

FIT3161: Computer Science Project 1 Semester 1, 2024

Project Proposal & Literature Review

Project Topic:

Singing Video Generation with Music Separation

Supervised by:

Dr. Arghya Pal

Group Name:

MCS08

Group Member:

Yeoh Ming Wei (32205449)

Toh Xi Heng (33200548)

Yew Yee Perng (32205481)

Words Counts: 9600

Table of Contents

Table of Contents.....	2
1. Introduction.....	4
2. Literature Review.....	5
2.1 Background.....	5
2.1.1 The Challenges of Music Separation.....	5
2.1.2 Limitation of Generating A Realistic Virtual Avatar.....	5
2.2 Rationale.....	6
2.2.1 How Do Both Backgrounds Relate to Each Other?.....	6
2.2.2 Comparison of Deep Learning and Traditional Method for Audio Separation.....	6
2.2.3 How Dynamic Scene Representations Relates Virtual Avatar?.....	7
2.3 Related Work.....	8
2.3.1 Audio Separation.....	8
2.3.1.1 Non-negative Matrix Factorization (NMF).....	8
2.3.1.2 U-Net architecture.....	8
2.3.2 Virtual Avatar Generation.....	9
2.3.2.1 Generative Adversarial Networks (GANs).....	9
2.3.2.2 AniPortrait.....	9
2.4 Conclusion.....	10
3. Project management plan.....	11
3.1 Project Overview.....	11
3.1.1 Major work activities and milestones of the project completed.....	11
3.1.2 Major work activities and milestones of the project yet to completed (to be completed in sem 2 2024).....	12
3.2 Project Scope.....	12
3.2.1 Project Scope.....	12
3.2.2 Product characteristics and requirements.....	12
3.2.3 Product acceptance criteria.....	13
3.3 Project Organisation.....	14
3.3.1. Process Model.....	14
3.3.2. Project Responsibilities.....	15

3.4. Management Process.....	16
3.5 Monitoring and controlling mechanisms.....	17
3.5.1 Communication Plan and task allocation.....	17
3.5.2 Progress Monitoring.....	18
3.5.3 Review and audit mechanisms.....	18
3.6 Schedule and resource requirements.....	19
3.6.1 Schedule.....	19
3.6.2 Resource requirements.....	19
3.6.2.1 Hardware Requirements.....	19
3.6.2.2 Software Requirements.....	20
4. External design.....	23
4.1 User Interface.....	23
4.2 Training Datasets and Training Resources.....	25
4.3 Performance.....	25
5. Proposed methodology.....	26
5.1 Introduction.....	26
5.2 Software and Tools.....	26
5.2.1 Programming Language and Libraries.....	26
5.3 Models.....	27
5.4 Data Processing.....	29
5.4.1 Audio Separation Processing.....	29
5.4.2 Virtual Avatar Generation Processing.....	31
6. Test planning.....	33
6.1 Unit Test.....	33
6.1.1 Audio Separation.....	33
6.1.2 Virtual Avatar Generation.....	33
6.2 Integration Test.....	33
6.3 System Testing.....	34
6.4 User Acceptance Testing.....	34
7. Conclusion.....	35
8. Reference.....	36
9. Appendix.....	38

1. Introduction

1.1 Aim of the project

The project, titled "Singing Video Generation with Music Separation," aims to develop an advanced system capable of generating high-quality human singing face videos synchronised with provided songs. The core objective is to leverage state-of-the-art artificial intelligence and audio processing techniques to decompose input music into separate streams of background music and human voice. This separation facilitates accurate lip synchronisation and enables the creation of realistic head movements and facial expressions that follow the rhythm and emotional content of the music.

1.2 Brief Summary of the Entire Project Plan

Over the next semester, our group intends to systematically develop and integrate the various components required for this project. The plan is divided into several key phases:

1. Development Phase:

- **Music Separation:** Implement a deep learning-based model to decompose input music into vocals and background music. This phase will involve preprocessing the audio data, training the model, and validating its performance.
- **Lip Synchronisation:** Utilise a model like Wav2Lip to generate lip-synced videos. This will require integrating the separated vocal tracks with facial movements, ensuring precise synchronisation.
- **Facial Animation:** Implement a model such as the First Order Motion Model to generate realistic head movements and facial expressions. This phase will focus on transferring motion from a driving video to a target face.

2. Integration and Testing:

- Combine the outputs of the music separation, lip synchronisation, and facial animation models into a cohesive system.
- Perform extensive testing using a variety of songs and videos to evaluate the system's performance and make necessary adjustments.

3. Optimization and Finalization:

- Optimise the model parameters to improve the quality and realism of the generated videos.
- Finalise the system and prepare comprehensive documentation and presentations for project submission.

2. Literature Review

2.1 Background

2.1.1 The Challenges of Music Separation

Music separation is one of the key challenges in music processes. In general, a music file contains broad categories which consists of audio mixture of the voice of a human and the background music which can be also known as the music instrumental. The objective of music separation is to correctly identify the sound events by isolating the source within a complicated mixed audio input. As music separation is popular and widely used by many applications such as the music industry, speech recognition and hearing aids, there are many researches being explored to improve the quality of separated audio or better efficiency. Some of the techniques include the earlier methods such as Fourier transform and Non-negative Matrix Factorisation(NMF) or the computer vision models which are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

However, there is no doubt that a music file cannot be separated perfectly with the presence of a trained system dedicated to music separation. This is known as the cocktail party problem which is a disability to perfectly recognize a single audio source from overlapping signals. The difficulty increases tremendously when a trained system not only needs to identify the audio source, it also needs to extract or isolate the source from a mixed audio input to produce separated audio source files.

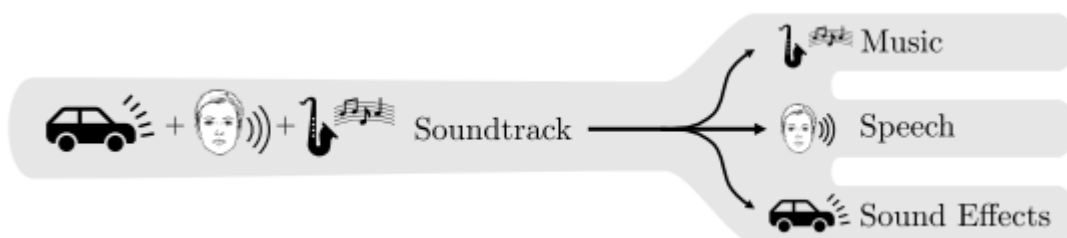


Figure 1: A process of Cocktail Fork Problem

2.1.2 Limitation of Generating A Realistic Virtual Avatar

Virtual avatar is currently demanding for various applications, such as the gaming industry, virtual try-on and many more. The concept of virtual avatars is already known during the early development such that various applications started their own development as a visual representation of a human object. This includes the gaming industry which is widely used as a visual representation of players and Non-Playable Characters (NPCs), or the usage of virtual reality (VR) equipment which allows the user to control the virtual avatar using motion capture and many more.

However there are some limitations that restrict the virtual avatar being captured poorly that causes some factor of the avatar to remain unrealistic compared to a normal human behaviour, such as lip synchronisation, head movement and emotion expression and it remains a challenge to solve real world problems. However, there are various approaches that help to resolve the issue by using Deep Learning methods such as Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN)s, Neural Rendering Fields (NeRF) for generating the dataset. More explanation will be provided at the related work section.

2.2 Rationale

2.2.1 How Do Both Backgrounds Relate to Each Other?

As a recap for the objective of our project, the project aims to generate a singing video which contains elements such as instrumental, vocal and a virtual avatar with movement. The generation can be done by providing input such as a music file for decomposing and an image for generation of virtual avatar. Additionally, we will improvise the system by having a software with better user interface and have a more effective generation.

With that said, separate backgrounds allow us to understand the key challenges of approach and identify different techniques to tackle the problem. Later, we will be able to analyse the best method to approach both backgrounds and combine the method to produce a robust software.

2.2.2 Comparison of Deep Learning and Traditional Method for Audio Separation

There are various approaches or methods that can be done to separate the audio. However, these methods have advantages and disadvantages that affect the result based on various aspects, such as the quality of the audio or the efficiency when separating the audio. Based on the comparison of both deep learning and traditional methods, it is mentioned that deep learning has more advanced techniques that surpassed the traditional method (Rincón-Trujillo & Córdova-Esparza, 2019).

