项目报告 1

Shandong University

March 1, 2020

高德琛

201705130099

Contents

1 立项准备

1.1 日志概述

根据实验要求，“实验一” 任务包括：团队建立及分工、需求分析及可行性分析、协作平台确定三部分。本文档中记录团队调研过程中，个人的工作记录。

1.2 团队组建

联系组队之后团队共有四人，经过投票选举确定队长为苑宗鹤同学接下来根据队员的专长以及软件工程所需职能确定团队分工。

Table 1: 团队分工

姓名	软件工作	调研工作
张火亘	前端技术	前端技术栈对比分析
苑宗鹤	后端、测试技术	服务器框架、后端业务技术选型
曹远	后端、测试技术	测试工作分析、测试计划
高德琛	算法、模型可行性	后端、算法、深度学习调研

1.3 协作平台

我们将协作平台分为三类：代码版本控制、共享文档、团队线上沟通及项目管理。并且分别对各类工具进行了调研。

1.3.1 代码版本控制

此类工具有 GitHub、SourceForge 等。其中 GitHub 以 Microsoft 为依托，目前是全球最大的开源生态社区，拥有大量开源工作者及优质代码资源。

我们团队选择 Git 的原因，主要原因在于复杂的软件工程需要强有力的版本控制工具，并且能够提供多人协作。而 Git 正符合此需求，如Figure 1中展示了同一项目当中，功能拓展、Bugs 修复在 Repo 当中留存的 commit 历史。通过 commit 记录，我们可以阅览多人协作的代码维护记录，了解项目的扩展过程。通过对照不同版本历史代码，如Figure 2，我们可以清晰了解队员的代码修改。此外还可以回退版本。这些特性得以让团队开发有条不紊地进行。

