Singing Video Generation with Music Separation

Supervised by: Dr Arghya Pal

Member Name:

1. Toh Xi Heng
2. Yew Yee Perng
3. Yeoh Ming Wei

Introduction

Avatar Face Generation

Output of Video Includes Virtual Avatar And Motion Expression
Through Generation By Inputting Single Audio And Image

Input

Output

Yeoh Ming Wei

Project Overview and Requirements

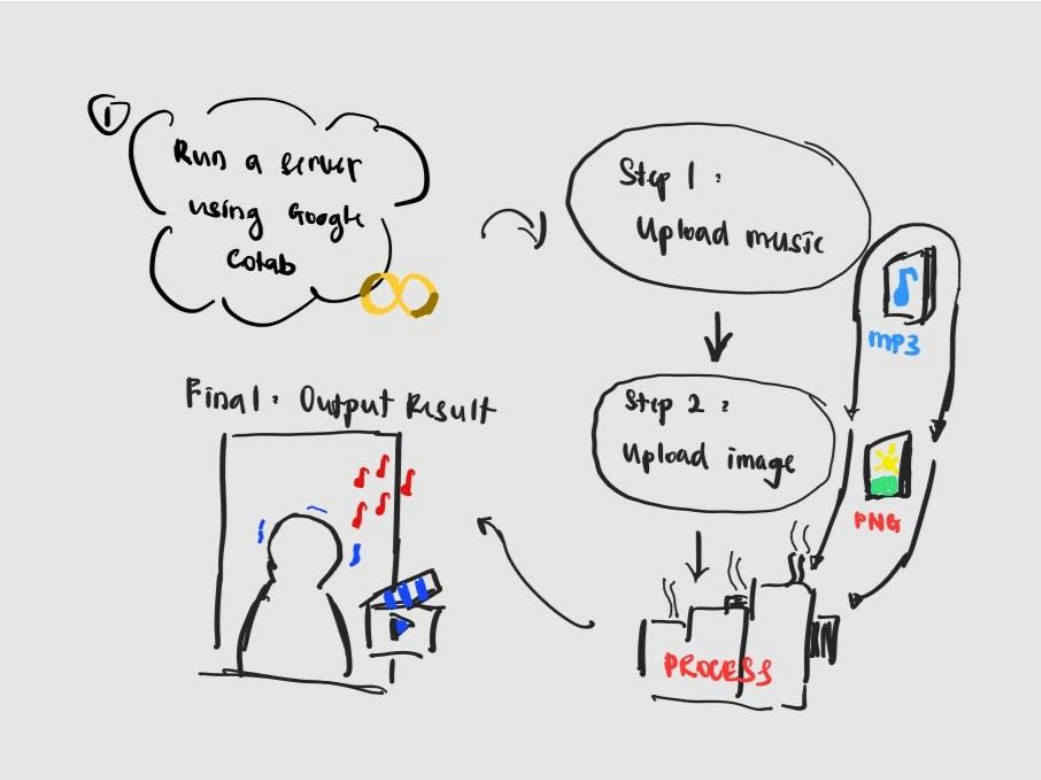
Aim

Develop a web application to generate a high-quality human singing face video with a provided song.

Requirement

- Decomposing the input music into human voice and background music components
- Generating a human singing face video with accurate lip synchronization to the vocals
- Test the system using variety of songs, Evaluate the performance

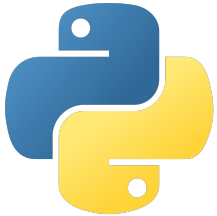
Project Outcome



Methodology

Tools and Frameworks

Programming Language



Python

Libraries & Frameworks:

Flask
(Web Framework)

Bootstrap Flask
(CSS Framework)

OpenCV
(Video Frames Capturing)

TensorFlow
(Prediction Model)

Methodology

Tools and Frameworks

Integrated Development Environment (IDE)

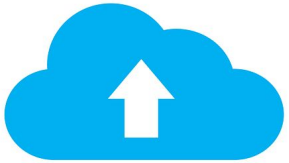
Visual Studio Code (Local)

Google Colab (Jupyter Notebook)

