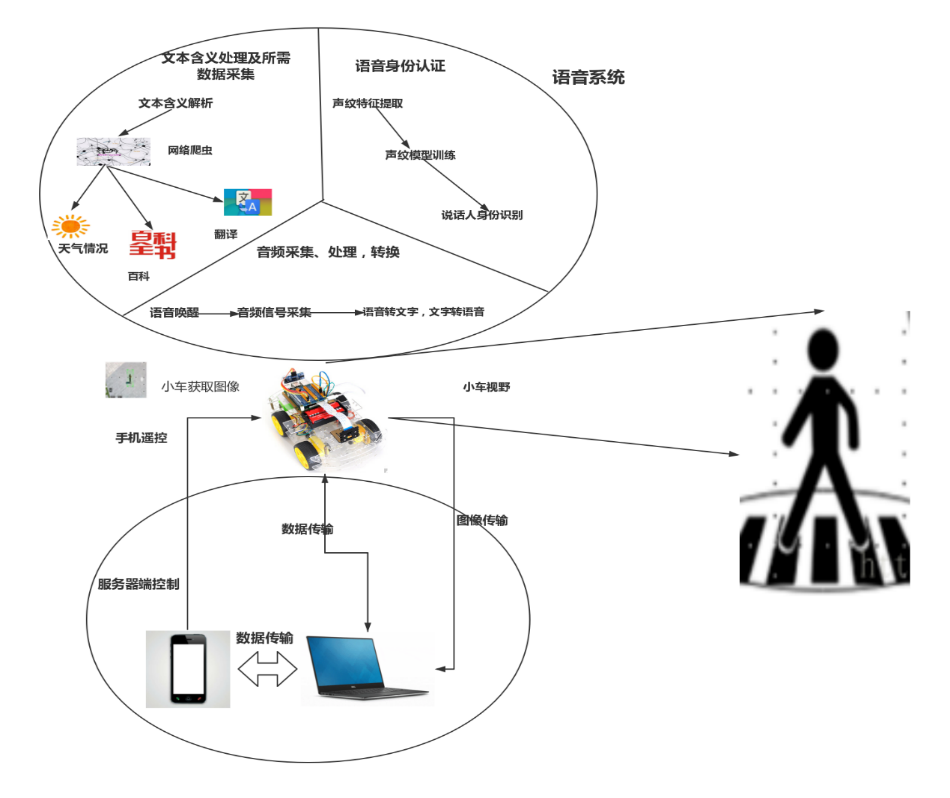


图4 .1 系统功能图

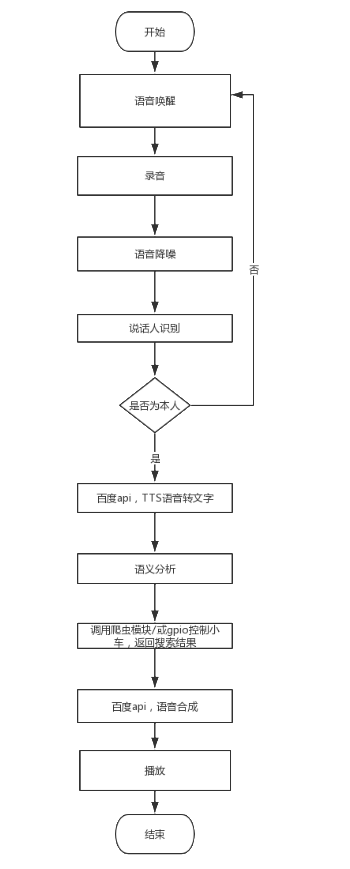


图4 .2 系统流程图

图4 .1为该系统设计的整体框架，图4 .2为语音识别和说话人认证的设计流程图。

* 1. **系统功能模块设计思路介绍**
     1. **小车基本功能模块**

针对功能需求的分析，该模块主要用来控制小车的运动以及图像传输和超声避障，小车可以根据指令实现前进、后退、左转，右转等功能。同时也可以通过车载摄像头，根据指令拍摄照片并且实时采集小车行进路上所拍摄的视频画面，将这些视频流传送到web服务器，以便人们观察。

由于树莓派开发板支持带有GPIO接口，所以支持外部传感器的接入。这里为实现小车的前后左右运动，选用L298n电机驱动模块，来接收树莓派的指令从而控制电机。同样，采用超声波模块来实现距离的采集，实现避障功能。视频传输采用树莓派官方摄像头和NGINX服务器搭建视频传输流。流程图如下：

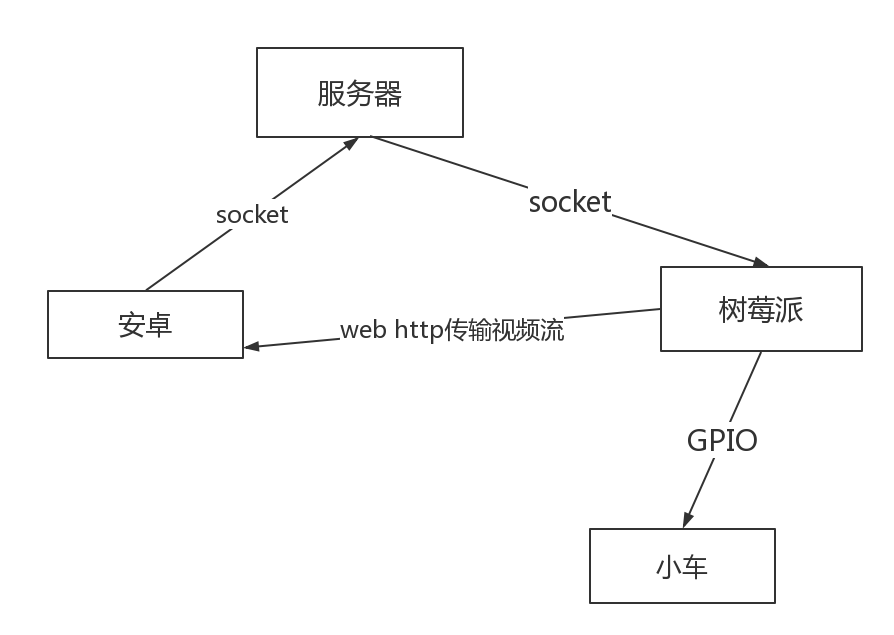


图4 .3 小车视频传输流程

* + 1. **语音唤醒模块**

针对功能需求的分析，语音唤醒模块应该保持在后台的持续运行，当用户说出关键字时，自动唤醒。语音唤醒模块的设计思路是执行运行在后台的程序监听用户的指令，一旦检测到用户的指令，就唤醒系统执行后续功能，针对性能需求语音唤醒模块应运行快，并且占用系统资源少。在运行唤醒功能前，应提前录制自己定制的关键词，系统运行时该程序会放置后台不间断运行，当程序检测识别到该关键词时调用回调函数，而这里的回调函数，为系统其它模块的接口。流程图如下：

