Cloud-based Remote Sensing Platforms in Remote Sensing Experiment Course

Jun Li [1], Jinjing Deng [2], Qiaozhi Wang [3], Xiangyang Zeng [4], Lin Hu [5], Yanzhong Liu [6*]

[1] College of Resource and Environmental Engineering, Wuhan University of Science and Technology, Wuhan, China – 0108026@wust.edu.cn
[2] Wuhan University of Science and Technology, Wuhan, China – 1164997685@qq.com
[3] Wuhan University of Science and Technology, Wuhan, China – wangqiaozhi@wust.edu.cn
[4] Wuhan University of Science and Technology, Wuhan, China – ZengXiangyang@wust.edu.cn
[5] Wuhan University of Science and Technology, Wuhan, China – LinHu@wust.edu.cn
[6] Wuhan University of Science and Technology, Wuhan, China – liuyanzhong@wust.edu.cn

Keywords: Remote Sensing Education, Cloud-based Remote Sensing Platform, Google Earth Engine, AI Earth.

Abstract:

Processing massive archives of satellite imagery has historically paralyzed traditional desktop-based remote sensing laboratories. The sheer volume of computationally heavy tasks-from bulk atmospheric correction to long-term radiometric calibration-frequently exceeds the hardware capacity of local campus networks and student laptops. To bypass these severe limitations, this study presents a dual-cloud pedagogical architecture that integrates Google Earth Engine (GEE) and Alibaba's AI Earth. This hybrid framework allows students to instantly access petabytes of analysis-ready data while maintaining low-latency processing for complex modelling via domestic servers. We operationalized this setup through four core practical modules: urbanization monitoring, urban heat island analysis, nighttime light assessment, and AI-driven road extraction. By entirely eliminating the overhead of raw data management and environment configuration, students can finally redirect their cognitive focus toward the actual physics and algorithmic logic of remote sensing—such as parameterizing radiative transfer equations and interpreting radiometric time-series. Furthermore, in light of AI Earth's recent policy shift in March 2026, which heavily restricts free access for educational usage, we critically evaluate the long-term sustainability of this curriculum. To maintain unhindered access to cloud-native geoprocessing, our future instructional designs will assess alternative localized platforms and open-source AI frameworks, ensuring the uninterrupted evolution of rigorous Earth observation education.

1. Introduction

With the rapid accumulation of satellite archives, Earth observation has transitioned into the Remote Sensing Big Data (RSBD) era. However, introducing these massive datasets into university experimental courses has exposed the fatal weakness of traditional local desktop software, such as ENVI or ERDAS IMAGINE. Physical remote sensing analysis usually requires computationally heavy preprocessing steps. For example, performing long-term pan-sharpening or atmospheric correction based on radiative transfer models consumes huge computing resources. Under the traditional teaching mode, students have to download dozens of gigabytes of raw images to their local hard drives, which is almost impossible to finish during limited class hours.

Taking our remote sensing experiment course in the autumn of 2023 as an example, we asked 48 students to simultaneously download a single 1GB Landsat 8 L1TP-level scene via the campus network. This directly caused severe network congestion and request timeouts. Nearly 45 minutes of a 90-minute class were completely wasted on basic data transmission. Besides, the computing power of students' personal laptops varies greatly. When running memory-intensive tasks like random forest classification or multi-temporal image mosaicking, system freezing, crashing, and Out of Memory (OOM) errors happened frequently. This kind of physical hardware limitation turned a class that should focus on land surface parameter retrieval and physical mechanism discussion into a highly inefficient "IT troubleshooting" session.

To break through these local storage and computing bottlenecks, geospatial analysis is irreversibly migrating to cloud-native computing architectures. Globally, Google Earth Engine (GEE) has become the cornerstone of planetary-scale Earth system science research, relying on its petabyte-scale Analysis-Ready Data (ARD) public archives and distributed computing APIs (Gorelick et al., 2017). In remote sensing education, the universality of GEE greatly lowers the technical threshold for long-time-series and large-scale environmental monitoring (Amani et al., 2020; Tamiminia et al., 2020; Dovgyi et al., 2025). However, in domestic teaching practices in China, a single international cloud platform often faces the dilemma of cross-border network latency. It also encounters ecological barriers when connecting with domestic high-resolution satellite series and the latest computer vision foundation models. To address this gap, localized cloud platforms like Alibaba's AI Earth and PIE-Engine have gained significant traction. These platforms not only provide extremely low-latency domestic network access but also natively embed pre-trained foundation models optimized for satellite imagery, allowing students to efficiently conduct complex intelligent feature extraction tasks on the cloud (He et al., 2024; Shen et al., 2025).

