小组成员信息

- 组号：实验课 第2小组
- 组长：巩羽飞
- 组员：廉涟，李玉昆，王铭慧，王菁
- 贡献度：每人20%

实验目标： 通过本实验，你将学习并实践信号处理中的三个重要任务：端点检测、基音周期检测以及 MFCC 特征提取。这些任务是语音信号处理和音频处理中的基础，对于语音识别和语音分析等应用至关重要。其中，端点检测、基音周期检测实验可以二选一，MFCC特征提取实验必须完成。

实验方法：

1. 端点检测：

- 阅读端点检测的相关文献或资料，了解双门限法的原理和实现方式。
- 使用 Python 编程语言，实现端点检测算法。
- 在代码中设定合适的阈值来检测端点，你可以尝试不同的阈值来获取更好的效果。
- 编写代码以可视化端点检测结果，将检测到的端点在原始信号上标出。

2. 基音周期检测：

- 学习基音周期检测的原理，理解基音周期在语音信号中的重要性。
- 使用 librosa 库或其他音频处理库加载音频数据。
- 探索基音周期检测的方法，并尝试实现其中一种方法。你可以使用自相关函数或其他方法来检测基音周期。
- 编写代码以可视化基音周期检测的结果，将检测到的基音周期在音频信号上标出。

3. MFCC 特征提取：

- 了解 MFCC 特征在语音信号处理中的重要性以及其在语音识别中的应用。
- 使用 librosa 库或其他音频处理库加载音频数据。
- 实现 MFCC 特征提取算法。
- 编写代码以可视化提取的 MFCC 特征，观察不同时间段内的特征变化。

实验要求：

录制一段音频，并开展上述实验，并可视化实验结果，并保存图片到本地。其中实验1和2可任选一个，实验3必做。

In [ ]:
```
# #安装依赖库，可以跳过已安装的库
%pip install -r requirements.txt
```

In [ ]:
```
%pip show numpy
%pip show matplotlib
%pip show scipy
%pip show pandas
%pip show pyaudio
%pip show wave
```

```
%pip show librosa
%pip show IPython
# 如果显示WARNING: Package(s) not found,则需要安装对应依赖库
```

```
Name: numpy
Version: 1.24.4
Summary: Fundamental package for array computing in Python
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD-3-Clause
Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
Requires:
Required-by: altair, biopython, contourpy, gradio, kneed, librosa, matplotlib, nu
mba, onnxruntime-gpu, openunmix, pandas, pyarrow, pydeck, scikit-learn, scipy, se
aborn, soxr, streamlit, torchlibrosa
Note: you may need to restart the kernel to use updated packages.
Name: matplotlib
Version: 3.7.4
Summary: Python plotting package
Home-page: https://matplotlib.org
Author: John D. Hunter, Michael Droettboom
Author-email: matplotlib-users@python.org
License: PSF
Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
Requires: contourpy, cycler, fonttools, importlib-resources, kiwisolver, numpy, p
ackaging, pillow, pyparsing, python-dateutil
Required-by: gradio, seaborn, voicefixer2
Note: you may need to restart the kernel to use updated packages.
Name: scipy
Version: 1.10.1
Summary: Fundamental algorithms for scientific computing in Python
Home-page: https://scipy.org/
Author:
Author-email:
License: Copyright (c) 2001-2002 Enthought, Inc. 2003-2022, SciPy Developers.
        All rights reserved.

        Redistribution and use in source and binary forms, with or without
        modification, are permitted provided that the following conditions
        are met:

        1. Redistributions of source code must retain the above copyright
           notice, this list of conditions and the following disclaimer.

        2. Redistributions in binary form must reproduce the above
           copyright notice, this list of conditions and the following
           disclaimer in the documentation and/or other materials provided
           with the distribution.

        3. Neither the name of the copyright holder nor the names of its
           contributors may be used to endorse or promote products derived
           from this software without specific prior written permission.

        THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
        "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
        LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
        A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
        OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
        SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
        LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
        DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
        THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
        (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
```

```
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----

This binary distribution of SciPy also bundles the following software:


Name: OpenBLAS
Files: .libs/libopenb*.so
Description: bundled as a dynamically linked library
Availability: https://github.com/xianyi/OpenBLAS/
License: 3-clause BSD
  Copyright (c) 2011-2014, The OpenBLAS Project
  All rights reserved.

  Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions are
  met:

     1. Redistributions of source code must retain the above copyright
        notice, this list of conditions and the following disclaimer.

     2. Redistributions in binary form must reproduce the above copyright
        notice, this list of conditions and the following disclaimer in
        the documentation and/or other materials provided with the
        distribution.
     3. Neither the name of the OpenBLAS project nor the names of
        its contributors may be used to endorse or promote products
        derived from this software without specific prior written
        permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
  AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, T
HE
  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURP
OSE
  ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS B
E
  LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CON
SEQUENTIAL
  DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
  SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWE
VER
  CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIAB
ILITY,
  OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF T
HE
  USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAG
E.


Name: LAPACK
Files: .libs/libopenb*.so
Description: bundled in OpenBLAS
Availability: https://github.com/xianyi/OpenBLAS/
License 3-clause BSD
  Copyright (c) 1992-2013 The University of Tennessee and The University
                          of Tennessee Research Foundation.  All rights
```