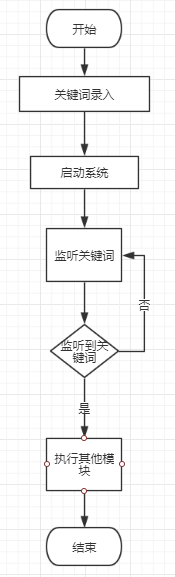


图4 .4语音唤醒流程

* + 1. **声纹识别模块**

针对功能需求的分析，该模块应实现用户声音特征提取，用户声纹模型的训练，以及说话人识别等功能。

考虑到系统的准确性，声音特征的提取采用MFCC算法，提取用户的声纹特征。

考虑到系统的局限性及准确性，声纹模型的训练采用传统的GMM机器学习模型。

该模块的设计思路为，模块应分为两大部分，一是用户信息的录入和用户声纹模型的训练及建立，二是测试语音身份的认证和指令的反馈。流程图如下：

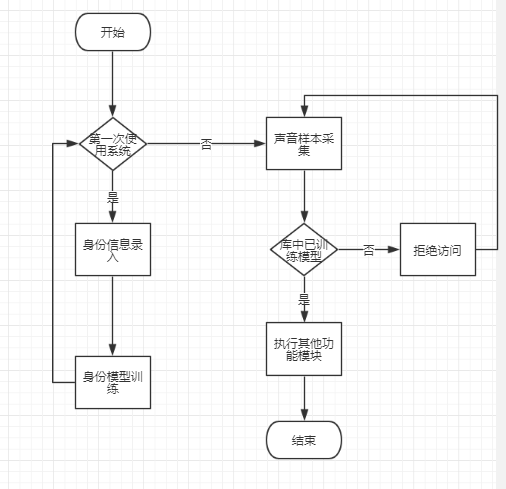


图4 .5 声纹识别流程

* + 1. **语音识别模块**

针对功能需求的分析，语音识别模块应实现音频录制，语音数据的降噪，语音数据转文字数据，文字数据转语音数据。设计思路为在每一次用户唤醒系统后，提示用户进行录音，随后对每一段录音进行去噪，将去噪后的录音进行语音数据转文字数据，并传送给数据抓取模块进行处理，最后将抓取到的文字数据转成语音数据并播放给用户。流程图如下：

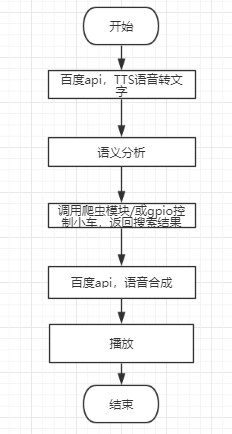


图4 .6 语音识别流程

* + 1. **数据爬取模块**

针对功能需求分析，数据爬取模块的作用是对说话人所描述的语义分析后，进行的反馈数据的查询和爬取。例如当说话人对小车说“北京的天气如何”并且通过了模型认证，这时候数据爬取模块就会根据北京和天气这两个关键字，去网上搜索相关的信息并返回相应的结果给用户。这个模块实际上是实现了一个简单语音聊天功能。

该模块的设计思路是，对用户的命令进行字符串匹配，当用户命令与某个字符串完全匹配或部分匹配时，就去执行相应的数据抓取。天气功能采用python爬虫，根据指令的城市，转换成城市代码，从中国天气网中爬取信息；百度百科模块，根据用户的指令，提取相应的关键字，从百度百科中爬取摘要信息；翻译模块，根据用户的指令，提取要翻译的内容，并送至谷歌翻译进行翻译，返回翻译后的结果。

* 1. **系统流程**

本系统是一个面向个人的说话人识别的声控小车系统，理想地情况下实现小车只对存入数据的人提供服务。下面是小车系统的主要流程：

* + 1. **身份识别及信息录入**

第一次使用该系统时，系统应要求用户提交手册上的秘钥来实现身份认证，认证通过后，系统应要求当前用户进行声纹的录入，作为用户的声音特性录入到系统中去。随后系统会立刻将录入的用户信息进行去噪，特征提取和模型训练，生成该用户对应的声纹模型。（在信息录入时，尽量保证在时相对安静的环境下录音）。

* + 1. **语音唤醒**

当用户身份信息正确录入后，系统会在后台自动并且不间断运行，监听用户的指令。这时可以通过唤醒词来唤醒系统。当系统监听到关键字后，便会唤醒系统回复用户提示音，并进入下一阶段。

* + 1. **输入指令及身份验证**

当系统被自动唤醒后，用户便可以通过语音对系统下达指令。系统在接收到指令后，对语音中的噪声进行处理，然后进行声纹识别，若识别是已有模型的人声，则进入下一阶段。若认证失败，则会语音输出“无权使用”。

* + 1. **语义分析及应答**

通过身份认证后，系统会继续向下一阶段进行，将用户的语音指令通过语音识别模块将用户的语音数据转成文字数据，随后将这些文字送至数据抓取模块进行功能匹配。功能分为两大类：一类是控制小车的运动，另一类是获取一些生活信息。系统会根据说话人的指令去控制小车的运动，图片的抓取。或提供一些如天气，词语意思，百度百科等信息。如果上述信息均未匹配到，则系统会提醒用户，系统还不支持当前命令或听不懂当前命令。最后将抓取到的文字数据通过语音识别模块转换成语音数据输出给用户。

