

公告

Since 2017.05.21:

Visitors

CN 164,792

US 6,915

TW 3,449

HK 2,890

JP 1,844

GB 1,255

DE 977

SG 833

AU 827

CA 650

Pageviews: 307,480

FLAG counter

昵称：桂。  
园龄：1年8个月  
粉丝：260  
关注：15  
+加关注

|    |         |    |    |    |    |    |   |
|----|---------|----|----|----|----|----|---|
| <  | 2018年9月 |    |    |    |    |    | > |
| 日  | 一       | 二  | 三  | 四  | 五  | 六  |   |
| 26 | 27      | 28 | 29 | 30 | 31 | 1  |   |
| 2  | 3       | 4  | 5  | 6  | 7  | 8  |   |
| 9  | 10      | 11 | 12 | 13 | 14 | 15 |   |
| 16 | 17      | 18 | 19 | 20 | 21 | 22 |   |
| 23 | 24      | 25 | 26 | 27 | 28 | 29 |   |
| 30 | 1       | 2  | 3  | 4  | 5  | 6  |   |

最新随笔

- 1. Mathematica简介
- 2. 概率论02
- 3. 概率论01
- 4. 小数据滤波器的诸多工程问题及改进措施
- 5. coon's patch
- 6. system generator学习笔记【02】
- 7. CZT变换 (chirp z-transform)
- 8. system generator学习笔记【01】
- 9. 硬件资源拆解
- 10. 基础008\_定浮点转化[floating point IP]

随笔分类

- 00-C 语言(2)
- 01-MATLAB(12)
- 02-Python(9)
- 03-前端相关(1)
- 10-HTK(1)
- 11-Linux(3)
- 12-Tensorflow(2)
- 13-scikit-learn(4)
- 14-工具使用(16)
- 15-音频分析工具(2)
- 16-硬件(39)
- 20-音频信号(30)
- 21-信号处理(55)
- 22-图像(3)
- 23-模式识别(18)
- 24-矩阵(18)
- 25-测向算法(32)
- 26-电磁(2)
- 27-数学(2)
- 待归档(4)
- 读书(26)
- 工作记录(12)
- 随手记(7)

随笔-239 文章-6 评论-174

音频特征提取——pyAudioAnalysis工具包

作者：桂。

时间：2017-05-04 18:31:09

链接：<http://www.cnblogs.com/xingshansi/p/6806637.html>



前言

语音识别等应用离不开音频特征的提取，最近在看音频特征提取的内容，用到一个python下的工具包——**pyAudioAnalysis**: An Open-Source Python Library for Audio Signal Analysis，该工具包的说明文档可以[点击这里](#)下载，对应的github链接[点击这里](#)。

