GeoInformatica (2025) 29:907–925
https://doi.org/10.1007/s10707-025-00552-0

RESEARCH

Secure spatial remote sensing image matching

Yi Wei[1] · Xuan Zhou[2] · Hao Huang[1] · Hao Wang[1] · Chuang Hu[1] · Jiawei Jiang[1,2]

Received: 9 April 2025 / Revised: 4 May 2025 / Accepted: 20 May 2025 /
Published online: 31 May 2025

Abstract
Spatial data generation, such as remote sensing image data, plays a vital role in applications like remote sensing imagery. Large language models (LLMs) are increasingly used to process and analyse such data. However, many existing data sources are redundant, and efficiently matching spatial remote sensing data in federated scenarios is crucial. In this work, we focus on the critical task of privacy-preserving image matching, which is widely used in applications such as object recognition, scene understanding, and image retrieval. We propose a spatial remote sensing image matching framework called SRSIM to realise this. SRSIM leverages a hierarchical matching mechanism, including image-level matching and pixel-level matching, to improve matching performance from both global and local perspectives. Furthermore, we designed a product quantization and cluster-based strategy to accelerate the matching process. The experimental results demonstrate that SRSIM significantly outperforms prior works, such as SEPIM and SESIM, in terms of both matching performance and efficiency.

Keywords Privacy preserving · Image matching · Federated learning · Large language model

✉ Jiawei Jiang
jiawei.jiang@whu.edu.cn

Yi Wei
weiyi96@whu.edu.cn

Xuan Zhou
xuanzhou33416@gmail.com

Hao Huang
haohuang@whu.edu.cn

Hao Wang
wanghao.cs@whu.edu.cn

Chuang Hu
handc@whu.edu.cn

[1] School of Computer Science, Wuhan University, Bayi Road, Wuhan 430064, Hubei, China

[2] Wuhu Overseas Students Pioneer Park, Wuhu Pioneer Park, Yinhu North Road, Wuhu 241000, Anhui, China

1 Introduction

LLM for spatial data generation In recent years, large language models (LLMs) have demonstrated significant potential in spatial data generation by leveraging their ability to learn patterns from large-scale datasets [1, 2]. LLMs have been applied to a range of tasks, including spatial text-to-data transformation, trajectory representation learning, and spatial reasoning. For example, the integration of LLMs with spatiotemporal information enables the generation of synthetic spatial trajectories and geospatial predictions, facilitating more efficient decision-making processes [3–5]. There has also been considerable interest in applications such as road trajectory generation and trajectory similarity joins, which create synthetic yet realistic trajectories to address challenges like high data collection costs and privacy concerns [6, 7]. LLMs offer a more flexible and scalable solution for generative spatial modeling than traditional approaches that rely on computational simulations or rule-based models. Additionally, LLMs assist in generating datasets for various purposes, such as filling data gaps, augmenting training data for physical experiments, and developing diverse samples to optimize model performance. Despite these promising advancements, this emerging field still faces challenges related to ensuring spatial accuracy, improving interpretability, and aligning generated data with real-world constraints.

When LLM Meets Image Matching With the development of generative models like LLMs, spatial data generation has also advanced rapidly, enabling the automated creation of diverse geospatial datasets such as satellite imagery and maps. These synthetic datasets play a crucial role in augmenting training data, supporting deep learning tasks, and simulating complex scenarios. However, generating high-quality spatial data poses challenges such as ensuring geographical consistency and minimizing redundant or unrealistic content. Effectively addressing these challenges-through robust redundancy detection and quality control-is essential for enhancing the value and applicability of generated spatial data in geospatial analysis. In this case, finding and reducing the redundancy between different data owners is vital. By using robust image-matching techniques, redundancy in spatial datasets can be reduced and key features aligned, thereby improving the effectiveness of LLMs in geospatial tasks. Over the past few decades, there have been various image-matching methods that can replace humans. In the field of image matching, one core idea is extracting features and then performing matching on the extracted feature descriptors. The feature extraction methods can be roughly divided into two categories: 1) manually designed feature extractors and 2) learning-based feature extractors. The manually designed feature extractors include SIFT, SURF, ORB, LATCH, and DeepDesc [8]. Unlike these traditional methods, a series of learning-based methods apply deep learning techniques to extract features, significantly improving the feature extraction performance. Specifically, learning-based techniques can be divided into pixel-level and image-level categories. The pixel-level methods generate representation descriptors for pixel-to-pixel matches in the raw image and image-level methods learn a representation from an entire image.

When Image Matching Meets Federated Learning When there is only one data owner, the data processing is straightforward and performed on a single machine. However, in the LLM era, multiple organizations, companies, and individuals have different available LLM-generated data [9]. In such scenarios, each data owner may hold a subset of the entire dataset. With the increasing demands for data protection, many countries have published data protection regulations, e.g., the European General Data Protection Regulation (GDPR), the China

Personal Information Protection Law, and the United States Privacy Act [10–12]. Federated learning (FL) emerges as a technique that collaboratively trains distributed models across data silos in a privacy-preserving manner. The popular federated learning algorithms, such as FedAvg, aggregate models trained by different data owners, yielding a better model than that trained by each data owner [13]. Some privacy protection techniques, including secure multi-party computation (SMC), homomorphic encryption (HE), and differential privacy (DP), are leveraged to protect data communication between data owners.

The Current Landscape Several existing works have studied privacy-preserving image matching. SEPIM [14] enables the user to send the query to the data owner and measures image similarity without revealing the actual images. SESIM [15] presents an improved protocol over SEPIM for similar image retrieval based on SMC. Each client chooses a pre-trained CNN model to extract image features. For query images, they calculate the distance or similarity metrics over the encrypted features. If we look at previous works, they often assume running a secure query over a data owner and are hard to work for queries between multiple data owners. Regarding the feature extractors, pixel-level methods (e.g., SEPIM) only extract local information and cannot provide the global information of an image. In contrast, image-level approaches, e.g., HomographyNet, concentrate on global patterns and hence may miss some similar images that are only similar within local regions. Furthermore, in the process of similarity search, previous work has predominantly relied on brute-force search, resulting in significant computational overhead. Currently, some approaches are proposed to speed up the search for similar embeddings. Specifically, vector similarity search techniques, including hash-based, graph-based, product quantization, and inverted file methods, are widely used in LLM workloads.

