

HOW TO CREATE PROFESSIONAL VIDEO CONTENT IN PYTHON

Kamalova Nilufar Ilkhomovna

*Bukhara State University Lecturer of the Department of Applied
Mathematics and Programming Technologies, Doctor of Philosophy (PhD)*

Annotation: The article discusses the use of the MoviePy library for creating and editing video content in the Python environment. The MoviePy library provides a wide range of possibilities for working with video files, including merging, slicing, resizing and speed of videos, as well as creating slideshows and animations. Examples of using MoviePy for various tasks are presented, such as creating animations, adding text to videos, cropping video fragments and creating a slideshow with music. The article also discusses the advantages and convenience of working with video files using MoviePy, emphasizing its ease of use, rich functionality and high performance.

Keywords: Python, moviepy, video editing, video effects

Introduction

Moviepy is a library for working with videos in Python, which allows you to create, edit and process video files. It has many features such as merging, slicing, resizing and speed of video, adding text, images, sound, etc. The library also supports working with various video, audio and image formats. Thanks to its simple and intuitive interface, Moviepy is ideal for both beginners and advanced users who need to process video files quickly and efficiently.

Ease of use: the library has a clean and understandable syntax, which makes the process of working with video files quite simple even for beginners. MoviePy provides a wide range of video editing features, including cropping, merging, adding sound, text and images, changing the speed and size of the video, and more. The library supports working with various video, audio and image formats, which makes it convenient to process and convert files. MoviePy is well optimized for working with videos and allows you to process large files quickly and efficiently. The library has an active community of users and developers, which provides support and the opportunity to get help if necessary. In general, MoviePy is a convenient and powerful tool for working with video files in Python, which allows you to easily and quickly create and edit video content.

Methods

To create a new video using the Movie Py library, follow these steps:

1. Install the MoviePy library using pip: `pip install movies`

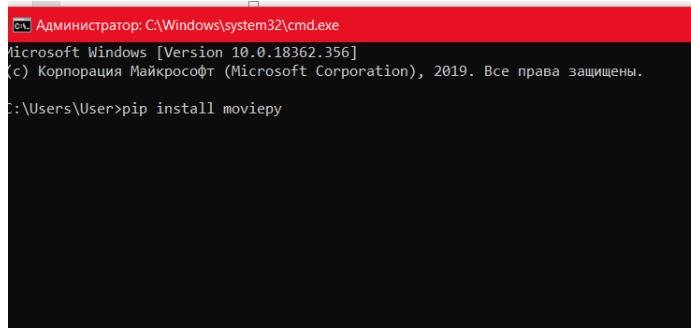


Figure 1. Installing moviepy via pip.

2. Import the necessary modules from the MoviePy library:

```
from moviepy.editor import *
```

3. Create a new video clip. For example, to create a video from an image, you can use the following code:

```
from moviepy.editor import *
image_uzb = "Uzbekistan.png"
clip = ImageClip(image_uzb, duration=5)
clip.write_videofile("Uzb.mp4", fps=24)
print("FINISH")
```

4. Create video clips that need to be combined into a new video. For example, to create a new video clip from two existing video files, you can use the following code:

```
from moviepy.editor import *
video1=VideoFileClip("bir.mp4")
video2=VideoFileClip("ikki.mp4")
birlashgan=concatenate_videoclips([video1, video2])
birlashgan.write_videofile("uch.mp4")
print("Bajarildi")
```

5. Save video by rotating 180 degrees:

```
from moviepy.editor import *
a=VideoFileClip("bir.mp4").rotate(180)
a.write_videofile("ikki.mp4")
```

6. The following code can be used to crop the video:

```
from moviepy.editor import *
a=VideoFileClip("bir.mp4").rotate(180)
a=a.subclip(5, 9)
a.write_videofile("uch.mp4")
```

7. To create video collages

```

from moviepy.editor import*
a=VideoFileClip("bir.mp4").subclip(1,3)
b=VideoFileClip("ikki.mp4").subclip(1,3)
c=VideoFileClip("bir.mp4").subclip(3,5)
d=VideoFileClip("ikki.mp4").subclip(3,5)
x=clips_array([[a,b],[c,d]])
x.write_videofile("uch.mp4")

```

After completing these steps, there will be a new video clip with a cropped fragment or a changed duration, saved in the format and quality you need.

When using the moviepy library, you can use various methods and functions provided by this library to apply filters and effects to videos. Below is an example showing how a filter can be applied to a video:

```

from moviepy.editor import *
video = VideoFileClip("Uzb.mp4")
filtered_video = video.fx(vfx.blackwhite)
filtered_video.write_videofile("oq-qora.mp4")

```

This example uses the `.fx()` method, which allows you to apply various effects to the video. In this case, a black and white filter is applied to the video and a new video clip with the applied effect is saved.

