A Secure and Efficient Remote Sensing Image Classification Framework Using Chebyshev-SHA Encryption and Fox-Optimized Fast Recurrent Neural Networks

Abdullah Ghanim Jaber *, Ravie Chandren Muniyandi, Khairul Akram Zainol Ariffin

Posted Date: 18 November 2025

doi: 10.20944/preprints202511.1171.v1

Keywords: remote sensing; image classification; chebyshev chaotic map; encryption; fox-optimized fast recurrent neural networks

 doi:10.20944/preprints202511.1171.v1

Article

A Secure and Efficient Remote Sensing Image Classification Framework Using Chebyshev-SHA Encryption and Fox-Optimized Fast Recurrent Neural Networks

Abdullah Ghanim Jaber [1,3,*], Ravie Chandren Muniyandi [1,2] and Khairul Akram Zainol Ariffin [1]

[1] Research Centre for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi 43600, Selangor, Malaysia
[2] College of Computing and Informatics, Universiti Tenaga Nasional, 43000 Kajang, Selangor, Malaysia
[3] University of Information Technology and Communications, 10067, Baghdad, Iraq
* Correspondence: p131289@siswa.ukm.edu.my

Abstract

The exponential rise of remote sensing and sensor network technologies has generated massive amounts of visual data, posing challenges in safe transmission, data integrity, and categorization. In real-time applications, existing approaches fail to reconcile strong encryption, classification accuracy, and computing economy. This study aims to develop a safe and effective remote sensing image classification system that addresses both data security and intelligent analysis, enabling automated, context-aware insights in dispersed, real-time settings. The proposed work introduces the Fox-Optimized Secure Hybrid Image Encryption and Learning-based Detection (FOX-SHIELD) framework, which effectively integrates advanced encryption techniques with deep learning-based image classification, ensuring both data security and high classification accuracy for remote sensing images in real-time, distributed environments. An upgraded Chebyshev chaotic map and the Secure Hash Algorithm (SHA-256) provide dynamic, stable encryption keys in the first phase, ensuring data secrecy and integrity throughout transmission and storage. A Fast Recurrent Neural Network (FRNN) coupled with the Fox Optimization Algorithm improves convergence rate, stability, and classification accuracy even for encrypted input in the second phase. This integration enables powerful object detection while ensuring anonymity, an essential feature for sensitive remote sensing tasks. The FOX-SHIELD framework outperforms traditional models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and other hybrid encryption-learning models, in terms of classification accuracy, training convergence, and computational efficiency when applied to standard remote sensing datasets. This work addresses the fundamental issue of data security in remote sensing image classification by integrating lightweight cryptographic methods with metaheuristic deep learning optimization to enhance model accuracy, convergence, and computational efficiency in real-time applications.

Keywords: remote sensing; image classification; chebyshev chaotic map; encryption; fox-optimized fast recurrent neural networks

1. Introduction

Satellite imagery, UAVs, and sensor networks are boosting remote sensing [1]. High-resolution photography improves environmental monitoring, agriculture, disaster management, urban planning, and military surveillance [2]. Falling imaging equipment costs and the availability of commercial satellites have accelerated the acquisition of large image libraries. Large datasets offer significant advantages, such as improved model generalization and greater accuracy, due to their greater volume

 doi:10.20944/preprints202511.1171.v1

and diversity. However, they also present challenges, including increased computational demands, longer training times, and potential difficulties in data storage and management [3]. Remote sensing image classification using deep learning and a cryptographic encryption method analyzes more data but requires high bandwidth for storage, transmission, and computation [4]. To extract meaningful insights, this vast and complex imagery must be classified efficiently, ensuring computational effectiveness while supporting future data growth and safeguarding sensitive information [5].

Remote sensing imaging for critical military, infrastructure, and disaster response applications necessitates stronger cyber security [6]. Encryption may inhibit categorization due to the computational cost, and the image classification algorithms must also be able to adapt to encrypted data, which may hide important photographic characteristics [7]. Maintaining strong encryption and high classification performance is difficult in real time systems because even a few milliseconds of delay can affect the overall performance of the operation [8]. The amount of data volume and the threat landscape is increasing, so we need the framework to protect, analyze, and categorize remote sensing images on a tighter timeframe while maintaining accuracy and scaling [9].

Although traditional encryption methods, such as AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman), are effective for securing general data, they may not always be suitable for protecting the specific attributes of image data. For instance, these methods often fail to preserve the visual characteristics of images (such as edges, textures, or fine details) after encryption, which can hinder image classification tasks [10]. Processing high-resolution images could take considerable computing resources, and thus may not be ideal for remote sensing in near real time [11]. Distributed schemes will be further threatened when distributing the images between multiple nodes inherently increases the chances for interception [2]. The problem with static functions, and low-entropy key protection supports the case for the need for encryption functions based on chaotic and dynamic systems for generating keys that are far less predictable, and with lower computational cost [13]. Low-entropy keys lack the randomness needed to secure data against sophisticated cryptanalysis. This supports the case for encryption functions based on chaotic and dynamic systems, which generate keys that are far less predictable and adapt to changing conditions, offering enhanced security while maintaining lower computational cost. Such encryption techniques would be invaluable for maintaining secure, pi-locked integrity and confidentiality in positively controlled sensitive operations.

