華中科技大學

创新训练项目结项报告

多模态视频信息分类系统

一级学科	新闻传播学类
负 责 人	吴宜檬
专　　业	传播学
院　　系	新闻与信息传播学院
指导教师	王然
导师单位	新闻与信息传播学院
实施时间	2021/03/20 - 2022/04/01
填表时间	2021/04/05

目　　录

1 项目概述

1.1 项目背景

随着技术水平的提升，视频这一媒介已成为人们生活中必不可缺的一部分。与此同时，许多人也纷纷加入视频编辑的浪潮，开始了由视频观赏者到视频创作者的转变。

在视频编辑过程中，对自主拍摄素材的筛选、切片往往是不可避免的第一步。在实践中，经常需要花费大量的工作时间与精力来实现对自主拍摄的素材的基础剪辑与分类。尤其在素材库较大的情况下，显得尤为繁琐。这样低技术水平，却高人工成本的需求，给大大小小的团队，尤其是非专业的个人视频创作者，带来了极大的不便。

基于这样的矛盾，本项目尝试提出一种多模态视频信息处理方法，为用户提供一种根据其需求文本，对原始素材进行自动剪辑切片的可能，进而提升视频编辑效率。

1.2 项目研究内容

本项目的研究内容可大致分为三个部分，包括需求与竞品分析、方法探索与产品实现。小组成员首先使用深度访谈的形式明确使用者的潜在需要，并对已出现的类似工具以及学界的近似处理思路进行学习与区别。而后，小组成员着重关注多模态视频信息处理方法相关内容，有针对性地对相关算法重点学习与研究。最终实现一个视频素材分类剪辑系统，该系统主要能实现两个功能：

1. 处理用户的检索需求

   用户在完成视频素材分类需求时，会有自身的分类标准，而从用户到机器，是本系统需要处理的第一部分。项目中，通过设置一定的输入规范，并通过自然语言处理适当优化与修正输入文本，从而明确之后识别所需要的关键内容。

2. 实现分类剪辑

   得到处理好的用户需求与用户素材后，即需要利用前期研究学习的算法，实现对视频素材的检索分类。对于不同模态信息进行整合理解，再使用视觉库实现场景识别与目标检测，最终完成关键视频提取剪切。

另外，在系统的实现中，小组成员还需要对平台进行功能规划与界面设计，制作一个交互友好、内容明晰的使用界面，并在初步实现后进行测试与修正，实现产品的迭代增强。

通过上述实践过程，项目预期实现一个能让用户实现本地素材的标签化切片工具。该工具基于现存的文科领域问题，采用工科的技术手段与创新方法，实现类似“自动剪辑提取”的效果，完成从人工检索到机器识别的改变，最终实际提升用户剪辑效率与剪辑体验，以智能化的方式适应当前视频剪辑需求大幅增长的时代趋势。

1.3 项目研究综述

1.3.1 智能剪辑技术现状

从视频检索以及自动剪辑技术的产品使用状况来看，以产品展示视频在电子商务领域的应用为例，徐德顺的《国际电商发展趋势与中国电商发展对策》中提到，据 eMarketer 预测，2019-2021 年间，全球移动电商销售额由 2.3 万亿美元攀升至 3.6 万亿美元。电商行业和短视频行业的快速发展促进了产品信息展示方式的进步，让视频这种信息动态展现的方式成为除文字、图片之外最重要的内容营销形式。

但是，视频的搜索以及剪辑效率，从市场的角度来看，依然有很大的问题。根据帅世辉的《产品展示视频自动剪辑方法研究》，对于视频的自动剪辑方法，目前的市场上依然缺乏系统化的设计框架。目前，视频分割都是基于完整镜头或对话等高级语义，不符合产品展示视频短小紧凑，以展示产品信息为主的特点；镜头组接则大多是按照时间线来组织，用来表达故事或纪实，无法对没有明显的时间关系的视频素材进行编辑；最佳镜头序列的求解算法大多是通过限制镜头前后关系进行的，缺少全局性约束。与此同时，电子商务领域的商品特别是服装类商品迭代速度很快，通常一个季度就需要全部更新一遍。一个电商平台的商家通常拥有数十到几千件商品。商家面对如此快的迭代速度和如此庞大的商品数量，传统的视频制作方式难以满足，商家需要找到视频更有效率的制作方案。视频制作自动化相关技术的发展，为解决这一问题带来了可能。

1.3.2 多模态信息处理技术现状

从技术层面看，现有的研究与实践使本项目的展开成为了可能。本项目预计通过文本理解、人脸识别、场景识别等技术，以实现对用户需求和视频语义的同步理解与对其，进而达成视频自动检索以及自动剪辑的目的。

