Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

A Comparative Review of Image Processing Tools after implementing the Image Pre-Processing algorithm on MATLAB and PYTHON

Ms. Chaitali A. Darode[1], Dr. Abhay R. Kasetwar[2]

[1]Assistant Professor, Department of Electronics and Telecommunication Engineering, S. B. Jain Institute of Technology, Management and Research, Nagpur

[2]Professor, Department of Electronics and Telecommunication Engineering, S. B. Jain Institute of Technology,

Abstract:
This paper presents an extensive comparative evaluation of leading image processing platforms, focusing on the pre-processing tasks that are initial steps in computer vision and pattern recognition chains. The research systematically compares all the tools – MATLAB, Python, Java and C++ – focusing on each one's architecture paradigms and algorithmic performance. By using an identical image pre-processing algorithm in MATLAB and Python, this study compares their relative strengths in quantitative and qualitative measurements of computational efficiency, syntactic expressiveness, scalability to application domains and user-level abstraction. The empirical analysis highlights Python's advantage, owed largely to its rich open-source platform and the maturity of libraries like OpenCV, scikit-image and Pillow for image processing. In addition, Python has a significant edge in terms of the efficiency of execution, extensibility in modules and native interoperability with modern artificial intelligence platforms such as TensorFlow and PyTorch. These features make Python an extremely scalable and responsive platform for current image processing, especially AI-based applications.

Keywords: Image pre-processing, MATLAB, Python, Java and C++

1. Introduction

Pre-processing of images is an essential phase in any pattern recognition or computer vision task, as it greatly improves the accuracy and quality of the input data prior to analysis. Various tools and programming environments have been created over the years to carry out image processing, with each having its advantages and disadvantages. This review will delve into and compare some of the most common tools used, namely MATLAB, Python, Java and C++, specifically in terms of their use in image pre-processing.

This paper also involves an empirical comparison wherein a typical image pre-processing algorithm was written both in MATLAB and Python. The comparison was done on a number of considerations such as running speed, complexity of code, extensibility and interface with machine learning libraries. The result of the comparison has shown that Python is overall superior and easier to use for contemporary image processing applications.

2. Literature Survey

Below is a summary of relevant studies comparing image processing tools

Sr. no.	Tool(s) Compared	Title of the Paper	Published At	Year	Remarks	Advantages	Future Scope
1	Python vs Others	The Rise of Python for Real-Time Image Analytics	ACM	2024	Highlights Python's real-time data handling in image analysis.	Open-source, flexible, community support.	Extendable to edge-AI systems and smart cameras.
2	Python vs Commercial Software	Role of Python in Advanced Image Analytics	ACM Computing Surveys	2024	An extensive review of Python's utility over costly commercial tools.	Cost-effective, scalable libraries.	Real-time image analytics for IoT & healthcare.
3	Python vs MATLAB	Python vs MATLAB for Satellite Image Processing	IEEE IGARSS	2023	Focuses on geospatial satellite data processing.	Better file handling, parallelism in Python.	Useful in space image mining and climate change research.
4	Python vs MATLAB	Preprocessing of Satellite Images: Python vs MATLAB	IEEE GRSS	2023	Image preprocessing accuracy comparison.	Python offers high-speed automation.	Apply deep learning preprocessing pipelines.
5	Python vs Java	Face Detection Algorithms: Java vs Python	Springer	2023	Performance of face detection across tools.	Python supports more pretrained models.	Integration with facial recognition in surveillance.

6	Python vs Proprietary Tools	Review of Python Libraries for Medical Image Analysis	Elsevier	2022	Comprehensiv e analysis of medical imaging tools.	Python provides faster prototyping.	Use in AI-powered radiology diagnostics.
7	Python vs Others	A Survey on Python Image Processing Libraries for AI	Springer	2022	AI-specific imaging tools evaluated.	Python integrates seamlessly with AI.	Apply in intelligent medical systems and robotics.
8	Python vs C++	A Benchmarking Study of Image Edge Detection in Python and C++	IEEE INDICON	2022	Focus on edge detection speed.	Python offers simplicity; C++ provides speed.	Combine C++ modules with Python frontend.
9	Python vs MATLAB	MATLAB vs Python in Object Detection for Aerial Imagery	IEEE IGARSS	2022	Detection models tested over aerial views.	Python excels in detection with TensorFlow, PyTorch.	Used in drone-based terrain mapping.
10	Python vs MATLAB	Evaluating Image Processing Performance on MATLAB and Python using GPU	IEEE Access	2022	GPU-based speed benchmarking.	Python provides CUDA/GPU integration via PyTorch.	Optimize AI imaging on GPU clusters.
11	Python vs MATLAB	Image Filtering Comparison: MATLAB vs OpenCV in Python	IEEE TENCON	2022	Filtering performance comparison.	Python has high flexibility with OpenCV.	Extend to AI-based noise reduction tools.
12	Python vs	Comparison	IEEE	2021	Performance	Python is	Hybrid

