PSSOH 2022, DOI: https://doi.org/10.5281/zenodo.7096251

Video analysis using open-source FFmpeg tool and selection of codecs

Ana Gavrovska

University of Belgrade - School of Electrical Engineering,
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
e-mail addresses: anaga777@gmail.com, anaga777@etf.rs

ABSTRACT

This paper presents a brief overview of a free and open-source multimedia framework called FFmpeg with emphasis on video analysis and selection of codecs (CoDec - Coding-Decoding/Compression-Decompression). Filtering, trimming, transcoding, streaming, multiplexing, and other possibilities are available using this tool which is commonly applied for multimedia handling, telecommunications and video traffic monitoring, forensics and, generally, practical projects. Commands can be used in the command line interface with input and output files, selected actions and parameters. There are several available FFmpeg libraries for developers, where some of them enable implementation of relatively novel encoding/decoding solutions. Quality conditions may be set differently according to specific control modes. This cross-platform and open solution is beneficial for understanding both basic and complex approaches for video and multimedia manipulation, and is relevant for educational purposes. Regardless of the framework, codecs are evolving towards the new generation and some of the general trends in this domain are presented here. Finally, attention is given to some of the licensing considerations.

Keywords: open-source, ffmpeg, video, codecs, multimedia compression, presets

1 Introduction

For content production and distribution, it is important to cope with a large amount of media [1-4]. During the last decades, video has become dominant in digital communication and a large part of global telecommunication traffic. Video production, visual advertising, sharing platforms, video conference, unified communication and collaboration products, immersive media, have led to the need of efficient video delivery models. Video demand on the internet has grown significantly to 80% of all traffic in 2021 compared to 67% in 2016 according to Cisco report [5]. Video web services have skyrocketed in the period of Covid-19 pandemic [6]. Thus, new media technologies have caused revising issues of content delivery congestion.

Encoding and decoding video and audio streams, i.e. codecs, and other relevant tools for developers enable practical implementations [1-2, 7-8]. In addition to coding-decoding, usually compression-decompression tasks are performed to deal with bandwidth bottleneck and limitations of storage. Lossless compression approaches are developed having in mind retaining all information without losing media quality, while mostly used lossy ones make the trade-off, meaning transmission channel or storage space is less occupied with decent approximation of encoded data. Methods for editing, control, quality evaluation, encapsulation, and many other processes are related to coding and media handling. Different encoded data can be combined by multiplexing and data elements (streams) should coexist within container formats. This provides interoperability within ICT (Information and Communication Technology) systems that is essential. Benefits of IP (Internet Protocol) communications provided flexibility and scalability in media distribution [9, 10].

Free and open-source frameworks and projects are necessary for providing basics of understanding multimedia technologies that are becoming more complex and of wide-ranging interest among engineers, developers, and stakeholders. FFmpeg is one of the leading multimedia framework available for implementation of codecs, processing, streaming, multiplexing and similar media handling options and manipulations with high portability over different platforms, environments, and machine architectures [11, 12].

In this paper several main aspects related to modern codecs in general and FFmpeg usage are discussed. The paper is organized as follows. Brief description of popular formats and main FFmpeg framework characteristics can be found in Section II after the introduction. Basic tools for treating media are described, such as setting video quality according to predefined parameters and chosen codec. Some of the examples, which include preset options, are presented in Section III. Video streaming is part of our daily lives, and adoption of new standards and licensing efforts are also important issues to sum things up in Section IV. Finally, Section V is dedicated to conclusion.

2 Codec tools

A large number of specific techniques nowadays make a solution satisfying for media delivery. Committees, corporations, communities, and alliances design variety of standards and formats. For example, MPEG stands for Motion Picture Experts Group, and is dedicated to media coding and compression, transmission and formats. In video and multimedia technology field MPEGx and H26x standards have taken the central place over the years [1, 13].

In order to analyze some of the codec tools, FFmpeg (Fast Forward MPEG) is applied here. It represents one of the most powerful multimedia environments with options suitable for many users and developers working in different fields that include video and multimedia processing. It supports different formats from old-fashioned ones to novel solutions, where most video programs include this solution as a part of their processing pipeline [10, 11]. Cross-platform libraries for processing like OpenCV can also use the FFmpeg results as backend for recording, media converting and audio and video streaming, and this backend usage is not the rare case [14]. This practical project is free to use and one of the most deserving for this are developers Fabris Belard and Michael Niedermaier, and many other project participants.