1. 文本信息抽取与理解

综合国内外信息抽取研究成果，常见的三类文本信息抽取方法分别为：基于自然语言处理的规则模板方法、基于传统统计方法和基于统计机器学习的方法。在本项目中，将主要使用基于自然语言处理的信息抽取方法。

对于基于自然语言处理的信息抽取，根据蔡皎洁的《AI 中的文本信息抽取方法进展研究》，该方法通过上下文词性分析、句法分析以及依存关系分析来抽取和总结频繁发生的规则模式实现信息抽取。该方法经历了基于“名词”、“复合性名词术语”、“文本结构加权术语”等判断重要概念的过程。在基于名词术语规则的抽取上，Hears 和 Marti 人工定义了上下文名词术语间的语义模板，抽取名词间的 is a(kind of) 关系。在基于复杂术语规则的抽取上，Justeson 和 Katz 提出术语也可以表示成“名词短语”，其中包括形容词、名词、偶尔有介词，很少包括动词、副词和连词，他们提出了由这些词语特征驱动的术语提取方法。

2. 人脸信息抽取与理解

视频图像有两个有效的独特性: 同个主体的多帧融合与实时信息。多帧融合保证了姿势的变化, 允许适当的选取一些高质量的帧 (如: 高质量的近距离正面姿势的人脸图像)。隐藏在视频中动态的面部运动被认为是实时信息。因此，以视频图像作为输入的具有高鲁棒性的人脸识别系统的发展成为近年来的热点及难点。

根据郭建华的《基于视频的人脸识别综述》，基于视频的人脸识别系统通常包括三个模块: 人脸检测模块、跟踪模块和识别模块。而在人脸识别模块，主要有基于时空信息、基于统计模型、多模生物特征认证等方法。

对于基于时空信息的方法，Zhou 和 Chellappa 提出了一种合成视频序列中实时信息的人脸识别方法, 利用含有跟踪状态向量和识别变量的状态空间模型用来描述主体, 再利用序贯重要采样 (SIS) 的方法有效估计出跟踪状态向量和识别变量的后验概率分布。同时，Krueger 和 Zhou 采用线性径向基函数方法, 从训练视频中选择有代表性的人脸图像作为样本模型, 这个模型可以有效捕捉小范围的

2D 运动, 但是不能处理大的 3D 姿态变化或者遮挡问题。

对于基于统计模型的方法，动态模型利用了人脸的时间和空间连续变化的信息, 能够更好地刻画人脸的动态变化特性。由于视频引起的局限可以由该人脸识别框架解决。

对于多模生物特征认证方法，可通过融合人脸和步态特征进行识别, 得到良好的识别率，从而实现自动视频特征识别；同时结合直方图归一化、boosting 技术和线性鉴别分析来解决光照、姿态和遮挡等问题, 并且可以用扩展的 kalman 滤波 (EKF) 方法，优化语音去噪算法。

3. 场景信息抽取与理解

根据《多模态视频信息检索》中应用于视频检索的方法论述，考虑到提取图像的低层特征仅仅能够“简化”计算机对图像的表示方式，并不能够真正得到该图像所代表的语义信息；且随着视频数据库的不断增大，采用人工标注语义的方法显然是不明智且是不切实际的，所以自动化提取图像语义的算法研究广泛展开起来。主要思路在于：首先需要比较全面地提取出图像的种种特征，只有特征比较全面的提取出来之后，才能较好地表征这幅图像，进而可以与图像的语义相联系；当特征提取出来之后，需要采用统计学习的方法建模，将特征和语义通过统计学习的方式对应起来。通常的统计学习建模方法有：贝叶斯分类，决策树，支持向量机（SVM），高斯模型（GMM）等。目前的研究主要着眼于建模方法的确定。

综上所述，已有的研究为本项目提供了良好的研究基础。对基于内容的视频检索, 国内外已开发出的原型系统：QBIC 系统、CORE 系统、VisualSEEK 系统、TV-FI(Tsinghua Video Find It) 系统和 iVideo 视频检索系统。对自动剪辑过程，目前的研究成果有：《产品展示视频自动剪辑方法研究》中，通过信息分类建模、基于图片相似度的方法进行镜头分割、滑动窗口分割算法进行子镜头分割等，来实现视频自动剪辑；《基于深度学习的视频自动剪辑》中，可以通过卷积神经网络、人脸识别经典模型-Face Net、生成对抗网络 GAN 等相关算法与理论来实现基于深度学习的视频自动剪辑。

在良好的研究基础上，从目前国内外有关多模态视频检索与自动剪辑技术的发展情况来看，多模态视频检索与自动剪辑依然在行业上、技术上存在一定的问题。在行业上，与视频检索以及自动剪辑相关的技术成为行业内探索的热点，