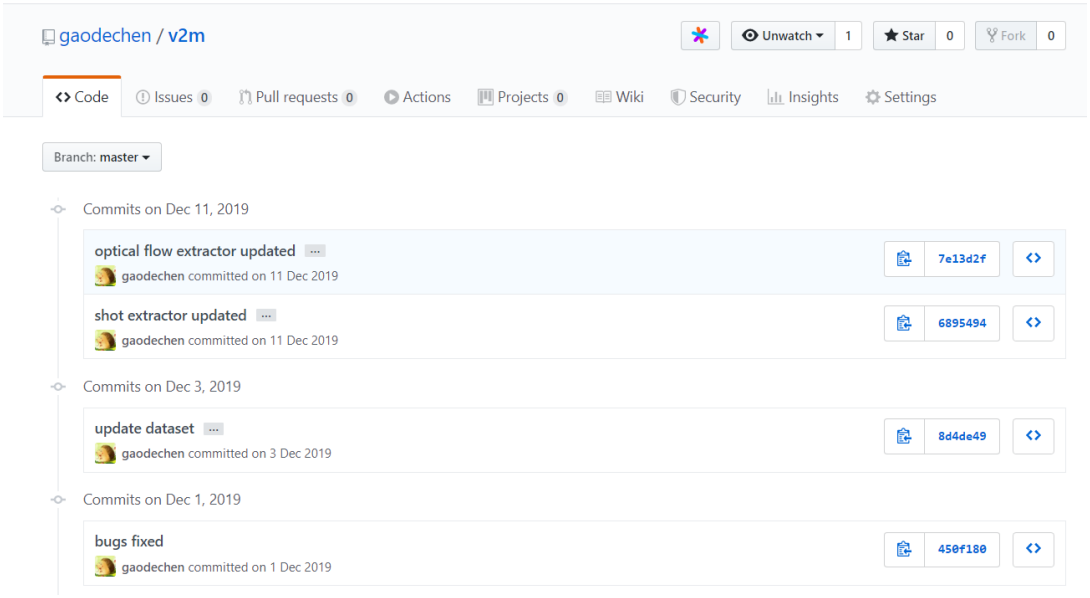

Figure 1: GitHub commit 版本历史

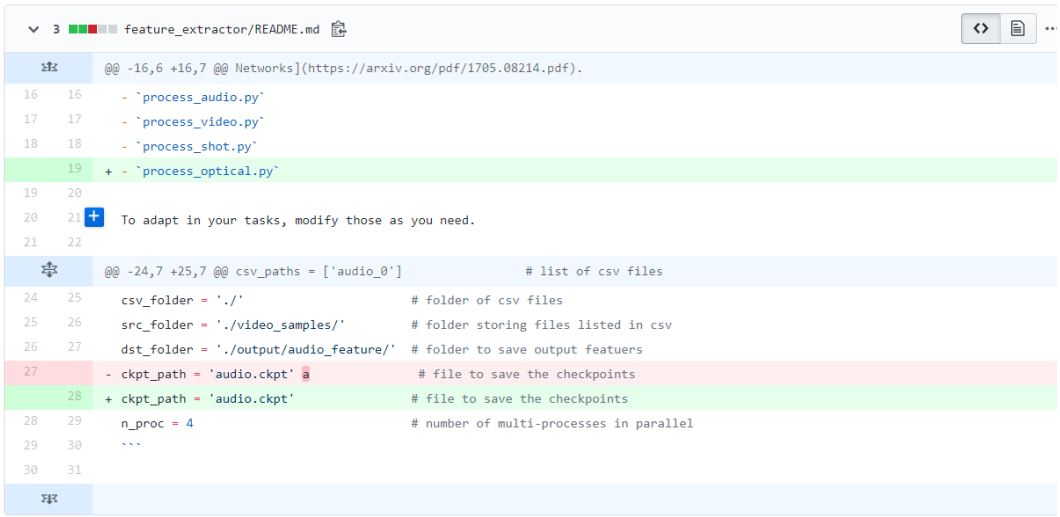

Figure 2: GitHub 版本对比

1.3.2 共享文档

考虑到快速开发下的时间成本，《技术文档》我们选择使用腾讯文档进行协作。腾讯文档允许多人共同编辑 Office 格式的文档。但是 Office 类富文本格式的问题在于，不同平台下兼容性不一致。Web 端的共享编辑时常出现格式与本地不符，造成编辑排版困难。

此外为了简化排版，节约时间成本。我们使用 LaTeX 这类纯文本格式进行个人日志的编写。而 Overleaf 提供了线上编辑 LaTeX 的服务。基于纯文本和线上平台的好处在于，不需要考虑兼容性问题。即插即用，完全免费，不需要安装配置。并且排版相比 Office 而言更为专业，节约时间成本同时效果更佳。

Figure 3为 Overleaf Panel，文档线上存储管理，可以共享；Figure 4为 Overleaf Editor，编辑器左侧为纯文本编辑器，可以在线实时渲染预览。