Although existing studies have explored the combination of encryption and deep learning for remote sensing image classification, challenges remain, including performance degradation, slower convergence, and increased computational costs when processed on encrypted images. These issues highlight the need for further research into lightweight encryption techniques and optimized models that can balance security and classification performance [14]. While combining deep learning and encryption offers significant security benefits, it also introduces challenges such as distortion of the feature space, slower convergence during training, and increased computational costs, making traditional deep learning approaches less effective when working with encrypted data [15]. Additionally, most deep learning models are not designed to operate in resource-constrained environments, such as those encountered in remote sensing applications, such as satellite or UAV processing in the field, where computational and memory resources are limited [16]. Performance is further reduced through the lack of integration between encryption-aware pre-processing methods and classification workflows. Moreover, optimization techniques, such as many metaheuristic search algorithms, are seldom employed to optimize neural network parameters for classifying encrypted data, leading to reduced performance on encrypted datasets [17]. These inefficiencies can be mitigated by creating specialized architectures that efficiently process encrypted data, achieving high accuracy without severely impacting the overall processing performance [18].

Remote Sensing (RS) imagery from UAVs and satellites provides crucial geographic information for applications such as infrastructure monitoring, environmental monitoring and and military observation. However, accurately classifying remote sensing (RS) images poses several challenges, including significant intra-class variability, similarities between classes, variations in object size, and

 doi:10.20944/preprints202511.1171.v1

overlapping ground objects [19]. High dimensionality, limited labeled training data, and increased costs for processing high-resolution RS remote sensing images are challenges. Additionally RS data could be intercepted or manipulated when transmitted or stored in a wireless sensor network. Although modern encryption systems provide security [20], they demand significant resources and are dependent on public keys, which can be a security risk in the remote sensing domain. In addition, while deep learning models are robust and powerful, they lack interpretability, as they are open to adversarial perturbation, and require a great deal of tuning of model parameters. The following issues may challenge the reliability of their outputs-encryption, contextualization, and misclassification of noise. It is essential that we design a more safe, robust, interpretable, and intelligent classification framework that takes advantage of the large volume of RS data, diminishes the risk of its secure processing, and can withstand related attacks [21].

There is a growing need for accurate and secure classification technologies in remote sensing (RS); environmental monitoring, disaster management, land administration, and military applications all utilize remote sensing imagery. In using remote sensor networks, it is also natural to want to intercept and modify important RS data [22]. While convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can enhance the classification process, especially RNNs, they often face challenges when applied to encrypted remote sensing (RS) data due to the lack of built-in encryption support or adaptation to encrypted domains, leading to performance inefficiencies and vulnerabilities in the RS system. Existing studies have shown that hyperspectral datasets, which involve remote sensing images captured across many narrow spectral bands (from visible to infrared), or large RS datasets, require significant computing power, prolonged training durations, and multiple optimization rounds to mitigate the negative impact of class imbalance and spatial heterogeneity.

1.1. Objectives of the research

The study aims to establish a system for remote sensing image classification that is secure and efficient, with a focus on protecting data as well as maintaining high performance based on analysis speed. To this end, an enhanced Chebyshev chaos-based encryption scheme combined with an SHA key generation mechanism is proposed, improving the encryption's security and computational efficiency while ensuring the confidentiality, integrity, and authenticity of RS image data. At the same time, a Fox-Optimized Fast Recurrent Neural Network (FRNN) is developed to make informed classifications of encrypted images, enhancing convergence speed, stability, and robustness against noise. The framework additionally incorporates region-based feature extraction and tentative contextual background information to enhance detection accuracy while lowering computational times and processing costs for large-scale applications. Extensive performance studies on real-world remote sensing datasets showed that the framework was effective in terms of classification accuracy, robustness against attacks, computational efficacy, and time to convergence for training.

2. Related Works

To shorten reconstruction time for high-resolution remote sensing images in the Internet of Things, Zhang et al. (2023) [23] proposed a lightweight Visual Cryptography-based privacy method. This study employed block-based encryption with error diffusion, Boolean-based decryption and custom training of CNN. They increased recognition accuracy while preserving privacy. Limitations include potential processing overhead in a large-scale deployment and/or the risk of perceptual distortion.

Alkhelaiwi et al. (2021) [24] presented an automated system for satellite imagery that uses a partially homomorphic Paillier encryption scheme, enabling CNN training on encrypted data while preserving privacy. The methods used were a custom CNN architecture and lighting and transfer learning for their study. They reported strong privacy guarantees while maintaining CNN accuracy. Limitations included increased computational overhead for encryption and practical issues with scaling their private solutions across large datasets obtained from satellite imagery.

Zhang et al. (2022) [25] presented a lightweight privacy-preserving framework for remote sensing images in the Internet of Things (IoT) based on visual cryptography with a stacking-to-see property.

 doi:10.20944/preprints202511.1171.v1

Approaches involve incorporating cargo information into block-based encryption and denoising neural networks to enhance quality and improve recognition accuracy. Experimental results demonstrate strong privacy preservation and classification effectiveness. Limitations include potential performance degradation under high levels of noise and computational requirements for large-scale, real-time applications.

