

# 信号处理原理 SRT 影视分割

## ——“豆包影视分割系统”

### 终结报告

付钊 陈俊男

# 目录

目录.....	2
摘要.....	3
介绍.....	3
• 画面方面.....	3
• 音频方面.....	4
• 人性化.....	4
基础技术及其原理分析.....	5
音频特征 mfcc.....	5
音频模型 GMM.....	5
聚类 k-means.....	5
颜色分布直方图.....	5
灰度图的边缘检测.....	6
算法流程.....	6
流程图.....	6
基于图像特征进行初步分割.....	6
基于音频特征合并分割点.....	7
综合图像和音频进行迭代合并.....	8
实验结果及分析.....	8
初步确定算法阶段.....	8
查文献阶段.....	9
最终结果.....	9
GUI 成果展示.....	11
人性化用户界面.....	11
研究过程中遇到的问题.....	14
初步确定算法.....	14
查文献.....	15
算法更新 1.....	15
算法更新 2.....	15
引入说话时间段判定.....	15
算法更新 3.....	16
算法更新 4.....	16
实验总结.....	16
鸣谢.....	16

## 摘要

本报告叙述了“豆包影视分割系统”。这是一个具有将影视剧自动按场景分割的视频播放系统，通过这个系统，用户可以：

- 任意给出影视作品，进行处理。
- 清晰的影视作品的情境变化。
- 根据影视分割的信息，自动切分出有主要演员的戏段。
- 通过一定训练，识别出演员的情绪（如笑）。
- 通过交互界面，迅速找到想看的部分。

在目前，做影视分割，尤其是影视剧的影视分割的成果几乎为零，这给我们的造成了一些困难，我们没有现成的算法思路可以循照。

为了达到影视分割的结果，我们做了许多尝试，最终，我们决定使用视频特征与音频特征相互结合的方法，经过训练、调试，最终，我们的系统分割的准确率较高。

“豆包影视分割系统”的另一大重要功能是与用户的交互，我们的 GUI 设计也提供了一种与用户交互的思路。对于每一个分割点，用户可以通过单击，查看该段情景的截图，从而判断是否要观看这段视频。

总的来说，我们的算法达到了较好的效果，在速度上仍有提升空间<sup>①</sup>（本系统使用的是 MATLAB，如果使用 PYTHON 重写，那么速度会有所提升）本系统的 GUI 提供了用户交互方式的基础思路，日后仍可以继续完善。

## 介绍

对于视频分割系统，一项重要的判断依据就是其将影视作品，分割成有意义的、逻辑清楚的、人类可接受的片段的能力。

### • 画面方面

从画面来看，一种分割片段是一段镜头（shot），或是一组由同一摄像角度拍摄的分镜。在大部分影视作品中，这种完整的镜头或分镜会被不同种类的边界、转换分隔开。简单的边界就是一组在独立帧层面的场景转换，这个可以较为容易的判断出来。而常规的转换更为复杂，它是逐渐的转换，我们把这种逐渐的转换一般分为两种类型：渐退（fade）和溶解（dissolve）。实际上，当我们考虑使用这样的模型将视频切分开时，不难发现许多镜头的变换，都会被错误的分割成切割的转换。

为了尽可能的提高准确率，避免这种错误的切割，我们要找到镜头的边界，一项重要的方法是帧的基于图形的距离计算，当距离超过某阈值时，记录这个变换点。临近帧距离的计算可以基于像素特性的统计计算，压缩算法，和边缘检测。我们使用的统计方法是基于直方图变化、加以边缘检测变化的方法。具体的检测算法会在下文中进行详细的阐述，目前可以发现的结果是基于直方图的变化

测是最为准确的。其结果比直接进行帧像素对比要科学很多。

## • 音频方面

经过以上计算，我们可以得到整部影视作品几乎所有的分镜及场景切换。找到有价值的切分点，我们需要依靠的主要是在音频方面的处理了。

对声音进行预处理，我们需要提取 MFCC，使用 GMM 模型、K-means 模型等作为主要方法，对音频模型进行分类。相对于声纹识别等研究，影视剧的音频处理显得更为复杂。主要原因有以下几点：

- ① 非纯人声，提取不准确；
- ② 背景音乐与噪声多，需要加以区分；
- ③ 样本容量有限，很难进行有效的训练。

经过一系列测试，我们最终还是使用 MFCC，使用 GMM 计算距离，运用直方图的方法。具体方法下文详细阐述，系统可以将音频分为安静、有人对话、背景音乐、噪声等模式，我们使用这些信息来鉴别基于图像特征直方图的镜头的边界，将不正确的分割点筛去。这是系统中最为重要的一项工作，由于基于图像特征，在一集影视剧中，系统会计算出两百个左右的分镜，我们需要准确的筛选算法，将有价值的分割点保留下来（一般十个左右），筛选率为 3%~5%。

