ISSN: 2424-8460(O)
2251-2608(P)

Original Research Article

Remote sensing image classification based on deep learning

Peng Yin
Henan Polytechnic University, Jiaozuo, Henan, 454003, China

Abstract: Remote sensing image classification is a core component of remote sensing technology applications, widely serving fields such as resource exploration and environmental monitoring. Traditional classification design schemes have limitations such as insufficient feature extraction and low classification accuracy, making it difficult to adapt to complex scene requirements. Deep learning, with its powerful ability to mine deep features, provides a new design concept for remote sensing image classification. This article focuses on the design of a remote sensing image classification system based on deep learning, reviews the relevant technical foundations, completes the design of each core module and overall architecture, solves the pain points of traditional design, and provides efficient and feasible design solutions for remote sensing image classification.
Keywords: remote sensing images; deep learning; design

1. Introduction

With the rapid development of remote sensing technology and aerospace industry, the efficiency of remote sensing image acquisition has been greatly improved. Its application in multiple fields such as resource exploration, environmental monitoring, urban planning, and agricultural yield estimation has become increasingly widespread. Remote sensing image classification, as the core link for interpreting image semantics and mining image value, its design rationality and classification effect directly determine the application quality of remote sensing technology.

2. Overall design of remote sensing image classification system based on deep learning

2.1. Overall system architecture design

The overall system adopts a hierarchical modular design, which clearly divides the data input layer, preprocessing layer, feature extraction layer, and classification output layer into four core levels. Clear design boundaries and functional positioning are defined at each level, and data exchange and collaborative work between modules are achieved through standardized interfaces. The data input layer is responsible for receiving various remote sensing image data, the preprocessing layer completes image optimization, the feature extraction layer mines image features based on deep learning architecture, and the classification output layer achieves accurate classification and outputs results. At the same time, draw an architecture diagram to clearly present the connection relationship and data flow of each module, ensuring that the architecture design is scientific, reasonable, and implementable.

2.2. System core design process planning

The core design process of the system revolves around the logic of "input processing output", clarifying the sequence, connection conditions, and key nodes of each design stage. The process begins with remote sensing image input, optimized by the preprocessing module, and then passed to the feature extraction module to extract deep features. The classification module then completes the category determination, and finally outputs the classification result and saves it. Simultaneously plan the design priorities of each stage, clarify preprocessing and feature extraction as the core key stages, optimize the logical connection between stages, avoid process redundancy, ensure the efficiency and orderliness of the entire design process, and adapt to the overall design goals and actual application scenarios of the system.

3. Design of remote sensing image preprocessing module

3.1. Design of core preprocessing process

The core preprocessing stage revolves around image quality optimization and standardization, and four key stages are designed based on the characteristics of remote sensing images. Each stage works together to ensure processing effectiveness. One is image denoising design, selecting denoising algorithms that are suitable for remote sensing images, clarifying the core design parameters of the algorithm, and eliminating random noise and salt and pepper noise in the image in a targeted manner. While denoising, the semantic information of image details is preserved to avoid loss of details that may affect subsequent classification. The second is the design of geometric correction for images, establishing standardized correction processes, clarifying correction standards, correcting geometric distortions caused by shooting angles and sensor deviations, and ensuring the accuracy of image spatial positions. The third is image normalization design, determining a reasonable range of pixel value normalization, designing an adapted normalization method, unifying the pixel scale of the image, and eliminating interference caused by scale differences. The fourth is image cropping/stitching design, which combines the input size requirements of subsequent deep learning modules to design cropping and stitching schemes to ensure that the processed images meet the input standards of the module.

3.2. Preprocessing module interface and output design

The core of the interface and output design of the preprocessing module is to ensure smooth connection between the module and other modules in the system, while clarifying the standards for processing results to meet the needs of subsequent use. The interface design is divided into input interface and interactive interface. The input interface supports mainstream remote sensing image formats, specifies input parameter requirements, ensures that the original image can be smoothly imported into the module, and has format compatibility to adapt to remote sensing images from different sources; The interactive interface design is simple and supports manual parameter adjustment, making it easy to optimize the preprocessing effect according to actual needs and connect with the system's interactive interface. The output design specifies the standards for preprocessed images, including core parameters such as image format, resolution, and pixel value range, to ensure that the output image specifications are uniform and the quality meets the standards. Simultaneously design an output storage scheme, clarify the storage path and naming rules for output data, and facilitate subsequent calling and management; The output results should also include preprocessing status prompts, clearly indicating whether the processing was successful and explaining any abnormal situations, to facilitate timely detection and handling of problems by staff. The overall design should be in line with the system architecture to ensure seamless connection between the module and subsequent feature extraction modules.