Cloud Server

Google Colab

Database

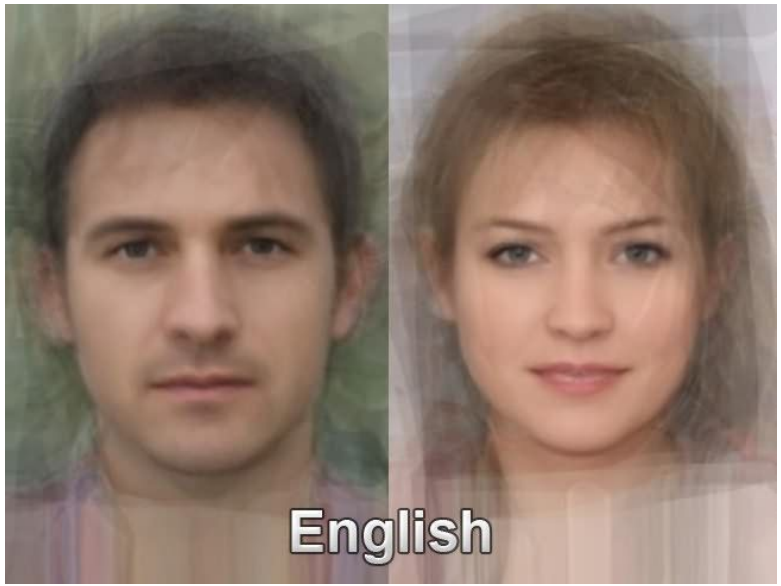


Cloud Storage in Google Colab (Temporary)

Methodology

Model Used:

- 3D Morphable Model (3DMM)
- PoseVAE
- ExpNet
- FaceRender



Toh Xi Heng

Deliverables

End-User Product:

- A Python-based software package for image/video preprocessing and video generation.
- An API that allows users to upload their image/audio and run processing automatically.

Usage:

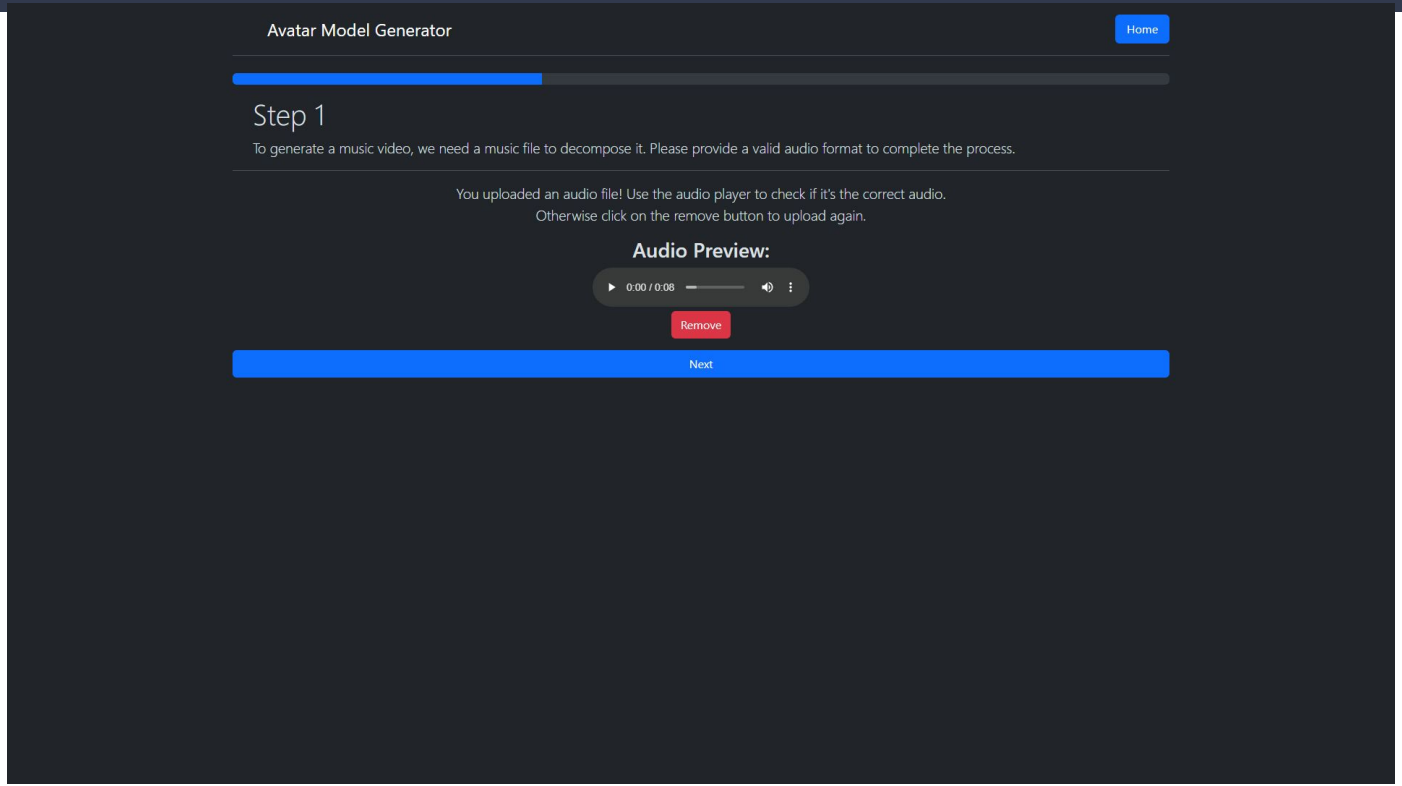
- Users can input image/audio via the web-based dashboard.
- The system processes the data, generate the video and return a video to show.