## • 人性化

本算法注重“人性化”，将视频与音频的信息处理结合，不完全依赖帧变换、声音变换，使得切割点尽量人性化。因为视频是分给人看的，只有基于人的思维的分割，才是有价值的分割点。

在设计算法方面，我们将人性化思维尽可能的加入算法当中，并将之转化为可量化的度量方式。我们通过观看大量的影视剧（重复观看，着重分析），对影视剧的拍摄手法进行了分析：

例如，即使分镜切换处于高频，但是演员的台词保持一定程度的连续，这种情况下情景是连续，并不是分割。

再如，情景发生转换，绝大多数情况伴随前后三秒内没有人说话。

再如，没有演员对话的场景切换，不是主要表达重点。

基于这样的考虑，我们在算法设计上，将这些性质带入了切分点的筛选，例如将没有人说话，量化为判断前后  $n$  秒内的 MFCC-Distance 等。

最终，本算法在分析国产影视剧上，切割断点的效果较好，我们将结果与优酷网、爱奇艺网网站上人工对于视频的断点分割进行了对比。结果表明，我们的切割点涵盖了视频网站上所有的人工切割点，一般比网站上的分割点多 2-3 个，召回率和准确率都有不错的表现。

# 基础技术及其原理分析

## 音频特征 mfcc

- 定义:

mfcc 即梅尔频谱倒谱系数,是将一段音频时域信号转化成频域信号后经过梅尔滤波器滤波得到的频谱的倒谱系数。

- 特点

梅尔频率倒谱的优势在于,其频带划分是在梅尔刻度上等距划分的,它比用于正常的对数倒频谱中的线性间隔的频带更能近似人类的听觉系统,因此这种频率弯曲可以更好的表示声音。

- 应用原因

在视频分割中,算法要模拟人的思维对视频进行分割,而人在对视频情节进行分割时,大多会根据两段音频中的语音是否相似来判断是否属于同一情景,因此在音频处理的过程中使用 mfcc 能够更好地模拟人对语音的识别,从而起到更好的效果。

## 音频模型 GMM

- 定义

GMM 即混合高斯模型,是统计学中的一种对数据建模的方法,通过多个高斯模型可以拟合复杂的数据分布。

- 特点

混合高斯模型可以很准确地拟合并表示各种复杂的高维数据分布,在大多数情况下都可以用它进行建模。

- 使用原因

在视频分割中,音频特征 mfcc 是一种复杂的高维分布,很适合用 GMM 拟合。

## 聚类 k-means

- 特点及其使用原因

k-means 是一种常用聚类方法,在给定距离矩阵的情况下使用 k-means 可以高效地进行聚类。在影视分割中,可以用 k-means 对分好段的音频进行聚类以此作为分割的判据。

## 颜色分布直方图

- 特点及其使用原因

如果将图像的颜色 RGB 三通道均匀分成 10 份, 那么色彩空间就会被分成 1000 份。颜色分布直方图就是指用色彩空间表示图像的颜色分布, 例如用 1000 维向量表示一幅图像。颜色分布直方图能够很好地反映出图像的色调以及颜色组成, 在计算视频帧与帧图像间距离时是重要参数之一。