Although the teaching potential of cloud platforms is widely recognized, existing teaching reforms are often limited to the introduction of a single platform or the construction of generalized WebGIS system architectures (Wolf et al., 2016; Yang et al., 2016). Our research team previously tried to build a Local Area Network (LAN) virtual experiment environment based on Python and the Flask framework (Li et al., 2020). Although this system reduced software installation time in the early stage, maintaining a local server capable of synchronously updating massive satellite data faced prohibitive hardware iteration costs. The high-concurrency requests easily broke down the local computing pool.

Drawing on these painful lessons, an overall leap from the LAN architecture to public cloud-native platforms is imperative. This paper proposes a dual-cloud platform symbiotic teaching architecture that deeply integrates the complementary advantages of GEE and AI Earth. Instead of treating these two major platforms as simple auxiliary viewing tools, we directly translate their powerful distributed computing power into four "problem-oriented" physical and applied remote sensing experiment projects. By constructing a complete implementation workflow covering pre-class, in-class, and post-class stages, this scheme aims to completely strip away the cognitive consumption of underlying software and hardware configuration on students, guiding them to substantively return to the logical design of remote sensing algorithms and the in-depth analysis of spatial phenomena's physical mechanisms.

2. Remote Sensing Experiment Course

Furthermore, the expanding volume of multi-temporal and hyper-spatial resolution satellite datasets has exacerbated the latency in traditional local computing environments. Upgrading campus laboratory hardware to continuously match the peak memory and GPU requirements for planetary-scale data processing imposes an unsustainable financial burden on most educational institutions.

To completely resolve the physical data storage and computing bottlenecks of our previous local network system, we rebuilt the experimental course around a layered cloud architecture explicitly designed for modern teaching environments. In our early teaching attempts between 2022 and 2023, we tried to deploy virtual computing environments based on Docker containers on local servers. However, physical remote sensing experiments often involve dense matrix operations (such as land surface temperature retrieval or aerosol optical depth calculations). When dozens of students simultaneously called the local GPU cluster for large-area image processing, Out of Memory (OOM) errors and extremely long task queue times practically destroyed the continuity of the class. To overcome these severe computation and storage bottlenecks, geospatial processing has transitioned toward cloud-native computing platforms. We constructed a four-tier teaching framework that transfers the technical complexity of server maintenance to commercial public clouds, providing students with an out-of-the-box experience.

As illustrated in the top-down workflow of Figure 1, the proposed architecture follows a four-tier design. Tier 1 ensures cross-platform educational access, allowing students to completely bypass local IT setup. Moving downward, Tier 2 serves as the dual-platform core engine, combining the global data ecosystem of GEE with the native Python and AI foundation models provided by Alibaba Cloud. Tier 3 acts as the invisible backend infrastructure, managing multi-petabyte data streams alongside complex cloud processing modules (such as atmospheric pre-processing and advanced classification). Finally, the bottommost level (Tier 4) translates these massive infrastructural capabilities into four tangible teaching applications: urban heat island monitoring, multi-decadal urbanization tracking, nighttime light observation, and AI-based road extraction. This tier-based design successfully shifts the heavy technical burden of server maintenance to commercial public clouds, providing students with a seamless, out-of-the-box Earth observation environment. Initial classroom trials demonstrated that all students could complete the full experimental workflow without any local software installation.