Hou et al. (2025) [26] proposed a fast and secure cloud-based configuration for the retrieval of remote sensing images with verifiability and traceability. Approaches comprise retrieval methods based on CNN feature extraction, spectral-hashing-based dimensionality reduction with spectral rotation (SHSR), encrypted searchable index utilizing asymmetric scalar-product preserving encryption (ASPE), pixel rearrangement for predictive error (PE) labeling in image encryption, watermark embedding, and Merkle tree verification. Results demonstrate increased retrieval accuracy and privacy protections. Shortcomings include increased computational complexity and storage overhead in large deployments.

Song, H. (2023) [27] proposed a fast and simple training CNN framework (FST-EfficientNet), a one-stage deep learning CNN for semantic scene categorization of remotely sensed images using EfficientNetV2-S. A systematic data augmentation strategy that employs either a constant or a progressively growing resolution is used to enhance training effectiveness. Results achieve state-of-the-art accuracy improvements (0.8–2.7%) on the Aerial Image Dataset (AID) and the Northwestern Polytechnical University Remote Sensing Image Scene Classification 45 Dataset (NWPU-RESISC45D). Limitations pertain to the probable inadequacy in performance on very intricate datasets that lack adequate augmentation variety.

Al-Khasawneh et al. (2022) [28] proposed a chaos-based parallel image encryption technique for remotely sensed images, using Henon, Logistic, and Gauss maps, along with an external secret key. Executed on Hadoop with improved file management for TIFF/GeoTIFF, it facilitates scalable, efficient encryption of large datasets. Results demonstrate greater efficacy than current methodologies. Limitations include reduced efficiency when image dimensions are significantly reduced across large batches.

Feng et al. (2024) [29] introduced the Real Time Detection Transformer (RTDETR) and combined it with the Soft-threshold and Cascaded-Group-Attention (CGA) (SC-RTDETR), a multi-source forest remote sensing data fusion system that employs the Real-Time Detection Transformer, Soft-threshold adaptive filtering, and Cascaded-Group-Attention. These modules are designed to reduce noise from attacks and minimize the need to focus on less relevant details, resulting in more robust detection. Experiments on datasets related to pine wilt disease observed a 12.9% gain in mean Average Precision (mAP) after attack. However, potential disadvantages include increased complexity and increased computational effort, which could be drawbacks in large-scale, real-time implementations.

Similarly, Albarakati et al. (2024) [30] proposed a framework that incorporates information into Land-Use and Land-Cover (LULC) classification by designing two novel CNN (convolutional neural network) models called the Residual Self-Attention Network with six residual blocks (ResSAN6), and the Remote Sensing Inverted Residual Self-Attention Network with six inverted residual blocks (RS-IRSAN), which was built from the ground using the Bayesian Optimization Algorithm (BOA). The frameworks included Data Augmentation (DA) to mitigate class imbalance, Mutual Information-based Serial Feature Fusion (MI-SFF), Median Normalization (MN), and, finally, Arithmetic Optimization (AO) for feature selection, followed by classification with a Shallow Wide Neural Network (SWNN). The trials of using RSI-CB128 resulted in accuracies of 95.7%, 97.5%, and 92.0% on the WHU-RS19 and NWPU-RESISC45 datasets respectively. Limitations include longer processing times and greater model complexity when applied to large datasets in real time.

Ahmed et al. (2024) [31] introduced XcelNet17, a deep learning framework with fourteen convolutional layers and three fully connected layers for remote sensing image classification. A hybrid feature selection algorithm, BA-ABC, that combines the Bat Algorithm (BA) with the Artificial Bee Colony (ABC) is also introduced. XcelNet17 achieves a range of 94.6% to 99.9% on the WHU-RS19 test dataset, surpassing the metrics of AlexNet, VGG16/19, ResNet50, and DarkNet19. The BA-ABC feature

 doi:10.20944/preprints202511.1171.v1

selection method with XcelNet17 outperformed the Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), Bat Algorithm (BA), Artificial Bee Colony (ABC), and Ant Colony Optimization (ACO) with increases of up to 8%. Constraints introduce additional complexity for model training and increased challenges in scaling for large RS datasets.

Liu et al. (2025) [32] proposed a lightweight convolutional network for Remote Sensing (RS) image classification named STConvNeXt, which features a split-based mobile blocks convolution module with a hierarchical tree structure and parameterized depthwise separable convolutions to reduce complexity. A fast pyramid pooling module provides a large context, and a dynamic threshold loss enhances class separability. Using AID as the primary dataset, STConvNeXt exhibits a reduction in parameters (56.49%) and FLOPs (49.89%) relative to ConvNeXt, with improvement in accuracy of 1.2–2.7%. Possible limitations are performing worse with extremely high-resolution remote sensing data or multi-modal remote sensing data.