Approximate Testing:

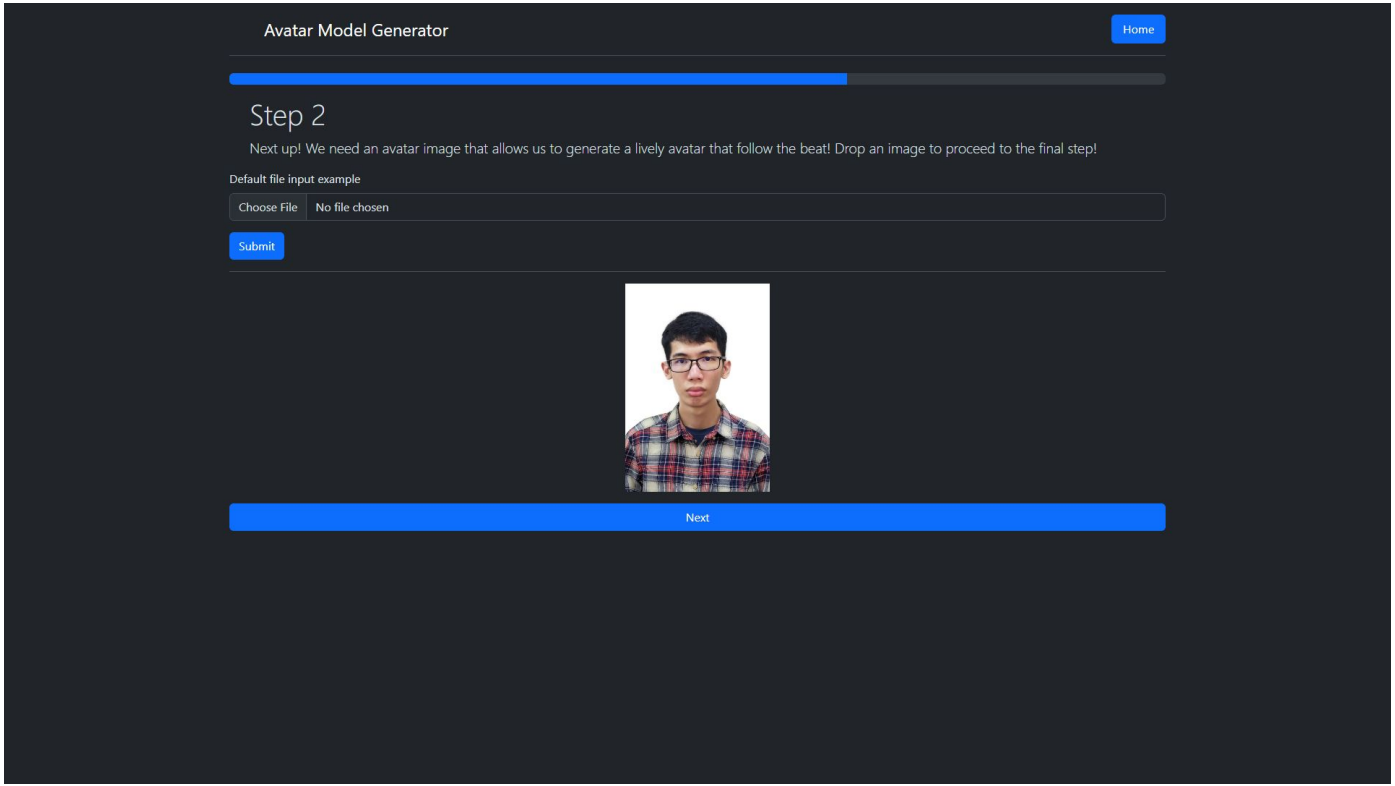
- Preprocessing module fully tested and validated with image(png) and audio(mp3).
- Image, audio, generated video can review.
- Full system testing done.

