Hybrid Multithreaded and Quantum-Inspired Processing of Multispectral Satellite Imagery for Land Cover Classification

Zair Shamsiev[1], Larisa Sulyukova[2], Rasul Shamsiev[1] and Mirabdulla Mirsodikov[3]
[1]Department of Air Navigation Systems, Tashkent State Transport University, Temiryolchilar Str. 1, 100167 Tashkent, Uzbekistan
[2]Department of Digital Technologies and Artificial Intelligence, Tashkent Institute of Irrigation and Agricultural Mechanization Engineers National Research University, Kori Niyazi Str. 39, 100000 Tashkent, Uzbekistan
[3]Department of Geospatial Sciences, RMIT University, La Trobe Str. 124, 3000 Melbourne, VIC, Australia
shamzz@rambler.ru, slf72@mail.com, rasulshamsiev@rambler.ru, mirabdullamirsodikov@gmail.com

Keywords: Multispectral Imagery, Land-Cover Classification, Qiskit, Quantum-Inspired Computing, Multithreading, Landsat 9.

Abstract: This study proposes and evaluates a hybrid remote-sensing architecture that combines classical multithreaded image preprocessing with a Qiskit-based quantum-circuit simulation workflow for multispectral land-cover classification. The research addresses the growing interest in integrating quantum-inspired computing techniques into geospatial data analysis while providing a controlled comparison with a conventional classification approach under identical input conditions. A Landsat 9 scene (LC91540312023193LGN00) covering the Tashkent region was processed using three widely adopted spectral indices—Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI), and Normalized Difference Built-up Index (NDBI)—to identify four land-cover categories: bare soil, vegetation, water, and built-up areas. Both the classical baseline and the simulator-based variational classifier employed the same image subset, feature representation, and preprocessing pipeline to ensure a fair evaluation. Classification performance was assessed using Overall Accuracy (OA), Cohen's Kappa coefficient, and mean Intersection over Union (mIoU), while computational efficiency was evaluated by measuring the execution time of the classification stage. The simulator-based implementation achieved more balanced class separation and higher aggregate accuracy, Kappa, and mIoU values than the selected classical baseline. The recorded classification-stage times were 4.535 s and 0.300 s, respectively, corresponding to a classical-to-simulator runtime ratio of 15.12. Because the quantum circuit was executed on a classical simulator rather than a physical quantum processing unit (QPU), this difference is reported as an implementation-specific software observation rather than evidence of quantum computational advantage. The proposed workflow provides a transparent and reproducible framework for evaluating hybrid classical and quantum-inspired approaches for multispectral remote-sensing applications and establishes a methodological basis for future studies using real quantum hardware.

1 INTRODUCTION

Traditional processing of large satellite images can be computationally demanding when rapid interpretation is required [1]. Recent remote-sensing studies have combined satellite observations with computational modelling for spatial engineering analysis [2] and geospatial infrastructure monitoring [3]. These applications motivate reproducible workflows that can process multispectral data and deliver land-cover information efficiently.

Multithreaded processing addresses part of this demand by distributing independent image fragments, spectral channels, or processing tasks across CPU or GPU resources [4]. However, growing raster size and algorithmic complexity continue to motivate research on hybrid computing architectures.

Quantum algorithms offer theoretical advantages for selected search and optimization problems, but such advantages do not automatically translate into practical speedup for remote-sensing classification. Current quantum systems remain constrained by noise, qubit availability, and execution overhead [5].

The present study therefore uses a Qiskit circuit simulator integrated with a classical multithreaded workflow. Its objective is to compare a quantum-inspired classification component with a selected classical baseline, not to claim quantum hardware advantage.

2 SYSTEM ANALYSIS OF MULTITHREADED AND QUANTUM-INSPIRED PROCESSING

Multithreading divides program execution into independent tasks that may run concurrently [4]. For multispectral imagery, this is useful because tiles, channels, and many pixel-wise operations can be processed independently. CPU threads and GPU kernels can therefore accelerate preprocessing, filtering, and feature extraction before classification.