In terms of flexibility and adaptability, deep learning methods introduced new sources that show a better improvement compared to traditional methods. A few examples are acoustic-only source and audiovisual source that are being proposed by many researchers and accomplished many results (Rincón-Trujillo & Córdova-Esparza, 2019).

Besides that as a continuation from the explanation above, having more sources provides advantages in stability of separation which also increases the quality of the audio, which

results in a higher accuracy compared to traditional methods. From a proposal (Gao, Du, Dai, & Lee, 2017), their system managed to provide a faster performance by using a framework based on joint learning that allows managing both background noise and speech.

However, there are some drawbacks such as audiovisual sources that require a great amount of training data in order to achieve a commendable result. It can be seen on one of the research which requires 60 speakers with around 200 videos each in TCD-TIMIT Dataset (Gabbay, Ephrat, Halperin, & Peleg, 2018). As a result, it will increase the computational resources as well. Researchers (Liu & Wang, 2018) proposed a solution that produces better results which is an advantage, however their algorithm didn't manage to perform in real-time computation.

2.2.3 How Dynamic Scene Representations Relates Virtual Avatar?

Dynamic Scene Representations has been the popular research in computer vision. It allows capturing the motion of objects over time by ensuring all three properties which are permanency, amodal completeness and spatiotemporal continuity. To further explain the properties, permanency ensures that the object still exists after being blocked from the view. Amodal completeness ensures that the object is available as a 3D model even though the view of the model is partially seen. Lastly, spatiotemporal continuity will track each of the objects over space and time simultaneously. It relates to a virtual avatar as dynamic scene representation is allowed to provide changing over time for a given avatar face by changing the surface of a human's face using transformation.

One of the research (Gafni, Thies, Zollhofer, & Niesner, 2021) shows the usage of dynamic scene representation through input frames. By allowing tracking of facial movement, a 3D morphable model will enable reconstruction of facial expressions. Through input frames, it will provide estimation of the movement and provide a list of transformations. Lastly, volumetric rendering will be used to synthesise the image of the face.

The issue of dynamic scene representation that involves multiple moving objects or the occlusion of the object from different viewing angles. One of the approaches that is proposed is using Neural Rendering Fields (NeRF) however it is not a wise approach as it needs a vast amount of queries to render an image.

For the next section, we will discuss various approaches or techniques available from the researchers to tackle the relevant problems that are mentioned in the background section. Later, we will review the technique available to ensure that the approach chosen is better through comparison and apply it for the design of the system.

2.3 Related Work

2.3.1 Audio Separation

2.3.1.1 Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization algorithm was firstly introduced by (Lee & Seung, 2000) which proved to be a useful decomposition of multivariate data. The algorithm was later adapted for audio separation. One of the methodologies from Smaragdis and Brown (2003) shows the usage of Non-negative Matrix Factorization for Polyphonic Music Transcription which aims to recognize the musical instrument and pitch from music signals generated by multiple instruments, and later generate it into symbolic representation (Smaragdis & Brown, 2003).

In detail, the principle used nonnegative decomposition of the spectrogram of the music signals onto a dictionary of components that consists of different sound units, such as music note, chord, percussion and many more. In addition, the architecture obtains the factorization and optimization of a cost function by using NMF models. Later, the process continues by reconstructing the components based on the nonnegative factorization (Makino, 2019).

As a result of the research above, it is mentioned that the process was successful without extensive computational resources and complicated system design. Hence, NMF is still being vastly researched in a situation where the data needed is small which is a better solution compared to Deep Neural Based techniques that require more data for training (Makino, 2019). However, this may be only useful for simple types of application and more data may be reliable for a better quality of audio separation.

2.3.1.2 U-Net architecture

A U-Net architecture is a pre-trained model or an encoder/decoder Convolutional Neural Network (CNN) architecture with skip connections (Hennequin, Khlif, Voituret, & Moussallam, 2020). This architecture allows stem or track separation from mixed audio source with the implementation of 1D and 2D convolution (Henning, Choudhry, & Ma, 2021). One of the deep learning tools called Spleeter which is a new tool for audio separation with a pre-trained model which is U-Net architecture that contains vocal separation, 4 stems separation and 5 stems separation. Splitter used a total of 12 layers for U-net, 6 layers for encoding and 6 layers for decoder.

It works by providing vocal and non-vocal components into the neural network system, comparing output inside the neural system, and performing adjustment to reduce comparison, allowing the system to estimate both of the components. The system aims to improve the estimation of identifying vocal and non-vocal components after the training of a neural network system. (Singing voice separation with deep U-net convolutional networks, 2024)

The usage of Spleeter for music separation has proven that the output speed shows better efficiency due to implementation in Tensorflow, which allows code execution on the Central Processing Unit (CPU) and Graphic Processing Unit (GPU). As a result, the computation will be parallelized which reduces the running time of processing. Additionally, the model is based on CNN. An experiment shows that Spleeter can process 100 seconds of stereo audio in less than a second. As a better efficiency of processing, Spleeter can be useful for processing large datasets.

2.3.2 Virtual Avatar Generation

2.3.2.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) is a deep learning based generation that assists in generation of realistic images with enhancement through training of two different neural networks modelled against each other. It is called a Generator that generates samples through input data. Another neural network which is called a Discriminator, that receives the samples from Generator. These two trained each other so that the Discriminator is able to identify data that are realistic or artificial. The final generated data can be achieved when it cannot differentiate between real data.

There are also improved GANs that are related to virtual avatar generation such as the StyleGANs. StyleGAN is an extension of the vanilla GANs that has an ability to generate controllable images of the human face over different aspects such as the facial features, hairstyles and accessories. There are alternative GANs methods such as CycleGAN which are appropriate for Image-to-Image translation.

GANs generation provides many advantages such as the high quality realistic image generation by using StyleGAN. However, there will be drawbacks which require large data datasets and high computational resources. The time spent for model training using CycleGAN is approximately 520 hours (García, 2023).

2.3.2.2 AniPortrait

Aniportrait (Wei, Yang, & Wang, 2024) is a framework to generate animation portraits by providing an input audio and reference image that resembles a human face avatar. This proposal aims to breakthrough the commonly used generation such as Generative Adversarial Networks (GANs) or Neural Radiance Field (NeRF).

AniPortrait requires 2 stages to generate the animation portrait. The first stage involves extraction of 3D facial mesh and head pose from the audio input by using a transformer-based model. Following up, these two elements will be transformed into 2D keypoints. The second stage transforms the keypoint into a portrait video using a diffusion model. There are two modules that will represent each of the stages used for the framework which is Audio2Lmk that extracts sequence of landmarks through facial expression and lip movement, Lmk2Video that helps to generate portrait videos through the landmarks obtained.

As a result, it can be seen that the video generated features smooth lip motion and natural head movements which shows human-like movement. However, the generated video contains an uncanny valley effect which the avatar generated is not fully natural compared to a real human movement. Additionally, it requires intermediate 3D representations, and the cost of high computation also shows the drawback of the generation.

2.4 Conclusion

After analysing the different techniques or research tools for both audio separation and virtual avatar generation, we identified the advantages and disadvantages for each of the techniques that are mentioned above. Despite having many techniques available for our project, we had come to the conclusion that U-Net architecture and AniPortrait is the most suitable for our project.

The usage of U-Net architecture for a framework such as Spleeter has provided benefits in terms of efficiency. The usage of TensorFlow allows the framework to process the code through parallelization by using both CPU and GPU. This will reduce the speed of performing audio separation and allows us to have more time for code testing.

As for generation of virtual avatar, we think that AniPortrait provides a solution that shows the most relation for our project. Although GANs method does not have much difference compared to Anipotrait, Anipotrait actually shows more resemblance to our project. Our initial plan was having inputs of a music file and an image to process the video which shows high similarity compared to AniPortrait.

Therefore through critical analysis, our project will be planned to use U-Net architecture first for audio separation. The generated music files will be transferred to AniPortrait as inputs to generate the portrait video based on its method.