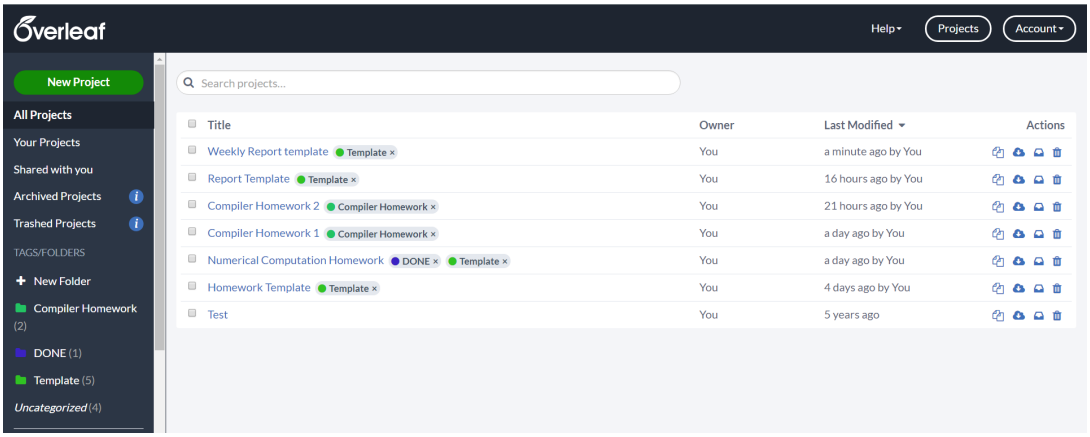

Figure 3: Overleaf Panel

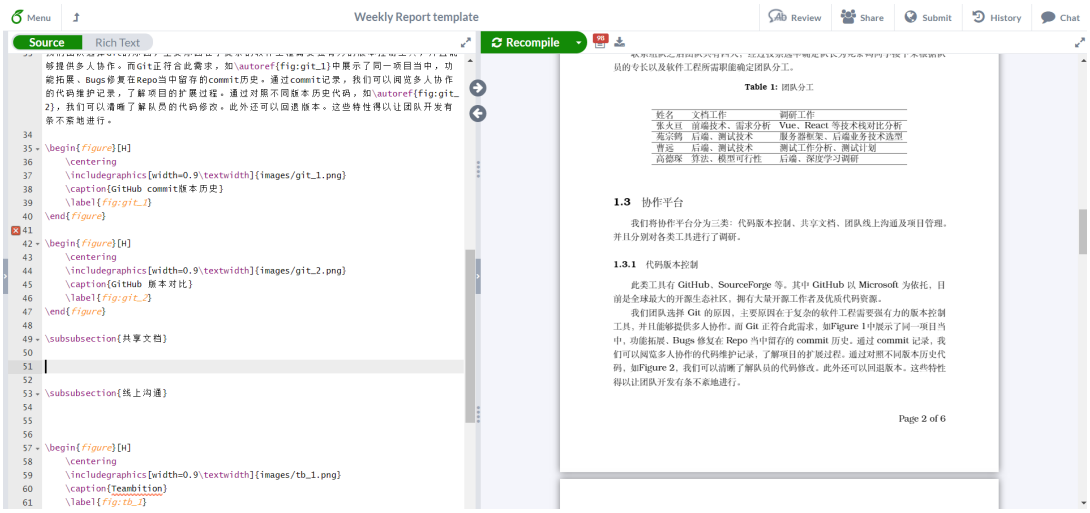

Figure 4: Overleaf Editor

1.3.3 线上沟通

考虑到团队较小，没有公司编制，所以我们首先放弃了 Microsoft Teams、Slack、Ding Talk 等工具。因为视频、音频使用 IM 工具提供的服务，即可满足沟通需求。此时若选择大型的团队沟通工具只会增加操作成本、时间成本。

尽管如此，我们还是需要专业的 TODO 类工具维护开发周期，为此基于轻便、易用的原则我选们定了 Teambition。Figure 5中我们将软件工程划分为多个阶段，每一阶段涉及不同任。Figure 6中项目管理者可以将 TODO 任务指派给不同队员进行负责、协作。