2.1 FFmpeg tools and libraries

FFmpeg includes three basic tools, i.e. commands that can be used in working with media content: *ffmpeg*, *ffprobe* and *ffplay*. By using *ffmpeg* command, some of the popular codec can be applied or format conversion can be performed. On the other hand, the *ffplay* command serves to reproduce files in order to visualize the content. The professionals parse video content and monitor traffic quantities via *ffprobe* command. FFmpeg comes with libraries: *libavutil* (intended for simplifying programmable approaches and routines), *libavcodec* (containing encoders and decoders), *libavformat* (dedicated to multiplexing and demultiplexing), *libavdevice* (containing input and output devices for grabbing and rendering), *libavfilter* (holding filters for multimedia needs), *libswscale* (for scaling and conversion between color and pixel based formats), and *libswresample* (for resampling, rematrixing and sample format conversions). FFmpeg software may be useful in overcoming the need of special hardware in order to use main codecs and formats especially the ones for distribution over the Internet. The project has technical documentation, automated testing environment (fate), bug tracker, and wiki page. The main licence is GPL (GNU General Public License) or LGPL (GNU Lesser GPU) [11].

The commands are easily applied in command line (CLI - *Command Line Interface/Interpreter*) with chosen input and output files, selected actions and parameters. In Fig. 1 typical video processing that includes demultiplexing (demux) of input file, decoding, encoding, and multiplexing (mux) obtaining output file. The media content is kept in containers. Input and output files can be: mpeg, avi (audio video interleaved), m4v that is similar to mp4 (MPEG-4 Part 14), etc. Some of the codec examples are also given in Fig. 1 [11, 12]. Extending the framework is possible as well [15].

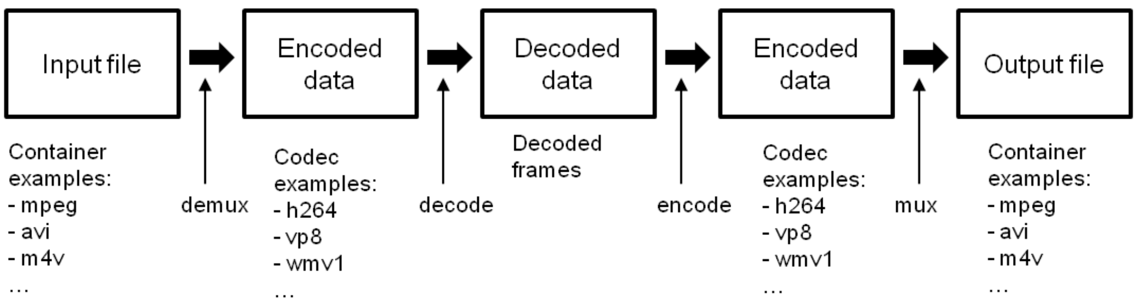

Figure 1. Typical video processing.

2.2 Video coding steps and standards

In video coding there are several main steps including: partitioning, prediction, data transformation and quantization, filtering improvements and entropy coding, as can be seen in Fig. 2. Each new standard brings novel advancements, like more partitions and larger blocks (superblocks), more intra and inter predictions, functions, filters and introducing AI (*Artificial Intelligence*) to coding and compression [6, 13, 15].

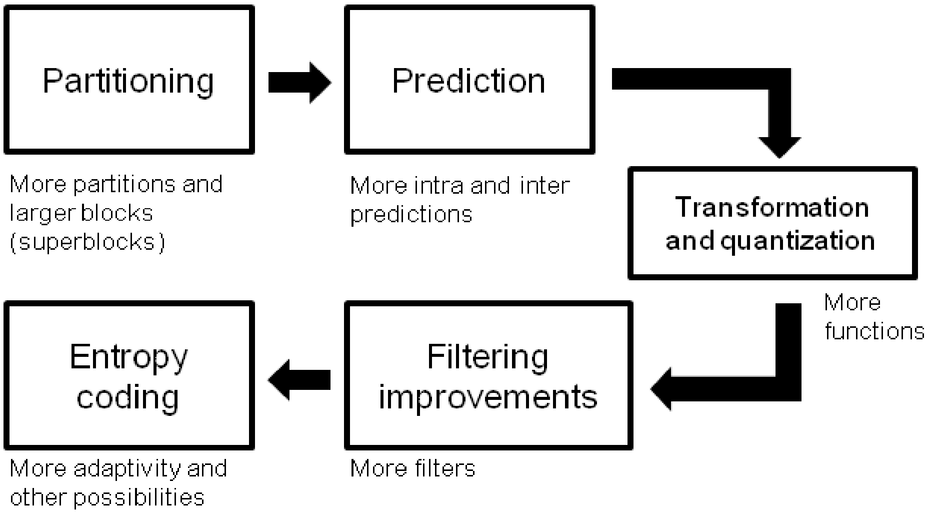

Figure 2. Illustration of main video coding steps and advancement.

