European Journal of Information Technologies and Computer Science
Vol 4 | Issue 5 | November 2024
ISSN 2736-5492

RESEARCH ARTICLE

Comparative Analysis of Power Consumption and Resource Utilization in Open-Source and Proprietary Media Players while using Raw Videos

Afzal Ahmed*, Mohammad Tariq Iqbal, and Mohsin Jamil

ABSTRACT

This study evaluates and compares the power consumption and resource utilization of open-source and proprietary media players during the playback of a large raw video file. Using real-time monitoring tools like HWiNFO, key metrics such as GPU power consumption, CPU power consumption, memory usage, and CPU usage percentage were collected and analyzed. The experiment was conducted on a system powered by a 12th Gen Intel(R) Core(TM) i7-12700H processor, and the media players were tested with a 2-minute, 14-second raw video file in .MOV format. A statistical analysis using t-tests was performed to assess the significance of the differences between the two categories. The results indicated that open-source media players generally exhibit lower GPU and CPU power consumption, with a potential for saving energy. Long-term power consumption analysis further demonstrated that users could achieve significant energy savings by opting for open-source media players, making them more suitable for energy-conscious environments. These findings highlight the trade-offs between power efficiency and performance while playing raw videos.

Keywords: CPU usage, GPU, Media players, raw videos.

Submitted: September 27, 2024
Published: November 20, 2024

10.24018/ejcompute.2024.4.5.140

Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, Canada.

****Corresponding Author:***
e-mail: afzala@mun.ca

1. Introduction

The technological landscape for media players has witnessed significant changes with the proliferation of both open-source and proprietary software solutions. This debate is especially pertinent when discussing media players capable of handling raw video formats, which are uncompressed and require substantial computational resources for smooth playback. Raw video files, commonly used in professional film production, are valued for their high image quality and extensive post-production flexibility. However, their size and complexity demand optimized software solutions to ensure efficient decoding, rendering, and playback.

In the world of open-source software, flexibility, cost savings, and community-driven innovation are often cited as primary advantages. Studies like those conducted by Panayides *et al*. [1] have shown that open-source solutions, including codecs like AV1 and tools such as VLC Media Player, provide adaptable frameworks that can be customized for various user needs. This flexibility, combined with the absence of licensing fees, makes open-source media players an attractive option for users looking to minimize costs. However, as noted by Mahmoud *et al*. [2], while open-source media players offer considerable adaptability, they may require more system resources than proprietary options to handle raw video efficiently.

On the other hand, proprietary media players like Adobe Premiere Pro or Apple's Final Cut Pro have been shown to outperform open-source alternatives in terms of performance and resource optimization. A study by Ohm *et al*. [3] highlights the superior compression techniques used in proprietary codecs like H.264 and H.265, which are specifically optimized for professional use cases that involve raw video formats. Proprietary software often comes bundled with professional-grade support and regular updates, ensuring that users can rely on structured long-term service agreements. This makes proprietary media players more appealing to professional users, especially those working in industries where performance and uptime are critical.

The choice between open-source and proprietary media players often boils down to a trade-off between cost and performance. Open-source software offers unparalleled flexibility and customization options, which are essential for users with unique requirements. However, as noted by studies on open-source educational platforms, the lack of professional support and potential instability in open-source projects can be a drawback, particularly for users who require high reliability and technical assistance [4]. In contrast, proprietary media players, while more expensive, typically offer better stability, performance, and comprehensive support, making them ideal for professional environments where reliability is paramount.

There are several Raw Video Formats available, such as YUV, RGB, RAW, CineForm, ProRes RAW, and CinemaDNG. These are designed to preserve high-quality video data with minimal or no compression. YUV separates luminance and chrominance, commonly used for color correction; RGB represents uncompressed video in red, green, and blue channels, offering maximum quality but large file sizes. RAW formats capture unprocessed sensor data, providing flexibility in post-production. CineForm offers compressed video for efficient editing, while ProRes RAW balances raw flexibility with efficient compression, optimized for Apple's ecosystem. CinemaDNG is used for high-end digital cinema, offering detailed image data but with limited support across non-professional players. Each format varies in its compression and compatibility, with ProRes balancing quality and efficiency for cross-platform use [5].