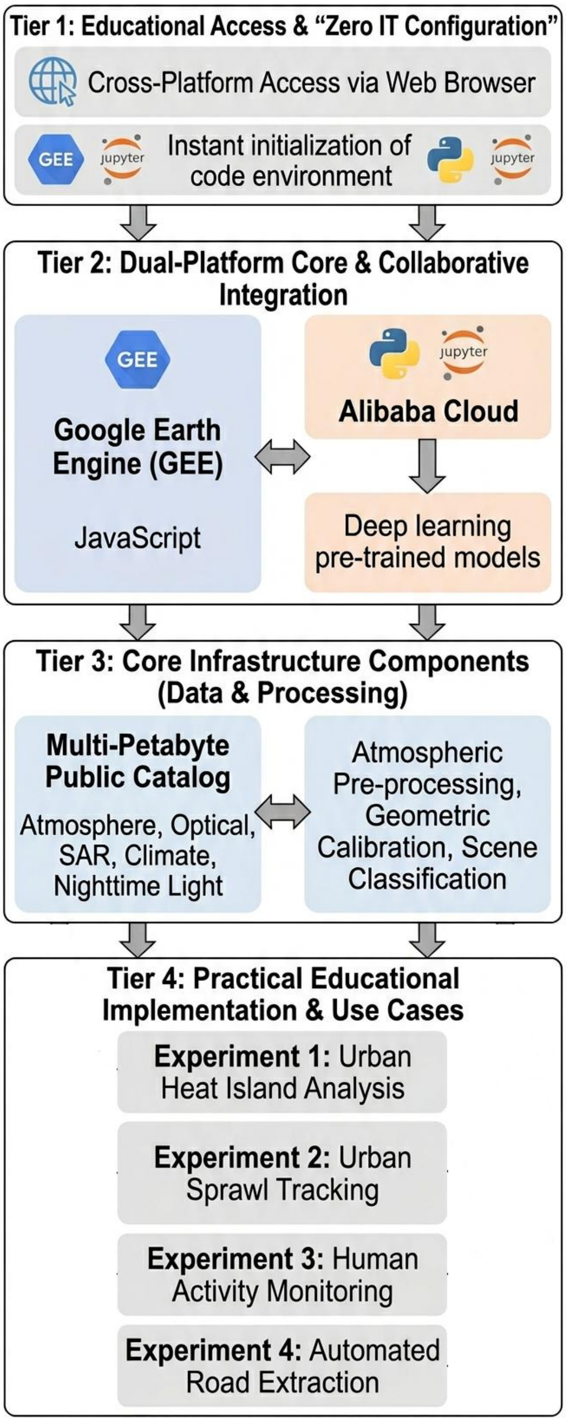

Figure 1. Architecture for the cloud-based remote sensing learning environment.

2.1 Teaching-Oriented Four-Tier Architecture

Tier 1: Educational Access and "Zero IT Overhead". In traditional teaching environments, requiring undergraduate students without a strong computer science background to configure Python environments with complex spatial dependencies on their personal Windows computers was usually a disaster. Dealing with notorious C++ compiler dependency conflicts when installing core spatial libraries like GDAL, Rasterio, or GeoPandas often wasted the first two weeks of the semester. Under this new tier, students no longer need to configure high-performance personal computers or install commercial software. By simply utilizing standard web browsers, they can access the experimental environment from anywhere, at any time, across various devices. This "zero IT overhead" approach fundamentally eliminates the technical friction before class begins.

Tier 2: The Dual-Platform Processing Engine. Tier 2 serves as the operational engine, leveraging the complementary strengths of GEE and AI Earth. GEE supports global-scale remote sensing analysis tasks through its historical data archives and mature APIs, which have become essential tools for planetary-scale geospatial analysis (Dong et al., 2016; Gorelick et al., 2017; Amani et al., 2020). Conversely, AI Earth utilizes a robust domestic cloud infrastructure (Alibaba Cloud) to support experiments requiring rapid network responses, particularly those involving AI algorithms and domestic high-resolution satellite constellations (Shen et al., 2025).

Tier 3: Core Infrastructure Components (Data and Processing). This tier operates invisibly on the backend, integrating cloud storage and computing capabilities to automatically connect data modules via APIs to petabyte-scale public data catalogs that update daily. Traditional remote sensing workflows required a tedious process of drawing bounding boxes on a webpage, downloading hundreds of gigabytes of raw compressed files, extracting them locally, and configuring complicated absolute file paths. Now, it provides instant, download-free access to massive multi-source datasets, including optical imagery, synthetic aperture radar, medium-resolution climate variables, and nighttime light data (He et al., 2024). The parallel cloud computing capabilities allow complex radiometric calibrations or multi-year pixel-level statistics to be executed simultaneously across thousands of server nodes (Genzano et al., 2020; Tamiminia et al., 2020).

Tier 4: Practical Educational Implementations. Our primary pedagogical goal is not to teach the intricacies of cloud infrastructure or distributed server management. Therefore, the bottom layer translates these abstract infrastructural powers into four specific teaching applications: urban heat island analysis, urbanization, nighttime light observation, and AI road extraction. By completely abstracting the underlying hardware constraints, this tier successfully shifts the students' cognitive focus away from tedious "IT troubleshooting"—such as managing local storage overflows, resolving software dependency conflicts, or configuring local environments. They can instead dedicate their full attention to core geospatial application design, focusing on algorithmic logic, parameter optimization, and the scientific interpretation of spatial phenomena.

