How to Add Text to Video with FFmpeg?

Master the art of integrating text into your video projects using Ffmpeg

Transkriptor 2023-04-03

What is FFmpeg?

FFmpeg is a free project that provides a collection of programs for handling data such as audio, video, and images. It is a tool that is e used to convert, record, stream, and play files.

FFmpeg supports a wide range of formats and codecs and is used for tasks such as transcoding, resizing filters, adding watermarks or subtitles , getting audio from video, and merging multiple videos or audio files. It is also possible to create a GIF using FFmpeg and combine MP4 files using FFMPEG on Windows

FFmpeg is widely used in the media industry and is available on most operating systems, including Windows, macOS, Linux, and Unix.

How to Use FFmpeg?

Here are some common examples of how to use FFmpeg:

1. Convert a video file from one format to another:

Code: ffmpeg -i input_ video.mp 4 output_video.avi

This command will convert `input_video.mp4` to `output_video.avi` . FFmpeg will automatically select the appropriate codecs for the output format.

2. Resize a video:

Code: ffmpeg -i input_ video.mp 4 -vf scale=640:480 output_ video.mp 4

This command will resize `input_video.mp4` to a resolution of 640×480 pixels and save the result to `output_video.mp4` .

3. Extract audio from a video:

Code: ffmpeg -i input_ video.mp 4 -vn output_ audio.mp 3

This command will extract the audio track from `input_video.mp4` and save it to `output_audio.mp3` .

4. Concatenate two videos:

Code: ffmpeg -i input_ video1.mp 4 -i input_ video2.mp 4 -filter_complex concat output_ video.mp 4

This command will concatenate `input_video1.mp4` and `input_video2.mp4` and save the result to `output_video.mp4` .

5. Add a watermark to a video:

Code: ffmpeg -i input_ video.mp 4 -i watermark.png -filter_complex "overlay=10:10" output_ video.mp 4

This command will add the `watermark.png` image to the top-left corner of `input_video.mp4` with a 10-pixel offset and save the result to `output_video.mp4` .

These are just a few examples of what to do with FFmpeg. It is also possible to use FFmpeg on Android. For more information, consult online tutorials, video streamings, or AI chatbots such as ChatGPT.

What are FFmpeg Useful Video Editing Commands?

1. Trim a video:

Code: ffmpeg -i input_ video.mp 4 -ss 00:00:05 -t 00:00:10 -c copy output_ video.mp 4

This command will trim `input_video.mp4` to start at the 5-second mark and end after 10 seconds and save the result to `output_video.mp4` .

2. Add an image or video overlay on my video:

Code: ffmpeg -i input_ video.mp 4 -i overlay_image.png -filter_complex “overlay=10:10” output_ video.mp 4

This command will add the `overlay_image.png` file as an overlay to `input_video.mp4` at the coordinates (10, 10), and save the result to `output_video.mp4` .

4. Change video resolution:

Code: ffmpeg -i input_ video.mp 4 -vf scale=1280:720 -c:a copy output_ video.mp 4

This command will change the resolution of `input_video.mp4` to 1280×720 pixels and save the result to `output_video.mp4` .

- Add the `textfile` and `reload` options for drawtext:

Code: ffmpeg -i input.mp 4 -vf "drawtext=fontfile=/path/to/font.ttf:textfile=text.txt:reload=1:fontcolor=white:fontsize=24:box=1:boxcolor=black@0.5:boxborderw=5:x=(w-text_w)/2:y=(h-text_h)/2" -codec:a copy output.mp 4

How to Add Text to Video with FFmpeg?

To add text to a video with FFmpeg, use the `drawtext` filter. The `drawtext` filter allows you to specify the font, size, color, and position of the text, among other parameters.