This study will delve into these issues by comparing the performance, customization capabilities, and resource usage of open-source and proprietary media players, with a specific focus on their ability to handle raw video formats efficiently. By analyzing existing literature and performance metrics, this research aims to provide a nuanced understanding of how each type of software fits into different user scenarios.

2. Literature Review

2.1. Customization and Flexibility

Open-source software is known for its adaptability and customization, which allows developers to modify the code to suit specific requirements. This flexibility makes open-source media players ideal for tailored environments that demand specific functionalities [6]. Studies show that open-source tools like MediaElement.js enable educators to build interactive environments, while proprietary software is more rigid but offers a smoother, out-of-the-box experience, which can be crucial in professional environments with less need for customization [7]. Proprietary solutions offer fewer customization options, which can be a limitation in dynamic environments where user needs frequently change [8].

2.2. Performance and Efficiency

In terms of performance, proprietary media players tend to have an edge, particularly when handling large and high-resolution raw video formats. Proprietary codecs such as H.264 and H.265 are well optimized for performance, ensuring better compression and high-quality playback with fewer resources [9], [10]. Open-source codecs like VP9 and AV1, while competitive, may require more memory and processing power in certain scenarios, which can be a drawback when playing raw video files [11]. Proprietary solutions like Adobe's and Apple's codecs are specifically designed for media professionals requiring high-quality output without sacrificing performance [12].

2.3. Security and Reliability

Security is a crucial factor when comparing open-source and proprietary media players. Open-source solutions benefit from transparency, allowing developers worldwide to spot vulnerabilities and fix them quickly. However, this model may result in inconsistent updates for less popular projects [13]. Proprietary software, despite its closed nature, provides a controlled environment with regular patches and vendor-based security [14]. Some studies indicate that proprietary players might lag in updating critical vulnerabilities compared to the open-source community, where peer-review mechanisms speed up security patches [15].

2.4. Support and Maintenance

Proprietary software often comes with robust support systems, including SLAs (Service Level Agreements), ensuring fast resolution of issues and minimizing downtime [16]. This is essential for enterprises that cannot afford significant downtime. Open-source media players, on the other hand, rely primarily on community-based support, which can vary in quality depending on the popularity of the software [17]. Paid professional support options for open-source tools exist, but they may still lack the structured consistency found in proprietary systems [18].

2.5. Cost and Sustainability

One of the main advantages of open-source media players is the absence of licensing fees, making them an affordable option for many organizations [19]. However, the hidden costs of maintenance, customization, and the need for skilled personnel to handle technical issues can add up [20]. Proprietary solutions, although more expensive upfront due to licensing fees, often bundle support and maintenance, offering a more predictable long-term cost structure [21]. In the long run, proprietary systems may prove more viable for organizations requiring high performance and stability without the complexity of managing open-source environments [22].

Power consumption has emerged as a critical consideration in evaluating software, particularly as the demand for energy-efficient systems grows. Software tools that monitor power consumption provide invaluable insights into how different software architectures perform under various workloads. One study explores how virtualization technologies, such as hypervisors and containers, differ significantly in their energy consumption based on system configurations, which is applicable to both open-source and proprietary software [23].

Furthermore, research highlights that open-source tools often provide more detailed energy profiling, allowing users to monitor energy consumption more effectively compared to proprietary systems. For example, open-source tools such as Powerstat and Open Hardware Monitor can be more accessible for power consumption monitoring [24]. In another analysis, metrics on energy consumption were systematically reviewed, offering a comprehensive view of how open-source and proprietary software can differ in energy efficiency. This study revealed that energy-efficient software development practices can impact whether organizations choose open-source over proprietary software, particularly in environments with limited energy resources [25].

The comparison of energy consumption across software systems under different workloads has also revealed that proprietary software often demonstrates better optimization for power-saving configurations. Empirical evidence suggests that, in many cases, proprietary software achieves greater energy efficiency, especially in high-performance scenarios [26]. Another comparative study specifically investigated the energy usage of open-source and proprietary software in various operating environments, concluding that proprietary solutions often have the advantage in energy optimization due to their more targeted resource management techniques [27]. These findings emphasize the importance of power consumption metrics when selecting software for both professional and personal use.