2.2 Practical Collaborative Advantages of the Dual-Cloud Strategy

In a real-world classroom setting in China, relying on a single commercial cloud provider often exposes critical engineering vulnerabilities. The dual-cloud ecosystem we constructed offers complementary advantages across three vital dimensions

Network Latency and Concurrency Handling. Restricted by physical distance and cross-border gateways, occasional packet loss is an objective reality when accessing international platforms like GEE from mainland China. During high-concurrency classroom sessions—for example, when 50 students simultaneously drag the map on their screens to request real-time rendering of complex cloud masking or atmospherically corrected dynamic tiles—GEE occasionally encounters cross-border network latency, causing connection anomalies. Conversely, AI Earth is highly optimized for domestic network environments. Backed by Alibaba Cloud's distributed servers, it delivers a low-latency access experience. In our teaching strategy, this localized infrastructure eradicates network barriers in foundational teaching, allowing for uninterrupted interactive demonstrations. We reserve GEE for advanced analytical tasks where students require massive historical archives but have flexible scheduling to accommodate network fluctuations.

Bridging Programming Ecosystems. The programming interfaces of the two platforms present distinct pedagogical advantages. The GEE code editor primarily utilizes JavaScript for rapid web-based visualization rendering, whereas AI Earth provides Python support alongside a built-in Jupyter Notebook environment (Li et al., 2020). The industry demand in modern geoinformatics has irreversibly shifted towards machine learning and deep data mining. Introducing Python bridges the long-standing gap between traditional remote sensing workflows and modern computer science curricula. This empowers students to utilize mainstream open-source data science libraries (like Pandas, NumPy, and SciPy), enabling a smooth transition from spatial operations to machine learning modelling within a unified workspace.

Reconstructing the Accessibility of Deep Learning Pipelines. To effectively integrate artificial intelligence (AI) into the curriculum, we must clearly define the technical boundaries of both platforms. From the perspective of data volume and traditional machine learning frameworks, GEE remains unparalleled for global long-term historical archives. Its built-in ee.Classifier module is extremely efficient when handling shallow machine learning tasks like Random Forest or Support Vector Machines (SVM). Furthermore, GEE possesses robust deep learning support, allowing users to conduct complex neural network inferences by bridging to Google Cloud Platform's (GCP) Vertex AI nodes (He et al., 2024; Shen et al., 2025).

However, introducing this specific deep learning pipeline into a basic undergraduate classroom in China is highly impractical. Calling Vertex AI requires teachers or students to establish GCP billing accounts (usually requiring foreign credit cards), configure extremely complex IAM (Identity and Access Management) permissions, and set up Cloud Storage Buckets for lengthy data transferring. For an introductory undergraduate course that only lasts 90 minutes, this cumbersome "IT wall" and commercial billing threshold are insurmountable obstacles.

In contrast, AI Earth possesses built-in, out-of-the-box pre-trained large models for remote sensing. This native AI integration lowers the technical threshold, allowing students to engage with cutting-edge intelligent methods (like automated feature extraction) without building complex neural networks from scratch. Students no longer need to spend weeks drawing thousands of annotation samples, nor do they need to abandon training a complex U-Net architecture from scratch due to insufficient local GPU memory. With just a few lines of Python code, they can call pre-trained foundation models in the cloud to complete high-precision intelligent feature extraction. This zero-config cloud deep learning pipeline truly brings frontier AI remote sensing technology down to the daily teaching of undergraduate students. Building on this accessible deep learning capability, we developed four practical use cases. These cover urbanization, urban heat island, nighttime light, and AI road extraction.

3. Implementation and Practical Use Cases

To prevent the course from becoming a basic software tutorial, we translated the dual-cloud platform's processing power into problem-oriented projects driven by a closed-loop instructional design. Drawing on project-driven teaching experiences, we formulated an integrated "pre-class, in-class, and post-class" workflow. This workflow shifts the instructional focus from learning software interfaces to scientific inquiry and algorithm design.

3.1 Teaching Implementation Process

This workflow fundamentally transforms the traditional experimental organization model by shifting the instructional focus from software interfaces to scientific inquiry and algorithm design (Figure 2).