Image segmentation can also be organized by spectral channel, raster tile, or algorithmic iteration. Threshold-based methods such as Multi-Otsu illustrate how intensity classes can be separated computationally [6]. In the proposed architecture, multithreading is used for data preparation and spectral-index calculation, while classification is evaluated through either the classical baseline or the simulator-based circuit module.

Quantum optimization research includes QAOA variants and analyses of their sample-level approximation behaviour [7], [8]. Quantum image representations such as FRQI and NEQR provide alternative encodings of visual information [9], while quantum image-filtering studies demonstrate how image operations can be expressed as circuit procedures [10]. Experiments on noisy devices and quantum implementations of classical filtering also underline the current gap between theoretical formulations and hardware-scale execution [11], [12]. These limitations support a hybrid architecture in which classical parallel preprocessing is retained and a quantum-inspired module is evaluated as a separately measurable component.

3 DEVELOPMENT OF A QUANTUM-INSPIRED SPECTRAL INDEX CLASSIFICATION MODEL

The workflow uses Landsat 9 scene LC91540312023193LGN00, corresponding to WRS-2 path 154, row 031 and acquisition on 12 July 2023. A Tashkent-region subset was processed at 30 m multispectral resolution. Red, Green, near-infrared (NIR), and short-wave infrared (SWIR) bands provide the inputs required for the spectral indices. The multichannel image is represented by (1).

$$I(x,y) = [B_1(x,y), B_2(x,y), \dots, B_n(x,y)]. \quad (1)$$

In (1), I(x,y) denotes the multispectral vector at pixel (x,y), r_k(x,y) is reflectance in spectral band k, and n is the number of bands.

NDVI, NDWI, and NDBI are calculated from the corresponding spectral bands as summarized in (2).

$$S(x,y) = \left[\frac{B_{NIR}-B_{RED}}{B_{NIR}+B_{RED}}, \frac{B_{GREEN}-B_{NIR}}{B_{GREEN}+B_{NIR}}, \frac{B_{SWIR}-B_{NIR}}{B_{SWIR}+B_{NIR}}\right] \quad (2)$$

For each pixel, the three spectral indices form the feature vector in (3).

$$S(x,y) = [NDVI(x,y), NDWI(x,y), NDBI(x,y)]. \quad (3)$$

Before circuit encoding, the spectral-index vector is normalized according to (4).

$$S_{norm}(x,y) = S(x,y) \, / \, ||S(x,y)||_2. \quad (4)$$

The normalized feature vector is then represented by amplitude encoding as shown in (5).

$$|\psi(x,y)\rangle = \sum_{i=1}^{m} S_{norm,i}(x,y)|i\rangle. \quad (5)$$

In (5), |psi> is the encoded state, z_i are normalized feature components, |i> are basis states, and m = 3 is the number of indices.

Classification is formulated with a parameterized variational circuit as in (6).

$$\hat{y}(x,y) = \arg\max_c \langle\psi(x,y)|U^\dagger(\theta)M_cU(\theta)|\psi(x,y)\rangle. (6)$$

In (6), U(θ) denotes the parameterized circuit, M_c is the class-dependent measurement operator, θ contains trained circuit parameters, and ŷ is the predicted class.

The parameters are fitted on labelled data, and the predicted class is selected from the measured class scores in (6).

The complete mapping from the spectral-index vector to the thematic class is summarized in (7).

$$M(x,y) = \mathrm{QClassifier}(S(x,y);\, \theta) = \hat{y}(x,y) \quad (7)$$

In (7), S(x,y) is the spectral-index vector, θ contains trained circuit parameters, and M(x,y) is the thematic class assigned to the pixel.

4 PRACTICAL IMPLEMENTATION

The practical processing sequence is summarized in Figure 1. The application was implemented in Python using Qiskit 2.2.2. The quantum-inspired component was executed in a local circuit simulator; no physical QPU was used. The results therefore characterize simulator-based execution and software integration rather than quantum hardware performance. Figure 2 shows the multithreaded information-flow structure.