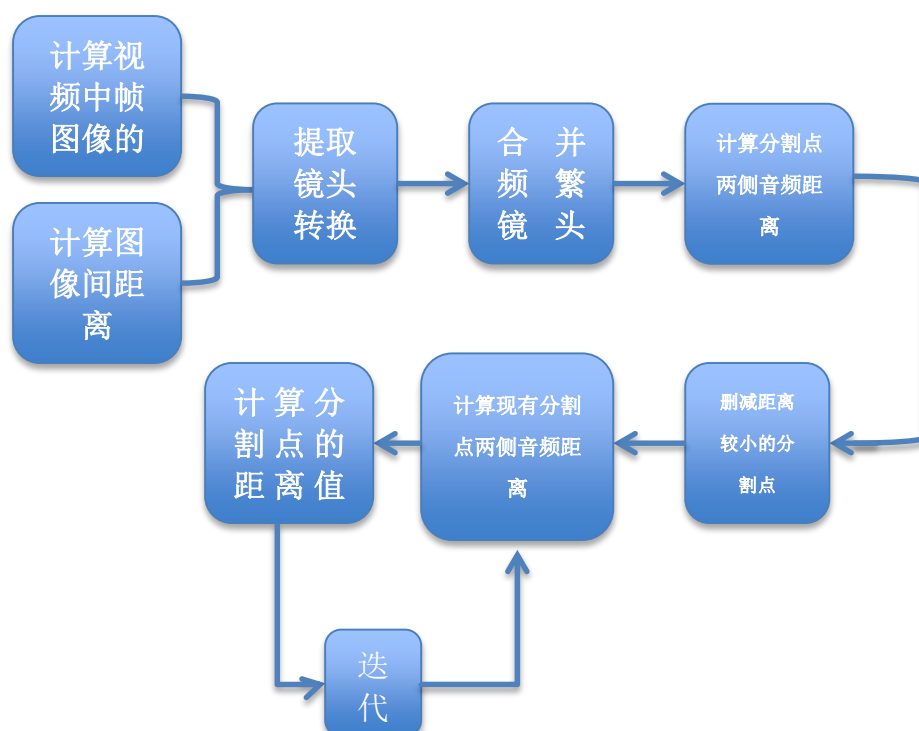
## 灰度图的边缘检测

- 特点及其使用原因

是指将彩图转化为灰度图后对其进行边缘检测, 这样得到的只包含边缘信息的黑白矩阵可以作为图像的另一重要特征信息。边缘特征值能够很好地反映出图像中的物体以及场景中前景的轮廓, 减少光线和背景颜色对图像特征向量的影响, 在计算视频帧与帧图像间距离时是重要的参数之一。

## 算法流程

### 流程图



## 基于图像特征进行初步分割

- 计算视频中帧图像的特征向量

对视频每隔  $\text{timeSlice}$  ( $\text{timeSlice}$  取 0.5s) 时间取出一帧图像  $I$ , 先计算色彩分布直方图得到表示颜色的特征向量  $\text{color\_feature\_origin}$  (1000 维), 再将其转化成灰度图后通过边缘检测技术计算出边缘矩阵, 并将得到的边缘矩阵分成  $\text{patchSize} \times \text{patchSize}$  的块后计算块平均值, 以此得到表示边缘的特征向量  $\text{edge\_feature}$ 。为减小视频分辨率对颜色特征向量的影响, 由

$$\text{color\_feature} = \text{color\_feature\_origin} / (H \times W / 2)$$

得到标准化的颜色特征向量  $\text{color\_feature}$ , 其中  $H$  和  $W$  分别表示图像的高和宽。

#### • 计算图像间距离并提取出镜头转换

计算相邻两帧 (间隔  $\text{timeSlice}$  时间) 图像的颜色特征向量和边缘特征向量的距离

$$\text{dist\_color} = |\text{color\_feature\_i} - \text{color\_feature\_j}|$$

和

$$\text{dist\_edge} = \text{sum}(\text{edge\_feature\_i} - \text{edge\_feature\_j} > \text{TH}_e),$$

其中  $\text{TH}_e$  表示阈值, 取 0.08,  $\text{sum}$  表示对大于阈值的块个数求和。再计算两帧间总距离

$$\text{dist} = \text{dist\_color} * \text{color\_weight} + \text{dist\_edge} * \text{edge\_weight},$$

其中  $\text{color\_weight}$  取 5,  $\text{edge\_weight}$  取 1.5。如果  $\text{dist}$  大于阈值  $\text{TH}_d$  ( $\text{TH}_d$  取 12), 那么将后一帧视为一个镜头转换, 记录所有的镜头转换时间于  $\text{scene\_cut\_time}$  数组。

#### • 合并切换频繁的镜头转换

对镜头转换的每一帧, 比较并得到前 30s 内与之相似度高于阈值  $\text{TH}$  ( $\text{TH}$  取 10) 的最前面的一帧, 合并在这之间的所有镜头转换。

## 基于音频特征合并分割点

#### • 计算现有分割点两侧音频距离

先计算出整个视频的一阶 mfcc 特征向量集, 每 20ms 分成一段。

对每个分割点  $t$ , 分别将  $t-dt$  至  $t$  和  $t$  至  $t+dt$  两段时间的音频对应的 mfcc 特征向量集用 gmm 模型进行建模 (其中  $dt$  取 5s), 再计算两个 gmm 模型间的距离, 记为  $\text{dist\_gmm}$ 。

#### • 删减距离较小的分割点

综合考虑  $\text{dist\_gmm}$  和  $\text{dist\_color}$ , 加权得到

$$\text{dist} = \text{dist\_gmm} + \text{dist\_color} * \text{weight},$$

其中  $\text{weight}$  取 10。对相邻两分割点间时间小于阈值  $\text{TH}_t$  ( $\text{TH}_t$  取 10s) 的所有分割点, 取其中  $\text{dist}$  值最大的分割点并保留, 删减其余分割点。

## 综合图像和音频进行迭代合并

- 计算现有分割点两侧音频距离

对每个分割点  $t$ ，分别将上个分割点到当前分割点和当前分割点到下个分割点两段时间的音频对应的 mfcc 特征向量集用 gmm 模型进行建模后，计算两个 gmm 模型间的距离，记为  $\text{dist\_gmm\_slice}$ 。