Challenges The prior works have not yet studied how to effectively and efficiently perform image matching over multiple data owners in a privacy-preserving way. Specifically, there remain several unsolved challenges. Since different types of feature extractors have shown their merits for different tasks in the literature, the first question is *how to extract proper features using pixel-level and image-level methods?* When multiple data owners jointly run image matching, they need to exchange image-related data with each other and incur data leakage risk. If data protection techniques are introduced, the image-matching task inevitably becomes costly. Therefore, it is necessary to design methods to reduce the computational complexity. To this end, our second question is *how to efficiently run image matching in a privacy-preserving way?* In this work, we study the above two challenges and try to design a holistic solution with system and algorithm co-design.

Summary of Contributions We propose a framework, called SRSIM, for image matching under privacy-preserving conditions. We design a hierarchical matching strategy that considers both global and local knowledge of the candidate images and hence improves the image-matching performance. To protect data privacy, we introduce HE and DP for data communication. To optimize system efficiency, SRSIM leverages product quantization to reduce the embedding size and distance computation. Additionally, by applying a clustering-based strategy, the number of matching image pairs can be significantly decreased. The major contributions of this work are summarized below:

- We propose a general framework termed SRSIM for distributed privacy-preserving image matching. SRSIM introduces a hierarchical matching mechanism, which includes pixel-level matching and image-level matching, to improve the matching performance.

- We design a series of system optimizations in SRSIM. A product quantization method is introduced to reduce the dimension of extracted features. Furthermore, a clustering-based method is designed to avoid pair-to-pair comparison.
- SRSIM protects data privacy with delicate design throughout the entire lifecycle. SRSIM protects the image identity by randomly shuffling all the images, and the transferred data by applying HE and DP. We also conduct a comprehensive security analysis for SRSIM.
- We evaluate the effectiveness and efficiency of SRSIM with an extensive empirical study. The experimental results on several datasets show that, compared to the baselines, SRSIM can achieve a better trade-off between matching accuracy and matching speed.

2 Related works

Image matching Image matching is to identify the same or similar content or structure from two or more images. Typically, image matching involves two crucial techniques — the feature extraction and the matching strategy [8]. Deep learning has significantly improved image matching in the phase of feature extraction. Brown et al. [16] proposed a learning framework that uses Powell minimization and linear discriminant analysis. Radenović et al. [17] proposed to fine-tune CNNs by generalizing max and average pooling and extracting features from the images, which can achieve better matching performance. Trzcinski et al. [18] utilized the boosting technique to train a series of weak learners and learn non-linear local visual feature representations. However, these methods may perform poorly on downstream tasks due to varying application scenarios. Several CNN-based detectors integrate feature extraction and image detection into a comprehensive matching process. After extracting feature descriptors, the matching process usually compares these descriptors based on distance-based metrics. The FT strategy estimates matching images via a threshold, leading to one-to-many matching cases. The NN strategy can retrieve more positive matches but incurs one-to-many scenarios. The MutualNN method seeks accurate results but may sacrifice missing other similar images. The NNDR method considers the distance between the first and second NN, yielding robust matching performance without sacrificing positive matching images [19]. However, NNDR depends on the stable distribution of descriptors such as SIFT [20].

Federated Learning Federated learning is a privacy-preserving machine learning paradigm. One of the earliest works from Google designed distributed model training across Android clients by exchanging model updates [21]. According to different data and task scenarios, federated learning can be categorized into different types: horizontal, vertical, federated transfer learning [22], etc. FedAvg is a well-known federated learning algorithm that aggregates model updates by computing their average. FedProx is more effective than FedAvg in handling non-i.i.d. (identically and independently distributed) issues [23]. Similarly, SCAFFOLD can address non-i.i.d. issues by introducing a control variate to decrease client drift [24].

Similarity search for vectors Neural networks extract embeddings as intermediate products to represent the features of the input data. The distance between them indicates the similarity between the corresponding data items. Neighbourhood search provides a direct and efficient mechanism for similarity search on item embeddings. For example, in hashing-based methods, Locality Sensitive Hashing (LSH) is an algorithm designed to reduce the dimensionality by hashing similar input items into the same hash buckets with high probability, facilitating efficient approximate nearest neighbour search. Graph structure-based methods, such as

Hierarchical Navigable Small World (HNSW), are proposed for approximate nearest neighbour search, which leverages a multi-layer graph structure [25]. Inverted File Index (IVF) efficiently group features or data points to their corresponding clusters according to their similarities, facilitating fast similarity searches. Quantization methods like Product Quantization (PQ) decompose the original large space into a Cartesian product of low-dimensional subspaces [26]. In the presence of diverse vector similarity search approaches, several easy-to-use vector similarity search libraries are developed to instantiate these methods, including but not limited to Faiss, Milvus, Proxima, and Vearch.

Privacy Protection Techniques Several HE libraries are designed to protect data privacy under the cryptography mechanism [27]. In practice, additive HE is widely used in machine learning algorithms [28]. Secure Multi-Party Computation (SMC) is a cryptographic technique that allows multiple parties to jointly compute a function over their inputs while keeping those inputs private. Another line of work uses DP [29] that adds noise to the data or k-anonymity that obscures attributes until the third party cannot distinguish the individual.

3 A general framework for federated image matching

3.1 Overview of SRSIM

System Roles As shown in Fig. 1, SRSIM consists of several clients that hold images and a server that runs the matching process.

- **Client.** Each client in our system has a set of images. The client extracts both pixel-level and image-level features from the images. We use a pre-trained model for pixel-level features that can adapt to various domains [30]. All the clients jointly train a CNN network as the image-level extractor for pixel-level features to make the network adapt to the downstream image domain. Dimensionality reduction techniques are leveraged to transform the embeddings into lower dimensions, which are then transferred to the server. Note that the client applies privacy protection approaches to ensure data safety.
- **Server.** The server receives data from the clients and performs the image-matching stage. The server is assumed to be honest but curious. It complies with the processing protocol but may try to guess the original data on the clients. The server provides both image-level matching and pixel-level matching and combines different matching strategies in a hierarchical framework.

Data Communication The clients send feature descriptors to the server during the feature extraction phase. Afterwards, the server computes the matching results and returns them to the clients. The matching phase may involve multiple communication rounds for both image-level and pixel-level matching strategies.

Data Protection To safeguard data communication, SRSIM provides two mainstream privacy protection techniques — HE and DP. HE offers a strong privacy guarantee at the expense of expensive computation. DP is computationally efficient, but the noise introduced inevitably causes less accurate results.

