

基于 MFCC 和 DTW 的频域语音识别问题

任泽华, 姬文虎, 程敏敏, 刘雨芊, 秦祯

(西安交通大学自动化学院 西安 710049)

摘要:

本次实验面向孤立字语音识别基本任务, 由频域角度进行探讨、分析、解决。 本文将通过能识别数字 0~9 的语音识别系统的实现过程详细阐述基于 MFCC、DTW 的特定孤立词识别的相关原理和关键技术。

在信号采集部分, 首先通过编程方式运用 `audiorecorder` 函数进行了语音信号的采集并建立了班级语音信号库。在预处理部分, 我们对语音信号进行了批量处理: 进行预加重滤波处理; 运用汉明窗函数进行分帧加窗处理, 短时域特性分析并对该特定时间段的语音信号进行离散时间傅里叶 (DTF) 变换, 得到相应的频谱结果; 求取 Mel 倒谱系数和 MFCC 系数, 运用模板匹配法、动态时间规整 (DTW) 进行分类识别。

通过实验对比、量化分析及改进, 本系统的语音识别平均准确率大致为 48%。

关键词: DTW; MFCC; 频域分析; 孤立词识别

Speech Recognition in Frequency Domain Based on DTW and MFCC

Abstract:

This experiment is aimed at the basic task of isolated word speech recognition, which is discussed, analyzed and solved from the perspective of frequency domain. In this paper, the principle and key technology of specific isolated word recognition based on MFCC and DTW are described in detail through the implementation of speech recognition system which can recognize digit 0-9

In the part of signal acquisition, firstly, audio recorder function is used to acquire the speech signal and set up the class speech signal database. In the preprocessing part, we deal with the speech signal in batches: pre weighting filtering; frame and window processing with Hamming window function, short-time-domain characteristic analysis and discrete-time Fourier (DTF) transformation for the speech signal in this specific period to get the corresponding spectrum results; calculate Mel cepstrum coefficient and MFCC coefficient, and use template matching method and dynamic time DTW is used for classification and recognition.

Through experiment comparison, quantitative analysis and improvement, the average accuracy of the system is about 48%.

Key words: DTW; MFCC; Frequency Domain Analysis; Isolated Word Recognition

1 问题分析

1.1 问题背景

语音信号处理是一个新兴的交叉学科, 是语音和数字信号处理两个学科的结合产物。

语音信号处理目的是得到一些语音特征参数, 以便高效的传输或存储, 或通过某种处理以达到特定目的, 如语音合成, 辨识出讲话者、识别出讲话的内容等。随着现代科学技术和计算机技术的发展, 除

了人与人的自然语言的沟通, 人机对话和智能机领域也开始使用语言。这些人造的语言拥有词汇, 语法, 语法结构和语义内容等。自动语音识别技术起源于 20 世纪 50 年代, 最早的商用系统是 IBM 在 90 年代推出的 `ViaVoice`。经过半个多世纪的发展, 语音识别技术目前已日趋成熟并成功应用到人们的日常生活之中, 如苹果手机的 `Siri` 体验、科大讯飞的迅速崛起等。

根据识别的对象不同, 语音识别任务大体可分为 3 类, 即孤立词识别 (`isolated`

word recognition), 关键词识别 (keyword spotting) 和连续语音识别。其中, 孤立词识别的任务是识别事先已知的孤立的词, 如“开机”、“关机”等。本题即属于孤立词识别。语音是一种典型的、易于获取的一维时序信号, 语音信号处理及识别技术也是数字信号处理课程绝佳的实践途径。时间序列分析、快速傅里叶变换、滤波器设计等多项数字信号处理的教学内容在语音识别核心技术中均占有重要地位。本次实验即面向语音识别基本任务, 需熟悉语音数据的基本形式及特点, 理解并应用离散时间信号的基本分析、处理方法, 理解语音识别技术的概貌。

1.2 问题提出

- 1) 采集“0” - “9”语音信号并构建语音库
- 2) 进行语音信号预处理: 预加重、分帧与加窗
- 3) 快速傅里叶变换, 把时域信息转变为频域信息
- 4) 计算 Mel 滤波器能量、经离散余弦变换 (DCT) 得到 MFCC 系数
- 5) 利用动态时间规整 (DTW) 进行模式匹配实现语音识别
- 6) 编程实现可视化界面, 实现在线语音识别

2 问题解决

2.1 采集“0” - “9”语音信号

$$(1) \text{矩形窗 } w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases}$$

$$(2) \text{汉明窗 } w(n) = \begin{cases} 0.54 - 0.46 \cos \left[\frac{2\pi n}{N-1} \right], & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases}$$

$$(3) \text{海宁窗 } w(n) = \begin{cases} 0.5 \left(1 - \cos \left[\frac{2\pi n}{N-1} \right] \right), & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases}$$