但大多数相关工作仅仅停留在研究的初级阶段，还需要进行进一步核心技术的研究。在技术上，国内外已经初步具备了视频检索、自动剪辑、多模态信息融合等技术，但是这些技术并未能很好地融合运用在视频检索与自动剪辑领域；另一方面，虽然多模态技术已经较为完善，但多模态技术与抽取并理解自采集视频信息技术方面还有一定的技术鸿沟，即多模态技术并没有很好地应用在采集视频信息技术上。

因此，从已有的研究为本项目奠定的良好根基上、从行业的需求上、从各项分立技术的成熟与完善上，本项目的研究都具有独特的价值与意义。

1.4 项目创新与特色

1. 内容创新：智能处理

过去的“视频粗剪”，往往需要耗费大量不必要的人工成本用于检索拍摄的海量素材，且存在以偏概全的可能。区别于传统人工检索海量素材，我们充分发挥大数据和机器的优势，通过视频内容精准理解，将重复、繁琐、不精准的生产过程，转为高效率，低人工成本，且标准化的生产体验。将时间还给灵感，让创作回归创作。

另外，项目的智能也体现在对用户意图的理解上。区别于搜索引擎上的固有的通过标签化语言进行检索的思路，项目中使用自然语言处理方法，给予用户自由表述的空间，并对其进行语义上的理解。最终通过对跨模态信息的融合对齐，实现用户所述即所搜，所搜即所得的使用效果。

2. 思路创新：文工交叉

本项目基于新闻学院学生的体验，在人文与新媒体的视域里发现问题。并在项目前期采用了新媒体方向产品设计的思路，即：通过访谈等方式确定需求，通过研究过往类似技术等发现不足与可借鉴的地方，从而搭建初步的功能结构。