此后小车会一直在2-4步骤间进行循环，不断等待唤醒，接收指令。流程图如下：

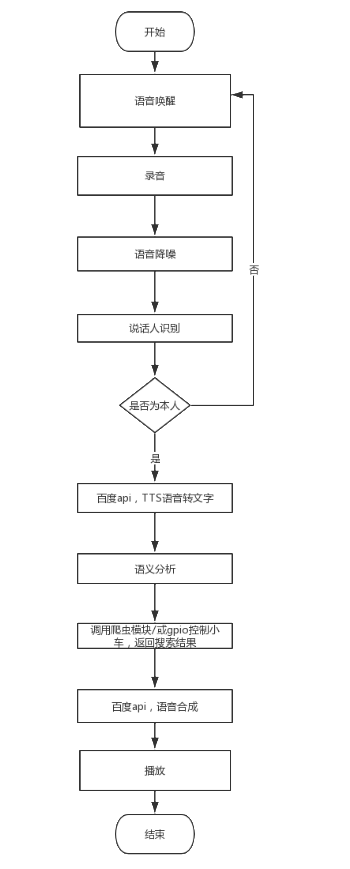


图4 .7 系统流程

1. **系统的具体实现与性能测试**
   1. **小车基本功能模块的实现**

小车的基本功能包括运动，超声测距以及视频传输和图像抓取。

本系统采用GPIO控制L298n电机驱动模块，来实现电机的转动，从而实现小车的运动。L298n驱动模块的引脚图如下：

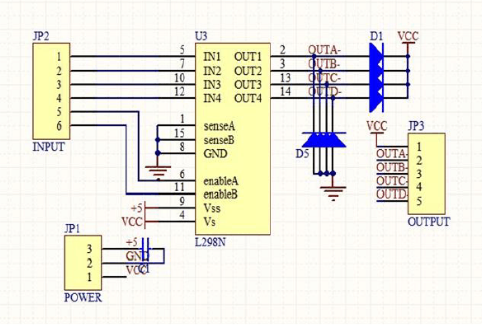


图5 .1 l298n接口

如图5 .1 所示l298n内部包含了四通道逻辑驱动电路。模块接口如下：

+5V：芯片电压5V。

VCC：电机电压，最大可接50V。

GND：共地接法。

A-~D-：输出端，接电机。

A~D+：模块上接VCC。

EN1、EN2：高电平有效，EN1、 EN2分别为IN1 和IN2、IN3 和IN4的使能端。

IN4：输入端，输入端电平和输出端电平是对应的。

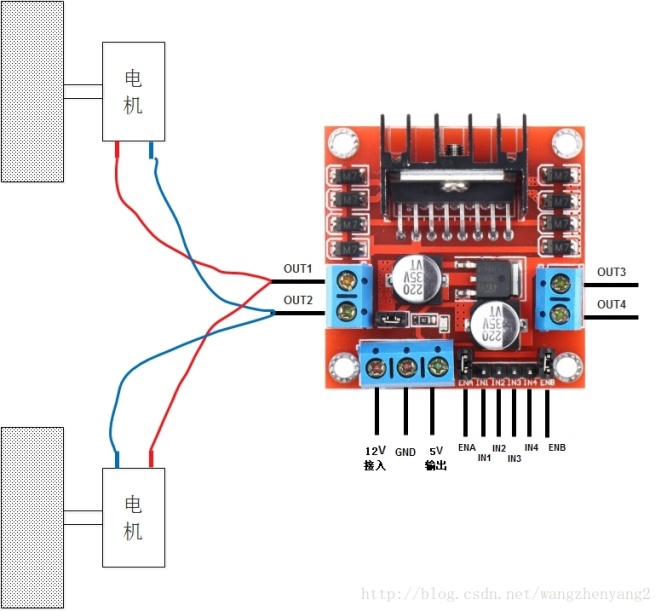


图5 .2 l298n线路连接图

图5 .2 为L298n模块与电机的接线图。

表5. 1



表5.1 为L298n的真值表。

超声测距采用超声波测距传感器，每3毫秒发送一次信号，并计算出距离。超声波传感器的引脚图如下图5 .4：



图5 .4 超声模块

超声波模块利用超声波反射的原理来检测距离，测量范围较小但是精度较高。图5 .4引脚分别为VCC GND是电源（5 v），TRIG是控制端（输入），ECHO是返回端（输出）。

使用时首先需要在trig引脚上输入一个长为20us的高电平方波，这样会触发模块发射8个40kHz的超声波，同时echo引脚的电平会由0变为1。当超声波返回被接收时，echo的电平由1变为0。用定时器记录下这个时长t，则距离=340\*t/2。

视频传输功能通过树莓派官方传感器配合开源的mgjp-stream搭建视频流服务器，效果图如下图5 .5：



图5 .5 视频采集

* 1. **语音唤醒功能的实现**

针对功能需求分析和性能需求分析，语音唤醒模块要保证准确快速同时低功耗，占有系统资源低的特征。本系统采用开源软件snowboy来实现语音唤醒功能。但系统将snowboy的开源代码进行了修改，因本系统第一次使用与后续使用存在差异，所以将第一次使用时的功能加入在唤醒功能之前，因此对代码进行了比较大的修改。修改后的程序在第一次使用时，不包括唤醒功能，并且此时系统会提醒用户录制自己用来唤醒系统的关键字。之后使用系统时，系统会不断监听关键字，并在监测成功后，调用其他功能模块的接口。运行图如下：

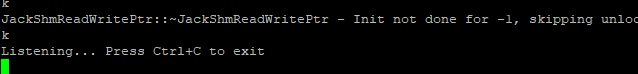


图5 .6语音唤醒



图5 .7系统应答

图5 .6为开启语音唤醒功能，系统正在不断监听用户是否正在唤醒系统。图5 .7显示，在何时系统检测到了用户的唤醒，并回复“小宗在的”，等待用户的下一步指令。

* 1. **声纹识别模块的实现**

针对功能需求分析，声纹识别模块要对用户进行声纹信息的提取，模型的训练以及说话人的识别。

本系统采用MFCC-GMM架构进行声纹特征的提取和模型的训练及匹配。具体实现方法为：

通过Python\_speech\_features库中MFCC类，来生成声纹特征向量。

python\_speech\_features.base.mfcc，其中主要参数如下：

signal-用于特征提取的音频特征信号，需是一个N\*1维数组。

samplerate-音频信号的采样率。

numcep-返回的倒谱数量。

nfft-FFT的大小，默认为13。

winstep-连续窗口的时间间隔，默认为1毫秒。

返回一个包含特征信息的numpy数组，数组的每一行代表一个特征向量。

通过sklearn库中的GMM模型来进行模型的训练和身份的匹配。

具体用法如下：

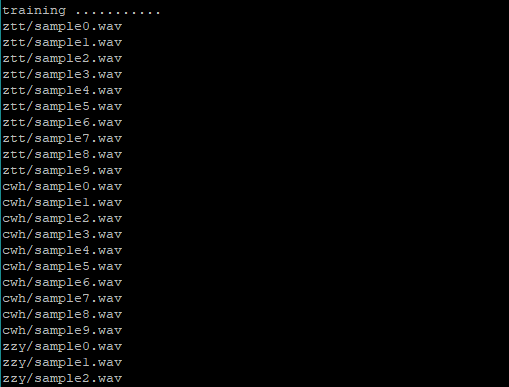


图5 .8 训练过程

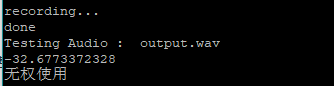


图5 .9拒绝访问

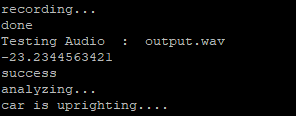


图5. 10认证成功

该模块分为用户信息的录入和音频身份的认证两大部分，图5 .8展示了系统正对用户的声纹特征进行训练，在此分别对5个人的10组声纹模型进行训练，大约花费30分钟。

图5. 9和图5 .9分别展示了当经过身份认证后该段语音所得的分数，系统会根据不同的得分来判断此用户是否为当前注册用户，以此来判断是否有权限继续使用系统。若无权使用，系统则会语音广播“无权使用”。

* 1. **语音识别模块的实现**

针对功能需求分析，语言识别模块应对录音进行语言去噪，语音数据转文字数据和文字数据转语音数据。

对于去噪，本系统采用了谱减法实现语音去噪，谱减法通过将录音前的一段时间当做背景噪声，进而将整段录音的频谱与噪音的频谱相减，得到纯净的语音信号。实现效果如下图5. 11和5 .12：

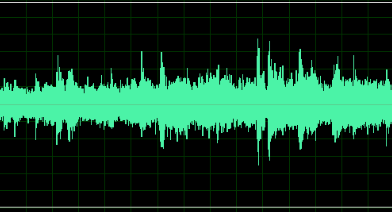


图5 .11加噪语音



图5 .12 去噪语音

图5 .11为带有噪音的信号，图5. 12为去除噪音的信号。

语音数据转文字数据和文字数据转语音数据由于工作量和难度较大，这里采用了通过百度语音识别api来实现。将语音数据发送给百度api即可得到相应的文字数据，同理可得到文字数据转语音数据。

* 1. **数据抓取模块**

针对功能需求分析，数据抓取模块主要实现对用户指令的各种信息的抓取。

本模块主要通过字符串的匹配和python爬虫来实现，实现效果如下：





图5. 13天气情况

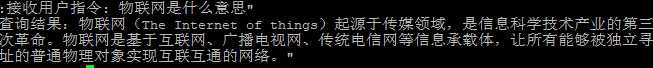


图5. 14 百科



图5. 15 翻译

图5. 14显示对于天气数据的抓取该模块返回的结果。图5 .14显示对于百科类指令的返回结果，图5. 15为翻译类指令的抓取。

* 1. **性能测试**

本系统基本完成系统设计时所提到的所有功能，从总体看上看，系统目前只能实现单一的说话人识别或说话人认证（说话人识别指在多个已知人员中，识别出当前的说话人，说话人认证指在已知一名用户的前提下，判断当然说话人是不是已知用户），不能同时保证两个功能的实现。除此之外，系统实现了小车的基本功能，语音唤醒功能，语音识别功能和数据抓取功能。完成了极大部分的功能需求。