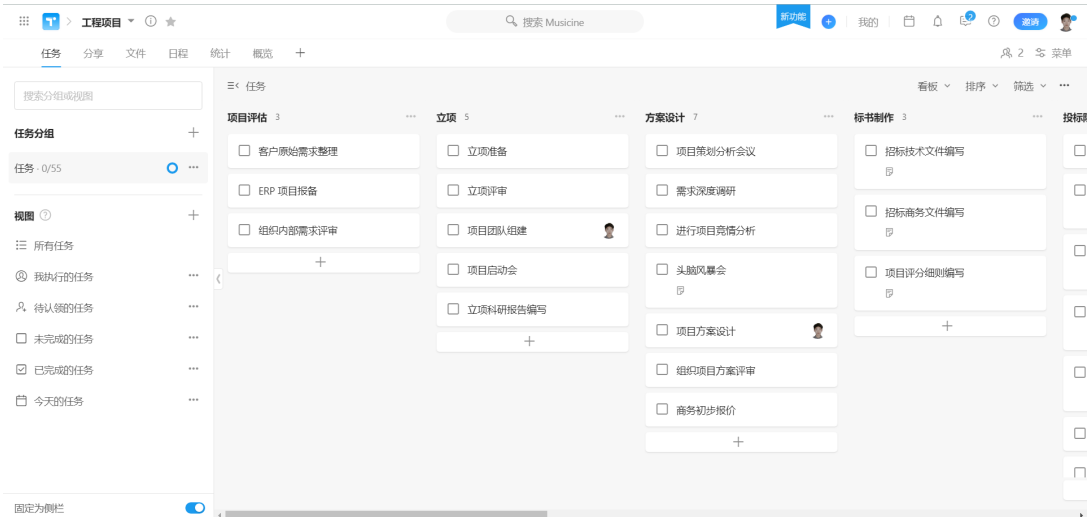

Figure 5: Teambition TODO List

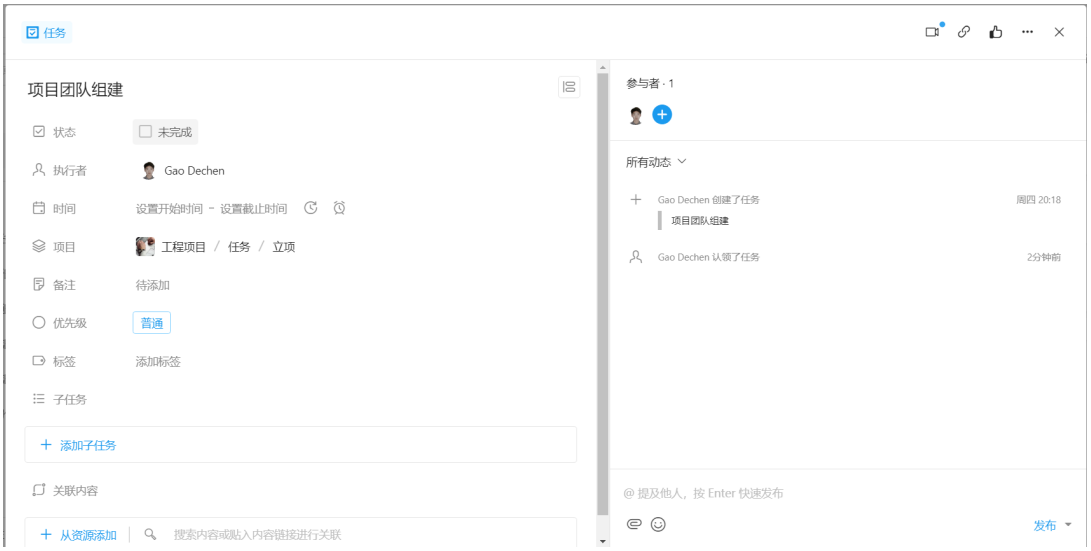

Figure 6: Teambition Task Manager

2 方案设计

2.1 项目及技术选型

经过投标我们命中了“基于深度学习的短视频智能配乐及剪辑系统”项目。该项目具有一定创新性及工作量。经过队员提议和讨论，我们初步确定了系统需求概要。

- 作为短视频平台，用户可以制作、编辑、上传短视频作品，意味着前后端需要提供相应流媒体技术支持
- 作为短视频社区，用户可以共享作品，阅览其他用户作品，意味着推荐系统及检索功能的需求存在
- 作为视频及配乐工具，具有 Music Transcription, Music Generation, Rhythm Chord Generation 需求，依赖深度学习及 Music Information Retrieval 技术

