

Part 1

HEVC: High Efficiency Video Coding (H.265)

Presented by: Phong Nguyen



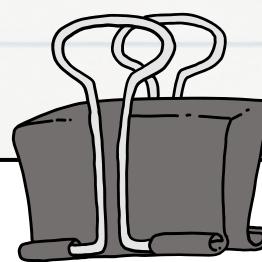
Skype

Phongsolo.Nguyen

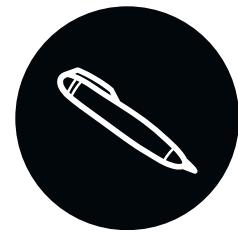
Overview

1. HEVC là gì?
2. Sự khác biệt của H.265 so với các chuẩn khác
3. HEVC có hiệu quả như thế nào
4. Mô phỏng với PYTHON
5. TLTK

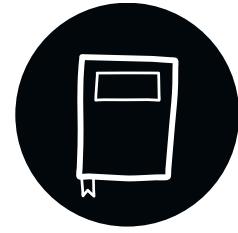
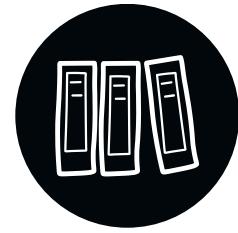




HEVC là gì?



- HEVC (High Efficiency Video Coding), còn được gọi là H.265, là một tiêu chuẩn nén video tiên tiến được phát triển để cải thiện hiệu suất so với chuẩn H.264/AVC. HEVC cung cấp chất lượng video cao hơn với cùng mức bitrate hoặc giảm kích thước file video đáng kể trong khi vẫn giữ nguyên chất lượng. Điều này làm cho HEVC trở thành lựa chọn phù hợp cho việc truyền tải video độ phân giải cao như 4K và 8K, đặc biệt trong các ứng dụng có băng thông hạn chế.
- HEVC/H.265 được phát triển bởi hai tổ chức lớn trong ngành công nghệ truyền thông là ITU-T (Liên minh Viễn thông Quốc tế – mảng Tiêu chuẩn hóa) và ISO/IEC (Tổ chức Tiêu chuẩn hóa Quốc tế cùng với Ủy ban Kỹ thuật Điện Quốc tế). Cụ thể, nhóm chuyên gia ITU-T Video Coding Experts Group (VCEG) và ISO/IEC (Moving Picture Experts Group -MPEG) đã hợp tác để tạo ra tiêu chuẩn này.
- Nhóm VCEG của ITU-T và MPEG của ISO/IEC trước đây cũng đã từng hợp tác để phát triển H.264/AVC, một tiêu chuẩn nén video phổ biến đang dùng hiện nay.



Sự khác biệt của H.265 so với các chuẩn khác là gì?

1



1. Hiệu quả nén cao hơn

2



2. Hỗ trợ độ phân giải
cao hơn

3



3. Các kỹ thuật mã
hóa tiên tiến

4



4. Kiểm soát bitrate và
truyền tải qua băng
thông giới hạn

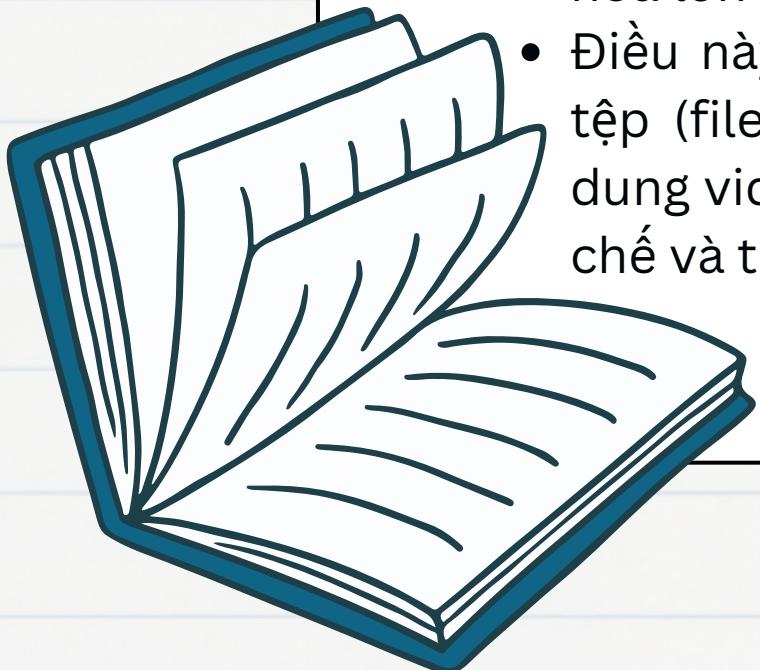
H.265/HEVC khác biệt so với các tiêu chuẩn nén video trước đó như H.264 (AVC) ở một số điểm quan trọng

1. Hiệu quả nén cao hơn

- H.265 cung cấp hiệu quả nén tốt hơn khoảng 50% so với H.264, nghĩa là nó có thể mang lại chất lượng tương đương với H.264 ở mức bitrate chỉ bằng một nửa. Hiệu quả này đạt được nhờ các kỹ thuật tiên tiến như khói mã hóa lớn hơn và cải tiến bù chuyển động.
- Điều này giúp H.265 giảm đáng kể kích thước tệp (file), rất hữu ích cho việc truyền tải nội dung video chất lượng cao qua băng thông hạn chế và tiết kiệm không gian lưu trữ.

2. Hỗ trợ độ phân giải cao hơn

- H.265 được thiết kế dành cho nội dung UHD (Ultra High Definition) và có thể xử lý độ phân giải lên đến 8K (7680×4320 pixel). Điều này khiến H.265 trở nên lý tưởng cho các nội dung video chất lượng cao với độ phân giải như 4K và 8K, trong khi H.264 chủ yếu tối ưu cho nội dung Full-HD.
- Các tiêu chuẩn trước đó H.264 và MPEG-2, chủ yếu hỗ trợ độ phân giải HD và SD (Standard Definition).



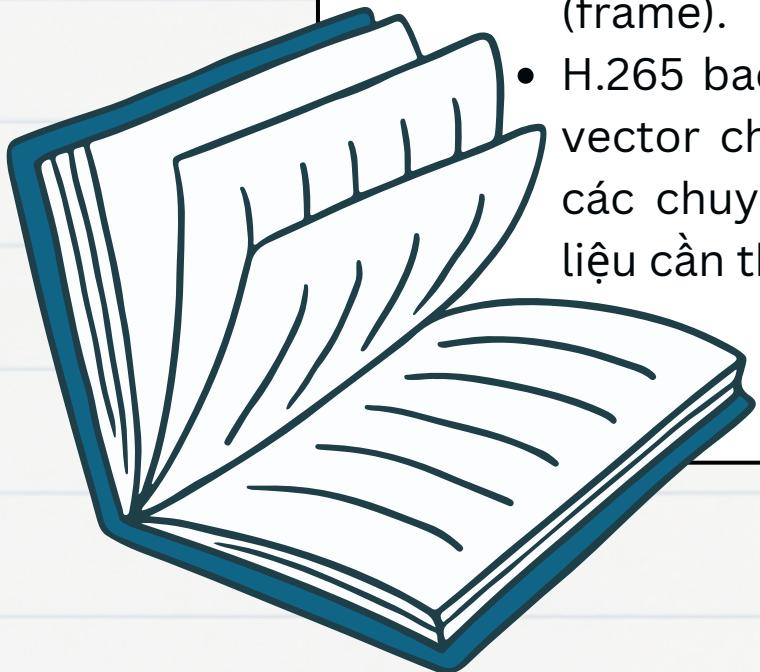
H.265/HEVC khác biệt so với các tiêu chuẩn nén video trước đó như H.264 (AVC) ở một số điểm quan trọng