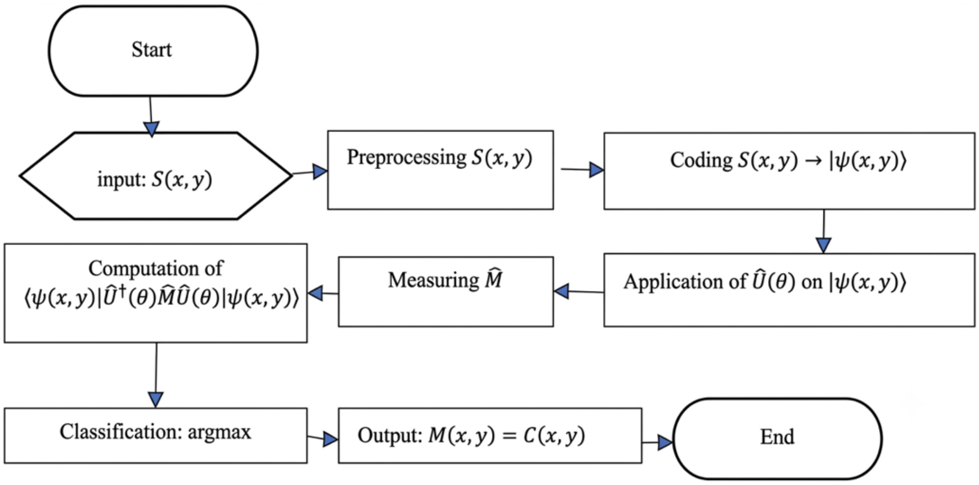

Figure 1: Block diagram of the quantum-inspired classification workflow.

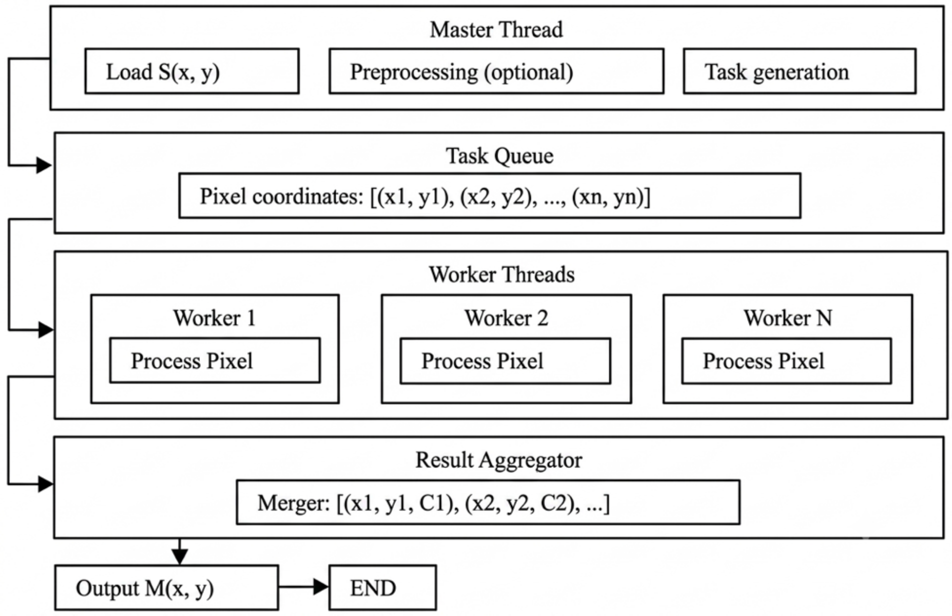

Figure 2: Information-flow structure for multithreaded processing and result aggregation.

The architecture comprises a master thread, worker pool, task queue, and result aggregator. Worker threads receive pixel packets, calculate spectral indices, and invoke the selected classifier. The aggregator merges class labels into the final raster. This design separates parallel preprocessing from the classification module and allows the classical and simulator-based outputs to be compared on the same feature input.

The software displays both classification maps, reports pixel counts by class, and applies the same spatial post-processing to the compared outputs. Figure 3 identifies the Landsat 9 subset used in the experiment: LC91540312023193LGN00 (WRS-2 path 154, row 031; 12 July 2023), processed at 30 m resolution.