但本项目并未止步于此，而是将构想转为实践与应用，通过对智能技术的学习与整合，实现已有技术的跨领域应用，让技术有作用，让问题有着落。

2　项目设计与具体实施

2.1　应用结构设计

整体应用按照核心功能，即用户体验流程出发，推及前后端的函数与算法，最终得到的流程规划与设计如下图：

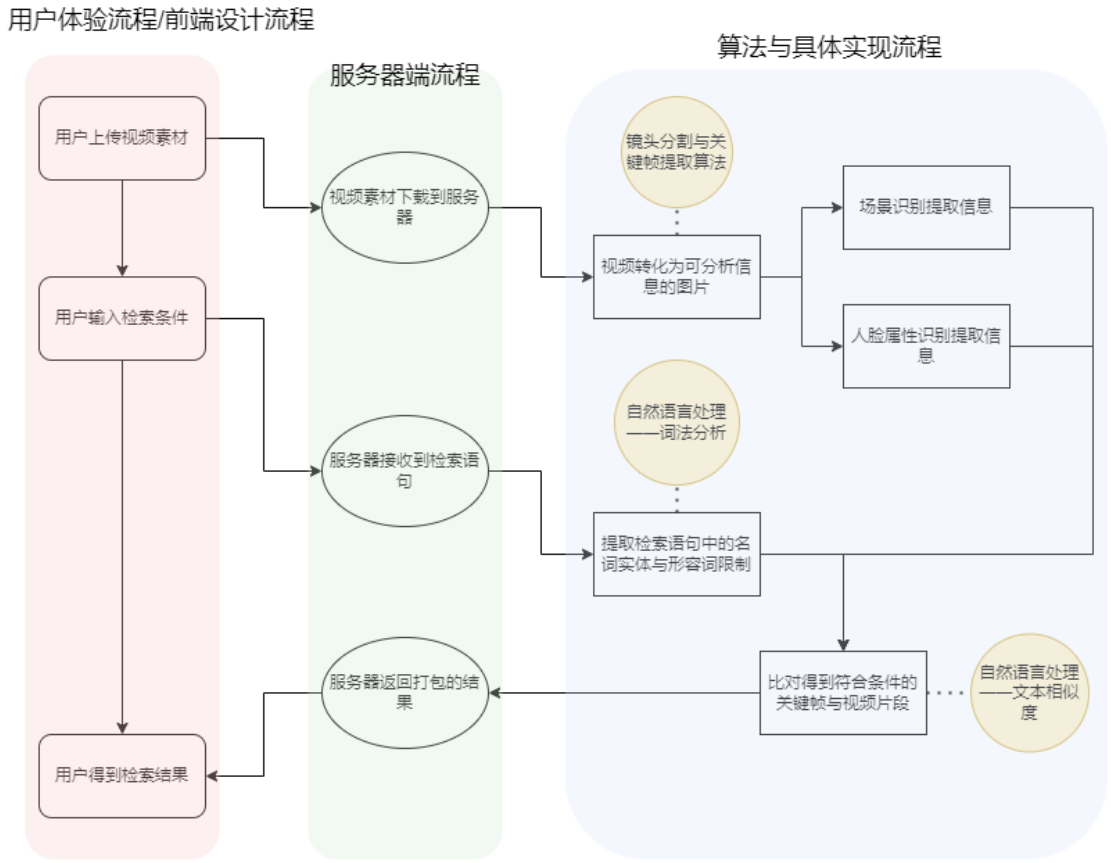

图 2-1　结构设计

2.2 核心代码设计

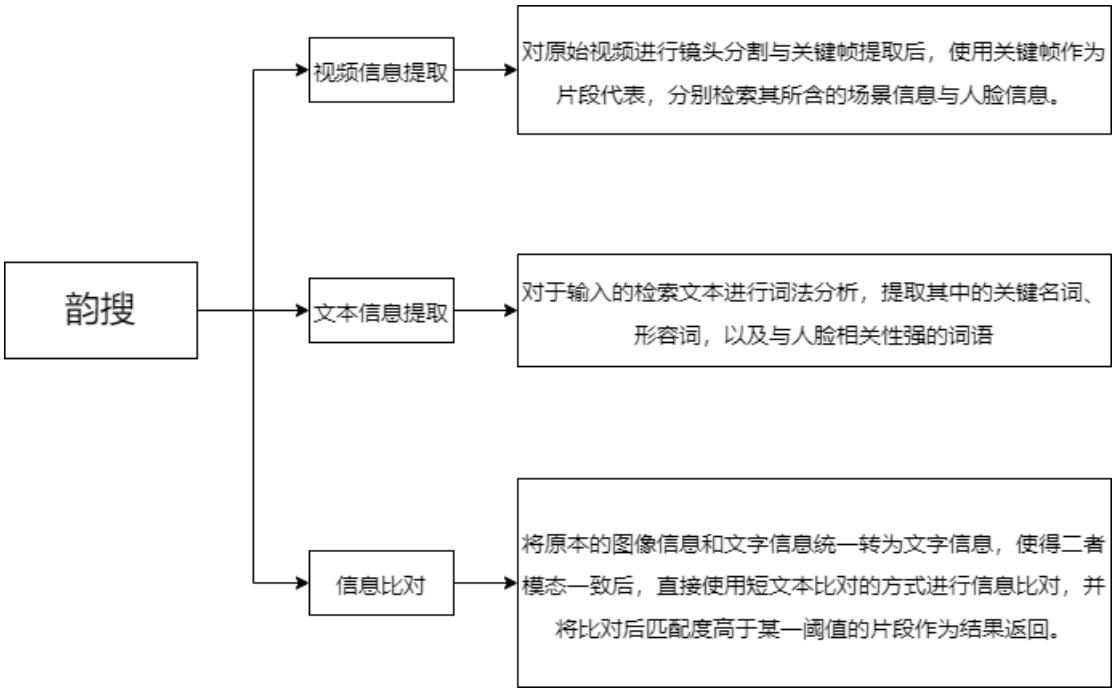

图 2-2 核心代码设计

2.2.1 视频信息提取

视频信息提取包括视频镜头分割和关键帧画面信息提取两部分，主要在 *extraction_work* 函数中完成执行视频镜头分割。因为视频在运动中可能产生大量的中间过渡帧，在视觉上可能有’模糊’等特性、同时这些过渡振能提供的信息与前后帧差别不大。因此对其进行信息检索时可能出现效果差且效率低的特性。而关键帧，即镜头分割时的分割帧，不仅往往因为是运动的始末而呈现出稳定的形态，也能在很大程度上代表当前镜头的画面信息。

具体操作上，使用 *ffmepg* 库能快速地在处理视频文件时，通过对帧做标记实现关键帧提取的功能。

```
def extraction_work(name,n_word,fc_att): #视频镜头分割与关键帧提取
        '''
        这里使用是基于ffmepeg的关键帧提取 对每一帧进行标记 得到I、P、B帧
        I帧具有以下特点:
        - 解码时仅用I帧的数据就可重构完整图像
        - I帧描述了图像背景和运动主体的详情
        - I帧不需要参考其他画面而生成
        - I帧所占数据的信息量比较大
        因此我们可以视I帧为我们所需的关键帧，并将其标记在res[]中
        '''
```

```
        #--------------------关键帧提取----------------
        fname=FILE_PATH+'videos/'+name
        res_path=FILE_PATH+'extract_result/'
        cmd='ffprobe -v error -show_entries frame=pict_type -of default=
            noprint_wrappers=1'.split()
        res=subprocess.check_output(cmd+[fname],shell=True).decode()
        res=res.replace('pict_type=','').split()
        idx=0
        cap=cv2.VideoCapture(fname)
        success,frame=cap.read()
        #--------------------关键帧信息查询----------------
        while(success):
                if (len(match)<=idx): match.append(0)
                else: match[idx]=0
                if res[idx]=='I': #只对I帧查询
                        req_imgs(frame2base64(frame),idx,n_word,fc_att,0.35)
                        #阈值设为0.35
                idx+=1
                success,frame=cap.read()
        cap.release()
        return res #返回关键帧标记
```