这个工具包原说明文档支持的是Linux安装，且不能与python3很好地兼容，注意啦

一、常用工具包简介

目前针对音频信号，C/C++、Python、MATLAB等常用的工具包有：

| Name                                   | Description  |
|--|--|
| Yaafe                                  | A Python library for audio feature extraction and basic audio I/O ( <a href="http://yaafe.sourceforge.net/">http://yaafe.sourceforge.net/</a> )  |
| Essentia                               | An open-source C++ library for audio analysis and music information retrieval. Mostly focuses on audio feature extraction, basic I/O, while it also provides some basic classification functionalities <a href="http://essentia.upf.edu/">http://essentia.upf.edu/</a>   |
| aubio                                  | A C library for basic audio analysis: pitch tracking, onset detection, extraction of MFCCs, beat and meter tracking, etc. Provides wrappers for Python. <a href="http://aubio.org/">http://aubio.org/</a>  |
| CLAM (C++ Library for Audio and Music) | A framework for research / development in the audio and music domain. Provides the means to perform complex audio signal analysis, transformations and synthesis. Also provides a graphical tool. <a href="http://clam-project.org/">http://clam-project.org/</a>  |
| Matlab Audio Analysis Library          | A Matlab library for audio feature extraction, classification, segmentation and music information retrieval <a href="http://www.mathworks.com/matlabcentral/fileexchange/45831-matlab-audio-analysis-library">http://www.mathworks.com/matlabcentral/fileexchange/45831-matlab-audio-analysis-library</a> . Can be used as companion material for the book [1]     |
| librosa                                | A Python library that implements some audio features (MFCCs, chroma and beat-related features), sound decomposition to harmonic and percussive components, audio effects (pitch shifting, etc) and some basic communication with machine learning components (e.g. clustering) <a href="https://github.com/bmcfee/librosa/">https://github.com/bmcfee/librosa/</a> |
| PyCASP                                 | This Python library focuses on providing a collection of specializers towards automatic mapping of computations onto parallel processing units (either GPUs or multicore CPUs). These computations are presented through a couple of audio-related examples.   |
| seewave                                | This is an R package for basic sound analysis and synthesis. Mostly focusing on feature extraction and basic I/O. <a href="https://cran.r-project.org/web/packages/seewave/index.html">https://cran.r-project.org/web/packages/seewave/index.html</a>  |
| bob                                    | An open general signal processing and machine learning library (C++ and Python). <a href="http://idiap.github.io/bob/">http://idiap.github.io/bob/</a>   |

二、pyAudioAnalysis工具包简介

pyAudioAnalysis是一个音频处理工具包，主要功能如图：

随笔档案

- 2018年6月 (4)
- 2018年5月 (20)
- 2018年4月 (4)
- 2018年3月 (2)
- 2018年2月 (11)
- 2018年1月 (4)
- 2017年12月 (3)
- 2017年11月 (11)
- 2017年10月 (13)
- 2017年9月 (14)
- 2017年8月 (18)
- 2017年7月 (14)
- 2017年6月 (16)
- 2017年5月 (37)
- 2017年4月 (29)
- 2017年3月 (25)
- 2017年2月 (10)
- 2017年1月 (4)

积分与排名

积分 - 165677  
排名 - 1841

最新评论

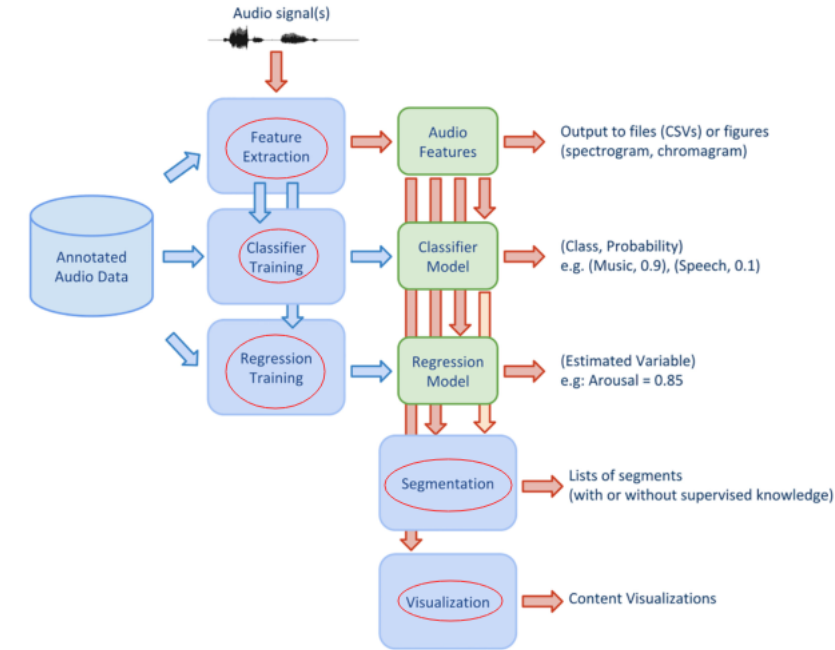
- 1. Re:MATLAB (2) ——小波工具箱使用简介  
@若知，问题解决了吗，求指点一二  
--大枫侠
- 2. Re:信号处理——EMD、VMD的一点小思考  
你好，博主，我做的轴承故障，帮忙看下VMD选择几层分解，根据图。  
--641573672
- 3. Re:信号处理——EMD、VMD的一点小思考  
博主你好，请问在VMD中（function [u, u\_hat, omega] = VMD(signal, alpha, tau, K, DC, init, tol)），每次K值下会求得相应的中心频率……  
--蓝色砂砾
- 4. Re:PCA算法  
真的很详细，  
--GXTon
- 5. Re:稀疏傅里叶变换（sparse FFT）  
@陈曦明你看的那个论文，P25， $\sigma^2 = B^2 \cdot \log 2$ ， $\sigma$ 与M有数学上的联系，自己可以换算一下，细节就不帮你看了。…  
--桂。