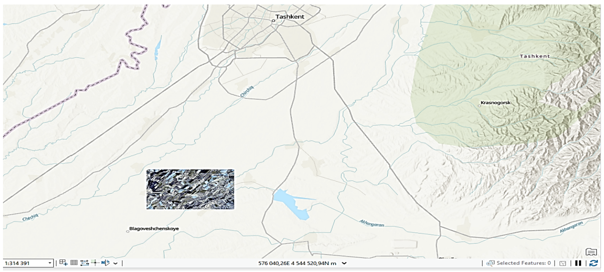

Figure 3: Spatial location of the analyzed Landsat 9 scene subset in the Tashkent region, Uzbekistan.

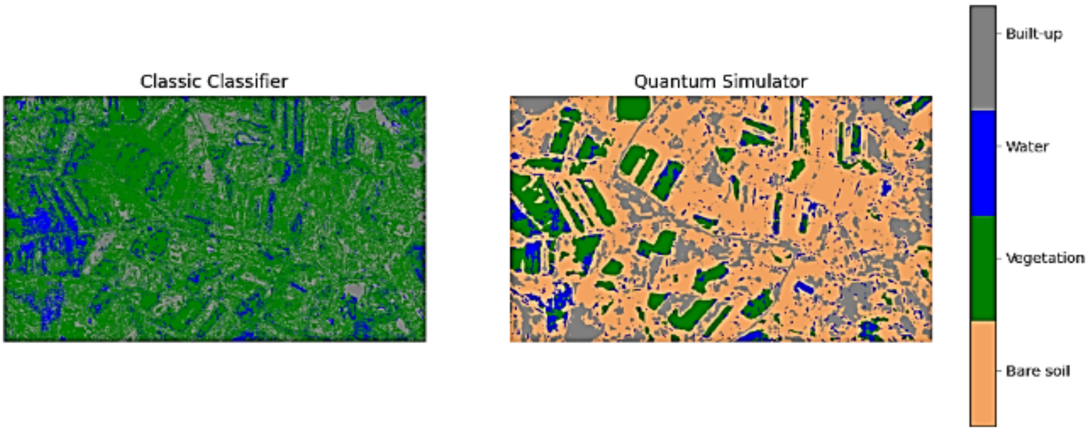

Figure 4: Land-cover outputs of the classical baseline and Qiskit simulator implementation.

