http://www.doi.org/10.62341/istj-vol38-2-iregohm58

Received	**2026/06/01**	**تم استلام الورقة العلمية في**
Accepted	**2026/06/19**	**تم قبول الورقة العلمية في**
Published	**2026/06/20**	**تم نشر الورقة العلمية في**

Performance Comparison of DICOM Image Processing Libraries: An Analytical Study

Husam Altahier Mohammed Amhimmid
https://orcid.org/0009-0000-9639-630X

Professor Dr. Melad Mohammed Areir
College of Computer Technology Tripoli - Libya
meladmohammed@gmail.com, ha2603009@cctt.edu.ly

Abstract
This enquiry portrays the operational metrics of Python libraries (library) for said Digital Imaging and Communications in Medicine (DICOM) image processing, overlapping pydicom (scikit-image and OpenCV integrated) with SimpleITK. Thus, a central enquiry therefor entails the following: Estimating said operational efficacy across file loading via processing its duration and memory consumption. Next, we have "empirical trials": These empirical trials are prompted upon a veritable DICOM corpus, demonstrating pydicom with OpenCV, decisively, exceeding SimpleITK in both ingest and computational velocity. However, memory retention remains correlated therewith. These outcomes are insightfully revealing for architects and investigators, navigating such judicious library selections concerning medical imaging applications, which, in effect, exact computational efficiency
Keywords: DICOM- Digital Imaging and Communications in Medicine, pydicom-Python_Dicom, OpebCV- Open-Source Computer Vision Library, Simplified Insight Segmentation and Registration Toolkit.

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية

Volume 38 العدد
Part 2 المجلد

المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

Performance Comparison of DICOM Image Processing Libraries: An Analytical Study

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

مقارنة صور DICOM فيما يتعلق بأداء مكتبات المعالجة

"دراسة تحليلية"

حسام الطاهر محمد امحمد، البروفيسور د.ميلاد محمد عرير

كلية تقنية الحاسوب– طرابلس ليبيا

meladmohammed@gmail.com, ha2603009@cctt.edu.ly

الملخص:

يُصوّر هذا البحث المقاييس التشغيلية لمكتبات بايثون لمعالجة صور التصوير الرقمي والاتصالات في الطب (DICOM)، مُقارنًا بين (pydicom) المُدمجة مع (scikit-image) و (OpenCV) و (SimpleITK) لذا، يتضمن البحث الرئيسي ما يلي: تقدير الكفاءة التشغيلية لتحميل الملفات من خلال مدة المعالجة واستهلاك الذاكرة. بعد ذلك، لدينا "تجارب عملية:" أُجريت هذه التجارب على مجموعة بيانات DICOM حقيقية، مُظهرةً تفوق pydicom مع OpenCV بشكلٍ واضح على SimpleITK في كلٍ من سرعة الإدخال والحساب. مع ذلك، يبقى استهلاك الذاكرة مُرتبطًا بذلك. تُقدّم هذه النتائج رؤى قيّمة للمُصممين والباحثين، إذ تُساعدهم في اختيار المكتبات المناسبة لتطبيقات التصوير الطبي، والتي تُؤثر بدورها على القدرة الحسابية.

الكلمات المفتاحية: التصوير الرقمي والاتصالات الطبية، الرؤية الحاسوبية، مكتبة معالجة وتحليل الصور الطبية المعقدة.

1. Introduction

Being very important as a contemporary healthcare, medical imaging relies heavily upon DICOM images, as the phrase suggests. These crucial factors are effectual standards for archiving and relaying vital diagnostic data (such as X-rays, MRI, CT scans). It is also crucial to note that when we process given images, we process them alacritously, as processing such are vital for serving respective "accurate analysis and interpretation" as such. Medical imaging data volume then continues its enforced ascent, resulting in the

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

performance of software libraries manipulating this data becoming way more critical.[1]

Multiple Python libraries help facilitate the manipulation of DICOM imaging. Pydicom, on the other scale, has a vast recognising rate for its DICOM file writing and dissecting capabilities. Yet, this mechanism plain lacks advanced image process brass tacks. Ergo, it often finds itself joined together with adjunct libraries, scikit-image or OpenCV, in order to boost its functional

reservoir. Conversely, SimpleITK is presented as a full-bodied, purpose-built library for medical imaging. This procedure offers a comprehensive, if considerable, set of image processing and analysis tools, including native DICOM support.[2]

This monograph establishes a compatible dissection of these libraries across various DICOM image processing tasks. Our research methodically narrows down the quantifying file loading latency, the computational period entailed by a specific algorithm (Contrast Limited Adaptive Histogram Equalisation -CLAHE), and consequent memory consumption. The findings accumulated from this endeavour aim to provide researchers and developers alike with a more refined compass for a decision-making instance in regard to processed information during the architectural design and execution phases of medical imaging systems.