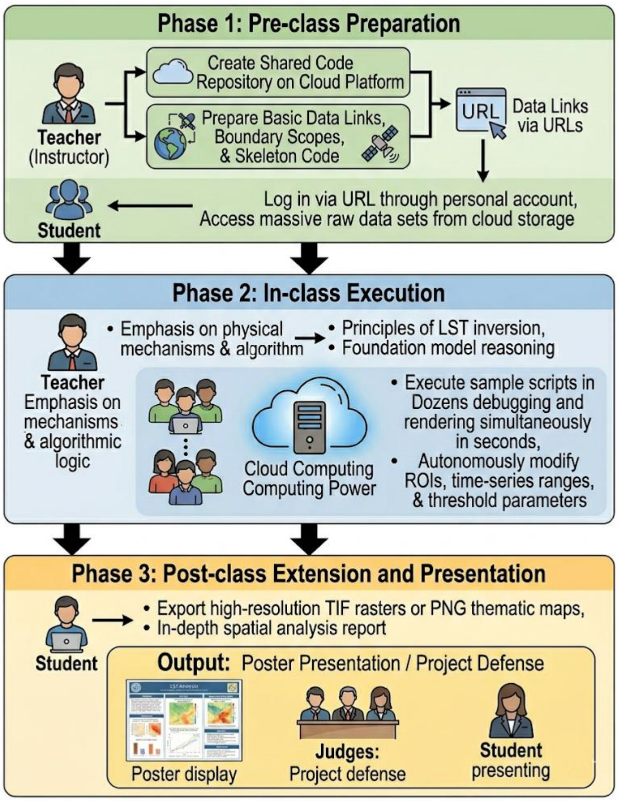

Figure 2. Workflow of the teaching implementation process

Pre-class: Instructors establish shared code repositories on the cloud platform and distribute basic data links, boundary extents, and skeleton codes to students via URLs. Students only need to log in via a web browser; they do not have to replicate massive raw datasets locally.

In-class: The classroom focus shifts significantly. Instructors emphasize the physical mechanisms and algorithmic logic of remote sensing. Students execute example scripts in the cloud and autonomously modify regions of interest (ROIs), time-series ranges, and threshold parameters. High-concurrency computing capabilities allow dozens of students to simultaneously render and debug within seconds, greatly enhancing classroom interactivity.

Post-class extension and presentation: The learning process extends beyond the classroom into a structured post-class analysis phase. Using the dual-cloud platforms, students export their experimental results, including high-resolution TIF rasters, custom PNG thematic maps, and quantitative charts. To accurately interpret the mechanisms behind their data, students are required to conduct independent literature reviews. They query academic databases for relevant research papers and experimental backgrounds to compare and substantiate their findings. Students then synthesize these literature insights with their cartographic outputs to compile comprehensive spatial analysis reports. Furthermore, to cultivate practical communication skills, we organize group-based poster presentations at the end of the semester. Students collaborate to design academic posters, where they visually display their processing workflows and thematic maps, and systematically explain their geographical interpretations to peers and instructors. This continuous process effectively consolidates their geospatial expression abilities and teamwork.

3.2 Specific Experimental Project Designs

To fully exploit the advantages of processing multi-source, large-scale data and intelligent algorithms, four core experimental projects were systematically designed. The detailed processing methodologies and parameter configurations for each experiment are comprehensively outlined below, with their respective cloud-based interfaces and outputs displayed in Figure 3.

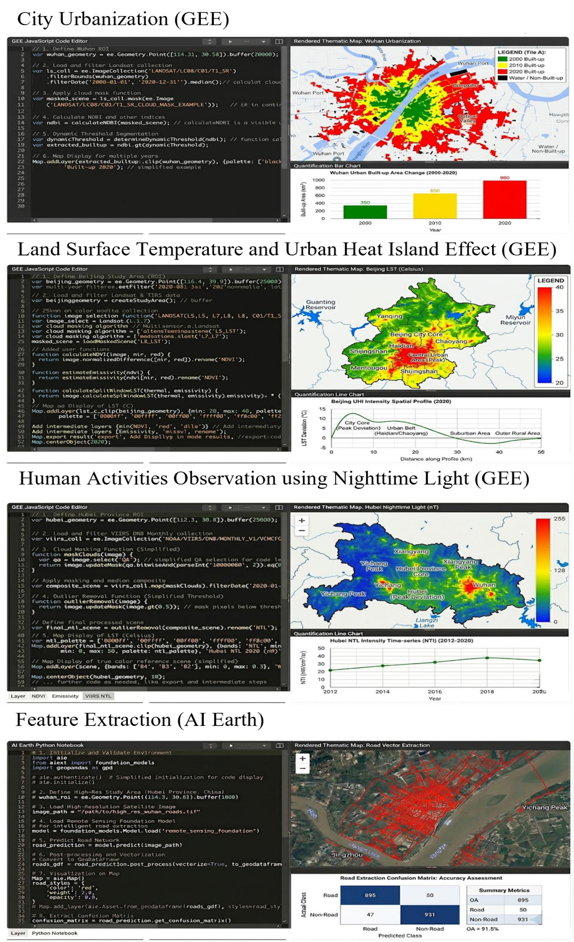

Figure 3. Cloud-based code interfaces and outputs for the four experimental projects