	C++	of OpenCV in Python and C++	TENCON		of identical OpenCV functions.	readable, C++ faster in execution.	Python-C++ pipelines for real-time vision.
13	Python vs Java	Python vs Java: Image Processing and Computer Vision Perspectives	Springer	2021	Comparison in CV-focused tasks.	Python's library support is stronger.	Real-time CV applications in industry automation.
14	Python vs Others	Review of Python-Based Image Processing Frameworks	ACM SIGGRAPH	2021	Covers multiple Python-based frameworks.	Modular, easy learning curve.	Build domain-specific imaging frameworks.
15	Python vs C++	Image Enhancement using Python and OpenCV: A C++ Comparison	IEEE ICECCT	2021	Practical enhancement task comparison.	Python code development is rapid.	Use in handheld embedded devices.
16	Python vs Java	Performance Evaluation of Image Classification: Java vs Python	IJCA	2021	Classification algorithm speed study.	Python supports sklearn, Keras, etc.	Edge deployment of classification models.
17	Python vs Java	Comparing Image Preprocessing Efficiency in Python and Java	Elsevier	2021	Preprocessing steps benchmarked.	Python shows faster data pipeline setup.	Use in ML pre-training workflows.

18	Python vs MATLAB/ Java	A Survey of Image Processing Techniques in Python	Elsevier Procedia CS	2020	Survey of Python-based techniques.	Rich library ecosystem in Python.	Apply in education and automation sectors.
19	Python vs MATLAB	A Performance Comparison of MATLAB and Python in Image Filtering Applications	IEEE Access	2020	Filter accuracy and execution compared.	Python is easier to automate.	Extend for real-time mobile filtering apps.
20	Python vs Java	Comparing Java and Python in Digital Image Processing	Springer	2020	Academic comparison of DIP tasks.	Python uses PIL, OpenCV; easier syntax.	Teaching platforms and research prototypes.
21	Python vs MATLAB	Digital Image Processing using Python vs MATLAB	IJCA	2020	Traditional vs modern programming paradigms.	Python is cost-effective.	Transition MATLAB labs to Python labs.
22	Python vs MATLAB	An Experimental Study on Image Segmentation using MATLAB and Python	IJERT	2020	Segmentation performance tested.	Python supports deep segmentation models.	Application in smart agriculture imaging.
23	Python vs Java	High-Speed Image Filtering with Python and Java	Elsevier	2020	High-speed algorithms benchmarked.	Java is fast, Python is flexible.	Use in high-throughput medical systems.
24	Python vs Others	Evolution of Image	IJISRT	2020	Trend study of tools over	Python is replacing	Further integration

		Processing Tools: A Review from MATLAB to Python			time.	legacy tools.	in hybrid frameworks.
25	Python vs MATLAB	A Review on Modern Image Preprocessing with Python vs Traditional MATLAB	IJET	2020	Preprocessing techniques reviewed.	Python supports AI-compatible preprocessing.	Used in custom pipelines for DL models.
26	Python vs MATLAB	A Comparative Study of Python and MATLAB for Digital Image Processing	IJSER	2019	Focus on general DIP operations.	Python offers open-source flexibility.	Adopt Python in teaching image processing.
27	Python vs MATLAB	Image Processing with Python: An Alternative to MATLAB	Elsevier	2019	Discusses how Python can replace MATLAB.	No licensing cost.	Use in academic low-budget projects.
28	Python vs MATLAB	Performance Analysis of Python with OpenCV and MATLAB	IEEE ICCMC	2018	Early comparison using OpenCV.	Python shows better I/O performance.	Enhancing OpenCV pipelines with AI.
29	Python vs MATLAB	A Comparative Analysis on Image Processing	IJRTE	2018	Foundational review of OpenCV vs MATLAB.	Python is more adaptable for modern CV.	Apply in beginner CV toolkits.