3. Các kỹ thuật mã hóa tiên tiến

- H.265 giới thiệu các đơn vị mã hóa khung (CTU) lớn hơn, có thể lên tới 64x64 pixel, so với các macroblock nhỏ hơn của H.264 (tối đa 16x16). Sự linh hoạt này cho phép H.265 xử lý tốt hơn các khu vực chi tiết và mượt mà trong hình ảnh, thích ứng với độ phức tạp của các phần khác nhau trong khung hình (frame).
- H.265 bao gồm ước lượng chuyển động và dự đoán vector chuyển động cải tiến, giúp nó xử lý tốt hơn các chuyển động trong khung hình, giảm lượng dữ liệu cần thiết để biểu diễn chuyển động.

4. Kiểm soát bitrate và truyền tải qua băng thông giới hạn

- H.265 cho phép kiểm soát bitrate chi tiết hơn và mã hóa bitrate thích ứng, điều này giúp nâng cao chất lượng video trong các điều kiện băng thông giới hạn.
 - Điều này đặc biệt hữu ích cho các công nghệ truyền phát thích ứng, vd như streaming, nơi chất lượng video điều chỉnh theo thời gian thực dựa trên điều kiện mạng, mang lại trải nghiệm truyền tải mượt mà hơn.



5. Độ phức tạp và yêu cầu xử lý

- H.265 phức tạp hơn so với các tiêu chuẩn trước. Việc mã hóa và giải mã video H.265 yêu cầu cần bộ xử lý mạnh hơn, điều này có thể gây khó khăn cho một số thiết bị cũ hoặc hiệu năng phần cứng thấp.
- Tuy nhiên, hỗ trợ phần cứng và phần mềm cho H.265 hiện đã phổ biến hơn, với nhiều thiết bị hiện đại, GPU, và ứng dụng phần mềm cung cấp khả năng mã hóa (coder) và giải mã (decoder) H.265.

HEVC có hiệu quả như thế nào?

i).

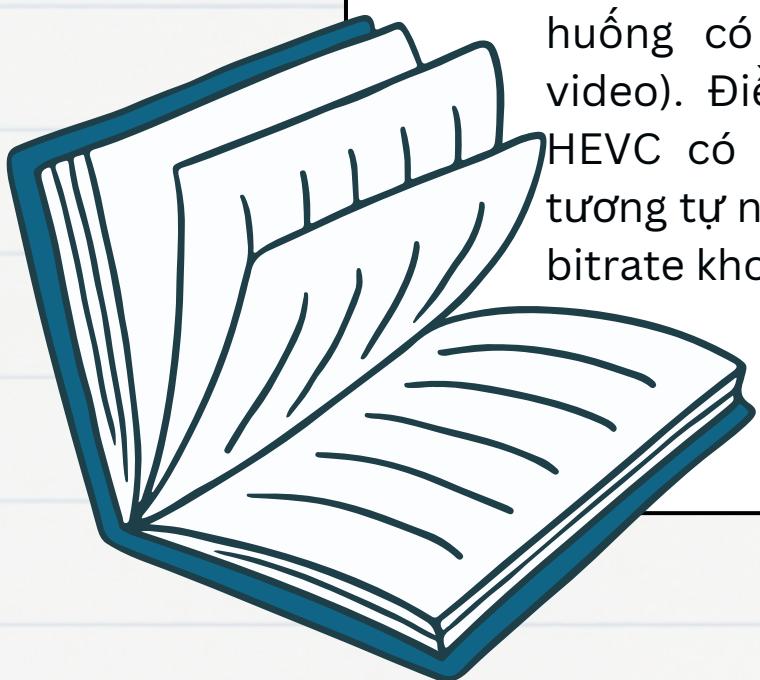
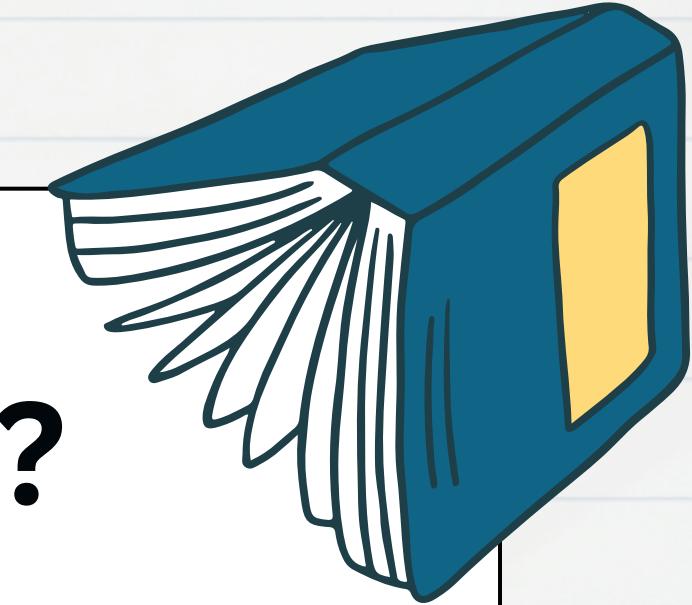
Bin Li (2011), so sánh giữa H.264/AVC và HEVC. Mô hình thử nghiệm HEVC ("HM") và mô hình thử nghiệm H.264 ("JM"). Trung bình, HEVC vượt trội hơn H.264 39% đối với các tình huống truy cập ngẫu nhiên (ví dụ: phát quảng bá) và 44% đối với các tình huống có độ trễ thấp (ví dụ: gọi video). Điều này có nghĩa là codec HEVC có thể đạt được chất lượng tương tự như H.264 với mức tiết kiệm bitrate khoảng 39-44%.

ii).

Ohm (2012): Khả năng nén của một số thê hệ tiêu chuẩn mã hóa video được so sánh bằng tỷ lệ tín hiệu trên nhiễu đỉnh (PSNR) và kết quả thử nghiệm đánh giá chủ quan. Một phương pháp tiếp cận thống nhất được áp dụng để phân tích các thiết kế, bao gồm Video H.262/MPEG-2, H.263, MPEG-4 Visual, Mã hóa video nâng cao H.264/MPEG-4 (AVC) và Mã hóa video hiệu suất cao (HEVC).