City Urbanization Monitoring (GEE Platform). In this project, the spatial expansion of the Wuhan metropolitan area was quantitatively analyzed using long-term optical satellite archives(Yan et al., 2023). The region of interest (ROI) was precisely delineated as the initial step. Following this, a multi-temporal ee.ImageCollection comprising Landsat 5, 7, and 8 surface reflectance data spanning a twenty-year period (2000-2020) was ingested via the GEE platform. To ensure rigorous data quality and eliminate atmospheric interference, a bitwise cloud and shadow masking algorithm was executed utilizing the QA_PIXEL band. Subsequently, the Normalized Difference Built-up Index (NDBI) was computed to effectively amplify the spectral signatures of impervious surfaces. A dynamic threshold segmentation technique was then systematically applied to the NDBI time series, functioning to separate urban built-up lands from background features. Finally, the extracted binary urban footprints were aggregated at distinct decadal intervals. This allowed for the dynamic illustration of rapid urbanization trajectories expanding toward peripheral zones, specifically highlighting the Dongxihu and Optics Valley districts

Land Surface Temperature and Urban Heat Island Effect (GEE Platform). The spatial heterogeneity of Beijing's Urban Heat Island (UHI) effect was rigorously investigated using thermal infrared remote sensing. Landsat 8 Collection 2 Level-2 datasets were selected as the primary data source. Within the GEE processing environment, the thermal band (Band 10) was initially converted from Digital Numbers (DN) to At-Satellite Brightness Temperature to reflect true thermal radiation. Concurrently, the Normalized Difference Vegetation Index (NDVI) was calculated to estimate the Fractional Vegetation Cover (FVC). This FVC was subsequently utilized as a critical parameter to derive pixel-specific land surface emissivity. By applying the simplified radiative transfer equation, the actual Land Surface Temperature (LST) was successfully retrieved. To quantitatively evaluate the UHI intensity, continuous spatial profile lines were generated, traversing directly from the highly urbanized core outward to vegetated suburbs. This approach allowed for the precise extraction and detailed statistical analysis of thermal deviation gradients across different land cover types.

Human Activities Observation using Nighttime Light Data (GEE Platform). The spatiotemporal evolution of socioeconomic human activities across the Hubei region was monitored using nighttime light (NTL) remote sensing techniques. Monthly average radiance composites from the NPP-VIIRS Day/Night Band (DNB) spanning from 2012 to 2020 were acquired for this purpose. To mitigate the adverse influence of ephemeral lights, stray light, and atmospheric noise, a strict and robust data cleaning protocol was implemented. Cloud-free observations were ensured via rigorous bitwise masking, which was then followed by a temporal median compositing algorithm to generate highly stable annual NTL datasets. Additionally, a dedicated background noise removal function was executed to successfully eliminate anomalous high-radiance pixels typically associated with industrial gas flares. Finally, the calibrated Nighttime Light Intensity (NTI) was segmented into distinct radiance thresholds. This segmentation facilitated a comprehensive, quantitative time-series analysis of human activity agglomeration.

Intelligent Feature Extraction using Foundation Models (AI Earth Platform). To demonstrate the powerful integration of artificial intelligence within modern geospatial analysis, automated urban road network extraction was conducted using the Python SDK natively supported within the AI Earth platform. High-spatial-resolution optical imagery was employed as the primary input dataset. Following proper environment initialization, a pre-trained remote sensing vision foundation model was loaded directly into the processing pipeline. Predictive inference was then executed directly on the input image paths to accurately delineate complex urban road networks. The resulting probability maps were subjected to necessary morphological post-processing and ultimately converted into vector GeoDataFrames for precise spatial alignment and mapping. To rigorously evaluate the algorithm's performance, a thorough accuracy assessment was conducted utilizing an independent validation dataset. A confusion matrix was constructed, allowing for the precise calculation of critical evaluation metrics.

4. Discussion: Technical and Educational Advantages

This section demonstrates the efficiency of the dual-cloud platform architecture over traditional teaching models through objective technical workflow comparisons and educational outcomes.

A comparison of technical workflows between the traditional desktop and dual-cloud experimental models highlights significant efficiency gains across four key dimensions. Regarding Data Acquisition, the traditional mode requires the manual retrieval and downloading of massive raw data archives, whereas the dual-cloud platform enables direct API calls to cloud data catalogs, achieving zero download time. In terms of Environment Setup, traditional models necessitate the installation of commercial software and local license configuration, whereas the cloud-based approach offers cross-platform access via standard web browsers with zero installation requirements. For Computing Power, traditional setups heavily rely on local PC CPUs, making them prone to software freezing or crashing during intensive tasks. In contrast, the dual-cloud architecture utilizes distributed server clusters, robustly supporting large-scale computing with second-level feedback. Finally, concerning Result Management, the traditional method involves the cumbersome copying of bulky project files or intermediate raster outputs, while the new platform allows for the seamless sharing of code scripts and visualized assets via cloud-based URLs.