本文在 MATLAB 环境中使用 audiorecorder 函数录制语音信号, 采集“0”、“1”...“9”这 10 个语音的 wav 文件, 每个类别应采集 10 组以上的样本。为获得更多数据集, 本班建立了班级语音库。

2.2 语音信号预处理

2.2.1 预加重

由于语音的高频分量对于识别具有特别的意义, 然而高频分量又通常能量较弱, 因此应对原始语音信号首先进行预加重滤波处理, 再进行后续的频谱计算。我们先对原先在实验一采集到的 0 到 9 的 100 条原始的语音数据信号进行预加重滤波处理。

在这里, 我们进行预加重处理时采用的滤波器的数学模型为:

$$H(z) = 1 - az^{-1}$$

其中, a 取 0.97。

2.2.2 分帧与加窗

贯穿于语音分析全过程的是“短时分析技术”。因为, 语音信号从整体来看其特性及表征其本质特征的参数均是随时间而变化的, 所以它是一个非平稳态过程, 不能用处理平稳信号的数字处理技术对其进行处理。但是, 由于不同的语音是由人的口腔肌肉运动构成声道某种形状而产生的响应, 而这种口腔肌肉运动相对于语音频率来说是非常缓慢的, 所以从另一个方面看, 虽然语音信号具有时变特性, 但是在一个短时间范围内 (一般认为在 10-30 ms 的时间内), 其特性基本保持不变即相对稳定, 因而可以将其看做是一个准稳态过程, 即语音信号具有短时平稳性。所以任何语音信号的分析 and 处理必须建立在“短时”的基础上, 即进行“短时分析”, 将语音信号分为一段一段来分析其特征参数, 其中每一段称为一“帧”, 帧长一般取 10-30 ms。这样, 对于整体的语音信号来讲, 分析出的是由每一帧特征参数组成的特征参数时间序列。分帧是用可移动的有限长度窗口进行加权的方法来实现的, 这就是用一定的窗函数 $w(n)$ 来乘 $s(n)$, 从而形成加窗语音信号 $S_W = s(n) * w(n)$ 。窗函数 $w(n)$ 的选择 (形状和长度), 对于短时分析参数的特性影响很大。为此应选择合适的窗口, 使其短时参数很好地反映语音信号的特性变化。在语音信号数字处理中常用的窗函数有三种:

本次实验，我们选取汉明窗作为加窗函数。

2.3 基于 FFT 的离散傅里叶变换

每一帧时间内的信号的离散傅里叶变换公式如下：

$$X_n(e^{j\omega}) = \sum_{m=0}^{N-1} x_n(m) e^{-j\omega m}$$

这里，我们定义 $\omega = 2k\pi/N$ ，可以得到离散的短时傅里叶变换（DFT），实质上是 $X_n(e^{j\omega})$ 在频域的取样，其式子如下所示：

$$X_n\left(e^{j\frac{2\pi k}{N}}\right) = X_n(k) = \sum_{m=0}^{N-1} x_n(m) e^{-j\frac{2\pi km}{N}}$$

为了降低该离散傅里叶变换公式计算时所带来的庞大计算量，提高语音信号进行傅里叶变换的效率，我们采用了快速傅里叶（FFT）算法进行实现。快速傅里叶变换（FFT）充分利用了离散时间信号频谱的周期性和对称性，可以大量减少了离散傅里叶中的复数运算所带来的庞大计算量。但该算法在 Matlab 的调用中，需要我们将离散傅里叶变换的点数设置为 $2L$ （ L 为整数），且要满足 $2L > N > 2L-1$ 。在实验中，我们取 $N=160$ ，则 $2L=256$ ，对每一帧进行快速傅里叶变换得到每一帧的频谱结果。

2.4 谱线能量的计算

对快速傅里叶变换（FFT）所得到的结果，根据如下数据谱线能量公式，计算谱线能量：

$$E(i, k) = [X_i(k)]^2$$

其中， i 表示第 i 帧， k 表示频域中第 k 条谱线。

2.5 Mel 频率倒谱系数

梅尔（Mel）频率谱是在已知信号频谱的基础上，基于人类听觉系统的感知特性，设计出的一种频谱分组方式。通过计

算 Mel 频谱，将得到比原始傅里叶频谱更加具有区分性的频域紧凑表达，从而有利于精确地实现识别任务。Mel 频率尺度的值大体上对应于实际频率的对数分布关系，与实际频率可用下式近似表示：

$$\text{Mel}(f) = 2595 \lg\left(1 + \frac{f}{700}\right)$$

式中， f 为频率，单位为 Hz。

语音频率可以划分成一系列三角形的滤波器序列，即 Mel 滤波器组，如图 2.1 所示。

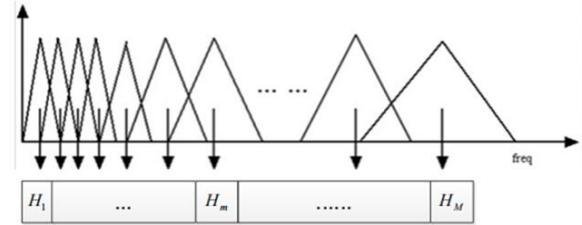


图 2.1 Mel 频率尺度滤波器组

设划分的带通滤波器为 $H_m(k)$, $0 \leq m \leq M$, M 为滤波器的个数。每个滤波器具有三角形滤波特性，其中心频率为 $f(m)$ ，在 Mel 频率范围内，这些滤波器是等带宽的（如图 2.2）。

