INHA UNIVERSITY IN TASHKENT

Multimedia Computing Course

Technical report
for
Video Editing Software

Submitted by

Zebo Botirova u1610255

Contents

Abstract

In order to technically explain the process of implementing our final term project, this report will be accomplished. Topic selection areas were freely chosen by the students to improve their skills in the fields of image processing, computer vision, machine learning and even in deep learning. Importance of idea of the project is marked highly rather than its development according to its useful and capturing features and future goals for improvement also considered. Furthermore, it is not compulsory to choose which programming language and usage of algorithms. Although it is small project or partly done, student should show his contribution and give explanation.

Introduction

This project implementation is done in the field of image processing. The reason why this area has been selected is that being informed about this course by the topics that were covered in other courses and encountering with problems related to this sphere in real life.

Report includes general explanation of the process creating my video editing software. Facing with difficulties while editing the videos or making short videos that consist of images urged me to make my own video editor program by python. The reason why I wanted to use python is that almost every type of software can be generated by this programming language. It has huge possibilities in many spheres including, image processing, as well and it has Moviepy library to help changing video streams easily.

Users have difficulties with the usage of existing video editors or it is impossible to find appropriate one for free. Therefore, this idea seemed me unusual and interesting. Additionally, it has many features that can be developed in my future career. Currently, short video editing can be done without challenges. Namely:

- User should give the path of the folder which consist of selected pictures
- Program initially, collect the pictures (pictures should be sorted and sized the same) from the given directory and make a video that sequentially displays the pictures
- In the next steps, user can use this video to add audio or text. For this also path of the input video, output video and audio, and watermark which, user wanted to inter in different font, color, are asked. Then program make ready the edited video.

Implementation method

1. Making initial video from images
    - Necessary libraries: moviepy.editor, PIL and os
    - image_folder_path object is created to ask the path from the user
    - input directory that includes all the images and output video will be specified

    ```
    thumbnail_dir=os.path.join(image_folder_path, "images")
    output_video=os.path.join(image_folder_path, "video1.mp4")
    ```

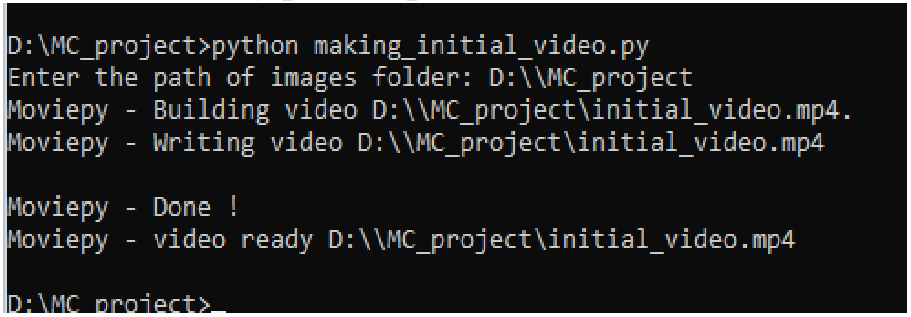
    - in order to make it easy to sort pictures can be reanmed with numbers and they will be in a specific oreder. In this way dictionary will be created to save the path and order. a new path is also created to include all image paths in order.
    - Then we will create a video of all the images using ImageSequenceClip by specifying either fps or the duration of the video.
2. Adding or changing audio of the video
    - Necessary libraries: AudioFileClip and VideoFileClip
    - Objects are created to ask from the user the video path, audio path, final video output directory, final video name, starting and ending time.
    - Objects of VideoFileClip and AudioFileClip are created then used part of the audio will be specified by user
    - Audio is integrated to video
3. Adding watermark and intro
    - Necessary libraries TextClip, CompositeVideoClip
    - Objects are created to ask from the user the video path, audio path, final video output directory, final video name, watermark.
    - object of the video and audio file is created and trim the audio according to video duration
    - intro is designed with TextClip class object with text, font size, color and size of clip.
    - The same thing is implemented for watermark
    - Videos will be combined.