Song et al. (2024) [33] introduced a differentiable Neural Architecture Search (NAS) for RS image classification that uses a binary gate to establish partial channel connections, thereby reducing parameter and memory usage. The study results show a 15.1% increase in validation accuracy at the search phase, while reducing search time by 88% and parameters by 84% when compared to DARTS. Limitations included a 4.5% reduction in accuracy when compared to DARTS, and possible reduced tuning capabilities across many RS datasets.

Rasheed et al.(2021) [34] proposed a trustworthy RS-embedded approach for classifying high-resolution satellite images, while accounting for photometric and geometric distortions. Features generated from the last fully connected layer of a Deep Neural Network (DNN), were classified using a multi-class Support Vector Machine (SVM) with a Gaussian kernel. This approach achieved an accuracy of 93.8% using data from China's Headwater Region, reported as a significant improvements over contemporary methods. The only limitations were based on manual kernel parameters and scalability with data in instances that utilize many different RS datasets. Table 1 below is summary of the reviewed papers across RS image processing methods.

Table 1. Summary of recent RS image processing methods, highlighting main contributions, applied techniques, achieved results, and identified limitations in privacy preservation, classification accuracy, and computational efficiency.

Ref. & Authors	Main Contribution	Methods	Results	Limitations
Zhang et al. (2023) [23]	Lightweight privacy-preserving detection in IoT RS images	Visual Cryptography (VC), block-based encryption, error diffusion, Boolean decryption, optimized DL models	High recognition accuracy + privacy	Processing overhead, visual distortions
Alkhelaiwi et al. (2021) [24]	Privacy-preserving CNN training on encrypted satellite images	Paillier encryption, bespoke CNN, transfer learning	Maintains accuracy with strong privacy	High computing cost, scalability issues

 doi:10.20944/preprints202511.1171.v1

Table 1 (continued)

Ref. & Authors	Main Contribution	Methods	Results	Limitations
Zhang et al. (2022) [25]	Lightweight VC framework with stacking-to-see	Block-based encryption, denoising NN	Strong privacy + recognition	Performance drop with high noise, real-time cost
Hou et al. (2025) [26]	Secure cloud-based RS image retrieval with traceability	CNN features, SHSR, ASPE, pixel rearrangement, watermark, Merkle tree	High retrieval precision + privacy	High computation and storage overhead
Song, H. (2023) [27]	Fast, simple CNN for RS scene categorization	FST-EfficientNet (EfficientNetV2-S), constant/progressive resolution augmentation	SOTA accuracy gain (0.8–2.7%)	May underperform on very complex datasets
Al-Khasawneh et al. (2022) [28]	Chaos-based parallel RS image encryption	Henon, Logistic, Gauss maps, Hadoop parallel processing	Higher encryption efficiency	Lower efficiency for very small images
Feng et al. (2024) [29]	Adversarially robust forest RS detection	SC-RTDETR, Soft-threshold filtering, Cascaded-Group-Attention	mAP ↑12.9% under attack	High complexity, computational cost
Albarakati et al. (2024) [30]	LULC classification with dual CNNs + fusion	ResSAN6, RS-IRSAN, DA, MI-SFF, MN, AO, SWNN	Acc: 95.7%, 97.5%, 92.0%	High computation, complex model
Ahmed et al. (2024) [31]	XcelNet17 + BA-ABC hybrid feature selection	14 conv + 3 FC layers, BA-ABC	Acc: 94.6–99.9%, +8% over baselines	Complex training, scaling issues
Liu et al. (2025) [32]	Lightweight RS classification CNN	STConvNeXt, depthwise separable conv, fast pyramid pooling, dynamic threshold loss	Params ↓56.49%, FLOPs ↓49.89%, Acc ↑1.2–2.7%	May struggle on high-res/multimodal data

 doi:10.20944/preprints202511.1171.v1

Table 1 (continued)

Ref. & Authors	Main Contribution	Methods	Results	Limitations
Song et al. (2024) [33]	Efficient NAS for RS classification	Differentiable NAS, binary gate, partial channel connection	Acc ↑15.1% vs DDSAS, Time ↓88%, Params ↓84% vs DARTS	Slightly lower acc than DARTS, tuning difficulty
Rasheed et al. (2021) [34]	Robust RS classification under geometric/photometric variation	DNN features + multi-class SVM (Gaussian kernel)	Acc: 93.8%	Manual kernel tuning, scalability issues
Jaber & Muniyandi (2021) [36]	Hybrid deep learning-based face recognition	Gabor filters, stacked sparse autoencoders (SSAE), DNN	Improved recognition accuracy and feature extraction efficiency	High training cost, illumination sensitivity
Rahman et al. (2020) [37]	ML-based feature selection and classification review for ASD	Comparative analysis of ML classifiers and FS techniques	Identified efficient ML-FS combinations	Not directly RS-related, lacks privacy context
Sihwail et al. (2021) [38]	Memory-based malware detection and classification	Feature engineering, DNN, memory forensics	High detection accuracy and robustness	Limited to malware domain, non-RS adaptation
Hasan et al. (2021) [39]	Lightweight encryption for medical image security	Stream cipher, key generation, hash-based encryption	High encryption speed, enhanced IoMT privacy	Limited scalability beyond medical domain
Talukdar et al. (2021) [40]	Secure communication via IDS and digital signature	IDS integration, AODV optimization, digital signature	Improved packet delivery and attack detection	Designed for ad hoc networks, limited RS applicability