阅读排行榜

- 1. python音频处理用到的操作(25774)
- 2. 音频特征提取——librosa工具包使用(18920)
- 3. 信号处理——Hilbert变换及谱分析(15995)
- 4. PCA算法(12337)
- 5. MATLAB (2) ——小波工具箱使用简介(12263)

推荐排行榜

- 1. 信号处理——EMD、VMD的一点小思考(9)
- 2. 音频特征提取——常用音频特征(7)
- 3. python音频处理用到的操作(6)



其中Feature Extraction包括（顺序有先后）：

| Feature ID | Feature Name       | Description   |
|------------|--------------------|---|
| 1          | Zero Crossing Rate | The rate of sign-changes of the signal during the duration of a particular frame.   |
| 2          | Energy             | The sum of squares of the signal values, normalized by the respective frame length.   |
| 3          | Entropy of Energy  | The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.   |
| 4          | Spectral Centroid  | The center of gravity of the spectrum.  |
| 5          | Spectral Spread    | The second central moment of the spectrum.  |
| 6          | Spectral Entropy   | Entropy of the normalized spectral energies for a set of sub-frames.  |
| 7          | Spectral Flux      | The squared difference between the normalized magnitudes of the spectra of the two successive frames.   |
| 8          | Spectral Rolloff   | The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.  |
| 9-21       | MFCCs              | Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.   |
| 22-33      | Chroma Vector      | A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing). |
| 34         | Chroma Deviation   | The standard deviation of the 12 chroma coefficients.   |

补充说明一下：

- 1-Zero Crossing Rate：短时平均过零率，即每帧信号内，信号过零点的次数，体现的是频率特性
- 2-Energy：短时能量，即每帧信号的平方和，体现的是信号能量的强弱
- 3-Entropy of Energy：能量熵，跟频谱的谱熵（Spectral Entropy）有点类似，不过它描述的是信号的时域分布情况，体现的是连续性
- 4-Spectral Centroid：频谱中心又称为频谱一阶距，频谱中心的值越小，表明越多的频谱能量集中在低频范围内，如：voice与music相比，通常spectral centroid较低
- 5-Spectral Spread：频谱延展度，又称为频谱二阶中心矩，它描述了信号在频谱中心周围的分布状况
- 6-Spectral Entropy：谱熵，根据熵的特性可以知道，分布越均匀，熵越大，能量熵反应了每一帧信号的均匀程度，如说话人频谱由于共振峰存在显得不均匀，而白噪声的频谱就更加均匀，借此进行VAD便是应用之一
- 7-Spectral Flux：频谱通量，描述的是相邻帧频谱的变化情况

```
1 function [vsf] = FeatureSpectralFlux (X, f_s)
2
3     % difference spectrum (set first diff to zero)
4     afDeltaX = diff([X(:,1), X],1,2);
5
```

4. 霍夫变换(4)
5. 统计学习方法: 感知机(3)

```

6         % flux
7         vsf         = sqrt(sum(afDeltaX.^2))/size(X,1);
8     end

```

- 8-Spectral Rolloff: 频谱滚降点, 给出定义:

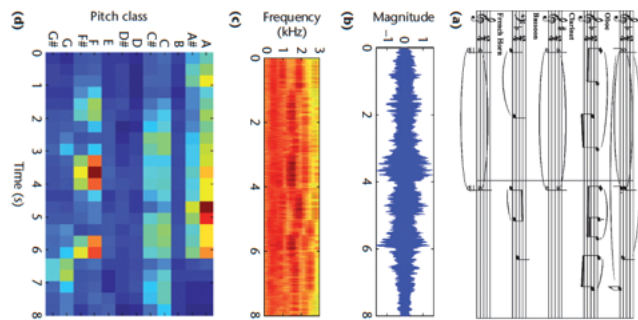
**Spectral Rolloff:** The spectral rolloff is defined as the frequency below which 85% of the distribution magnitude is concentrated [76]

$$\arg \min_{f_c \in \{1, \dots, N\}} \sum_{i=1}^{f_c} m_i \geq 0.85 \cdot \sum_{i=1}^N m_i \quad (16)$$

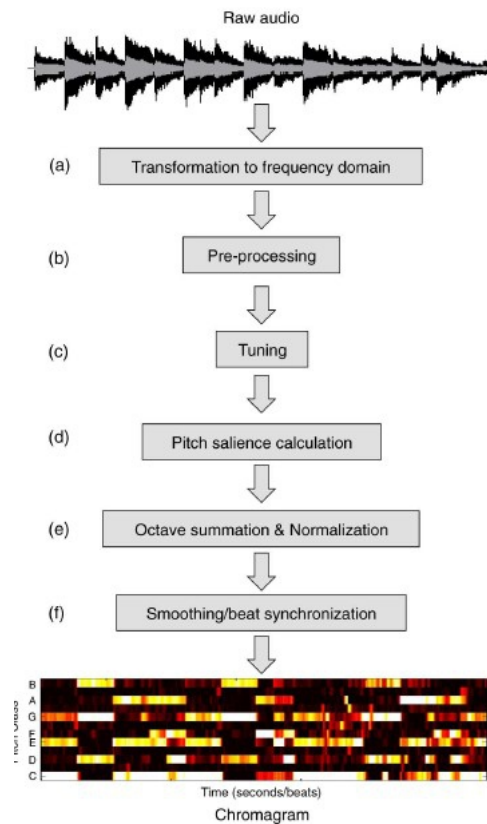
where  $f_c$  is the rolloff frequency and  $m_i$  is the magnitude of the  $i$ -th frequency component of the spectrum. The

- 9~21-MFCCs: 就是大名鼎鼎的梅尔倒谱系数, 这个网上资料非常多, 也是非常重要的音频特征。
- 22~33-Chroma Vector: 这个有12个参数, 对应就是12级音阶, 还是看原文解释: A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
- 34-Chroma Deviation: 这个就是Chroma Vector的标准方差。

这个在音乐声里可能用的比较多, 目前没有接触这类特征:



什么是Chroma特征呢? 给出一个示意图



code示例:

```

1 from pyAudioAnalysis import audioBasicIO
2 from pyAudioAnalysis import audioFeatureExtraction
3 import matplotlib.pyplot as plt
4 [Fs, x] = audioBasicIO.readAudioFile("sample.wav");
5 F = audioFeatureExtraction.stFeatureExtraction(x, Fs, 0.050*Fs, 0.025*Fs);

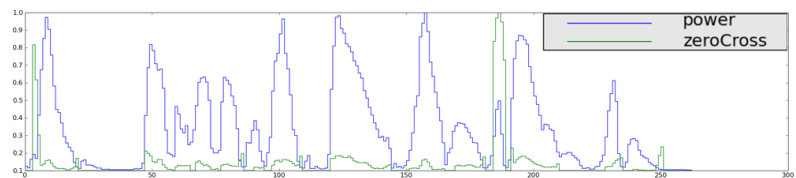
```

```
6 plt.subplot(2,1,1); plt.plot(F[0,:]); plt.xlabel('Frame no'); plt.ylabel('ZCR');
7 plt.subplot(2,1,2); plt.plot(F[1,:]); plt.xlabel('Frame no'); plt.ylabel('Energy'); plt.show
```