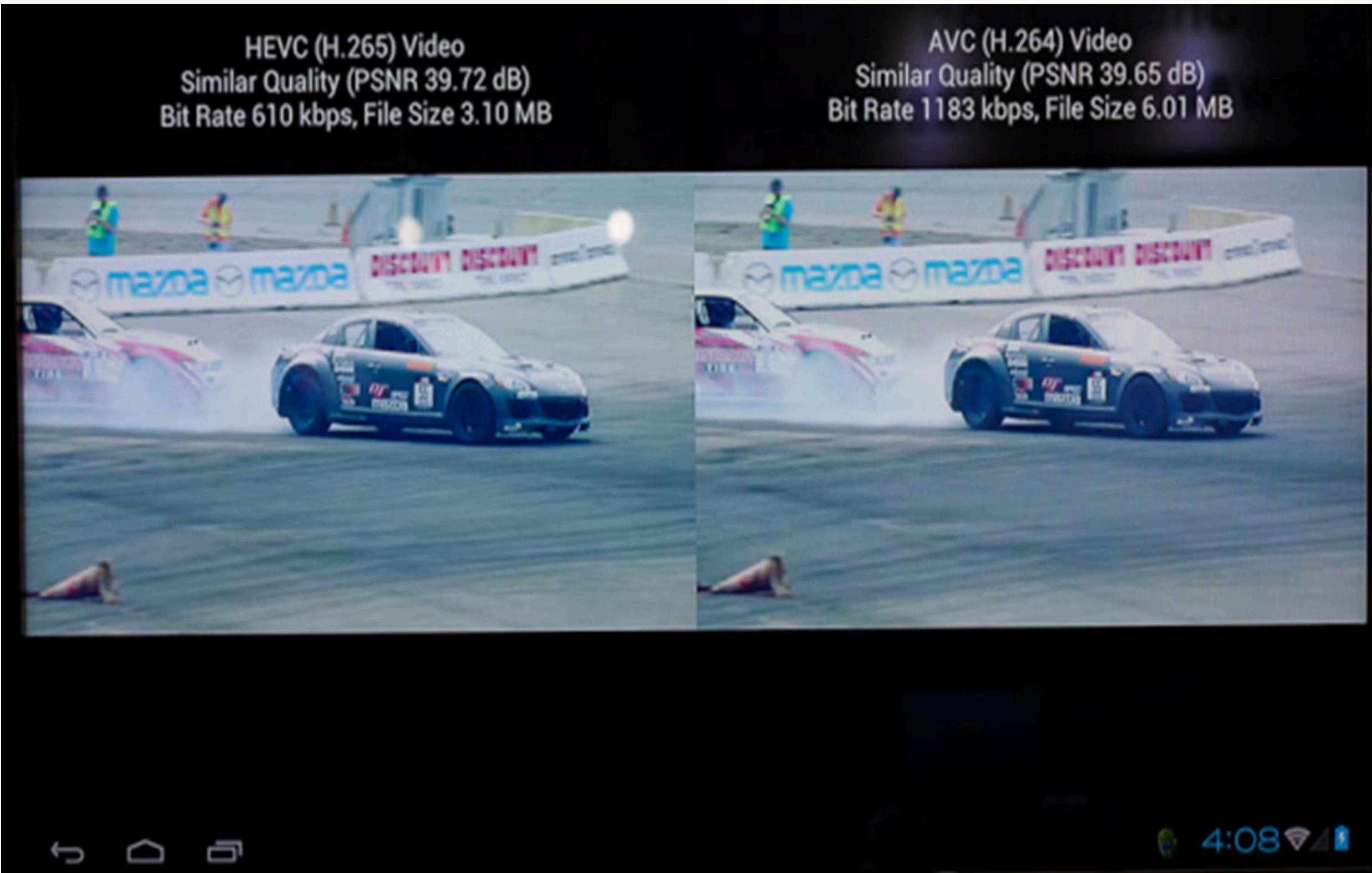
iii).

Kết quả của các thử nghiệm phương pháp đánh giá chủ quan đối với chuỗi WVGA và HD cho thấy bộ mã hóa HEVC có thể đạt được chất lượng tái tạo chủ quan tương đương với bộ mã hóa tuân thủ H.264/MPEG-4 AVC khi sử dụng tốc độ bit thấp hơn trung bình khoảng 50%.



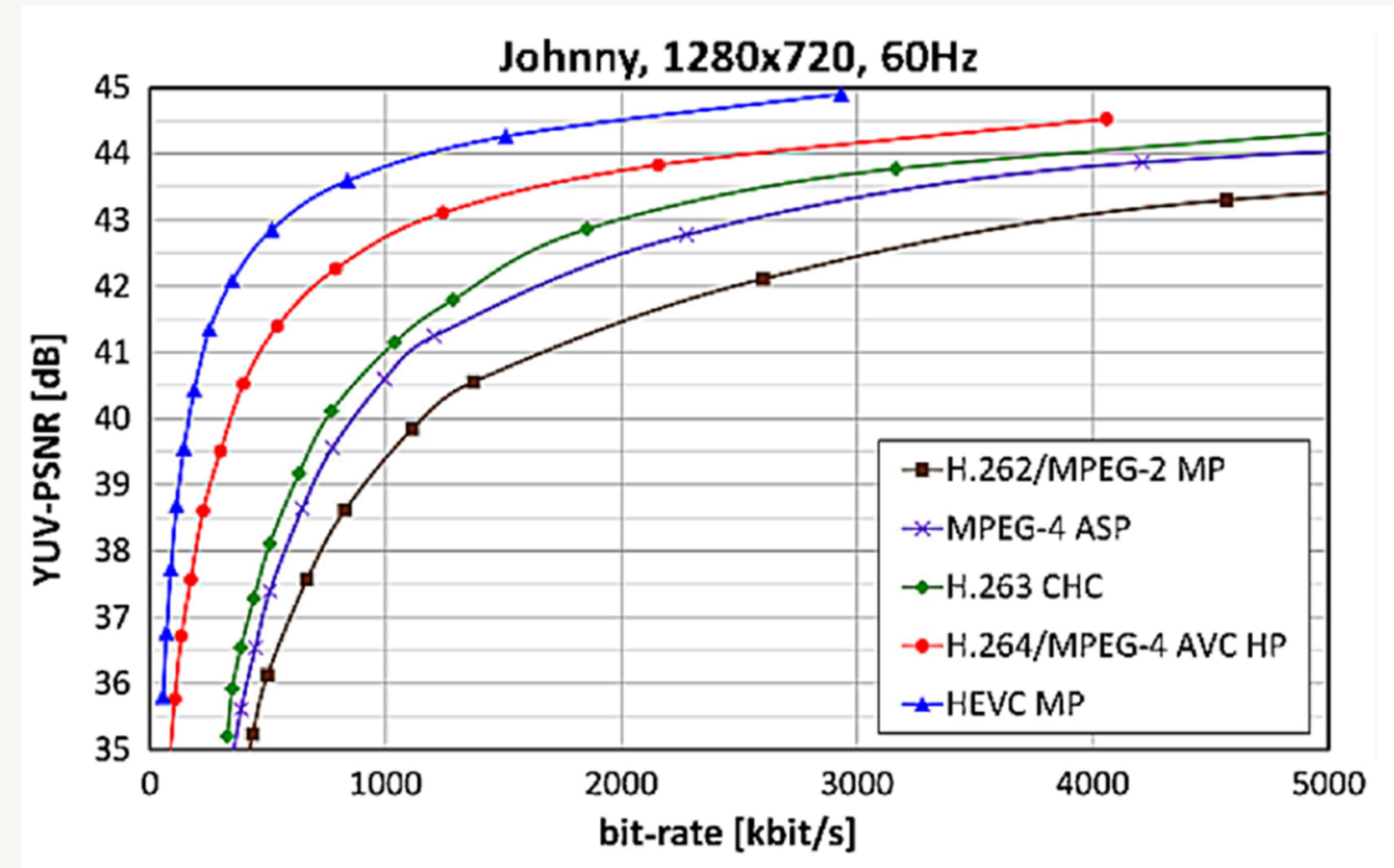
01

So sánh H.265 và H.264 cùng mức chất lượng (PSNR=39.72 dB. Tốc độ bit rate của H.265 là 610 kbps, vs H.264 là 1183 kbps. Điều này làm giảm kích thước file đáng kể (tiết kiệm 50%) lưu trữ



