文件编号：从容应队-SWC2020-从容应队

受控状态：■受控 □非受控

保密级别：□公司级 □部门级 ■项目级 □普通级

采纳标准：CMMI DEV V1.2





素音

**Divoice**

项目开发文档

**Version 1**

2019.12.19

**Written by 从容应队**



**All Rights Reserved**

目录

[1 项目概述 1](#_Toc22846313)

[1.1 项目背景 1](#_Toc22846314)

[1.2 项目定位 1](#_Toc22846315)

[1.2.1 应用场景 1](#_Toc22846316)

[1.2.2 目标人群 1](#_Toc22846317)

[1.3 项目方案 1](#_Toc22846318)

[1.4 项目目标 1](#_Toc22846319)

[1.5 项目价值 1](#_Toc22846320)

[2 开发计划 1](#_Toc22846321)

[2.1 最终呈现形式 1](#_Toc22846322)

[2.2 主要功能描述 1](#_Toc22846323)

[2.3 运行环境 1](#_Toc22846324)

[2.4 验收标准 1](#_Toc22846325)

[2.5 关键问题 1](#_Toc22846326)

[2.6 进度安排 1](#_Toc22846327)

[2.7 开发预算 1](#_Toc22846328)

[3 可行性分析 2](#_Toc22846329)

[3.1 技术可行性分析 2](#_Toc22846330)

[3.2 资源可行性分析 2](#_Toc22846331)

[3.3 市场可行性分析 2](#_Toc22846332)

[4 需求分析 2](#_Toc22846333)

[4.1 数据需求 2](#_Toc22846334)

[4.1.1 静态数据 2](#_Toc22846335)

[4.1.2 动态数据 2](#_Toc22846336)

[4.1.3 数据词典 2](#_Toc22846337)

[4.1.4 数据采集 2](#_Toc22846338)

[4.2 功能需求 2](#_Toc22846339)

[4.2.1 \*\*功能模块 2](#_Toc22846340)

[4.3 性能需求 4](#_Toc22846341)

[4.3.1 时间特性 4](#_Toc22846342)

[4.3.2 适应性 4](#_Toc22846343)

[4.4 界面需求 4](#_Toc22846344)

[4.5 接口需求 4](#_Toc22846345)

[4.5.1 硬件接口 4](#_Toc22846346)

[4.5.2 软件接口 4](#_Toc22846347)

[4.6 其他需求 4](#_Toc22846348)

[5 概要设计 4](#_Toc22846349)

[5.1 处理流程 4](#_Toc22846350)

[5.2 总体结构设计 4](#_Toc22846351)

[5.3 功能设计 4](#_Toc22846352)

[5.4 用户界面设计 4](#_Toc22846353)

[5.5 数据结构设计 4](#_Toc22846354)

[5.6 接口设计 4](#_Toc22846355)

[5.6.1 外部接口 4](#_Toc22846356)

[5.6.2 内部接口 4](#_Toc22846357)

[5.7 错误/异常处理设计 5](#_Toc22846358)

[5.7.1 错误/异常输出信息 5](#_Toc22846359)

[5.7.2 错误/异常处理对策 5](#_Toc22846360)

[5.8 系统配置策略 5](#_Toc22846361)

[5.9 系统部署方案 5](#_Toc22846362)

[5.10 其他相关技术与方案 5](#_Toc22846363)

[6 数据库设计 5](#_Toc22846364)

[7 详细设计 5](#_Toc22846365)

[7.1 \*\*功能模块 5](#_Toc22846366)

[7.1.1 功能描述 5](#_Toc22846367)

[7.1.2 性能描述 5](#_Toc22846368)

[7.1.3 输入 5](#_Toc22846369)

[7.1.4 输出 5](#_Toc22846370)

[7.1.5 程序逻辑 5](#_Toc22846371)

[7.1.6 限制条件 5](#_Toc22846372)

记录更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **更改原因** | **版本** | **作者** | **更改日期** | **备 注** |
| 1 | 创建 | 1.0.0 | 队员A | 2019/12/19 | 创建项目开发文档 |
| 2 | 补充 | 1.0.1 | 队员A | 2019/12/22 | 文档的第一、二部分撰写 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 项目概述