The fundamentals in streaming are made by MPEG-2 standard that was popular in practical implementations like broadcasting. This practical usefulness has been continued by MPEG-4 by introducing additional enhancements. H.264 or Advance Video Coding (AVC) standard (MPEG-4 Part 10) is a well known standard from 2003 found by ITU-T and ISO/IEC. AVC is still in force for generic audiovisual services [16]. One of the most popular formats is still mp4 (MPEG-4 Part 14) representing digital multimedia container format for data encapsulation. A new standard is developed after H.264/AVC called H.265 or HEVC (High Efficiency Video Coding) or MPEG-H Part 2. HEVC is initially published in 2013 as the new generation standard. It has offered bitrate reduction of thirty to fifty percent to achieve comparable quality to H.264 [17]. Unfortunately, it has not proved to be the right solution as its predecessor despite its technical characteristics.

In 2013 Google released VP9 video coding format. Its development is based on previous VP8 standard (initally On2 technologies) and is intended for internet media delivery [18-20]. It has similar quality ratings with HEVC but with different implementation and traffic effects. Google's VP9 has been supported by popular web browsers and video platforms. In 2018 AOMedia (Alliance for Open Media) developed AV1 standard (AOMedia Video 1), and united top tech leaders for the next generation of media delivery over internet. The developed AV1 is considered a general-purpose open and royalty-free video coding format, and a successor of VP9 [18].

When efficient delivery over Internet is the target, the most common web codec formats and corresponding containers listed by Mozilla are presented in Table 1 [21]. Depending on source video format and set configuration parameters different coding results can be obtained. Supported codecs and formats by FFmpeg can be listed by using adequate commands [11]. This is shown in Fig. 3, where explanation of abbreviations is also given.

Table 1. Popular codecs and containers [21].

Codec	**Full codec name**	**Container support**
AV1	AOMedia Video 1	MP4, WebM
AVC (H.264)	Advanced Video Coding	3GP, MP4
H.263	H.263 Video	3GP,
HEVC (H.265)	High Efficiency Video Coding	MP4
MP4V-ES	MPEG-4 Video Elemental Stream	3GP, MP4
MPEG-1	MPEG-1 Part 2 Visual	MPEG, Quick Time
MPEG-2	MPEG-2 Part 2 Visual	MP4, MPEG, Quick Time
Theora	Theora	Ogg
VP8	Video Processor 8	3GP, Ogg, WebM
VP9	Video Processor 9	MP4, Ogg, WebM

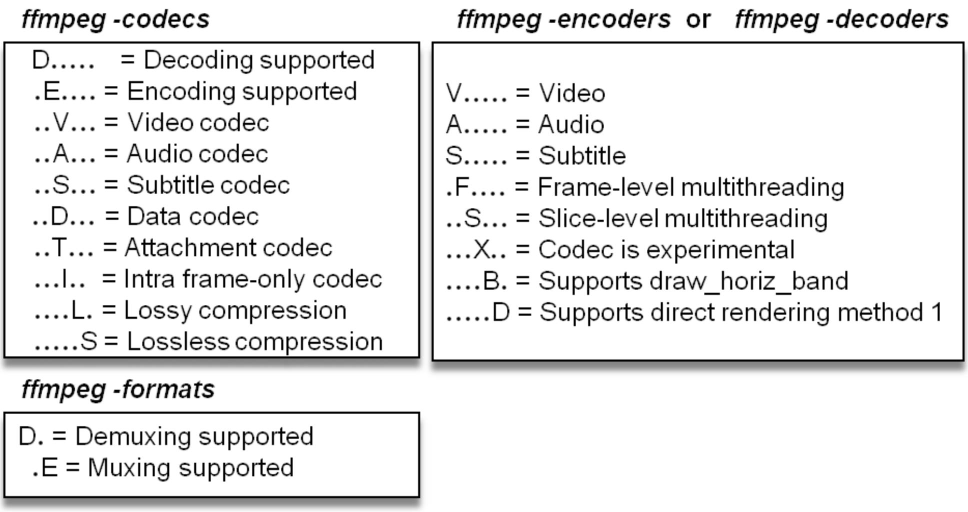

Figure 3. Commands for listing supported codecs and formats.