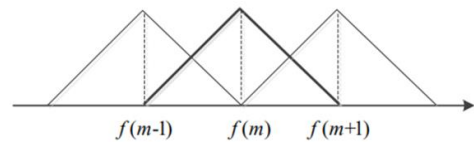


图 2.2 相邻三角滤波器之间的关系

每个带通滤波器的传递函数为：

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

其中， $\sum_{m=1}^{M-1} H_m(k) = 1$

Mel 滤波器的中心频率 $f(m)$ 定义为：

$$f(m) = \frac{N}{f_s} F_{\text{Mel}}^{-1}\left(F_{\text{Mel}}(f_i) + m \frac{F_{\text{Mel}}(f_h) - F_{\text{Mel}}(f_l)}{M+1}\right)$$

其中, f_h 和 f_l 分别是滤波器组的最高频率和最低频率, f_s 为采样频率, 单位为 Hz。 M 是滤波器组的数目, N 为 FFT 变换的点数

$$F_{Mel}^{-1}(b) = 700(e^{\frac{b}{1125}} - 1)$$

2.6 语音特征参数 MFCC

经离散余弦变换(DCT)得到 MFCC 系数:

$$mfcc(i, n) = \sqrt{\frac{2}{M} \sum_{m=0}^{M-1} \ln[S(i, m)] \cos\left[\frac{\pi n(2m-1)}{2M}\right]}$$

式中, $S(i, m)$ 是求出的 Mel 滤波器能量, m 是指第 m 个 Mel 滤波器(共有 $M=24$ 个); i 是指第 i 帧, n 是 MFCC 系数阶数, 在实验中, 我们选取了原始数据以及一阶差分, 所以取 $n=24$ 。

最终, 一段语音信号的每一帧都会计算出一个 MFCC 系数结果 $mfcc(i, n)$, 由此 MFCC 系数所构成的数组, 就是我们所提取的该段语音信号在频域上的特征。

2.7 模板匹配法语音识别

2.7.1 模板匹配法语音识别系统构成

图 3.1 为利用模板匹配法进行语音识别的原理框图。在训练阶段, 用户将词汇表中的每一个词一次说一遍, 并且将其矢量特征时间序列作为模板存入模板库。在识别阶段, 将输入语音的特征矢量时间序列依次与模板库中的每一个模板进行相似度比较, 将相似度最高者作为识别结果输出。

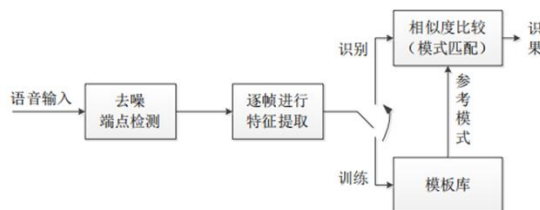


图 2.3 模板匹配法语音系统的原理框图

在特征提取阶段, 本实验选用实验二的 Mel 频率倒谱系数(MFCC)作为识别特征: 在识别阶段, 实验选用动态时间规整(DTW)技术进行模式匹配。

2.7.2 动态时间规整

由于每一个孤立语音信号的时长各不相同, 其计算得到的频谱特征向量长度也将各不相同。然而, 对于一般的模式识别系统而言, 要求待比对的特征向量应具有相同的长度。Dynamic Time Warping(DTW)技术是基于动态规划的思想可以实现不等长特征向量的距离计算, 因此在语音识别中得到了广泛应用。

动态时间规划是一个典型的最优化问题, 它用满足一定条件的动态时间规整函数描述输入模板和参考模板之间的时间对应关系, 求解两模板匹配时累计距离最小所对应的规整函数。假设词库中输入语音的特征矢量序列为:

$$a_1, \dots, a_m, \dots, a_M$$

输入语音的特征矢量序列为:

$$b_1, \dots, b_n, \dots, b_N, M \neq N, \text{ 那么动态}$$

时间规整就是要找到时间规整函数 $m=T(n)$ 。该函数把输入模板的时间轴非线性映射到参考模板的时间轴 m , 并满足下式:

$$D = \min_{T(n)} \sum_{n=1}^N d[n, T(n)]$$

式中, $d[n, T(n)]$ 表示两帧矢量之间的距离; D 是最佳时间路径下两个模板的距离测度。本实验中距离测度可采用欧式距离:

$$d(x, y) = \frac{1}{k} \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

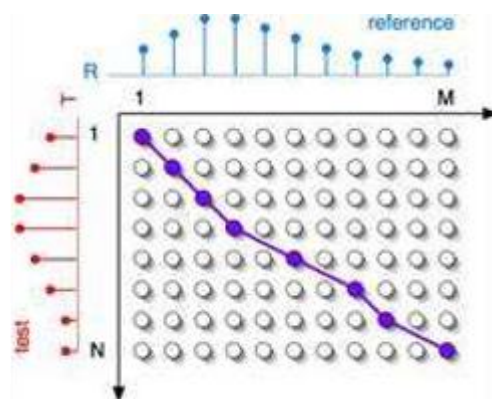


图 2.4 DTW 的算法原理图

由图 2.4 可知,把测试模板的各个帧号 $n=1\sim N$ 在一个二维直角坐标系中的横轴上标出,把参考模板的各帧 $m=1\sim M$ 在纵轴上标出,通过这些表示帧号的整数坐标画出一些横纵线即可形成一个网络,网格中的每一个交叉点表示测试模式中某一帧与训练模式中某一帧的交汇。DTW 算法分两步进行,一是计算两个模式各帧之间的距离,即求出帧匹配距离矩阵,二是在帧匹配距离矩阵中找出一条最佳路径。搜索这条路径的过程可以描述如下:搜索从 $(1,1)$ 点出发,对于局部路径约束,点 (i_n, i_m) 可到达的前一个格点只可能是 (i_n-1, i_m) 、 (i_n, i_m-1) 和 (i_n-1, i_m-1) 。那么一定选择这三个距离中的最小者所对应的点作为其前续格点,这时此路径的累积距离为:

$$D(i_n, i_m) = d(T(i_n), R(i_m)) + \min\{D(i_{n-1}, i_m), D(i_{n-1}, i_{m-1}), D(i_n, i_{m-1})\}$$

这样从 $(1,1)$ 点 $(D(1,1)=0)$ 出发搜索,反复递推,直到 (M, N) 就可以得到最优路径,而且 $D(N, M)$ 就是最佳匹配路径所对应的匹配距离。在进行语音识别时,将测试模板与所有参考模板进行匹配,得到最小匹配距离 所对应语音即为识别结果。

3 问题结果

3.1 可视化界面



图 3.1 可视化界面

可视化界面展示了语音的录入、处理、

识别的过程,从而我们比较完整的实现了数字的识别过程。

3.2 实验改进

3.2.1 设定阈值范围选取识别结果

两数之间的最小距离作为判别标准

在第一次试验中,为减小随机误差,我们通过观察法选取了一个较小的阈值范围 $[0, 2e^{-0.5}]$,当最短距离小于该范围时记为有效。最后,选取处于阈值内个数最多的数值记为识别结果。

经测试计算,该方案识别率为 10%,识别率比较低。

3.2.2 采用最小距离作为判别标准

考虑到方案一的阈值由观察法得到,并不严谨。

我们考虑两数之间的最小距离作为判别标准。即在识别阶段,每次比较时,若待识别数字和其正在比较的数之间欧式距离小于 d ,则更新 d 的值(令 d 等于此欧式距离),不断这样比较并更新 d 的值,计算完毕后将 d 的值对应的数字作为匹配的结果输出。

经测试计算,该方案识别率为 30%,识别率比较低。识别率有显著提高。

3.2.3 采用距离均值作为判别标准

经小组讨论及老师建议,本文采用距离均值作为判别标准进行识别。相比方案二,该方案进一步减小了偶然误差导致的错误匹配结果。

经测试计算,该方案识别率约为 50%,识别率再次显著提高。

3.3 结果与量化分析

本文将通过运用 DTW 及数字信号处理相关原理搭建了能识别数字 0~9 的语音识别系统。经调试改进后平均准确率大致为 50%。

经统计,识别错误情况如下:

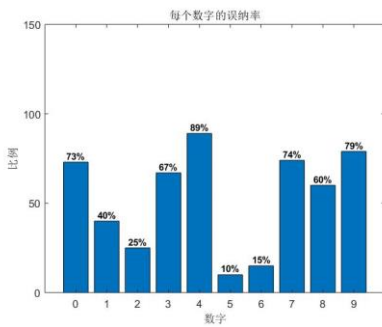


图 3.2 数字 0-9 的错误率

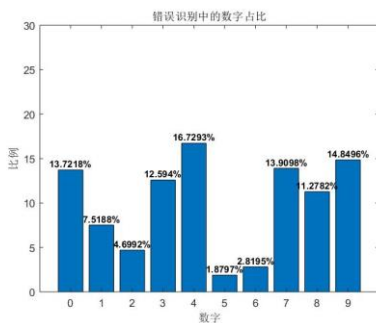


图 3.3 错误识别中每个数字占比情况

经分析,平均准确率仅为 48%,不太理想,可能有以下原因:特征值太多而出现拟合现象;模板长度不一致造成影响;模板数量较少,且代表性不足,不能完整的体现出相应数据的全部特征等等,从而会引起识别率降低。本系统可针对以上原因进一步改进。