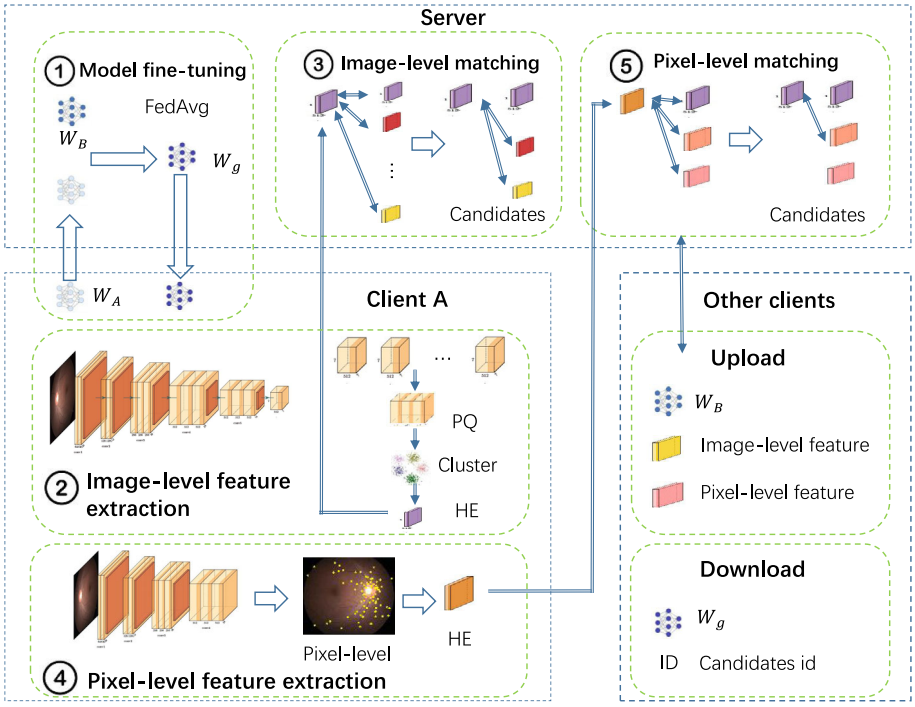

Fig. 1 The overall framework of the proposed SRSIM. ① SRSIM fine-tunes the image-level model between the clients with FedAvg. ② The clients extract image-level features. After applying PQ and k-means clustering, the encrypted descriptors are sent to the server. ③ The server conducts image-level matching and returns the candidate images. ④ The clients extract pixel-level feature descriptors from the candidate images. The encrypted descriptors are sent to the server. ⑤ The server conducts pixel-level matching and returns the matching results

3.2 Processing procedure

Below, we describe the processing procedure of SRSIM. Different from the prior works, SRSIM adopts a hierarchical matching scheme — the images are firstly matched using image-level descriptors, and the similar candidates are further compared using pixel-level features. In this way, both local and global information are extracted.

1. **Model fine-tuning.** SRSIM relies on a CNN to extract image-level descriptors. The chosen CNN model should be robust to image transformation so that images that are similar in the global context will not be overlooked. To this end, we choose HomographyNet [31] as our image-level feature extractor. A further question is — *should we use pre-trained model parameters?* When deployed in different domains, the pre-trained model may encounter performance degradation due to data heterogeneity and distribution drift. Therefore, we choose to fine-tune the pre-trained model and the local model replicas are aggregated on the server via FedAvg.
2. **Image-level feature extraction.** Once HomographyNet is fine-tuned till convergence, SRSIM generates image-level descriptors. Specifically, SRSIM extracts the output of the hidden layer (size of $7 \times 7 \times 512$) before the fully connected layer, generating 7×7 feature vectors of size 512. To decrease computation and communication costs, we apply

dimensionality reduction to these descriptors. We will elaborate on this optimization in Section 3.4.

3. **Image-level matching.** Each client generates image-level descriptors for its local images applies either HE or DP and randomly shuffles the original image identifiers. Afterwards, it sends pseudo-identifiers and encrypted descriptors to the server. The server conducts the matching process by comparing the received descriptors. To avoid brute-force matching and reduce the number of candidate image pairs, we design a clustering-based scheme, which will be elaborated in Section 3.4.
4. **Pixel-level feature extraction.** After the server sends the pseudo identities of candidates to the clients, each client extracts their pixel-level descriptors using VGG16. We do not choose HomographyNet here because it is specially designed to handle image deformation. Instead, we use the output of the conv4-3 layer in VGG16. To choose high-quality descriptors, we use D2Net [30] to generate a detection score d_{ij} for pixel (i, j) of descriptor f_{ij}. Then, a pixel is treated as a key point if its detection score is a local maximum compared to its neighbouring pixels. We rank the key points by their detection scores and choose the first 512 feature descriptors as the output. Note that, we do not apply dimensionality reduction on pixel-level descriptors since pixel-level matching requires fine-grained feature extraction.
5. **Pixel-level matching.** The server receives pixel-level descriptors from the clients and performs pixel-level matching. Assuming each image has a set of k descriptors $F = \{f_1, f_2, ..., f_k\}$, for each descriptor in the first set, our scheme needs to find the closest descriptor in the second set (and vice-versa for cross-checking). To reduce the number of comparing pairs, we filter descriptors that are unlikely to be similar. Specifically, each client ranks them by their L2-norm in descending order. The client then sends the feature descriptors and L2 norms to the server. The server runs the NNDR strategy by leveraging the ratio of distances between the best match and the second-best match for each feature. As advised in [19], we set this threshold to 0.8. After NNDR, the server counts the matched pixels. If the matching number is over a certain threshold, they are considered as a matching image pair. This threshold is decided according to the inlier rate of datasets [8].

3.3 Implementation of privacy protection

SRSIM implements HE to calculate the similarity of image descriptors without exposing the actual data. This is achieved by encrypting the original data and performing mathematical operations on the ciphertext. In SRSIM, we encrypt integers using the BFV protocol and real numbers using the CKKS protocol. In addition to HE, SRSIM also offers DP to introduce noises (e.g., Laplace noise and Gaussian noise), preventing others from speculating the original data. In SRSIM, we choose Laplace noise and perturb the descriptors.

3.4 System optimizations

Challenge Despite the hierarchical matching paradigm, SRSIM may still suffer from expensive overhead for a large volume of images. Assuming two parties have N and M images, and the image- and pixel-level descriptors have n and m dimensions, the matching complexity is $O(NMn^2) + O(Nm^2)$. The former refers to the image-level matching and the latter refers to the pixel-level matching.

