



链滴

开源的自动剪辑视频工具——跳跃剪辑 (JumpCutter)

作者: [HaujetZhao](#)

原文链接: <https://ld246.com/article/1590476704142>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

这是一个 python 写的小工具，我下载 [jumpcutter](#) 下后，修补了一些 bug，并加入了自动转字幕和动剪辑的功能，成了现在的版本。

不过我也是不专业程序员，目前也没时间学 Github，也懒得写英文 Readme 了，代码也简单，就发这里，看下有没有志愿者，能把这个程序发到 Github，进行维护。

工具源代码在这里：[自动跳跃剪辑JumpCutter.zip](#)

为了方便理解这个工具，我录制了一个视频，将自动剪辑前后的两个版本放到这里，供对比下：

- 自动剪辑后：[剪辑后new.mp4](#)
- 自动剪辑前的原视频：[剪辑前new.mp4](#)

因为使用的笔记本自带的麦克风收音，录音时风扇又在大转，所以会有些杂音，请见谅。

这个教程的两个版本，一个是经过了本工具剪辑后的效果，一个是原始视频。经过自动剪辑后，视频精炼，表达更好。

建议上面两个视频对比看一下，就能理解该工具有多强大。

所谓的自动剪视频有两个方面：

1. 根据音频大小，分出静音和有声音部分。将静音部分加速到比如 999 倍，就相当将静音片段剪了了。剩下的就是有声音、信息量较多的视频部分。

2. 在拍视频时，比如：

在 01:00 时，我说了声“保留”；在 01:30 时，我说了声“删除”；在 02:00 时，我说了声“保留”；视频拍摄到了 02:30。

对视频使用脚本，通过阿里云 api，将音频转成字幕，设置保留关键词为“保留”、保留切掉的关键为“删除”，在上述拍摄的时长 02:30 的视频中：

- 00:00 ~ 01:00、01:30 ~ 0 2:00、02:00 ~ 02:30 这三个片段会被保留
- 01:00 ~ 01:30 这个片段会被删掉
- 包含关键词“保留”、“删除”的片段，也会被删掉

这样，我就可以在拍视频时决定哪些片段要 cut 掉，比如有一句话说错了，只要说一声“删除”，这一小段就会在后期自动被删掉（完全不用手动 cut），要确保刚刚这一小段保留下来，只要说一声“保留”就行。

在拍视频时就可以决定要保留哪些片段、删除哪些片段，很 cool 不是吗？

也正是由于这个功能需要知道视频音频说了什么，才需要使用阿里云的录音文件识别 api，将音频转带时间戳的字幕。

通过联合使用上述两个方法，就可以自动剪出一个非常棒的 vlog、视频教程（因为这类视频中，一般有讲话的时候才有信息量，而因为我们大脑反应速度的限制，视频中会有不少空白的时间，我们不是说话，而是在想下一句说什么）

甚至，只使用将静音片段加速的方法，就能得到不错的结果，比如，将 15 分钟的视频，自动剪辑成 8 分钟，而且不损失任何重要信息！

我能想到最好的用处就是，对于程序员，可能

- 想通过视频给自己的项目、创意一个好的介绍

- 自己语言能力不大行，上镜后，录出的视频有许多不满意的片段
- 不大会、也不想剪视频

就可以用这个工具，粗糙、肆意、随便地录制一个教程视频后，用该工具处理一下，仅需一条命令，能生成一个信息量高、精炼的视频教程。

中文帮助

简介

JumpCutter 是一个 python 脚本。

它可以对视频中 **有声音** 和 **没声音** 的部分施以 **不同的播放速度**。

原作者是 karykh，但是有一些 bug。 [Github_jumpcutter](#)

本版本修补了各个已知 bug，加入了：

- 使用阿里云、腾讯云 API 自动转 srt 字幕
- 根据字幕自动剪辑

它现在可以根据 **srt字幕文件** 中的关键词 **自动剪辑**（也就是 **删除** 和 **保留**） 视频片段。

只有当有 srt 输入文件的时候，才会做自动剪辑。

阿里云 API 前 3 个月免费每天 2 小时，之后，按 2.5 元/小时收费。

你可以到 <https://www.bilibili.com/video/av97093907/> 查看它的工作原理。p4 视频是 jumpcutter 原作者的视频搬运。

下面是选项的帮助：

- input_file inputFILE 指定一个输入的视频文件
- url URL 如果要处理的输入文件是一个 YouTube 在线视频，就用这个选项，输入 URL
- input_subtitle SUBTITLE 如果要依据字幕来自动剪辑，就输入字幕文件路径，要求 srt 字幕
- cut_keyword 字幕中的关键字，这是切除片段的关键词。默认是“咔嚓”
- save_keyword 字幕中的关键字，这是保留片段的关键词。默认是“保留”
- output_file 输出视频文件路径，可选，如果没有选这个，会有默认自动的输出文件名
- silent_threshold 静音阈值，低于多少的音量可以被认为是静音（取值在 0 - 1 之间，是相对于个音频中的最大音量的相对值，不是绝对值。）
- sounded_speed 有声音部分的速度，默认是 1.00
- silent_speed 没有声音部分的速度，默认是 5.00
- frame_margin 加速留白。就是在静音区间两端留几帧，不要加速，防止音频没有停顿。默认是 1