2.2 项目方案设计

我们选择了前后端分离的开发方式，解耦前后端逻辑，前后端队员可以分别负责，方便对接与测试。

2.2.1 前端

Figure 7: React Vue

经过调研对比 React 及 Vue 技术栈，我们选择了 Vue 框架编写前端。张火亘同学为前端负责人。Vue 为目前主流的前端框架，具有良好的生态及优质的文档支持。

由于有过 React、Redux 开发经验，也曾考虑过 React 技术栈。但是经过对比，我们发现 React 组件设计较为繁琐，加入 Redux 之后尽管数据流逻辑清晰，但是编码成本依旧较高。故而团队希望通过 Vue 最大化开发效率，适应快速开发需求。

2.2.2 后端

Python、Node.JS 作为便捷开发首选，两者都具有良好的生态及文档支持，且队员语言基础较好。但是考虑到需求涉及大量的流媒体处理算法、深度学习，为了降低对接的时间成本，我们直接选用了 Python 作为后端语言。

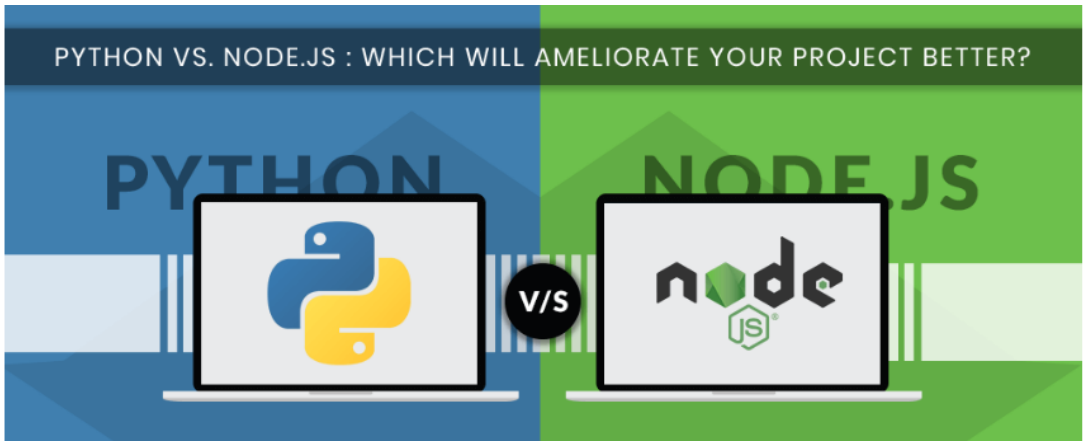

Figure 8: Python Node

Python 作为服务器而言，有 Django、Flask、Tornado 等诸多优质框架，其可扩展性运行我们对后端逻辑根据业务需求进一步优化拓展。接著 Nginx 等服务器我们可以进一步快速定制服务器以提升性能。

同时 Python 对于各类数值运算、音频算法支持极佳，有 librosa、numpy 等运算库的强力支持。同时深度学习方面更是原生支持 PyTorch 及 TensorFlow，故而 Pytho 是后端的不二之选。

数据库设计上，我们采用了双数据库，作为快速开发，考虑到 MongoDB 以及 Redis 本身良好的生态及性能优化，软件工程中不再涉及数据库的底层优化。

Figure 9中可以看出二者都是目前各类应用的数据库技术主流选择。我们选择 MongoDB 的原因在于 NoSQL 特性灵活且可读性强，易于后期业务扩展，并且性能也很优异。Redis 作为内存行数据库，可以满足检索、推荐系统等业务需求，使用缓存加速检索业务。