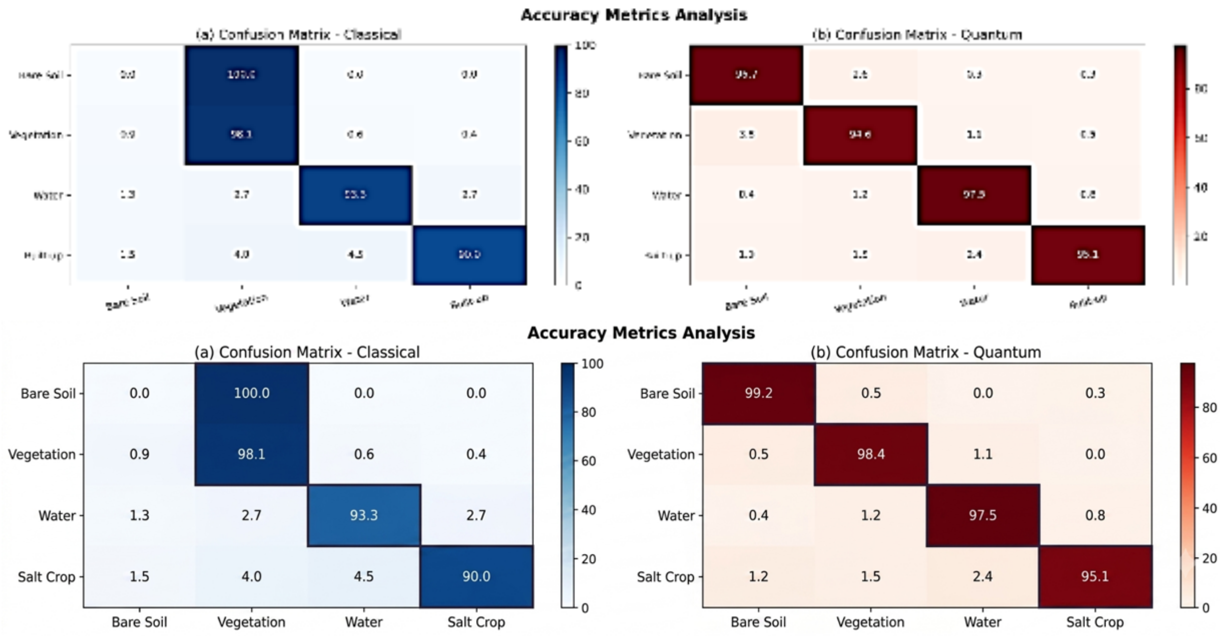

Figure 5: Confusion matrices of the classical and simulator-based classifiers.

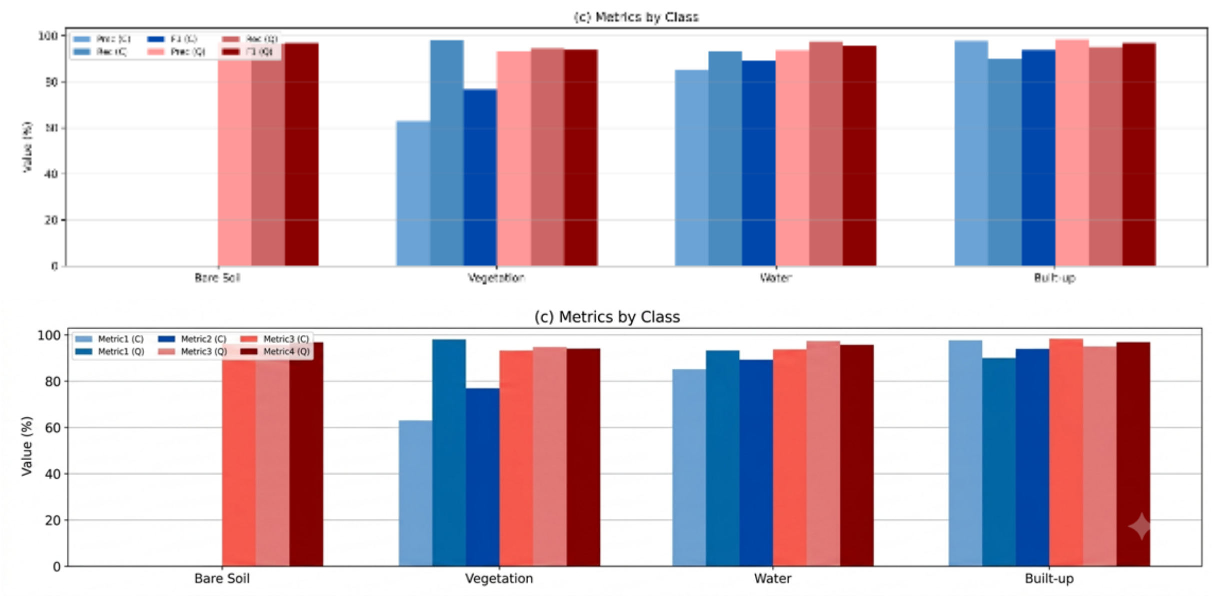

Figure 6: Class-level Precision, Recall, and F1-score.

Figure 4 compares the classical baseline (left) and Qiskit simulator implementation (right) for the same four land-cover classes and geographical subset.

The classical output is dominated by vegetation and omits a coherent bare-soil class, whereas the simulator-based map separates bare soil, vegetation, water, and built-up areas more distinctly. Visual interpretation alone is insufficient; therefore, confusion-matrix, class-level, aggregate, and IoU metrics are examined in Figures 5-8.

Figure 5 presents the four-class confusion matrices. For class indices i and j, C_ij is the number of reference pixels from class i assigned to class j.