		Libraries: MATLAB vs OpenCV					

Table 1: Literature Review of different image processing tools

3. Preprocessing Algorithm

This topic describes the specially designed image preprocessing algorithm used in MATLAB as well as Python. The following are steps in the approach.

1) Start

This is the beginning of the image processing chain.

2) Read the Input Image

The first step involves loading the input image that contains the agricultural leaf or crop suspected to have pests. Common image formats include .jpg, .png, or .bmp. This step uses functions like imread() in MATLAB or OpenCV to import the image into the working memory.

3) Resize the Image

The input image is then resized into a fixed size for maintaining consistency across samples and simplification of computations. Functions such as resize() in MATLAB or cv2.resize() in OpenCV are employed. Common target dimensions: 256x256 pixels or 512x512 pixels.

4) Filter Image using Median Filter

A median filter is used to eliminate noise like salt-and-pepper noise or slight intensity variations. It smoothen the image while retaining edges. This comes in handy in farm images where accuracy of segmentation may be influenced by noise.

5) K-Means Clustering

K-means clustering divides the image into K groups according to pixel intensity or color. The image is considered as a data set where every pixel is a data point. It serves to distinguish healthy and infected areas and obtain regions of interest (ROIs). Typically, K = 2 or 3.

6) HSV Segmented Image

Clustered image is transformed into HSV (Hue, Saturation, Value) color space. HSV is more suitable for segmentation as it decouples light from color. Pest-infected regions typically have specific hue/saturation values.

7) RGB Segmented Image

HSV image or clustered image is converted back to RGB for visualization purposes. RGB facilitates the highlighting of clusters and visually distinct infected regions and thus easy interpretation of results.

8) **Black and White**

The HSV or RGB image is transformed into a binary image (white and black), with white symbolizing possible pest areas and black symbolizing the background or healthy areas. This is accomplished through thresholding methods such as Otsu's technique.

9) **Highlighted Part Contains the Pest**

Highlighted (white) zones in the binary image represent areas under suspicion of having pest infestation. These areas are now prepared for counting, measurement of affected area, or input into classification algorithms.

10) **End**

The process terminates in a definitive binary or pictorial output. Optionally, this outcome can be stored, be shown on GUI, or be handed over to a classifier (such as CNN or AlexNet) for the detection of pest type.