New advances in secure image processing have brought attention to the need for efficient frameworks that can strike a balance between security, accuracy, and efficiency in categorization. For face recognition, Jaber and Muniyandi [36] proposed hybrid deep learning architectures to promote the deployment of top-tier stacked deep models in critical domains. Both feature extraction and autoencoders were incorporated into these designs. Machine learning is essential for selecting relevant features and making accurate classifications, as Rahman et al. [37] previously discussed. The idea is closely related to the improvement of remote sensing models that rely on neural networks. The ability to swiftly review data and memory is crucial for processing massive amounts of information, including data collected from remote sensing. Research by Sihwail et al. [38] proved the importance of malware detection. In order to secure medical images, Hasan and colleagues [39] developed small encryption techniques. They demonstrated the efficacy of hybrid encryption in safeguarding sensitive information.

 doi:10.20944/preprints202511.1171.v1

Additionally, Talukdar et al. [40] demonstrated how intrusion detection and optimization could improve network security by studying secure communication protocols. Problems with combining ideal neural networks with durable encryption have been uncovered by these studies' combined findings. This enhancement was made possible by the well-coordinated plan that FOX-SHIELD implemented.

2.1. Identified Research Gaps

A comprehensive review of the existing literature on secure and efficient remote sensing image classification reveals several significant research gaps. While many studies have investigated privacy-preserving mechanisms such as visual cryptography, chaos-based encryption, and homomorphic encryption, there is a clear lack of integration between strong encryption and deep learning approaches designed for encrypted input data, which often results in decreased classification performance, prohibitive computational resources, or both [38,39]. Second, many modern encryption schemes use static or predictable keys, which defeats the security purpose of encryption and also lacks a dynamic, high-entropy backdoor to be effective against successful cryptanalytic attacks. Lastly, many privacy-preserving schemes impose high computational and storage burdens and can't be replicated or run on real-time or resource-constrained platforms, such as UAS and onboard satellites.

Fourth, deep learning algorithms designed for unencrypted remote sensing (RS) images often exhibit lower accuracy on encrypted datasets at the model's output space, yet lack an appropriate procedure to address this [23,24]. Fifth, there is limited use of metaheuristic optimization to tune classification network parameters in the encrypted space, which could benefit models by improving network convergence and resilience. Additionally, a few of the approaches use either contextual modeling or region-specific features, which are actually essential in complex RS environments. Lastly, most studies are evaluated under stable conditions, with little testing in dynamic, real-time operational environments—leaving a large gap in research on robustness and implementation feasibility.

3. Proposed Methodology

The proposed FOX-SHIELD framework offers a secure and effective method for remote sensing image classification by integrating an enhanced Chebyshev chaotic map with dynamic key generation utilizing the Secure Hash Algorithm (SHA) and a Fox-Optimized Fast Recurrent Neural Network (FRNN). Classification makes use of FRNN's spatial-temporal correlations in encrypted imagery. The Fox Optimization Algorithm (FOA) tailors network parameters concurrently in order to enhance the convergence speed, stability, as well as resistance against noise and attacks. Figure 1 illustrates the entire suggested architecture of the system.