如果希望了解更多的音频特征，这里给出一个链接，[点击这里](#)，包含的特征有：

- mfccEnv – the first 8 mfccs describing spectral envelope ([graph](#), [sound](#))
- mfccFrq – the first 8 mfccs describing frequencies ([graph](#), [sound](#))
- power – spectral power ([graph](#), [sound](#))
- rms – rms amplitude ([graph](#), [sound](#))
- centroid – spectral centroid ([graph](#), [sound](#))
- centroidsubs – a collection of four subbanded spectral centroids ([graph](#), [sound](#))
- peakCentroid – spectral centroid of the peak frequency spectrum ([graph](#), [sound](#))
- zeroCross – zero crossing rate ([graph](#), [sound](#))
- flux – non-normalized spectral flux ([graph](#), [sound](#))
- fluxsubs – a collection of four sub-banded spectral fluxes ([graph](#), [sound](#))
- flatness – spectral flatness ([graph](#), [sound](#))
- slope – spectral slope ([graph](#), [sound](#))
- rolloff – spectral rolloff ([graph](#), [sound](#))
- variance – spectral variance ([graph](#), [sound](#))
- kurtosis – spectral kurtosis ([graph](#), [sound](#))
- skewness – spectral skewness ([graph](#), [sound](#))
- crest – spectral crest ([graph](#), [sound](#))
- noisiness – spectral noisiness ([graph](#), [sound](#))
- yin – monophonic pitch estimate as MIDI number ([graph](#), [sound](#))
- peakFreqs-N – a collection of the N loudest spectral peaks ([graph](#) of peakFreqs-4, [sound](#))

对应都有graph、sound可以点击，sound是对应的音频，graph对应的是特征的效果图，比如打开zeroCross：



### 三、pyAudioAnalysis工具包安装

pyAudioAnalysis对应链接[点击这里](#)。安装这个工具包需要依赖：

- NUMPY `pip install numpy`
- MATPLOTLIB `pip install matplotlib`
- SCIPY `pip install scipy`
- SKLEARN `pip install sklearn`
- hmmlearn `pip install hmmlearn`
- Simplejson `pip install simplejson`
- eyeD3 `pip install eyed3`
- pydub `pip install pydub`

#### A-hmmlearn安装

hmmlearn的链接[点击这里](#)。安装hmmlearn有几个前提：

The required dependencies to use hmmlearn are

- Python >= 2.6
- NumPy (tested to work with >=1.9.3)
- SciPy (tested to work with >=0.16.0)
- scikit-learn >= 0.16

You also need Matplotlib >= 1.1.1 to run the examples and pytest >= 2.6.0 to run the tests.

下载之后，我把hmmlearn-master放在python-3.5.2-0\Lib\目录，cmd窗口下cd进去，输入：

```
1 | pip install -U --user hmmlearn
```

即可安装成功：

```
Collecting hmmlearn
  Downloading hmmlearn-0.2.0-cp35-cp35m-win_amd64.whl (81kB)
    50% |#####| 40kB 100kB/s eta 0:00
    62% |#####| 51kB 89kB/s eta 0
    75% |#####| 61kB 106kB/s
    87% |#####| 71kB 104k
   100% |#####| 81kB
113kB/s
Installing collected packages: hmmlearn
Successfully installed hmmlearn-0.2.0
```