Full text of license texts referred to above follows (that they are listed below does not necessarily imply the conditions apply to the present binary release):

GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

This GCC Runtime Library Exception ("Exception") is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual

```
target processor architecture, in executable form or suitable for
input to an assembler, loader, linker and/or execution
phase. Notwithstanding that, Target Code does not include data in any
format that is used as a compiler intermediate representation, or used
for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in
non-intermediate languages designed for human-written code, and/or in
Java Virtual Machine byte code, into Target Code. Thus, for example,
use of source code generators and preprocessors need not be considered
part of the Compilation Process, since the Compilation Process can be
understood as starting with the output of the generators or
preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or
with other GPL-compatible software, or if it is done without using any
work based on GCC. For example, using non-GPL-compatible Software to
optimize any GCC intermediate representations would not qualify as an
Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by
combining the Runtime Library with Independent Modules, even if such
propagation would otherwise violate the terms of GPLv3, provided that
all Target Code was generated by Eligible Compilation Processes. You
may then convey such a combination under terms of your choice,
consistent with the licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general
presumption that third-party software is unaffected by the copyleft
requirements of the license of GCC.

----

                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
```

```
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.
```

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy. The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy. Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies. Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License. If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it. "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form. A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities. However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work. For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,

```
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.
```

```
  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
```

medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied

```
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
```

```
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
```

covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so

```
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you ma
y
not convey it at all.  For example, if you agree to terms that obligate y
ou
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.
```

```
  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRA
M
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING A
NY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIR
D
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
```

```
        copy of the Program in return for a fee.

                             END OF TERMS AND CONDITIONS

                    How to Apply These Terms to Your New Programs

          If you develop a new program, and you want it to be of the greatest
        possible use to the public, the best way to achieve this is to make it
        free software which everyone can redistribute and change under these term
s.

          To do so, attach the following notices to the program.  It is safest
        to attach them to the start of each source file to most effectively
        state the exclusion of warranty; and each file should have at least
        the "copyright" line and a pointer to where the full notice is found.

            <one line to give the program's name and a brief idea of what it doe
s.>
            Copyright (C) <year>  <name of author>

            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 3 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 should have received a copy of the GNU General Public License
            along with this program.  If not, see <http://www.gnu.org/licenses/>.

        Also add information on how to contact you by electronic and paper mail.

          If the program does terminal interaction, make it output a short
        notice like this when it starts in an interactive mode:

            <program>  Copyright (C) <year>  <name of author>
            This program comes with ABSOLUTELY NO WARRANTY; for details type `sho
w w'.
            This is free software, and you are welcome to redistribute it
            under certain conditions; type `show c' for details.

        The hypothetical commands `show w' and `show c' should show the appropria
te

        parts of the General Public License.  Of course, your program's commands
        might be different; for a GUI interface, you would use an "about box".

          You should also get your employer (if you work as a programmer) or scho
ol,

        if any, to sign a "copyright disclaimer" for the program, if necessary.
        For more information on this, and how to apply and follow the GNU GPL, se
e

        <http://www.gnu.org/licenses/>.

          The GNU General Public License does not permit incorporating your progr
am

        into proprietary programs.  If your program is a subroutine library, you
        may consider it more useful to permit linking proprietary applications wi
```

```
th
        the library.  If this is what you want to do, use the GNU Lesser General
        Public License instead of this License.  But first, please read
        <http://www.gnu.org/philosophy/why-not-lgpl.html>.
Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
Requires: numpy
Required-by: kneed, librosa, scikit-learn, voicefixer2
Note: you may need to restart the kernel to use updated packages.
Name: pandas
Version: 2.0.3
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: 
Author: 
Author-email: The Pandas Development Team <pandas-dev@python.org>
License: BSD 3-Clause License

        Copyright (c) 2008-2011, AQR Capital Management, LLC, Lambda Foundry, In
c. and PyData Development Team
        All rights reserved.

        Copyright (c) 2011-2023, Open source contributors.

        Redistribution and use in source and binary forms, with or without
        modification, are permitted provided that the following conditions are me
t:

        * Redistributions of source code must retain the above copyright notice,
this
          list of conditions and the following disclaimer.

        * Redistributions in binary form must reproduce the above copyright notic
e,
          this list of conditions and the following disclaimer in the documentati
on
          and/or other materials provided with the distribution.

        * Neither the name of the copyright holder nor the names of its
          contributors may be used to endorse or promote products derived from
          this software without specific prior written permission.

        THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS I
S"
        AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
        IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOS
E ARE
        DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIA
BLE
        FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIA
L
        DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS O
R
        SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVE
R
        CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABIL
ITY,
        OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
        OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
```

```
Requires: numpy, python-dateutil, pytz, tzdata
Required-by: altair, gradio, seaborn, streamlit
Note: you may need to restart the kernel to use updated packages.
WARNING: Package(s) not found: pyaudio
Note: you may need to restart the kernel to use updated packages.
WARNING: Package(s) not found: wave
Note: you may need to restart the kernel to use updated packages.
Name: librosa
Version: 0.10.1
Summary: Python module for audio and music processing
Home-page: https://librosa.org
Author: Brian McFee, librosa development team
Author-email: brian.mcfee@nyu.edu
License: ISC
Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
Requires: audioread, decorator, joblib, lazy-loader, msgpack, numba, numpy, pooc
h, scikit-learn, scipy, soundfile, soxr, typing-extensions
Required-by: torchlibrosa, voicefixer2
Note: you may need to restart the kernel to use updated packages.
Name: ipython
Version: 8.12.0
Summary: IPython: Productive Interactive Computing
Home-page: https://ipython.org
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD-3-Clause
Location: /home/bbs/miniconda3/envs/mdx/lib/python3.8/site-packages
Requires: backcall, decorator, jedi, matplotlib-inline, pexpect, pickleshare, pro
mpt-toolkit, pygments, stack-data, traitlets, typing-extensions
Required-by: ipykernel
Note: you may need to restart the kernel to use updated packages.
```

我们鼓励大家使用自己录制的音频进行实验，下面是录制音频的简单示例，可根据自己设备的情况，对录制时长和采样率进行调整

In [ ]:

```python
#录制音频
def record_audio(duration):
    import pyaudio
    import wave

    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 44100
    WAVE_OUTPUT_FILENAME = "demo.wav"

    p = pyaudio.PyAudio()

    stream = p.open(format=FORMAT,
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    frames_per_buffer=CHUNK)

    print("* 录音中...")

    frames = []

    for i in range(0, int(RATE / CHUNK * duration)):
```

```
        data = stream.read(CHUNK)
        frames.append(data)

    print("* 录音结束!")

    stream.stop_stream()
    stream.close()
    p.terminate()

    wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
    wf.setnchannels(CHANNELS)
    wf.setsampwidth(p.get_sample_size(FORMAT))
    wf.setframerate(RATE)
    wf.writeframes(b''.join(frames))
    wf.close()

record_audio(duration=5) #录音5秒,并保存为demo.wav
```

In [ ]:
```
import IPython
IPython.display.Audio('demo.wav') #播放demo.wav
```

Out[ ]:

0:00 / 0:04

In [ ]:
```
#读取音频
import numpy as np
import scipy.io.wavfile as wav
sr, signal = wav.read('demo.wav')
print('采样率: ', sr)
print("信号长度: ", signal.shape)
```

```
采样率:  8000
信号长度:  (32000,)
```