2.6. Media Format

When conducting this study, one of the key decisions was selecting the appropriate media format for testing the power consumption and resource utilization of different media players. Raw video formats vary in terms of file size, quality, compression, and compatibility across platforms. The media format chosen impacts the accuracy of the study, as some formats are more demanding on system resources, while others may not be supported natively by all media players. Table I shows the comparison of the most relevant raw video formats and their corresponding containers,codecs, and compatibility with the media players used in this study.

2.7. Reasons for Choosing ProRes in .MOV

- *Cross-Platform Compatibility:* ProRes in the .MOV container is widely supported on both macOS and Windows through popular software like Adobe Premiere Pro and VLC, ensuring compatibility across all media players tested:
- *High-Quality Compression:* ProRes balances excellent image quality with efficient compression, making it ideal for testing resource-intensive files without overwhelming system resources.
- *Optimized for iPhone 15 Pro:* Since the video was recorded on an iPhone 15 Pro, ProRes is the natural choice, offering professional-level quality directly from the device.
- *Widespread Player Support:* Most media players in the study natively support ProRes or can handle it with external codec packs like K-Lite, ensuring seamless testing.
- *Balanced File Size:* ProRes offers a manageable file size compared to uncompressed formats like RGB, while still being resource-intensive enough to assess power consumption effectively.

3. Experiment

The purpose of this experiment was to evaluate the power consumption and resource utilization of various open-source and proprietary media players during the playback of a large raw video file. The media players tested were divided into two categories: open-source and proprietary, and the experiment aimed to gather key performance metrics such as GPU power consumption, CPU power consumption, memory usage, and CPU utilization. Additionally, statistical analysis was performed to understand the significance of the differences between the two categories, and long-term power consumption was calculated to assess potential energy savings over time.

3.1. Hardware Used

The tests were conducted on the following hardware:

- Processor: 12th Gen Intel(R) Core(TM) i7-12700H
- Base Clock Speed: 2300 MHz
- Cores: 14 cores
- Logical Processors: 20 logical processors
- Physical Memory Available: 16 GB DDR4
- Operating System: Microsoft Windows 11 Home
- OS Version: 10.0.22631 Build 22631
- Display: Intel(R) Iris(R) Xe Graphics
- Adapter Type: Intel(R) Iris(R) Xe Graphics Family, Intel Corporation compatible
- Driver Version: 31.0.101.4575

TABLE I: Comparison of Raw Media Formats

Raw video format	Container	Description	Codecs required
RGB [29]	.avi, .mov	Uncompressed RGB video	RGB
RAW [30]	.raw, .r3d	Camera sensor data	RAW
CineForm [31]	.mov, .avi	Compressed raw format	CineForm codec
ProRes RAW [32]	.mov	Apple raw format	ProRes RAW
MOV [33]	.mov	Multimedia container	Varies (ProRes, H.264, etc.)
CinemaDNG [34]	.dng	Digital cinema raw format	CinemaDNG codec
Raw video format	Container	Description	Codecs Required
YUV [28]	.yuv	Color components	YUV

3.2. Tools Used

The following tools were used to measure and analyze the results:

- *HWiNFO:* For real-time monitoring of hardware metrics such as GPU and CPU power consumption, memory usage, and CPU utilization.
- *Microsoft Excel:* Used to compile, visualize, and analyze the data, including generating comparative graphs.

3.3. Media File

The test media file was a raw .MOV video file with a duration of 2 minutes and 14 seconds and a total size of 3.22 GB. With a resolution of 4K (3840×2160), color depth of 10-bits, which offers a wide dynamic range and better color accuracy compared to 8-bit, and uncompressed file was selected to simulate a high-demand workload for the media players, ensuring the system's resources were taxed during the playback.

3.4. Media Players Tested

The experiment was conducted on two categories of media players:

- Open-Source Media Players: Kodi, MPC (Media Player Classic), MPV, SMP, VLC.
- Proprietary Media Players: Windows Media Player, ACG, ALLPlayer, GOM, KMPlayer, LAPlayer, POT Player.