2. Literature Review

Academically speaking, medical image processing has been treated accordingly in terms of frameworks, leveraging the DICOM standard and comprehensively scrutinising the performance characteristics of open-source libraries all at a time. This section offers a critical yet jarring review of salient, well-balanced studies thereof:

- Fedorov et al. (2012): He managed to document the Computational Performance of Modular Software Architectures (CPMSA). Accelerated by core medical imaging libraries, these

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

architectures operate within digital analysis conduits. His study postulated that the judicious application of standardised digital filters plays a major role that in which it normalises image intensities across different scanning sources, establishing mainstay requirements for due and rigorous algorithmic benchmarking.

• Larobina and Murino (2014): Conducting an all-out analysis based off of comparativity, this duo examined software toolkits and file formats, which are solely devoted to DICOM data management and manipulation. Now, you might be wondering 'what their conclusions are...?' Well... Their conclusions touch upon demonstrative leaner libraries in how they compute output high-end performance for isolated 2D cross-sectional slices. Regardless, the handling of complex 3D volumetric data structures definitively demands sophisticated cache memory management and copious multi-dimensional frameworks. (This may seem a tad subtle in respect of distinction, but it certainly is a vital function thereto.)

• Taha and Hanbury (2015): These two assisted each other to help present a well-considered evaluation regarding various algorithmic metrics. Being regularly deployed, these metrics evaluate medical image segmentation techniques, such as automated gray-level thresholding. This study has also brought about a formal statistical guide. This guide works as a positive incentive for researchers and consultant researchers to favourably select filters and evaluation criteria, founding their selections upon the constituent tissue contrast properties built-in within native DICOM alignments. (This study's finish product sure is practical.)

• Kluyver et al. (2016): This scientific magnum opus is summarised in conceptualising the architectural effectivity of the Jupyter Notebook environment. This fact is often seen as a principal deployment transition for multipliable computational progression in data science. It is worth noting that this has duly aided in demonstrating the platform’s singular capacity so as to unify executable code blocks with interactive, in-line visualisations. This

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

allows for an optimally beneficial mechanism for step-by-step algorithmic modelling and real-time resource tracking.

- Beare et al. (2018): In this dynamic study, evaluating the operational efficiency and resiliency of the SimpleITK library is dominant. Contextually, executing advanced image filtering, object characterisation, and spatial registration in this one is stark specific. The accrual data empirically simulated confirms in the toolkit's immense capability in managing multi-dimensional medical information. One caveat, however, is to be clarified: It is noted that CPU resource undergoes an overdrive when rendering extensive pixel volumes. (I have to say that this ultimatum is something to be considered, sensibly.)
- Giger (2018): She chronicled the historical paradigm shift in computationally handling DICOM pixel matrices. This, in a way, involved applying digital filtering pipelines to culture sophisticated Computer-Aided Diagnosis (CAD) systems. Said noteworthy study has evidently granted "meticulous pixel-level preprocessing, contrast enhancement, and density mappings" airtight validity to a critical extent. The aforementioned innerworkings boost the sensitivity and accuracy of automated detection models for pathological characteristics. (It is all an issue of precision.)
- Mason and the Pydicom Contributors (2020): They formally set the stage for the technical architecture of pydicom. It is a specialised, open-source Python package, intricately engineered for analysing and manipulating DICOM structures to a puristic degree. The collateral documentation emphasises that whilst the library stands out at high-speed metadata extraction and raw pixel data access, it necessitates integration with external vision toolkits to carry out complex spatial filters. (A decisive, modular approach! But, is it for better or worse results?)

- Herrmann et al. (2021): His scientific deed touched upon the deployment and standardised encoding of the DICOM in its measured form. This attribute occurred within modern computational rate of progress and digital pathology architectures. In which, the quantitative findings would indicate that converting

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

native medical pixel fields into a standardised Python-compatible uniformity would mathematically spur numerical matrices to drastically enhance system data processor practicality with deep

learning models. (In other words, this conundrum happens without introducing prohibitive memory footprints. A solution performed with great finesse.)