MoviePy also provides many other built-in effects (such as overlay effects, color correction, blur, and more) that can be applied to videos. In addition to this, you can also create your own custom effects using the functions and methods of the MoviePy library.

To save and export the finished video file using the MoviePy library, you must use the `write_videofile()` method:

```

from moviepy.editor import *
video_clip = VideoFileClip("video.mp4")
processed_clip = video_clip.fx(vfx.rotate, angle=90)
processed_clip.write_videofile("output_video.mp4")

```

In this example, a video clip is created, then a 90-degree rotation effect is applied to the video clip, and finally the processed video clip is saved in mp4 format under the name "output_video.mp4".

When using the `write_videofile()` method, you can specify various parameters, such as frame rate (fps), video bitrate, codec, etc., to customize the saved video. For example:

```

processed_clip.write_videofile("output_video.mp4",
codec='libx264', fps=30, bitrate='5000k')

```

This code will save the processed video clip with the libx264 codec, a frame rate of 30 frames per second and a bitrate of 5000 kbps.

Thus, after editing the video using MoviePy, you can easily save and export the finished video file with the necessary settings.

Results

When working with video and maintaining its quality, it is important to consider several factors that may affect the final video quality. To optimize and maintain video quality:

- When saving a video, you need to specify the correct encoding parameters, such as codec, bitrate, frame rate and resolution. These parameters can significantly affect the quality and size of the video file.
- It is necessary to choose the optimal bitrate. Too low a bitrate leads to a loss of quality, and too high a bitrate leads to large files.
- A suitable resolution must be set for the video. It is important to use a resolution that matches the original video, or focus on the requirements of the platform on which the video will be used.
- You also need to select the appropriate codec for the video. Some codecs provide better quality with a smaller file size.
- The frame rate must be set to the optimal value. The frame rate of 24-30 frames per second is the standard for most videos.
- When storing videos, it is necessary to use variable bitrate instead of constant bitrate (CBR) for more efficient use of space and excellent quality.
- When re-editing a video, avoid multiple encoding, as this may lead to loss of quality. To find out the optimal parameters for the video, you need to experiment with various encoding settings.

By following these concepts, you can maintain high video quality and effectively optimize its size and parameters.

Discussion

The MoviePy library provides a wide range of features for working with videos in Python.

You can use MoviePy to create video presentations of images with the addition of text labels, background music and additional effects.

MoviePy makes it easy to resize videos, add captions and logos to optimize videos for posting on social networks.

You can use MoviePy to create simple animations, cinematic effects, and video clips.

MoviePy can be useful for video processing in scientific research, including motion analysis, timelapse creation, and data visualizations.

You can use MoviePy to create creative content, including video reviews, tutorials, music videos, and more.

MoviePy allows you to create scripts for automated video processing, which can be especially useful when working with a large number of video files.

This is just a small list of examples of using MoviePy in various tasks. Thanks to its flexibility and powerful features, the MoviePy library provides ample opportunities to work with videos and create high-quality video content.

Conclusion

Using the MoviePy library in Python opens up wide possibilities for creative processing and creation of video content. Thanks to its user-friendly interface, rich functionality and high performance, users can easily and efficiently edit video files, add special effects, text, sound and images, creating professionally designed video content. MoviePy is suitable for both beginners and experienced users, opening up new possibilities in the field of video processing and creating unique content. Modern video tools are becoming more and more accessible thanks to innovative solutions such as the MoviePy library, and allow us to expand the boundaries of creativity and bring any ideas and projects to life. As a result, MoviePy turns out to be an indispensable tool for working with videos and creating high-quality video content in Python.

List of literature used

1. Siahaan V., Sianipar R. H. DIGITAL VIDEO PROCESSING PROJECTS USING PYTHON AND TKINTER. – BALIGE PUBLISHING, 2024.
2. Satpute S. S. et al. Smart Video Summarization using Subtitles //International Research Journal Of Modernization In Engineering Technology And Science. – 2022. – T. 4. – №. 07.
3. Ilxomovna K. N. PYTHON DASTURLASH TILINI O ‘QITISHDA INTELLEKT TESTLARIDAN FOYDALANISH //Proceedings of International Conference on Educational Discoveries and Humanities. – 2023. – T. 2. – №. 5. – C. 274-278.
4. Qodirov A. Ta’limda Python dasturlash tilidan foydalanish //Engineering problems and innovations. – 2023.