PQ-based Dimension Reduction To solve the above challenge, an intuitive solution is to reduce the dimensionality of the image descriptors. We employ product quantization (PQ) to construct a codebook for representing high-dimensional descriptors. The key idea is to decompose the original vector space into the Cartesian product of M low-dimensional sub-spaces and quantize each sub-space into k code words. The codebook is, therefore, defined as $\mathcal{C} = \mathcal{C}^1 \times \cdots \times \mathcal{C}^M$, where $\mathcal{C}^M$ represents each sub-space's Cartesian product. To map the feature descriptors $\mathbf{X} \in \mathcal{R}^D$ into the space of the codebook, as shown in Fig. 2, the processing of PQ includes 3 major steps — vector partition, vector quantization, and codebook generation.

1. **Vector partition.** To meet the assumption of PQ, every subvector should have an equal number of dimensions. Hereby we partition the vector $\mathbf{X}$ into $\left[\mathbf{x}^1, \ldots \mathbf{x}^M\right]$. As suggested in prior works [26], we choose $M = 8$.
2. **Vector quantization.** Once all the vectors are partitioned in Step 1, PQ runs KMeans to cluster the subvectors into k parts and generate a sub-codeword for each cluster. We choose $k = 256$ and use the cluster IDs to represent subvectors [26]. Each sub-codeword has 256 digits.
3. **Codebook generation.** By concatenating all the sub-codewords $\mathbf{c}^m$, we generate a codebook $\mathbf{c} = \left[\mathbf{c}^1, \ldots \mathbf{c}^m, \ldots \mathbf{c}^M\right]$ with each $\mathbf{c}^m \in \mathcal{C}^m$.

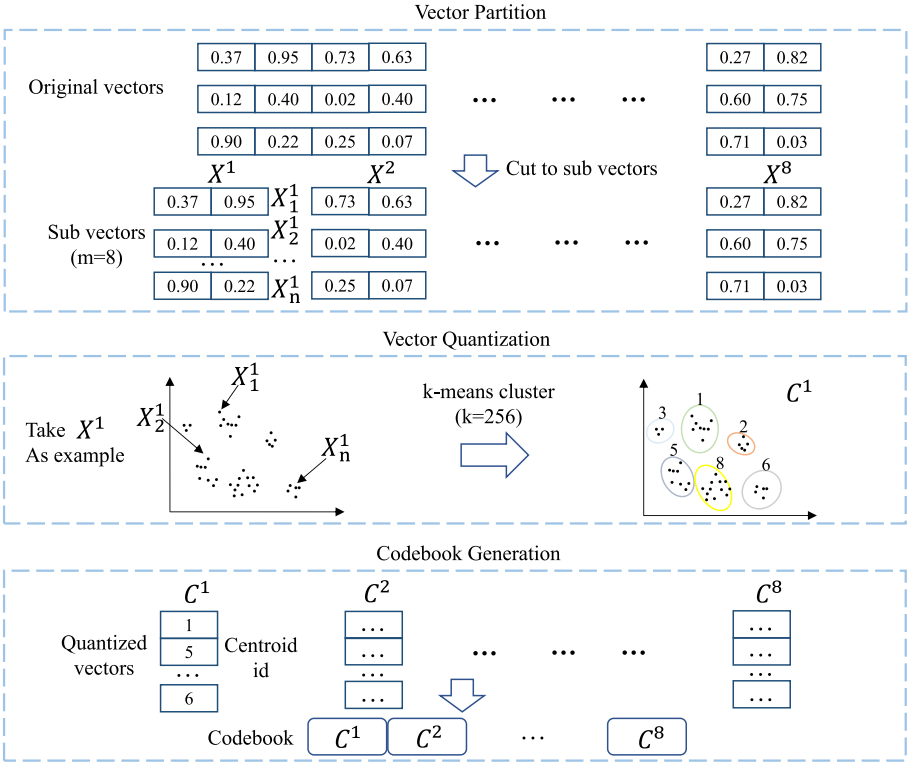

Fig. 2 PQ-based Dimension Reduction. The indices of centroids, so-called centroids id, vary from 0 to 255 within the sub-space

Inverted Index-based Matching After dimensionality reduction, the next problem is *how to reduce the number of matching?* The number of candidate image pairs can be extraordinarily high when dealing with a large dataset. To improve this, we introduce a clustering approach to build an inverted index for the matching phase [32].

1. **Cluster initialization.** For the two image sets from clients A and B, each client first runs clustering over the extracted image-level feature descriptors. This step partitions the local images into k sets and each set will generate a corresponding cluster centroid.
2. **Cluster set selection.** Each client sends the cluster centroids and image descriptors to the server. For a query image from client A (and vice versa), the server computes its similarity to the cluster centroids of client B. It only continues comparing a cluster if the similarity is larger than a threshold. If HE is applied, the server needs to send the HE-encrypted similarity values to a client for decryption. We use K-means as our clustering algorithm ($k = 8$).
3. **In-cluster matching.** After comparing all the images in a cluster, the server returns the matching image pairs to the clients (refer to Section 3.2 for details).

3.5 Discussion

Scalability Although we assume two parties, SRSIM can easily support the scenario of multiple clients. For multiple clients, each client sends its feature descriptors to the server, and the server finds similar images from the received set. For example, if the task is to find all similar images between any two clients, the server compares all possible pairs of descriptor sets.

Security Analysis We next conduct a security analysis of SRSIM, including identity security, image security, feature security, and result security.

- **Threat model.** We assume three cases of threat models. **(Honest-but-curious)** The server and clients are honest but curious which means no collusion in this case. **(Server-client collusion)** The server may collude with one client and intend to acquire data from other clients. **(Client-client collusion)** Clients may conspire together to acquire data from the rest clients. **(Malicious server)** The server is no longer "honest-but-curious," but instead behaves maliciously; it can modify message content, distribute malicious model parameters, manipulate the aggregation process, or even deliberately implant backdoors. **(Byzantine faults)** Some clients or the server may behave in an arbitrary or Byzantine manner, sending completely incorrect or fabricated updates.
- **Identity Security.** In SRSIM, the image identifiers are randomly shuffled on each client before they are sent to the server. Hence, the server only receives pseudo identifiers. **(Honest-but-curious)** The server cannot map pseudo-identifiers back to the original image identifiers, ensuring identity security. **(Server-client collusion)** Even if the server colludes with one client, it can only access the identifiers of that particular client. **(Client-client collusion)** Colluding clients know their pseudo-identifiers and could acquire the identity from the rest of the clients if matching results are positive. **(Malicious server)** It may actively attempt various methods to undermine identity protection mechanisms in order to locate or associate specific identities. **(Byzantine faults)** Byzantine nodes may manipulate the protocol process, such as prematurely disclosing identities or sending specific messages to mislead others, thereby access the identifiers of that particular client.