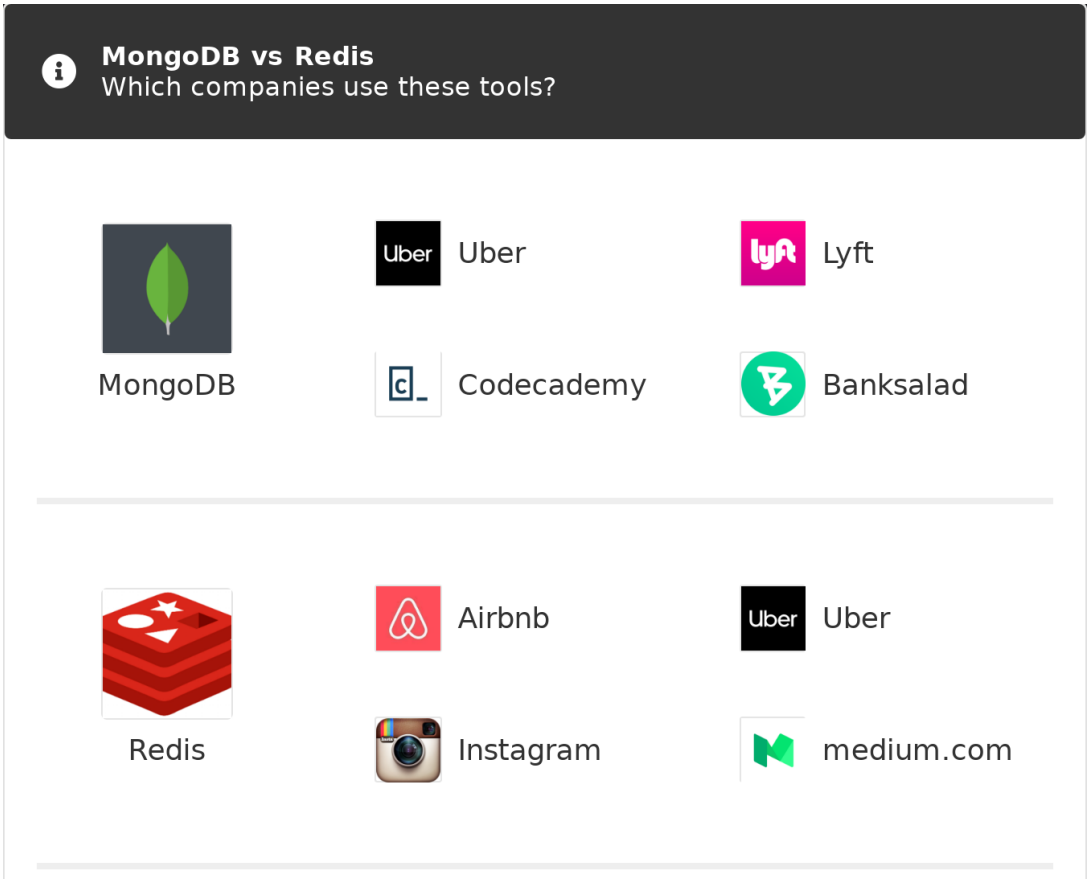

Figure 9: MongoDB Redis

2.3 预算评估

单元测试、系统测试阶段采用云服务器进行，其他时间内进行本地的开发调试。故而开发成本集中在测试阶段的云服务器开销以及深度学习模型训练方面。我们积极的联系了学校实验室，实验室的 GPU 资源可以帮助团队节约花费。据此估计支出约为 500 元。

3 算法及模型可行性

除了进行平台、前后端技术选型之外的调研工作，我的主要工作是算法、模型方面的调研。根据业务需求，我将算法及模型技术分为音频技术、视频技术、检索技术。

3.1 符号音乐生成技术

作曲家创作乐谱的过程，即创作符号音乐的过程。如今“人工智能的音乐创作”技术也多指代“符号音乐生成”技术。

音乐本身具有序列性，NLP 领域的方法工具同样被尝试运用在符号音乐的研究当中，下面我们列举音乐生成自 2017 年以来的四项代表性工作。

1. *MusicVAE*: Google Magenta 将 VAE 变分自编码器应用于音乐生成当中。利用隐空间性质可以完成乐曲的变奏；该架构同样可以用于伴奏生成。