4 问题收获与感想

秦祯: 本次语音识别实验很有难度也很有意义,我在课程学习、实验项目和小组合作中收获颇丰。语音识别实验不仅与课程紧密结合,还进一步拓宽了我的知识面,看到信号处理知识如何应用于我们的生活。在实验过程中,也深刻体会到了学习能力、编程能力以及解决问题的重要性。当我们按照实验指导书的思路和步骤完成时,发现识别效果并不太好。对于如何提高识别准确率也常常觉得很难有角度思考。后来,在王芳芳老师的引导建议以及其他优秀小组的展示中,看到了很多不同的思考问题

角度和解决方法。我不仅为大家的项目成果感到惊叹,更欣赏大家思维的严密性和创新性,觉得自己在这些方面需要不断地学习和操练。

程敏敏: 本次实验难度较大但很有意义,让我们对语音识别有了初步的了解和学习,同时我们也不再只是局限于课本,而是将生活中的例子与我们所学的知识联系起来,帮助我们更深刻地理解相关内容,极大地提高了我们的学习兴趣。在实验中,我们对相关算法进行了学习,同时在解决问题的同时也对其他组的想法进行了了解和探讨,对我们的能力有了很大的提升。在老师的指导和帮助下,识别的准确率也得到了极大的提升。在本次实验中我们感受到了语音识别的极大魅力,在实验结束后我们也会更深入地对此进行学习和练习。

姬文虎: 本次语言识别实验很有挑战而且有很大的实用意义,通过这次实验,让我对语言识别的一些常用方法有了一些认识,不局限于课本的算法,要通过这个问题找到合适的算法,而不是把算法往问题中生搬硬套,通过实验一我能感受到同一个问题使用不同算法带来的一些不同的改变,比如时间和正确率之间如何权衡,去到一个相对满意的指标。通过实验二三,我学习到了一个完整的比较简单的单字识别的思路,了解了如何去比较不同的长度的矩阵。深刻认识到编程能力对于实现自己想法的助力,要加强自己的编程能力。在实验别的组的展示中我也看到了他们的对于问题的自己的认识,学习到了一些我没有注意的方法和思路,这让我受益匪浅。需要多多了解他人的看法,来启发自己。通过这次实验我更加深入理解了语言识别的优点和魅力。

刘雨芊: 在进行完整个实验后,我对 MATLAB 软件运用的更加熟练,对语音信号识别的原理和方法有了更深刻的理解。先学习了语音信号的采集和预处理,然后从时域出发对语音信号进行识别,由于识别

的准确度不高，又学习了从频域提取信号特征对信号进行识别。通过几次实验掌握了多个信号识别的方法收获很大。

本次实验进一步培养了我实事求是、严谨认真的作风，我认识到了对自己要严格要求，不能够一知半解，要力求明明白白。在此次的实验中，我最大的体会就是进一步认识到了理论联系实际的重要性。一份耕耘，一份收获。通过这段时间的学习，也让我明白科学的思维方法是多么重要，只有只有掌握科学的方法才能够更高效地完成任任务，总而言之，此次实验让我学到了好多平时在课堂上学不到的东西，提高了我的知识运用能力，增强我的实际操作能力。谢谢老师给我们提供这么好的机会，为我们今后的实践奠定了一个好的基础。本次实验让我学到了很多，也学会到了要怎么样去面对困难，不要对知识一知半截，要有的求实的能力，通过老师的帮助我学到了很多在平时没有注意到、没有深入去理解的知识。通过这次实验我要更加明确自己的优点和不足，更要注重自己在各方面能力的锻炼。最后感谢数字信号处理这门课的老师对我的指导和帮助。

任泽华：这次实验收获很大，就像郑院士说的，我们本科阶段做实验最重要的不是结果，而是享受做实验的过程，为将来的科研积累经验。在整个过程中我们一个团队为了一个目标共同努力，这一点让我印象深刻。在我们的模型识别率仅为 10%，大家没有情绪低落，而是共同分析，一起寻找问题，这也让我很是感动。通过与其他小组的沟通交流和老师的指导，我们意识到了特征参数提取的错误，由于我们录音采用的是 44100Hz 的采样频率，而算法直接才用普遍使用的 8000Hz，这就导致许多频率信息被忽略。在 DTW 进行识别的过程中，由于参数选取的不科学也走了不少弯路。总的说来，这次实验带给我的不仅仅是做出的成果，更多的是一种用科学方法分析问题的能力和一份坚忍不拔的毅力，还有团队之间通力合作的宝贵品质。