In [ ]:
```
# 下面是一些会用到的音频处理函数
from speechlib import *
import os
```

实验1-1：基于双门限法的端点检测 需要补充下列方法: STEn、STZcr、vad_TwoThr

所有需要补充的代码都注释有TODO, 按要求补充即可【未选择该实验】

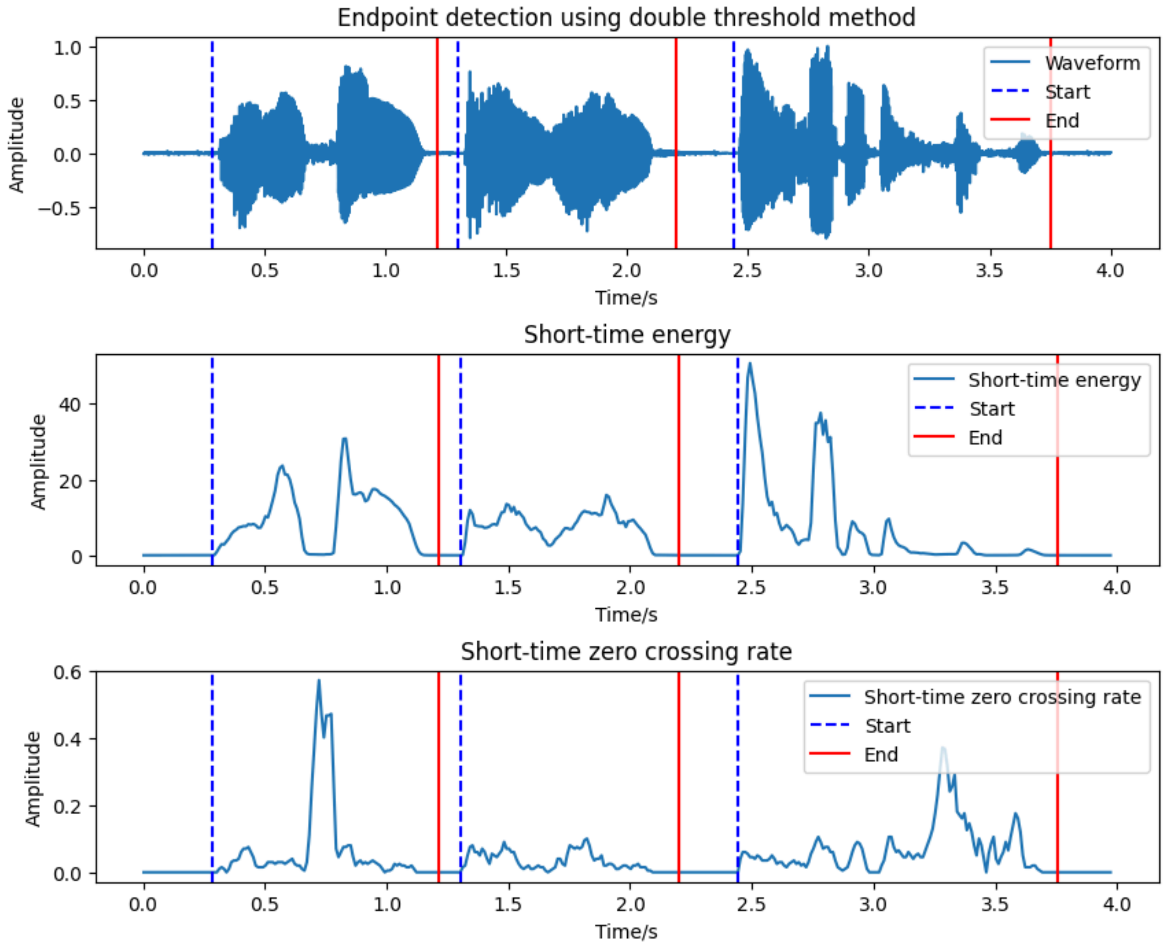
In [ ]:
```
def STEn(x: object, win: object, inc: object) -> object:
    """
    计算短时能量函数
    :param x: 语音信号
    :param win: 窗长
    :param inc: 帧移
    :return: 短时能量，即每一帧的能量
    """
    # TODO 需要补充
    # 1.分帧
    # 2.计算短时能量
    # 3.返回短时能量

def STZcr(x, win, inc, delta=0):
    """
    计算短时过零率
```

```
    :param x: 语音信号
    :param win: 窗长
    :param inc: 帧移
    :return: 短时过零率，即每一帧的过零率
    """
    # TODO 需要补充
    # 1.分帧
    # 2.计算短时过零率
    # 3.返回短时过零率

def vad_TwoThr(x, wlen, inc, NIS):
    """
    使用门限法进行端点检测
    :param x: 语音信号
    :param wlen: 分帧长度
    :param inc: 帧移
    :param NIS: 用于计算门限的无话段长度
    :return: 语音段的起始点和终止点，语音段数，语音段标记，非语音段标记，短时能量
    """
    maxsilence = 15
    minlen = 5
    status = 0
    y = # TODO 需要补充：调用enframe方法进行分帧
    fn = y.shape[0] # 帧数
    amp = STEn(x, wlen, inc) # 计算短时能量，需要在上方方法定义里补充
    zcr = STZcr(x, wlen, inc, delta=0.01) # 计算短时过零率，需要在上方方法定义里
    ampth = np.mean(amp[:NIS]) # 计算短时能量的平均值
    zcrth = np.mean(zcr[:NIS]) # 计算短时过零率的平均值
    amp2 = 2 * ampth
    amp1 = 4 * ampth
    zcr2 = 2 * zcrth
    xn = 0
    count = np.zeros(fn)
    silence = np.zeros(fn)
    x1 = np.zeros(fn)
    x2 = np.zeros(fn)
    for n in range(fn):
    # 0-静音，1-可能开始，2-确定开始，3-语音段

        if status == 0 or status == 1:
            if amp[n] > amp1: # 语音段的起始点
                x1[xn] = # TODO 需要补充
                status = 2
                silence[xn] = 0
                count[xn] += 1
            elif amp[n] > amp2 or zcr[n] > zcr2: # 可能开始
                status = 1
                count[xn] += 1
            else: # 静音
                status = 0
                count[xn] = 0
                x1[xn] = 0
                x2[xn] = 0

        elif status == 2:
            if # TODO 需要补充
                count[xn] += 1
            else:
                silence[xn] += 1
                if silence[xn] < maxsilence:
```

```
                    count[xn] += 1
                elif count[xn] < minlen:
                    status = 0
                    silence[xn] = 0
                    count[xn] = 0
                else:
                    status = 3
                    x2[xn] = x1[xn] + count[xn]
        elif status == 3:
            status = 0
            xn += 1
            count[xn] = 0
            silence[xn] = 0
            x1[xn] = 0
            x2[xn] = 0
    el = len(x1[:xn])
    if x1[el - 1] == 0:
        el -= 1
    if x2[el - 1] == 0:
        print('Error: Not find endding point!\n')
        x2[el] = fn
    SF = np.zeros(fn)
    NF = np.ones(fn)
    for i in range(el):
        SF[int(x1[i]):int(x2[i])] = 1 # 语音段标记
        NF[int(x1[i]):int(x2[i])] = 0 # 非语音段标记
    voiceseg = findSegment(np.where(SF == 1)[0]) # 找到语音段
    vsl = len(voiceseg.keys())
    return voiceseg, vsl, SF, NF, amp, zcr # 返回语音段的起始点和终止点，语音段数
```