- 计算分割点的距离值并进一步删减分割点

综合考虑  $\text{dist\_gmm\_slice}$ 、 $\text{dist\_gmm}$  和  $\text{dist\_color}$ ，加权得到  $\text{dist\_value} = \text{dist\_gmm\_slice} + \text{dist\_gmm} + \text{dist\_color} * \text{weight}$ ，其中  $\text{weight}$  取 10。如果分割点的距离值  $\text{dist\_value}$  大于阈值  $\text{TH}_v$  ( $\text{TH}_v$  取 185)，则保留，否则删减该分割点。

- 迭代

重复步骤 a 和 b，直到没有分割点被删除为止。

## 实验结果及分析

### 初步确定算法阶段

对某一集老友记音频进行 mfcc 提取和聚类后，得到如下结果：

音频时长：56 秒；聚类数：8

类别如下：

- ① 空指针（视频开始时）
- ② 女性轻叹声
- ③ Ross 的一句（男性）
- ④ Joey 的台词（男性）
- ⑤ 所有的笑声
- ⑥ Phoebe 和 Rachael 的台词（女性，这两位女演员的声音比较接近）
- ⑦ 所有的对话开始与结尾（一半为人声，一般为环境音）
- ⑧ Ross 的大部分台词（男性）

断点如下：

我们的程序将音频按照特征向量判断出视频场景发生转折的断点。由程序自动分段。从结果来看，部分段落区分度还可以，部分段落的区分度不是很高。在我们使用的样例中，断点位置大概有：

- ① 男女演员对话的交接处（4处）
- ② 从环境音到演员开始说话的交接处（2处）
- ③ 从演员说完话到开始环境音的交接处（3处）

断点的成功率为  $9/16$ ，即 56.25%。

总的来说，本阶段实验的效果比较明显，在聚类中，由于我们是按








照每秒作为单位区分，因此，如果这一秒钟的内容不由同一场景构成，那么将会生成不同的特征。因此，我们提取出了对话的开始时刻，这在实际运用中非常有用。我们还提取出了所有的笑声，也就是说判断出了“笑点”的位置，在实际运用中也比较实用。

但即便如此，我们并不能让程序自动判断并给每一类加上标签，必须人工听过才能确定每一类分别是什么，这对于编程实现视频分割并没有起到什么作用，因此算法必须改进。

## 查文献阶段

经过无数关键词的尝试，查到 5 篇内容相关度较高的文献：

 audio and video clues.pdf	2013/12/9 18:47	Adobe Acrobat ...	394 KB
 audio based speech detection.pdf	2013/12/9 12:57	Adobe Acrobat ...	625 KB
 content analysis.pdf	2013/12/9 15:02	Adobe Acrobat ...	261 KB
 features models and time scales.pdf	2013/12/9 15:00	Adobe Acrobat ...	394 KB
 HMM for video segmentation.pdf	2013/12/9 14:58	Adobe Acrobat ...	415 KB

## 最终结果

我们对热播电视剧《咱们结婚吧》中的第 8 集前 10 分钟进行了透彻的分析后，人工设置规则集并确定了算法中的各个阈值和权重参数，并对第 1 集到第 9 集进行了测试(其中第 1 集到第 7 集分为上下集，分别用 \_1 表示上集、\_2 表示下集)，结果如下：

### • 单集结果(详见附件 outcome.xlsx)

例：第 7 集上下集结果

	A	B	C	D	E	F	G	H
1	time				time			
2	0:7	r	right	8	1:2	w(应该0:54)	right	14
3	1:17	r	wrong	7	2:12	r	wrong	4
4	5:38	r	notcut	1	5:11	w	notcut	0
5	6:56	r	precisior	0.533333	5:32	w	precisior	0.777778
6	7:48	w	recall	0.888889	5:55	r	recall	1
7	8:29	r			7:45	r		
8	8:49	w			8:17	w		
9	9:30	r			9:1	r		
10	11:36	w			12:50	r		
11	12:32	r			13:12	r		
12	14:1	w			14:30	r		
13	16:37	w			15:23	r		
14	17:49	n			15:52	r		
15	20:40	w			16:6	r		
16	21:7	w			16:40	r		
17	21:48	r			17:1	w		
18					17:26	r		
19					18:17	r		
20								

其中，每个时间点对应的 r 表示分割正确(right)，w 表示分割错误

(wrong), n 表示没分出来(not cut)。分割错误又分为两种情况：多余的分割和相差 10 秒以内的错位分割。实际上，错位分割给观众造成的影响是很小的(也很少出现在我们的结果中)，观众可以看很短的时间就发现真正的分割点。