在性能需求上，也基本保证了系统的准确性，稳定性和高效性。

在系统准确性方面，在所测试的100个样例中，其中声纹识别的准确率可以达到97%，同时声纹认证的准确率达到了83%，这主要是由于噪声的影响，在安静的环境下，声纹认证的准确率也可达到90%以上。基本满足了性能需求中准确性的要求。如图5. 16：

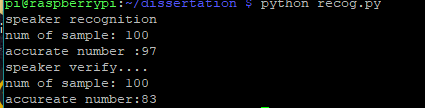


图5. 16 性能测试

在系统稳定性上，在连续供电运行十个小时，并使用100次以上的情况下，系统没有出现过崩溃，并且可以按照系统设计流程正常运行。系统满足性能需求中稳定性的要求。

在系统高效性上，经测试在训练10组每组1分钟的用户语音数据的前提下，系统大约花费七到八分钟进行模型的学习和建立，在身份认证时大约花费2秒到3秒，在数据抓取及语音合成时大约每次花费2秒。所以经测验，系统大致满足性能需求中高效性的要求，但由于硬件设备条件有限，还不能做到用户指令下达的同时返回结果。

经测试，系统基本完成了所有的功能需求以及性能需求。

1. **总结与展望**
   1. **总结**

本系统初步实现了需求分析所要求的功能，基本实现了一个基于说话人认证的声控小车系统。本系统在测试的过程中提取了15个不同说话人的语音特征并进行模型训练，并对100个测试样例进行了测试，其中对97%的测试样例进行了正确的匹配。在说话人认证方面，系统需反复观察测试人与模型直接的得分情况，经过大部分的实验，在大多数情况下测试人员的分数在22与25之间，而其他声源的得分一般大于29，在对用户单人身份确认的情况下测试了100组样例，该系统也对其中83%的测试用例进行了正确的匹配。系统基本完成了预期目标。当然，本系统还存在着很多的不足之处，如噪音对系统的影响仍然很大，如在噪音非常大的公共场所，经常出现误判错判等问题。同时，对于混合了多人说话的场景，该系统仍不能解决这个问题，当用户和其他人员一同说话时，系统会将这段语音识别为陌生人而拒绝掉。除此之外，还存在人机聊天的功能过少，声纹认证所需时间过长等问题。

* 1. **展望**

本系统是建立在已有理论和知识模型上的一次设计，基本实现了一个基于MFCC-GMM声纹身份认证的树莓派声控小车，但如上所说还存在着很多方面的问题，如噪音问题，混合说话人问题等。目前比较处于比较先进的课题说话人聚类（speaker diarization）可以解决混合人说话问题，说话人聚类的目的是为了解决谁在什么时候说话（who speak when）的问题，可以在没有任何已知说话人声纹特征和模型数据的情况下，将一段语音中不同的说话人在何时开始说话，并在何时结束说话标注出来。这可以很好的解决混合说话人问题，通过说话人聚类把混合说话人中不同说话人在何时说话进行标注，从而再将不同的标注进行说话人识别从而解决本系统中说话人混音的问题，并且此种方法可以不讲其他人的语音当成噪音进行处理。但目前说话人聚类还处于研究阶段，还没有得到很好的应用。在之后的研究中，希望可以深入研究说话人识别相关的算法和技术应用已及说话人聚类，从而更好的完善该系统。

**致谢**

光阴似箭，岁月如梭。转眼已到了毕业的时候，四年大学生活即将结束，回首往事，在这四年的学习和生活中有过心酸，有过感动。值此毕业论文完成之际，我谨向所有关心和帮助我的人们致以衷心的感谢。

首先感谢我敬爱的导师房胜教授，导师严谨的教学态度，乐观开朗的生活态度，深深的影响了我。从最初的选题、开题，到文章的写作和修改，到最终论文的定稿。感谢房胜老师对我论文的写作和完成过程提供了大量的帮助和指导。

感谢与我一起生活的舍友，在生活中遇到困难时，他们给予我无私的帮助，使我度过了丰富多彩的大学生活。

最后感谢我的家人。他们是我前进的动力，在我困惑和迷茫时给我鼓励，我将继续努力，不负你们期望。

感谢我的母校-山东科技大学。

**附录一 英文原文**

**Mel-frequency cepstrum**

In [sound processing](https://en.wikipedia.org/wiki/Sound_processing), the mel-frequency cepstrum(MFC) is a representation of the short-term [power spectrum](https://en.wikipedia.org/wiki/Power_spectrum) of a sound, based on a [linear cosine transform](https://en.wikipedia.org/wiki/Cosine_transform) of a [log power spectrum](https://en.wikipedia.org/wiki/Power_spectrum) on a [nonlinear](https://en.wikipedia.org/wiki/Nonlinear_system) [mel scale](https://en.wikipedia.org/wiki/Mel_scale) of frequency.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC[[1]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-1). They are derived from a type of [cepstral](https://en.wikipedia.org/wiki/Cepstrum) representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the [cepstrum](https://en.wikipedia.org/wiki/Cepstrum) and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in [audio compression](https://en.wikipedia.org/wiki/Data_compression#Audio).

MFCCs are commonly derived as follows:[[2]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-2)

1. Take the [Fourier transform](https://en.wikipedia.org/wiki/Fourier_transform) of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the [mel scale](https://en.wikipedia.org/wiki/Mel_scale" \o "Mel scale), using [triangular overlapping windows](https://en.wikipedia.org/wiki/Window_function#Triangular_window).
3. Take the [logs](https://en.wikipedia.org/wiki/Logarithm) of the powers at each of the mel frequencies.
4. Take the [discrete cosine transform](https://en.wikipedia.org/wiki/Discrete_cosine_transform) of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

There can be variations on this process, for example: differences in the shape or spacing of the windows used to map the scale,[[3]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-3) or addition of dynamics features such as "delta" and "delta-delta" (first- and second-order frame-to-frame difference) coefficients.[[4]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-4)

The [European Telecommunications Standards Institute](https://en.wikipedia.org/wiki/European_Telecommunications_Standards_Institute) in the early 2000s defined a standardised MFCC algorithm to be used in [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone).[[5]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-etsi01-5)

Application

MFCCs are commonly used as [features](https://en.wikipedia.org/wiki/Features_(pattern_recognition)) in [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition)[[6]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-6) systems, such as the systems which can automatically recognize numbers spoken into a telephone.

MFCCs are also increasingly finding uses in [music information retrieval](https://en.wikipedia.org/wiki/Music_information_retrieval) applications such as [genre](https://en.wikipedia.org/wiki/Genre) classification, audio similarity measures, etc.[[7]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-7)

noise sensitivity

MFCC values are not very robust in the presence of additive noise, and so it is common to normalise their values in speech recognition systems to lessen the influence of noise. Some researchers propose modifications to the basic MFCC algorithm to improve robustness, such as by raising the log-mel-amplitudes to a suitable power (around 2 or 3) before taking the DCT ([Discrete Cosine Transform](https://en.wikipedia.org/wiki/Discrete_cosine_transform)), which reduces the influence of low-energy components.[[8]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-8)

history

Paul Mermelstein[[9]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum" \l "cite_note-merm76-9)[[10]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-merm80-10) is typically credited with the development of the MFC. Mermelstein credits Bridle and Brown[[11]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-11) for the idea:

Bridle and Brown used a set of 19 weighted spectrum-shape coefficients given by the cosine transform of the outputs of a set of nonuniformly spaced bandpass filters. The filter spacing is chosen to be logarithmic above 1 kHz and the filter bandwidths are increased there as well. We will, therefore, call these the mel-based cepstral parameters.[[9]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-merm76-9)

Sometimes both early originators are cited.[[12]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-12)

Many authors, including Davis and Mermelstein,[[10]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-merm80-10) have commented that the spectral basis functions of the cosine transform in the MFC are very similar to the [principal components](https://en.wikipedia.org/wiki/Principal_components) of the log spectra, which were applied to speech representation and recognition much earlier by Pols and his colleagues.[[13]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-13)[[14]](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#cite_note-14)

**Hidden Markov Model**

Hidden Markov Model(HMM) is a [statistical](https://en.wikipedia.org/wiki/Statistical_model) [Markov model](https://en.wikipedia.org/wiki/Markov_model) in which the system being modeled is assumed to be a [Markov process](https://en.wikipedia.org/wiki/Markov_process)with unobservable (i.e. hidden) states.