3. Project management plan

3.1 Project Overview

This project aims to push the boundaries of DeepFake with the aid of generative AI on developing an application that generates singing videos with accurate lip synchronisation. This project will mainly focus on the decomposition of the input music into human voice and background music components then ensure the accuracy on lip synchronisation between the human singing face and the vocals by testing the system using a variety of songs to evaluate its performance.

3.1.1 Major work activities and milestones of the project completed

Analyzation of the project

Our team did thorough research and analysis on the project to develop a further understanding of the background knowledge regarding this project. Our team also did meetings with our supervisor to understand the reason behind the project and the direction of the research we would be expected to do.

Models and libraries related to the project

Some models we shall be using include the U-Net architecture which serves as the deep learning model for our project and libraries such as Annipotrait that would be used throughout the project.

Initial design and concept

Our team was able to create a design and concept which allows us to produce an insight for our application.

Literature review of the topic

Our team performed a literature review on the topic to be updated with the existing knowledge known in the field. Besides that, this ensures that we have the basic knowledge of the topic so that we are able to develop an understanding of the project.

Project proposal

Our team was able to articulate the project and its objective by performing a project proposal.

3.1.2 Major work activities and milestones of the project yet to completed (to be completed in sem 2 2024)

- Training and testing the model to be used for our project to achieve optimal accuracy
- Explore and improve on audio generators for decompositions of high quality audio
- Develop a working application for our project with the given libraries and models

3.2 Project Scope

3.2.1 Project Scope

The scope of the project is to have the team create an application that is able to generate high accuracy and quality singing videos with an image or virtual avatar with lip synchronisation in it. The application will be able to run on a user's computer without much issues. The team will not create a model from scratch and will instead train existing models from an established source. The team will also create the user interface for the software. After development of the software, the team tests the performance and enhances it based on the performance to ensure that it reaches an acceptable level of accuracy. All details during the development process shall be documented and reported to relevant stakeholders.

3.2.2 Product characteristics and requirements

Below is the identification of the in-scope and out-of-scope characteristics and requirements for the product, with the relevant assumptions, limitations and constraints. This is to ensure that the team has a clear vision of the implementation of the final product.

In scope(high priority):

1. Accepts a music audio input of the user's desired music audio. This file must be in mp4, flac and wav.
2. Able to decompose the music audio into background music streams and human voice.
3. Able to produce a high quality singing video generation that produces lifelike singing face videos synchronised with the provided song.
4. Able to have realistic facial movements in the video's generation driven by music signals.

In scope(low priority):

1. Simple user interface design.
2. The system has high performance and is able to compute the singing video quickly, accurately and naturally.

Out of scope:

1. Super resolution of a low resolution video.
2. Full body animation of the virtual avatar created.
3. Real-time processing optimised for quality.
4. Extensive language support for the produced singing video.

Assumptions

1. Not building a model from scratch. The model is trained on a pre-existing model from several reliable sources.
2. Input video resolution is at least 720p.
3. A fixed set of predefined basic facial expressions and head movements that will be sufficient to convey the necessary emotions and realism.
4. It is assumed that the project will comply with all relevant copyright laws and ethical guidelines, particularly concerning the misuse of generated videos and the use of music and facial data.

Limitations/constraints

- 1) Hardware to train models.
- 2) Python language is used.
- 3) Spleeter library and Anipotrait model used to develop and train the model.

Functional and Non-Functional Requirements

The product requirements are derived from the project requirements from the project supervisor, which is Dr Arghya Pal . The requirement traceability matrix can be found in the appendix.

3.2.3 Product acceptance criteria

User acceptance criteria are requirements that our product must meet in order for it to be accepted by the client. Below are several user acceptance criteria associated with its corresponding user story that have been identified and received from the project supervisor and can be found in the appendix.

User stories

- 1) As a developer, I would like to decompose the music to vocal and background music so that I can synchronise the facial expressions and lip movements of the singing face video with the vocals extracted from the input music.
- 2) As a developer, I would like to catch the lip movement so that the avatar's lip singing the vocal can sync with it.

- 3) As a developer, I would like to catch the facial expression so that I can ensure that the facial expressions align with the emotional content and dynamics of the input music.
- 4) As a user, I wish the decomposition of music is precise so that the sound of the resulting video is not garbled.
- 5) As a user, I wish I can upload videos and images and generate videos easily so that I do not need to follow a strict instruction.

For the user stories mentioned above, below is the user acceptance criteria respectively based on the number.

User acceptance criteria

- 1) The application is able to decompose the music audio to vocal and background music so that the developers can synchronise the facial expressions and lip movements of the singing face video with the vocals extracted from the input music.
- 2) The application is able to catch the lip movement so that the avatar's lip can sync with the vocals.
- 3) The application is able to catch facial expressions so that the developers can ensure that the facial expressions align with the emotional content and the dynamics of the input music.
- 4) The application is able to decompose the music precisely so that the sound of the resulting video is not unclear.
- 5) The application allows the users to upload videos and images as well as generate videos easily without any strict instructions.

3.3 Project Organisation

The team consists of 3 members. The table below shows the team member's details, roles and responsibilities. The team is supervised by Dr Arghya Pal and his PHD student Ai Fang.

3.3.1. Process Model

For our project, we decided to employ the Agile process model as it provides a huge advantage due to its inherent flexibility, emphasis on collaboration and the ability to integrate continuous feedback. Besides that, Agile's iterative and incremental approach aligns seamlessly with the project's requirements for progressive development and frequent refinements. Given our current stage, we had been learning and grasping the knowledge regarding the fields related to the project. With all the tasks such as audio decomposition and lip synchronisation, Agile allows our team to work in manageable sprints, delivering functional components of the in manageable sprints and giving us a chance to work as we learn.

Additionally, Agile promotes strong communication and collaboration among cross-functional teams, which is vital for integrating diverse expertise in machine learning, audio processing, and computer vision. Daily standups and sprint retrospectives facilitate ongoing dialogue and problem-solving, enhancing team cohesion and efficiency. By structuring the project into fixed-length sprints with clear goals and deliverables, Agile provides a disciplined yet flexible framework that helps manage the project's complexity and ensures continuous progress. This structured approach, combined with Agile's principles of iterative improvement and stakeholder collaboration, makes it an ideal process model for delivering a high-quality, user-centric system for singing video generation with music separation.

Thus, our team shall use an Agile process model as it is beneficial and advantageous for the current project.

3.3.2. Project Responsibilities

To ensure that everyone in the team has a clear understanding of the goals and objectives of the project, tasks related to roles are recorded on a table for future reference.

Name	Role	Responsibility
Yeoh Ming Wei	Project Manager	Monitor progress and set deadlines, ensures the satisfaction of stakeholders, evaluate project performance and takes lead in discussions and meetings
Yee Perng Yew	Quality Assurance	Ensures the quality and performance of deliverables
Toh Xi Heng	Technical Lead	Guides team members in technical matters and supervising system modifications
Dr Arghya Pal	Supervisor	Provides guidance and advice to team members on the project

Although every team member has their own roles and responsibilities, the team will still perform some of the activities together regardless of the roles. For example, the team will gather information and resources on different models or libraries from reliable resources, then validate it with our supervisor before attempting to implement it and improve it on the given source code for the project. Besides that, documentation of our work, progress and the UI implementation will be done together as a team too as planned.

3.4. Management Process

3.4.1 Risk Management

According to Schwartz in 2021, Project risk management is the process of identifying, examining and responding to the risk that arises over the life cycle of a project to help the project remain on track and achieve the goal. In order to minimise the risk while taking advantage of the possible advantages, the team decided to make use of SWOT analysis to identify the potential risks. Our team held meetings to voice out the potential risks involved in this project. SWOT analysis is a strategic planning technique that provides assessment tools to identify the core strengths, weaknesses, opportunities and threats that lead to fact-based analysis, fresh perspectives and new ideas. Strengths and weaknesses are internal factors brought to by the team, while opportunities and threats are external factors that could affect the project positively or negatively. The table below displays our SWOT analysis as our risk identification technique for this project.