- Image Security. In SRSIM, the client's images are not directly transmitted. Instead, feature extraction is encrypted. **(Honest-but-curious)** The server only receives encrypted feature descriptors, preventing access to the original images. **(Server-client collusion)** The server only receives encrypted feature descriptors, preventing access to the original images. **(Client-client collusion)** Colluding clients can only share their feature descriptors, not the original images of others. **(Malicious server)** The server only receives encrypted feature descriptors, like the Honest-but-curious case, preventing access to the original images. **(Byzantine faults)** Byzantine nodes can not access the image from other clients, thereby preventing their access to the original images.
- Result Security. After the pixel-level matching phase, the server sends the pseudo identifier of matching images and the number of matched pixels to the clients. **(Honest-but-curious)** The server cannot infer the content of the images, preserving result security. **(Server-client collusion)** If the server colludes with a client, it may learn about the matching results of that client but not others. **(Client-client collusion)** Colluding clients can only share their matching results, not those of other clients. **(Malicious server)** A malicious server can modify the results, which may lead to incorrect or problematic matching outcomes. **(Byzantine faults)** Byzantine nodes can send incorrect descriptors, which lead to incorrect matching result.

Conclusion Compared with formal privacy-preserving techniques such as differential privacy, homomorphic encryption, and federated learning, SRSIM adopts a lightweight mechanism that focuses on practical data protection without incurring substantial computational or communication overhead. SRSIM balances privacy and efficiency by leveraging encrypted features and identifier shuffling; however, it does not provide quantitative measures but only formal analysis. In practice, this means SRSIM may be more suitable for scenarios where resource constraints are a priority and formal privacy guarantees are stringent. Meanwhile, our method can defend against most mainstream attack types, but its effectiveness is limited against certain malicious attacks. Nevertheless, users must be cautious about potential privacy risks in highly adversarial environments. Integrating or extending SRSIM with these formal approaches remains an important direction for future work.

4 Experiment

4.1 Experimental settings

Environment We use three machines in our experiments — two act as the clients and one as the server. Each machine has a Gen Intel(R) Core(TM) i5-12400F CPU, two Nvidia GeForce RTX 3090TI GPUs, with 128 GB memory.

Datasets As shown in Table 1, we use four datasets to evaluate SRSIM. The Retinal dataset consists of 70 retinal image pairs with non-grid transformation. The average matching pixels and inlier rate are 158.4 and 41.56 [33]. The remote sensing dataset has 161 image pairs, including color-infrared, SAR, and panchromatic photos with an average of 767.7 and an inlier rate of 68.50 [33]. The DTU dataset involves several scenes for objects and has 130 image pairs with an average number of 729.3 and an inlier rate of 58.83 [33] The Gaofen dataset contains 4,000 images of 10 views taken from a certain location [34]. We partition the dataset into two partitions, each with images of 5 views. The inlier rate cannot be calculated since

Table 1 The statistics of evaluated datasets

Datasets	Retinal	Remote Sensing	DTU	Gaofen
# image pairs	70	161	130	2000
# matching pixels	158.4	767.6	729.3	586.2
Inlier rate	41.56	68.50	58.83	–

"# image pairs" refers to the number of matching image pairs in ground truth. "# matching pixels" refers to each image's average number of matching pixels. "Inlier rate" refers to the ratio of matching pixels in each image, according to the ground truth

Gaofen does not provide matching pixels. By aligning two partitions, we thereby generate 2,000 image pairs.

Baselines SRSIM has two versions — SRSIM with HE and SRSIM with DP. We first evaluate SRSIM using different feature descriptors and then compare it with three state-of-the-art privacy-preserving image matching approaches — SYBA [35], SEPIM [14] and SESIM [15].

Protocols We implement Adam as the optimization algorithm with an initial learning rate of 10^{-3}, which is decayed by 0.5 every 10 epoch and set batch size to 256. The HomographyNet and VGG16 models are fine-tuned for 50 epochs. For these part of hyper-parameters, our fine-tuning is mainly based on the values reported in previous papers [31, 36]. In the PQ part, the hyper-parameters are introduced in Section 3.4, and the other hyper-parameters are set as default. Specifically, the number of subvectors is 8 and the number of sub-codewords is 256. In the clustering-based optimization, the size of the cluster is 8. We gradually adjusting parameters based on experience and experimental results for these hyper-parameters are in small parameter space and experienced practitioners. The other hyper-parameters are set as default.

4.2 Exploring feature descriptors

In SRSIM, feature descriptors are critical for the model performance and system efficiency. To this end, we compare different descriptors, including both image-level and pixel-level, in SRSIM that establishes the hierarchical matching strategy. Specifically, to compare the image-level descriptors, we run the image-level matching and generate matching results, without running the pixel-level matching stage.

Image-level matching We compare image-level descriptors, including LF-Net, RF-Net, HomographyNet, LIFT and TCDET. Tables 2 and 3 show the accuracy and efficiency of the evaluated image-level descriptors. As can be observed, five evaluated descriptors show different performances. HomographyNet achieves the highest accuracy on three datasets out of four. For example, on the DTU dataset, the matching accuracy of HomographyNet is 97.64%, 1.7% higher than the second-best method TCDET. Regarding time cost, LIFT and TCDET are relatively faster, followed by HomographyNet. Overall, HomographyNet obtains the best trade-off between matching performance and efficiency, verifying our choice of HomographyNet as the image-level feature extractor in SRSIM. In Fig. 3, we showcase the result of HomographyNet on four query images. In the first row of Fig. 3, the retina images

Table 2 The time cost of different image-level descriptors. We highlight the best results in bold

Methods	Descriptors	Retina	DTU	RemoteSensing	Gaofen
Image-level	LF-Net	84.1	93.4	158.6	289.4
	RF-Net	76.4	64.7	147.1	203.0
	Homo-Net	69.6	74.1	120.8	186.6
	LIFT	**65.7**	64.6	**98.0**	186.4
	TCDET	67.4	**58.1**	103.5	**184.7**

We report the average matching time per image pair in seconds

on the right side have deformation. On the second and fourth rows, the images are rotated. On the third row, the images are displaced. As can be observed, HomographyNet can handle diverse image scenarios, such as deformation, rotation, and displacement.