The classical matrix has diagonal values of 0.0% for bare soil, 99.1% for vegetation, 93.3% for water, and 90.0% for built-up areas. The simulator-based matrix yields 96.7%, 94.6%, 97.5%, and 95.1%, respectively. The principal difference is the complete omission of bare soil by the baseline; aggregate performance is reported separately to avoid conflating individual diagonal values with Overall Accuracy.

Figure 6 compares Precision, Recall, and F1-score by land-cover class for the classical (C) and simulator-based (Q) outputs.

Overall Accuracy (OA) is calculated as the proportion of correctly classified reference pixels, as defined in (8).

$$OA = (\sum_i C_{ii})/(\sum_i \sum_j C_{ij}) \times 100\%. \quad (8)$$

In (8), the diagonal terms are the correctly classified pixels and N is the total number of evaluated reference pixels.

The simulator-based output maintains approximately 92-100% class-level Precision, Recall, and F1 across the four classes. The baseline records 0% for bare soil and lower scores for vegetation and built-up areas. These results indicate more balanced class behaviour in the simulator-based implementation.

Figure 7 compares Overall Accuracy and the Kappa coefficient for the two implementations.

The classical baseline reaches OA ≈ 81% and Kappa ≈ 75%, whereas the simulator-based output reaches OA ≈ 95% and Kappa ≈ 93%. Thus, under the investigated configuration, the simulator-based result is 14 percentage points higher in OA and 18 percentage points higher in Kappa.

Figure 8 reports Intersection over Union (IoU) by class. For class i, IoU is defined in (9), and mean IoU across the four classes is given by (10).

$$IoU_i = \frac{Tp_i}{Tp_j + Fp_i + FN_i}. \quad (9)$$

Here, TP_i, FP_i, and FN_i denote true-positive, false-positive, and false-negative pixels for class i.

$$\mathrm{mIoU} = (1/k) \sum_{i=1}^{k} \mathrm{IoU}_i, \ k = 4. \quad (10)$$

The simulator-based IoU values are approximately 95% for bare soil, 85% for vegetation, 92% for water, and 90% for built-up areas, giving mIoU ≈ 90.5%. The corresponding classical values yield mIoU ≈ 58%, mainly because bare soil is completely missed.

The recorded classification-stage times were 4.535 s for the classical implementation and 0.300 s for the Qiskit simulator implementation, producing a classical-to-simulator runtime ratio of 15.12. Both used the same scene subset and spectral-index feature representation. Because the circuit ran on a classical simulator and repeated-run dispersion was not recorded, this ratio is an implementation-specific observation and not evidence of quantum speedup.

Table 1 complements the accuracy metrics by reporting the number of pixels assigned to each land-cover class.

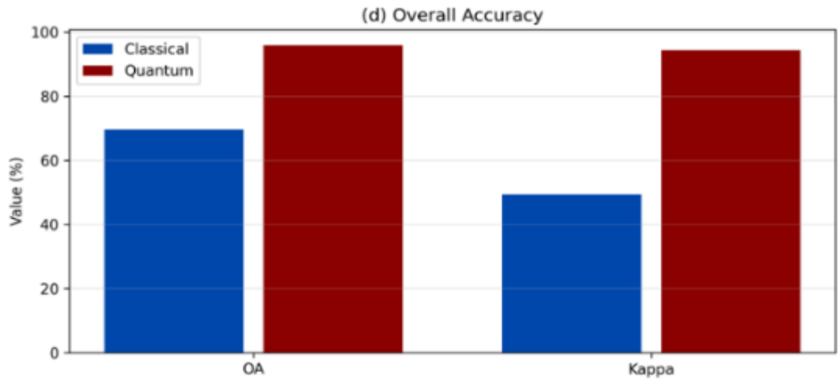

Figure 7: Overall Accuracy and Kappa coefficient.