## 项目背景

随着互联网的不断普及，中国互联网和移动互联网的发展也愈发成熟，人们手中的手机也愈发智能化，网络成为了我们接触世界的一扇极其重要的门扉，而手机则是人们叩开这扇大门的钥匙——相比电脑更加方便、灵巧。同时，在互联网这座庞大的信息海洋里，为了更好地展示自己的才能以及不同的生活场景，使更多的人能够足不出户便遨游世界，网络主播、up主等新型职业应运而生，并不断发展壮大，网络直播、录播、小视频、vlog等新型娱乐方式也早已走入人们的生活之中。

然而，在观看这些大多由个人制作的产品时，我们却发现了一个问题，大多数主播、up主受制于设备不专业的问题，且并没有掌握专业的音频编辑技能，从而导致产出音频、视频作品往往难以如人意，在吸引观众、粉丝上的竞争力也就难免逊色于同行了。这一问题也就将大多数欲闯入主播、up主这一行业的人拒之门外了。

普遍来说，解决音频降噪、混音等等音频问题的方法总结起来有两个：

1. 买更好的硬件，例如声卡、话筒等，这类设备依据性能也分为高中低三个档次，价格 从几百元到几万元不等，而价格偏低的产品也往往性能堪忧。而高价的产品也只是依据通用的算法来处理普遍的问题，对一些特殊的情况也会出现无法满足用户需求的情况，例如背景噪声实在太过嘈杂，甚至大于人（主）声的情况。
2. 寻找专业音频处理师或者调音师，为自己提供更加精细的、更加优质的音频处理服务，这样的情况下音频处理会更加多选以及符合用户的特定要求，但随着服务而来的是价格也愈发昂贵，而大多数个人主播、up主难以承受这样庞大的支出。

而除了上述情况外，主播要求不高的话，还可以通过找寻一个安静度极高的环境来降低环境噪音，但此举也只是满足了降噪的需求，无法实现对音频的多元化修饰，这样产出的作品吸引力也往往较低而无法达到预期。一些up主也会自学音频编辑处理的技能，但还是免不了每一次音频处理后长时间的编辑处理。

为解决用户降噪难，音频处理难的问题，结合如今正大热的机器学习技术以及新兴的快应用技术，本团队构想出Divoice，期望能够依据用户提供的音频，标记并分离出不同的声音，已方便用户实现降噪、主音凸显的功能，并提供一系列主流音频处理功能：混音，增强选定波段声音，添加音轨效果等辅助音频编辑功能。满足主播、up主以及普通用户对音频处理的需求，节省出80%以上音频编辑时间，降低音频、视频录制的门槛，使更多的普通大众也能享受到创作出优良的作品，给自媒体行业加速。

除此之外，Divoice也会随着用户提供个人专属音频素材以及反馈继续学习，更加精确识别与提取用户声音，为用户提供更加优质的音频处理服务。

## 项目定位

### 应用场景

无论是主播、up主还是普通用户，当用户需要在嘈杂的环境中录制音频时，当用户想要获取一段音频的伴奏或是纯净人声时，当用户苦于电脑处理音频太过繁琐时，当用户渴望听清嘈杂环境中的特定声音是，Divoice都可以为用户提供专业、优质、个性化的音频分离服务和强大的音频处理服务，用户仅需要提供自己录制的音频，并选择性的提供单一音色的声音素材，Divoice便可以依据强大的计算能力，智能分离当前音频中的不同声音并生成不同的音轨存储这些分离出的声音，再依据用户的选择对当前所有音频进行处理，快速、便捷地生成用户所需音频文件，极大地节省用户的时间。

### 目标人群

此项目适用于有音频处理需求的人，主要包括：

1. 主播、up主，帮助他们能够仅凭手机就能实现对自己录制文件的降噪、混音处理， 节省时间、降低成本，大大提高工作效率。激励普通大众也生成属于自己的特色音频/ 视频文件。
2. 企业的宣传部门与音频工作有关的工作人员，打造优质的企业之声。
3. 专业音频处理人员，为其提供纯净的声音素材，大大降低工作难度，灵活工作时间 与办公地点。

## 项目方案

对于如何实现音频分离功能的问题，主要涉及到两个方面，音色识别与音轨分离。其中重点就是如何在一段混音中识别出不同音色的声音。对于音色识别问题，本团队联想到Siri的语音操控的功能，决定采取采样甄别进行识别，系统库中有预先训练好的一系列常见声音样本模型；此外，用户在使用过程中，对于自己的声音也可以进行采样录入，加强对自己的声音的识别度，从而满足用户的个性化的需求。

音轨分离功能则是基于音色识别的基础上，对识别出音色的波段进行比对提取。

【参考文献】Uhlich, S. , Porcu, M. , Giron, F. , Enenkl, M. , & Mitsufuji, Y. . (2017). Improving music source separation based on deep neural networks through data augmentation and network blending. *ICASSP 2017*.