Experiment result

Although this is simple project, result is worth to make you happy. User interface implementation is expected later for this project but for now it works on terminal.

As addition it should be mentioned that proper installation of libraries plays important role in taking the output, especially, ImageMagick library installation.

1. User enters only path of pictures

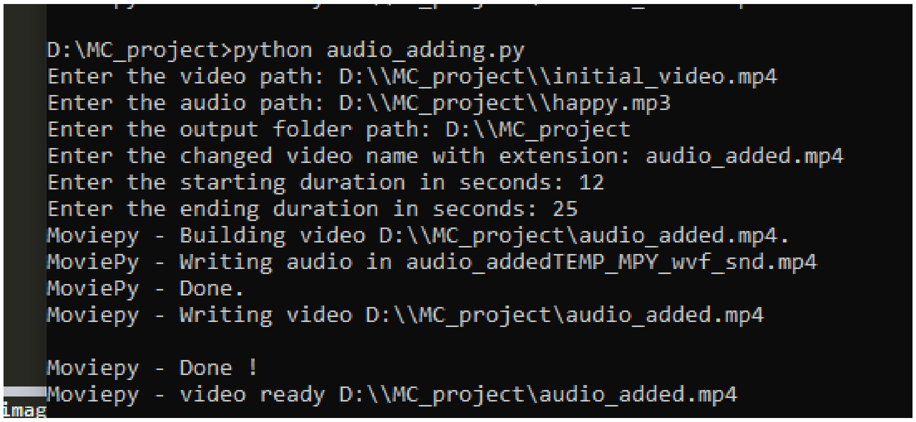


```
D:\MC_project>python making_initial_video.py
Enter the path of images folder: D:\\MC_project
Moviepy - Building video D:\\MC_project\initial_video.mp4.
Moviepy - Writing video D:\\MC_project\initial_video.mp4

Moviepy - Done !
Moviepy - video ready D:\\MC_project\initial_video.mp4

D:\MC_project>_
```

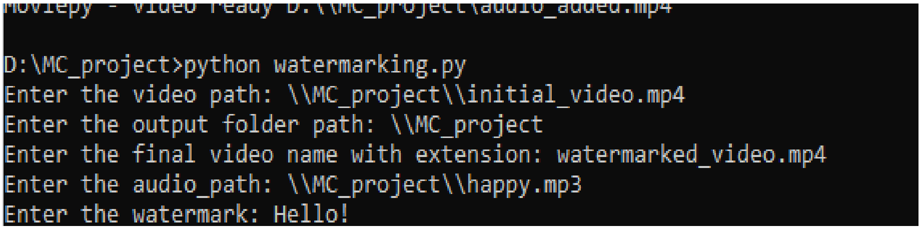
2. Here is shown what user should enter



```
D:\MC_project>python audio_adding.py
Enter the video path: D:\\MC_project\\initial_video.mp4
Enter the audio path: D:\\MC_project\\happy.mp3
Enter the output folder path: D:\\MC_project
Enter the changed video name with extension: audio_added.mp4
Enter the starting duration in seconds: 12
Enter the ending duration in seconds: 25
Moviepy - Building video D:\\MC_project\audio_added.mp4.
MoviePy - Writing audio in audio_addedTEMP_MPY_wvf_snd.mp4
MoviePy - Done.
Moviepy - Writing video D:\\MC_project\audio_added.mp4

Moviepy - Done !
Moviepy - video ready D:\\MC_project\audio_added.mp4
```

3. Adding watermark and intro also is the same



```
Moviepy - video ready D:\\MC_project\audio_added.mp4

D:\MC_project>python watermarking.py
Enter the video path: \\MC_project\\initial_video.mp4
Enter the output folder path: \\MC_project
Enter the final video name with extension: watermarked_video.mp4
Enter the audio_path: \\MC_project\\happy.mp3
Enter the watermark: Hello!
```

Output file will be included in zip file.

Reference

https://pypi.org/project/moviepy/

https://python.org

https://medium.com

https://zulko.github.io/moviepy/