The hidden Markov model can be represented as the simplest [dynamic Bayesian network](https://en.wikipedia.org/wiki/Dynamic_Bayesian_network). The mathematics behind the HMM were developed by [L. E. Baum](https://en.wikipedia.org/wiki/Leonard_E._Baum) and coworkers.[[1]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-3)[[4]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-5)HMM is closely related to earlier work on the optimal nonlinear [filtering problem](https://en.wikipedia.org/wiki/Filtering_problem_(stochastic_processes)) by [Ruslan L. Stratonovich](https://en.wikipedia.org/wiki/Ruslan_L._Stratonovich),[[6]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-Stratonovich1960-6) who was the first to describe the [forward-backward procedure](https://en.wikipedia.org/wiki/Forward%E2%80%93backward_algorithm).

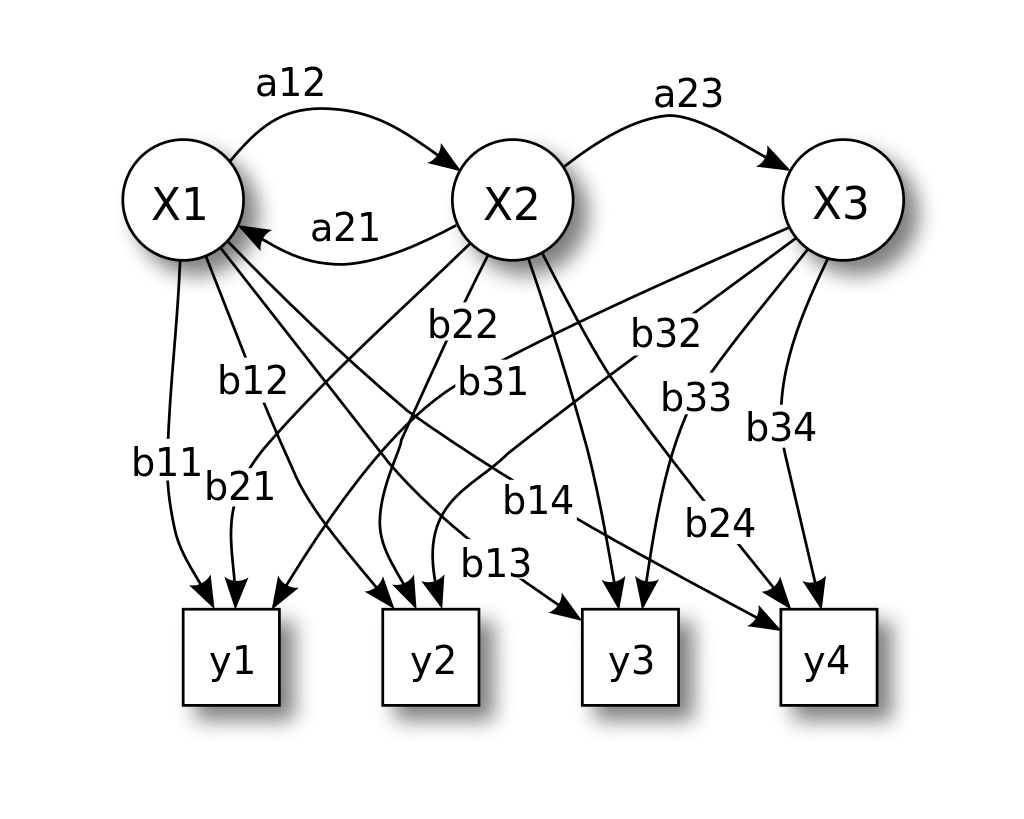
In simpler [Markov models](https://en.wikipedia.org/wiki/Markov_model) (like a [Markov chain](https://en.wikipedia.org/wiki/Markov_chain)), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters, while in the hidden Markov model, the state is not directly visible, but the output (in the form of data or "token" in the following), dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by an HMM gives some information about the sequence of states; this is also known as pattern theory, a topic of [grammar induction](https://en.wikipedia.org/wiki/Grammar_induction).

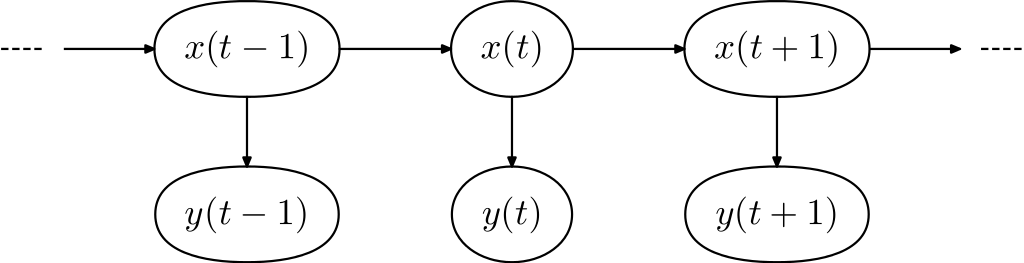
The adjective hidden refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a hidden Markov model even if these parameters are known exactly.

Hidden Markov models are especially known for their application in [reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) and [temporal](https://en.wikipedia.org/wiki/Time) [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition) such as [speech](https://en.wikipedia.org/wiki/Speech_recognition), [handwriting](https://en.wikipedia.org/wiki/Handwriting_recognition), [gesture recognition](https://en.wikipedia.org/wiki/Gesture_recognition),[[7]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-7)[part-of-speech tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging), musical score following,[[8]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-8) [partial discharges](https://en.wikipedia.org/wiki/Partial_discharge)[[9]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-9) and [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics).[[10]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-10)

A hidden Markov model can be considered a generalization of a [mixture model](https://en.wikipedia.org/wiki/Mixture_model) where the hidden variables (or [latent variables](https://en.wikipedia.org/wiki/Latent_variables)), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Recently, hidden Markov models have been generalized to pairwise Markov models and triplet Markov models which allow consideration of more complex [data structures](https://en.wikipedia.org/wiki/Data_structure)[[11]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-TMMEV-11)[[12]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-JASP-12) and the modeling of nonstationary data.[[13]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-TSP-13)[[14]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-SPL-14)