Toh Xi Heng

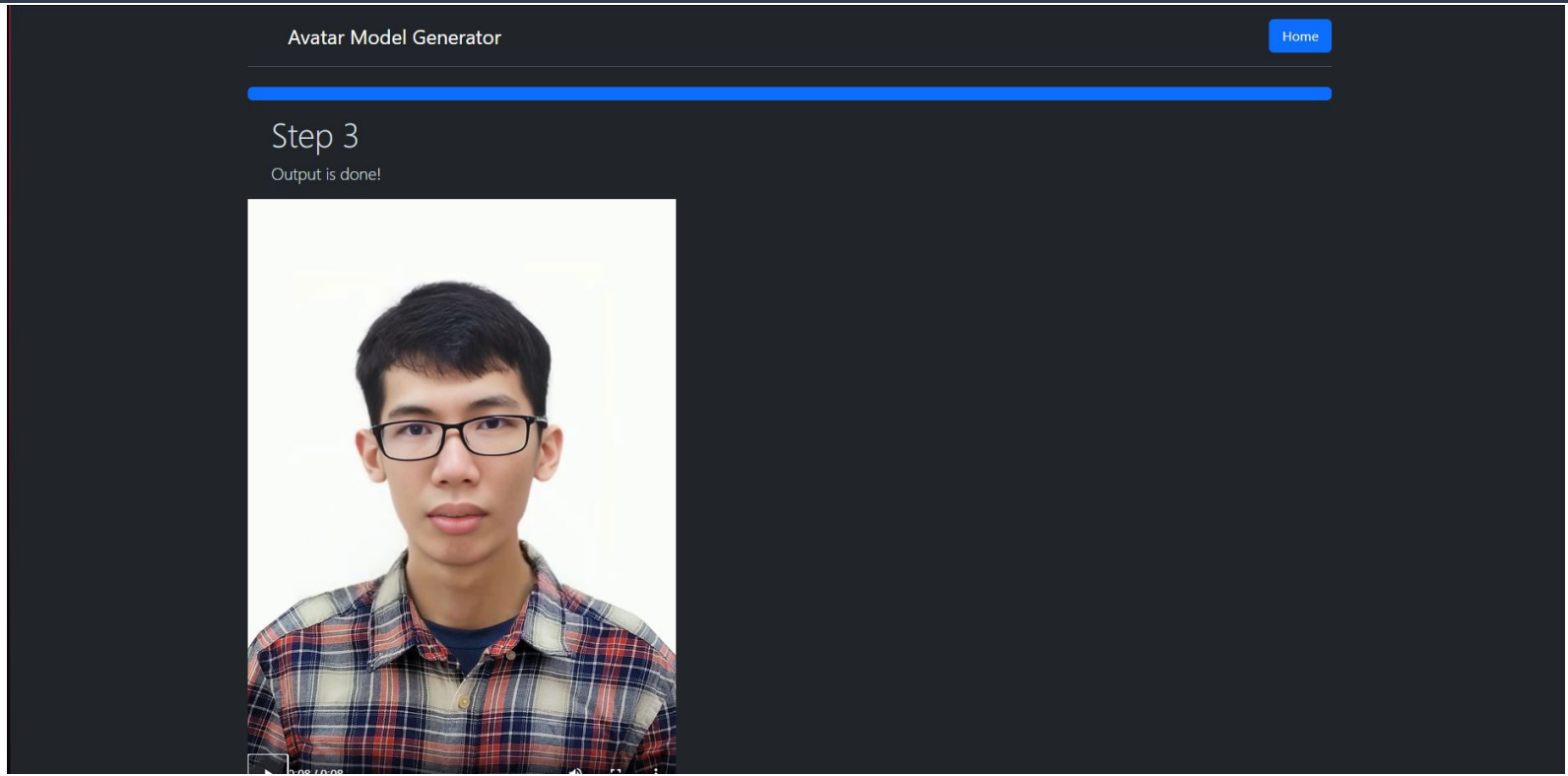
Software Showcase – Upload Audio



Software Showcase – Upload Image



Software Showcase – Output



Common Software Quality Attributes

Robustness

- error handling on input format
- stable performance on heavy workload

Security

- no personal information stored (Temporary storage)

Usability

- Simplistic User Interface

Scalability

- Allow users to use simultaneously without affecting performance

Benefits of Deliverables

1. **Who benefits from the deliverables?**

2. **What can they use the app for?**

Content Creators

Benefit:

- Easily create animated videos from static images, saving time and effort in video production.

Use Case:

- YouTubers, influencers, or social media managers can create personalized talking head videos without the need for complex animation skills.

Demo

Yeoh Ming Wei