02

Hình này minh họa kết quả thử nghiệm mối liên hệ giữa PSNR so với tốc độ bit với codec: MPEG-2; H.263, H.264/AVC và H.265. Với cùng 1 bit rate, đường màu xanh dương (blue) của H.265 có PSNR cao nhất, tương ứng chất lượng hình ảnh tốt nhất



Đặc điểm	H.264 (AVC)	H.265 (HEVC)	MPEG-2
Hiệu quả nén	Trung bình	Cao (tốt hơn đến 50%)	Thấp
Độ phân giải tối đa	4K	8K	SD/HD
Kích thước khối mã hóa	16x16 (macroblock)	64x64 (CTU)	8x8/16x16 blocks
Kiểm soát tốc độ bit	Biến đổi nhưng kém hiệu quả	Biến đổi nâng cao	Hạn chế
Yêu cầu xử lý	Trung bình	Cao	Thấp

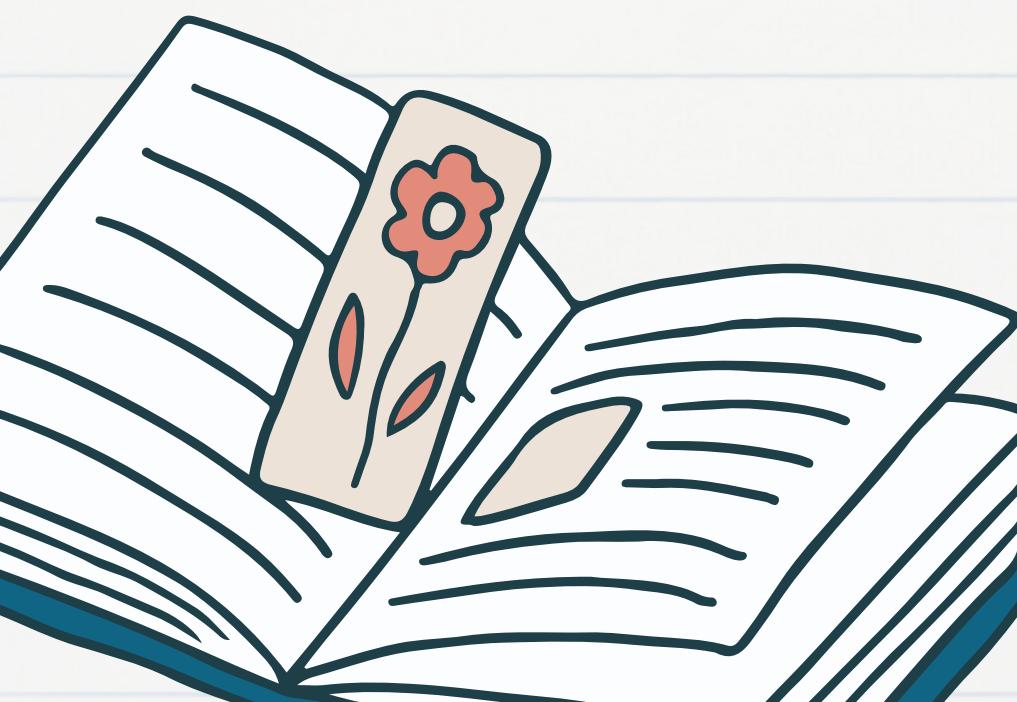
Ứng dụng	Mô tả	Bitrate khuyến cáo	Tài liệu tham khảo
Truyền hình 4K/8K	Truyền tải video độ phân giải cao qua mạng cáp, vệ tinh hoặc truyền hình cáp.	15-25 Mbps cho 4K, 50-80 Mbps cho 8K	ITU-T Recommendation H.265 (HEVC), ETSI TR 101 290, EBU Tech 3335
Live Streaming (4K)	Truyền phát trực tiếp video chất lượng cao qua nền tảng như YouTube, Twitch.	20-25 Mbps cho 4K	ITU-T H.265, EBU Tech 3335, ETSI TR 101 290
Video trên di động (4K)	Cung cấp video chất lượng cao trên điện thoại di động, tiết kiệm băng thông.	4-6 Mbps cho 4K	ITU-T Recommendation H.265 (HEVC), ETSI TR 101 290
Video Conferencing (HD/4K)	Cung cấp cuộc gọi video chất lượng cao cho các cuộc họp trực tuyến.	2-6 Mbps cho HD, 8-12 Mbps cho 4K	ITU-T H.265, EBU Tech 3335, ETSI TR 101 290

Mô phỏng với PYTHON

Cài đặt FFmpeg: Hỗ trợ mã hóa và giải mã H.265

B1

- pip install scikit-image
- pip install ffmpeg-python



```
C:\ Command Prompt
C:\Users\PNH>pip install ffmpeg-python
Collecting ffmpeg-python
  Downloading ffmpeg_python-0.2.0-py3-none-any.whl.metadata (1.7 kB)
Collecting future (from ffmpeg-python)
  Downloading future-1.0.0-py3-none-any.whl.metadata (4.0 kB)
Downloading ffmpeg_python-0.2.0-py3-none-any.whl (25 kB)
Downloading future-1.0.0-py3-none-any.whl (491 kB)
Installing collected packages: future, ffmpeg-python
Successfully installed ffmpeg-python-0.2.0 future-1.0.0

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\PNH>
```