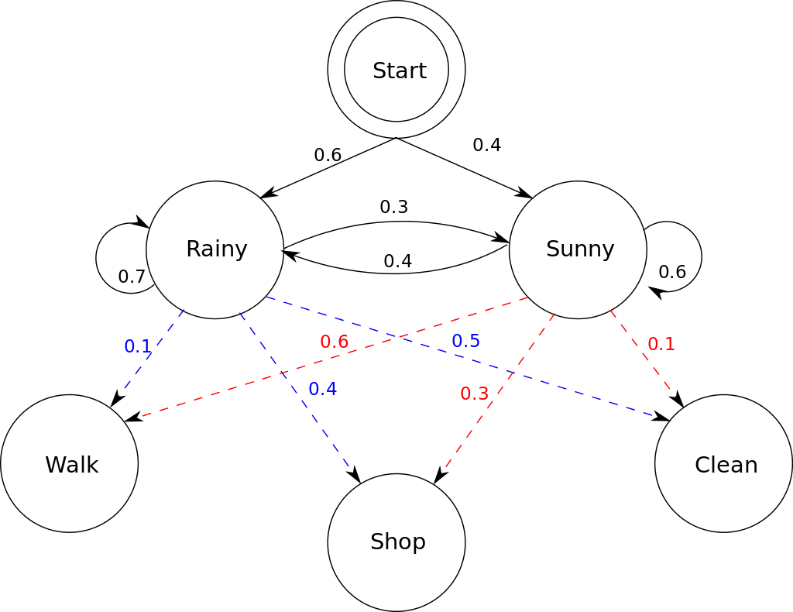
In its discrete form, a hidden Markov process can be visualized as a generalization of the [Urn problem](https://en.wikipedia.org/wiki/Urn_problem) with replacement (where each item from the urn is returned to the original urn before the next step).[[15]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-15) Consider this example: in a room that is not visible to an observer there is a genie. The room contains urns X1, X2, X3, … each of which contains a known mix of balls, each ball labeled y1, y2, y3, … . The genie chooses an urn in that room and randomly draws a ball from that urn. It then puts the ball onto a conveyor belt, where the observer can observe the sequence of the balls but not the sequence of urns from which they were drawn. The genie has some procedure to choose urns; the choice of the urn for the n-th ball depends only upon a random number and the choice of the urn for the (n−1)-th ball. The choice of urn does not directly depend on the urns chosen before this single previous urn; therefore, this is called a [Markov process](https://en.wikipedia.org/wiki/Markov_process). It can be described by the upper part of Figure 1.

The Markov process itself cannot be observed, only the sequence of labeled balls, thus this arrangement is called a "hidden Markov process". This is illustrated by the lower part of the diagram shown in Figure 1, where one can see that balls y1, y2, y3, y4 can be drawn at each state. Even if the observer knows the composition of the urns and has just observed a sequence of three balls, e.g. y1, y2 and y3 on the conveyor belt, the observer still cannot be sure which urn (i.e., at which state) the genie has drawn the third ball from. However, the observer can work out other information, such as the likelihood that the third ball came from each of the urns. 



Learning

The parameter learning task in HMMs is to find, given an output sequence or a set of such sequences, the best set of state transition and emission probabilities. The task is usually to derive the [maximum likelihood](https://en.wikipedia.org/wiki/Maximum_likelihood) estimate of the parameters of the HMM given the set of output sequences. No tractable algorithm is known for solving this problem exactly, but a local maximum likelihood can be derived efficiently using the [Baum–Welch algorithm](https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm) or the Baldi–Chauvin algorithm. The [Baum–Welch algorithm](https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm) is a special case of the [expectation-maximization algorithm](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm). If the HMMs are used for time series prediction, more sophisticated Bayesian inference methods, like [Markov chain Monte Carlo](https://en.wikipedia.org/wiki/Markov_chain_Monte_Carlo)(MCMC) sampling are proven to be favorable over finding a single maximum likelihood model both in terms of accuracy and stability.[[17]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-17)Since MCMC imposes significant computational burden, in cases where computational scalability is also of interest, one may alternatively resort to variational approximations to Bayesian inference, e.g.[[18]](https://en.wikipedia.org/wiki/Hidden_Markov_model#cite_note-18) Indeed, approximate variational inference offers computational efficiency comparable to expectation-maximization, while yielding an accuracy profile only slightly inferior to exact MCMC-type Bayesian inference.



**Deep learning**

Deep learning(also known as deep structured learning or hierarchical learning) is part of a broader family of [machine learning](https://en.wikipedia.org/wiki/Machine_learning)methods based on artificial neural networks. Learning can be [supervised](https://en.wikipedia.org/wiki/Supervised_learning), [semi-supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) or [unsupervised](https://en.wikipedia.org/wiki/Unsupervised_learning).[[1]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIO2012-1)[[2]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-SCHIDHUB-2)[[3]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-NatureBengio-3)

Deep learning architectures such as [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network), [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_networks) have been applied to fields including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Automatic_speech_recognition), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), audio recognition, social network filtering, [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design](https://en.wikipedia.org/wiki/Drug_design), medical image analysis, material inspection and [board game](https://en.wikipedia.org/wiki/Board_game)programs, where they have produced results comparable to and in some cases superior to human experts.[[4]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-:9-4)[[5]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-krizhevsky2012-5)[[6]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-6)

Artificial Neural Networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological [brains](https://en.wikipedia.org/wiki/Brain). Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.[[7]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-7)[[8]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-8)[[9]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-9)

Most modern deep learning models are based on an artificial neural networks, specifically, [Convolutional Neural Networks](https://en.wikipedia.org/wiki/Convolutional_Neural_Network)(CNN)s, although they can also include [propositional formulas](https://en.wikipedia.org/wiki/Propositional_formula) or latent variables organized layer-wise in deep [generative models](https://en.wikipedia.org/wiki/Generative_model) such as the nodes in [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network) and deep [Boltzmann machines](https://en.wikipedia.org/wiki/Boltzmann_machine).

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own. (Of course, this does not completely obviate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.)[[1]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIO2012-1)[[12]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-12)

The "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a [feedforward neural network](https://en.wikipedia.org/wiki/Feedforward_neural_network), the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_network), in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.[[2]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-SCHIDHUB-2) No universally agreed upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth > 2. CAP of depth 2 has been shown to be a universal approximator in the sense that it can emulate any function.[[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Beyond that more layers do not add to the function approximator ability of the network. Deep models (CAP > 2) are able to extract better features than shallow models and hence, extra layers help in learning features.

Deep learning architectures are often constructed with a [greedy](https://en.wikipedia.org/wiki/Greedy_algorithm) layer-by-layer method.[[clarification needed](https://en.wikipedia.org/wiki/Wikipedia:Please_clarify)][[further explanation needed](https://en.wikipedia.org/wiki/Wikipedia:Please_clarify)][[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Deep learning helps to disentangle these abstractions and pick out which features improve performance.[[1]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIO2012-1)

For [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) tasks, deep learning methods obviate [feature engineering](https://en.wikipedia.org/wiki/Feature_engineering), by translating the data into compact intermediate representations akin to [principal components](https://en.wikipedia.org/wiki/Principal_Component_Analysis), and derive layered structures that remove redundancy in representation.

Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors[[13]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-scholarpedia-13)and[deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network).[[1]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIO2012-1)[[14]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-SCHOLARDBNS-14)

**附录二 中文翻译**

**梅尔频率倒谱系数**

在声音处理领域中，梅尔频率倒谱(Mel-Frequency Cepstrum)是基于声音频率的非线性[梅尔刻度](https://zh.wikipedia.org/wiki/%E6%A2%85%E5%B0%94%E5%88%BB%E5%BA%A6" \o "梅尔刻度)(mel scale)的[对数](https://zh.wikipedia.org/wiki/%E5%B0%8D%E6%95%B8" \o "对数)能量频谱的线性变换。

梅尔频率倒谱系数 (Mel-Frequency Cepstral Coefficients，MFCCs)就是组成梅尔频率倒谱的系数。它衍生自音讯片段的[倒频谱](https://zh.wikipedia.org/wiki/%E5%80%92%E9%A0%BB%E8%AD%9C)(cepstrum)。倒谱和梅尔频率倒谱的区别在于，梅尔频率倒谱的频带划分是在[梅尔刻度](https://zh.wikipedia.org/wiki/%E6%A2%85%E5%B0%94%E5%88%BB%E5%BA%A6" \o "梅尔刻度)上等距划分的，它比用于正常的对数[倒频谱](https://zh.wikipedia.org/wiki/%E5%80%92%E9%A0%BB%E8%AD%9C)中的线性间隔的频带更能近似人类的听觉系统。 这样的非线性表示，可以在多个领域中使声音信号有更好的表示。例如在[音讯压缩](https://zh.wikipedia.org/wiki/%E9%9F%B3%E8%A8%8A%E5%A3%93%E7%B8%AE" \o "音讯压缩)中。

梅尔频率倒谱系数（MFCC）广泛被应用于[语音识别](https://zh.wikipedia.org/wiki/%E8%AA%9E%E9%9F%B3%E8%AD%98%E5%88%A5" \o "语音识别)的功能。他们由Davis和Mermelstein在1980年代提出，并在其后持续是最先进的技术之一。在MFCC之前，线性预测系数（LPCS）和线性预测倒谱系数（LPCCs）是自动语音识别的的主流方法。

MFCC通常有以下之过程:

1. 将一段语音信号分解为多个[讯框](https://zh.wikipedia.org/wiki/%E8%A8%8A%E6%A1%86" \o "讯框)。
2. 将语音信号[预强化](https://zh.wikipedia.org/w/index.php?title=%E9%A0%90%E5%BC%B7%E5%8C%96&action=edit&redlink=1" \o "预强化（页面不存在）)，通过一个[高通](https://zh.wikipedia.org/wiki/%E9%AB%98%E9%80%9A)[滤波器](https://zh.wikipedia.org/wiki/%E6%BF%BE%E6%B3%A2%E5%99%A8)。
3. 进行[傅立叶变换](https://zh.wikipedia.org/wiki/%E5%82%85%E7%AB%8B%E5%8F%B6%E5%8F%98%E6%8D%A2" \o "傅立叶变换)，将信号变换至频域。
4. 将每个[讯框](https://zh.wikipedia.org/wiki/%E8%A8%8A%E6%A1%86" \o "讯框)获得的频谱通过梅尔滤波器(三角重叠窗口)，得到[梅尔刻度](https://zh.wikipedia.org/wiki/%E6%A2%85%E5%B0%94%E5%88%BB%E5%BA%A6" \o "梅尔刻度)。
5. 在每个梅尔刻度上提取[对数](https://zh.wikipedia.org/wiki/%E5%B0%8D%E6%95%B8" \o "对数)能量。
6. 对上面获得的结果进行[离散傅里叶反变换](https://zh.wikipedia.org/w/index.php?title=%E7%A6%BB%E6%95%A3%E5%82%85%E7%AB%8B%E8%91%89%E5%8F%8D%E5%8F%98%E6%8D%A2&action=edit&redlink=1" \o "离散傅里叶反变换（页面不存在）)，变换到[倒频谱](https://zh.wikipedia.org/wiki/%E5%80%92%E9%A0%BB%E8%AD%9C)域。
7. MFCC就是这个[倒频谱](https://zh.wikipedia.org/wiki/%E5%80%92%E9%A0%BB%E8%AD%9C" \o "倒频谱)图的幅度(amplitudes)。一般使用12个系数，与讯框能量叠加得13维的系数。

MFCC主要作为[语音识别](https://zh.wikipedia.org/wiki/%E8%AF%AD%E9%9F%B3%E8%AF%86%E5%88%AB" \o "语音识别)系统中的特征，这样的系统可以自动识别语音中的数字内容。MFCC同样也用于[说话人识别](https://zh.wikipedia.org/w/index.php?title=%E8%AF%B4%E8%AF%9D%E4%BA%BA%E8%AF%86%E5%88%AB&action=edit&redlink=1)，该技术尝试通过语音该鉴别说话人。

MFCC也被用于[语音信息检索](https://zh.wikipedia.org/w/index.php?title=%E8%AF%AD%E9%9F%B3%E4%BF%A1%E6%81%AF%E6%A3%80%E7%B4%A2&action=edit&redlink=1)领域，如流派分类(genre classification)、音频相似性计算等。

MFCC特征在加性噪声的情况下并不稳定，因此在语音识别系统中通常要对其进行归一化处理(normalise)以降低噪声的影响。一些研究人员对MFCC算法进行修改以提升其强健性，如在进行DCT之前将log-mel-amplitudes提升到一个合适的能量(2到3之间)，以此来降低低能量成分的影响.[[3]](https://zh.wikipedia.org/wiki/%E6%A2%85%E5%B0%94%E9%A2%91%E7%8E%87%E5%80%92%E8%B0%B1%E7%B3%BB%E6%95%B0#cite_note-10)

**隐马尔可夫模型**

隐马尔可夫模型（Hidden Markov Model；[缩写](https://zh.wikipedia.org/wiki/%E7%B8%AE%E5%AF%AB" \o "缩写)：HMM）或称作隐性马尔可夫模型，是[统计](https://zh.wikipedia.org/wiki/%E7%BB%9F%E8%AE%A1)[模型](https://zh.wikipedia.org/wiki/%E6%A8%A1%E5%9E%8B)，它用来描述一个含有隐含未知参数的[马尔可夫过程](https://zh.wikipedia.org/wiki/%E9%A9%AC%E5%B0%94%E5%8F%AF%E5%A4%AB%E8%BF%87%E7%A8%8B)。其难点是从可观察的参数中确定该过程的隐含参数。然后利用这些参数来作进一步的分析，例如[模式识别](https://zh.wikipedia.org/wiki/%E6%A8%A1%E5%BC%8F%E8%AF%86%E5%88%AB)。

在正常的马尔可夫模型中，状态对于观察者来说是直接可见的。这样状态的转换概率便是全部的参数。而在隐马尔可夫模型中，状态并不是直接可见的，但受状态影响的某些变量则是可见的。每一个状态在可能输出的符号上都有一[概率分布](https://zh.wikipedia.org/wiki/%E6%A6%82%E7%8E%87%E5%88%86%E5%B8%83)。因此输出符号的序列能够透露出状态序列的一些信息。

隐马尔可夫模型尤其以其在强化学习和时间模式识别中的应用而闻名，例如语音，手写，手势识别，词性标注，乐谱跟随，局部放电]和生物信息学。

隐马尔可夫模型可以被认为是混合模型的推广，其中控制要为每个观察选择的混合成分的隐藏变量（或潜在变量）通过马尔可夫过程而不是彼此独立相关。 最近，隐马尔可夫模型已经推广到成对马尔可夫模型和三重马尔可夫模型，这些模型允许考虑更复杂的数据结构[和非平稳数据的建模。

学习过程

HMM中的参数学习任务是在给定输出序列或一组这样的序列的情况下找到状态转换和发射概率的最佳集合。该任务通常是在给定输出序列集的情况下导出HMM的参数的最大似然估计。没有可处理的算法来准确地解决这个问题，但是可以使用Baum-Welch算法或Baldi-Chauvin算法有效地导出局部最大似然。 Baum-Welch算法是期望最大化算法的特例。如果将HMM用于时间序列预测，则更准确的贝叶斯推理方法，如马尔可夫链蒙特卡罗（MCMC）采样被证明有利于在准确性和稳定性方面找到单个最大似然模型。由于MCMC施加了显着的计算负担，在计算可扩展性也是感兴趣的情况下，可以替代地采用贝叶斯推断的变分近似。实际上，近似变分推断提供了与期望最大化相当的计算效率，同时产生的精度曲线仅略微低于精确的MCMC类型贝叶斯推断。