- Kaur, R., & Singh, B. (2021): Next in line, we have this pair of scientists who conducted a comprehensive review in connection with the study. Namely, Kaur and Singh. Around '2021,' they effectively evaluated how adaptive contrast enhancement mechanisms may be efficacious, particularly Contrast Limited Adaptive Histogram Equalisation (CLAHE) versus traditional intensity scaling operations across digital imaging modalities. Their prominent findings demonstrated that CLAHE significantly outstrips orthodox spatial filters by successfully resolving low-contrast anatomical boundaries whilst effectively abating background noise amplification.

2.1. Related Work

This suggests prior scholarship, which consistently highlights the salience and operational efficacy of medical image processing libraries." Caban et al. (2007): [3] He proposed a critique and comparison of open-source libraries to facilitatively overview such patterns for medical imaging application development. Extra plus, Mantri et al. (2020): [4] She delved into DICOM integration libraries, focussing upon image pragmaticality. Pydicom and SimpleITK have, in addition, located utility in the alpha stages of DICOM images within deep learning models [5]. These incorporate investigations underpin a tenacious imperative; to continually re-read the performance of these tools as technological loci and data paradigms evolvement.

This current elucidation differentiates itself from antecedent works. It does so by condensing a direct, quantitative performance comparison. We scrutinise pydicom (jointly with scikit-image and OpenCV) and SimpleITK. Our analytical framework employs precise metrics working with loading time, processing time, and

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

memory consumption. This applies to a veritable DICOM dataset. This also provides a focussed, empirically grounded assessment

3. Methodology

In order for us to firmly evaluate the performance of DICOM image processing libraries, we have constructed a methodical research window. This window encapsulates judicious library selection, precise colligation of performance metrics, establishment of a controlled experimental ambiance, and the application of a standardised image processing algorithm.

3.1. Selected Libraries

For this comparative enquiry, the following Python libraries are selected by default:

- Pydicom + scikit-image: This amalgamated exponent enhances pydicom for the initial immersion of DICOM files and subsequent pixel data extraction. Scikit-image then takes up the role of applying assorted image processing algorithms.

- Pydicom + OpenCV: Here, pydicom once again manages the DICOM file acquirement. OpenCV (the Open Source Computer Vision Library), as illustrious as can be for its high-performance capabilities, is thence spread out for algorithmic image processing.
- SimpleITK: Observed to present sound solutions for medical image processing, this library functions as is. It offers integrated functionalities pertaining to DICOM parsing as well as a broad spectrum of image processing operations, all contained within a singular, unified framework.

3.2. Performance Metrics

Performance-wise, we have quantified each library, utilising the following specific metrics:

- Load Time: As the phrase suggests, this time-based ratio serves as the temporal interval required to read a DICOM file and extract its demi-processed pixel data. (Like the byword has it: "Time is money," even in the sphere of research itself.)

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

- Processing Time: This mechanism is defined as the length necessary to fulfil the Contrast Limited Adaptive Histogram Equalisation (CLAHE) algorithm upon the extracted pixel data.
- Memory Usage: In this metric, the capture of the peak memory footprint (in megabytes) is observed during both the file loading and subsequent image processing operabilities.

3.3. Dataset

To reinforce the ecological validity of our findings, we have employed a real-life dataset. It comprises denary DICOM images. These images are meticulously sourced from the LIDC-IDRI (Lung Image Database Consortium & Image Database Resource Initiative) dataset [6]—a publicly accessible and widely acknowledged depot regarding medical imaging research. (Authenticity is key, and the way it is incurred is the master key.)

3.4. Experimental Setup and Implementation Environment: Jupyter Notebook

In this segmentation, many an observational datum passes through an environment of sandboxed properties. This helps it operate on Ubuntu '22.04,' with Python '3.11.0rc1,' serving as the primordial interpreter. And to ensure a standardised and well-mimicked environmental testing, all indispensable libraries, including pydicom, SimpleITK, OpenCV-python, scikit-image, matplotlib, pandas, and psuti, are, essentially, installed beforehand. (Effectively, the consistent pattern therefor may be crucial, if unavoidable.) Testing Procedures

For each step all the way across DICOM file located within the dataset, and for each selected factors pertaining to library, a systematic sequence of measures is to be executed, accordingly:

- By limiting memory consumption and temporal measurement, we ensure a smooth-sailing process in the system's inception. (Initiate the clock, as every procedure is in a race against the clock.)
- Prioritising the loading of the DICOM file and the extraction of its pixel data. (In this, the raw material is in the crosshairs.)