In [ ]:
```
(fs, data) = wavfile.read('demo.wav') # 读取音频
data1 = data/np.max(np.abs(data))     # 幅值归一化

N = len(data1)
wlen = 200
inc = 80
IS = 0.1
overlap = wlen - inc
NIS = int((IS * fs - wlen) // inc + 1)
fn = (N - wlen) // inc + 1

frameTime = FrameTimeC(fn, wlen, inc, fs) # 计算每帧的时间刻度
time = [i / fs for i in range(N)]

voiceseg, vsl, SF, NF, amp, zcr = vad_TwoThr(data1, wlen, inc, NIS) # 语音段检测

# 打印语音段信息
fig, axs = plt.subplots(3, 1, figsize=(10, 8))  # Adjust figsize as needed 调整
plt.subplots_adjust(hspace=0.5)  # Adjust hspace for vertical spacing between su

axs[0].plot(time, data1)
axs[0].set_title('Endpoint detection using double threshold method')
axs[0].set_ylabel('Amplitude')
axs[0].set_xlabel('Time/s')

axs[1].plot(frameTime, amp)
axs[1].set_title('Short-time energy')
axs[1].set_ylabel('Amplitude')
axs[1].set_xlabel('Time/s')
```

```
axs[2].plot(frameTime, zcr)
axs[2].set_title('Short-time zero crossing rate')
axs[2].set_ylabel('Amplitude')
axs[2].set_xlabel('Time/s')

# 画出语音段的起始和终止点
for i in range(vsl):
    nx1 = voiceseg[i]['start']
    nx2 = voiceseg[i]['end']

    axs[0].axvline(frameTime[nx1], np.min(data1), np.max(data1), color='blue', l
    axs[0].axvline(frameTime[nx2], np.min(data1), np.max(data1), color='red', li
    axs[0].legend(['Waveform', 'Start', 'End'])

    axs[1].axvline(frameTime[nx1], np.min(amp), np.max(amp), color='blue', lines
    axs[1].axvline(frameTime[nx2], np.min(amp), np.max(amp), color='red', linest
    axs[1].legend(['Short-time energy', 'Start', 'End'])

    axs[2].axvline(frameTime[nx1], 0, 1, color='blue', linestyle='--')
    axs[2].axvline(frameTime[nx2], 0, 1, color='red', linestyle='-')
    axs[2].legend(['Short-time zero crossing rate', 'Start', 'End'], loc='upper

os.makedirs('figs', exist_ok=True) # 创建文件夹
plt.savefig('figs/vad.png') # 保存图片
plt.show()
```