其余音频编辑处理使用前端音频处理的核心模块ffmpeg，使用其提供的录制，转化以及流化音频的解决方案，再此基础上再丰富Divoice的功能。

流程：获取音频文件->解码后传送给worker->计算处理->以事件的方式返回处理结果

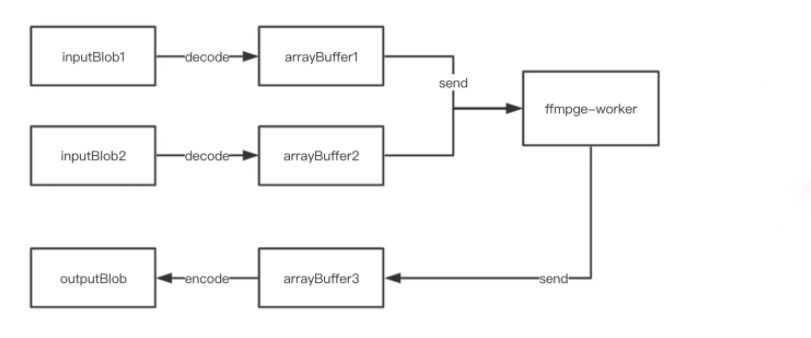


图1.3.1ffmpeg模块音频处理流程

## 项目目标

Divoice期望实现一个集自动分离音频、智能降噪、个性化、综合性的音频编辑系统。

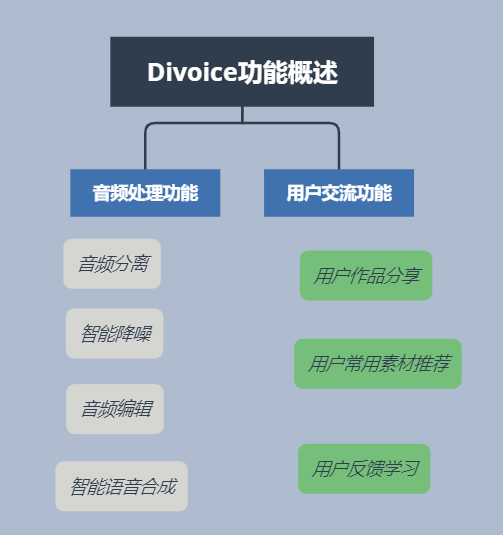


图1.4.1 Divoice功能概述

Divoice会首先对用户提供的音频进行接收分类，分类包括直接采用云端数据模型比对、用户提供原音素材比对两种方式。前者更加便利，但是在精确度上不及后者，后者更加适合个性化要求较强的专业音频处理人员。

之后Divoice会根据识别出的声音种类分离提取出不同声音的文件供用户选择。

然后Divoice也会自带有常用的音频处理工具，用户可以利用这些工具就给自己的音频加渲染、整合、剪切等等。方便用户进一步处理音频文件。

最后Divoice会生成用户所需的音频文件，选择保存到本地、分享在Divoice社区或者分享到其他平台，实现对作品的保存、传播。

Divoice还具备一定的云端素材库，可以推荐给用户使用，让用户的作品更加丰富立体。

## 项目价值

在当今主播、up主行业兴起，且一系列如喜马拉雅、全名k歌等等音频类app社区兴起之后，越来越多的人们认识到了声音的魅力与能量，也都纷纷加入其中，分享自己的声音与体悟。

但是，大多草根出身的主播、up主缺乏音频处理的专业知识，也没有足够的资金去购买以及雇佣专业人员协助自己创作出优秀的音频作品，受限与设备以及资金的问题，产品往往达不到预期，也更加难以吸引观众。

根据发放的调查问卷显示，70%以上的调查人群都认为在观看或者收听他人上传的作品时，音频质量是他们选择的一个重要参考点，超过半数的受访对象对会在音频质量不佳时通常选择放弃收听/看当前作品。

在受访主播中，我们发现58.9%的主播或多或少认为音频质量不佳是自己作品受限的一大原因之一。且无论是观众还是主播都有超过半数的人群表示很期待Divoice的出现，帮助他们更好地“操控”声音。

