基于 SV2TTS 的在线慕课视频自动生成系统设计与实现

郭辉[1] 刘丽景[1,2] 陈群心[1] 刘怡凡[1] 杨智钦[1] 薛婷之[1]

（1. 西安培华学院 陕西省西安市 710125；2. 西安交通大学 陕西省西安市 710049）

摘要：本文设计一种基于 SV2TTS（Speaker Verification to Multispeaker Text-To-Speech）技术的在线慕课视频自动生成系统，能够自动地将 PPT 课件和相应的讲解文本转化为慕课视频，通过语音克隆技术合成讲解者的语音并嵌入到视频中，最终生成适用于在线教学的慕课视频。系统基于 SV2TTS、POI、FFmpeg 等技术进行实现，本文对系统总体架构设计和系统处理流程与具体实现进行了详细的阐述。本系统能够快速的将线下课程转化为线上慕课，极大提高学校数字化课程建设进程。

关键词：SV2TTS；在线慕课；语音克隆；FFmpeg

Design and implementation of automatic MOOC video generation system based on SV2TTS

Absrtact：This paper designs an online MOOC video automatic generation system based on SV2TTS (Speaker Verification to Multispeaker Text-To-Speech) technology, which can automatically convert PPT materials and corresponding explanatory text into MOOC videos, the narrator's voice is also synthesized and embedded into the video through voice cloning technology, then finally generate MOOC videos which are suitable for online course. This system is implemented based on SV2TTS, POI, ffmpeg and other technologies, and in this paper, the overall system architecture design and system processing flow and specific implementation are elaborated. This system can quickly convert offline courses into online MOOC courses, which greatly improves the process of digital curriculum construction in schools.

Keywords：SV2TTS；online MOOC video；Voice Cloning; FFmpeg

1. 引言

近年来，随着教育行业的大力发展，微课已经成为教育发展的趋势[1]，慕课作为一种数字化资源的在线课程，逐渐融入学校课堂，为学生提供了更广阔的学习空间。慕课符合开放教育的理念，降低学习的门槛，在一定程度上符合“以学生为中心的”教学理念[2]。然而，在慕课的建设过程中，存在着一系列繁琐而重复的操作和较长的课程建设周期。这包括课程设计、视频制作、教材准备、在线平台的搭建和维护等工作，需要耗费大量的时间和精力。伴随着人工智能的快速发展，尤其在文本转语音以及语音克隆领域的发展，在线慕课的自动化构建成为可能。本文所设计的系统在一定程度上节省了慕课制作者的时间和精力，帮助教师打造高质量的慕课视频。

2. 系统开发设计

慕课视频自动生成系统主要由客户端服务、API 网关、业务逻辑系统、领域服务以及底层基础设施五部分组成。系统整体架构如图 1 所示，其中第一层为客户端，分为 Web 端和移动端，用户可以通过这两种方式提交 PPT 课件、讲解文本和教师个人音频数据，APIG 网关对收到的数据和请求做认证校验，并将不同的请求转发到相应的业务逻辑中；业务逻辑层，主要包含四种核心功能：PPT 课件管理、讲解管理、Speaker 管理以及自定义音频录制。这些功能由业务逻辑层处理，根据不同的功能要求，调用领域服务来实现具体功能。领域服务以微服务的方式提供文件管理、格式转换、语音生成、语音 AI 等功能，为系统的功能提供了强大的支持。最底层是系统的基础支撑层，主要提供存储、计算等基础能力，确保系统的稳定性和可靠性。

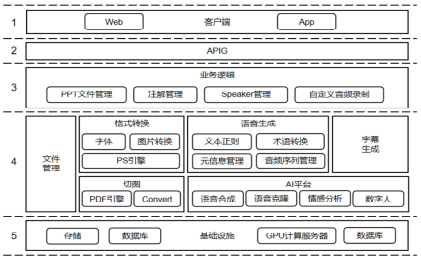

图 1 系统架构

3 系统流程设计

基于 SV2TTS 的慕课视频生成系统主要流程包括 PPT 转换图片，文本合成语音，字幕生成以及视频合成。系统处理流程如图 2 所示。

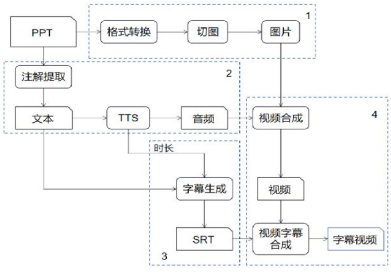

图 2 视频生成流程

3.1PPT 格式转换

为保证 ppt 格式不失真，避免因转换过程中出现格式错误，本系统将 ppt 转换为 PDF 文档格式，对其进行按页码裁剪，生成所对应需要的图片类型，以供系统使用。