## B-Simplejson工具包安装:

Simplejson是Python的JSON编码和解码器,它具有简单、快速、完整、正确和易于扩展的特点,对应的链接[点击这里](#)。Simplejson工具包直接conda安装即可:

```
C:\Users\Nobleding>conda install simplejson
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment C:\Users\Nobleding\Anaconda3:

The following NEW packages will be INSTALLED:

    simplejson: 3.10.0-py35_0

The following packages will be UPDATED:

    conda:      4.3.16-py35_0 --> 4.3.17-py35_0

Proceed [y/n]? y

simplejson-3.1 100% |#####| Time: 0:00:00 198.19 kB/s
conda-4.3.17-p 100% |#####| Time: 0:00:06 91.49 kB/s
```

## C-eyed3安装:

eyed3: A tool for working with audio files, specifically MP3 files containing ID3 metadata. 它提供了读写 ID3 标签(v1.x 和 v2.3/v2.4)的功能。同时可检测 MP3 文件的头信息,包括比特率、采样频率和播放时间等。eyed3直接conda install没有成功,对应的链接[点击这里](#)。选择了这个版本:

[eyeD3-0.8.0a5-py2.py3-none-any.whl](#) 23-Jan-2017 21:53 163K

放在了python库的Lib文件夹下: C:\Users\Nobleding\Anaconda3\pkgs\python-3.5.2-0\Lib, cd到对应目录下, pip install 文件名.whl, 即可完成安装

```
C:\Users\Nobleding>cd C:\Users\Nobleding\Anaconda3\pkgs\python-3.5.2-0\Lib
C:\Users\Nobleding\Anaconda3\pkgs\python-3.5.2-0\Lib>pip install eyeD3-0.8.0a5-py2.py3-none-any.whl
Processing c:\users\nobleding\anaconda3\pkgs\python-3.5.2-0\lib\eyeD3-0.8.0a5-py2.py3-none-any.whl
Collecting grako (from eyeD3==0.8.0a5)
  Downloading grako-3.22.0-py2.py3-none-any.whl (82kB)
    49% |#####| 40kB 62kB/s eta 0:00:
    61% |#####| 51kB 75kB/s eta 0
    74% |#####| 61kB 84kB/s e
    86% |#####| 71kB 98kB
    99% |#####| 81kB
   100% |#####| 92kB
123kB/s
Installing collected packages: grako, eyeD3
Successfully installed eyeD3-0.8.0a5 grako-3.22.0
```

## D-pydub安装:

pydub是音频处理常用的工具包,例如:

打开一个wav格式文件:

```
1 from pydub import AudioSegment
2 song = AudioSegment.from_wav("never_gonna_give_you_up.wav")
```

打开一个mp3格式文件:

```
1 song = AudioSegment.from_mp3("never_gonna_give_you_up.mp3")
```

或者其他音频、视频格式:

```
1 ogg_version = AudioSegment.from_ogg("never_gonna_give_you_up.ogg")
2 flv_version = AudioSegment.from_flv("never_gonna_give_you_up.flv")
3
4 mp4_version = AudioSegment.from_file("never_gonna_give_you_up.mp4", "mp4")
5 wma_version = AudioSegment.from_file("never_gonna_give_you_up.wma", "wma")
6 aac_version = AudioSegment.from_file("never_gonna_give_you_up.aiff", "aac")
```

更多细节信息可以访问[其主页](#)。我在[github](#)上下载对应的工具包,里边有对应的安装说明。

```
C:\Users\Nobleding\Anaconda3\pkgs\python-3.5.2-0\Lib\pydub-master>pip install pydub
Collecting pydub
  Downloading pydub-0.18.0-py2.py3-none-any.whl
Installing collected packages: pydub
Successfully installed pydub-0.18.0
```