参考文献

- [1] 郑南宁.数字信号处理简明教程西安.西安交通大学出版社,2015
- [2] 数字信号处理实验指导书.西安
- [3] 胡广书 . 数字信号处理 [M]. 北京: 清华大学出版社 , 2003
- [4] 奥本海姆.信号与系统.电子工业出版社,2012

代码附录

```
function varargout = main(varargin)
% MAIN MATLAB code for main.fig
%     MAIN, by itself, creates a new
%     MAIN or raises the existing
%     singleton*.
%
%     H = MAIN returns the handle to a
%     new MAIN or the handle to
%     the existing singleton*.
%
%
% MAIN('CALLBACK',hObject,eventData,handl
es,...) calls the local
%     function named CALLBACK in
MAIN.M with the given input arguments.
%
%     MAIN('Property','Value',...) creates
a new MAIN or raises the
%     existing singleton*. Starting from
the left, property value pairs are
%     applied to the GUI before
main_OpeningFcn gets called. An
%     unrecognized property name or
invalid value makes property application
%     stop. All inputs are passed to
main_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools
menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the
response to help main

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State     = struct('gui_Name',
mfilename, ...
                       'gui_Singleton',
gui_Singleton, ...
                       'gui_OpeningFcn',
@main_OpeningFcn, ...
                       'gui_OutputFcn',
```

```

@main_OutputFcn, ...
    'gui_LayoutFcn',
[], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before main is made visible.
function main_OpeningFcn(hObject, ~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main (see VARARGIN)

% Choose default command line output for main
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = main_OutputFcn(~, ~, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, ~, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
file_path = get(hObject, 'String');
handles.file_path = file_path;
guidata(hObject, handles);

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, ~, ~)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
%%选择识别样本
function pushbutton1_Callback(~, ~, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global k fs
[file1,pathname]=uigetfile('*.wav','请选择要识别的样本');%跳出对话框
fname=fullfile(pathname,file1);
[xx,fs]=audioread(fname);
[m,n]=size(xx);
k=xx+eps*ones(m,n);
sound(k,fs);%播放声音

```



```

set(handles.edit3,'string','识别结果');

function pushbutton2_Callback(~, ~,
handles)
% hObject handle to pushbutton1 (see
GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)
global k fs
set(handles.edit3,'string','请说话');
[k,fs]=luyin1();

set(handles.edit3,'string','结果');

% --- Executes on button press in
pushbutton3.
%%开始识别按钮
function pushbutton3_Callback(~, ~,
handles)

% hObject handle to pushbutton3 (see
GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)
global k fs

h = waitbar(0,'正在识别, 请稍等...');
cc=MFCC2(k,fs);

j=shibie(cc);

fprintf('与样本测试距离最小得分即为识
别结果: %d\n', j);
close(h);
set(handles.edit3,'string',j);

function edit3_Callback(~, ~, ~)
% hObject handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of edit3 as text
% str2double(get(hObject,'String'))
returns contents of edit3 as a double

% --- Executes during object creation, after
setting all properties.
function edit3_CreateFcn(hObject, ~, ~)
% hObject handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created
until after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
% See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
pushbutton4.
function pushbutton4_Callback(~, ~, ~)
% hObject handle to pushbutton4 (see
GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and
user data (see GUIDATA)
close

% --- Executes during object creation, after
setting all properties.
function figure1_CreateFcn(hObject,
eventdata, handles)
% hObject handle to figure1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created
until after all CreateFcns called
ha=axes('units','normalized','pos',[0 0 1 1]);
uistack(ha,'down');
ii=imread('timg.jfif');
image(ii);
colormap gray
set(ha,'handlevisibility','off','visible','on');

% --- Executes during object creation, after
setting all properties.
function pushbutton2_CreateFcn(hObject,
eventdata, handles)
% hObject handle to pushbutton2 (see
GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB

