

Installing ffprobe and ffmpeg

```
(base) Penda@M2-14-inch-Macbook-Pro ~ % brew install ffmpeg
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man b
rew`).
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
evans          iguana          jenkins-cli    kbld           otel-cli       restish
==> New Casks
cloudflare-warp@beta      font-special-gothic-expanded-one
deskttime                 sokim
font-source-han-code-jp   soundanchor
font-special-gothic-condensed-one

You have 3 outdated formulae installed.

Warning: ffmpeg 7.1.1_1 is already installed and up-to-date.
To reinstall 7.1.1_1, run:
  brew reinstall ffmpeg
```

Installing brew install ffmpeg

```
Last login: Mon Mar  3 20:10:20 on ttys002
(base) Penda@M2-14-inch-Macbook-Pro ~ % ffprobe -version

ffprobe version 7.1.1 Copyright (c) 2007-2025 the FFmpeg developers
built with Apple clang version 16.0.0 (clang-1600.0.26.6)
configuration: --prefix=/opt/homebrew/Cellar/ffmpeg/7.1.1 --enable-sha
le-pthreads --enable-version3 --cc=clang --host-cflags= --host-ldflags=
classic' --enable-ffplay --enable-gnutls --enable-gpl --enable-libaom --
libaribb24 --enable-libbluray --enable-libdav1d --enable-libharfbuzz --e
jxl --enable-libmp3lame --enable-libopus --enable-librav1e --enable-lib
able-librubberband --enable-libsrt --enable-libssh --enable-libsvtav1 --enable-libtesseract --enable-libtheora --enable-libvidstab
...

(base) Penda@M2-14-inch-Macbook-Pro ~ % ffmpeg -version
ffmpeg version 7.1.1 Copyright (c) 2000-2025 the FFmpeg developers
built with Apple clang version 16.0.0 (clang-1600.0.26.6)
configuration: --prefix=/opt/homebrew/Cellar/ffmpeg/7.1.1_1 --enable-
able-pthreads --enable-version3 --cc=clang --host-cflags= --host-ldfl
d_classic' --enable-ffplay --enable-gnutls --enable-gpl --enable-liba
-libaribb24 --enable-libbluray --enable-libdav1d --enable-libharfbuzz
ibjxl --enable-libmp3lame --enable-libopus --enable-librav1e --enable
enable-librubberband --enable-libsrt --enable-libssh --enable-libsv
...

```

Verify with ffmpeg -version and ffprobe -version

```
In [1]: !pip install ffprobe-python ffmpeg-python
Requirement already satisfied: ffprobe-python in /Users/Penda/anaconda3/lib/python3.11/site-packages (1.0.3)
Requirement already satisfied: ffmpeg-python in /Users/Penda/anaconda3/lib/python3.11/site-packages (0.2.0)
Requirement already satisfied: future in /Users/Penda/anaconda3/lib/python3.11/site-packages (from ffmpeg-python) (0.18.3)
```

These libraries allow interaction with ffmpeg and ffprobe via Python.

```
In [2]: import subprocess

try:
    subprocess.run(["ffprobe", "-version"], check=True)
    print("ffprobe is working correctly!")
except FileNotFoundError:
    print("Error: ffprobe not found. Make sure it is installed and in your system's PATH.")
```

Error: ffprobe not found. Make sure it is installed and in your system's PATH.

The script attempts to check if ffprobe is correctly installed. If ffprobe is not found, it prints an error message indicating that it may not be in the system's PATH.

```
In [3]: import os
os.environ["PATH"] += os.pathsep + "/opt/homebrew/bin"
```

The PATH environment variable is modified to ensure ffmpeg and ffprobe are accessible.

Brief Analysis

The application is designed to analyze and standardize video files by ensuring they conform to a predefined set of technical specifications. It identifies problematic fields within the video and audio streams and applies corrective measures using FFmpeg filters.