PPT 转换包括一下步骤：

PPT 导入：将 PPT 文件上传到系统中；

格式转换：系统后台利用 POI 接口将上传的 PPT 文件转换为 PDF 文档。

页面提取：将 PDF 每个页面提取为一张图片，并按照页面顺序生成图片序列；

图片处理：对每张图片进行处理，包括裁剪、压缩等操作，以提高图像质量。

3.2 文本转语音

文本转语音是将文本内容转换为自然流畅的语音输出的过程。文本转语音包括一下步骤：

语音模型训练：利用 WaveNet 算法[3][4]根据慕课数据训练基础语音 Encoder 模型，同时根据个人录制的音频数据生成个人语音声纹(Mel 特征)。

文本预处理：去除特殊字符和标点符号、词干提取或词形还原、移除数字和文本规范化等。

文本到语音的转换：将预处理后的文本输入到训练好的语音模型中，生成对应的语音信号。

语音信号处理：对生成的语音信号进行处理，包括噪声去除、音调调整、音量平衡等操作。

3.3 字幕生成

视频字幕能够帮助学生提高学习效率，字幕生成的主要工作是根据讲解文本和讲解音频，按照时间轨道生成固定标准格式的文件。本系统所支持的字幕文件有 SRT、SSA、SMI、ASS 四种格式。

3.4 视频合成

视频合成是将音频、视频帧合并为视频文件的过程，主要包含两个步骤：一是将 PDF 切好的图片序列作为视频帧，与音频合成为可连续播放的视频文件；二是将生成好的视频通过导入字幕文件生成带有字幕的视频文件。

视频帧合并音频功能实现使用 FFmpeg 工具[5][6]来生成，具体步骤如下：

1.先对 PPT 图片的音频文件按照时间序列生成排序文件；

2.利用 FFmpeg 对每一页 PPT 图片和对应的语音文件生成独立的指定分辨率的视频文件；

重复步骤 1 和 2，将每一页 PPT 都生成一个视频文件，并按页数序号排序；

将排序的视频合并为一个视频文件。

视频字幕合成是将字幕文件与视频进行结合以在视频中显示对应的字幕内容。这一步骤主要是将文本字幕与视频进行同步，使得字幕在合适的时间点显示指定位置。其主要包括：

字幕时间轴对齐：通过对视频和字幕的时间轴进行匹配，确保字幕在正确的时间点出现，并与语音和图像内容同步；

字幕渲染与叠加：将字幕按照时间轴的顺序逐帧渲染，并与视频图像叠加在一起。这样字幕就能在对应的时间点出现在视频中。

4. 系统实现

4.1 客户端

本系统采用了 Vue 组件技术实现 Web 客户端[7][8]，移动 App 端采用 UniApp 框架，并通过 H5 进行前端页面渲染。Vue 框架是一种流行的 JavaScript 框架，用于构建交互性强的单页面应用（SPA），Vue 提供了一种组件化的开发方式，使开发者可以将页面划分为独立的组件，提高了代码的可维护性和重用性。UniApp 框架是一个跨平台的应用开发框架，它允许使用 Vue 来开发同时运行在多个平台上的应用程序，UniApp 使用了一套基于 Web 标准的组件库和 API，使开发者能够轻松地将应用程序发布到不同的移动平台。

4.2 APIG

本系统采用了前后端分离和微服务架构，其中 APIG 是关键组件，用于解决微服务调用的问题。系统选择了分布式网关 Kong[9]，主要用于认证、API 请求限流和流量统计等功能。

认证：认证是一个关键的过程，用于验证用户身份并授权其访问 API。Kong 提供了多种认证插件，包括基于令牌的身份验证、基于密钥的访问控制和基于用户凭据的认证等，认证插件可以实现用户身份验证和授权访问的功能。