实验1-2：基于自相关法的基音周期检测 需要补充下列方法: pitch_Corr【已补充✅】

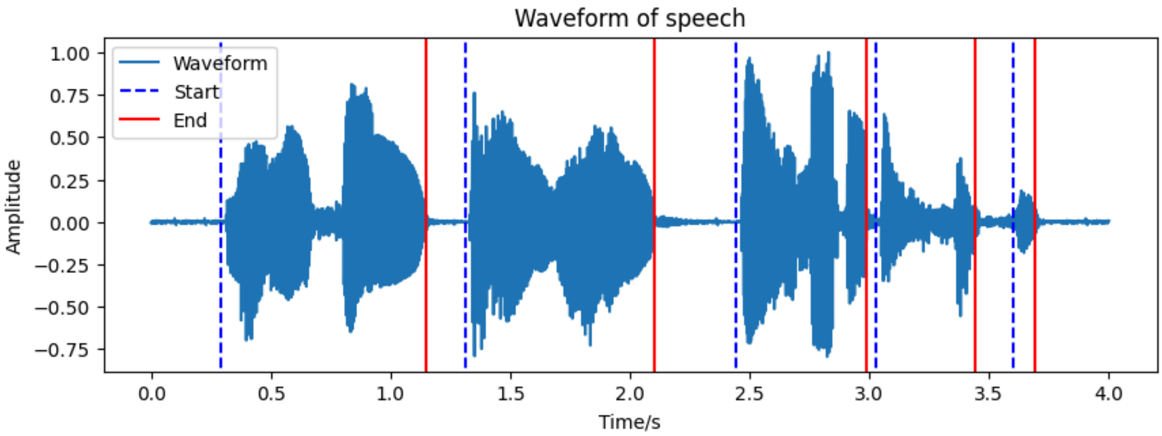
In [ ]:
```
def pitch_Corr(x, wnd, inc, T1, fs, miniL=10):
    """
    自相关法基音周期检测函数
```