```

% handles empty - handles not created
until after all CreateFcns called

```
function getmfcc= MFCC2(x,fs)
%=====
% 提取 MFCC 特征参数
% 无去噪及端点检测
% Input:音频数据 x,采样率 fs
% Output: (N,M)大小的特征参数矩阵
% 其中 N 为分帧个数, M 为特征维度
% 特征参数: M=24 倒谱系数 12 维, 一
% 阶差分 12 维
%=====
% tic
x = double(x);
x = x / max(abs(x));

%常数设置
FrameLen = 256;
FrameInc = 80;

amp1 = 10;
amp2 = 2;
zcr1 = 10;
zcr2 = 5;

maxsilence = 8; % 8*10ms = 80ms
minlen = 15; % 15*10ms = 150ms
status = 0;
count = 0;
silence = 0;

%计算过零率
tmp1 = enframe(x(1:end-1),
hamming(FrameLen), FrameInc);%变汉明窗
tmp2 = enframe(x(2:end),
hamming(FrameLen), FrameInc);%变汉明窗
signs = (tmp1.*tmp2)<0;
diffs = (tmp1 -tmp2)>0.02;
zcr = sum(signs.*diffs, 2);

%计算短时能量
amp = sum(abs(enframe(filter([1 -0.9375], 1,
x), hamming(FrameLen), FrameInc)), 2);%变汉明窗
%调整能量门限
amp1 = min(amp1, max(amp)/4);
amp2 = min(amp2, max(amp)/8);
```

```
%开始端点检测
x1 = 0;
% x2 = 0;
for n=1:length(zcr)
% goto = 0;
switch status
case {0,1} % 0 =
静音, 1 = 可能开始 % 确
信进入语音段
x1 = max(n-count-1,1);
status = 2;
silence = 0;
count = count + 1;
elseif amp(n) > amp2 || ... % 可能
处于语音段
zcr(n) > zcr2
status = 1;
count = count + 1;
else %
静音状态
status = 0;
count = 0;
end
case 2 % 2 =
语音段
if amp(n) > amp2 || ... % 保持
在语音段
zcr(n) > zcr2
count = count + 1;
else %
语音将结束
silence = silence+1;
if silence < maxsilence % 静音还
不够长, 尚未结束
count = count + 1;
elseif count < minlen % 语音
长度太短, 认为是噪声
status = 0;
silence = 0;
count = 0;
else %
语音结束
status = 3;
end
end
case 3
break;
end
end

count = count-silence/2;
x2 = x1 + count - 1;

xx=tmp1(x1:x2,:); %截取有效段
%归一化 mel 滤波器组系数
bank=melbankm(24,256,fs,0,0.5,'m');%Mel
```

```

滤波器的阶数为 24, fft 变换的长度为 256,
采样频率

bank=full(bank);

bank=bank/max(bank(:));%[24*129]

%设定 DCT 系数

for k=1:12

n=0:23;

dctcoef(k,:)=cos((2*n+1)*k*pi/(2*24));

end

%归一化倒谱提升窗口

w=1+6*sin(pi*[1:12]./12);

w=w/max(w);

%计算每帧的 MFCC 参数

for i=1:size(xx,1)

y=xx(i,:);%取一帧数据

s=y'.*hamming(256);

t=abs(fft(s));%fft 快速傅立叶变换 幅度谱

t=t.^2;%能量谱

%对 fft 参数进行 mel 滤波取对数再计算倒谱
c1=dctcoef*log(bank*t(1:129));%对能量谱
滤波及 DCT %t(1:129)对一帧的前 128 个数
(帧移为 128)

c2=c1.*w';%归一化倒谱

%mfcc 参数

m(i,:)=c2';

end

%求取一阶差分系数

dtm=zeros(size(m));

for i=3:size(m,1)-2

dtm(i,:)=-2*m(i-2,:)-m(i-1,:)+m(i+1,:)+2*m(i+
2,:);

end

dtm=dtm/3;

%合并 mfcc 参数和一阶差分 mfcc 参数

ccc=[m dtm];

%去除首尾两帧, 因为这两帧的一阶差分
参数为 0
ccc=ccc(3:size(m,1)-2,:);

getmfcc=ccc;%返回特征值
%'MFCC 特征参数提取时长'
% toc

% subplot(2,1,1)
% plot(ccc(100,:));
% hold on
% plot(ccc(200,:), 'r');
% plot(ccc(300,:), 'g');
% plot(ccc(400,:), 'y');
% plot(ccc(500,:), 'b');
% xlabel('维数');
% ylabel('幅值');
% title('帧数 100->500');
% subplot(2,1,2)
% plot(ccc(:,1));
% hold on
% plot(ccc(:,2), 'r');
% plot(ccc(:,3), 'g');
% plot(ccc(:,5), 'y');
% plot(ccc(:,7), 'b');
% xlabel('帧数');
% ylabel('幅值');
% title('维数 1->7')

% subplot(2,1,1)
% ccc_1=ccc(:,1);
% plot(ccc_1);title('MFCC');ylabel('幅值');
% [h,w]=size(ccc);
% A=size(ccc);
% subplot(2,1,2)
% plot([1,w],A);
% xlabel('维数');
% ylabel('幅值');
% title('维数与幅值的关系')
end

function f=enframe(x,win,inc)
%ENFRAME split signal up into (overlapping)
frames: one per row. F=(X,WIN,INC)