API 请求限流：限流是一种控制流量的机制，可以防止过多的请求对后端服务造成负载压力或破坏系统稳定性。Kong 自带的限流插件 Rate Limiting 可以基于时间窗口、请求速率或并发连接数等指标对请求进行限制，确保服务的可用性和稳定性。

此外，还可以在路由上进行限流，或在 consumer 上进行限流。

4.3 业务逻辑层

业务逻辑层主要采用 Python 语言开发，使用 Swagger 定义系统的 API 接口。系统使用 Gunicorn 作为 WSGI 服务器来运行应用程序，提供高性能的请求处理和并发能力。本架构使得业务应用程序不依赖于特定的会话状态来处理请求，每个请求相互独立，无需依赖共享状态，因此本设计方案有助于实现应用程序的水平扩展，能够通过增加更多的实例来扩展应用程序的处理能力，提高系统的可伸缩性和性能。

业务逻辑层服务通过 rest 接口和 SDK 方式调用领域层服务，实现业务和基础功能的解耦。

从功能上讲，业务逻辑层主要包含四部分：PPT 文件管理、讲解管理、Speaker 管理和自定义音频录制。

其中实体对象包含 PPT，讲解文档，Speaker，Voice，图片，字幕和视频。一个 PPT 对象拆分出多个讲解文档对象，一个 PPT 对象的每一页都会拆分为一张图片，一个 Voice 对象都是使用一个讲解文档对象，并选择一个 Speaker 对象，根据 Speaker 对象生成讲解文档的声音，PPT 的视频对象都是由其中的每一张 PPT 的图片对象和对应的 Voice 对象合成

PPT 文件管理

主要完成对 PPT 文件的 MetaData 元数据管理、上传、下载、格式转换、文件操作、文件搜索和权限管理等。

元数据结构设计主要包括文件名、唯一标志 id、所属人、大小、缩略图、创建时间、最近更新时间、对象 id、对象类型、uri、状

态和标签。

对文件的上传、下载、格式转换、搜索等 Swagger 定义的 API 接口如下：

表 1 文件操作 API

API	Description	Request type
update	Update an existing PPT file.	PUT
upload	Upload PPT files.	POST
findSlideByID(id)	Find by slide id.	GET
updateSlide	Update slide content.	POST

(id)		
deleteSlide (id)	Delete slide by id.	DELETE
findByStatus	Find slides by status.	GET
findByTags	Find slides by tags	GET

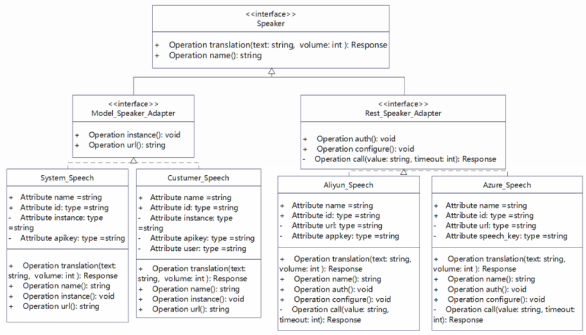

图 3 Speaker 管理类图

讲解管理

其功能支持多种讲解方式，包括独立的讲解和 PPT 注解方式。

其 API 接口实现如下：

表 2 讲解操作 API

API	Description	Request type
update	Update an existing interpretation.	PUT
add	Add a new slides interpretation.	POST
findSlideByID(id)	Find by slide id.	GET
updateSlide(id)	Update slide content.	POST
deleteSlide(id)	Delete slide by id.	DELETE

Speaker 管理

主要负责对各种类型的语音文件进行管理，包括适配系统预置模型、用户克隆模型以及与外部第三方语音转换服务的集成。Speaker 管理具体实现 UML 图如图 3 所示：

自定义音频录制

系统提供自定义音频录制，音频文件管理，音频文本管理等功能。为语音克隆提供源数据或原语音支持。音频数据结构约束主

要包含 id、name、maxtime、objectid、object_type 等。其中对 maxtime 的约束定义如下所示，

maxtime:

type: 浮点数

format: float

example: 500

此定义表示录制音频的最大时长为 500 秒，即 5 分钟：

音频录制主要 API 接口如下：

表 3 音频数据接口

API	Description	Request type
createVoice	Create voice.	POST
getTmplVoice	Get template voice.	GET
findVoiceByID(id)	Find voice by ID.	GET
updateVoice(id)	Update a voice.	PUT
deleteVoice(id)	Delete voice.	DELETE
uploadVoice	Upload voice file.	POST