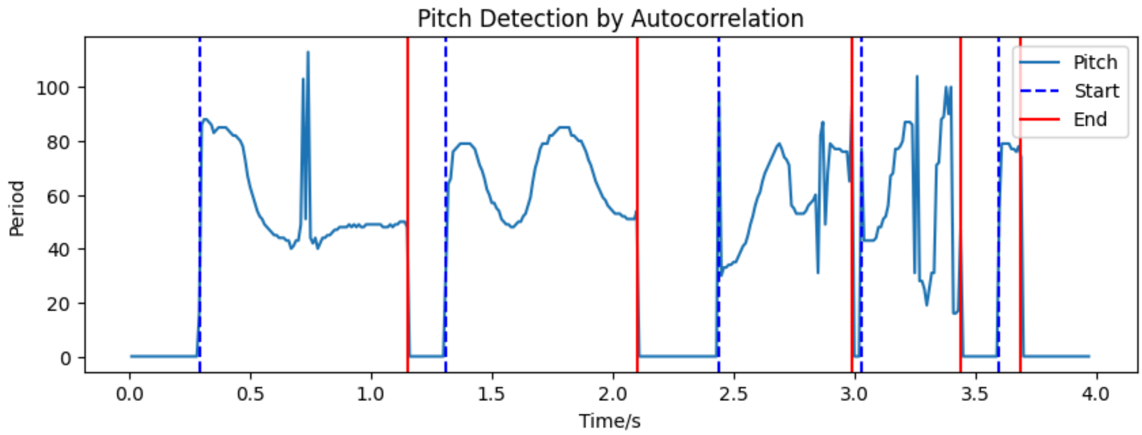
```
    :param x: 语音信号
    :param wnd: 窗函数或窗长
    :param inc: 帧移
    :param T1: 门限
    :param fs: 采样率
    :param miniL: 语音段的最小帧数
    :return voiceseg, vsl, SF, Ef, period: 语音段的起始点和终止点，语音段数，语音
    """
    y = enframe(x, wnd, inc)    # 调用enframe方法进行分帧
    fn = y.shape[0]
    if isinstance(wnd, int):
        wlen = wnd
    else:
        wlen = len(wnd)
    voiceseg, vsl, SF, Ef = pitch_vad(x, wnd, inc, T1, miniL)  # 语音分段
    lmin = fs // 500  # 基音周期的最小值
    lmax = fs // 60  # 基音周期的最大值
    period = np.zeros(fn)
    for i in range(vsl):    # 在所有语音段中
        ixb = voiceseg[i]['start'] # 语音段的起始帧
        ixd = voiceseg[i]['duration'] # 语音段的帧数
        for k in range(ixd):
            ru = np.correlate(y[ixb + k, :], y[ixb + k, :], mode='full')   # 计算
            ru = ru[wlen:]
            tloc = np.argmax(ru[lmin:lmax])
            period[ixb + k] = tloc + lmin
    return voiceseg, vsl, SF, Ef, period
```

In [ ]:

```
import numpy as np
from scipy.io import wavfile
import matplotlib.pyplot as plt

# 读取WAV文件
(fs, data) = wavfile.read('demo.wav')

# 去除直流偏移
data = data - np.mean(data)

# 幅值归一化
data = data / np.max(data)

# 分析参数
wlen = 320  # 分析窗口长度
inc = 80    # 连续窗口间隔
N = len(data)
time = [i / fs for i in range(N)]  # 时间向量
T1 = 0.05  # 用于基音校正的阈值

# 进行基音校正并检测有声段
voiceseg, vsl, SF, Ef, period = pitch_Corr(data, wlen, inc, T1, fs)

# 计算帧数
fn = len(SF)

# 计算帧时间位置
frameTime = FrameTimeC(fn, wlen, inc, fs)

# 设置子图布局
fig, axs = plt.subplots(2, 1, figsize=(10, 8))
plt.subplots_adjust(hspace=0.5)  # 调整垂直间距
```

```
# 绘制波形图
axs[0].plot(time, data)
axs[0].set_title('Waveform of speech')
axs[0].set_ylabel('Amplitude')
axs[0].set_xlabel('Time/s')

# 绘制自相关基音周期检测图
axs[1].plot(frameTime, period)
axs[1].set_title('Pitch Detection by Autocorrelation')
axs[1].set_ylabel('Period')
axs[1].set_xlabel('Time/s')

# 标记有声段
for i in range(vsl):
    nx1 = voiceseg[i]['start']
    nx2 = voiceseg[i]['end']

    # 在波形图上标记有声段
    axs[0].axvline(frameTime[nx1], np.min(data), np.max(data), color='blue', lin
    axs[0].axvline(frameTime[nx2], np.min(data), np.max(data), color='red', line

    # 在自相关基音周期检测图上标记有声段
    axs[1].axvline(frameTime[nx1], np.min(period), np.max(period), color='blue',
    axs[1].axvline(frameTime[nx2], np.min(period), np.max(period), color='red',

axs[0].legend(['Waveform', 'Start', 'End'])
axs[1].legend(['Pitch', 'Start', 'End'])

os.makedirs('figs', exist_ok=True) # 创建文件夹
plt.savefig('figs/pitch.png') # 保存图片
plt.show()
```