4.1 Process- and Result-Based Validation

The traditional desktop-based experimental model presents severe technical bottlenecks. Students must spend extensive extracurricular time retrieving tens or hundreds of gigabytes of raw image archives and navigating tedious commercial software installations. During processing, varying PC hardware performance frequently causes software unresponsiveness or system crashes during complex tasks. For instance, during random forest classification on multi-season Sentinel-2 data, over 30% of student laptops crashed after exceeding local RAM capacities, forcing instructors to reduce the data volume.

In contrast, the dual-cloud architecture drastically improves the technical workflow. Students can invoke cloud-based data dictionary catalogs simply by opening a web browser, achieving zero download time and zero installation costs. By leveraging massive distributed server clusters, analyses that originally required hours now return feedback in seconds or minutes. This shift fundamentally eliminates the hardware and network barriers that previously restricted teaching progress, providing quantifiable improvements in technical efficiency.

The vibrant atmosphere of the class is illustrated in Figure 4. Students are depicted interacting with peers and instructors, utilizing high-quality thematic maps printed directly from the GEE and AI Earth platforms to explain their analyses of complex geospatial phenomena (e.g., urban expansion and thermal environments).

Figure 4. Students presenting their cloud-based remote sensing project results in 2025.

4.2 Significant Shift in Teaching Methodology

From the perspective of educational psychology, minimizing technical friction directly reduces extraneous cognitive load. Beyond technical metrics, the dual-cloud platform architecture fundamentally restructures the pedagogical approach in remote sensing experimental teaching methodology. In traditional classrooms, troubleshooting software errors, incorrect data paths, or insufficient computing power consumes too much time, leaving students with almost no opportunity to contemplate the scientific connotations of the algorithms themselves.

Our proposed architecture fundamentally alters this status quo. Because the platforms encapsulate the computational complexity, students can focus 100% of their energy on remote sensing application design, parameter sensitivity analysis, and the scientific interpretation of geospatial phenomena. As demonstrated by the four case studies, students can independently produce high-quality spatiotemporal thematic maps and quantitative statistical charts for their final academic poster presentations. This transition from passive software training to active Earth science exploration perfectly aligns with the research-oriented objectives of higher engineering education.

4.3 Limitations and Future Prospects

Constrained by current course evaluation systems, this paper primarily validates the proposed scheme based on system architecture design, technical workflow optimization, and specific case outputs. As emphasized in certain pedagogical studies, we have not yet conducted rigorous double-blind testing based on quantitative score comparisons across multiple consecutive student cohorts, nor have we administered large-scale student satisfaction surveys. In future teaching practices, introducing a quantitative evaluation mechanism based on platform operation logs, combined with a statistical analysis of final grades, will be an important direction for refining our evaluation system.

5. Conclusions and Future Work

To address the local computing bottlenecks and complex software configuration environments of traditional courses in the remote sensing big data era, this study innovatively proposes and successfully implements a dual-cloud experimental teaching architecture based on GEE and AI Earth. This architecture effectively integrates the global long-time-series data advantages of international platforms with the ultra-fast network access and native AI ecosystems of localized cloud platforms. By translating powerful distributed computing capabilities into four specific, problem-oriented experimental projects-urbanization monitoring, LST retrieval, nighttime light analysis, and intelligent feature extraction-we established a novel organizational workflow.

Comparative analysis demonstrates that this layered architecture eliminates the cumbersome costs associated with massive data downloads and local software installations inherent in the traditional model. Students can now conduct cross-platform, large-scale spatial analysis anywhere, anytime, using solely standard web browsers. This leap in technical accessibility successfully shifts students' cognitive load from tedious IT troubleshooting to core algorithm logic and scientific exploration, thereby significantly enhancing the continuity and professional depth of experimental teaching.

The ubiquity of cloud platforms has substantially lowered the entry barrier for Earth observation technology. We anticipate expanding this dual-cloud architecture beyond traditional remote sensing courses to empower interdisciplinary students in fields like resources and environment, economics, urban planning, and public health. Furthermore, we acknowledge that due to recent policy adjustments announced by AI Earth in March 2026, free educational use of its core functionalities will face significant restrictions. Consequently, to ensure long-term sustainability and universal accessibility, we will systematically evaluate alternative localized cloud platforms or open-source remote sensing AI frameworks as potential substitutes in future teaching practices.