图1.5.1调查报告结果分析（音频质量对观众的影响）

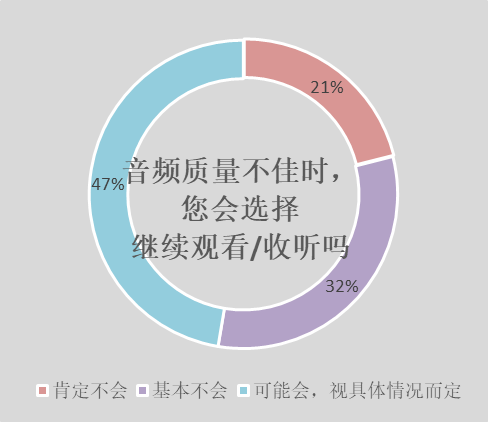
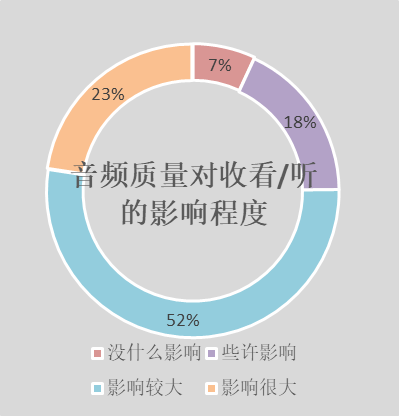


图1.5.1调查结果分析（对Divoice期待值）

# 开发计划

## 最终呈现形式

Divoice最终呈现为一个能流畅运行的快应用程序，运用深度学习技术为用户实现音频自动分类、分离，进一步协助用户实现智能降噪，混合、编辑音频的功能。整个快应用将由两个核心模块构成——音频处理模块和社区分享模块。音频处理模块能够为用户提供有效、便捷的音频处理功能，保证原文件不丢失的情况下，网络端处理生成分离出的音频文件，根据音色识别分离的音频在降噪以及混音方面将呈现出更加强大的功能。社区分享模块将为用户提供便捷的作品分享服务，编辑经验分享服务，以及优秀作品、声音素材推荐服务。

## 主要功能描述

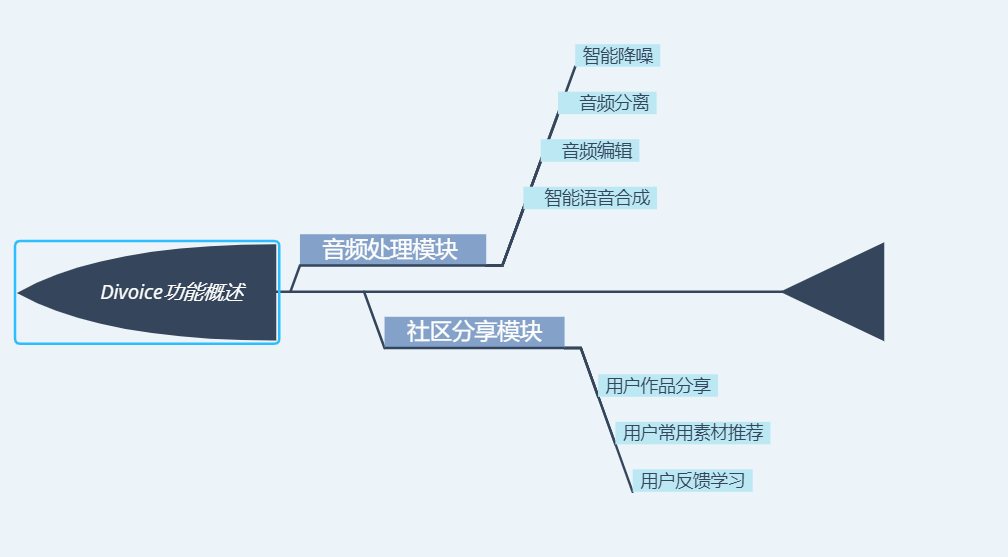


图2.2.1 Divoice两大功能模块概述

Divoice主要分为两大功能模块——音频处理模块和社区分享模块。

音频处理模块会对用户提供的音频进行接收分类，之后根据识别出的声音种类分离提取出不同声音的文件供用户选择编辑。同时此模块也配备有常用的音频处理工具，方便用户使用这些工具直接给自己的音频加渲染、整合、剪切等。

社区分享模块对于用户处理完毕后的音频文件，提供保存到本地以及分享到其他平台的功能，以方便作品的保存与传播。Divoice也提供云端素材库，给予用户在处理音频时所需素材，使用户使用更加便捷。