关键帧画面信息提取则主要通过两次查询完成，一次是针对“环境”、“静物”等的场景检测，另一次则主要是针对人脸信息。

两次查询分别调用百度开放平台的通用物体和场景识别接口和 $Face++$ 的人脸检测接口，得到如下图的结果。

{'result_num': 5, 'result': [{'keyword': '海鸥', 'score': 0.804717, 'root': '动物-鸟类'}, {'keyword': '红嘴鸥', 'score': 0.627296, 'root': '动物-鸟类'}, {'keyword': '热带海鸟', 'score': 0.459738, 'root': '动物-鸟类'}, {'keyword': '天鹅', 'score': 0.282051, 'root': '动物-鸟类'}, {'keyword': '江鸥', 'score': 0.101176, 'root': '动物-鸟类'}], 'log_id': 1430045193844742930}

图 2-3　调用结果

2.2.2　文本信息处理

输入文本处理主要涉及 $words\_edit$ 函数。对于输入的文本首先使用 $req\_extraction$ 函数通过调用词法分析接口进行分词与名词和形容词的提取。一般情况下，对于场景静物的描述集中于名词，而针对人脸的描述则是通过形容词实现。

而提取文本中与人脸相关的信息则主要通过 get_fc_att 函数实现。对人脸相关的信息进行提取主要采用将文本信息与人工设定的人脸属性代表词典进行相似度分析，通过不同的阈值以及在各属性中的不同相似情况，判定语句中是否包含人脸相关信息。

```
def words_edit(words): # 输入文本处理
        '''
```

```
        对于输入文本需要进行两方面的操作:分词+部分信息提取
        req_extraction: 对输入文本进行基于词法的分词和提取
        得到名词列表 n_word 与 形容词列表a_word
        get_fc_att : 基于提取出的名词和形容词
        进行人脸相关的属性 ( 是否对人有需求 / 人的性别、年龄、情绪、颜值 )
        最终得到名词列表和人脸属性用于后续比对
        '''
        n_word,a_word = req_extraction(words)
        fc_att=get_fc_att(n_word,a_word)
        return n_word,fc_att

def req_extraction(sentence): #返回用户输入中的所有名词与形容词 request
        params = {"text": sentence}
        request_url = Extraction_url+"?access_token="+nlp_access_token
        headers = {'content-type':'application/x-www-form-urlencoded'}
        response = requests.post(request_url, data=params, headers=headers)
        n_words = []
        a_words = []
        if(response):
                result = response.json()
                if "items" in result.keys():
                        data_list = result["items"]
                        for res_dic in data_list:
                                if 'n' in res_dic["postag"]:
                                        n_words.append(res_dic["word"])
                                if 'a' in res_dic["postag"]:
                                        a_words.append(res_dic["word"])
        return n_words,a_words

def get_fc_att(n_word,a_word): #得到检索语句有有关"人"的信息
        '''
        文本可提取的信息与自定义词典进行比对
        包括 "是否对人有需求 / 人的性别、年龄、情绪、颜值"五个方面
        '''
        fc_att={}
        '''
        是否对人有需求: 即可能并没有限定人脸相关的属性 但提到了"人"
        如: 教室里的人等
        比对词为"人" 相似阈值为0.5
        '''
        exist_dict={'1':"人"}
        res=n_fc_att(n_word,exist_dict,0.5)
        if (res): fc_att['exist']=res
        '''
        性别: 即语句中包含的显性/隐形性别检索条件
        如: 姑娘、美女、帅哥等
        比对词为"女性"、"男性" 相似阈值为0.5
        '''
        gender_dict={"Female":"女性","Male":"男性"}
```

```
        res=n_fc_att(n_word,gender_dict,0.5)
        if (res): fc_att['gender']=res
        '''
        情绪：即语句中包含的情绪限制 一般为形容词
        如：高兴的小孩、伤心的老人等
        比对词分别为各个情绪的翻译词
        实测发现 形容词之间的相似度相对较高 相似阈值设为0.6
        '''
        emo_dict={"anger":"愤怒",
        'disgust':'厌恶',
        'fear':'恐惧',
        'happiness':'高兴',
        'neutral':'平静',
        'sadness':'伤心',
        'surprise':'惊讶'}
        res=n_fc_att(a_word,emo_dict,0.6)
        if (res): fc_att['emo']=res
        '''
        年龄：即语句中包含的显性/隐形年龄检索条件
        如：学生、老爷爷等
        比对词分别为各个年龄阶段的统称
        考虑到人脸年龄检测的结果为具体的数值如24，置信度相对较低
        因而设置各个年龄阶段存在部分重叠
        相似阈值设为0.5
        '''
        age_dict={"0-10":"幼儿",
        '5-15':'少儿',
        '10-30':'青少年',
        '20-40':'青年',
        '30-60':'中年',
        '50-200':'老年'}
        res=n_fc_att(n_word,age_dict,0.5)
        if (res): fc_att['age']=res
        '''
        颜值：即语句中包含的显性/隐形颜值检索条件
        如：美女、帅哥
        比对词为"漂亮"
        实测发现这一比对词与其他描述人的词汇相似度普遍偏高
        且这一需求的检索情况往往会直接表明对颜值的需求
        相似阈值设为0.8
        '''
        score_dict={'85':"漂亮"}
        res=n_fc_att(n_word,score_dict,0.8)
        if (res): fc_att['score']=res
        return fc_att
```