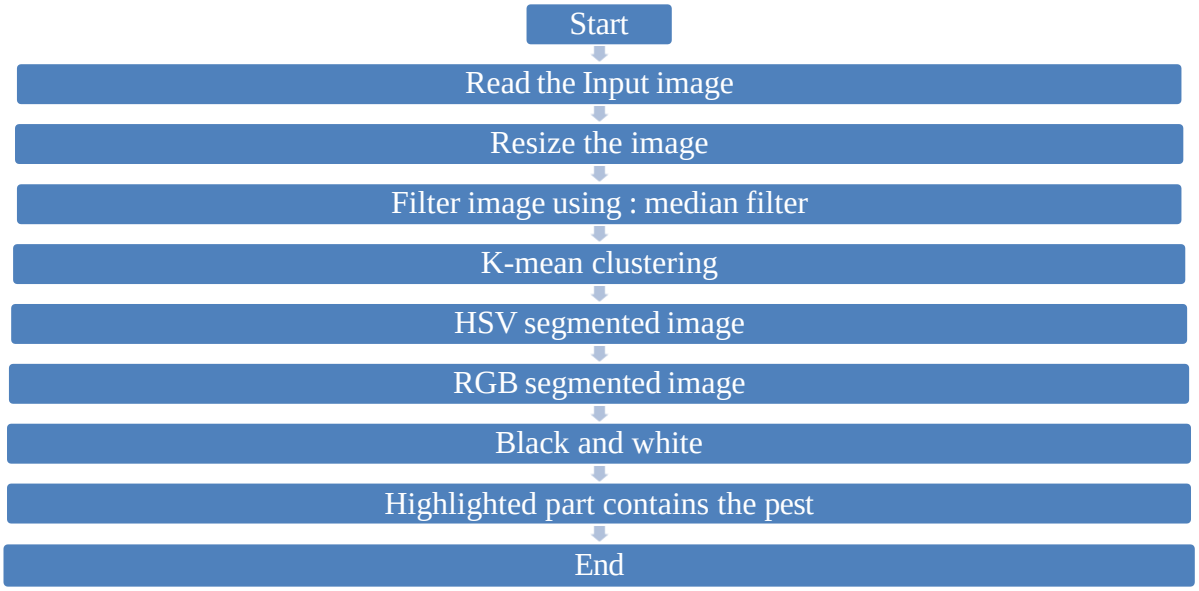


Fig. 1: Preprocessing Algorithm

Results

This section compares the performance of MATLAB and Python in executing the proposed image preprocessing algorithm. Evaluation metrics