Cài đặt FFmpeg

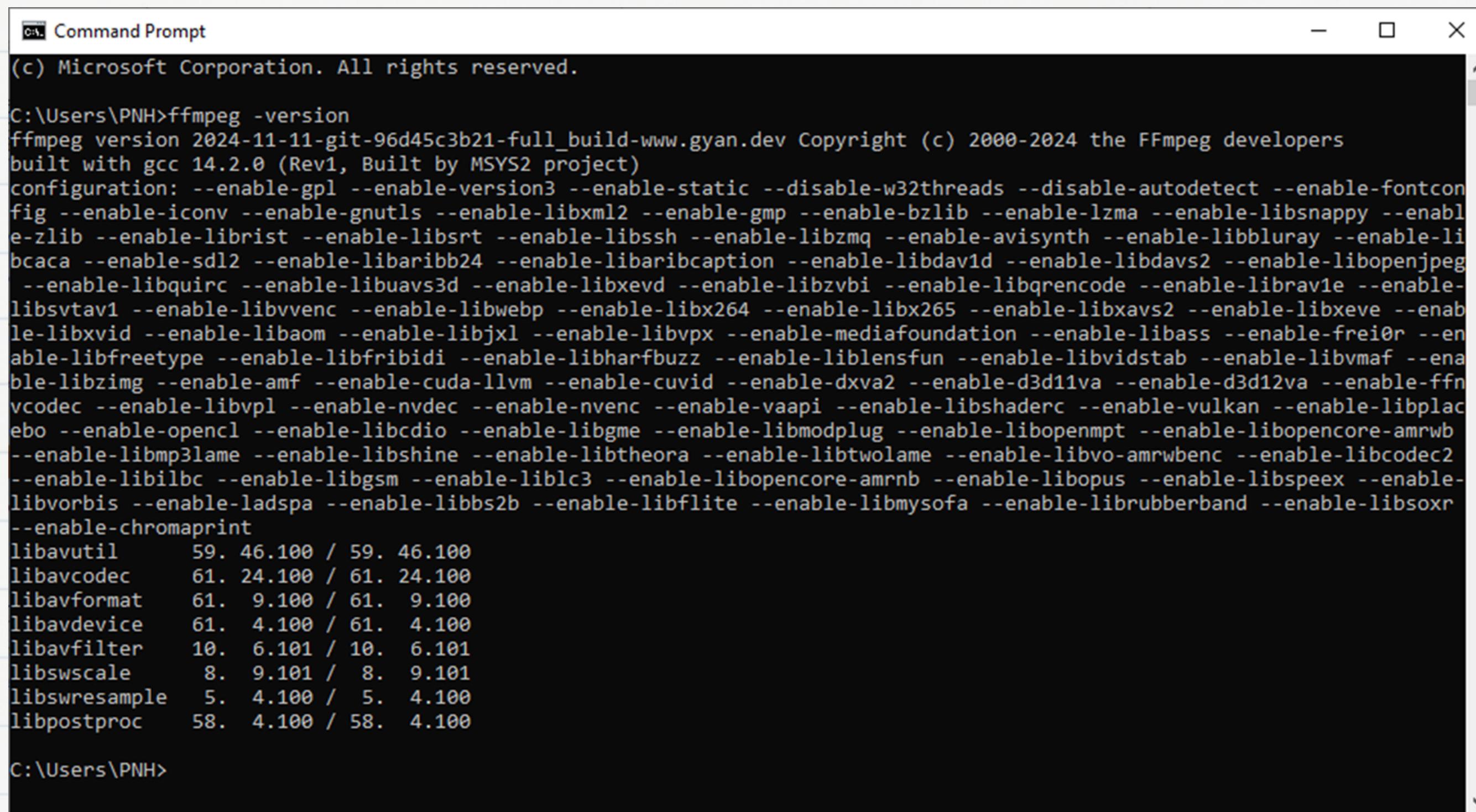
B2

ModuleNotFoundError: No module named 'ffmpeg'
Xem hướng dẫn cài đặt ffmpeg tại đây:

<https://www.wikihow.com/Install-FFmpeg-on-Windows>
<https://wwwffmpeg.org/download.html>
<https://www.gyan.dev/ffmpeg/builds/>

C:\ffmpeg
setx /m PATH "C:\ffmpeg\bin;%PATH%"
ffmpeg -version

Cài đặt FFmpeg



```
Command Prompt
(c) Microsoft Corporation. All rights reserved.

C:\Users\PNH>ffmpeg -version
ffmpeg version 2024-11-11-git-96d45c3b21-full_build-www.gyan.dev Copyright (c) 2000-2024 the FFmpeg developers
built with gcc 14.2.0 (Rev1, Built by MSYS2 project)
configuration: --enable-gpl --enable-version3 --enable-static --disable-w32threads --disable-autodetect --enable-fontconfig --enable-iconv --enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-lzma --enable-libsnappy --enable-zlib --enable-librist --enable-libsrt --enable-libssh --enable-libzmq --enable-avisynth --enable-libbluray --enable-libcaca --enable-sdl2 --enable-libaribb24 --enable-libaribcaption --enable-libdav1d --enable-libdavs2 --enable-libopenjpeg --enable-libquirc --enable-libuavs3d --enable-libxevd --enable-libzvbi --enable-libqrencode --enable-librav1e --enable-libsrtav1 --enable-libvvenc --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavcs2 --enable-libxeve --enable-libxvid --enable-libaom --enable-libjxl --enable-libvpx --enable-mediaproxy --enable-libass --enable-frei0r --enable-libfreetype --enable-libfribidi --enable-libharfbuzz --enable-liblensfun --enable-libvidstab --enable-libvmaf --enable-libzimg --enable-amf --enable-cuda-llm --enable-cuvid --enable-dxva2 --enable-d3d11va --enable-d3d12va --enable-fnvcodec --enable-libvpl --enable-nvdec --enable-nvenc --enable-vaapi --enable-libshaderc --enable-vulkan --enable-libplacebo --enable-opencl --enable-libcdio --enable-libgme --enable-libmodplug --enable-libopenmpt --enable-libopencore-amrwb --enable-libmp3lame --enable-libshine --enable-libtheora --enable-libtwolame --enable-libvo-amrwbenc --enable-libcodec2 --enable-libilbc --enable-libgsm --enable-liblc3 --enable-libopencore-amrnb --enable-libopus --enable-libspeex --enable-libvorbis --enable-ladspa --enable-libbs2b --enable-libflite --enable-libmysofa --enable-librubberband --enable-libsoxr --enable-chromaprint
libavutil      59. 46.100 / 59. 46.100
libavcodec     61. 24.100 / 61. 24.100
libavformat    61.  9.100 / 61.  9.100
libavdevice    61.  4.100 / 61.  4.100
libavfilter     10.  6.101 / 10.  6.101
libswscale       8.  9.101 /  8.  9.101
libswresample    5.  4.100 /  5.  4.100
libpostproc     58.  4.100 / 58.  4.100

C:\Users\PNH>
```

Mô phỏng với PYTHON

(Visual Studio Code)

The screenshot shows a Visual Studio Code interface with the following details:

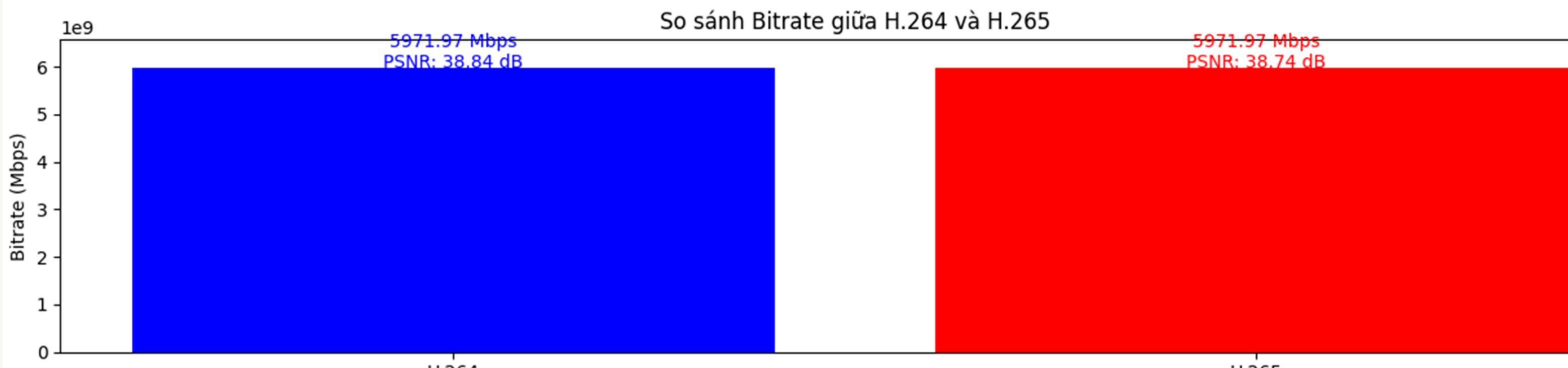
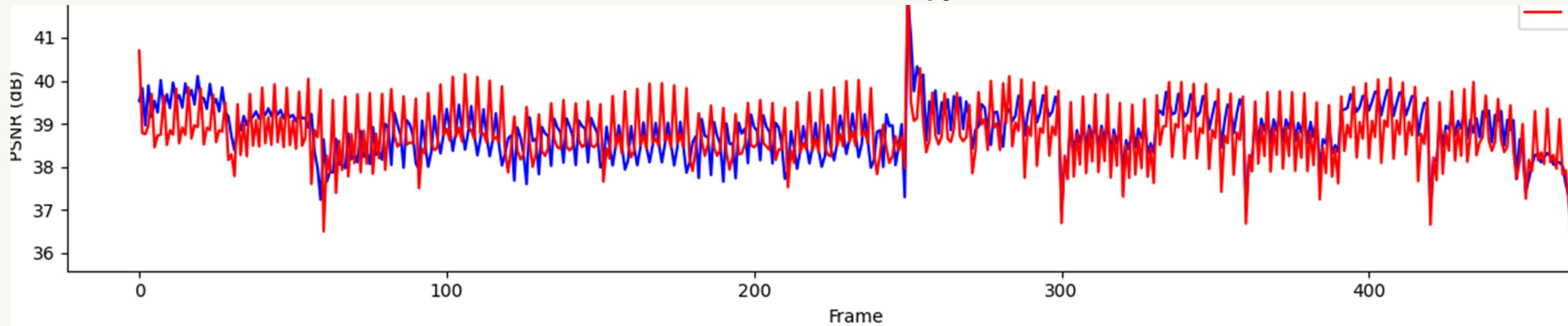
- File Explorer:** Shows a folder named "PYTHON HEVC" containing files: compressed_h264.mp4, compressed_h265.mp4, HEVCvsH264.py, HEVCvsH264v2.0.py, PSNRvsBitrate.py, PSNRvsBitrateV2.0.py, videotest.mp4, and videotest02.mp4.
- Code Editor:** Displays the content of the file "HEVCvsH264v2.0.py". The code uses the `ffmpy` library to capture frames from original and compressed videos, calculate PSNR, and release the video capture objects.
- Terminal:** Shows the output of the script execution, including frame statistics and final PSNR values.
- Status Bar:** Provides information about the current file (HEVCvsH264v2.0.py), encoding (Python - Video Co...), and system status (Windows_NT x64 10.0.19045).
- Information Dialog:** A modal dialog titled "Visual Studio Code" displays the application's version and build information.

```
3 import ffmpy
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from skimage.metrics import peak_signal_noise_ratio as psnr
7
8 # Function to calculate PSNR between the original and compressed videos
9 def calculate_psnr(original_video, compressed_video):
10     cap_orig = cv2.VideoCapture(original_video)
11     cap_comp = cv2.VideoCapture(compressed_video)
12
13     psnr_values = []
14     frame_count = 0
15
16     while True:
17         ret_orig, frame_orig = cap_orig.read()
18         ret_comp, frame_comp = cap_comp.read()
19
20         if not ret_orig or not ret_comp:
21             break
22
23         psnr_value = psnr(frame_orig, frame_comp)
24         psnr_values.append(psnr_value)
25         frame_count += 1
26
27     cap_orig.release()
```

Version: 1.95.1 (user setup)
Commit: 65edc4939843c90c34d61f4ce11704f09d3e5cb6
Date: 2024-10-31T05:14:54.222Z (1 wk ago)
Electron: 32.2.1
ElectronBuildId: 10427718
Chromium: 128.0.6613.186
Node.js: 20.18.0
V8: 12.8.374.38-electron.0
OS: Windows_NT x64 10.0.19045

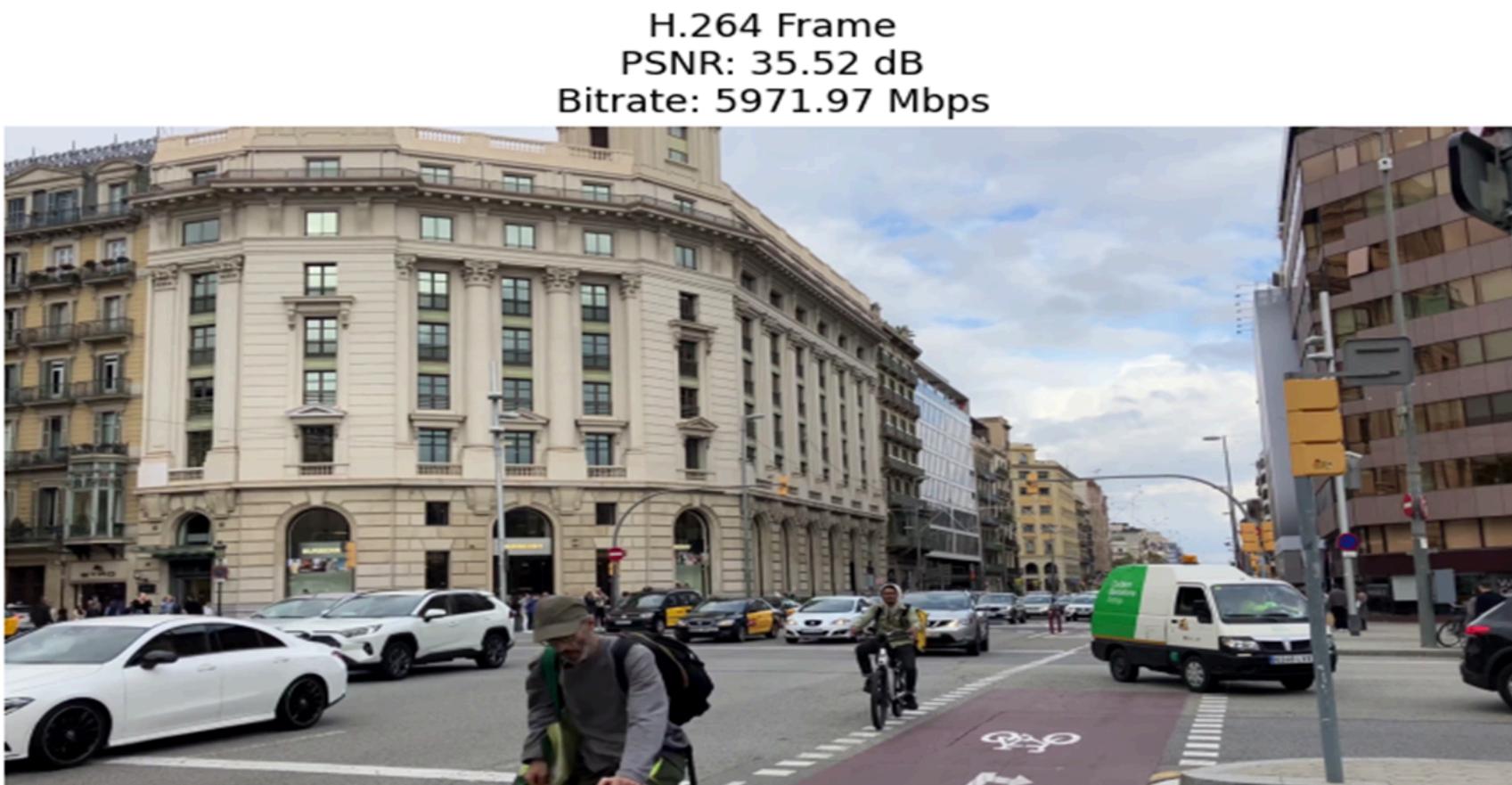
Minh họa về PSNR và Bitrate cho H.264 và H.265