2.2.3 双方信息比对

得到图像信息和文本信息后，将视觉模态的信息和文字模态信息转化至同一维度，最终可将其转化为语义上的向量表示，使用文本之间相似度完成计算。

经测试发现对于文本中的名词实体与场景检测的相似阈值设为 0.35 能有效地排除与需求不符的镜头，同时尽可能保留与需求有关的镜头。另外，由于人脸属性是基于预设的词典进行比对，其无法直接统一使用语义上的类似，因此对于人脸属性进行的有针对性的比对在 *sim_attr* 函数中实现。

```
def req_similarity(word1, word2): #两次相似度获取 request
        '''
        计算传入两词的相似度
        :param word1:
        :param word2:
        :return:
        '''
        time.sleep(0.6)
        params = json.dumps({"text_1":word1, "text_2":word2}).encode('GBK')
        request_url = SL_url+"?access_token="+nlp_access_token
        headers = {'content-type':'application/json'}
        response = requests.post(request_url, data=params, headers=headers)
        if(response):
                result = response.json()
                print(result)
                if "score" in result.keys():
                        return result["score"]
        return 0

def isSame(word_list, word): #词与词合集的文本相似度计算
        '''
        计算word与word_list中所有词的最大相似度
        返回相似度值，根据当前情况阈值再判断
        :param word_list: 在之前已经获取过的名词列表
        :param word: 传入的词
        :return: similarity值
        '''
        similarity = 0
        for word_a in word_list:
                similarity = max(similarity, req_similarity(word_a, word))
        return similarity

def sim_attr(gender,age,emotion,score,fc_att): #判断是否与人脸属性相似
        '''
        性别属性：直接判断是否一致
        颜值属性：视频中人脸是否符合高颜值标准(分值>85)
        年龄属性：判断年龄值是否在需求的年龄区间内
```

```
        情绪属性：判断视频人脸出现的最高分情绪与需求是否一致
        如上述条件出现不一致则返回 0
        如全部保持一致则返回 1
        如没有上述属性限制 则表示只是单纯对 "人出现了" 这件事存在需求
        那么直接返回 1
        '''
        if ('gender' in fc_att) and (gender!=fc_att['gender']): return 0
        if ('score' in fc_att) and (score<fc_att['score']): return 0
        if ('age' in fc_att):
                edge=fc_att['age'].split('-')
                if age>int(edge[1]) or age<int(edge[0]): return 0
        if ('emo' in fc_att):
                mx=mx_name=0
                for name,value in emotion:
                        if value>mx:
                                mx=value
                                mx_name=name
                if mx_name!=fc_att['emo']: return 0
        return 1
```