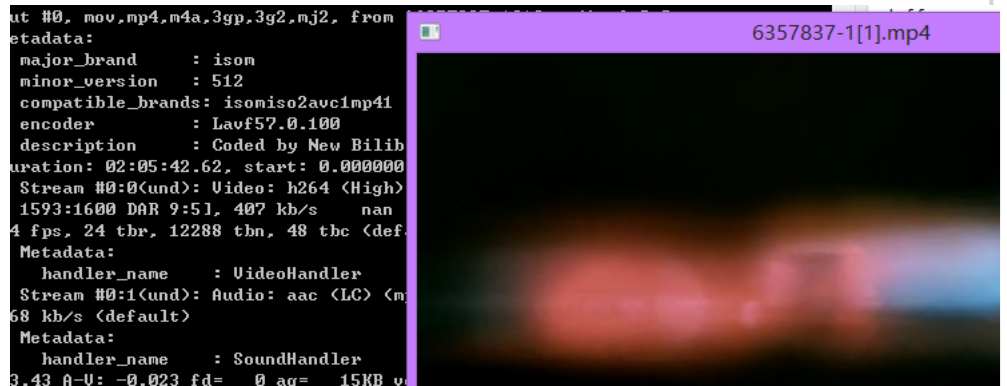
如果处理wav文件,没有其他要求,如果音频是其他格式它要求电脑安装 [ffmpeg](#) [orlibav](#).如果没有安装,运行会有提示:

```
In [7]: import pydub
C:\Users\Nobleling\Anaconda3\lib\site-packages
\pydub\utils.py:165: RuntimeWarning: Couldn't
find ffmpeg or avconv - defaulting to ffmpeg,
but may not work
warn("Couldn't find ffmpeg or avconv -
defaulting to ffmpeg, but may not work",
RuntimeWarning)
```

ffmpeg下载, 选择版本

| Version          | Architecture | Linking |
|------------------|--------------|---------|
| 20170503-a75ef15 | 64-bit       | Static  |

解压并添加环境变量, 并利用ffplay测试一下打开一个mp4文件:



ffmpeg安装成功。这个时候import pydub,不再有warning信息

```
In [9]: import pydub
In [10]: |
```

### E-pyAudioAnalysis安装

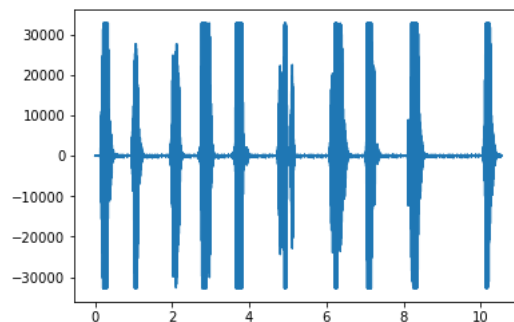
Github给出的是linux下的安装思路, 这里下载之后将pyAudioAnalysis放在了\Anaconda3\Lib\site-packages文件夹下, 输入指令:

```
In [11]: import pyAudioAnalysis.audioBasicIO
In [12]: |
```

成功调用, 原数据是支持Python2的, 很多细节要修改, 给出一个简单读取wav的测试:

```
1 from pyAudioAnalysis import audioBasicIO
2 import numpy as np
3 import matplotlib.pyplot as plt
4 [Fs, x] = audioBasicIO.readAudioFile("count2.wav");
5 time = np.arange(0, len(x))*1.0/Fs
6 plt.plot(time, x)
```

效果图:



分类: [02-Python,20-音频信号](#)

标签: [pyAudioAnalysis](#), [python](#), [音频](#)

好文要顶

关注我

收藏该文



桂...  
关注 - 15  
粉丝 - 260

+加关注



« 上一篇: [【转】python2与python3的主要区别](#)

» 下一篇: [2017随记——5月](#)

posted @ 2017-05-04 23:53 桂。 阅读(5842) 评论(11) 编辑 收藏

## 评论列表

#1楼 2017-05-05 09:07 寻风问雨

讲的不错。希望多搞点demo，新手学习了

支持(0) 反对(0)

#2楼[楼主] 2017-05-06 09:08 桂。

@ 寻风问雨

新写了一篇，讲了些demo，感兴趣可以一起讨论讨论

支持(0) 反对(0)

#3楼 2017-05-08 21:33 onsunsl

楼主pyaudioanalysis不支持python3吧。  
能出一些demo吗？

支持(0) 反对(0)