Strength: <ul style="list-style-type: none">- Guidance and assistance from our supervisor and his PHD student- Source code to be given by our supervisor's PHD student	Weaknesses: <ul style="list-style-type: none">- Shallow understanding on the models to compute the singing video- Unfamiliar to work with the libraries used for AI
Opportunities: <ul style="list-style-type: none">- The application will support audio from multiple languages hence increasing user experience.	Threat: <ul style="list-style-type: none">- Ethical or legal issues concerns might be a potential issue that could arise in the future

Furthermore, a risk register is used to identify potential risks within our project throughout our whole project duration. By doing so, we would have a plan to counter any potential risks that arise during the project and mitigate it safely without causing many issues. The risk register table would be placed in the Appendix.

3.4.2 Stakeholder analysis and communication plan

A stakeholder analysis is a process used to identify, assess, and prioritise all the individuals, groups, and organisations that can affect or be affected by a project. It involves understanding the stakeholders' interests, influence, and potential impact on the project to ensure their needs and expectations are appropriately managed. This analysis is crucial for successful project management as it helps in building strong relationships, mitigating risks, and ensuring stakeholder support and engagement throughout the project lifecycle.

3.4.2.1 Stakeholders involved in this project

Project Supervisor - In charge of overseeing the project, ensures the project is on the right track, guides and assist the project and provides feedback on the progress and performance to the team

Project Team - Ensures the project is on track and fulfilling the requirements and needs of the clients.

End-Users - Provides feedback on the user experience such as on the functionality and usability in the system.

Teaching Team - Marks and grades the documents that are used for this project and assesses the presentations and showcases of the project.

3.5 Monitoring and controlling mechanisms

3.5.1 Communication Plan and task allocation

An effective communication plan is vital towards the efficiency of the project. To achieve an effective communication plan, there are some key steps that we would follow. The identification of the stakeholders and determination of communication needs greatly reduces the doubts of each meeting held during our initialisation stage. By assigning responsibilities, it also helped in the identification of tasks for specific roles to take note of during the meeting. Our team follows a schedule for our meetings and fixed communication methods for our regular meetings and information sharing.

When we first started to plan for our project, we had meetings regularly to discuss and explore the project topic by gaining more information thus increasing our knowledge in that field. Our project supervisor also held regular meetings to ensure that we understood the project and the initiative of the project so that the team does not get in the wrong direction, ensuring the actual requirements and thoughts of the clients. Google chat was our main communication platform while communicating with our supervisor, whereas Discord and Whatsapp were our internal group communication for meetings and sharing information.

Initially, the team did work together on all given tasks during the first assignment. However, it was soon to be noticed that as time went by, it got inefficient as everyone had to understand every part of the task. Besides that, every team member had their own working time therefore finding a common time to do the assignment was not an easy task. Hence, we decided to split up our work according to the expertise and experience of the team members, and progress was reported regularly through meetings and our communication platform. When all the tasks

had been completed, we would then conduct a final meeting to finalise the project, where team members explained the part they did to the other team members to ensure that everyone was on track and the work was satisfactory for all of the team members.

3.5.2 Progress Monitoring

The main method of monitoring our progress for the project is to have weekly meetings. During our initial starting phase, we had meetings to assign roles, responsibilities and tasks for our project. We also held progress meetings and information sharing sessions to ensure everyone is on track and up to date. Besides that, we would also have meetings with our supervisor on a weekly basis. We would update our supervisor on our findings throughout the week and he would share some sources or ask us to read up some information to keep us updated. Besides that, our team also checks our progress with our Gantt Chart, which will be added to the appendix. This ensures that our team is on track with the initial plan.

3.5.3 Review and audit mechanisms

Review Mechanisms and Audit Mechanisms are processes used to ensure that a project is performing as expected, adhering to standards, and achieving its goals. This includes version control, quality assurance and documentation and training.

Version control

For version control, we will be using GitHub as our code repository. By doing so, it allows us to code on separate branches and do work without having the need of being together. Besides that, we will be able to track changes in our code across different versions and acknowledge the contributions made for every team member.

Quality Assurance

To ensure the quality of the project is up to standard, code reviews are done regularly during the meetings. By doing so, we are able to have easily understandable code and work with comments or descriptions to explain the particular work. In addition, we can also check if the code fulfils the requirements fully and test the code to ensure that it is error-free.

Documentation

Documentations are noted down for all activities performed in the project, acting as a memo for future references. These documentations would usually be noted down by one member of the team and then sent to our communication channel so that everyone has a copy of it. After the documentation has been reviewed by each member, the document is finalised and kept by every member of the team locally.

Training

As the team had no prior knowledge with DeepFake and generative AI, all members of the team will gain knowledge in these fields by performing thorough research and readings to enhance their knowledge. Meetings with our project supervisor also helps to a certain extent as he provides us with some background information and some resources for us to study.

3.6 Schedule and resource requirements

3.6.1 Schedule

A schedule is a list of activities that are planned or required to be accomplished within the existing time. By following a schedule, the project can be completed within the planned time, progress could be done according to the time and ensures the team is on track.

For our team, our project schedule dates start on 4 March 2024 until 15 November 2024. The duration of the project would roughly be 9 months as planned.

As for the project tasks, our team decided to break down the project into phases in Work Breakdown Structure (WBS) and our project timeline in a Gantt Chart. The WBS and Gantt charts can be found in the appendix below.

3.6.2 Resource requirements

Below are the hardware and software requirements for this project. These requirements represent a minimum requirement after research by the team.

3.6.2.1 Hardware Requirements

Hardware	Justification
Student's Laptop	Although our laptop cannot handle heavy tasks like training large-scale deep learning models, we still can use it to do prototyping and do testing for code.

Graphics Processing Unit	GPUs are used to train the deep learning models. Although we are using the GPUs via Google Colab, we can still locally run our code or train models using our own GPUs. If we have enough budget to use, we can buy a NVIDIA GeForce RTX 4090 for local use.
--------------------------	--

3.6.2.2 Software Requirements

Software	Justification
Operating System Windows 10 or later	Our team will use the Windows 10 or better version as our operating system(OS) which is the only type of OS our team has. The other reason why we use this OS is because of its popularity with users. Globally, about 71% of users use Windows as their main and also with company. The documentation[1] has given some reasons for using Windows compared to Linux or MacOS such as user-friendly interface and extensive software compatibility. Therefore, designing the project using Windows makes it easier to ensure that users' devices are compatible with our project.
Programming Language Python	We are going to use Python as our programming language in this project. First of all, it has a vast ecosystem of libraries or deep learning frameworks and will explain further in software libraries. Besides that, Python has been used by our team members in many units in our computer science study career. Furthermore, Python has a large number of developers and researchers. This helps us find out some useful tutorials or pre-trained models to improve the quality and accelerate the speed of designing projects.
Software Libraries PyTorch, Spleter	PyTorch is used for irregular input data such as graphs, point clouds and manifolds. PyTorch seamlessly integrates with Python that allows us to use libraries and tools in Python. This streamlines the development process and facilitates interoperability with other technologies used in AI which is our project needed for AI models. Also, PyKale is a library in PyTorch for multimodal learning and transfer learning with deep learning and

	dimensionality reduction on images and videos. There are few more libraries[2] that can be used and PyTorch is just the main used in our project. Also, Splitter can be used to separate human voice and background music.
Programming Language Environment VSCode	VSCode is now commonly used by our team members for coding. For Python, we can also use PyCharm but we decided to use VSCode. As known, VSCode has lighter weight and faster performance compared to PyCharm. Besides that, VSCode is easier to connect with GitHub which has also been used in this project to store and backup the whole project. One extension that is useful is we can do live coding by using VSCode which makes our team easy to discuss. Furthermore, VS Code provides a Data Viewer that allows us to explore the variables within the code and notebooks, including PyTorch. [3]
Frontend Programming Language HTML & CSS	These two languages are the standard language used to create a website application. HTML provides web page structure whereas CSS is used to control web page styling. [4]
Backend Programming Language Python	Because it is mainly using PyTorch, so the backend programming language will use Python.
Cloud Platform Service Google Colab	Google Colab offers free access to GPU(Graphics Processing Unit) and TPU (Tensor Processing Unit) resources, along with pre-installed libraries and frameworks like TensorFlow and PyTorch. We can directly write code in notebooks in its web browser and help us to avoid needing powerful local hardware. Because the free version has some limitations such as session time limits and restrictions on resource availability, we are suggested by the supervisor to have a pro version of Colab. We would need to pay for RM47.16 to get more units, memory and faster GPUs.
Cloud Storage	Google Colab has integration with Google Drive which

Google Drive	allows us to save and share our Google Colab notebooks easily.[5]
Version Control GitHub	We use GitHub to control our version as it is easier to clone its repository with VSCode. Besides, we can read the history commit from GitHub which means that we have a backup to our project.
Project Management Tool Jira	We will use Jira as our project management tool. We chose Jira because the management process we learn is Agile and Jira supports all needed for the Agile process like kanban board, scrum and creating sprint[6]. While the free version has some limitations such as the limit of kanban boards created for each sprint, it still can be used nicely for the process.