4.4 领域服务层

4.4.1 文件管理

本系统的文件服务采用了 Java 语言，并利用 Apache VFS 库进行开发和实现。通过 Apache VFS 能够实现处理不同文件系统的读取、写入、复制、移动、删除等操作，实现统一管理文件系统的访问，并且通过扩展接口，可以实现对其他文件系统的适配，只需编写适配器来实现针对特定文件系统的操作。

VFS 对所有文件名都被视为 URI, 最基本的本地物理文件系统上的文件的访问 URI 格式。

4.4.2 格式转换

本系统格式转换服务采用 Python 语言 + FFmpeg 的方式开发。FFmpeg 是一个强大的开源多媒体处理工具，可以进行音视频的编解码、格式转换、剪辑等操作。结合 Python 和 FFmpeg 可以实现各种格式转换等服务，例如将不同音视频格式相互转换、提取音频或视频流、合并多个音视频文件等。Python 提供了丰富的库和工具，可以与 FFmpeg 进行集成，并通过调用 FFmpeg 的命令行接口或使用相应的 Python 封装库来执行各种媒体处理操作。

4.4.3 AI 平台

本系统的 AI 平台主要依赖于一个由多块 T4 GPU 组成的 Kubernetes 集群[10]。利用 Kubernetes 的 GPU 分片共享调度能力，实现了多个 GPU 算法实例的高效运行，从而提升了 GPU 利用率和多个 AI 推理服务的可靠性。主要的 AI 模型推理服务包括以下两种：

SV2TTS 是一种基于深度学习的算法模型，通过分析和学习特定说话人的语音特征，能够生成与该说话人相似的语音样本。SV2TTS 技术可以被应用于语音克隆、语音合成和语音转换等多种应用场景，为用户提供高质量的语音处理服务。

Erlangshen-Roberta-330M-Sentiment 是中文的 RoBERTa-wwm-ext-large 在 8 个中文领域的情感分析数据集，总计 227347 个样本上微调的一个 Bert 模型[11][12]。

通过利用 T4 显卡的并行计算能力，可以加速算法模型的推理过程，提高语音克隆和语音合成的效率和性能，能够加速语音相关任务的处理速度，显著提高语音应用的响应性能和用户体验。

4.4.4 语音生成

本文所设计的系统使用 Encoder 模型对文本进行 Embedding，然后将每个预处理的句子经过 Synthesizer 合声器和编码器 Vocoder 生成音频文件，并以 UUID 名称保存。语句预处理工作包括以下步骤：

文本正则化：文本正则化是解决文本发音一致性问题的关键步骤。本系统基于开源的 chinese_text_normalization 库实现了文本正则化操作。主要针对以下问题做正则化处理：非标准词的歧义；将特殊的文本转换为口语词；剔除掉无用字符；中文进行全半角字符转化等。

术语转换：通过术语对照表，将领域内的特定术语转换为正确的文本描述。例如“ML”转换为“机器学习”等。

音频序列管理：将预处理的文本按照句子结果拆分，每一个句子单独进行音频生成，并对整个音频序列进行

排序管理。基本实现逻辑如下：

表 4 音频序列管理

文本块	句子编号	文本内容	音频序列编码	音频文件	音频时长
文本块 1	句子 1	文本 1	1001	1001.	00:03:52,230
	句子 2	文本 2	1002		00:05:30,100
	句子 3	文本 3	1003		00:01:50,100
文本块 2	句子 1	文本 1	2001		00:01:30,100

Step1：对输入的 PPT 中的每一页完整讲解进行段落划分

Step2：每个段落看作为一个文本块，并对段落当中的句子进行切分

Step3：按照 XXXX 的数据格式来编码每一个句子。第一位为文本块的顺利编码，后三位为文本块中句子的顺序编码，001 代表第一个句子。

4.4.5 字幕生成

本系统以 SRT 文本字幕格式[13]为例。SRT 是最简单的文本字幕格式，后缀名为.srt，SRT 每个字幕段有四部分：字幕序号、字幕显示的起始时间、字幕内容（可多行）和空白行（表示本字幕段的结束）。字幕序号是顺序增加的，表示字幕是一系列连续的序列。字幕序号的值可以随意，起始值为 1 或 100 都能够支持，并不会影响字幕的显示。字幕序号是字幕段的一部分，不可删除。