3 Use case for media transcoding and preset options

Video (and audio) can be manipulated by (trans-)coding and controlling its quality. This can be done in many ways like setting constant factor quality, buffer size, or using constant, constrained or variable bit rate [11]. In Fig. 4 commands applied here are presented. Firstly, audio is removed, and then constant rate factor (crf) is selected as a model for controlling output. The option is available for popular codecs and keeps the output quality level by rate control method. For VP9 and AV1, the crf usually takes values from 23 to 63, and in the case popular codecs x264 and x265 it matches quality ranges from 19 to 41 [22]. Lower crf values correspond to higher quality. The parameter crf chosen here is 35, and *libsvtav1* which represents the AV1 codec or to be more precisely SVT-AV1 (Scalable Video Technology for AV1) codec [23, 20]. Video of about ten minutes length (19036 frames) and 30fps [24] is tested according different preset options (codec Lavc59.39.100 libsvtav1). A preset is a collection of options that can provide certain encoding speed that slower preset may provide higher quality per filesize with more time needed to encode [12]. Similar preset options are available for AVC and HEVC. The preset options are focused on speed and codec complexity. In the case of SVT-AV1 in Fig. 4 preset option (value 10) is shown with no additional tuning applied here. Supported preset options ranges from 0 to 13 with 13 for debugging and higher speed for higher preset value.

```
ffmpeg -i av_input -c copy -an  v_file
ffmpeg -i v_file  -c:v libsvtav1 -crf 35 v_output
ffmpeg -i v_file  -c:v libsvtav1 -preset 10 -crf 35 v_output
```

Figure 4. Examples of ffmpeg commands.

The obtained results for SVT-AV1 and five preset options are given in Table 2, without direct measuring the video quality, and there seems that there are no significant difference after reproduction to a standard viewer under common circumstances (ten volunteers using LED 23" monitor). On the other hand, time or speed needed for transcoding is quite different. Here, when no preset option is applied, as in Fig. 4, the same results as with preset 10 are obtained. The lowest filesize is for preset option 8, as shown in Table 2.

Table 2. SVT-AV1 results for different coding speed.

No.	Preset option	fps	Lsize	bitrate	speed
1	12	69	151584kB	1957.1kbits/s	2.29x
2	10	40	141200kB	1823.0kbits/s	1.32x
3	8	17	136125kB	1757.5kbits/s	0.583x
4	6	6.3	145011kB	1872.2kbits/s	0.212x
5	4	1.5	147742kB	1907.5kbits/s	0.0485x

Table 3. H.264 results for different coding speed.

No.	Preset option	fps	Lsize	bitrate	speed
1	ultrafast	199	645603kB	8335.3kbits/s	6.63x
2	superfast	116	249509kB	3221.7kbits/s	3.87x
3	veryfast	96	143810kB	1856.9kbits/s	3.21x
4	faster	58	169341kB	2186.6kbits/s	1.92x
5	fast	50	179782kB	2321.4kbits/s	1.68x
6	medium	43	178491kB	2304.7kbits/s	1.44x

Number of frame per second (fps) for output is 30fps, but for processing the video it varies. When applying the preset option for the libx264 and crf 25, without tuning, from medium to ultrafast, higher speeds are obtained, as in Table 3. Medium is default preset option and it may range from veryslow to ultrafast. For the settings here, the lowest file size and bitrate are obtained for preset option veryfast. The quality difference exists, meaning often higher quality for slower coding [25]. This trend is illustrated in Fig. 5 based on VMAF (Video Multimethod Assessment Fusion) scores. One may still consider that there is no significant difference to a standard viewer in this 1080p case. So, preset default option may respond to higher speed at the price of slightly decrease of quality. Some of the details of the presets can be found in [26].

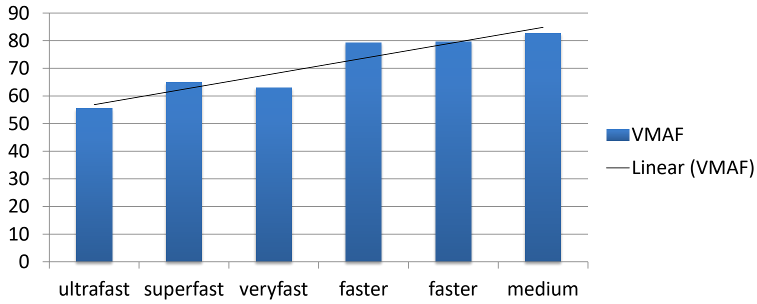

Figure 5. VMAF scores using six presets.