#plot Fig.1 slide Mo phong voi PYTHON
HEVCvsH264v2.0.py



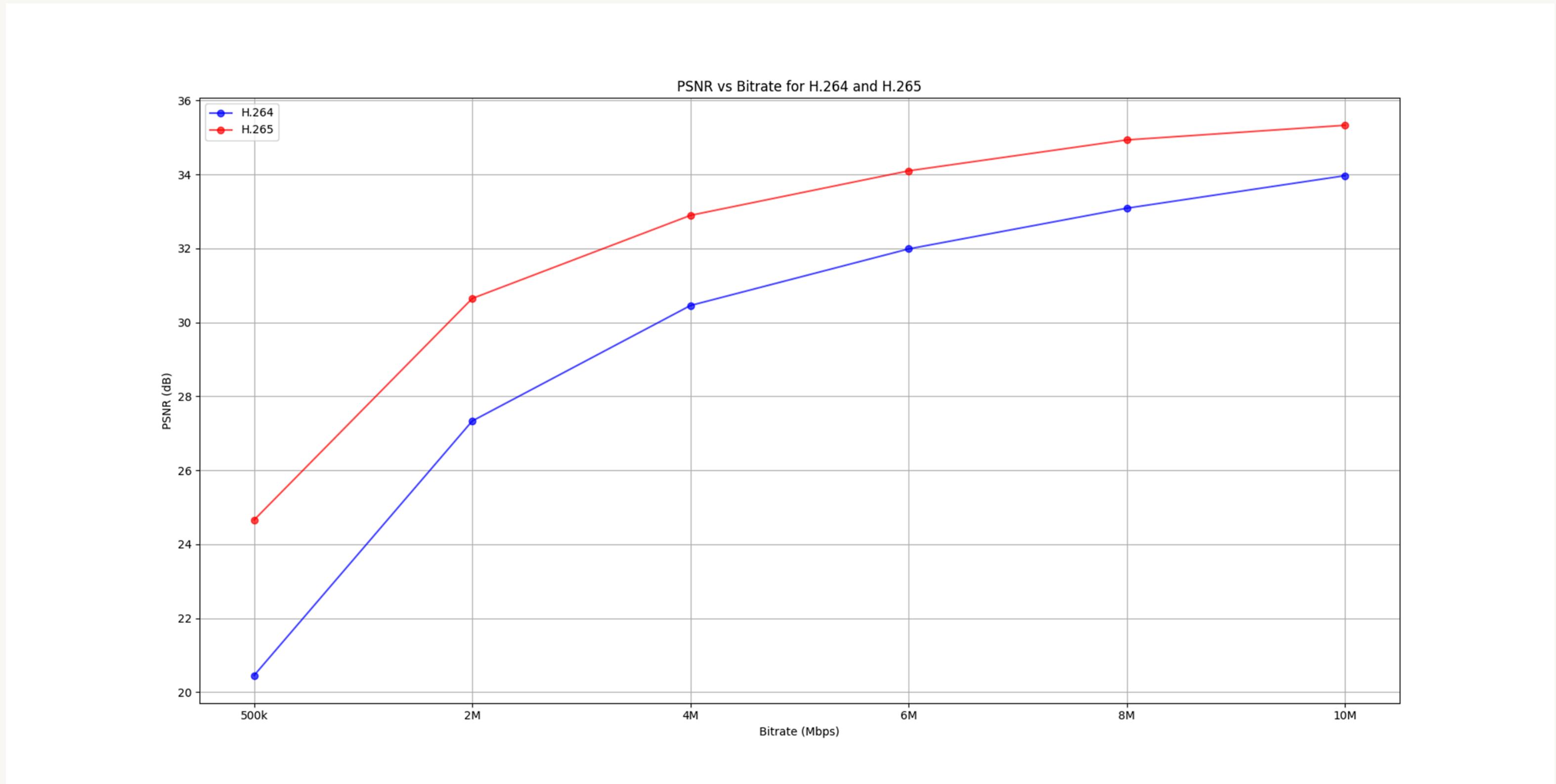
Mô phỏng với PYTHON

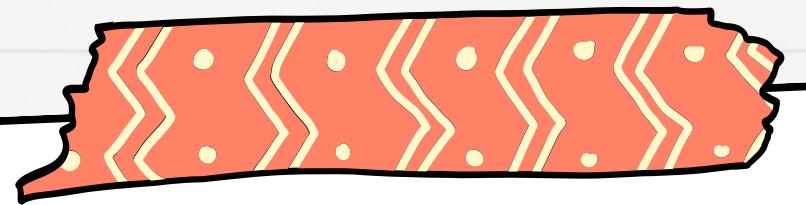
(PSNRvsBitrate.py) Fig.2



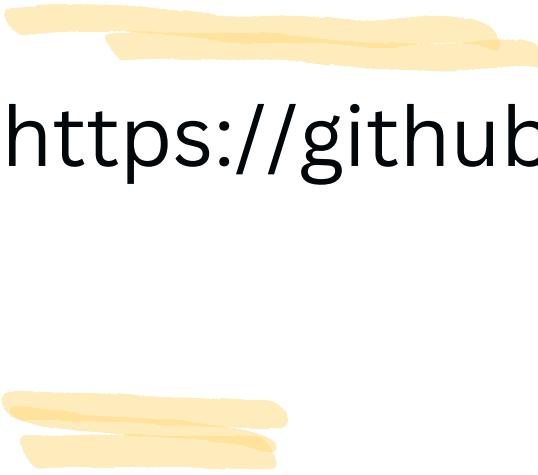
#plot Fig03 slide Mo phong voi PYTHON

So sánh H.264 vs H.265, cùng tốc độ bitrate, H.265 có giá trị PSNR cao hơn, tương ứng chất lượng hình ảnh tốt hơn H.264 File: PSNRvsBitrateV4.0.py