#### • 错误原因分析：

对于没分出来的情况，一般是因为在根据图像切分镜头时就漏掉了，说明画面过度很柔和，所以没分割出来；对于分割多余的情况，一般是由于在同一个场景中镜头切换前后说话人由男变女或情绪大变造成的，分割出来也在情理之中；对于分割错位的情况，一般是因为真实分割点与错位分割点相距比较近，且错位分割点的分割值大于真实分割点的分割值，所以程序会错把真实分割点删除而保留错位分割点。

#### • 总体结果

Video No.	1_1	1_2	2_1	2_2	3_1	3_2	4_1	4_2
Right	17	14	9	8	9	9	8	9
Wrong	2	1	2	1	5	2	3	2
NotCut	2	1	1	1	0	0	2	1
Precision(%)	89.5	93.3	81.8	88.9	64.3	81.8	72.7	81.8
Recall(%)	89.5	93.3	90.0	88.9	100	100	80.0	90.9

Video No.	5_1	5_2	6_1	6_2	7_1	7_2	8	9
Right	12	7	10	9	8	14	18	19
Wrong	2	2	1	3	7	4	10	3
NotCut	1	0	0	0	1	0	3	4
Precision(%)	85.7	77.8	90.9	75.0	53.3	77.8	64.3	86.4
Recall(%)	92.3	100	100	100	88.9	100	85.7	82.6

平均准确率为 79.1%，平均召回率为 92.6%，可见准确率比较令人满意，召回率相对更好一些。原因是算法在判定删减分割点时的阈值设置得比较小，使得差异本身不大的两段也不会轻易被合并成一段，这也达到了影视分割的目的：

尽量在保持准确率的基础上提高召回率，这样使用者就可以在跳过剧情的过程中尽量少地漏掉剧情。而准确率维持在一个较高的值也保证了不会设置过多的分割点让使用者感到烦躁，且每两个分割点之间尽量保证是一段连续的不短的情节，以保证使用者在跳过一段后能感受到明显的情节递进。

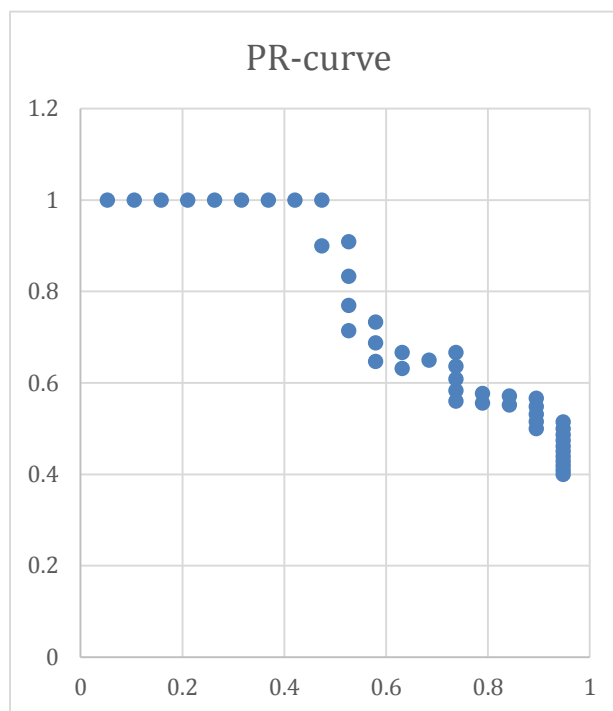
#### • 其他电视剧结果：

虽然我们的算法几乎是针对像《咱们结婚吧》这类肥皂剧而设置的参数，但我们也尝试分割了不同风格的电视剧，例如前段时间热播的《笑傲江湖》中第一集，部分结果如下(详见附件 outcome.xlsx)：

Right	19
Wrong	27
NotCut	1
Precision(%)	41.3
Recall(%)	95.0

可以看出，虽然准确率有些不尽人意，但也不算太低，而召回率依然是很高的，也就是说依然符合我们影视分割的最终目的，只不过需要观众跳更多的次数。但平均每分钟一个分割点的数目也是在可以接受的范围内。