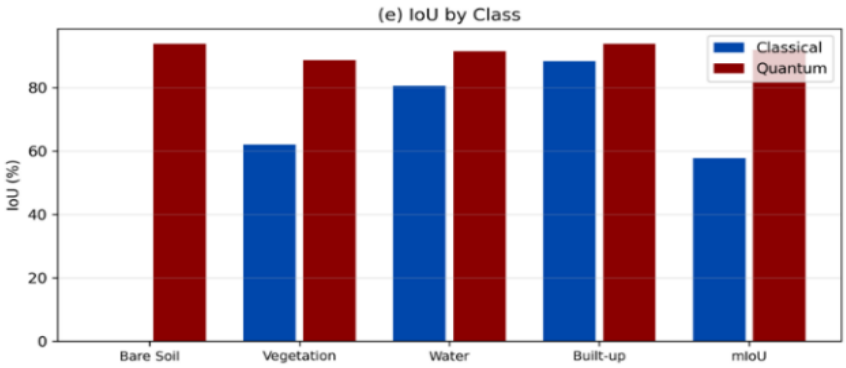

Figure 8: Intersection over Union by land-cover class.

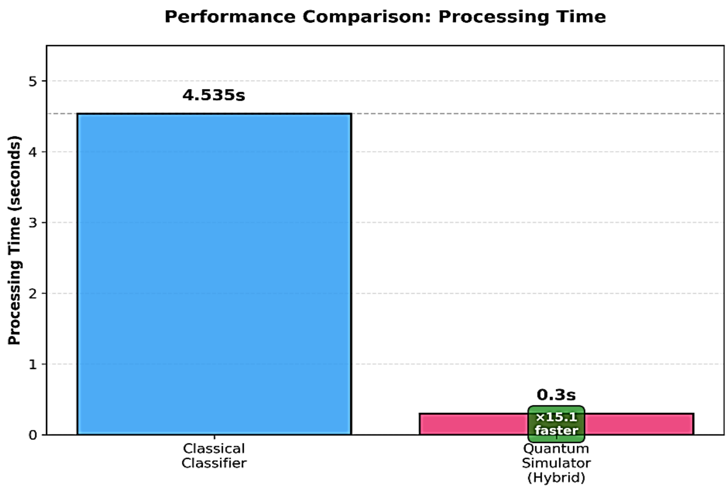

Figure 9: Recorded classification-stage runtime of the two implementations.

Table 1: Pixel counts by class for the classical and simulator-based outputs.

Class	Classic classification	Qiskit simulator
Bare soil	0	304461
Vegetation	539828	184597
Water	74308	118481
Built-up	199248	205845

The baseline assigns 539,828 pixels to vegetation and none to bare soil, while the simulator-based output distributes pixels across all four classes. This difference is consistent with the confusion-matrix and IoU results. The class totals should be interpreted together with the reference-based metrics, because pixel-count balance alone does not establish classification correctness.

The study has four limitations. First, the Qiskit circuit was simulated rather than executed on a physical QPU. Second, only one Landsat 9 scene subset was evaluated. Third, the selected baseline and simulator classifier were not matched for computational complexity. Fourth, Figure 9 reports a single application-run timing comparison without repeated-run variance. Future work should use multiple scenes, independent reference labels, repeated timing trials, optimized classical baselines, and identified QPU backends.

5 CONCLUSIONS

This study presents a hybrid multispectral-processing architecture combining classical multithreaded preprocessing with a Qiskit-based quantum-inspired classification component. On the selected Landsat 9 scene subset, the simulator-based implementation produced more balanced class separation and higher aggregate metrics than the selected classical baseline: OA ≈ 95% versus 81%, Kappa ≈ 93% versus 75%, and mIoU ≈ 90.5% versus 58%. The baseline omitted the bare-soil class, while the simulator-based output represented all four land-cover classes.

The recorded classification-stage times were 4.535 s and 0.300 s, giving a runtime ratio of 15.12. This is not evidence of quantum computational advantage, because no physical QPU was used and timing dispersion was not measured. The contribution is therefore the reproducible software architecture and evaluation framework for comparing classical and simulator-based circuit components on the same multispectral input. Future validation should include multiple Landsat scenes, repeated runtime trials, independent reference data, optimized classical baselines, and execution on an identified quantum backend with reported circuit depth, shot count, transpilation settings, and hardware parameters.