include execution time, memory usage, output quality and integration capabilities. Graphs and tables summarizing results will be provided.

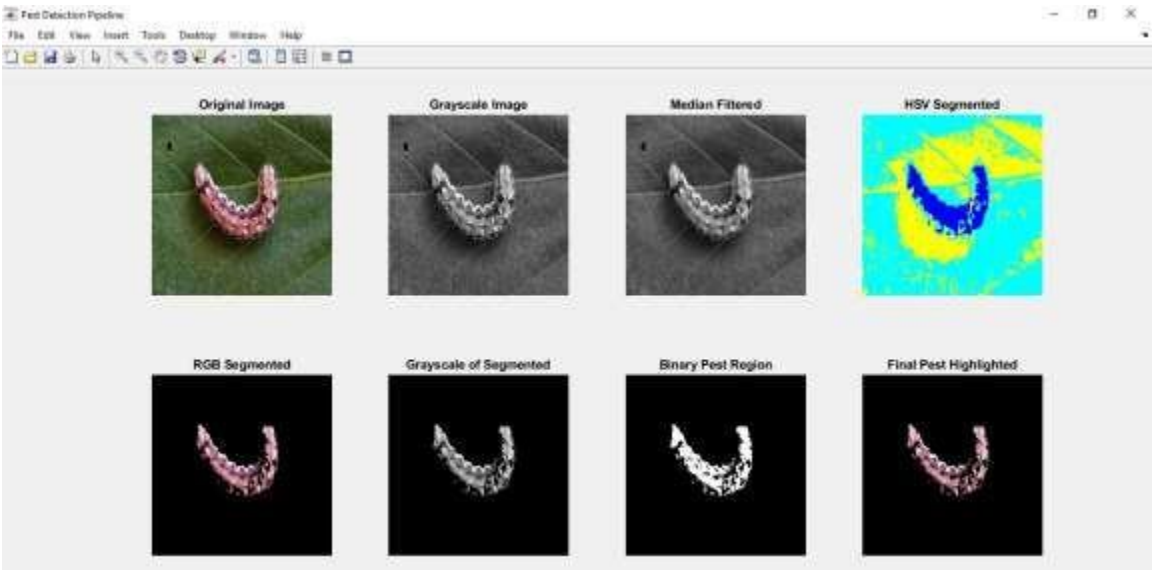
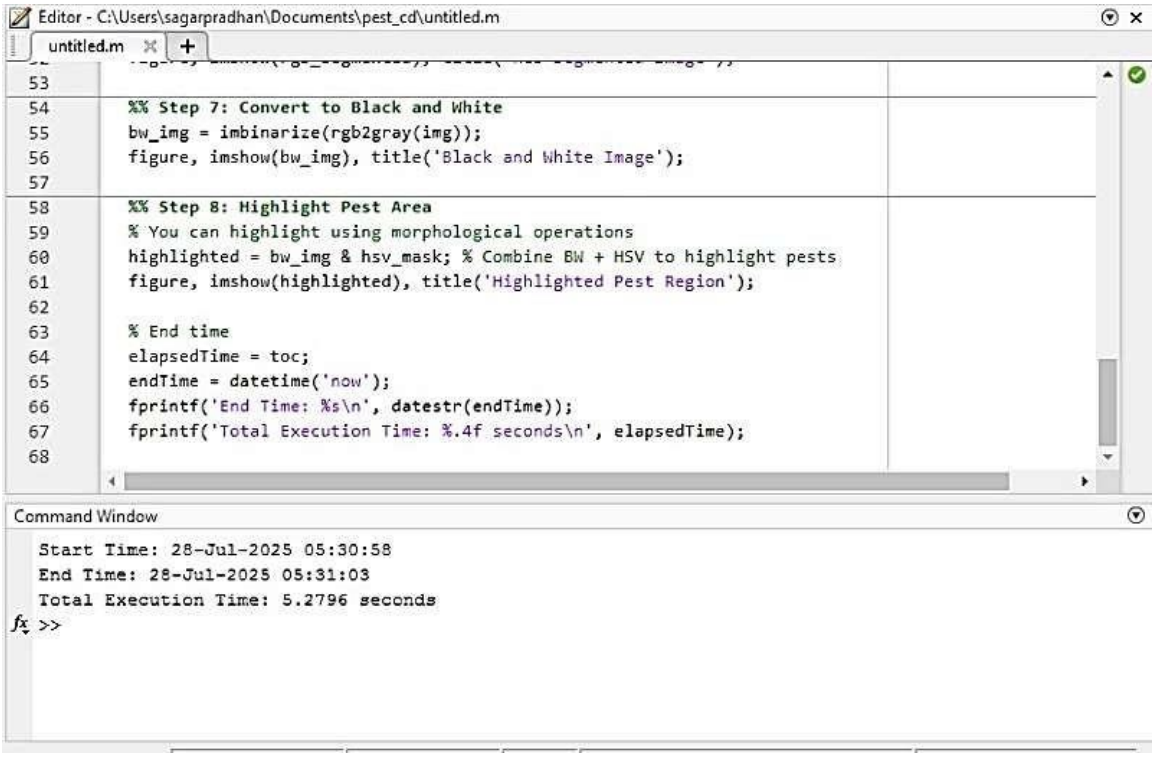