4. External design

4.1 User Interface

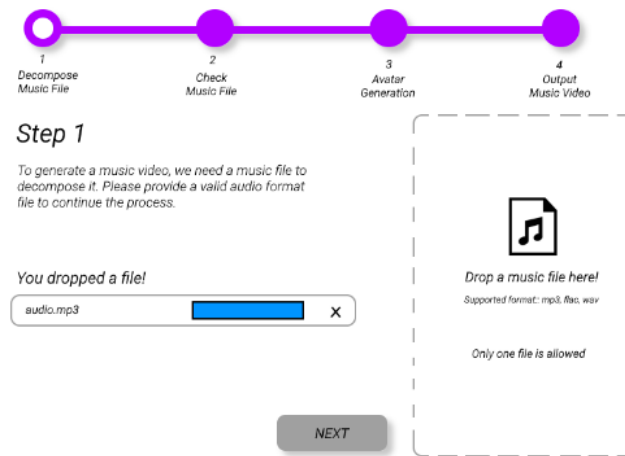


Figure 2. Import Video

The application supports users to upload a music file. There is only a music file accepted for each generation. It commonly supports music files such as MP3, FLAC, and WAV. Users are able to delete or change the music uploaded if users need to do any changes.

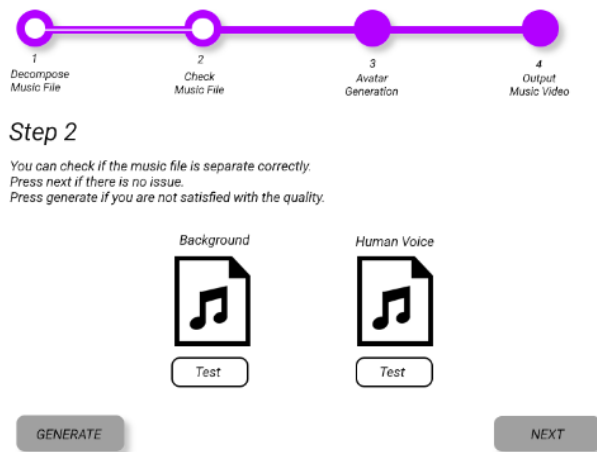


Figure 3. Music Separation Testing

In this step, the music uploaded will be generated into background music and vocals. There are two clickable buttons for users to test the decomposition audio after the default test in the model created. If the user is not satisfied with the result, the user can click on the GENERATE button to regenerate the decomposition audio until it gets the quality that user is satisfied.

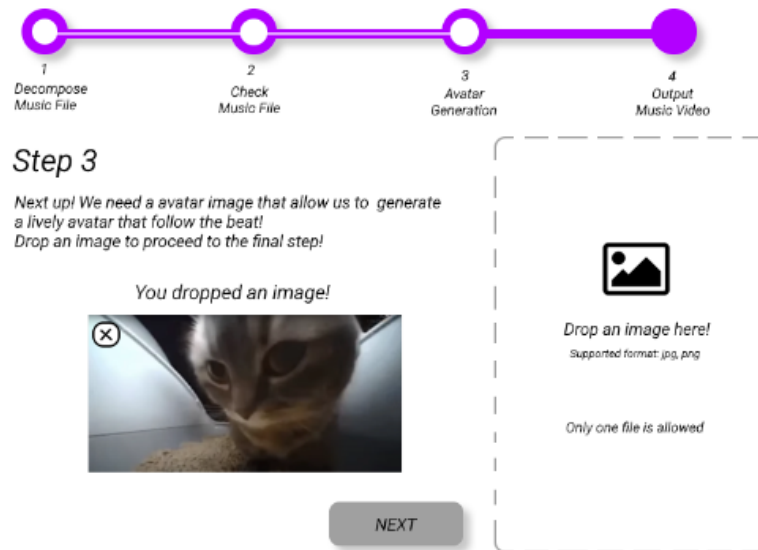


Figure 4. Avatar Generation

For the next step, it is going to generate the avatar movement. Users will be asked to upload only an image for generation. The image is only allowed in JPG, JPEG and PNG files. To do lip synchronisation and face animation, it will ask to provide a clear facial features image, so there will be an automatic check for users to reupload the image if the image is not up to par.

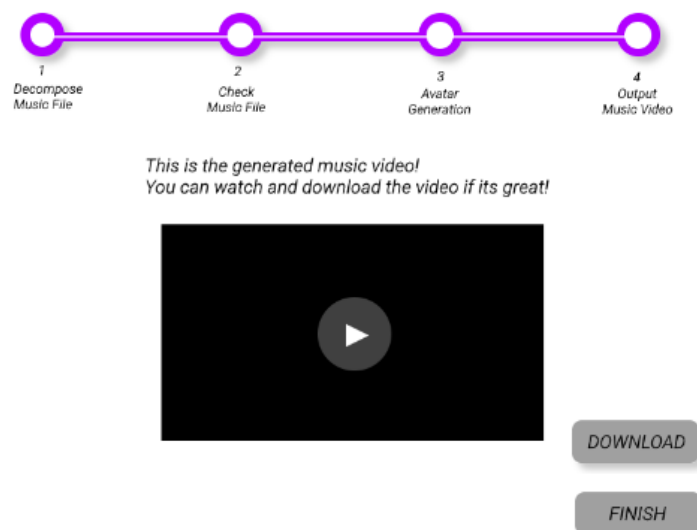


Figure 5 Result of Video Generation

At the last panel, it will have a screen to show the preview of the result of the video generated if the video is successfully generated. Then, users can download the video or click FINISH to do the next generation.

4.2 Training Datasets and Training Resources

VoxCeleb dataset and LRS3-TED dataset These two dataset can be used to train lip synchronisation and facial animation models while the second one is more focusing on the speaking and singing part. Also, the MUSDB18 dataset can be used to train a music decomposition model.

For training the models, a significant amount of computational power is required. Then, we were advised and decided to use Google Colab as our training resource. It contains a free cloud service with GPU support, suitable for initial experiments and smaller models that is enough for our project.

4.3 Performance

Spleeter's performance characteristics are impressive, with real-time processing times ranging from 1-10 seconds and batch processing capabilities for multiple audio files. Its memory usage is relatively low, typically between 1GB to 4GB, depending on the input audio file size and desired precision. Additionally, Spleeter can take advantage of GPU acceleration to reduce processing time and memory usage. With its flexible architecture, Spleeter can be adapted to different music genres and styles, making it suitable for a wide range of applications. Overall, Spleeter's performance characteristics make it an efficient and effective tool for music source separation.

AniPortrait's performance characteristics are impressive, with real-time animation generation capabilities, rendering high-quality animations in 10-30 seconds, and frame rates up to 60 FPS. The system's memory usage is relatively low, ranging from 100MB to 500MB, with moderate texture memory requirements. AniPortrait also takes advantage of multithreading and GPU acceleration, making it suitable for use in real-time applications such as video games, virtual reality, and augmented reality. With its optimised performance and efficiency, AniPortrait is well-suited for demanding applications where high-quality facial animations are required.

5. Proposed methodology

5.1 Introduction

Our model is a fusion model that incorporates two distinct models that are trained for different models. One each for audio separation and virtual avatar generation. We will then take a closer look at each model's architecture, the technical aspects of running these models such as preprocessing and the tools or software required and how they work together to achieve the aim for the project.

5.2 Software and Tools

5.2.1 Programming Language and Libraries

We required a programming language that is used widely by team members and easy to understand. Therefore, we decided to use Python due to the reason above as well as we are using libraries Spleeter and PyTorch. For visualisation tools, we decided to use VSCode because it can be easily cloned with the GitHub which we will use to do version control. Jupyter notebook may be used also due to we are going to use GPU via google colab.