Acknowledgements

The authors would like to acknowledge the financial support from the Teaching Research Project of Wuhan University of Science and Technology (Grant No. 2022X058) and WUST Science and Technology Innovation Project for College Students (Grant No. z25059). We also extend our gratitude to Google and Alibaba Cloud for providing access to the Google Earth Engine (GEE) and AI Earth computing platforms, which were instrumental in conducting the cloud-based experimental courses described in this study. Furthermore, we thank the United States Geological Survey (USGS), the National Aeronautics and Space Administration (NASA), and the European Space Agency (ESA) for providing the open-access Landsat, NPP-VIIRS, and Sentinel-2 satellite datasets.

References

Amani, M., Ghorbanian, A., Ahmadi, S.A., Kakooei, M., Moghimi, A., Mirmazloumi, S.M., Moghaddam, S.H.A., Mahdavi, S., Ghahremanloo, M., Parsian, S., Wu, Q., Brisco, B., 2020: Google Earth Engine cloud computing platform for remote sensing big data applications: A comprehensive review.

IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., 13, 5326-5350. doi.org/10.1109/JSTARS.2020.3021052

Dong, J., Xiao, X., Menarguez, M.A., Zhang, G., Qin, Y., Thau, D., Biradar, C., Moore, R., 2016: Mapping paddy rice planting area in northeastern Asia with Landsat 8 images, phenology-based algorithm and Google Earth Engine. *Remote Sens. Environ*., 185, 142-154. doi.org/10.1016/j.rse.2016.02.016

Dovgyi, S., Babiichuk, S., Tomchenko, O., Davybida, L., Biletska, M., 2025: Educational aspects of using Google Earth Engine: Training course for educators by the Junior Academy of Sciences of Ukraine. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-M-7-2025, 81-87 . doi.org/10.5194/isprs-archives-XLVIII-M-7-2025-81-2025

Genzano, N., Pergola, N., Marchese, F., 2020: A Google Earth Engine tool to investigate, map and monitor volcanic thermal anomalies at global scale by means of mid-high spatial resolution satellite data. *Remote Sens*., 12(19), 3232 . doi.org/10.3390/rs12193232

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau ,D., Moore, R., 2017: Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sens. Environ.*, 202, 18-27. doi.org/10.1016/j.rse.2017.06.031

He, S.Y., Zhou, B., Wang, B., Ding, Z.P., Zhang, J.F., 2024: Land classification research based on the improved Deeplabv3+ model and the AI Earth remote sensing cloud platform. *IEEE Access*, 12, 182516-182531. doi.org/10.1109/ACCESS.2024.3468384

Li, J., Sheng, J.L., Chen, Y., Ke, L.H., Yao, N., Miao, Z.H., Zeng, X.Y., Hu, L., Wang, Q.Z., 2020: A web-based learning environment of remote sensing experimental class with python. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLIII-B5-2020, 57-61. doi.org/10.5194/isprs-archives-XLIII-B5-2020-57-2020.

Shen, S.Q., Su, Q.Q., Lei, H., Yu, Z.F., Cheng, P.Y., Gu, W.X., Zhou, B., 2025: A web-based national-scale coastal tidal flat extraction system using multi-algorithm integration on AI Earth platform. *Remote Sens.*, 17(16), 2911. doi.org/10.3390/rs17162911

Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S., Brisco, B., 2020: Google Earth Engine for geo-big data applications: A meta-analysis and systematic review. *ISPRS J. Photogramm. Remote Sens.*, 164, 152-170. doi.org/10.1016/j.isprsjprs.2020.04.001

Wolf, N., Fuchsgruber, V., Riembauer, G., Siegmund, A., 2016: An adaptive web-based learning environment for the application of remote sensing in schools. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLI-B6, 53-56. doi.org/10.5194/isprsarchives-XLI-B6-53-2016

Yan, X.G., Li, J., Smith, A.R., Yang, D., Ma, T.Y., Su, Y.T., 2023: Rapid land cover classification using a 36-year time series of multi-source remote sensing data. *Land*, 12(12), 2149. doi.org/10.3390/land12122149

Yang, Z.L., Cao, J., Hu, K., Gui, Z.P., Wu, H.Y., You, L., 2016: Developing a cloud-based online geospatial information sharing and geoprocessing platform to facilitate collaborative education and research. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLI-B6, 3-7. doi.org/10.5194/isprs-archives-XLI-B6-3-2016

https://doi.org/10.5194/isprs-archives-XLIX-B5-2026-97-2026 |