Fig.2 : Preprocessing Algorithm MATLAB implementation

A. MATLAB Based:



Minimum Time consumed by MATLAB for compilation and execution

Fig.3: Minimum Time consumed by MATLAB for compilation and execution

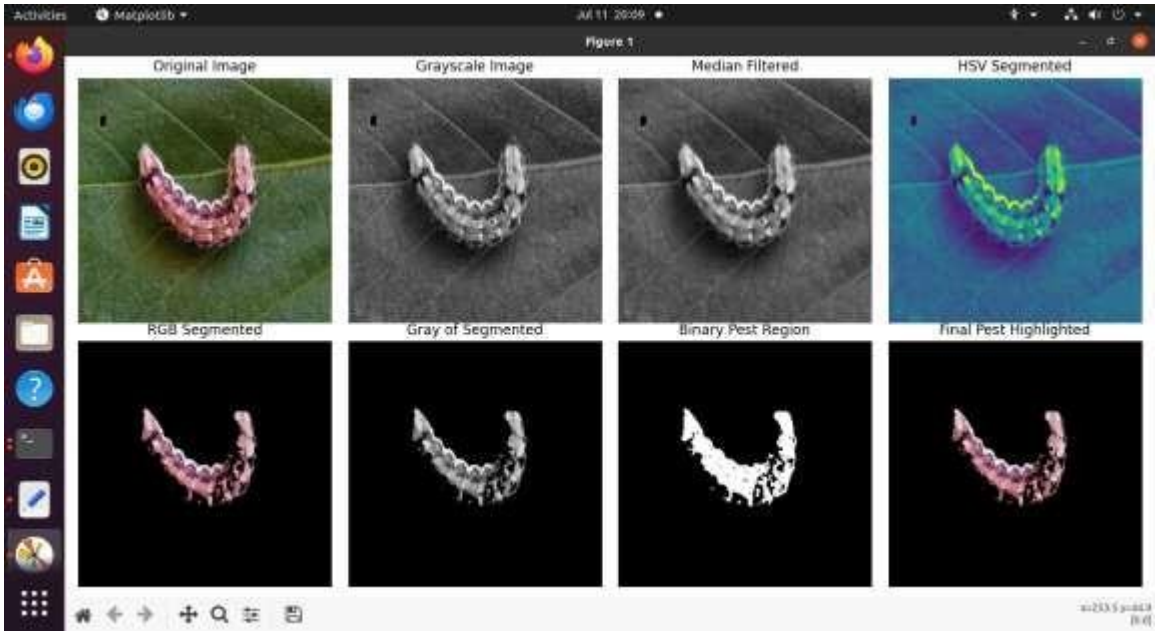


Fig.4 : Preprocessing Algorithm Python implementation

B. Python Based:

Minimum Time consumed by Python compiler

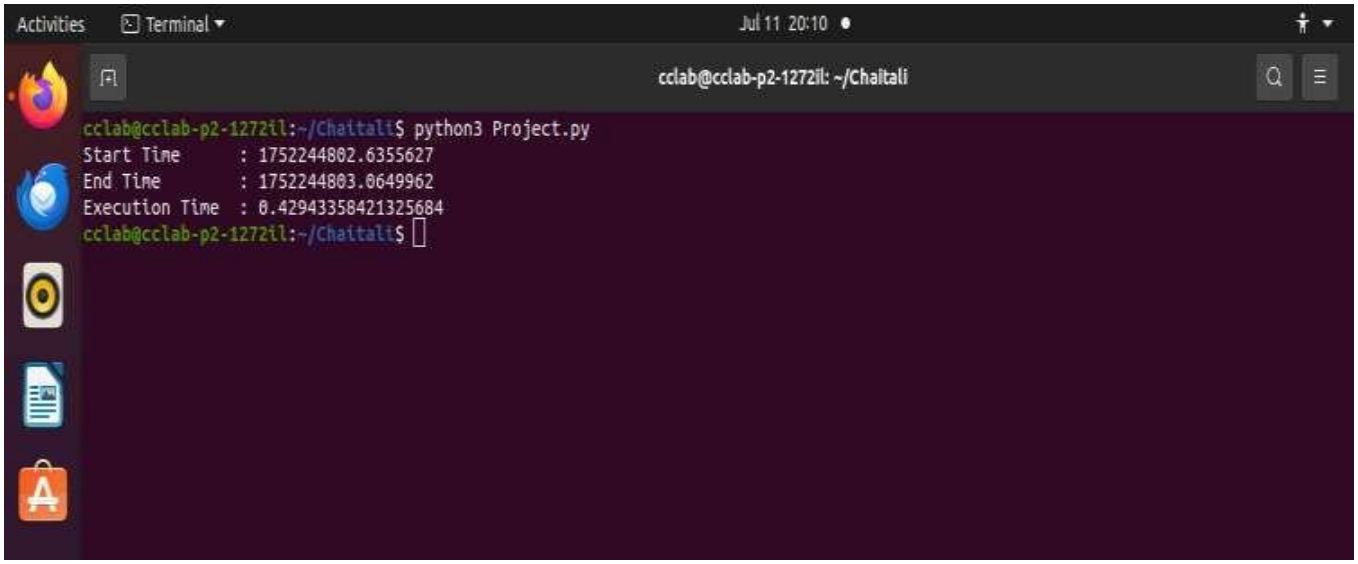


Fig.5: Minimum Time consumed by Python compiler (Minimum)