右图为 PR 曲线，通过图像可以看出：召回率在基本不变的情况下也还是能够提升准确率的，说明算法的适应性还是很强的。



## GUI 成果展示

GUI 使用 JAVAScrip 和 HTML 编写。

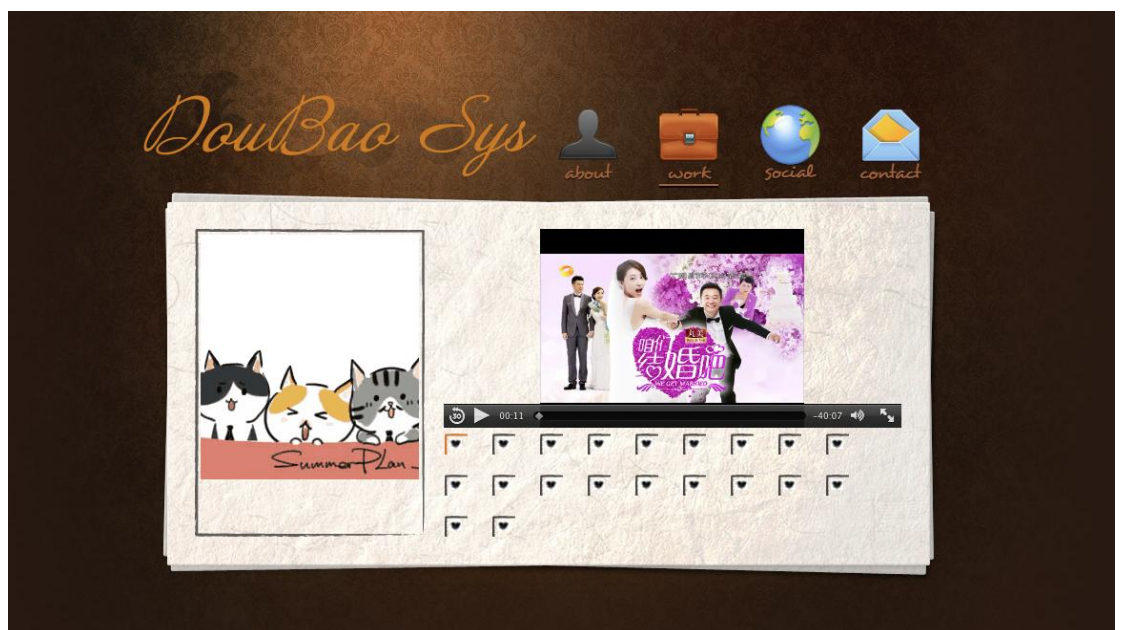
使用方式为点击 OUT.HTML，使用浏览器打开。

### • 人性化用户界面

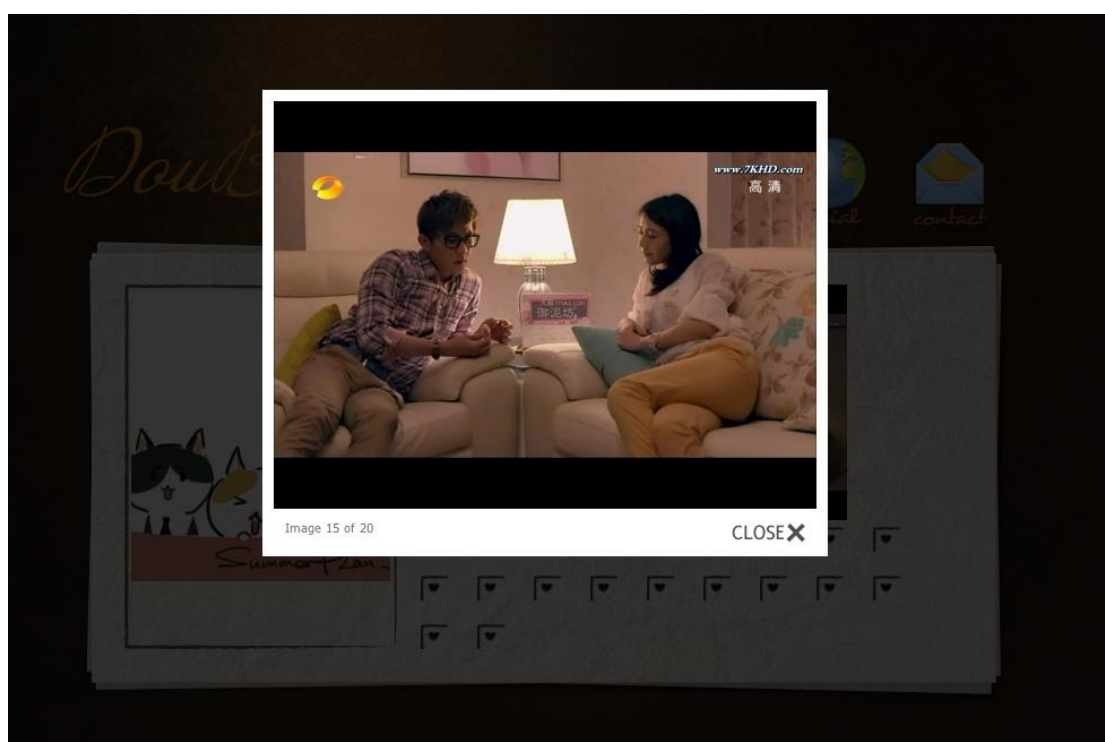