Here is a basic command to add text to a video using the `drawtext` filter:

Code: ffmpeg -i input_ video.mp 4 -vf "drawtext=text='Hello World':fontfile=/path/to/font.ttf:fontsize=50:fontcolor=white:x=100:y=100" -codec:a copy output_ video.mp 4

In this command:

- `-i input_video.mp4` specifies the input video file.
- `-vf "drawtext=..."` specifies the video filter to apply, which is the `drawtext` filter in this case.
- `text='Hello World'` specifies the text to add to the video.
- `fontfile=/path/to/font.ttf` specifies the font file to use.
- `fontsize=50` specifies the font size in points.
- `fontcolor=white` specifies the font color.
- `x=100:y=100` specifies the position of the text in the video, with the top-left corner at (100, 100) pixels.
- `-codec:a copy` specifies to copy the audio stream from the input video to the output video without re-encoding it.
- `output_video.mp4` specifies the output video file.

What is Fontfile?

The `fontfile` option in FFmpeg is used to specify the path to a font file that will be used to render text in a video or image.

When you use the `drawtext` filter in FFmpeg to add text to a video, specify the `fontfile` option to choose a specific font to use for the text.

How to Include Font in FFmpeg Command without Using the Fontfile Option?

In order to use a font in an FFMPEG command without specifying the fontfile option, use a system-installed font.

Here is an example command that uses a system-installed font:

Code: ffmpeg -i input_ video.mp 4 -vf “drawtext=text='Hello World':fontcolor=white:x=100:y=100:font=Arial” -codec:a copy output_ video.mp 4

What is GitHub?

GitHub is a web-based platform that allows developers to host and manage their software projects, collaborate with others, and track and manage changes to their code. It provides tools for version control, issue tracking, code review, and team collaboration.

How to Add “StackOverflow” Text with FFmpeg?

Code: ffmpeg -i input_ video.mp 4 -vf “drawtext=text='StackOverflow':fontfile=/path/to/font.ttf:fontsize=30:fontcolor=white:x=100:y=100” -codec:a copy output_ video.mp 4

Here is what the various options in this command do:

- `-i input_video.mp4` specifies the input video file.
- `-vf "drawtext=text='StackOverflow':fontfile=/path/to/font.ttf:fontsize=30:fontcolor=white:x=100:y=100"` sets up the video filtergraph for drawing the text. This command draws the text “StackOverflow” on the video at the coordinates (100, 100), using a font file located at `/path/to/font.ttf` , with a font size of 30 and a white font color.
- `-codec:a copy` copies the audio stream from the input video to the output video without re-encoding it.
- `output_video.mp4` specifies the name of the output video file.

How to Display TimeCodes or TimeStamps Using FFmpeg’s Drawtext Filter?

Code: ffmpeg -i input_ video.mp 4 -filter_complex “drawtext=fontfile=/path/to/font.ttf:text=’%{pts/:hms}’:fontsize=24:fontcolor=white:x=10:y=10” -codec:a copy output_ video.mp 4

Here is what the various options in this command do:

- `-i input_video.mp4` specifies the input video file.
- `-filter_complex` allows you to use multiple filters in the same command.
- `drawtext` is the name of the filter.
- `fontfile=/path/to/font.ttf` specifies the path to the font file we want to use.
- `text='%{pts/:hms}'` sets the text to be displayed. `%{pts/:hms}` is a placeholder that will be replaced by the current timecode in hours:minutes:seconds format.
- `fontsize=24` sets the font size to 24.
- `fontcolor=white` sets the font color to white.
- `x=10` and `y=10` set the position of the text on the video.
- `-codec:a copy` copies the audio stream from the input video to the output video without re-encoding it.
- `output_video.mp4` specifies the name of the output file.

Share Post

Related Blogs

Best Video Editing Apps to Add Text to Video (2023)

How to Add Text to Video on Samsung Devices?

How to Add Text to Video on Vegas Pro?

Show More

Resources

Audio to Text

Video Transcription

Transcription Service

Links

Support

Blog

Privacy Policy

Terms of Service

Contact Information

customer@transkriptor.com

GET IT ON Google Play

Download on the App Store

Available in the Chrome Web Store

Dubai, UAE