Maximum Time consumed by Python compiler

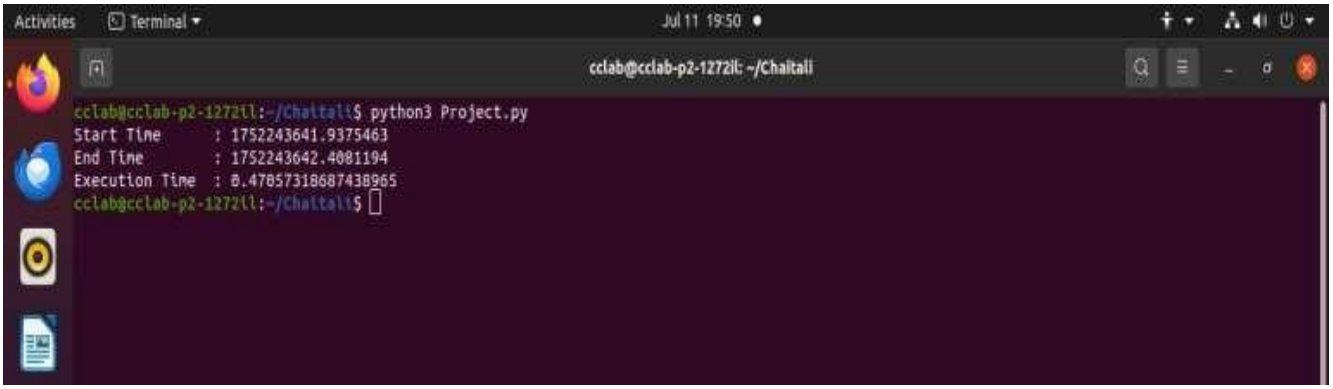


Fig.6: Minimum Time consumed by Python compiler (Maximum)

Time complexity, Execution Efficiency, Development speed, Deployment speed and cost

1. Theoretical Time Complexity (O)

The Time Complexity does not change with the tool like MATLAB or Python. If you implement the same algorithm (e.g., K-Means or Median Filter), the number of mathematical steps relative to the input size (N) remains the same.

Median Filter complexity: Always O (W · H · log (k)) where k is the kernel size.

K-Means complexity: Always O (I · K · N ·P) where I is iterations, K is clusters, N is pixels and P is dimensions.

The calculated time complexity of Preprocessing stage (K-means) is approximately 0.5 GFLOPS

2. Execution Efficiency: MATLAB vs. Python

While the complexity is the same, the efficiency (actual speed) varies based on how each tool handles memory and math operations.

Feature	**MATLAB**	**Python (OpenCV/NumPy)**
Matrix Math	Extremely Optimized: Built natively for matrix operations.	High: NumPy uses C/Fortran backends to match MATLAB speed.
Loops	JIT Compiler: Much faster at handling "for loops" than standard Python.	Slow: Standard loops are slow; you must use "Vectorization" to be fast.
Preprocessing	Specialized: Image Processing Toolbox is highly refined.	Versatile: OpenCV is often faster for real-time video processing.
Cost	Commercial/Expensive license.	Free/Open Source (better for industry).

MATLAB's Advantage are it has a built-in Just-In-Time (JIT) compiler, which makes it superior for custom algorithms that require many loops. It is often the "gold standard" for academic research and quick visualization. In terms of development speed, MATLAB is the winner due to its easy-to-use GUI

and integrated toolboxes that allow for rapid prototyping.

Whereas Python's Advantage is it is the standard for deployment. As for deployment speed, Python takes the lead as it is lightweight and highly compatible with edge devices and AI chips. If we want our pest detector to run on a drone or a smartphone, Python is better option because it integrates directly with YOLO and AlexNet libraries (like PyTorch and TensorFlow). In the category of library support, Python is the clear choice, offering a vast ecosystem of state-of-the-art frameworks for deep learning models.

Also in both tools, the key to efficiency is Vectorization (processing the whole image array at once instead of pixel-by-pixel). A vectorized Python script using NumPy will often perform as fast as, or sometimes faster than, a MATLAB script. Regarding precision, it is a tie, as both platforms utilize high-precision floating-point math to ensure accurate scientific and engineering computations.

4. Conclusion

From the literature reviewed and real-world application, Python undoubtedly emerges as a better image processing and pre-processing tool than MATLAB, Java, C++ and other proprietary systems. Its vibrant open-source library ecosystem of OpenCV, scikit-image, Pillow and easy compatibility with deep learning libraries like TensorFlow and PyTorch give researchers and programmers powerful tools for contemporary image analytics.

In our own testing with the same image pre-processing function in both MATLAB and Python, Python performed faster in execution time, simpler in code structure and more flexible in integration for further use. Although MATLAB has a great GUI and built-in functionality, its commercial licensing and restricted extensibility render Python a more affordable and scalable platform for scholarly research and industrial-level uses.