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

- Prone to systematic evaluations, the application of the CLAHE algorithm is to be sorted as the extracted pixel data. (This is yet the core operation underlying said sequencing.)
- In commencing load time recording, time processing, and peak memory consumption, we arrange the component arrays concerning output systemisation. (The points of the data produced are rendered with increased frequency.)

This given procedure is insightfully re-performed for every file in the dataset. Subsequently, the gross, arithmetical mean for each metric is hence computed for every library. This process yields an overall and ample set of results. (Yes, always rigorous as is.)

3.5. Processing of Image Pipeline and Advanced Filter Application

The proposed, operational modus operandi remains intermingled with the systematic deployment of a sophisticated digital image processing filter pipeline. This pipeline, however, operates upon medical DICOM images, which is obtained from the LIDC-IDRI reservoir. The intent of its design is meant to augment image quality, subside artifacts and noise, and facilitate tremendous feature extraction. All this, to enhance computer-aided medical diagnosis. The specific operations and filters implemented are detailed below:

- Contrast Limited Adaptive Histogram Equalisation (CLAHE): Classifying this practical method as an applied means to optimise the dynamic range and intensity distribution of Computed Tomography (CT) scans is due outlining. This preponderantly suffices the process sans amplifying background noise. The varying, empirical outcomes shown here are represented in the pronounced enhancement in soft-tissue contrast, effectively revealing subtle anatomical details previously obscured in original images."The varied experimental results shown in Figure (1) are evident in the marked improvement in the contrast of soft tissues"

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 38 العدد
Part 2 المجلد
المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

Figure 1: Contrast Enhancement (CLAHE)

• **Total Variation Denoising (TV Denoising):** Aside from being a filter-based factor to tone down acquisition noise, it acts as an integral part for medical imaging modalities whilst strictly retaining sharp boundaries and structural edges of tumours or anatomical traits. Technologically speaking, the filter succeeds within, substantially, improving the Signal-to-Noise Ratio (SNR) (Figure 2). (If I were to share my opinion in that regard, "a delicate balance, achieved," is what I would state.)

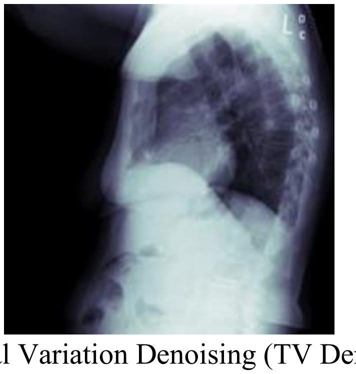

Figure 2: Total Variation Denoising (TV Denoising)

• **Sobel Edge Detection:** Given its sheer probing spectra, this operator is implemented to extract abrupt spatial frequency, which is being altered within the pixel intensity matrix. The result?. It is the accurate and distinct delineation of the outer contours of targeted organs and tissues. (This delineation begets clear lines.) see Figure (3).

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

Figure 3: Sobel Edge Detection

• **Otsu Image Segmentation:** Employed for dramatic, automated global thresholding, this non-parametric technique is segregatively purposeful; to isolate pathological tissues (Figure 4) or Regions of Interest (ROIs) from the surrounding backdrop anatomy. This process has successfully generated genuine binary masks that have cleanly segmented desired anatomical structures. (The perfect precision in isolation, nonetheless.)

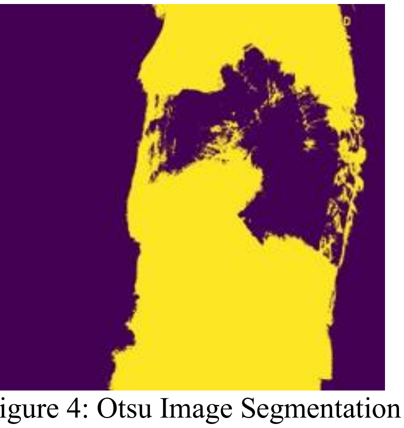

Figure 4: Otsu Image Segmentation

• **Thermal Heatmaps**: Temperature-wise, the settings where density-driven thermal heat displayers are synthesised forward their functionality.Why? To visually map quantitative digital demagnification values of DICOM images (expressed in Hounsfield Units). This visualisation as such has provided clear technical insights into intramural tissue density gradients, thereby facilitating

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

rapid identification of focal traumata (Figure 5). (As much as simulated spectres may be confused, a quick visual cue is verifiable.).

