

Video Editing Application using Python (OPEN-CV)

G.Akshith
B.Tech
School of Engineering
Hyderabad, India
2111CS020037@mallareddyuniversity.ac.in

K.Akshith
B.Tech
School of Engineering
Hyderabad, India
2111CS020038@mallareddyuniversity.ac.in

Ch.Akshitha
B.Tech
School of Engineering
Hyderabad, India
2111CS020039@mallareddyuniversity.ac.in

P.Akshitha
B.Tech
School of Engineering
Hyderabad, India
2111CS020040@mallareddyuniversity.ac.in

M.Akshitha
B.Tech
School of Engineering
Hyderabad, India
2111CS020041@mallareddyuniversity.ac.in

E.Akshaya
B.Tech
School of Engineering
Hyderabad, India
2111CS020042@mallareddyuniversity.ac.in

Guide: A. Kalyani
Asst. Professor
School of Engineering,
Mallareddy University
kalyani.a@mallareddyuniversity.ac.in

Abstract: The aim of the project is we will get to know the method detecting to play a video in reverse mode using OpenCV Python. Here, we will learn about video streaming and playing in reverse mode, Resizing and Rescaling Frames, Drawing Shapes & Putting Text. Also, we will see a Python program for the same.

OpenCV (Open Source Computer Vision Library) is Python library to solve computer vision related problems. OpenCV is a vast library which provides various functions for video operations. Using OpenCV, we can capture a video from the webcam. In this Project we use Tkinter and Moviepy modules for our Graphical interface. It lets you create a video capture object which helps to capture videos through webcam and then we can perform desired operations on that video.

We can solve many real time problems using image processing applications. In this paper, sample real time image processing applications of OpenCV are discussed along with steps.

Keywords:- Tkinter, Moviepy

I. INTRODUCTION

Video editing applications are software tools used to manipulate and enhance video footage. These applications can be used by individuals or professionals to create high-quality videos for various purposes, such as social media content, marketing campaigns, or movies

However, despite their advanced capabilities, video editing applications have limitations that users should be aware of. For example, these tools require a significant amount of processing power and storage space, making them less accessible for those with lower-end hardware. Additionally, mastering the various features and techniques of these applications can be time-consuming and challenging, especially for beginners.

Moreover, some advanced features like artificial intelligence-based editing tools or complex 3D compositing can require expensive add-ons or dedicated hardware, which may not be affordable for everyone.

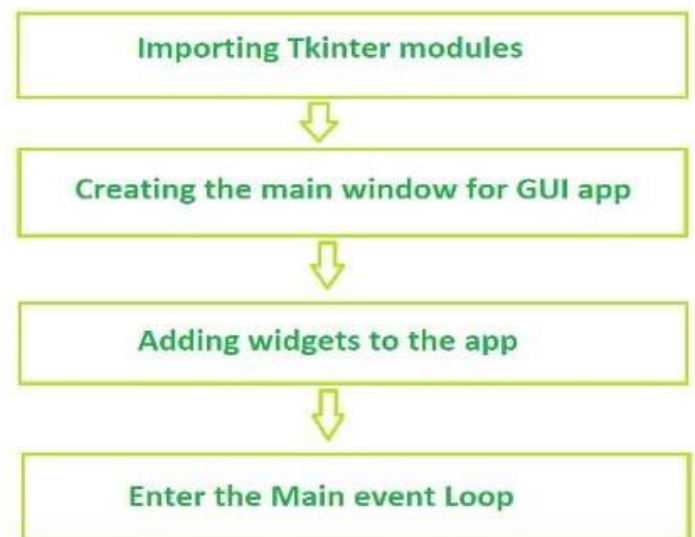
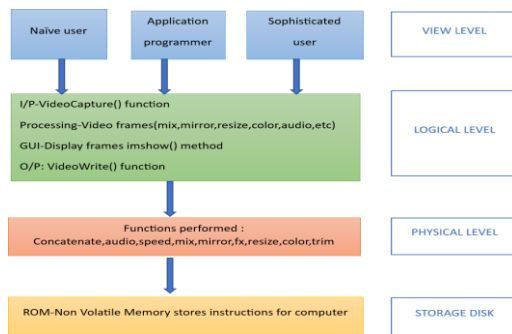


Fig: Basic Video editing file

II. PROBLEM STATEMENT

Take a video as input and break the video into frame by frame and save those frames. Now, a number of operations can be performed on these frames. Like reversing the video file or cropping the video etc. For playing video in reverse mode, we need only to store the frames in a list and iterate reverse in the list of frames. Use the reverse method of the list to reverse the order of frames in the list. Use Python Open

(a) Architecture



I. LITERATURE REVIEW

In today's digital age, video editing has become a crucial tool for professionals and enthusiasts alike. With the rise of social media platforms, online content creation, and video marketing, the demand for video editing tools has increased dramatically. However, many existing video editing functions are designed to function individually, which can limit their potential and usefulness. Individual video editing functions are those that operate as standalone tools, and they do not integrate or interact with other editing functions. For example, a video editing software that only allows users to trim or crop videos is an individual editing function. Similarly, a color correction tool that only applies a single effect is also an individual editing function. While these functions may serve their intended purpose, they have limitations in terms of versatility, efficiency, and creativity.