4. Design of feature extraction module based on deep learning

4.1. Deep learning feature extraction architecture design

The core of designing a deep learning feature extraction architecture is to combine remote sensing image features, select an adapted basic network, and perform targeted optimization to construct an architecture system that balances feature extraction accuracy and efficiency. Firstly, the selection of basic networks should be based on the characteristics of remote sensing images, and priority should be given to convolutional neural network (CNN) architectures with strong spatial feature mining capabilities, which can adapt to the spatial distribution characteristics of surface cover in remote sensing images, while also considering network complexity and processing efficiency to avoid excessively complex architectures leading to high computational costs. Secondly, to address the issue of insufficient deep semantic feature extraction in remote sensing images, network structure improvement design is carried out, optimizing the parameters and connection methods of convolutional layers, pooling layers, and activation layers to increase the depth and breadth of feature extraction; Introduce a multi-scale feature fusion mechanism to integrate image features at different levels, improve feature integrity and recognition, and avoid missing key semantic information.

4.2. Function design and output specifications of feature extraction module

The functional design and output specifications of the feature extraction module focus on clarifying the positioning of the core functions of the module, unifying output standards, ensuring that the module functions meet standards and smoothly connect with subsequent classification modules. In terms of functional design, the module needs to implement three core functions: firstly, automatically receive preprocessed remote sensing image data, complete data validity verification, filter out unqualified images, and avoid affecting the feature extraction effect; The second is based on a deep learning architecture designed to automatically mine and extract deep features of images without the need for manual intervention, thereby improving the automation level of the module; The third is to achieve preliminary screening and optimization of features, remove redundant feature information, retain key effective features, reduce the computational load of subsequent classification modules, and have the function of manually adjusting feature extraction parameters to adapt to the application requirements of different scenarios. In terms of output specifications, specific standards for output features should be clarified: the dimension, data format, and numerical range of feature vectors should be unified to ensure that the classification module can read and process them normally; The output content includes extracted feature vectors, feature extraction status prompts (successful/failed), abnormal information explanations, etc., which are convenient for staff to troubleshoot problems; Output storage should follow standardized rules, clarify storage paths and naming formats, and facilitate subsequent feature calls, queries, and management. In addition, the output specifications need to conform to the input requirements of the classification module, ensuring that the specifications of the feature vectors match the input parameters of the classification module.

5. Design of remote sensing image classification module

5.1. Classifier design and adaptation

Firstly, the selection of classifiers should be based on the characteristics of remote sensing images and feature vectors, with priority given to classifiers adapted to deep learning architectures. The mainstream choices include fully connected layer classifiers and Softmax classifiers, balancing classification accuracy and computational efficiency, avoiding overly complex and computationally expensive classifiers, and adapting to system efficiency design principles.

Secondly, the classifier parameters are designed with specificity, and reasonable output dimensions are set based on the number of categories classified in remote sensing images; Choose an appropriate activation function to enhance the non-linear fitting ability of the classifier and improve the classification performance of complex features; Design a reasonable loss function to measure the deviation between classification results and real categories, providing direction for classification optimization.

5.2. Optimization design of classification results

Firstly, in response to potential redundant classification results during the classification process, a redundancy removal scheme is designed to screen for effective classification results, eliminate duplicate and invalid classification information, simplify the output results, and improve the readability and practicality of the results.

Secondly, in response to the misjudgment problem caused by fuzzy classification boundaries, a classification boundary optimization scheme is designed to refine classification rules, adjust classifier parameter thresholds, clarify the classification boundaries of different categories, reduce classification errors in fuzzy areas, and improve the discrimination of similar feature categories.

In addition, a mechanism for handling abnormal classification results is designed to label and perform secondary verification on results with low classification confidence and fuzzy judgments. Combined with the spatial correlation and texture features of remote sensing images, it assists in correcting classification results and reducing the probability of misjudgment.

6. System interaction interface design

6.1. Core interface module design

The core goal is to build a concise, intuitive, and smooth interactive platform that supports users in completing core functions such as remote sensing image input, preprocessing parameter settings, classification operation startup, and classification result viewing, reducing user operation barriers and improving operational efficiency; Simultaneously achieving visual display of classification processes, processing progress, and result data, allowing users to clearly grasp the system's operating status, facilitating operational control and result verification.

The interface design follows four core principles: first, the principle of usability, simplifying the operation process, reducing redundant steps, and making key functions clear at a glance; Secondly, the principle of aesthetics is to maintain a neat interface layout and coordinated colors to enhance the user experience; The third principle is consistency, which unifies the style and interaction logic of buttons, menus, and prompt information to avoid confusion in user operations; The fourth principle is practicality, focusing on core functional requirements and rejecting useless designs.