3.5. Data Collected

The following performance metrics were gathered for each media player during video playback:

1. *GPU Power Consumption (Watts):* Measures the power drawn by the GPU during video playback.
2. *CPU Power Consumption (Watts):* Measures the power consumed by the CPU during playback.
3. *Memory Usage (MB):* Reflects the total amount of physical memory (RAM) used by each media player during playback.
4. *CPU Usage Percentage (%):* Indicates the percentage of the system's CPU resources utilized by each player.

4. Results

The experiment provided valuable insights into how different media players, categorized as either open-source or proprietary, manage system resources, including power consumption and memory usage, during raw video playback. The following section details the results of GPU and CPU power consumption, memory usage, and CPU utilization. It also includes extended calculations for potential long-term energy savings.

4.1. GPU Power Consumption

The GPU power consumption was measured in watts for each media player. The average GPU power consumption was lower for open-source media players compared to proprietary players. Table II summarizes the average GPU power consumption for each media player while playing .MOV file.

From Table II, it can be seen that open-source media players have a higher average GPU power consumption of 0.36 W compared to proprietary players, which have an average of 0.21 W. To further understand the impact of GPU power consumption, Fig. 1 shows the comparison of the average GPU power consumption of each media player.

4.2. CPU Power Consumption

The CPU power consumption was another key metric analyzed during the playback of the raw video file. Table III shows the average CPU power consumption for each media player.

TABLE II: Average GPU Power Consumption

Media player	Average GPU power consumption (W)
Kodi	0.3
MPC	0.3
MPV	0.5
SMP	0.5
VLC	0.2
WindowsMP	0.2
ACG	0.1
ALLPlayer	0.1
GOM	0.2
KMPlayer	0.3
LAPlayer	0.2
POT Player	0.3

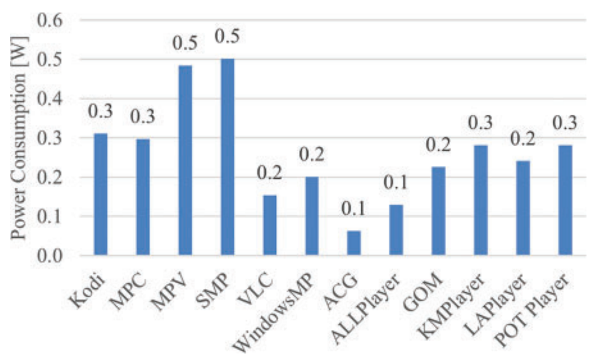

Fig. 1. Comparison of average GPU power consumption.

TABLE III: Average CPU Package Power Consumption

Media player	Average CPU power consumption (W)
Kodi	10.9
MPC	6.8
MPV	8.3
SMP	8.4
VLC	13.6
WindowsMP	13.9
ACG	6.7
ALLPlayer	11.3
GOM	15.7
KMPlayer	8.6
LAPlayer	13.7
POT Player	7.9

The average CPU power consumption for open-source media players was approximately 9.6 W, whereas proprietary media players consumed an average of 11.2 W. This difference suggests that open-source media players tend to be more energy-efficient in terms of CPU usage. Figs. 2 and 3 represent CPU power consumption across open-source and proprietary media players, respectively, over time. These graphs show the trends and fluctuations, particularly players like VLC and GOM exhibiting higher CPU usage than others.

4.3. Memory Usage

Memory usage, measured in megabytes (MB), is a key performance indicator for how much of the system's RAM each media player consumes during playback. Table IV summarizes the results.

The average memory usage for open-source media players was 8085.3 MB, whereas proprietary players averaged 8797.5 MB. Proprietary players, particularly ACG, exhibited the highest memory usage at 10,263.9 MB. Figs. 4 and 5 compare the memory usage of open-source and proprietary media players, respectively.