One of the biggest drawbacks of individual video editing functions is their limited versatility. Since each function operates independently, there is often a lack of integration and compatibility between them. As a result, users may have to switch between different editing tools to accomplish a single task, which can be time-consuming and frustrating. For instance, if a user wants to apply multiple effects to a video, they may have to use several different individual functions, which can create a disjointed editing process.

II. REQUIRED TOOLS

- VS CODE
- Jupyter notebook
- Python3

- Tkinter
- Movie py

III. METHODOLOGY

However, the limitations of individual video editing functions have become apparent, as they lack versatility, efficiency, and creativity. To address these limitations, our application has integrated all video editing features together, allowing for a more seamless and efficient editing experience. This report will discuss the benefits of this approach and how it can improve the video editing experience.

MoviePy is a popular Python library used for video editing tasks. It provides a simple and intuitive API for creating, editing, and processing videos, making it an excellent choice for beginners and experienced users alike.

Video Editing Functions: MoviePy provides a wide range of video editing functions, such as cutting, resizing, and rotating videos. These functions are easy to use and can be chained together to create complex editing operations.

Tkinter is a popular Python GUI toolkit used to create graphical user interfaces. While it is not specifically designed for video editing, Tkinter can be used to build custom interfaces for video editing applications.

Custom UI: Tkinter can be used to create custom user interfaces for video editing applications. This can include buttons, menus, and sliders for controlling video playback, editing, and effects.

Preview Windows: Tkinter can be used to create preview windows for displaying the edited video. Users can play, pause, and scrub through the video to see how it looks with the applied effects and edits.