```

```

% 分帧函数
% win 一般取 10-30ms,inc 一般取 10ms
% win 和 inc 的取值和采样频率有关

% F = ENFRAME(X,LEN) splits the vector X
up into
% frames. Each frame is of length LEN and
occupies
% one row of the output matrix. The last
few frames of X
% will be ignored if its length is not
divisible by LEN.
% It is an error if X is shorter than LEN.
%
% F = ENFRAME(X,LEN,INC) has frames
beginning at increments of INC
% The centre of frame l is
X((l-1)*INC+(LEN+1)/2) for l=1,2,...
% The number of frames is
fix((length(X)-LEN+INC)/INC)
%
% F = ENFRAME(X,WINDOW) or
ENFRAME(X,WINDOW,INC) multiplies
% each frame by WINDOW(:)

% Copyright (C) Mike Brookes 1997
% Version: $Id: enframe.m,v 1.3
2005/02/21 15:22:12 dmb Exp $
%
% VOICEBOX is a MATLAB toolbox for
speech processing.
% Home page:
http://www.ee.ic.ac.uk/hp/staff/dmb/voice
box/voicebox.html
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program is free software; you can
redistribute it and/or modify
% it under the terms of the GNU General
Public License as published by
% the Free Software Foundation; either
version 2 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope
that it will be useful,
% but WITHOUT ANY WARRANTY;
without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the
% GNU General Public License for more
details.
%
% You can obtain a copy of the GNU
General Public License from
%
ftp://prep.ai.mit.edu/pub/gnu/COPYING-2.0
or by writing to
% Free Software Foundation, Inc.,675
Mass Ave, Cambridge, MA 02139, USA.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nx=length(x);
nwin=length(win);
if (nwin == 1)
    len = win;
else
    len = nwin;
end
if (nargin < 3)
    inc = len;
end
nf = fix((nx-len+inc)/inc);
f=zeros(nf,len);
indf= inc*(0:(nf-1)).';
inds = (1:len);
f(:)
x(indf(:,ones(1,len))+inds(ones(nf,1),:));
if (nwin > 1)
    w = win(:)';
    f = f .* w(ones(nf,1),:);
end

function dist = dtw1(t, r)
n = size(t,1);
m = size(r,1);
% 帧匹配距离矩阵
d = zeros(n,m);
for i = 1:n
    for j = 1:m
        d(i,j) = sum((t(i,:)-r(j,:)).^2);
    end
end
% 累积距离矩阵
D = ones(n,m) *realmax;
for i=1:n
    for j=1:m
        if i==1&&j==1
            D(i,j)=d(1,1);
            D1=0;
            D2=0;
            D3=0;

```

```

end
if i==1&&j>1
    D1=D(i,j-1);
    D2=realmax;
    D3=realmax;
end
if j==1&&i>1
    D1=D(i-1,j);
    D2=realmax;
    D3=realmax;
end
if i>1&&j>1
    D1=D(i-1,j);
    D2=D(i,j-1);
    D3=D(i-1,j-1);
end
D(i,j)=d(i,j)+min([D1,D2,D3]);
end
end
dist = D(n,m);

```

```

tic
load character24_plus.mat;
true=0;
sample=20;
for m=1:100
    num=round(rand*9);
    [k,fs]=audioread(['D:\大三上\ (实验)
数字信号处理 \ 改进
\test\int2str(num),'.wav']);
    cc=MFCC2(k,fs);
    dist=ones(10,sample)*realmax;
    for k=1:10
        for i=1:sample
            j=randi(20);
            dist(k,i)=dtw1(cc,
Character{k,j});
            while isnan(dist(k,i))
                j=randi(20);
                dist(k,i)=dtw1(cc,
Character{k,j});
            end
        end
    end
    number=zeros(10,1);
    for i=1:10
        for j=1:20
            if dist(i,j)<2e+05
                number(i)=number(i)+1;
            end
        end
    end
end
% for i=1:10
%     minum(i,1)=1/min(dist(i,:));

```

```

% end
%
minum=(minum-min(minum))/(max(minum)
-min(minum));
%
number=(number-min(number))/(max(num
ber)-min(number));
%     final=0.5*minum+0.5*number;
[~,j]=max(number);
j=j-1;
if j==num
    true=true+1;
end
fprintf(' 真实值为 %d , 识别
为%d\n',num,j);
end
result=true/100
toc

```

```

function j=shibie(cc)
% load character1.mat;
% dist=zeros(10,1);
% for k=1:10
%     for i=1:18
%         j=randi(100);
%         dist(k) = dist(k)+dtw1(cc, p{
k,j});
%     end
% end
% [~,j] = min(dist);
% j=j-1;
% end
load character24_plus.mat;
true=0;
sample=20;
dist=ones(10,sample)*realmax;
for k=1:10
    for i=1:sample
        j=randi(20);
        dist(k,i)=dtw1(cc, Character{
k,j});
        while isnan(dist(k,i))
            j=randi(20);
            dist(k,i)=dtw1(cc, Chara

```

```
cter{k,j});
    end
end
end
number=zeros(10,1);
for i=1:10
    for j=1:20
        if dist(i,j)<2e+05
            number(i)=number(i)+1;
        end
    end
end
end
[~,j]=max(number);
j=j-1;
end
```