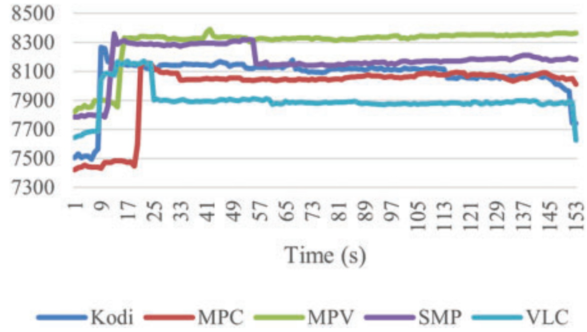

Fig. 4. Memory usage (MBs) of open-source media players.

4.4. CPU Usage Percentage

The CPU usage percentage represents the amount of CPU resources consumed by each media player. Table V presents the results.

The average CPU usage for open-source players was 12.2%, while proprietary players consumed an average of 14.9% of the CPU resources. KMPlayer, a proprietary player, exhibited the highest CPU usage at 22.9%.

As exhibited in Table V, raw video formats demand significantly more CPU resources than GPU during playback because of the sheer volume of uncompressed data they contain. The CPU is responsible for decoding and

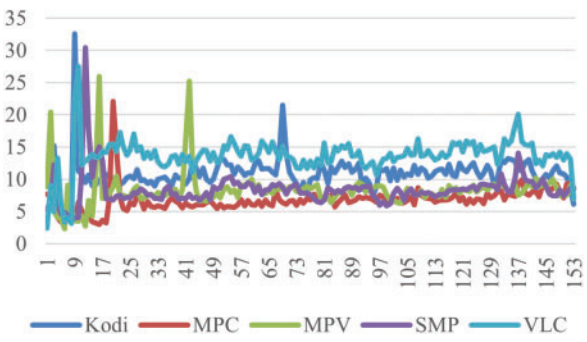

Fig. 2. CPU power consumption (W) of open-source media players.

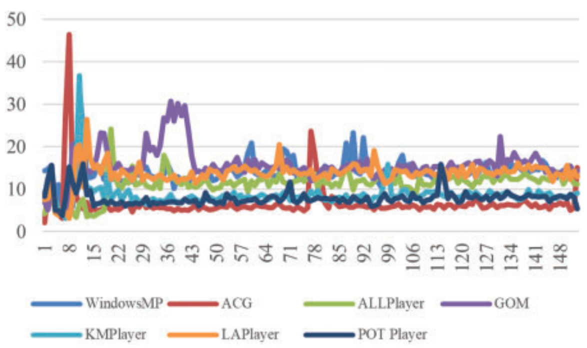

Fig. 3. CPU power consumption (W) by proprietary media players.

TABLE IV: Average Memory Consumption

Category	Media player	Average memory usage (MB)
Open-Source	Kodi	8,074.0
	MPC	7,984.2
	MPV	8,293.3
	SMP	8,176.0
	VLC	7,899.0
Proprietary	WindowsMP	9,097.4
	ACG	10,263.9
	ALLPlayer	8,608.4
	GOM	8,754.7
	KMPlayer	8,051.7
	LAPlayer	7,938.6
	POT Player	7,872.5

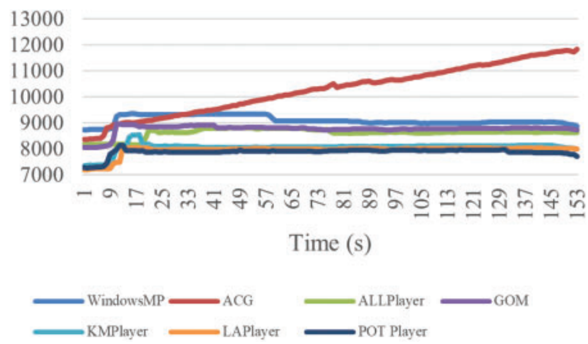

Fig. 5. Physical memory usage (MBs) by proprietary media players.

TABLE V: Average CPU Usage Percentage for Each Media Player

Category	Media player	Average CPU usage (%)
Open-Source	Kodi	11.6
	MPC	15.5
	MPV	11.4
	SMP	10.8
	VLC	11.8
Proprietary	WindowsMP	11.2
	ACG	17.8
	ALLPlayer	15.7
	GOM	12.1
	KMPlayer	22.9
	LAPlayer	10.4
	POT Player	16.9

processing this data, handling tasks like color space transformations and data handling, which require precise and intensive computation. Unlike compressed formats that leverage GPU hardware acceleration, raw video does not offload much work to the GPU. As a result, the CPU handles most of the real-time processing, especially in workflows involving high color depth and resolution like 4K video.