Pixel-level matching In this section, we choose HomographyNet as the image-level feature descriptor and compare pixel-level descriptors (Sift, Surf, Superpoint, ORB, and TILDE). Tables 4 and 5 show the accuracy and efficiency of the evaluated pixel-level descriptors. As observed, SRSIM achieves the highest accuracy on all four datasets. For example, on the Retina dataset, the matching accuracy of SRSIM is 97.62%, which is 0.74% higher than the second-best method Superpoint. As for the time cost, the classical Sift and Surf are relatively faster. In general, SRSIM achieves the best trade-off between matching performance and efficiency. Figure 4 shows examples of matching results. These two image pairs are treated similarly in the image-level matching stage; however, the left pair has many more matched pixel-level descriptors. Therefore, SRSIM can effectively reduce the mismatch cases at the image level, with a moderate increase in matching time.

4.3 Exploring similarity search methods

In this section, we conduct experiments to evaluate the components in GPPIM with a popular similarity search library Faiss, which provides techniques introduced in Section 2 and hybrid of them [25]. Here we compared SRSIM with similarity search baselines and hybirds of above methods. The hyperparameters of various methods have already been fine-tuned to the best of our knowledge. Specifically, The FlatL2 here represents brute force matching based on the L2-norm. The number of cells/clusters to partition data into is equal to 128 for the inverted file index; 16 for the hybrid method of inverted file and PQ, and 8 for the hybrid

Table 3 The accuracy of different image-level descriptors. We highlight the best results in bold

Methods	Descriptors	Retina	DTU	RemoteSensing	Gaofen
Image-level	LF-Net	94.78%	93.84%	**98.94**%	98.76%
	RF-Net	93.46%	94.52%	97.51%	98.73%
	Homo-Net	**96.53**%	**97.64**%	98.64%	**99.12**%
	LIFT	95.52%	94.16%	97.42%	98.42%
	TCDET	95.71%	95.94%	98.12%	98.46%

We report the accuracy of predicted matching images

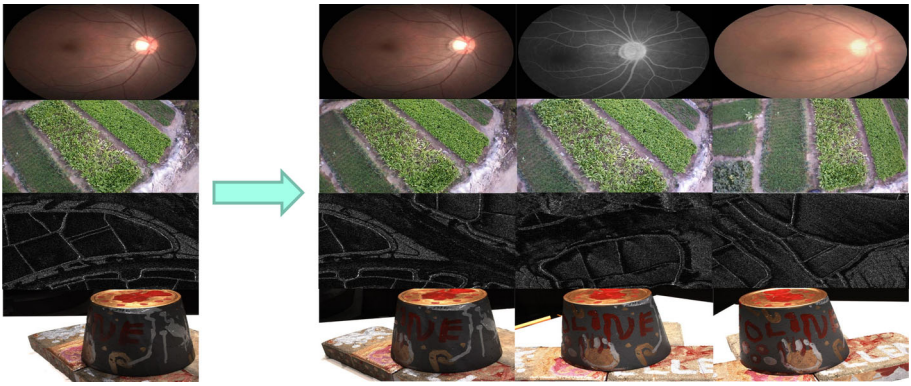

Fig. 3 Examples of image-level matching results using HomographyNet as the descriptor. The images on the right side show deformation. On the second and the fourth row, the images are rotated. On the third row, the images are displaced

method of HNSW and PQ. The number of bits required to encode a single cluster-ID is equal to 32 for the Locality Sensitive Hashing method and 8 for the rest of the methods. For the HNSW method, we fine-tuned the number of connections each vertex will have equal to 128 and 16 for the hybrid of HNSW and PQ. We choose 256 as cluster centroids for the PQ method and 128 for the rest. As shown in Table 6, SRSIM achieves the best efficiency result. For example, when applying the FlatL2 index, the accuracy increased from 98.4% to 99.2% compared to our method, while the matching time increased from 8.34 seconds to 395.27 seconds. When applying the IVF, the matching accuracy is 0.3% higher than SRSIM, and the matching speed is 10.1× slower. The matching accuracy is slightly higher than our methods however the matching time is still very high. For high-dimensional data, the performance of LSH may be suboptimal, especially when the vector dimensionality is large in this case. As the dimension increases, the stored vectors become larger, leading to excessively long search times. When applying PQ, the matching accuracy is 0.2% higher than our method, and the matching speed is 6.2× slower. When applying HNSW, the matching accuracy is 5.9% lower than our method, and the matching speed is 2.2× slower. When applying the hybrid method of IVF or HNSW with PQ, the matching accuracy will decrease compared to their original methods but the matching efficiency will also increase as a trade-off. In conclusion, our methods can achieve a good trade-off between accuracy and efficiency.

Table 4 The accuracy of different pixel-level descriptors. We highlight the best results in bold

Methods	Descriptors	Retina	DTU	RemoteSensing	Gaofen
Pixel-level	Sift	96.25%	97.72%	99.11%	98.97%
	Surf	96.17%	96.85%	98.24%	98.86%
	Superpoint	96.88%	97.64%	98.74%	99.12%
	ORB	96.36%	96.73%	98.26%	99.08%
	TILDE	96.22%	96.71%	98.74%	99.12%
	SRSIM	**97.62%**	**98.51%**	**99.22%**	**99.43%**

We report the accuracy of predicted matching images

Table 5 The time cost of different pixel-level descriptors. We highlight the best results in bold

Methods	Descriptors	Retina	DTU	RemoteSensing	Gaofen
Pixel-level	Sift	**113.2**	**121.2**	188.8	371.3
	Surf	111.4	121.9	192.9	375.9
	Superpoint	159.5	169.6	227.2	422.3
	ORB	121.0	127.3	143.6	382.1
	TILDE	134.9	132.2	**124.3**	**369.3**
	SRSIM	125.7	130.6	164.3	395.6

We report the average matching time per image pair in seconds

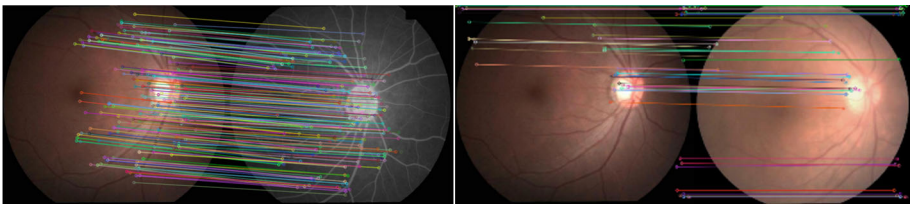

Fig. 4 Examples of pixel-level matching results using SRSIM. The patches on the left side are the correct matches according to the ground truth. The SRSIM shows that the left side compares two images with 428 matched pixels and the right side compares another with only 51 matched pixels