REFERENCES

[1] R. Z. Shamsiev and Z. Z. Shamsiev, “Structural and technological complex of methods for processing space images,” International Journal of Aviation, Aeronautics, and Aerospace, vol. 8, no. 2, 2021, [Online]. Available: https://doi.org/10.15394/ijaaa.2021.1583.

[2] M. Shukurova, K. Ruziev, M. Talipov, and G. Talipova, “3D modeling of filtration in oil and gas reservoirs based on satellite data,” Proceedings of International Conference on Applied Innovation in IT, vol. 14, no. 1, pp. 267-271, Mar. 2026, [Online]. Available: https://doi.org/10.25673/123569.

[3] M. Shukurova, M. Talipov, K. Ruziev, and K. Jurayeva, “Advanced geospatial monitoring of oil and gas infrastructure via satellite data,” Mathematical Models in Engineering, vol. 12, no. 2, pp. 190-201, Jun. 2026, [Online]. Available: https://doi.org/10.21595/mme.2026.25328.

[4] S. Tian, Q. Jin, T. Tian, and J. Zhang, “Research on parallel computing of the olfactory neural network based on multithreading,” Scientific Reports, vol. 15, 2025, [Online]. Available: https://doi.org/10.1038/s41598-025-19408-6.

[5] H. Hassani, H. Khedmatgozar, A. Hussein, A. Monem, S. Rahma, R. Arapbaev, R. K. Ahmed, A. Ahmed, F. Abdul, and K. Khazal, “Harnessing quantum computing for real-time data analytics: A 2025 perspective,” Iranian Journal of Information Processing Management, vol. 40, 2025, [Online]. Available: https://doi.org/10.22034/jipm.2025.728117.

[6] E. Suryani, E. Asmari, and B. Harjito, “Image segmentation of acute myeloid leukemia using multi Otsu thresholding,” Journal of Physics: Conference Series, vol. 1803, Art. no. 012016, 2021, [Online]. Available: https://doi.org/10.1088/1742-6596/1803/1/012016.

[7] G. Koßmann, L. Binkowski, L. van Luijk, T. Ziegler, and R. Schwonnek, “Deep-Circuit QAOA,” Quantum, vol. 9, Art. no. 1882, 2025, [Online]. Available: https://doi.org/10.48550/arXiv.2210.12406.

[8] J. Larkin, M. Jonsson, D. Justice, and G. G. Guerreschi, “Evaluation of QAOA based on the approximation ratio of individual samples,” Quantum Science and Technology, vol. 7, 2022, [Online]. Available: https://doi.org/10.1088/2058-9565/ac6973.

[9] J. Silos-Sanchez, J. Rodriguez Flores, J. Martínez-Mireles, M. A. Márquez, A. Austria-Cornejo, and J. Ruiz-Vanoye, “Comparison between FRQI and NEQR quantum algorithms applied in digital image processing: Master’s Thesis Summary,” International Journal of Combinatorial Optimization Problems and Informatics, vol. 16, pp. 213-225, 2025, [Online]. Available: https://doi.org/10.61467/2007.1558.2025.v16i1.898.

[10] S. Yuan, X. Mao, J. Zhou, and W. Xiaofa, “Quantum image filtering in the spatial domain,” International Journal of Theoretical Physics, vol. 56, no. 8, pp. 2495-2511, 2017, [Online]. Available: https://doi.org/10.1007/s10773-017-3403-x.

[11] F. Rehm, S. Vallecorsa, K. Borras, D. Krücker, M. Grossi, and V. Valle, “Precise image generation on current noisy quantum computing devices,” Quantum Science and Technology, vol. 9, 2023, [Online]. Available: https://doi.org/10.1088/2058-9565/ad0389.

[12] J. Mu, X. Li, X. Zhang, and P. Wang, “Quantum implementation of the classical guided image filtering algorithm,” Scientific Reports, vol. 15, 2025, [Online]. Available: https://doi.org/10.1038/s41598-024-84211-8.