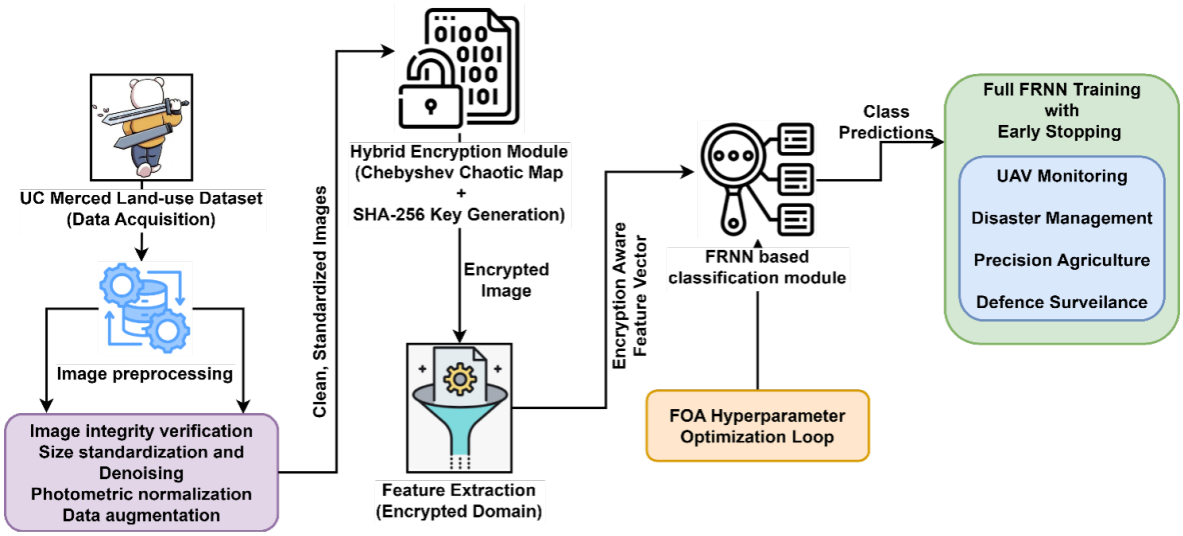


Figure 1. Overall Architecture of the FOX-SHIELD Framework

 doi:10.20944/preprints202511.1171.v1

The proposed FOX-SHIELD (Fox-Optimized Secure Hybrid Image Encryption and Learning-based Detection) framework integrates advanced encryption, feature learning, and optimization modules to achieve secure and accurate remote sensing image classification. The overall architecture consists of five primary stages: data acquisition, pre-processing, hybrid encryption, feature extraction and classification, and optimization-assisted model training. The process begins with data acquisition, where the UC Merced Land Use Dataset is employed, comprising 21 distinct land-use classes representing diverse urban and peri-urban environments. Following this, an image pre-processing module ensures uniformity and data quality through integrity verification, image resizing, denoising, photometric normalization, and data augmentation. These steps standardize the dataset and enhance intra-class variability, which is crucial for improving model generalization. Next, the images are processed through the Hybrid Encryption Module, which combines a Chebyshev Chaotic Map with SHA-256 key generation to perform secure encryption. This hybrid approach ensures strong data confidentiality and resistance to cryptographic attacks while retaining structural integrity suitable for feature extraction in the encrypted domain. The resulting encrypted images maintain privacy without compromising analytical usability. The Feature Extraction Module operates directly in the encrypted domain, utilizing a Fast Recurrent Neural Network (FRNN) to learn discriminative representations from the encrypted data. The FRNN-based classification module captures both spatial and temporal dependencies, facilitating accurate categorization of encrypted remote sensing images. A Fox Optimization Algorithm (FOA) Hyperparameter Optimization Loop is employed to fine-tune the FRNN's parameters such as learning rate, number of recurrent units, and regularization coefficients ensuring optimal performance and convergence stability. This metaheuristic optimization mechanism enhances classification accuracy and reduces computational redundancy. Finally, full FRNN training with early stopping is performed to prevent overfitting and ensure efficient convergence. The trained FOX-SHIELD model demonstrates strong applicability across multiple domains, including UAV monitoring, disaster management, precision agriculture, and defense surveillance, where secure, high-accuracy classification and data confidentiality are critical.

This method simplifies analysis and reduces overhead, enabling real or near real-time operation, particularly in resource-constrained environments such as drones or onboard satellite computers. The FOX-SHIELD framework offers a distinct advantage by integrating robust privacy-preserving mechanisms with high classification accuracy, maintaining performance consistency even under complex operational conditions and across heterogeneous datasets. FOX-SHIELD could be used effectively in environmental monitoring, disaster response, precision agriculture, urban planning, and defense monitoring applications where a high level of classification accuracy and research-level capabilities for security and privacy protection are critical [41]. FOX-SHIELD presents a unique study that combines effective, yet chaotic encryption with a more robust form of deep learning to solve two related problems at the same time: data privacy and processing efficiency for current remote sensing applications.

3.1. Data Acquisition

The experimental data utilized in this study is derived from the UC Merced Land [35], an open-access benchmark dataset designed for land-use and remote sensing studies. The dataset comprises 21 distinct land-use classes, each representing specific urban and peri-urban environments such as agricultural fields, highways, golf courses, port areas, and residential neighborhoods. Each individual class contains 100 RGB images which are 1 foot by 1 foot resolution per pixel (256 × 256 pixels). All images were individually extracted from the large aerial photos provided by the USGS National Map metropolitan Areas Imagery collection (which also includes several metropolitan areas in the US) which creates a rich dataset that has a wide variety of geographic and environmental conditions for researchers to inspect classification performance in a more realistic setting.

The dataset is carefully organized and balanced to ensure equal representation across all 21 land-use classes, thereby preventing class dominance and ensuring unbiased model training and evaluation. This balance is achieved by maintaining an identical number of samples per class and

 doi:10.20944/preprints202511.1171.v1

employing data augmentation techniques such as rotation, flipping, scaling, and noise injection to enhance sample diversity while preserving the overall class distribution. The high resolution, diverse classes, and equal balance of attributes all characterize this dataset as ideal for the secure and effective testing of classification systems as proposed by the FOX-SHIELD system. Figure 2 shows example images from our dataset.