Table 6 Ablation study of SRSIM with vector similarity search methods from Faiss. We highlight the best results in bold

Method	Matching Accuracy	Matching Time (s)
Ours	98.4%	**8.34 ± 0.33**
FlatL2	**99.2**%	395.27 ± 0.65
IVF	98.7%	84.75 ± 2.15
LSH	87.6%	274.46 ± 2.83
PQ	98.6%	52.19 ± 1.71
IVF & PQ	97.8%	11.48 ± 1.64
HNSW	92.5%	18.73 ± 0.62
HNSW & PQ	86.5%	12.47 ± 0.97

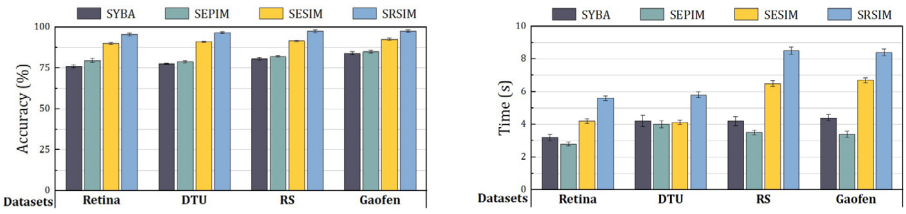

Fig. 5 (a) The accuracy of the state-of-the-art methods on different datasets. (b) The efficiency of the state-of-the-art methods on different datasets. We report the matching time per query image in seconds

4.4 Performance comparison with the state-of-the-art methods

Next, we compare the end-to-end performance of SRSIM and the state-of-the-art baselines regarding accuracy and efficiency. As illustrated in Fig. 5(a), this evaluation encompasses a detailed analysis of the matching accuracy across four distinct datasets. Remarkably, our proposed SRSIM consistently achieves the highest performance across all datasets. For example, when applied to RemoteSensing dataset, SRSIM demonstrates a remarkable matching accuracy of 99.22%. It outperforms the matching accuracy of the second-best method, SESIM, by a significant margin of 6.23%. This substantial improvement can be attributed to SRSIM's adoption of a hierarchical framework, thereby enhancing the overall matching quality and reducing mismatches. Additionally, we compare the runtime of the baselines and show the results in Fig. 5(b). Unsurprisingly, SRSIM is slower than the other baselines since SRSIM compares both image-level and pixel-level features, leading to considerable extra time overhead. The extra time overhead is moderate considering the benefit of enhanced matching accuracy. Overall, SRSIM outperforms three baselines considering both accuracy and efficiency.

4.5 Ablation study

In this section, we conduct an ablation study and evaluate the components in SRSIM, including HE, DP, PQ, and IVF optimization. As shown in Table 7, when using the original descriptors without applying privacy protection techniques, the matching accuracy is the highest and the matching time is also relatively low. If HE is applied, the server does not know the original descriptors and the matching accuracy remains the same at the expense of longer matching time. In particular, DP does not significantly enlarge the matching time but causes lower matching accuracy (up to 0.136) due to the introduced noises. This implies that the choice of DP raises a trade-off between efficiency and security — adding larger noises can better protect data but hurt model quality. The matching time will be significantly reduced when adopting PQ and IVF. For example, when using PQ and IVF strategies for HE encrypted descriptors, the matching time is reduced from 441.93 seconds to 8.34 seconds — a 55× improvement. Meanwhile, introducing IVF and PQ slightly decreases the accuracy from 99.2% to 98.8% and 98.4%, respectively.

Table 7 The ablation study of SRSIM regarding different privacy-preserving and optimization methods over all four datasets and the average matching time

Method	Matching Accuracy	Matching Time (s)
Original	$99.2 \pm 0.02\%$	29.12 ± 0.21
HE	$99.2 \pm 0.02\%$	441.93 ± 1.72
DP	$85.6 \pm 0.63\%$	32.75 ± 0.75
HE & IVF	$98.8 \pm 0.07\%$	79.46 ± 0.94
DP & IVF	$81.5 \pm 0.54\%$	8.25 ± 0.80
HE & PQ	$98.9 \pm 0.25\%$	53.36 ± 0.31
DP & PQ	$83.1 \pm 0.86\%$	7.28 ± 0.27
HE & PQ & IVF	$98.4 \pm 0.06\%$	8.34 ± 0.33
DP & PQ & IVF	$78.2 \pm 0.72\%$	4.17 ± 0.26

5 Conclusion

In this work, we propose SRSIM, a general privacy-preserving image-matching framework. SRSIM designs a hierarchical matching mechanism, including pixel-level and image-level matching, and leverages privacy protection techniques to protect data security. Furthermore, SRSIM proposes a series of system optimizations to improve the matching efficiency. We conduct extensive experiments over various datasets. The experimental results show that SRSIM significantly outperforms the prior works.

Beyond conventional image-matching tasks, our framework SRSIM holds significant potential for application in spatial data generation. In this context, efficiently and securely identifying redundant or similar images is crucial for curating high-quality, diverse geospatial datasets that underpin downstream analysis and model training. This makes SRSIM especially valuable for large-scale geospatial data synthesis workflows, where data integrity, privacy, and reduced redundancy are essential for reliable and practical spatial analysis.

Acknowledgements Not applicable.

Author Contributions Yi Wei and Jiawei Jiang conceived and designed the study. Yi Wei and Xuan Zhou conducted the experiments and collected the data. Hao Huang analyzed the data and prepared the statistical results. Hao Wang and Chuang Hu wrote the main manuscript text. All authors reviewed the manuscript.