2. *Music Transformer*: Transformer 将注意力机制成功应用在自然语言处理当中，同样地，Transformer 技术也被成功运用在了符号音乐生成当中。2018 年，Google Magenta 提出了 Music Transformer。无论音乐性及生成性能都达到了 SoTA 水平。

3. *XiaoIce Band*: KDD 2018 提出的 XiaoIce Band，使用 Encoder-Decoder 架构对和弦、节奏、旋律序列分别建模，以和弦为基础生成其他轨道。之后对多轨道联合建模，实现了多轨流行音乐的编曲技术。

4. *MuseGAN*: MuseGAN 旨在通过多轨道建模直接完成和谐、对位的多轨道音乐。

3.2 音频转谱技术

“基于生成的视频配乐技术”目前相关工作及应用较少，且相关工作中普遍需要“特定作曲规则”这类较强条件加成的原因，原因在于“视频-符号音乐”关联数据集的缺失。

广告配乐、电影配乐、MV 尽管可用数据量大，但是其中的音乐形式全部为“音频音乐”（Audio Music）。故而无法直接利用配乐数据进行训练。

目前处于 SoTA 水平的转谱技术，属于 Google Magenta 2017 年提出的 Onsets and Frames 架构，而该技术也仅仅是钢琴数据集上进行了模型设计、训练及测试。尽管对于钢琴音乐有着较高准确率，然而对于配器复杂、包含人声的音频，准确率依旧不够理想。

3.3 高保真音频合成技术

接下来我们将阐述为何目前的音乐生成技术限于“符号音乐”，而非“音频音乐”，这同样能够解释为何 uVloger 保留符号音乐生成这一方案。

音频音乐的分析研究当中，往往使用时频方法将音频转换为图像表示，故而音频合成技术实质上属于图像的生成技术，然而频谱图等图像与真实世界图像不同，“音符”作为音乐世界的“物体”，其图像构成复杂，且音符之间会产生变形。这就导致了转谱技术、音频合成技术的困难。

2019 年 Google 在 ICLR 上提出了目前音频音乐合成技术的 SoTA 技术，GANSynth。在此之前的 SoTA 架构基于 2016 年提出的 WaveNet。

尽管 GANSynth 相对 WaveNet 性能有极大提升，且合成音频效果更为理想，但以单音为单位生成实质上还是需要“符号音乐”序列作为输入，且性能无法满足整首音乐序列合成。

3.4 基于生成的视频智能配乐技术

基于生成的配乐技术工作集中在学界，且近年来解决方案并没有实质性变化。问题在于“视频-符号音乐”数据集的缺失，而音频音乐尽管数据量大，但是转谱技术不够理想，无法通过音频音乐转换为符号音乐来构造数据集。

故而相关工作多采用条件限制较强的作曲、配乐规则来限制配乐输出。2016 年的实时作曲系统工作当中将情感、动作、颜色特征，直接映射到配器方案、调性、节奏方案当中。与 1994 年的一项电影配乐工作相比，尽管加入了更先进的高级语义以及实时性，其本质思想依旧类似。

可以看出，较强的条件限制尽管可以一定程度上使作品契合视频，大大简化了问题的难度，但同时会丧失作品多样性。

3.5 视频-音频跨域研究

1. MM 2018 年发表的 Dance with Melody 进行了舞蹈动作与音乐的联合学习。
2. ECCV 18 年 Learn, Listen and Learn 当中给出了声音-视觉对应学习的方法。
3. 2019 年 VideoBERT 提出，利用 BERT 的 MASK 技术设计视频-文本多模态预训练任务，并在“视频描述生成”问题当中取得了 SoTA 水平。

3.6 视频分析技术

相比音频分析技术而言，视频技术近年来更为成熟。

1. 高层语义抽取：视频高层语义信息应用于场景识别、动作识别等问题当中。uVloger 预处理过程当中，采用 CVPR 2018 年发表的 3D ResNet。

2. 镜头切换检测：视频场景切换检测属于视频分析领域的经典问题。此项工作与 uVloger 的关联性在于，场景切换应尽可能契合音乐鼓点，使得配乐作品更具有节奏感，这同样是专业视频配乐的要求。预处理的过程中，我们采用了 2017 提出的 SBD 算法加速场景分析。