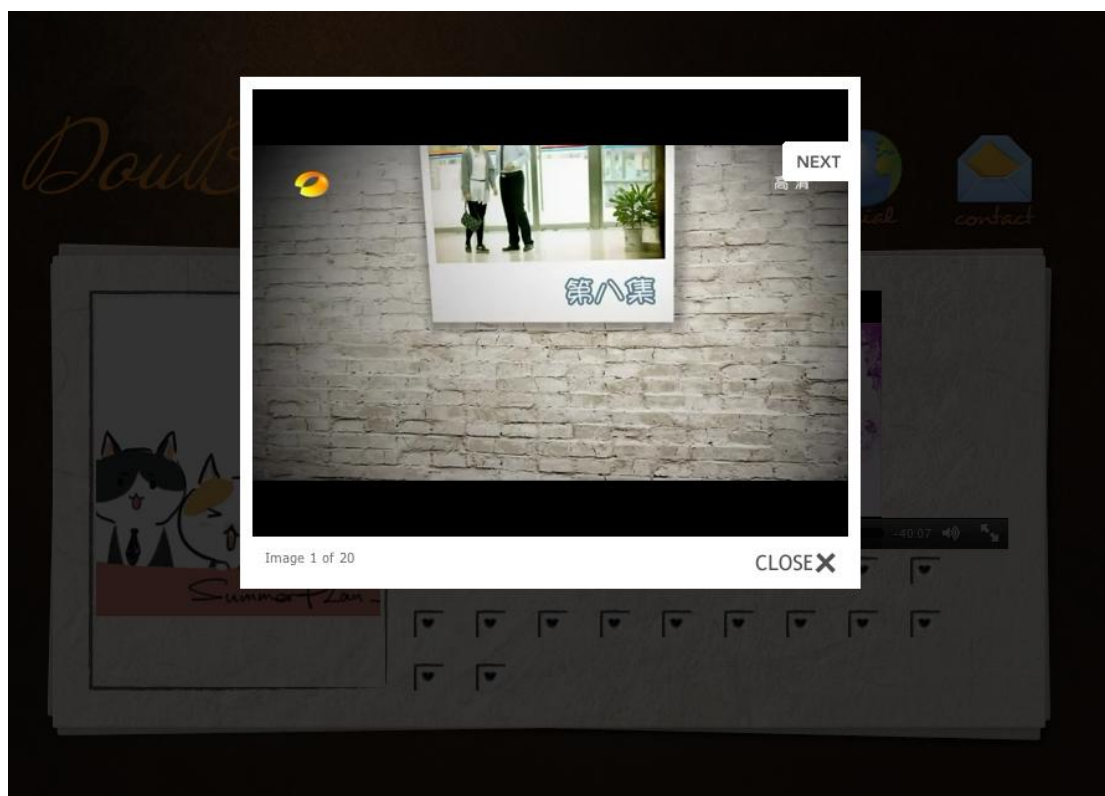
GUI 设计提供了一种与用户交互的思路。对于每一个分割点，用户可以通过单击，查看该段情景的截图，从而判断是否要观看这段视频。如果想要观看这段，直接点击这段分割即可。



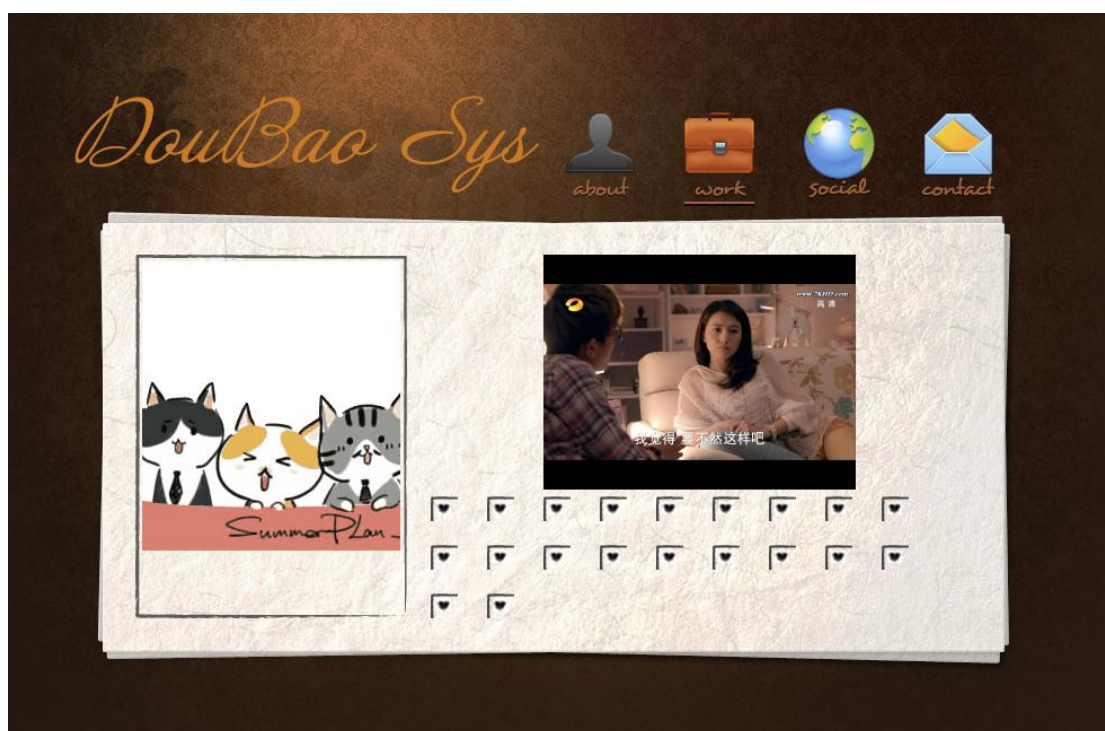
- 视频播放界面  
右下方的点击区域即为断点，点击之，可以查看该处的截图，并使播放器跳转至该处。