Funding This work was sponsored by Key R&D Program of Hubei Province (2023BAB077), National Natural Science Foundation of China (62472327), and the Fundamental Research Funds for the Central Universities (2042025kf0040). This work was supported by Sichuan Clinical Research Center for Imaging Medicine (YXYX2402).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Wang C, Chen L, Shang S, Jensen CS, Kalnis P (2024) Multi-scale detection of anomalous spatio-temporal trajectories in evolving trajectory datasets. In: Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. KDD '24, Association for Computing Machinery, New York, NY, USA, pp 2980–2990
2. Zhou S, Shang S, Chen L, Han P, Jensen CS (2024) Grid and road expressions are complementary for trajectory representation learning
3. Zhou S, Shang S, Chen L, Jensen CS, Kalnis P (2024) RED: Effective trajectory representation learning with comprehensive information
4. Feng S, Meng F, Chen L, Shang S, Ong YS (2024) Rotan: A rotation-based temporal attention network for time-specific next poi recommendation. In: Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. KDD '24, Association for Computing Machinery, New York, NY, USA, pp 759–770
5. Rao X, Jiang R, Shang S, Chen L, Han P, Yao B, Kalnis P (2025) Next point-of-interest recommendation with adaptive graph contrastive learning. IEEE Trans Knowl Data Eng 37(3):1366–1379
6. Rao X, Shang S, Jiang R, Han P, Chen L (2025) Seed: Bridging sequence and diffusion models for road trajectory generation. In: The web conference 2025
7. Ding Z, Li K, Chen L, Shang S (2025) Parallel online similarity join over trajectory streams. In: The web conference 2025
8. Ma J, Jiang X, Fan A, Jiang J, Yan J (2021) Image matching from handcrafted to deep features: A survey. Int J Comput Vision 129(1):23–79

9. Liu Q, Chen C, Qin J, Dou Q, Heng PA (2021) Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1013–1023
10. Hoofnagle CJ, Sloot B, Borgesius FZ (2019) The European union general data protection regulation: What it is and what it means. Inf Commun Technol Law 28(1):65–98
11. Feng Y (2019) The future of china's personal data protection law: Challenges and prospects. Asia Pac Law Rev 27(1):62–82
12. Rindfleisch TC (1997) Privacy, information technology, and health care. Commun ACM 40(8):92–100
13. Li X, Huang K, Yang W, Wang S, Zhang Z (2020) On the convergence of FedAvg on non-iid data
14. Abduljabbar ZA, Jin H, Ibrahim A, Hussien ZA, Hussain MA, Abbdal SH, Zou D (2016) Sepim: Secure and efficient private image matching. Appl Sci 6(8):213
15. Janani T, Brindha M (2022) Secure similar image matching (sesim): An improved privacy preserving image retrieval protocol over encrypted cloud database. IEEE Trans Multimed 24:3794–3806
16. Brown M, Hua G, Winder S (2011) Discriminative learning of local image descriptors. IEEE Trans Pattern Anal Mach Intell 33(1):43–57
17. Radenović F, Tolias G, Chum O (2018) Fine-tuning cnn image retrieval with no human annotation. IEEE Trans Pattern Anal Mach Intell 41(7):1655–1668
18. Trzcinski T, Christoudias M, Lepetit V, Fua P (2012) Learning image descriptors with the boosting-trick, vol 25
19. Yan H, Lv G, Ren X, Dong X (2018) Improved nearest neighbor distance ratio for matching local image descriptors. In: Zhou Z-H, Yang Q, Gao Y, Zheng Y (eds) Artif Intell. Springer, Singapore, pp 185–197
20. Austin PC (2014) A comparison of 12 algorithms for matching on the propensity score. Stat Med 33(6):1057–1069
21. Li L, Fan Y, Tse M, Lin KY (2020) A review of applications in federated learning. Comput Ind Eng 149
22. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: Concept and applications. ACM Trans Intell Syste Technol (TIST) 10(2):1–19
23. On convergence of fedprox (2022) Yuan, X., Li, P. Local dissimilarity invariant bounds, non-smoothness and beyond 35:10752–10765
24. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT (2020) Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning, PMLR, pp 5132–5143
25. Douze M, Guzhva A, Deng C, Johnson J, Szilvasy G, Mazaré PE, Lomeli M, Hosseini L, Jégou H (2024) The Faiss library
26. Jégou H, Douze M, Schmid C (2011) Product quantization for nearest neighbor search. IEEE Trans Pattern Anal Mach Intell 33(1):117–128
27. Tan H, Xuan S, Chung I (2020) Hcda: Efficient pairing-free homographic key management for dynamic cross-domain authentication in vanets. Symmetry 12(6)
28. Benaissa A, Retiat B, Cebere B, Belfedhal AE (2021) TenSEAL: A library for encrypted tensor operations using homomorphic encryption
29. Ouadrhiri AE, Abdelhadi A (2022) Differential privacy for deep and federated learning: A survey. IEEE Access 10:22359–22380
30. Dusmanu M, Rocco I, Pajdla T, Pollefeys M, Sivic J, Torii A, Sattler T (2019) D2-Net: A trainable CNN for joint detection and description of local features
31. DeTone D, Malisiewicz T, Rabinovich A (2016) Deep image homography estimation. arXiv preprint arXiv:1606.03798
32. Algergawy A, Massmann S, Rahm E (2011) A clustering-based approach for large-scale ontology matching. In: Advances in databases and information systems: 15th international conference, ADBIS 2011, Vienna, Austria, September 20-23, 2011. Proceedings 15, Springer, pp 415–428
33. Ma J, Zhao J, Jiang J, Zhou H, Guo X (2019) Locality preserving matching. Int J Comput Vision 127:512–531
34. Sun X, Lv Y, Wang Z, Fu K (2022) Scan: Scattering characteristics analysis network for few-shot aircraft classification in high-resolution sar images. IEEE Trans Geosci Remote Sens 60:1–17
35. Wang Y, Miao M, Shen J, Wang J (2019) Towards efficient privacy-preserving encrypted image search in cloud computing. Soft Comput 23:2101–2112
36. Zitova B, Flusser J (2003) Image registration methods: a survey. Image Vis Comput 21(11):977–1000

Yi Wei is currently working toward the PhD degree with Wuhan University. His research interests include federated learning, machine learning systems and data valuation.

Xuan Zhou is a research manager with Wuhu Overseas Students Pioneer Park, Wuhu, Anhui, China.

Hao Huang received his PhD degree in computer science from Zhejiang University, China, in 2012. He is currently a professor with the School of Computer Science, Wuhan University, China. His research interests include big data management and analytics, data mining, and intelligent information systems.

Hao Wang received his Ph.D. degree in computer science from the University of Hong Kong in 2014. He is currently a Full Professor with the School of Computer Science, Wuhan University. His research interests include spatiotemporal big data analytics and machine learning.

Chuang Hu received his B.S and M.S. degrees in Computer Science from Wuhan University in 2013 and 2016, and Ph.D. degree from the Hong Kong Polytechnic University in 2019. He is an Associate Researcher in the School of Computer Science at Wuhan University. His research interests include edge learning, federated learning analytics, and distributed computing.

Jiawei Jiang received his PhD degree in computer science from Peking University, China, in 2018. He is currently a professor with the School of Computer Science, Wuhan University, China. His research interests include database, big data management and analytics, and machine learning systems.