This application's main objective is to check and adjust video files to meet certain technical specifications. Using the MP4 container and preserving the H.264 video and AAC audio codecs, the application makes sure that video files adhere to a standard format. It also makes sure that the audio channel configuration is set to stereo, manages video and audio bit rates within predetermined bounds, adjusts the resolution to 640x360, enforces a 16:9 aspect ratio, and checks for a steady frame rate of 25 FPS.

In order to process video files, the application first retrieves them from a specified directory. After that, it checks to see if the audio and video streams meet the necessary requirements. The program uses FFmpeg filters to process the files and modify characteristics like frame rate, resolution, and bit rate if any differences are detected. Following the application of the corrections, the output is produced in a formatted state that complies with the necessary requirements.

Python is used in conjunction with the FFmpeg and FFprobe libraries to implement the application. To guarantee correct operation, it starts with installing and configuring these tools. The video and audio streams are inspected using a core function called `find_problematic_fields`, which finds any violations from the necessary standards. FFmpeg is used to alter the video stream by changing its frame rate, resolution, and bit rate if any problems are found. In a similar

manner, the audio parameters are adjusted as needed to guarantee adherence to codec and channel layout specifications. The application performs a final check after the changes are performed to ensure that the processed video now satisfies the required standards.

Brief Description

A video format, sometimes referred to as a container, is a type of file format that contains information, audio, video, and subtitles all in one. It controls the storage and playback of these many media components. MP4, AVI, MKV, and MOV are a few popular video containers; each offers varying degrees of compression, compatibility, and support for multiple codecs.

Video files can be compressed and decompressed using a video codec, which lowers their size without sacrificing visual quality. Video files would be too big for effective streaming and storage if they weren't compressed. H.264, which is frequently used for streaming and playback, and H.265 (HEVC), which offers superior compression while maintaining quality, are examples of popular video codecs.

Similar to a video codec, an audio codec is used for sound compression and decompression. It has an impact on an audio track's overall sound quality and file size. MP3 is still a commonly accepted standard for compressed audio, but AAC (Advanced Audio Codec) is one of the most popular audio codecs, providing effective compression with high-quality output.

The number of individual frames shown in a video every second is referred to as the frame rate. It affects the video's motion smoothness and is measured in frames per second (FPS). While 25 frames per second is the norm for PAL television broadcasts and 30 frames per second is the norm for NTSC television and web videos, a lower frame rate, such as 24 frames per second, is sometimes utilised in movies to create a theatrical atmosphere.

The proportionate relationship between a video frame's width and height is referred to as its aspect ratio. It affects how a video looks on various screens. While older televisions and some archival content use a 4:3 aspect ratio, the majority of current screens, including HDTVs and internet video platforms, use a 16:9 aspect ratio.

The number of pixels in a video frame, or resolution, dictates how clear and detailed the image is. Typically, it is represented in pixels as width × height. High clarity, for instance, is provided by a resolution of 1920 x 1080, or Full HD, but a smaller resolution, like 640 x 360, produces a smaller file size but worse image quality.

Megabits per second (Mbps) is the common unit of measurement for video bit rate, which is the amount of data processed each second in a video file. Better video quality is possible with a greater bit rate, but the file size also grows. For typical streaming, a video with a bit rate of 2 to 5 Mbps is seen to strike a decent compromise between efficiency and quality.

Conversely, audio bit rate, which is commonly expressed in kilobits per second (kbps), quantifies the data rate of an audio stream. While lower bit rates might result in apparent compression artefacts, particularly in content with a lot of music and dialogue, higher bit rates—like 256 kbps—offer better audio quality.

The quantity of distinct sound sources in an audio track is referred to as audio channels. There is only one channel in a mono audio file, therefore all of the speakers play the same sound. The two channels of stereo audio enable directional sound, making for a more engaging listening experience. More sophisticated configurations, like 5.1 or 7.1 surround sound, which are frequently used in home theatres and movie theatres, divide audio among several speakers to produce a three-dimensional soundscape.

<https://hub.labs.coursera.org:443/connect/sharedqfkwhzgw?forceRefresh=false&isLabVersioning=true>