4 Towards new trends in video codecs

Joint Video Exploration Team (JVET) has been working on the Versatile Video Codec (VVC) or H.266 as the successor of HEVC. It is expected that new VVC could deliver the same quality with up to 50% improvement compared to HEVC. MPEG released this new standard in 2020, where UHD 4k/8k, high dynamic range (HDR), VR (Virtual Reality) and omnidirectional 360 degrees video are taken into account [13, 27]. MPEG also proposed Coding-independent code points (CICP) and Versatile Supplemental Enhancement Information (VSEI) for improved interpretation of bitstreams and its parameters [13]. EVC (Essential Video Coding) or MPEG-5 Part 1 addresses cases where coding standards have not be adopted despite technical characteristics. MPEG-5 part 2 or LCEVC (Low Complexity Enhancement Video Coding) is developed, where enhancement layer is specified for streaming when combined with a base encoded video [13, 28]. Wireless and mobile multimedia delivery solutions are especially valuable, and there is a need for focusing on real-time transmission [28-29]. According to Bitmovin's Industry Report H.264 is dominant in 2020 [30]. In [31] it is obtained that besides x265 and VP9, which can qualify for real-time performance, AV1, VVC, and HM (HEVC test model) incur long compression times, and their compression efficiency is the advantage rather than real-time applications.

Patents do have affects on standards and applications [32]. Different patent holders and patent pools have been related to standards implementation, like [33, 34]. AI is also an important trend, and MPAI community is interested in AI-based coding with licensing formats [35]. Information of legal status of some of the solutions is difficult to answer whether one is a lawyer or not. Industrial aspects and, generally, profit should not affect the speed of general acceptance of the new standards. The novel standards are mostly results of united work, as well as the selection of core technologies, but the cooperation often makes conflicts [36]. FRAND means fair, reasonable and non-discriminatory, and licenses may be considered FRAND or non-FRAND. Having in mind impartiality challenge existed in some standards, the licensing for new solutions are of great interest. Fair, adaptive, but balanced approaches are expected for future media codec market, industry, and research needs.

5 Conclusion

Open-source and free video and multimedia technology frameworks, tools and projects play important role in education and research, especially when it comes to video and multimedia engineering fields. The necessity of such projects should be recognized at a larger scale due to necessity of future video and multimedia experts. Moreover, the impression is that such tools can be extremely useful to the professionals for further research and making extensions. New codec solutions should have in mind different presets and optimizations in order to deal with transmission challenges, as well as practical licensing approaches.

References

[1] Chellappa R, Theodoridis S. Academic Press Library in Signal Processing, Volume 7: Array, Radar and Communications Engineering. Academic Press; 2017 Dec 1.

[2] Zhang T, Mao S. An overview of emerging video coding standards. GetMobile: Mobile Computing and Communications. 2019 May 2;22(4):13-20.

[3] Ma N. Distributed video coding scheme of multimedia data compression algorithm for wireless sensor networks. EURASIP Journal on Wireless Communications and Networking. 2019 Dec;2019(1):1-9.

[4] Hossain K, Roy S. A data compression and storage optimization framework for iot sensor data in cloud storage. In2018 21st International Conference of Computer and Information Technology (ICCIT) 2018 Dec 21 (pp. 1-6). IEEE.

[5] Cisco VNI Complete Forecast Highlights. Global - 2021 Forecast Highlights.https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf (last accessed 01.07.2022.)

[6] Pelurson S, Cozanet J, Guionnet T, Abdoli M, Biatek T. AI-Based Saliency-Aware Video Coding. SMPTE Motion Imaging Journal. 2022 May 10;131(4):21-9.

[7] Gavrovska A. Uvod u savremene video tehnologije i sisteme. Akademska misao, 2021.

[8] Ozer J. What is Codec? https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74487 (last accessed 01.07.2022.)

[9] Ulas D. Digital transformation process and SMEs. Procedia Computer Science. 2019 Jan 1;158:662-71.

[10] Kale V. Digital transformation of enterprise architecture. CRC Press; 2019 Jul 8.

[11] FFmpeg. https://ffmpeg.org/, https://trac.ffmpeg.org/ (last accessed 01.07.2022.)

[12] Ferrando N. "FFmpeg - From Zero to Hero". https://ffmpegfromzerotohero.com/ (last accessed 01.07.2022.)