帧

--sample_rate 音频采样率，默认是 44100，目前程序不支持自动获得采样率，如果你的采样率不这个值，需要手动填写

--frame_rate 视频帧速，默认30，一般会自动识别，不用管。如果帧速出了问题，再来这里调节

--frame_quality 帧质量。处理时，会先把视频中的每一帧提取出来，保存到 jpg 格式的图片，这选项决定了保存图片的质量。取值在 1 - 31 之间，1 代表质量最高，31代表质量最差。默认是3

--online_subtitle 是否要使用在线云服务转字幕，可选项有：1, 0。默认是 0，也就是否

--subtitle_language 选择在线识别字幕的语言。可选项有："Chinese", "English"。默认是 "Chinese"

--cloud_engine 用哪个语音识别引擎。可选项有："Alibaba"、"Tencent"。默认是"Alibaba"

--delete_cloud_file 识别完成后，是否删除保存在云端的音频文件。可行项有：True, False。默认 True

安装

脚本嘛，直接 copy 下来，装上 FFmpeg、python、需要的库

一条使命装需要的库：

```
pip install -r requirements.txt## 使用
```

查看帮助：

```
python jumpcutter.py -h
```

示例1：

仅把视频静音片段加速，所有参数为默认值，输出文件为"我的vlog_new.mp4"

```
python jumpcutter.py --input_file "我的vlog.mp4"
```

示例2：

仅把视频静音片段加速，手动设置参数

```
python jumpcutter.py --input_file "视频.mkv" --silent_speed 99 --sounded_speed 1 --frame_margin 2 --silent_threshold 0.025 --output_file "视频.mp4"
```

示例3：

把视频静音片段加速，手动设置参数，并自动转字幕，利用字幕中的关键词自动剪辑

```
python jumpcutter.py --input_file "视频.mkv" --online_subtitle 1 --subtitle_language "Chinese" --cloud_engine "Alibaba" --delete_cloud_file True --cut_keyword "删除" --save_keyword "保留" --silent_speed 8 --sounded_speed 1 --frame_margin 2 --silent_threshold 0.025 --output_file D:\视频_剪辑后.mp4"
```

示例4:

把视频静音片段加速，手动设置参数，并自动转字幕，利用字幕中的关键词自动剪辑。这是在 Windows cmd 窗口，加了换行符，更好看点

```
python jumpcutter.py ^
--input_file "视频.mkv" ^
--online_subtitle 1 ^
--subtitle_language "Chinese" ^
--cloud_engine "Alibaba" ^
--delete_cloud_file True ^
--cut_keyword "删掉" ^
--save_keyword "保留" ^
--silent_speed 8 ^
--sounded_speed 1 ^
--frame_margin 2 ^
--silent_threshold 0.025 ^
--output_file "D:\视频_剪辑后.mp4"
```

视频转字幕引擎配置

打开 `moduels` 文件夹，打开 `CloudEngine.ini` 文件，按里面说的填好 api 相关信息。

毕竟，你不填好 api，怎么用阿里云/腾讯云的语音识别服务呢，对吧？

；在这里存放云引擎的参数

；oss 管理控制台用于管理 bucket，bucket 就是你的云文件存放的地方，我们要先将音频文件上传这个云空间，才能识别。所以你需要在阿里云开发者控制台开通 oss 管理控制台服务，再在控制台中建一个 bucket。新建好之后，将这个 bucket 的权限设为“公共读”（只有将权限设为“公共读”你的“智能语音交互”服务才能访问这里的录音文件），记下它的：外网访问EndPoint、bucket 字

；再在控制台搜索“智能语音交互”，开通服务后，新建两个项目，一个识别语言为中文，一个为英文，音频采样率为 16k，保存。记下这两个项目的：appkey

；上面两个服务弄好后，还用不了，因为我们没权限！所以我们要开通在主账号下开通一个能操控这个服务的用户，搜索“RAM访问控制”，开通服务，在控制面板，创建一个用户，记下它的：“AccessKey ID”、“AccessKey Secret”，并给这个用户添加以下两个权限：“管理对象存储服务（OSS 权限”、“管理智能语音交互（NLS）的权限”

[Alibaba RAM Access Control User]

；这里存放阿里巴巴 RAM 访问控制中有权限的用户密钥

yourAccessKeyId = xxxxxxxxxxxxxxxxxxxxxxxxxxxx

yourAccessKeySecret = xxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Alibaba Cloud OSS]

；这里存放阿里巴巴 oss 智能对象的设置

yourEndpointDomain = xxxxxxxxxxxxxxxxxxxxxxxxxxxx

yourBucketName = xxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Alibaba Cloud Chinese Transcribe Engine]

；中文项目的appKey

appKeyForChinese = xxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Alibaba Cloud English Transcribe Engine]

; 英文项目的appKey

appKeyForEnglish = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Tencent Cloud Sub User]

; 这里存腾讯有权限的子用户密钥

yourSecretId = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

yourSecretKey = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

[Tencent Cloud OSS]

; 这里存放腾讯 oss 智能对象的设置

yourBucketDomain = https://audio-recognition-xxxxxxxxxxxxxxxxxxxxx.myqcloud.com

yourBucketName = audio-recognition-xxxxxxxxxxx

yourBucketRegion = ap-xxxxxxxxx

yourEndPoint = cos.ap-xxxxxxxxx.myqcloud.com