**深度学习**

深度学习（也称为深层结构学习或分层学习）是基于人工神经网络的更广泛的机器学习方法系列的一部分。学习可以是监督，半监督或无监督。深度学习架构，如深度神经网络，深度置信网络，递归神经网络和卷积神经网络，已应用于计算机视觉，语音识别，自然语言处理，音频识别，社交网络过滤，机器翻译，生物信息学，药物设计等领域。 ，医学图像分析，材料检查和棋盘游戏程序，它们产生的结果可与人类专家相媲美，在某些情况下优于人类。

人工神经网络（ANN）受到生物系统中信息处理和分布式通信节点的启发。人工神经网络与生物大脑存在各种差异。具体而言，神经网络往往是静态的和象征性的，而大多数生物体的生物大脑是动态的和类似的。

大多数现代深度学习模型基于人工神经网络，特别是卷积神经网络（CNN），尽管它们还可以包括在深度生成模型中逐层组织的命题公式或潜在变量，例如深信念网络中的节点和深Boltzmann机器。

在深度学习中，每个级别都学习将其输入数据转换为更抽象和复合的表示。在图像识别应用中，原始输入可以是像素矩阵;第一个表示层可以抽象像素并编码边缘;第二层可以组成并编码边缘的排列;第三层可以编码鼻子和眼睛;并且第四层可以识别出图像包含面部。重要的是，深度学习过程可以了解哪些特征可以最佳地放置在哪个级别上。

“深度学习”中的“深度”是指数据转换的层数。更确切地说，深度学习系统具有（CAP）深度。 CAP是从输入到输出的转换链。 CAP描述了输入和输出之间潜在的因果关系。对于前馈神经网络，CAP的深度是网络的深度，是隐藏层的数量加1（因为输出层也是参数化的）。对于递归神经网络，其中信号可能不止一次传播通过一层，CAP深度可能是无限的。除此之外，更多的层不会增加网络的功能近似能力。深度模型（CAP> 2）能够提取比浅模型更好的特征，因此，额外的层有助于学习特征。

深度学习架构通常采用贪婪的逐层方法构建。并挑选出哪些功能可以提高性能。

对于有监督的学习任务，深度学习方法通​​过将数据转换为类似于主成分的紧凑中间表示来消除特征工程，并导出去除表示中的冗余的分层结构。

1. 参考文献

   Schmidhuber, J. (2015). "Deep Learning in Neural Networks: AnOverview". Neural Networks. 61: 85–117. [arXiv](https://en.wikipedia.org/wiki/ArXiv):[1404.7828](https://arxiv.org/abs/1404.7828). [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1016/j.neunet.2014.09.003](https://doi.org/10.1016%2Fj.neunet.2014.09.003). [PMID](https://en.wikipedia.org/wiki/PubMed_Identifier) [25462637](https://www.ncbi.nlm.nih.gov/pubmed/25462637). [↑](#endnote-ref-1)
2. *Mikolov, T.; et al. (2010).*["Recurrent neural network based language model"](http://www.fit.vutbr.cz/research/groups/speech/servite/2010/rnnlm_mikolov.pdf)(PDF)*. Interspeech.* [↑](#endnote-ref-2)
3. *Sahidullah, Md.; Saha, Goutam (May 2012). "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition". Speech Communication. 54 (4): 543–565.*[doi](https://en.wikipedia.org/wiki/Digital_object_identifier)*:*[10.1016/j.specom.2011.11.004](https://doi.org/10.1016%2Fj.specom.2011.11.004)*.* [↑](#endnote-ref-3)
4. [Lawrence R. Rabiner](https://zh.wikipedia.org/w/index.php?title=Lawrence_Rabiner&action=edit&redlink=1), [A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition](https://web.archive.org/web/20070209162249/http:/www.caip.rutgers.edu/~lrr/Reprints/tutorial%20on%20hmm%20and%20applications.pdf). Proceedings of the [IEEE](https://zh.wikipedia.org/wiki/IEEE), 77 (2), p. 257–286, February 1989. [↑](#endnote-ref-4)
5. Christopher M. Bishop etc., Pattern Recognition and Machine Learning, Springer, 2006 [↑](#endnote-ref-5)
6. 李航, 统计学习方法, 清华大学出版社, 2012 [↑](#endnote-ref-6)
7. Bush, Steve. [Dongle computer lets kids discover programming on a TV](http://www.electronicsweekly.com/Articles/2011/05/25/51129/Dongle-computer-lets-kids-discover-programming-on-a.htm). [Electronics Weekly](https://zh.wikipedia.org/w/index.php?title=Electronics_Weekly&action=edit&redlink=1). 2011-05-25 [↑](#endnote-ref-7)
8. van Rossum, Guido (1993). "An Introduction to Python for UNIX/C Programmers". Proceedings of the NLUUG Najaarsconferentie (Dutch UNIX Users Group). [CiteSeerX](https://en.wikipedia.org/wiki/CiteSeerX) [10.1.1.38.2023](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.2023). even though the design of C is far from ideal, its influence on Python is considerable. [↑](#endnote-ref-8)
9. [Pearson, K.](https://zh.wikipedia.org/w/index.php?title=Karl_Pearson&action=edit&redlink=1) [On Lines and Planes of Closest Fit to Systems of Points in Space](https://web.archive.org/web/20131020092552/http:/stat.smmu.edu.cn/history/pearson1901.pdf)(PDF). Philosophical Magazine. 1901, 2 (6): 559–572.  [↑](#endnote-ref-9)
10. [Abdi. H., & Williams, L.J.](https://zh.wikipedia.org/w/index.php?title=AbdiWilliams&action=edit&redlink=1) Principal component analysis.. Wiley Interdisciplinary Reviews: Computational Statistics,. 2010, 2: 433–459. [↑](#endnote-ref-10)
11. Min Xu; 等. HMM-based audio keyword generation. (编) Kiyoharu Aizawa, Yuichi Nakamura, Shin'ichi Satoh. [Advances in Multimedia Information Processing - PCM 2004: 5th Pacific Rim Conference on Multimedia](http://cemnet.ntu.edu.sg/home/asltchia/publication/AudioAnalysisUnderstanding/Conference/HMM-Based%20Audio%20Keyword%20Generation.pdf) (PDF). Springer. 2004. [ISBN 3-540-23985-5](https://zh.wikipedia.org/wiki/Special:%E7%BD%91%E7%BB%9C%E4%B9%A6%E6%BA%90/3-540-23985-5) [↑](#endnote-ref-11)
12. Sahidullah, Md.; Saha, Goutam. [Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition](http://www.sciencedirect.com/science/article/pii/S0167639311001622). Speech Communication. May 2012, 54 (4): 543–565. [↑](#endnote-ref-12)
13. V. Tyagi and C. Wellekens (2005), [On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition](http://dx.doi.org/10.1109/ICASSP.2005.1415167) , in Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on, vol. 1, pp. 529–532. [↑](#endnote-ref-13)
14. HMM, 2019,Wikipedia [↑](#endnote-ref-14)
15. Artificial Intelligence: A modern approach, 3rd edition, 2010, Stuart Russell and Peter Norvig. [↑](#endnote-ref-15)
16. Pattern Recognition and Machine Learning, 2006 ,Christopher M. Bishop. [↑](#endnote-ref-16)
17. 于海涛，网络上的信息传播模型与算法研究，青岛大学报2013 [↑](#endnote-ref-17)
18. 蔡宇，采用子带谱减法的语音增强，计算机应用，2015 [↑](#endnote-ref-18)