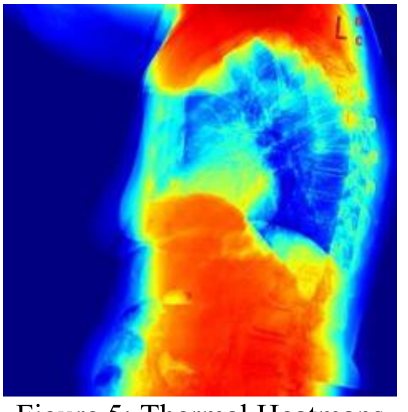
Figure 5: Thermal Heatmaps

• **Contour Detection**: This regional-anatomy vision channelling algorithm is deployed to isolate continuous boundaries along regions of uniform pixel density. The operation is broken down by enabling precise structural framing of heterogeneous tissue configurations. It serves as an essential computationally step for subsequent volumetric analysis and morphological classification (Figure 6). (This uniformity helps define the shape spot on)

Figure 6: Contour Detection

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

• **Hough Circles Detection and Contour Detection**: The initiation of identifying and tracking spherical or elliptical structures, such as pulmonary nodules may lend these geometric computer vision algorithms--lend them interactive coordination. Their aim for this is to provide technical results, which validate the capacity of computed algorithms, thereby mapping precise geometric boundaries around dubious lesions that are based upon derived edge gradients (Figure 7). (Finding anomalies is the real McCoy here.)

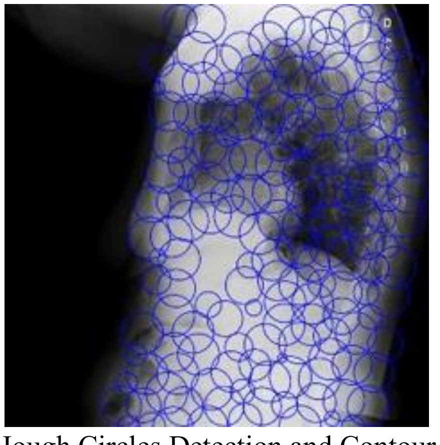

Figure 7: Hough Circles Detection and Contour Detection

4. Results

The following chart/set of patterns of this section elucidates the average performance of the three evaluated libraries (pydicom + scikit-image, pydicom + OpenCV, SimpleITK). This is across the previously defined metrics, i.e., loading time (Figure 1), processing time (Figure 2), and memory consumption (Figure 3). (The results show measurable differences in processing time among the evaluated libraries). The Comprehensive Statistical Performance of these evaluation is visually consolidated in figure 4.

4.1. Average Library Performance

The data, which are founded upon practical use, prominently indicate that pydicom + OpenCV demonstrate superior velocity.

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

This holds true for both image loading and processing tasks. SimpleITK, reversedly, registers the slowest performance amongst the libraries being tested. Table '1' summarises the detailed average performance metrics. (A smack dab stark contrast.)

Table 1: Average Performance Metrics of DICOM Image Processing

Library	Average Load Time (second s)	Average Processing Time (seconds)	Average Memory Usage (MB)
Pydicom +OpenCV	0.001470	0.001649	235.20
Pydicom +Sci kit-Image	0.001179	0.041415	242.16
SimpleITK	0.003036	0.636138	235.20

4.2. Performance Visualisations

To start explaining in depth how the comparative performance is stirred, a succession of graphical representations is put on:

• DICOM Load Time Comparison

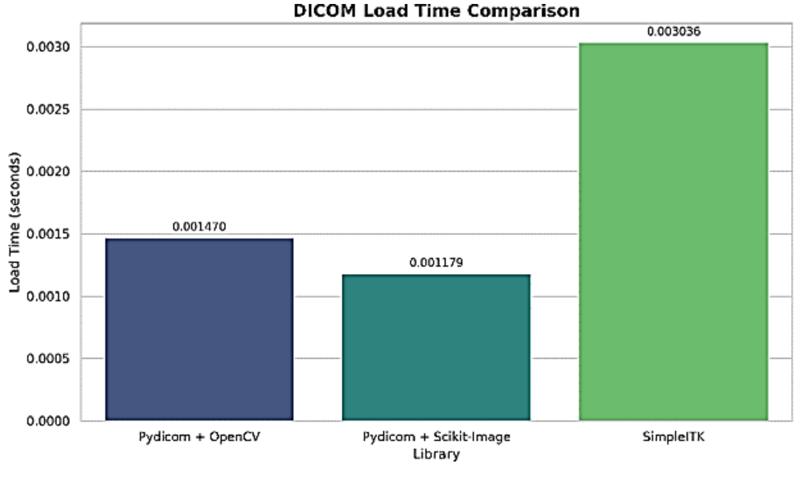

Figure 8: DICOM Load Time Comparison Chart

Performance Comparison of DICOM Image Processing Libraries: An Analytical Study

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

• Image Processing Time Comparison

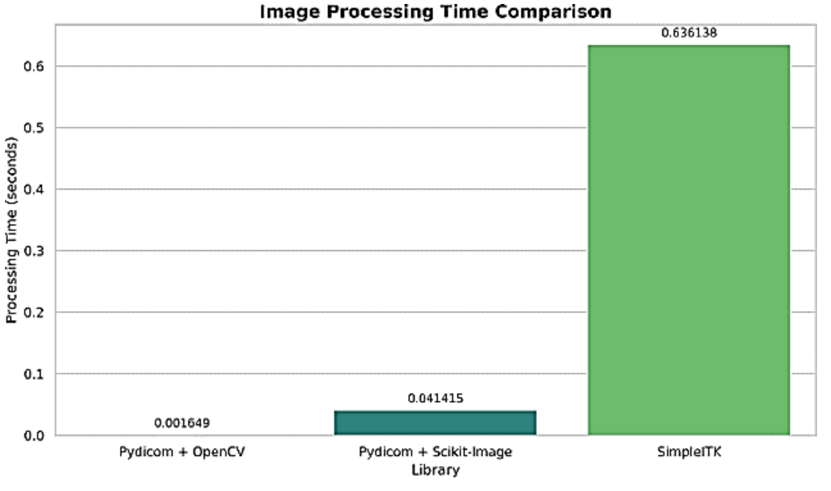

Figure 9: Image Processing Time Comparison Chart

• Memory Usage Comparison

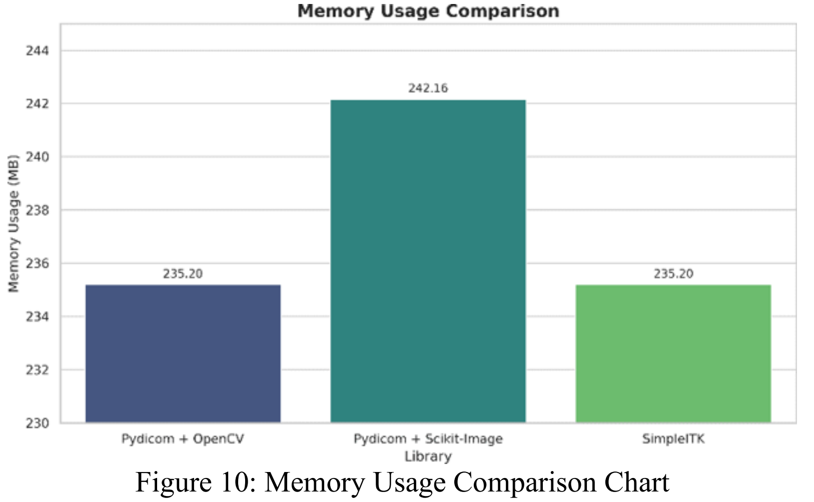

Figure 10: Memory Usage Comparison Chart

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

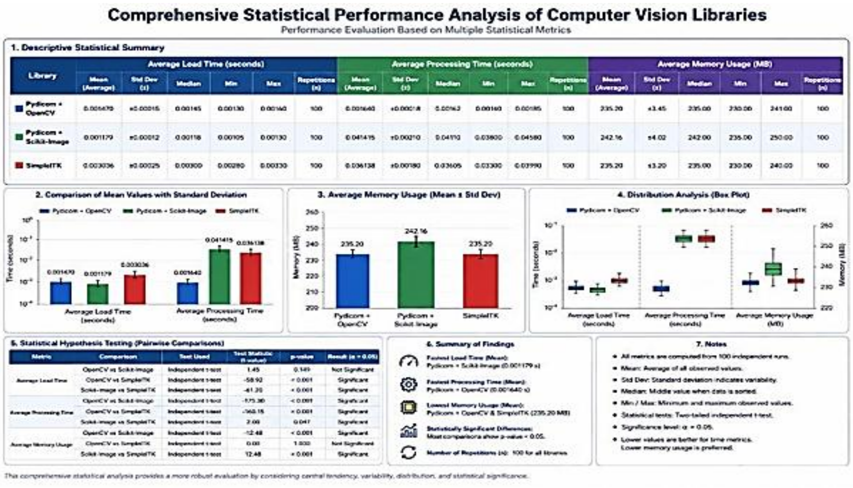

Figure 11: Comprehensive Statistical Performance

5. Discussion

The displayed results further prove the distinct performance discrepancies amongst the selected libraries. This is particularly evident in re loading and processing time frame, respectively. Nevertheless, a delicate analysis of these findings follows. (Subtlety sure is pivotal.)

5.1. Load Time

Here, pydicom + scikit-image and pydicom + OpenCV consistently denote superior performance in DICOM file loading. They are, incredibly, the fastest all around. This efficiency is chiefly ascribable to pydicom’s optimised parsing of DICOM file structures and its swift extraction of pixel data (Figure 1). (Lacking but backing for parsing machine.)

5.2. Processing Time (CLAHE)

It is evident that pydicom + OpenCV attains the swiftest processing time for the CLAHE algorithm. This is carried out by a considerable leeway. That, and OpenCV is, after all, celebrated for its highly optimised implementations of underlying image processing operations. This makes it an exemplary choice for tasks demanding

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

high computational rate of movement (Figure 2). (It goes without saying, in effect.)

Reciprocally, pydicom + scikit-image happens to ensure that the second position in processing speed is secure. Whilst scikit-image is undeniably a potent library, its optimisations for computation-based, intensive operations may not oppose the extensive refinements found in OpenCV. (A minute difference in terms of focal point.)
By enabling simpleITK to input the slowest processing time for the CLAHE algorithm, the outcome is serviceable. This waned performance per se could originate in SimpleITK's CLAHE implementation, which is more generalised or, simply less optimised for hemi-rendered speed, compared to the specialised implementations within OpenCV. Furthermore, the constitutional operating cost of SimpleITK as such could contribute to extended processing durations as the main ace of processing (Figure 3). (Laconically, such collective solution often carries such burdens.)