Critical Discussion

Successes

- implementation of key features

Challenges

- performance bottlenecks

Problems experienced

- unmet objectives
- unexpected bugs

Problem Issues – Video Quality Optimization

Make Video to Frames - Using OpenCV

Raster Image to Vector Image

- Convert the Frames(JPG) to Frames(SVG)
- Raster image is image with pixels
- Vector image is image with scalable vector graphics


RASTER


VECTOR

Toh

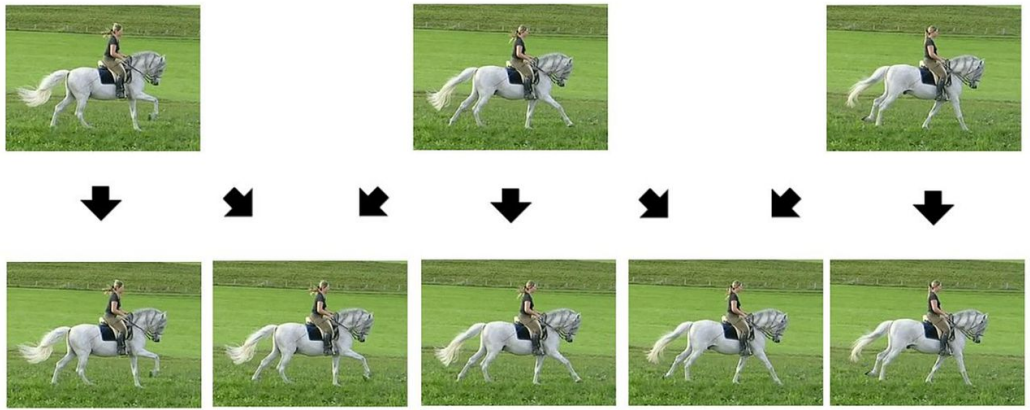
Future work and improvement

- Video Quality Optimization
- Improvements on Expressive Animation
- Accepting more input formats
- Temporary storages to store the previous image/audio uploaded

Future work and improvement

- Preprocessing image to optimize image quality before integrating image for generating video

- Improve the smoothness of the generated video (FPS)



Thanks!

Q&A