Download: Code and slide at GitHub



<https://github.com/phongsolo/Code-Learning-2024>

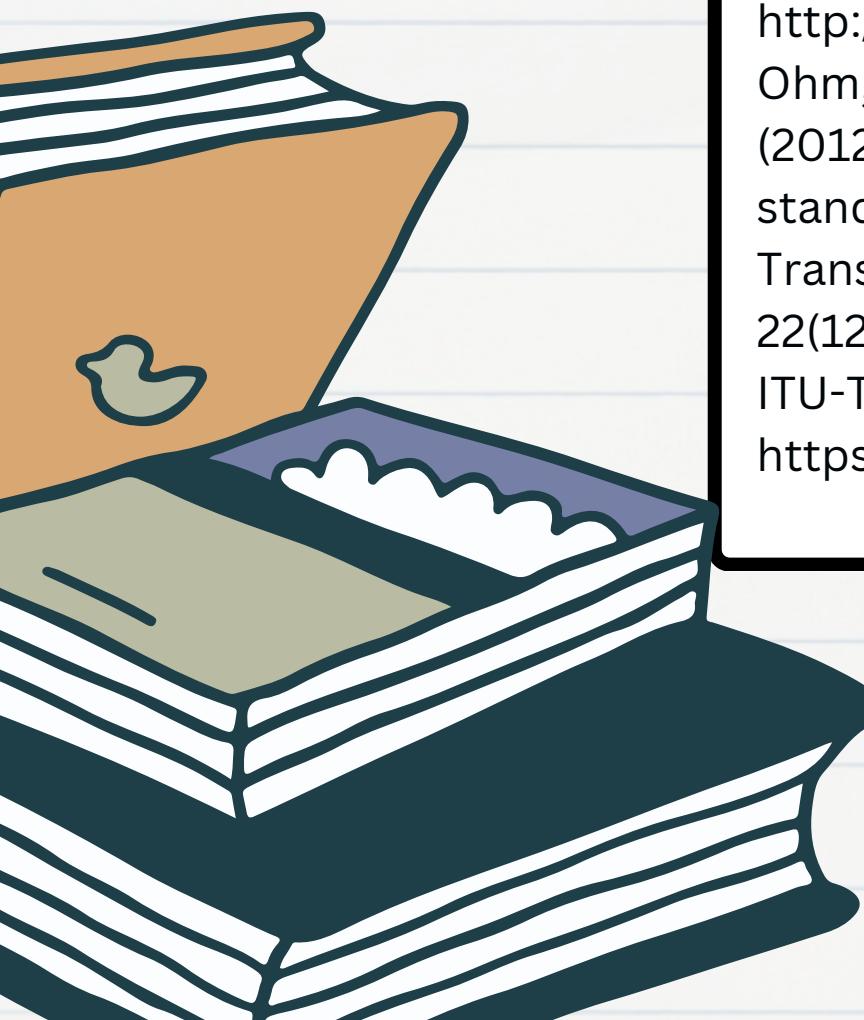


Versatile Video Coding (VVC), also known as H.266, [1] ISO/IEC 23090-3, [2] and MPEG-I Part 3, is a [video compression standard](#) finalized on 6 July 2020, by the Joint Video Experts Team (JVET) [3] of the [VCEG](#) working group of [ITU-T Study Group 16](#) and the [MPEG](#) working group of [ISO/IEC JTC 1/SC 29](#).

Origin	Type	ISO / IEC	ITU-T	Nicknames
MPEG-1	Video	11172-2	H.261	MPEG-1 video
MPEG-1	Audio	11172-3		MPEG audio layer 1
MPEG-2	Video	13818-2	H.262	MPEG-2 video
MPEG-2	Audio	13818-3		Layer 2: MPEG audio layer 2 Layer 3: MP3
MPEG-2	Audio	13818-7		AAC (Advanced Audio Coding)
Dolby Digital	Audio			AC-3 (Audio Coding 3)
MPEG-4	Video	14496-2	H.263	DivX, Xvid (codecs)
MPEG-4	Audio	14496-3		HE-AAC, EAAC (High Efficiency, Enhanced AAC)
MPEG-4	Video	14496-10	H.264	AVC (Advanced Video Coding)
MPEG-H	Video	23008-2	H.265	HEVC (High Efficiency Video Coding)
Dolby Digital	Audio			AC-4
MPEG-I	Video	23090-3	H.266	VVC (Versatile Video Coding)

TLTK

01



http://www.telcogroup.ru/files/materials-pdf/High_Efficiency_Video_Coding_H265.pdf
<http://www.vcodex.com/h265/72-how-efficient-is-hevc.html>
http://phenix.intevry.fr/jct/doc_end_user/current_document.php?id=3656
<http://www.hardwaremania.com/forum/hm/haberler/h-265-codec-8k-7680x4320-piksel-cozunurluk-mumkun-152235.html>
<http://mmediatech.blogspot.com/2012/03/qualcomm-shows-hevc.html>
<http://www.youtube.com/watch?v=pt-pZVKeoGg>
Ohm, J. R., Sullivan, G. J., Schwarz, H., Tan, T. K., & Wiegand, T. (2012). Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). *IEEE Transactions on circuits and systems for video technology*, 22(12), 1669-1684.
ITU-T Recommendations: ITU-T H.265 (V10) (07/2024)
<https://www.itu.int/itu-t/recommendations/rec.aspx?rec=H.265>

02



1.

IEEE:

- "High Efficiency Video Coding (HEVC) and Its Applications in Video Transmission" - IEEE Access, 2018.
- "Efficient Video Streaming Using HEVC for 4K and 8K Resolution" - IEEE Transactions on Broadcasting, 2017.
- "An Overview of HEVC and Its Potential in Streaming and Video Conferencing" - IEEE Communications Magazine, 2019.

2. **ITU (International Telecommunication Union):**

- **ITU-T Recommendation H.265:** Cung cấp thông tin chi tiết về các chuẩn HEVC và khuyến nghị bitrate cho các ứng dụng video.
- **ITU-T P.910:** Tiêu chuẩn khuyến nghị cho việc truyền tải video chất lượng cao qua mạng.

3. **EBU (European Broadcasting Union):**

- **EBU Tech 3335:** Tài liệu về khuyến nghị cho việc áp dụng HEVC trong truyền hình 4K và 8K.
- **EBU Tech 3307:** Hướng dẫn về nén video HEVC và yêu cầu bitrate cho truyền hình độ phân giải cao.

4. **ETSI (European Telecommunications Standards Institute):**

- **ETSI TR 101 290:** Báo cáo kỹ thuật về việc sử dụng HEVC trong truyền tải video và các khuyến nghị về bitrate cho các ứng dụng trực tuyến và truyền hình.