## 运行环境

Android端快应用运行环境，无需下载。暂无iOS端应用。

## 验收标准

## 关键问题

音频分离的重难点在于对音色特征的识别与提取。在一段混合有多种音色的音频文件中，音频文件存储的音色并非单一可分的音色，而是由于不同音色的声音谱频相互叠加从而被认定为是另一种音色的声音。这是由于声音的收集与存储方式所引起的。对于单一的声源，其音色特征值能够方便地被提取出来，可一但出现多种声音相互混杂的情况就会显得颇为棘手。对于盲源等未知特征值的声源，这样的处理识别就会显得更为艰难。

所以本团队选择针对性地提出解决方案——针对人声与背景人声、背景音乐的分离降噪问题。在语音信号处理领域，目前的语音识别与降噪增强算法仅能够识别和处理混油环境噪声的语音信号，对于“鸡尾酒会问题”则显得束手无策。在查阅相关资料后，发现了“盲源分离算法”，可以满足在源信号矢量的各个分量相互独立的情况下，允许信号幅度的不确定性和信号分量顺序的不确定性来解决这一问题。

同时，对于主播以及up主们，他们通常碰上的问题在于分离背景音与人声，而本团队的机器学习模型恰恰能够满足他们大多数的要求，以更好地契合他们的工作。

为了进一步增强声音分离的精确度以及效率，对于跟高需求的用户，本团队也提供样本采样的选项，由用户提供原音样品，交付与Divoice采样后，能够更加高效、精准地满足用户的需求。随着用户的不断使用，Divoice也会记录下用户的使用记录，对用户常用音色进行特征值存储，提高模型运算效率。

## 进度安排

## 开发预算

# 可行性分析

## 技术可行性分析

## 资源可行性分析

人力资源包括队员四人，包括深入学习过深度学习技术、掌握软件构架搭建、PPT制作以及视频制作能力突出的技术人员，也有组织能力突出的领导人员。

模型训练数据集：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Devoice需要一台服务器，由学校提供。\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 市场可行性分析

在当前市场上，手机端的音频处理软件并不多，并且大多以音频剪辑、合成以及语音识别为主，并没有能够达到音频分离功能的软件，这是市场的一大空白。在PC端的音频处理软件中，Adobe Audition、Audacity等部分软件能达到简单的音频分离，但分离效果较差，会造成音质的损坏，并且此类软件普遍操作复杂，入门困难，使用难度较大。

Divoice作为以音频分离为主的手机端快应用软件，主要有以下几点优势：

1. 以手机快应用的形式，无需下载，使用便捷，用户能够随时随地使用，而不需要求助于电脑等笨重的设备。
2. 以人手一把的手机为载体，注定了潜在用户数量众多，用户群体也不会局限于专业的音频处理人员，普通用户也能够在观看视频、直播时，分离出所需要的音频，获得更好的体验。
3. Divoice的分离方法是基于深度学习技术，用户只需输入原始音频文件，分离功能由软件自动实现，不需要用户自身额外的操作，使用方法简单便捷，入门容易。
4. 系统在运营过程中会不断地收集用户的反馈和评价，进而动态地更新模型，实现语音分离的高效率、高质量，为用户提供更加智能便捷的服务。

# 需求分析

## 数据需求

### 静态数据

### 动态数据

### 数据词典

### 数据采集

## 功能需求

### \*\*功能模块

表 核心功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| **功能模块** | **功能** | **功能描述** | **优先级** |
|  |  |  |  |
|  |  |  |
|  |  |  |

表 \*\*\*\*用例规约

|  |  |
| --- | --- |
| 用例名称 |  |
| 功能简述 |  |
| 用例编号 |  |
| 执行者 |  |
| 前置条件 |  |
| 后置条件 |  |
| 涉众利益 |  |
| 基本路径 |  |
| 扩展路径 |  |
| 字段列表 |  |
| 设计规则 |  |
| 未解决的问题 |  |
| 备注 |  |

## 性能需求

### 时间特性

### 适应性

## 界面需求

## 接口需求

### 硬件接口

### 软件接口

## 其他需求

# 概要设计

## 处理流程

## 总体结构设计

## 功能设计

## 用户界面设计

## 数据结构设计

## 接口设计

### 外部接口

### 内部接口

## 错误/异常处理设计

### 错误/异常输出信息

### 错误/异常处理对策

## 系统配置策略

## 系统部署方案

## 其他相关技术与方案

# 数据库设计

# 详细设计

## \*\*功能模块

### 功能描述

### 性能描述

### 输入

### 输出

### 程序逻辑

### 限制条件