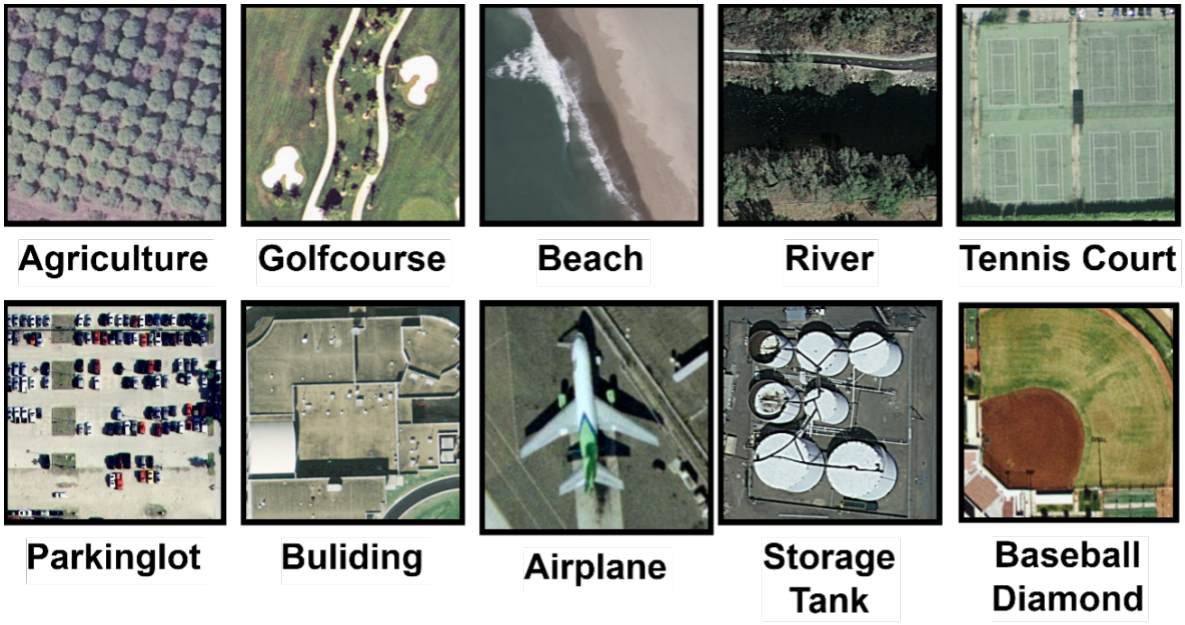


Figure 2. Sample images from the dataset [35] suitable for the FOX-SHIELD framework

3.2. Image Preprocessing

The first step in pre-processing the UC Merced Land Use Dataset involves obtaining all 21 scene types, each with exactly 100 RGB images of dimensions 256×256 pixels at a spatial resolution of 1 foot per pixel. The integrity of the dataset is confirmed by a mix of SHA-256 hashing for precise byte-level identification verification and perceptual hashing (pHash) for visual similarity, hence verifying the integrity of the image set, $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$, where N represents the entire number of images, devoid of duplic where N represents the entire number of images, devoid of duplicates or corrupted entries. A stratified splitting approach is utilized to partition I into training (D_{train}), validation $(\mathcal{D}_{\text{val}})$, and evaluation $(\mathcal{D}_{\text{test}})$ subsets, which maintain the class distribution for each category $c \in \mathcal{C}$, as defined in Equation (1).

$$\frac{|\mathcal{D}_{\text{train}} \cap c|}{|\mathcal{D} \cap c|} = \alpha, \quad \frac{|\mathcal{D}_{\text{val}} \cap c|}{|\mathcal{D} \cap c|} = \beta, \quad \frac{|\mathcal{D}_{\text{test}} \cap c|}{|\mathcal{D} \cap c|} = \gamma \tag{1}$$

where $\alpha = 0.70$, $\beta = 0.15$, and $\gamma = 0.15$ denote the proportions for training, validation, and testing, respectively. All images are verified to possess consistent dimensions; any discrepancies are rectified using center cropping C or reflection padding P to fit into the fixed domain $\Omega = [0, 256] \times [0, 256]$. If image noise is identified, light denoising is executed utilizing a Gaussian smoothing kernel $G_\sigma * I$ for uncorrelated noise or a Wiener filter $W(I)$ for spatially linked artifacts. Subsequently, photometric standardization is implemented as defined in Equation (2), wherein each pixel intensity $I_c(x, y)$ is adjusted. The value (x,y) in channel $c \in \{R, G, B\}$ at spatial coordinates (x, y) is normalized.

$$\hat{I}_c(x, y) = \frac{\frac{I(x,y)}{255} - \mu_c}{\sigma_c} \tag{2}$$

where $\mu_c = \frac{1}{|D_{train}| \cdot |\Omega|} \sum_{I \in D_{train}} \sum_{(x,y) \in \Omega} \frac{I_c(x,y)}{255}$ denotes the average pixel value for channel c across the training dataset, and σ_c denotes the associated standard deviation. Equation (3) applies data augmentation to the training set to increase generalization.

 doi:10.20944/preprints202511.1171.v1

$$T(I) = \mathcal{N}_\sigma\Big(\mathcal{B}_{\delta_b}\big(\mathcal{F}_{p_f}(\mathcal{R}_\theta(I))\big)\Big), \tag{3}$$

where $\mathcal{R}_\theta$ rotates the images by $\theta \in \{0°, 90°, 180°, 270°\}$, $\mathcal{F}_{p_f}$ applies flips, either vertical or horizontal, with probability p_f, $\mathcal{B}_{\delta_b}$ modifies the brightness within$[-\delta_b, +\delta_b]$, and $\mathcal{N}_\sigma$ incorporates Gaussian noise $\mathcal{N}(0, \sigma^2)$ where $\sigma \leq 2/255$. For encryption preparation, each normalized image $\hat{I}$ is separated into non-overlapping chunks $B_k \subset \Omega$ of size $b \times b$, where b is a divisor of 256, and then sent to the encryption function of Chebyshev-SHA, $\mathcal{E}_{\text{ChebSHA}}(\hat{I})$ to generate the encrypted image E. Post-encryption normalization, as defined in Equation (4), is used when encrypted images are to be used for classification.