5.2.2 Database

While we do not explicitly need a database, we only used it to store our models and datasets for training and testing. So, we will directly store those things in our own local devices or stored inside the Google Drive.

5.3 Models

As what we mentioned before, there are three models that will be used for this project which are Demucs and AniPortrait.

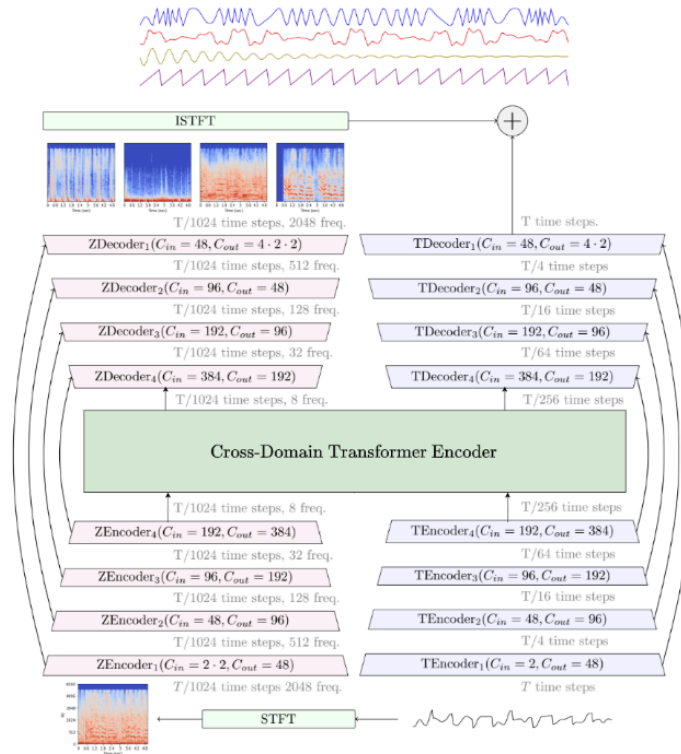


Figure 6: How Demucs works

For audio separation, we will be using Demucs . Demucs is a state-of-the-art music source separation model that can effectively separate drums, bass, and vocals from the rest of the accompaniment. The model uses a Hybrid Transformer Demucs architecture, which combines spectrogram and waveform separation techniques using Transformers. This results in a high-quality separation of audio sources, achieving a SDR of 9.00 dB on the MUSDB HQ test set. The model is available in various pre-trained versions, including htdemucs, htdemucs_ft, and htdemucs_6s, which offer different levels of fine-tuning and additional features such as 6-source separation.

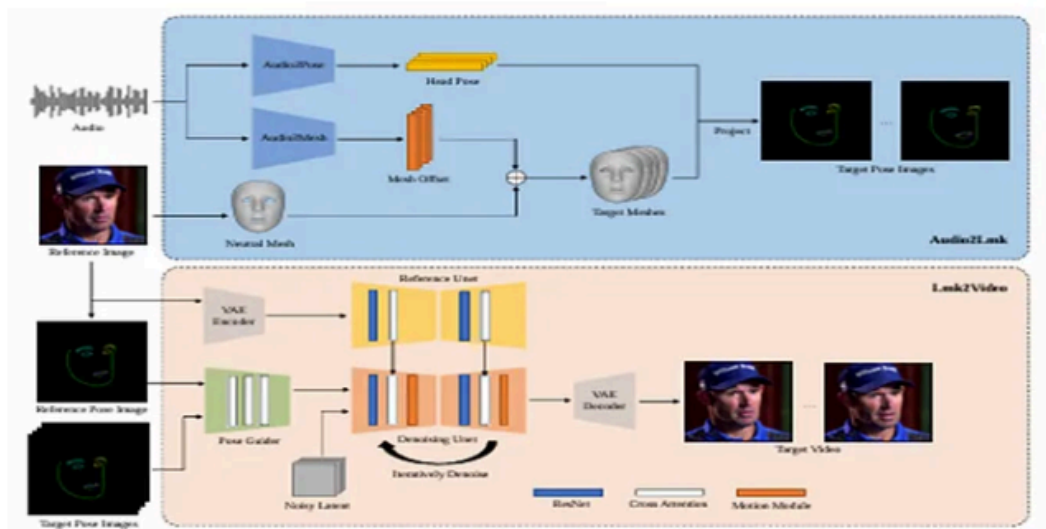


Figure 7 : How AniPortrait works

AniPortrait is particularly effective in lip synchronisation because it can accurately capture subtle expressions and lip movements from audio input. The framework uses a pre-trained audio processing model to extract a sequence of facial landmarks and head poses, which allows for precise tracking of lip movements and facial expressions. This means that the generated animation can accurately match the spoken words, creating a more natural and realistic experience. Traditional animation methods often require manual keyframe animation, which can be time-consuming and labour-intensive. AniPortrait's ability to automatically generate lip synchronisation from audio input reduces the need for manual animation, making it a more efficient and cost-effective solution.

AniPortrait is also useful for virtual avatar generation because it can create realistic and photorealistic animated portraits. The framework's ability to transform a sequence of facial landmarks and head poses into a temporally consistent and realistic animated portrait enables the creation of highly realistic virtual avatars. Virtual avatars can be used in various applications, such as virtual reality (VR) and augmented reality (AR) experiences, video games, and even social media platforms. By using AniPortrait to generate realistic virtual avatars, developers can create more immersive and engaging experiences for users. In addition, AniPortrait's ability to manipulate intermediate 3D representations allows for flexibility in creating different virtual avatars with varying facial expressions, head poses, and body language. This makes it an ideal solution for applications that require customised virtual avatars, such as virtual influencers or personalised characters in video games.

5.4 Data Processing

The flow chart at the appendix shows how the data process from starting processing towards the result needed. Below's flow chart will separate the flow chart as audio processing and virtual avatar generation processing.

5.4.1 Audio Separation Processing

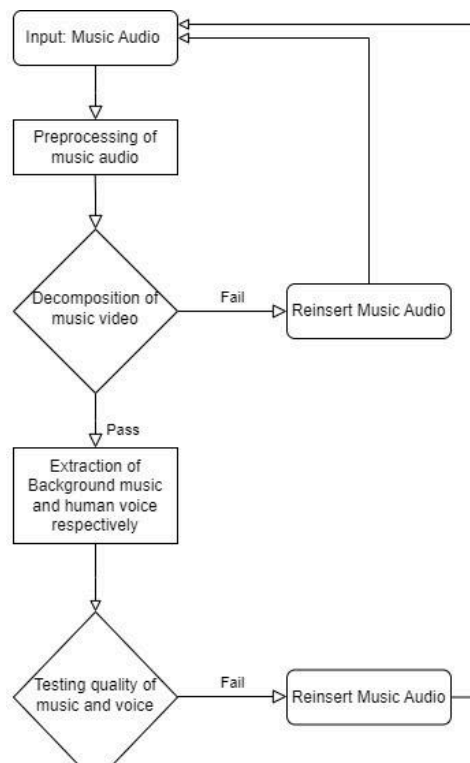


Figure 8: Flow chart for audio separation process

```

1 import spleeter
2
3 # Load the Input Audio
4 audio_file = "path/to/input_audio.wav"
5 audio, sr = spleeter.load(audio_file)
6
7 # Create a Separation Model
8 separator = spleeter.Demucs()
9
10 # Set Model Parameters
11 separator.set_params(iterations=100, min_magnitude=0.1, std=0.5)
12
13 # Separate the Audio
14 bgm, vocal = separator.separate(audio)
15
16 # Post-processing
17 bgm = filters.apply_filter(bgm, "low_pass")
18 vocal = filters.apply_filter(vocal, "high_pass")
19
20 # Save the Separated Audio
21 bgm.write("path/to/bgm.wav", sr)
22 vocal.write("path/to/vocal.wav", sr)

```

Figure 9 : Way to use Demucs model by Spleeter

The music audio is preprocessed and decomposed into background music and human voice by using Spleeter. To separate an input audio file into its background music (BGM) and vocal components using Spleeter, start by loading the audio file into Spleeter using the `spleeter.load()` function, which takes the path to the audio file as an argument and returns a tuple containing the audio data and the sampling rate. Next, create a separation model, such as the Demucs model, which is a deep learning-based model specifically designed for music separation tasks, and adjust its parameters to fine-tune the separation process, including the number of iterations, minimum magnitude threshold, and standard deviation. Once the model is set up, use the `separate()` function to separate the input audio into its individual components, namely BGM and vocal, and then apply post-processing techniques to improve the quality of the separated audio streams by filtering out noise or artifacts, normalising the levels, and equalising the tone and balance. Finally, save each separated audio stream as a separate file using the `write()` function, allowing you to easily work with the BGM and vocal tracks separately. By following these steps, you can successfully separate an input audio file into its individual components using Spleeter, enabling a wide range of applications, including music remixing, music analysis, and speech enhancement. If the decomposition fails, the user will be redirected to the input phase and will be required to reinsert a valid music audio to be preprocessed. The decomposed background music and human voice will then be tested by allowing the user to listen to the extracted music and voice for checking purposes. Similarly, if the user dislikes the decomposed music and voice, the user will be able to reinsert another desired music audio for decomposition purposes.