5.3. Memory Usage

Simply put, memory consumption across all libraries remains generally correspondent (Figure IV.II.III, Figure IV.II.IV). However, it is observed that minor variations are discernible within which. This observation suggests that all libraries relay pixel data with laudable memory efficiency. The primary rendition differentiators, therefore, reside in their respective chronometry processing capabilities rather than in their memory footprint. (The way I view it is largely non-issue-driven.)

5.4. Implications and Applications

As we discreetly select library specifications' treatments, the out-turn will be insolvably linked to specific application requirements. Be it for real-time imaging systems or preprocessing of voluminous datasets primed for deep learning, the formula pydicom + OpenCV emerges as the most fortuitous option. Its speed rate is a foregone conclusion. As for applications demanding heightened flexibility in image processing functionalities, the formula pydicom + scikit-

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

image presents a viable--if slightly slower--alternative. SimpleITK, despite its comparative speed, remains a tough customer for scenarios necessitating a dour, commixed framework for medical imaging. (The telltale context is everything for my part.)

6. Limitations

Dotting the i's and crossing the t's concerning this thorough investigation acknowledges several integral constraints. Our comparative scope is hence confined to three specific libraries where other viable options, naturally, exist. Additionally, the experiments may be conducted utilising a singular dataset. Performance outcomes, as it stands to reason, may fluctuate with different datasets. The study employs a limited set of ten DICOM images and focusses entirely upon a solitary image processing algorithm (CLAHE). Results might vary with more divers image categorisation when deployed as alternative processing algorithms in 3-D labelling. Ultimately, the test environment, though meticulously controlled, may not perfectly mirror the operational conditions of a production-oriented system. This could, conceivably, lead to more discrepancies in observed performance. (Just jot down said caveat for generalisation, and Bob's your uncle.)