#4楼[楼主] 2017-05-09 23:20 桂。

@ onsunsl

不好意思，才看到你的问题。我在前言里提到了与python3不能很好兼容，如果想调用需要修改一些指令。一般的音频处理，scipy工具包以及Librosa应该也是够用的了。pyAudioAnalysis只不过是多了些特征的后处理，这个完全可以借助其他工具包单独实现。

支持(0) 反对(0)

#5楼 2017-05-11 09:44 onsunsl

嗯，我看到您文章里提了py3所以问问支不支持：没关系用py2也可以的：

支持(0) 反对(0)

#6楼 2017-05-11 09:48 onsunsl

我另外想实现一个识别人说话的端点，也就是录一段话，然后用pyAduioAnalysis现有的库来实现，录音频里每一个字的起始和结束位置提取出来，不知道应该是个什么思路，望楼主指点一二。

支持(0) 反对(0)

#7楼 2017-05-11 09:49 onsunsl

录音人说话的内容和语速不确定。

支持(0) 反对(0)

#8楼 2017-05-11 09:56 onsunsl

是不是应该多录一些样本，加于标定，来训练模型，然后用这个模块来测试新的音频。不过应该选择什么模块，怎么训练怎么测试，我还不清楚。

支持(0) 反对(0)

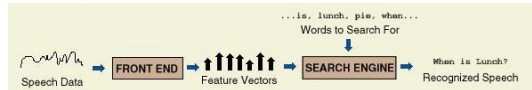
#9楼[楼主] 2017-05-11 10:14 桂。

@ onsunsl

我也是刚接触没多久，只能说一说自己的思路，供你参考一下。

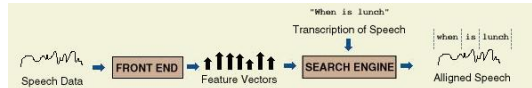
有话帧检测（VAD）很多是利用能量的信息、频率的分布特性来实现，更多是切割 有用信号/非有用信号。如果根据清音、浊音特性来识别单词。比如：啊，就是一个单词，发：是清音+浊音，但妈却又是两个浊音。个人觉得每一个字的起始、结束，目前来讲直接VAD处理应该做不到。

语音识别中技术——



利用统计模型的概率估计，找出最优解。

另外有forced alignment技术，Forced alignment与识别原理类似，不同点是已经知道句子内容，从而进行内容切割对齐：



但这个跟你说的可能还是有区别。

可以Google：Forced alignment + github，应该可以找到一些有用的资料

支持(0) 反对(0)

#10楼 2017-05-12 12:17 onsunsl

在github上找了一个强制对齐的工具，正在安装环境，不过看了一下，支持的38语言里不支持中文。很遗憾。

<https://github.com/readbeyond/aeneas>

支持(1) 反对(0)

#11楼 2017-06-14 10:03 caimj

ffmpeg的安装有什么详细的步骤吗？以及有具体使用的方式吗？

支持(0) 反对(0)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【免费】要想入门学习Linux系统技术, 你应该先选择一本适合自己的书籍

【前端】SpreadJS表格控件, 可嵌入应用开发的在线Excel

【直播】如何快速接入微信支付功能



#### 最新IT新闻:

- 少儿编程市场的冰与火
  - Visual Studio Code 1.27.2发布, Bug修复版
  - OpenStack的八年之痒
  - 库克谈最贵iPhone定价: 够创新所以很合理
  - AMD或于本月底前宣布对32bit显卡驱动停更
- » 更多新闻...



华为全联接大会 | 上海 | 2018.10.10-12

[ 大会门票+云服务器 ] 专属套餐0.35折起



#### 最新知识库文章:

- 为什么说 Java 程序员必须掌握 Spring Boot ?
  - 在学习中, 有一个比掌握知识更重要的能力
  - 如何招到一个靠谱的程序员
  - 一个故事看懂“区块链”
  - 被踢出去的用户
- » 更多知识库文章...