2.3 使用流程展示

1. 进入软件对应网址，如图 2-4 所示。

图 2-4 韵搜首页

2. 点击上方导航栏中的 Editor 按钮，或页面右下角标注了向右箭头的圆形按钮，如图 2-5，图 2-6 所示。进入主要操作页面，如图 2-7 所示。

图 2-5　导航栏点击操作

图 2-6　圆形按钮点击操作

图 2-7　主要操作页面

3. 点击上传视频按钮，如图 2-8 所示，并在本地选择视频文件进行上传，如图 2-9 所示。

图 2-8　点击上传视频按钮

图 2-9　上传本地视频

4. 在输入框输入检索条件，如图 2-10 所示。

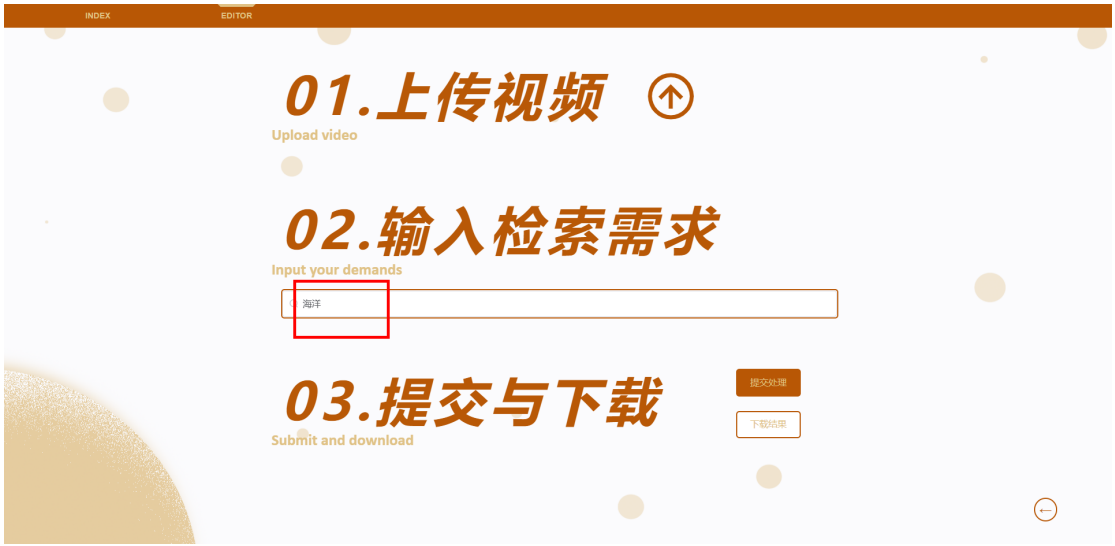

图 2-10　输入检索需求文本

5. 点击提交处理并等待，如图 2-11 所示。

图 2-11　点击提交处理

6. 后端处理完毕后，原本禁用的下载结果按钮被激活，进行点击即可下载结果，如图 2-12 所示。

图 2-12　点击下载结果后自动下载结果压缩包

7. 结果压缩包中包含符合条件的视频片段与一个包含这些视频片段所在的时间戳信息的文本文件，用户可解压该压缩包后自行进一步处理，如图 2-13 所示。

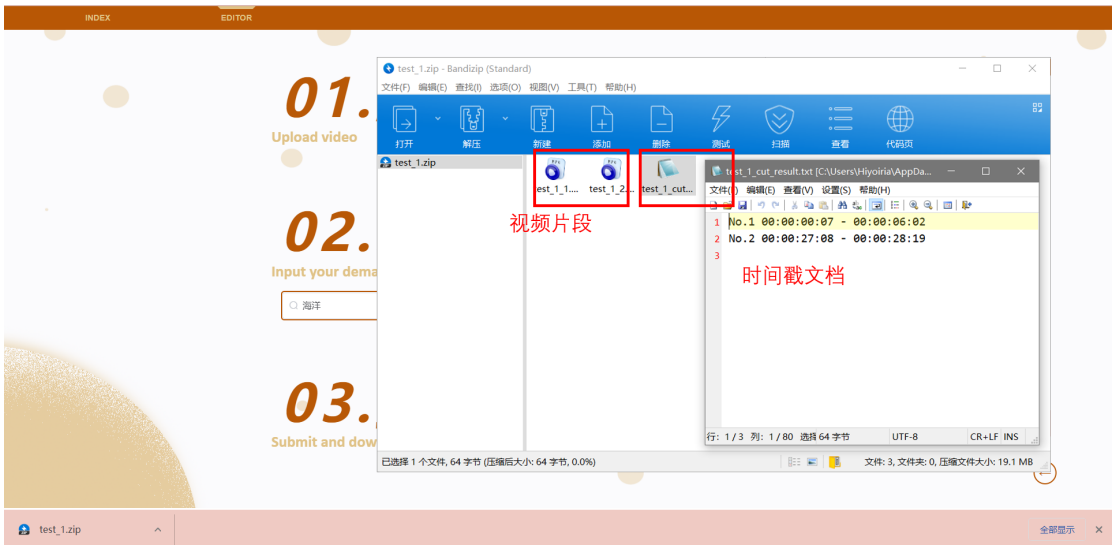

图 2-13　打开结果压缩包

3 测试结果、反思与展望

3.1 部分案例测试结果

	Case1	Case2	Case3
视频类型	博主拍摄的 vlog 片段节选	新闻学院 30 周年院庆 MV	小组成员摄像课作业
视频特点	画面干净，主体突出	有后期转场画面，画面元素多	画质差，抖动明显，人物脸部有覆盖物
测试内容	对优质素材的场景检索能力	对有多个属性（性别、年龄）的人物的检索能力	多个视频的处理能力，低质量生活化素材处理能力
原始视频时长	30 秒	56 秒	4 秒 6 秒 4 秒
应用处理时长	222 秒	263 秒	153 秒
搜索关键词	海洋	女学生	人物和单车
原片符合要求片段时长	9 秒	30 秒	4 秒 6 秒 4 秒
处理得到片段总时长	8 秒	28.5 秒	4 秒 3 秒 1 秒
处理得到符合要求片段时长	8 秒	20.5 秒	4 秒 3 秒 1 秒
处理得到不合要求片段时长	0 秒	8 秒	0 秒
查准率	100%	70%	100%
查全率	89%	68%	58%

3.2 反思与展望

经过程序编写与多种情况下的测试，目前的应用已能基本较好地实现少量视频与相对复杂的检索条件的匹配，但整体仍存在以下问题：

用户需求提取与语义匹配精度问题

目前的需求提取采取的是词法分析的方式，主要通过对词性的筛选来进行需求实体的识别。其中场景和绝大部分的人脸属性与名词进行匹配，而人脸中的情绪识别则主要与形容词进行匹配。基于词法分析的提取效果忽略了实体之间的关系，如“女孩和海洋”，“海洋边的女孩”所表达的检索需求其实存在较大的差异，前者需要出现女孩或海洋的镜头，而后者则是要求两个限制条件重叠。目前小组成员的改进思路是对用户短文本进行句法层面的分析，通过句法依存、关系词定义等手段明确各个搜索实体间的关系。