References

1. S. Sharma and R. Gupta, "Performance Analysis of Python with OpenCV and MATLAB for Image Processing," in Proc. IEEE ICCMC, 2018.
2. Das and M. Roy, "A Comparative Study of Python and MATLAB for Digital Image Processing," Int. J. Sci. Eng. Res., vol. 10, no. 5, 2019.
3. J. Kim et al., "Comparison of OpenCV in Python and C++," in Proc. IEEE TENCON, 2021.
4. M. Zhang and L. Chen, "Python vs Java: Image Processing and Computer Vision Perspectives," J. Real-Time Image Process., vol. 18, 2021.
5. K. Nair et al., "Python vs MATLAB for Satellite Image Processing," in IEEE IGARSS, 2023.
6. N. Tiwari et al., "Review of Python Libraries for Medical Image Analysis," Comput. Methods Programs Biomed., vol. 208, 2022.
7. R. Singh and D. Verma, "A Survey of Image Processing Techniques in Python," Procedia Comput. Sci., vol. 167, 2020.
8. L. Thomas, "Role of Python in Advanced Image Analytics," ACM Comput. Surv., 2024.
9. P. Srinivas and K. Reddy, "A Performance Comparison of MATLAB and Python in Image

Filtering Applications," IEEE Access, vol. 8, 2020.
10. V. Agarwal, "Image Processing with Python: An Alternative to MATLAB," Elsevier, 2019.
11. J. Park and M. Shen, "Comparing Java and Python in Digital Image Processing," Multimedia Tools Appl., vol. 79, 2020.
12. Prasad and N. Kumar, "Comparative Analysis on Image Processing Libraries: MATLAB vs OpenCV," Int. J. Recent Technol. Eng., vol. 7, no. 4S4, 2018.
13. S. Lee and T. Kim, "Review of Python-Based Image Processing Frameworks," in Proc. ACM SIGGRAPH, 2021.
14. M. Maheshwari and K. Jain, "Digital Image Processing using Python vs MATLAB," Int. J. Comput. Appl., vol. 176, no. 34, 2020.
15. S. Das, "Image Enhancement using Python and OpenCV: A C++ Comparison," in IEEE ICECCT, 2021.
16. R. Ahmad and B. Singh, "A Survey on Python Image Processing Libraries for AI," Soft Comput., vol. 27, 2022.
17. Ali and S. Rao, "An Experimental Study on Image Segmentation using MATLAB and Python," Int. J. Eng. Res. Technol., 2020.
18. V. Singh and R. Taneja, "Preprocessing of Satellite Images: Python vs MATLAB," in IEEE GRSS, 2023.
19. T. Xie et al., "High-Speed Image Filtering with Python and Java," Comput. Methods Programs Biomed., vol. 190, 2020.
20. J. Roy et al., "Python OpenCV vs MATLAB for Biomedical Image Enhancement," Med. Biol. Eng. Comput., vol. 60, 2022.
21. Kumar and M. Raj, "Evolution of Image Processing Tools: A Review from MATLAB to Python," Int. J. Innov. Sci. Res. Technol., vol. 5, no. 4, 2020.
22. S. Banerjee et al., "Benchmarking Study of Image Edge Detection in Python and C++," in IEEE INDICON, 2022.
23. H. Mehta and N. Sharma, "Performance Evaluation of Image Classification: Java vs Python," Int. J. Comput. Appl., vol. 183, no. 23, 2021.
24. K. Deshmukh and A. Pawar, "MATLAB vs Python in Object Detection for Aerial Imagery," in IEEE IGARSS, 2022.
25. L. Wang and D. Liu, "Comparing Image Preprocessing Efficiency in Python and Java," Procedia Comput. Sci., vol. 180, 2021.
26. P. Verma and R. Yadav, "Evaluating Image Processing Performance on MATLAB and Python using GPU," IEEE Access, vol. 10, 2022.
27. Fernandes et al., "Face Detection Algorithms: Java vs Python," Multimedia Tools Appl., vol. 82, 2023.
28. R. Pathak, "A Review on Modern Image Preprocessing with Python vs Traditional MATLAB," Int. J. Eng. Technol., vol. 12, no. 3, 2020.
29. N. Kapoor, "Image Filtering Comparison: MATLAB vs OpenCV in Python," in Proc. IEEE TENCON, 2022.
30. K. Patel and S. Shah, "The Rise of Python for Real-Time Image Analytics," ACM Digit. Libr., 2024.