[13] MPEG. https://www.mpeg.org/ (last accessed 15.07.2022.)

[14] OpenCV - Open source Coputer Vision, Video I/O with OpenCV Overview, https://docs.opencv.org/4.x/d0/da7/videoio_overview.html (last accessed 15.07.2022.)

[15] Wu X, Qu P, Wang S, Xie L, Dong J. Extend the FFmpeg Framework to Analyze Media Content. arXiv preprint arXiv:2103.03539. 2021 Mar 5.

[16] ITU-T H.264 : Advanced video coding for generic audiovisual services. https://www.itu.int/rec/T-REC-H.264-202108-I/en (last accessed 15.07.2022.)

[17] Bitmovin to Bitmovin's Video Developer Report. https://go.bitmovin.com/video-developer-report-2020 (last accessed 15.07.2022.)

[18] Alliance for Open Media. https://aomedia.org/ (last accessed 15.07.2022.)

[19] Chen Y, Murherjee D, Han J, Grange A, Xu Y, Liu Z, Parker S, Chen C, Su H, Joshi U, Chiang CH. An overview of core coding tools in the AV1 video codec. In2018 Picture Coding Symposium (PCS) 2018 Jun 24 (pp. 41-45). IEEE.

[20] Gavrovska AM, Milivojevic MS, Zajic G. Analysis of SVT-AV1 format for 4k video delivery. In 2020 28th Telecommunications Forum (TELFOR) 2020 Nov 24 (pp. 1-4). IEEE.

[21] Web video codec guide, https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs (last accessed 15.07.2022.)

[22] Wu PH, Katsavounidis I, Lei Z, Ronca D, Tmar H, Abdelkafi O, Cheung C, Amara FB, Kossentini F. Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications. InApplications of Digital Image Processing XLIV 2021 Aug 1 (Vol. 11842, pp. 236-256). SPIE.

[23] SVT-AV1. https://gitlab.com/AOMediaCodec/SVT-AV1 (last accessed 15.07.2022.)

[24] Big Buck Bunny, h HYPERLINK "https://peach.blender.org/"ttps://peach.blender.org (last accessed 01.07.2022.)

[25] Ozer J. Introduction to ABR Production & Delivery, https://www.streamingmedia.com Streaming Media West 2019 (last accessed 01.07.2022.)

[26] OBS, Streaming with x264, https://obsproject.com/blog/streaming-with-x264#presets 2017 (last accessed 01.07.2022.)
[27] Bross B, Wang YK, Ye Y, Liu S, Chen J, Sullivan GJ, Ohm JR. Overview of the versatile video coding (VVC) standard and its applications. IEEE Transactions on Circuits and Systems for Video Technology. 2021 Aug 2;31(10):3736-64.
[28] Minopoulos G, Memos VA, Psannis KE, Ishibashi Y. Comparison of video codecs performance for real-time transmission. In2020 2nd International Conference on Computer Communication and the Internet (ICCCI) 2020 Jun 26 (pp. 110-114). IEEE.
[29]] Fujihashi, T., Koike-Akino, T. and Watanabe, T., 2021. Soft Delivery: Survey on A New Paradigm for Wireless and Mobile Multimedia Streaming. *arXiv preprint arXiv:2111.08189*.
[30] Bitmovin to Bitmovin's Video Developer Report. https://go.bitmovin.com/video-developer-report-2020 (last accessed 01.07.2022.)
[31] Panayides, A. S., Pattichis, M. S., Pantziaris, M., Constantinides, A. G., & Pattichis, C. S. (2020). The battle of the video codecs in the healthcare domain-a comparative performance evaluation study leveraging VVC and AV1. *IEEE Access*, *8*, 11469-11481.
[32] Pfeiffer, S., 2009. Patents and their effect on Standards: Open video codecs for HTML5. *International Free and Open Source Software Law Review*, *1*(2), pp.131-138
[33] MPEG-LA. https://www.mpegla.com/ (last accessed 01.07.2022.)
[34] Access Advance. https://accessadvance.com/ last accessed 01.07.2022.)
[35] MPAI community. https://mpai.community/standards/mpai-evc/about-mpai-evc/ last accessed 01.07.2022.)
[36] Jones SL, Leiponen A, Vasudeva G. The evolution of cooperation in the face of conflict: Evidence from the innovation ecosystem for mobile telecom standards development. Strategic Management Journal. 2021 Apr;42(4):710-40.