5.4.2 Virtual Avatar Generation Processing

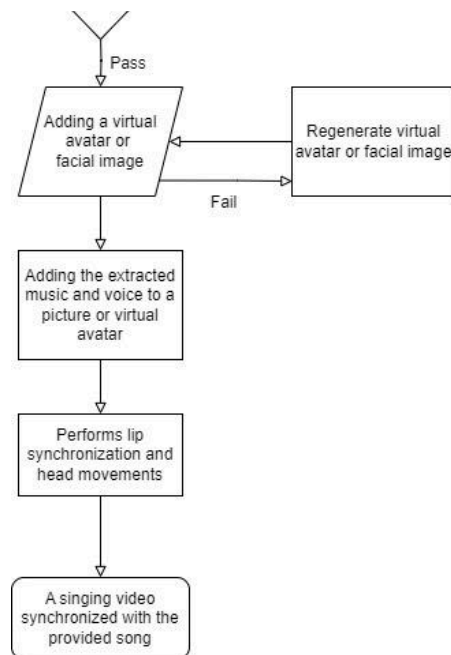


Figure 10: Flow chart for video generation

```
import cv2
import mediapipe as mp

# Load the input image
img = cv2.imread('input_image.jpg')

# Load the AniPortrait model
model = mp.solutions.AnipPortrait()

# Run the model on the input image
results = model.process(img)

# Get the generated avatar
avatar = results.get('avatar')

# Create a video writer
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output_video.mp4', fourcc, 30.0, (640, 480))

# Add animations to the avatar
for i in range(30):
    # Update the avatar's facial expression
    avatar.set_expression(cv2.FACE_EXPRESSION_HAPPY)
    # Update the avatar's hairstyle
    avatar.set_hairstyle(cv2.FACE_HAIRSTYLE_SHORT)
    # Add animation frames
    out.write(cv2.cvtColor(avatar.get_frame(), cv2.COLOR_RGB2BGR))

# Release resources
out.release()
```

Figure 11: Way to use AniPortrait

To generate a video using the AniPortrait, the process begins with image processing, where the input image is resized, normalized, and fed into a deep neural network trained on a vast dataset of faces. The network then detects the face within the image, aligns it to a standard pose, and analyzes the facial expression to determine the person's emotional state. With this

information, AniPortrait generates a 3D avatar of the person, which can be customized with various facial expressions, hairstyles, and clothing options. The avatar is then animated using a combination of keyframe animation, physics-based simulation, and machine learning algorithms to create realistic movements and actions. Finally, the animated avatar is rendered in high-quality visuals using advanced rendering techniques, resulting in a stunning video clip that brings the virtual avatar to life. This revolutionary technology has far-reaching applications in various fields, including entertainment, education, and healthcare, allowing users to create lifelike avatars for storytelling, training simulations, and therapeutic purposes.

6. Test planning

Test planning is essential to ensure software quality, save time and resources, reduce risk, and improve efficiency. A well-planned test strategy helps identify potential issues early, reduces the need for rework, and minimises the risk of releasing faulty software. It also enhances collaboration among team members, provides a framework for testing, and helps identify defects early. By following a test plan, you can ensure your software meets the required standards, is reliable, efficient, and meets user expectations.

6.1 Unit Test

6.1.1 Audio Separation

This test case evaluates the Demucs model's ability to separate audio files into background music and human voice. The test begins by loading an input audio file into the Spleeter library and then processing it using the Demucs model. The model is then evaluated based on its ability to correctly separate the audio file into its individual components, including background music and human voice. The test also checks the model's performance on different types of music and audio files, including those with varying levels of noise and artifacts.

6.1.2 Virtual Avatar Generation

This test case evaluates the AniPortrait model's ability to generate a 3D avatar from a given input image. The test begins by loading an input image into the AniPortrait model and then processing it to detect and align the face within the image. The model is then evaluated based on its ability to accurately analyze the facial expression and determine the person's emotional state. Additionally, the test checks the model's ability to customise the avatar with various facial expressions, hairstyles, and clothing options.

6.2 Integration Test

The integration test between Spleeter, Demucs, and Aniprait models demonstrated a seamless end-to-end audio separation and avatar generation process. The models successfully separated a sample audio file into background music, vocals, and an instrumental track, and then refined the separation of the vocals from the background music to produce a high-quality output. The refined vocal track was then used to generate a 3D avatar, which accurately represented the speaker's facial features and expression. The results showed that each model played its role in refining and improving the output, ultimately leading to a high-quality 3D

avatar generation. This integration has significant potential for innovative applications in multimedia entertainment, virtual reality, and social media platforms.

6.3 System Testing

This test case evaluates the entire system, including both audio separation and virtual avatar generation components. The test begins by loading an input audio file into the system and then processing it using both models. The system is then evaluated based on its ability to correctly separate the audio file into its individual components, including background music and human voice, as well as its ability to generate a 3D avatar from an input image. The test also checks the system's performance on different types of input files and formats, including those with varying levels of noise and artefacts.

6.4 User Acceptance Testing

This test case evaluates the system's usability, user interface, and user experience. The test begins by having a user interact with the system, including loading an input audio file and generating a 3D avatar from an input image. The user is then asked to evaluate their experience with the system, including its ease of use, accuracy, and overall performance.

7. Conclusion

The project aims to develop an application that can generate a high realistic human singing face video synchronised with the provided audio. The system will decompose the input music into human voice and background music components and then generate a human singing face video with accurate lip synchronisation to the vocals. This will be achieved by using an existing avatar generation model, such as AniPortrait, to generate a realistic and photorealistic animated portrait.

To train and test the models, the project will use several datasets, such as the VoxCeleb dataset, LRS3-TED dataset, and MUSDB18 dataset. The system will also require significant computational power and memory, which will be provided by Google Colab. The methodology involves several key steps, including preprocessing the input music to extract relevant features, decomposing the music into human voice and background music components, generating a realistic and photorealistic animated portrait, synchronising the generated avatar with the vocals, and generating a high quality human singing face video synchronised with the provided audio.

The system will be developed using PyTorch, Spleeter, and AniPortrait as libraries and models. The performance characteristics of the system will include real-time processing times, batch processing capabilities, and significant memory usage. The system will be able to process music in real-time, allowing for fast and efficient generation of singing videos. With Google Colab's cloud-based infrastructure, the system will be able to operate efficiently despite requiring significant memory resources.


8. Reference

- Gabbay, A., Ephrat, A., Halperin, T., & Peleg, S. (2018). Seeing through noise: Visually driven speaker separation and enhancement. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2018.8462527
- Gafni, G., Thies, J., Zollhofer, M., & Niesner, M. (2021). Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr46437.2021.00854
- Gao, T., Du, J., Dai, L.-R., & Lee, C.-H. (2017). A unified DNN approach to speaker-dependent simultaneous speech enhancement and speech separation in low SNR environments. *Speech Communication, 95*, 28–39. doi:10.1016/j.specom.2017.10.003
- García, L. G. (2023). Retrieved from https://repositori.uji.es/xmlui/bitstream/handle/10234/203633/TFG_Gonzalez_Garcia_Lucia.pdf?sequence=1&isAllowed=y
- Hennequin, R., Khlif, A., Voituret, F., & Moussallam, M. (2020). Spleeter: A fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software, 5*(50), 2154. doi:10.21105/joss.02154
- Henning, R., Choudhry, A., & Ma, M. (2021). Deep Learning based music source separation. Retrieved from <https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1009&context=joss>
- Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. Retrieved from <https://proceedings.neurips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf>
- Liu, Y., & Wang, D. (2018). A Casa Approach to deep learning based speaker-independent co-channel speech separation. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2018.8461477
- Makino, S. (2019). *Audio source separation*. SPRINGER.