4.4.5 视频合成

系统采用 Python 和 FFmpeg 开发，FFmpeg 是一款功能强大的开源多媒体处理工具，支持音视频编解码、格式转换、剪辑等操作。Python 与 FFmpeg 集成后，可实现各种格式转换服务，如音视频格式相互转换、提取音视频流、合并文件等。Python 提供了丰富的库和工具，可通过调用 FFmpeg 的命令行接口或使用 Python 封装库执行各种媒体处理操作。

5 系统部署

本系统的服务均采用容器化部署在 kubernetes 上，并通过 Deployment 和 HPA 实现服务实例的动态扩缩容，以确保服务的高可用性。其中 Deployment 是 Kubernetes 中的一个资源对象，它定义了如何创建和更新应用程序的副本集。通过配置 Deployment 能够指定要运行的容器镜像、副本数量以及其他部署相关的配置。HPA 是 Kubernetes 的一个控制器，它根据定义的指标自动调整 Deployment 的副本数量。通过配置 HPA 可以用来监测服务的指标，例如 CPU 使用率、内存消耗或自定义的指标。当指标达到阈值时，HPA 会自动增加或减少 Deployment 的副本数量，以满足服务的需求，并保持高可用性和性能。HPA 定义示例如下：

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: sv2tts
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: svtts-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      targetAverageUtilization: 50
  - type: Resource
    resource:
      name: memory
      targetAverageUtilization: 70
```

6 结束语

本文设计并实现一种利用 SV2TTS 技术的慕课视频自动生成系统。该系统通过集成文本处理、语音克隆和视频合成等模块，实现了将输入的文字内容转化为特定讲解人的语音样本，并将语音与相应的视频进行合成。基于该系统，用户能够根据文字内容生成具有特定讲解人声音的慕课视频，提供更加丰富和个性化的学习体验。此外，本系统存在一些限制和改进空间。例如，系统在语音合成过程中可能出现一些语音不自然或说话人失真的问题，需要进一步优化算法和模型。另外，系统的规模和性能还可以进行更进一步的扩展和优化，以满足大规模用户访问和高并发请求的需求。

参考文献：

[1]雷巧娟.智慧理念视域下的微课自动生成系统设计[J].自动化技术与应用, 2020(06): 144-147.

[2]曾永安.基于多技术融合的在线教育平台设计.自动化技术与应用[J]. 2019,38(10),142-145.

[3]Changyan Z,Jibin Y,Xiongwei Z, et al.Improving the performance of speech waveform synthesis using WaveNet fused with phase information[J]. Chinese Journal of Acoustics, 2022,41(01):1-19.

[4]丁云涛,才让卓玛,贡保加等.一种基于 WaveNet 的藏语语音合成方法[J].计算机仿真, 2023, 40(01): 295-299+538.

[5]孟利,沈郑燕,张泰雯.基于 FFmpeg 提取目标人物语音的应用研究[J].信息系统工程, 2023(03): 74-76.

[6]余海鑫,丁航,李文邦.基于 Vapoursynth 和 FFmpeg 的视频编辑[J].电子世界, 2022(01): 164-165+167

[7]褚建萍.基于 Vue 的数据可视化系统研究[J].电子技术与软件工程, 2022(18):234-237.

[8]郭艳华.基于Vue框架的海量数据处理系统设计[J].信息与电脑(理论版),2022,34(23):16-18.

[9]何运田,张青清.基于Kong和Elasticsearch的私有云API 网关及监控系统的设计与实现[J].计算机应用与软件,2022,39(11):136-140.

[10]刘祥,胡瑞敏,王海滨.基于 Kubernetes 的 AI 调度引擎平台[J/OL].计算机系统应用:1-9[2023-07-07].

[11]刘祥,胡瑞敏,王海滨.基于 Kubernetes 的 AI 调度引擎平台[J/OL].计算机系统应用:1-9[2023-07-07].

[12]刘斐瑜,俞卫琴.融合 BERT 与注意力的文本情感分析模型[J/OL].软件导刊:1-6[2023-07-07].

[13]佟国香,李乐阳.基于图神经网络和引导向量的图像字幕生成模型[J].数据采集与处理,2023,38(01):209-219.