IV. EXPERIMENT RESULTS

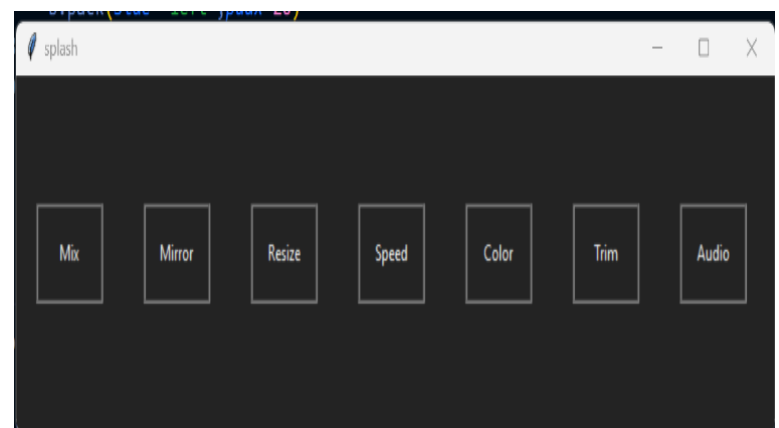


Fig. 2. Main Screen

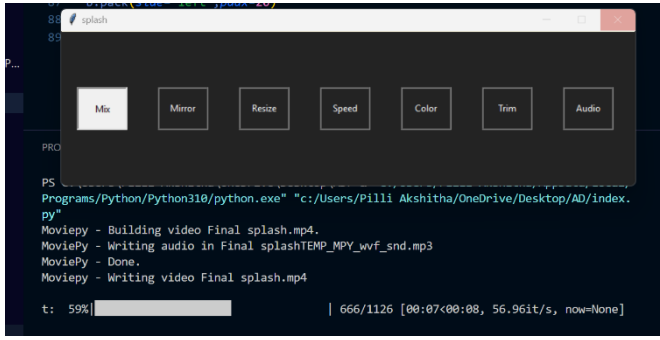


Fig. 3. Executing GUI

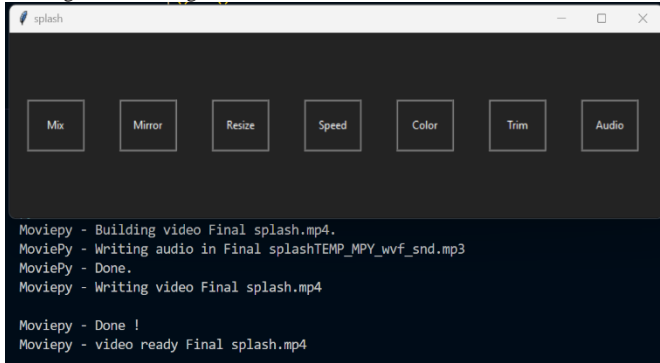
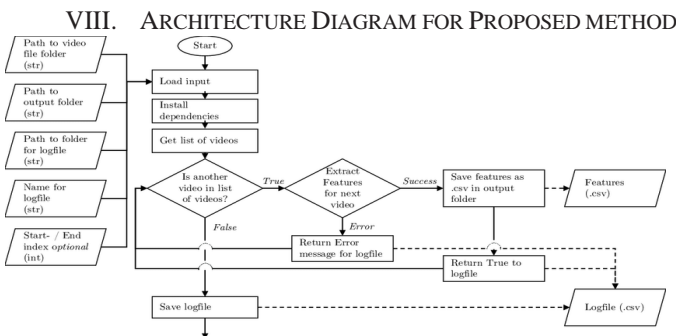


Fig. 4. Output Screen



VII. MERITS OF PROPOSED SYSTEM

Versatility: By integrating all video editing features together, our application offers a broader range of editing tools and features that can work together seamlessly. This allows users to have more options when it comes to editing their videos, such as applying multiple effects simultaneously or creating complex compositions involving multiple layers and effects.

Efficiency: Our application eliminates duplication of effort and reduces data transfer and management costs by integrating all video editing features together. Users no longer have to switch between different editing tools to accomplish a single task, which saves time and reduces frustration.

Creativity: By integrating all video editing features together, our application allows users to experiment with different editing techniques and effects without having to switch between different programs. This can lead to more creative and innovative videos that stand out in a crowded marketplace.

Fig. 5. Architecture Diagram

IX. CONCLUSION:

According to this research paper In conclusion, our application's integrated video editing features offer several benefits over individual editing functions. They are more versatile, efficient, and creative, allowing users to create high-quality videos with ease. By offering a comprehensive suite of video editin features , our application provides users with the tools they need to create professional-rade videos that stand out in today's digital landscape.

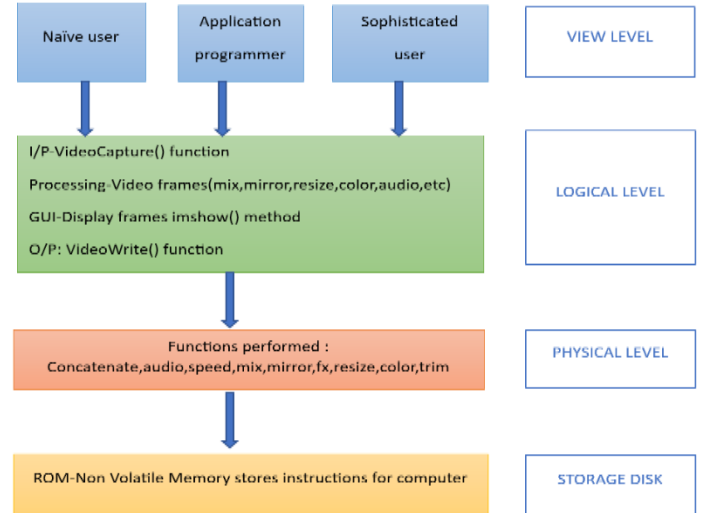


Fig. 6. Views

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and appreciation to my project guide and Head of Department for their invaluable support and guidance throughout the development of this project. Their unwavering encouragement and valuable insights have been instrumental in shaping my ideas and ensuring the successful completion of this project.

My guide's expertise and dedication have been pivotal in guiding me through the various stages of this project. They have provided me with the necessary resources and feedback to refine my work and bring out the best in me. Likewise, the Head of Department's constant encouragement and support have been crucial in motivating me to stay focused and dedicated to my work. Without their support, this project would not have been possible, and I am deeply grateful to them for their Guidance and mentorship

REFERENCES

- [1] GUI Reference : <https://youtu.be/yRAw-XderV8>
Modules and Theory Segment : ChatGPT
- [2] Tkinter module : Geek for Geeks
<https://www.geeksforgeeks.org/python-gui-tkinter/>
- [3] Moviepy module : Geek for Geeks website
<https://www.geeksforgeeks.org/introduction-to-moviepy/>
- [4] OpenCV : <https://www.geeksforgeeks.org/opencv-python-tutorial/>