在进行语义模糊匹配的过程中，小组成员最初选择的是通过计算单独的词语间的相似度来判断，而事实上这样的相似度判断结果并不适用于我们所需要的场景。如“男性”和“女性”两个搜索条件截然不同，但由于词性一致，且表达的都是判定人的性别的值，在进行词与词单独相似度分析的时候得到的相似度较高，不符合我们所预期得到的效果。因此小组成员将词与词之间的相似度计算，改为按照“短文本”之间的相似度计算，这种方式虽然没有直接抛去词性等与语义不相关的其他条件，但通过这种方式，变相地降低了“词”本身相关的属性权重，使得语义的权重上升，测试发现如“男性”、“女性”的相关度降至 0.2 左右。

尽管做出了这样的改变，词语之间的匹配仍然会出现部分“异常”的情况，如“海洋”和“江河”的相似度与“车”和“樱花”的相似度是接近的，这可能取决于相似度计算的原本样本情况与计算方法，之后改进需要进一步增强对于语义的理解，并结合具体的可能出现的“匹配”词库进行进一步的精准优化。

整体运行速度较慢，不能适应大规模批量处理

测试结果中，运行时长情况整体不尽人意。尽管可能有测试设备的性能原因，但整体的处理时长仍有较大的改进空间。因为这样的处理速度和项目最初所构想的实现大批量、高效率搜索的目标是有所冲突的。

小组成员认为，一方面应提升项目主体处理部分的运行效率，包括但不限于

选择更加有效的镜头分割、关键帧识别方式；对用户提交的视频进行降采样等预处理；提升使用接口时的 qps；通过提前更加精准地提取用户输入的语义信息避免重复检索等。另一方面则应集中于改进前后端编写方式，例如提升网页的响应速度，减少不必要的读写操作等。

搜索词粒度较大，难以实现精确搜索

在测试的过程中小组成员曾提出经常会有的搜索需求是基于人物的衣着和动作进行的。对于前者，目前没有一个较好的对衣着的表述与理解方式，在搜索提取画面信息时也很难对于这一级别的信息进行分析。而人物动作实际上是由多帧的变化构成的，这对关键帧提取的频率提出了较高的要求。针对具体的问题小组成员提出了以下构想：

1. 增加以图片为模板的输入，进而实现对难以描述的信息、具体的人物等的追踪
2. 通过人体定位和人体关节定位等，在确定大的动作起始与终结帧后，对中间的帧进行各个定位点的变化情况分析，进而理解物体的运动趋势。

附录

1. 场景识别接口官方文档：https://ai.baidu.com/ai-doc/IMAGERECOGNITION/Xk3bcxe21
2. 人脸检测接口官方文档:https://console.faceplusplus.com.cn/documents/4888373
3. 词法分析接口官方文档：https://ai.baidu.com/ai-doc/NLP/fk6z52f2u
4. 文本相似度接口官方文档：https://ai.baidu.com/ai-doc/NLP/ek6z52frp
5. 测试视频、系统使用演示视频以及源代码：https://pan.baidu.com/s/1UFbWPTmdQVOmccuCLlZbyw 提取码：ry4p