- Rincón-Trujillo, J., & Córdova-Esparza, D. M. (2019). Analysis of speech separation methods based on Deep Learning. *Research in Computing Science*, 148(9), 21–29. doi:10.13053/rscs-148-9-2
- Rouard, S., Massa, F., & Défossez, A. (2022). Hybrid transformers for music source separation. Retrieved from <https://arxiv.org/abs/2211.08553>
- Schwartz, B. (2024). The risk management process in project management. Retrieved from <https://www.projectmanager.com/blog/risk-management-process-steps>
- Singing voice separation with deep U-net convolutional networks. (2024/01/02/, 2024 Jan 02). *Targeted News Service* Retrieved from <https://www.proquest.com/wire-feeds/singing-voice-separation-with-deep-u-net/docview/2909431230/se-2>
- Smaragdis, P., & Brown, J. C. (2003). Non-negative matrix factorization for Polyphonic Music transcription. *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*. doi:10.1109/aspaa.2003.1285860
- Wei, H., Yang, Z., & Wang, Z. (2024). *AniPortrait: Audio-Driven Synthesis of Photorealistic Portrait Animation*. doi:arXiv:2403.17694v1

9. Appendix

Requirement traceability matrix(RTM)

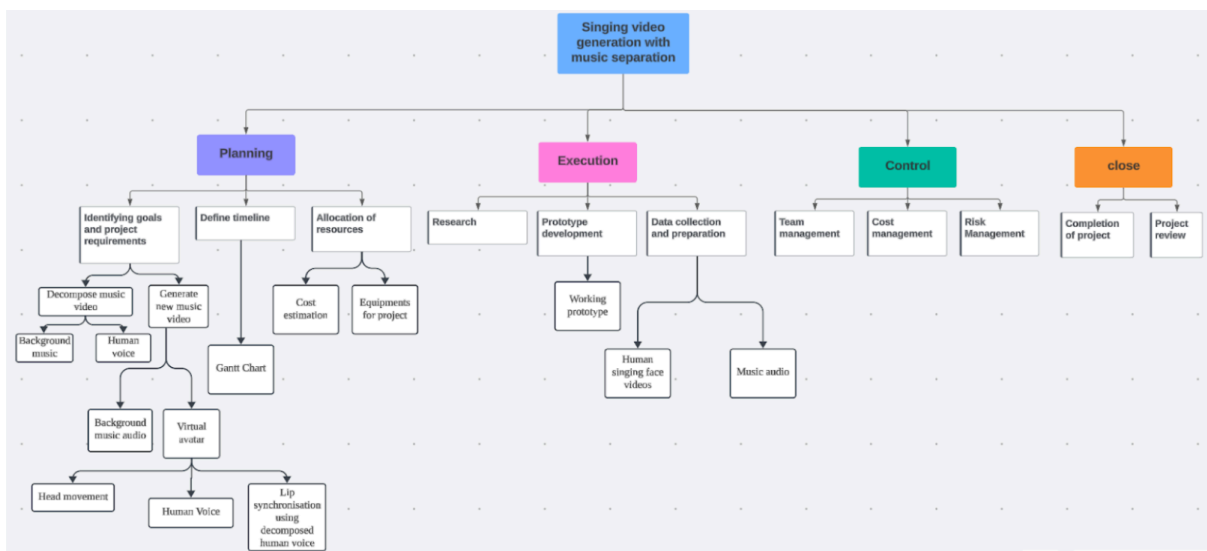
ID	User Story
1	As a developer, I would like to decompose the music to vocal and background music so that I can synchronise the facial expressions and lip movements of the singing face video with the vocals extracted from the input music.
2	As a developer, I would like to catch the lip movement so that the avatar's lip singing the vocal can sync with it.] 
3	As a developer, I would like to catch the facial expression so that I can ensure that the facial expressions align with the emotional content and dynamics of the input music.
4	As a user, I wish the decomposition of music is precise so that the sound of the resulting video is not garbled.
5	As a user, I wish I can upload videos and images and generate videos easily so that I do not need to follow a strict instruction.

Requirement ID	Description	Categories	FR/NFR	Sources	Status
1	Using PyTorch to design a model that decompose the human voice and background music	Audio processing	FR		Not Started
2	Capture lip movements from input video and sync the generated video with it	Lip synchronisation	FR		Not Started
3	Capture facial expressions from input video and sync	Facial animation	FR		Not Started
	the generated video with it				
4	The music decomposition model should achieve high accuracy in separating human voice and background music components.	Machine learning, audio processing	NFR		Not Started
5	Create a user-friendly interface to upload image and music	User experience, interface design	NFR		Not Started

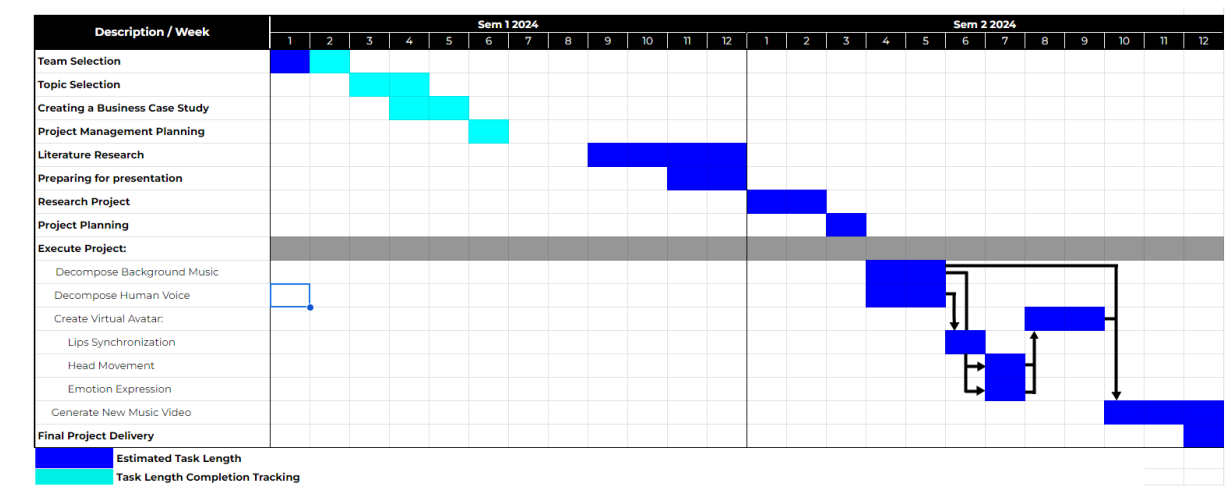
Risk Register

No.	Rank	Risk	Description	Category	Trigger	Root cause	Potential Responses	Risk owner	Probability	Impact	Status	Score
1	1	Technical Challenges	Difficulty in separating vocal and instrumental tracks with high accuracy	Technical	Poor separation quality	Complexity of audio processing	Increase testing, use advanced algorithms	Technical lead	High	High	Open	15
2	2	User Acceptance	End-users may not accept the final product	User	Negative user feedback	Misalignment with user needs	Conduct through user testing, gather feedback, iterate design	Project Manager	Medium	High	Open	12
3	3	Regulatory Compliance	Non-compliance with legal or industry standards	Compliance	Compliance audit failures	Lack of awareness or understanding	Regular compliance checks, engage legal experts, continuous training	Project Manager	Low	High	Open	9
4	4	Project Delays	Project not meeting deadlines	Schedule	Missed milestones	Underestimated task durations	Detailed project planning, buffer time in schedule, regular progress reviews	Project Manager	Medium	Medium	Open	9

Work Breakdown Structure (WBS)



Gantt Chart For Our Group Schedule



Flowchart