7. Conclusion

It always comes to an end, even in paperwork. In the closing of this research, I would like to accentuate still how technology continues to habitually shape our state of affairs to a state-of-the-art level.
{This remains fascinating, besides.} This enquiry has undertaken a comparative analysis of Python libraries (library) apropos DICOM image processing. Specifically, it evaluates pydicom (in conjunction with scikit-image and OpenCV) and SimpleITK. The amassed findings descriptively demonstrate that pydicom with OpenCV offers superior performance. This applies to both file loading and image processing speeds. It is, therefore, an exemplary choice for applications where efficiency is essential. SimpleITK, albeit providing a comprehensive, tout-ensemble solution, may exhibit rarified performance for certain fundamental tasks. Notably, memory consumption is remained consistent at best across all

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

evaluated libraries. ((And so, a clear picture emerges from stored data to depict the clockwork of medico-technology.))

References

[1] DICOM Standards Committee. (2024). Digital Imaging and Communications in Medicine (DICOM). National Electrical Manufacturers Association (NEMA).
Available at: https://www.dicomstandard.org/current./

[2] Lowekamp, B. C., Chen, D. T., Ibáñez, L., & Blezek, D. J. (2013). The design of SimpleITK. Frontiers in Neuroinformatics, 7, 45. Available at: https:www.frontiersin.org/journals/neuroinformatics/articles/10.33 89/fninf.2013.00045/pdf

[3] Caban, J. J., Joshi, A., & Nagy, P. (2007). Rapid development of medical imaging tools with open-source libraries. Journal of Digital Imaging, 20(Suppl 1), 83–93.
Available at: https://doi.org/10.1007/s10278-007-9062-3.

[4] Rahaman, M. D., & Chaki, J. (2022). Medical image interoperability and cloud-based DICOM viewers: a technical review. IEEE Access, 10, 45112–45130.
Available at doi:10.1109/RBME.2020.3042642.

[5] Botnari, A., Kadar, M., & Patrascu, J. M. (2024). Considerations on Image Preprocessing Techniques Required by Deep Learning Models. The Case of the Knee MRIs. Maedica (Bucur), 19(3), 526–535.
Available at:https://pmc.ncbi.nlm.nih.gov/articles/PMC11565144./

[6] Armato, S. G., III, McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., ... & VanMeerbeeck, J. (2011). The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans. Medical Physics, 38(2), 915–931.
Available at: https://doi.org/10.1118/1.3528204.

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

[7] Beare, R., Lowekamp, B., & Yaniv, Z. (2018). Image segmentation, registration and characterisation in R with SimpleITK. Frontiers in Neuroinformatics, 12, Article 1. Available at: https://doi.org/10.3389/fninf.2018.00001

[8] Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J. C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J. V., D'Alesandro, S., Giger, M., & Kikinis, R. (2012). 3D Slicer as an image computing platform for the Quantitative Imaging Network. Magnetic Resonance Imaging, 30(9), 1323–1341. Available at: https://doi.org/10.1016/j.mri.2012.05.001.

[9] Giger, M. L. (2018). Machine learning in medical imaging. RadioGraphics, 38(6), 1712–1731. Available at: https://doi.org/10.1148/rg.2018180127.

[10] Herrmann, M. D., Clunie, D. A., Fedorov, A., Doyle, S. W,. Pieper, S., Klepeis, V., Le,L. P., Mutter, G. L., Milstone, D. S., Golden, J. A., Kikinis, R., & Lennerz, J. K. (2021). Implementing the DICOM standard for digital pathology. Journal of Pathology Informatics, 12(1), 44. Available:https://doi.org//10.4103/jpi.jpi_83_20

[11] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & Jupyter Development Team. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Structures, 87–90 . Available at doi: 10.3233/978-1-61499-649-1-87.

[12] Larobina, M., & Murino, L. (2014). Medical image file formats. Journal of Digital Imaging, 27(2), 200–206. Available at: https://doi.org/10.1007/s10278-014-9715-y.

[13] Mason, N. R., & pydicom contributors. (2020). Pydicom: An open-source Python package for working with DICOM files. Zenodo. Available at: https://zenodo.org/records/8034250.

[14] Kaur, R., & Singh, B. (2021). Contrast enhancement of medical images using CLAHE: a review. Multimed Tools and Applications, 80(2), 2415–2437.

المجلة الدّولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

http://www.doi.org/10.62341/istj-vol38-2-iregohm58

Available at: doi: 10.21037/qims.2019.08.19

[15] Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: A systematic review. BMC Medical Imaging, 15, Article 28. Available at:https://doi.org/10.1186/s12880-015-0068-x