点击，查看图片，在图片界面可以翻页，来选择想看的截图部分。







## 研究过程中遇到的问题

### 初步确定算法

- 环境

对声音进行预处理，我们需要提取 MFCC，使用 GMM 模型、K-means 模型等作为主要方法，对音频模型进行分类。相对于声纹识别等研究，影视剧的音频处理显得更为复杂。主要原因有以下几点：①非纯人声，提取不准确，②背景音乐与噪声多，需要加以区分，③样本容量有限，很难进行有效的训练。

- 问题描述

音频数据难以进行处理。

- 解决

由于无法做到精确提取各部分的声音，进行相应的训练。由于影视剧音频的特殊性，使得我们无法进行语料库相关的训练。我们只能在无法训练的前提下，找到最优的方法，对音频进行切割和筛选。

为了达到相同的效果，我们采取了分步的策略。首先，按秒进行粗略分割，再对每秒提取 mfcc。提取后，我们利用每秒提取出的数据训练 gmm 模型，最后进行聚类。这样的效果比较好，结果在上文已详细给出。

## 查文献

- 环境

很难查到关于影视分割的文献。做影视分割，尤其是影视剧的影视分割的成果几乎为零，这给我们的造成了一些困难，我们没有现成的算法思路可以循照。

即使查到 **video segmentation** 也都是关于物体分割或者语音识别的，参考价值并不明显。

- 问题

难以查到有价值的文献。

- 解决

多试关键词，将 **audio** 和 **video segmentation** 结合起来。

在摸索中学习，询问徐老师，不断实验和测试，找出一条思路。

## 算法更新 1

- 问题

音频聚类不准，聚好的类别不易做标签，说话中和开头结尾不能聚成一类

- 原因

1s 时间过短，不易建立 **gmm** 模型；音频不易划分成段

- 解决

先基于图像切割，再按分割好的段训练 **gmm** 模型

## 算法更新 2

- 问题

图像切割的片段过小，不能用于训练 **GMM**

- 解决

按图像先进行合并；不考虑短的时间段，在按图像切分时控制相邻两个切分点间距不能过小（1s 以上），以保证能够训练 **GMM**。

## 引入说话时间段判定

- 环境

根据每 0.5s 的 **mfcc** 值算出其 **mfcc** 波动和频谱波动以及过零率，加权得到说话值，以此得到可能的说话时间段，在计算分割点两侧距离时也将两侧时段中的所有说话时段的 **mfcc** 合在一起训练 **gmm** 并算出距离。

- 问题

大部分时间段中没有说话时段，且对说话时段的判定不够准，有噪音，有漏判。

- **解决**  
没说话的直接按照背景音算，无视噪音和漏判。

## 算法更新 3

- **问题**  
很多时间段过短，即使通过背景音训练的 gmm 模型，算出的距离也很不靠谱。
- **解决**  
先通过分割点两侧定长音频对间隔较短的分割点计算距离，并删减，以保证最后留下的分割点间隔足够大（10s）。

## 算法更新 4

- **问题**  
引入说话对结果无正面作用。
- **解决**  
删掉这个因素。

## 实验总结

本系统实现了一个具有影视分割功能的人性化的 GUI，研究出一套可以很好地分割影视剧尤其是肥皂剧的算法，可以在保证准确率的情况下达到高召回率，能够满足视频网站的用户需求，达到了影视分割的实际应用目的。

## 鸣谢

感谢徐明星老师为我们提供的重要指导！  
感谢 308 不熄灯的桌子和灯！  
感谢一起刷夜的小伙伴们！