实验1-3：MFCC提取 需要补充下列方法: Nmfcc【已补充✅】

In [ ]:
```
def Nmfcc(x, fs, p, frameSize, inc, nfft=512, n_dct=12):
    """
    计算mfcc系数
    :param x: 输入信号
    :param fs: 采样率
    :param p: Mel滤波器组的个数
    :param frameSize: 分帧的每帧长度
    :param inc: 帧移
    :return: mfcc系数
    """
    # 预加重处理
    x_preemphasized = lfilter([1, -0.9375], [1], x)

    # 分帧
    frames = enframe(x_preemphasized, frameSize, inc)

    # 加窗
    frames = np.multiply(frames, np.hanning(frameSize))

    # 计算FFT
    fft_result = np.fft.rfft(frames)

    # 计算能量谱
    power_spectrum = np.abs(fft_result) ** 2

    # 计算通过Mel滤波器的能量
    bank = melbankm(p, nfft, fs, 0, 0.5 * fs, 0)
    ss = np.matmul(power_spectrum, bank.T)
```

```
    # 计算DCT倒谱
    M = bank.shape[0] # 滤波器个数
    m = np.array([i for i in range(M)])
    mfcc = np.zeros((ss.shape[0], n_dct)) # 初始化mfcc系数
    for n in range(n_dct):
        mfcc[:, n] = np.sqrt(2 / M) * np.sum(np.multiply(np.log(ss), np.cos((2 *
    return mfcc
```

In [ ]:

```
import librosa
import matplotlib.pyplot as plt
from scipy.io import wavfile
import numpy as np

# 读取WAV文件
(framerate, wave_data) = wavfile.read("demo.wav")

# 参数设置
wlen = 256
inc = 128
num = 8
nfft = 256
n_dct = 24

# 归一化处理
x = wave_data / max(np.abs(wave_data))
time = np.arange(0, len(wave_data)) / framerate

# 绘制原始波形
plt.figure(1)
plt.subplot(411)
plt.plot(time, x, 'b')
plt.title("(a) Waveform")
plt.ylabel("Amplitude")
plt.xlabel("Time/s")

# 计算MFCC特征
ccc1 = librosa.feature.mfcc(y=x,
                            n_fft=wlen,
                            sr=framerate,
                            n_mfcc=24,
                            fmax=4000,
                            dct_type=2,
                            hop_length=inc,
                            win_length=wlen)
ccc2 = np.transpose(ccc1)

# 进行NMFCC计算
ccc1 = Nmfcc(x, framerate, num, wlen, inc, nfft, n_dct)
fn = ccc1.shape[0]
cn = ccc1.shape[1]
frameTime = FrameTimeC(fn, wlen, inc, framerate)

# # 计算语谱图
D = librosa.stft(x, n_fft=wlen, hop_length=inc)
D_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
# 绘制语谱图
plt.subplot(412)
librosa.display.specshow(D_db, sr=framerate, hop_length=inc, x_axis='time', y_ax
plt.title("(b) Spectrogram")
```

```
plt.colorbar(format='%+2.0f dB')

# 绘制MFCC系数
plt.subplot(413)
plt.plot(frameTime, ccc1[:, 0:int(cn/2)])
plt.title("(c) MFCC Coefficients")
plt.ylabel("Amplitude")
plt.xlabel("Time/s")


# 绘制MFCC特征图
plt.subplot(414)
plt.imshow(ccc1, cmap='hot', interpolation='nearest', aspect='auto')
plt.xlabel('Frame')
plt.ylabel('MFCC Coefficient')
plt.title('(d) MFCC Features')
plt.colorbar(label='Magnitude')

# 调整子图间距
plt.subplots_adjust(hspace=2)

os.makedirs('figs', exist_ok=True) # 创建文件夹
plt.savefig('figs/mfcc.png') # 保存图片
plt.show()
```