The results of all the above experiments demonstrate that while playing raw videos, open-source media players are generally more energy-efficient, particularly in terms of CPU and GPU power consumption. Proprietary media players tend to consume more memory and CPU resources, though some proprietary players also manage power efficiently under specific conditions. Over the course of a year, choosing open-source players could result in notable energy savings, particularly for users who rely on media players for extended periods (like a kiosk left on 24/7).

5. Conclusions and Future Work

The results of this study highlight that while playing .MOV video file open-source media players tend to be more energy-efficient than proprietary media players, particularly in terms of CPU and GPU power consumption. While proprietary media players generally consume more memory and CPU resources, open-source alternatives offer more efficient power usage, which could lead to notable energy savings over time. Based on the findings, users who prioritize energy efficiency, particularly in resource-constrained or environmentally conscious settings, may find open-source media players more favorable. On the other hand, proprietary players may still have advantages in specialized performance optimization, though at the cost of higher resource usage.

Future work could expand on this research by testing a broader range of media formats and file types to see if these patterns hold across various workloads. Additionally, incorporating a wider variety of hardware configurations, including GPUs from different manufacturers, could provide more generalizable results. Further studies could also explore the impact of different codecs and playback settings on power consumption, as well as investigate more advanced energy-saving techniques employed by proprietary players. Lastly, longer-term tests on actual energy savings in multi-device environments or across larger user bases could provide deeper insights into the broader environmental impact of choosing open-source over proprietary software.

Conflict of Interest

The authors declare that they do not have any conflict of interest.

References

[1] Panayides AS, Pattichis MS, Pantziaris M, Constantinides AG, Pattichis CS. The battle of the video codecs in the healthcare domain-a comparative performance evaluation study leveraging VVC and AV1. *IEEE Access*. 2020;8:11469–81.

[2] Mahmoud M, Rizou S, Panayides AS, Kantartzis NV, Karagiannidis GK, Lazaridis PI, *et al.* A Survey on Optimizing Mobile Delivery of 360° Videos: Edge Caching and Multicasting. *IEEE Access*. 2023;11:68925–42. doi:10.1109/ACCESS.2023.3292335.

[3] Ohm JR, Sullivan GJ, Schwarz H, Tan TK, Wiegand T. Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). *IEEE Trans Circ Syst Video Technol*. 2012;22(12):1669–84.

[4] Valiandi I, Panayides AS, Kyriacou E, Pattichis CS, Pattichis MS. A comparative performance assessment of different video codecs. *International Conference on Computer Analysis of Images and Patterns*, pp. 265–75, Cham: Springer Nature Switzerland, 2023 September.

[5] Seshadrinathan K, Soundararajan R, Bovik AC, Cormack LK. Study of subjective and objective quality assessment of video. *IEEE Trans Image Process*. 2010;19(6):1427–41.

[6] Anand A, Krishna A, Tiwari R, Sharma R. Comparative analysis between proprietary software vs. open-source software vs. free software. *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 144–7, Solan, India: IEEE, 2018 December.

[7] Bamhdi A. Requirements capture and comparative analysis of open source versus proprietary service oriented architecture. *Computer Standards Interf*. 2021;74:103468.

[8] Meng Z, Lee SY. Open source vs. proprietary software: competition and compatibility. *Paper provided by Econ WpA in its series Industrial Organization with number 0508008*, 2005.

[9] Nguyen-Duc A. The impact of software complexity on cost and quality-a comparative analysis between Open source and proprietary software. arXiv preprint arXiv:1712.00675. 2017.

[10] Reinhard E, Francois E, Boitard R, Chamaret C, Serre C, Pouli T. High dynamic range video production, delivery and rendering. *SMPTE Motion Imaging J*. 2015;124(4):1–8.

[11] Eswer V, Dessai S. Processor performance metrics analysis and implementation for MIPS using an open source OS. *Int J Reconfig Embedded Syst*. 2021;10(2):137.