3. 光流表示法：光流提取时间、空间开销大。uVloger 的预处理 YouTube-8M 数据集的过程当中，采用了 2016 年提出来的 DIS 稠密光流算法加速光流信息的提取。

4. 视频摘要技术：视频摘要近 20 年以来的研究主要集中于无监督式方法，AAAI 2018 提出的强化学习视频摘要方法目前取得了 SoTA 水平，作为 uVloger 计划参考的解决方案。

3.7 检索技术及推荐系统

检索技术、推荐系统如今应用极为广泛，同时具有许多算法的优质开源工作及框架。

在推荐系统选型上，我们选择了资源较多、实现更为便捷的 Model-Based 推荐系统，并按照软件架构标准设计推荐引擎。由于前后端各类逻辑解耦，我们可以将推荐引擎便捷地接入后端架构当中。

References

[1] JunIchi I Nakamura et al. “Automatic background music generation based on actors’ mood and motions”. In: *The Journal of Visualization and Computer Animation* 5.4 (1994), pp. 247–264. ISSN: 10991778.

[2] Jiashi Feng, Bingbing Ni, and Shuicheng Van. “Auto-generation of professional background music for home-made videos”. In: *Proceedings of the 2nd International Conference on Internet Multimedia Computing and Service, ICIMCS’10* (2010), pp. 15–18.

[3] Joan Serrà et al. “Unsupervised detection of music boundaries by time series structure features”. In: *Proceedings of the National Conference on Artificial Intelligence* 2.2009 (2012), pp. 1613–1619.

[4] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. “SoundNet: Learning sound representations from unlabeled video”. In: *Advances in Neural Information Processing Systems* Nips (2016), pp. 892–900. ISSN: 10495258.

[5] Jian Wu et al. “A Hierarchical Recurrent Neural Network for Symbolic Melody Generation”. In: Mozer 1994 (2017).

[6] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: Mlm (2018).

[7] Hao Wen Dong et al. “Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment”. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* (2018), pp. 34–41.

[8] Cheng-Zhi Anna Huang et al. “Music Transformer”. In: (2018), pp. 1–14.

[9] Adam Roberts et al. “A hierarchical latent vector model for learning long-term structure in music”. In: *35th International Conference on Machine Learning, ICML 2018* 10 (2018), pp. 6939–6954.

[10] Taoran Tang, Jia Jia, and Hanyang Mao. "Dance with melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis". In: *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference* (2018), pp. 1598–1606.

[11] Hongyuan Zhu et al. "XiaoIce band: A melody and arrangement generation framework for pop music". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2018), pp. 2837–2846.

[12] Patrick Hutchings and Jon Mccormack. "Adaptive Music Composition for Games". In: July (2019), pp. 1–10.

[13] Zhong Ji et al. "Video Summarization with Attention-Based Encoder-Decoder Networks". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2019), pp. 1–1. ISSN: 1051-8215.

[14] Jong Wook Kim and Juan Pablo Bello. "Adversarial Learning for Improved Onsets and Frames Music Transcription". In: (2019).

[15] Chen Sun et al. "VideoBERT: A Joint Model for Video and Language Representation Learning". In: (2019).

[16] Hsin-ying Lee Xiaodong et al. "Dancing to Music". In: NeurIPS (2019), pp. 1–11.

[17] Nelson Yalta et al. "Weakly-Supervised Deep Recurrent Neural Networks for Basic Dance Step Generation". In: (2019), pp. 1–8.

[18] Yi Yu and Simon Canales. "Conditional LSTM-GAN for Melody Generation from Lyrics". In: (2019).

[19] Relja Arandjelovi. "Look , Listen and Learn". In: (), pp. 609–617.