6.2. Interface interaction logic design

The core of interface interaction logic design is to follow the principles of "simplicity, efficiency, and coherence", sort out the user operation process, clarify the interaction relationship and operation rules of each interface module, ensure smooth and seamless user operation, and reduce the error rate of operation, in line with the overall design logic of the system. Firstly, standardize the user operation process and strictly follow the logical design interface jump relationship of "image input → parameter settings → start running → view progress → view results". Each link is seamlessly connected and supports functions such as returning to the previous step, pausing operations, and canceling operations to enhance operational flexibility; Secondly, design a clear interactive feedback mechanism that provides visual prompts in a timely manner (such as button color changes, prompt pop ups) after users click buttons, upload files, and adjust parameters. Clear feedback is given on whether the operation is successful or failed to avoid users repeating the operation; Once again, design exception interaction handling logic to provide concise and easy to understand prompt messages for scenarios such as file format errors, unreasonable parameter settings, and system operation exceptions. At the same time, provide solution guidance to help users quickly solve problems; Finally, optimize the detailed interaction design, support shortcut key operations, drag and drop viewing of result images, unify interface jump animations and operation response speed, ensure smooth interaction, while maintaining consistency in the interaction logic of each module, allowing users to quickly familiarize themselves with the operation mode, achieve efficient interaction between users and the system, and fully utilize the interactive support role of the interface.

7. System testing plan design

7.1. Test scope and test type design

The design of testing scope and types should be in line with the overall system design, scientifically defining boundaries and types, ensuring comprehensive coverage and highlighting key points, without involving specific testing execution and data verification. The testing scope focuses on the core modules of the system, including preprocessing, feature extraction, classification, interactive interface modules, as well as design content such as inter module interfaces, overall architecture connections, and extended interface adaptability. The testing types are divided into three categories according to design verification requirements: functional testing, which verifies the completeness and correctness of the design functions of each module; Usability testing to verify the rationality of interactive interface design and smooth operation; Compatibility and stability testing is conducted to verify the adaptability of the system to different image formats and operating environments, as well as the stability of interface interaction and system operation, ensuring that the testing is highly targeted and in line with the overall design objectives.

7.2. Design ideas for test cases

The design of test cases follows the principle of "fitting the design, covering comprehensively, and highlighting key points", focusing on the testing objectives and scope, clarifying the design logic, and not involving specific case writing and result judgment. The core idea is to be guided by system design requirements and ensure that use cases accurately verify the key design points of each module; Balancing comprehensiveness and emphasis, covering the core functions, interface connections, and abnormal scenarios of each module, with a focus on key design stages such as feature extraction and classification; Balancing normal and abnormal scenarios, verifying the normal functionality and system fault tolerance of the verification module, and improving the verification approach for exception handling mechanisms.

8. Conclusion

This article focuses on the design of a remote sensing image classification system based on deep learning, and completes the overall architecture of the system and the design of each core module, covering preprocessing, feature extraction, classification, interactive interface, and testing scheme. The design has solved the pain points of traditional classification design, with smooth connection and strong adaptability of each module, balancing practicality and scalability, achieving the full process design implementation of classification, and providing feasible design solutions.

About the author

Peng Yin (1997.10.02), male, Han nationality, Kaifeng City, Henan Province, postgraduate student, Research direction: Computer remote sensing.

References

[1] Zhi Yanli, Zhou Yu, Liu Qing, et al. Research on Hyperspectral Remote Sensing Image Classification Algorithm Based on Spectrum Space Multi Feature Deep Learning [J]. Remote Sensing Technology and Applications, 2025, 40 (05): 1232-1242.

[2] Qin Xian, Hao Rong, Xing Peng, et al. Classification Model of Land and Resources Remote Sensing Images Based on Deep Learning [J]. Geospatial Information, 2024, 22 (10): 34-37.

[3] Zheng Dan, Song Sinan, He Qiang Optimization of Deep Separable Convolutional Networks in Remote Sensing Image Classification in Deep Learning [J]. Information Systems Engineering, 2024, (10): 132-135.

[4] Li Jiaojiao, Zhang Min Research on Global Feature Extraction for Remote Sensing Image Classification Based on Deep Learning Extended Convolutional Networks [J]. Information Systems Engineering, 2024, (08): 140-143.

[5] Wang Xiaoyan, Li Jinglan, Bai Yanping, et al. Research on Remote Sensing Image Classification Method Based on Deep Learning [J]. China Soil and Water Conservation, 2023, (12): 7-10.