[12] Jordan L. *Final Cut Pro Power Tips*. New Riders; 2022.

[13] Yılmaz N, Kolukısa Tarhan A. Quality evaluation models or frameworks for open source software: a systematic literature review. *J Softw: Evol Process*. 2022;34(6):e2458.

[14] Boulanger A. Open-source versus proprietary software: is one more reliable and secure than the other? *IBM Syst J*. 2005;44(2):239–48.

[15] Angermeir F, Voggenreiter M, Moyón F, Mendez D. Enterprise-driven open source software: a case study on security automation. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 278–287, IEEE, 2021 May.

[16] Huss R, Coupland SE. Software-assisted decision support in digital histopathology. *J Pathol*. 2020;250(5):685–92.

[17] Wermke D, Klemmer JH, Wöhler N, Schmüser J, Ramulu HS, Acar Y, *et al.* "Always contribute back": a qualitative study on security challenges of the open source supply chain. *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1545–1560, IEEE, 2023 May.

[18] Avatavului C, Cucu AI, Gherghescu AM, Boiangiu CA, Stanica IC, Tudose C *et al.* Open-source and closed-source projects: a fair comparison. *J Inf Syst Oper Manag*. 2023;17(2):1–22.

[19] Giera J, Brown A. *The Costs and Risks of Open Source*. Cambridge, MA: Forrester Research Inc, 2004.

[20] Prana GAA, Sharma A, Shar LK, Foo D, Santosa AE, Sharma A, Lo D. Out of sight, out of mind? How vulnerable dependencies affect open-source projects. *Empir Softw Eng*. 2021;26:1–34.

[21] Murciano-Goroff R, Zhuo R, Greenstein S. Hidden software and veiled value creation: illustrations from server software usage. *Res Policy*. 2021;50(9):104333.

[22] Carpenter M, Daidj N. Game console manufacturers: the end of sustainable competitive advantage? *Digiworld Econ J*. 2014;(94):39.

[23] Morabito R. Power consumption of virtualization technologies: an empirical investigation. *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pp. 522–7, IEEE, 2015 December.

[24] Ebrahim R, Luvhengo F, Vilakazi M, Mamushiane L, Lysko AA. Software tools for power consumption monitoring for open 5G and beyond research: brief review. doi:10.1109/AFRICON55910.2023.10293355. https://www.researchgate.net/publication/373640241.

[25] Ergasheva S, Khomyakov I, Kruglov A, Succil G. Metrics of energy consumption in software systems: a systematic literature review. *IOP Conference Series: Earth and Environmental Science*, vol. 431, no. 1, pp. 012051, IOP Publishing, 2020 February.

[26] Economides N, Katsamakas E. Linux vs. Windows: a comparison of application and platform innovation incentives for open source and proprietary software platforms. In *The Economics of Open Source Software Development*. Elsevier, 2006. pp. 207–18.
[27] Capra E, Francalanci C, Slaughter SA. Is software "green"? Application development environments and energy efficiency in open source applications. *Inf Softw Tech*. 2012;54(1):60–71.
[28] Bovik AC. *The Essential Guide to Video Processing*. 2nd ed. Academic Press, 2014. pp. 89–95.
[29] Debevec PE. Rendering with natural light. *J Graphics Process*. 2015;14(3):28–37.
[30] Gress J. *Digital Video Production*. 4th ed. Taylor & Francis, 2018. pp. 129–32.
[31] Corliss R. High-performance video compression formats. *J Digit Media Process*. 2020;23(2):45–53.
[32] Ahmed MS. Record motion picture in different formats to get the best possible quality conciding with less storage space. *Int Des J*. 2022;12(6):249–56.
[33] Apple Inc. TN3177: Understanding Alternate Audio Track Groups in Movie Files. Accessed on: Nov. 9, 2024. Available from: https://developer.apple.com.
[34] Carson PE. CinemaDNG: the open source raw video format. *Digit Cinema Rev*. 2021;18(5):123–9.
[35] Bovik AC. *The Essential Guide to Video Processing*. 2nd ed. Academic Press; 2009.