$$\hat{E}_c(x, y) = \frac{E_c(x, y) - \mu_c^{\text{enc}}}{\sigma_c^{\text{enc}}}. \tag{4}$$

in which μ_c^{enc} and σ_c^{enc} represent the channel-c mean and standard deviation throughout the encrypted training set. Lastly, a manifest file that links each image to its class label and encryption key, without disclosing sensitive relationships, is included with all processed (and encrypted) images, which are stored in efficient formats such as `LMDB` or `WebDataset` shards. For later Fox-Optimized FRNN classification, this preparation pipeline guarantees dataset cleanliness, spatial and photometric consistency, security preparedness, and adequate storage.

Figure 3 illustrates the comprehensive preprocessing workflow used for raw remote sensing images before categorization. Each step has its own explicit goal; standardizing the size assures that models receive images all of the same size, denoising will minimize noise from the sensor, normalization will standardize contrast, augmentation will increase variability in the dataset, and obfuscation encrypts sensitive data for security. These steps can all be modified, of course, but the spatial structural representation is applicable to encrypted domain classification and demonstrates that our offline methodology can maintain a level of compatibility with machine learning while preserving data safety.

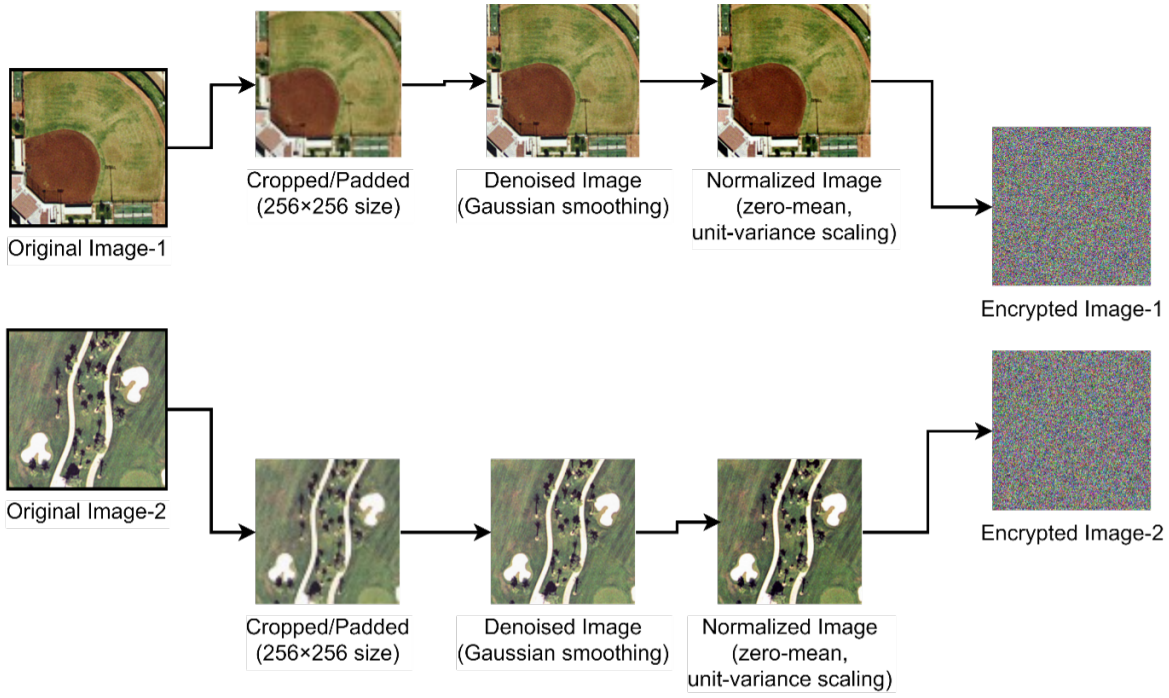


Figure 3. Stepwise preprocessing transformations for two UC Merced Land Use dataset images

3.3. Hybrid Encryption using Chebyshev–SHA

The proposed encryption framework uses an advanced Chebyshev chaotic map in combination with the Secure Hash Algorithm (SHA) to generate dynamic, per-image encryption keys and to apply

 doi:10.20944/preprints202511.1171.v1

strong permutation-diffusion encryption to RS images, while maintaining structural compatibility for classification in the encrypted domain. The combination method provides for the confidentiality, integrity, and validity of images during transmission, while also retaining the critical statistical characteristics of RS images required for classification. The entire procedure may be articulated as a linked transformation on the pre-processed image tensor $P \in \mathbb{Z}^{256\times256\times3}$, integrating pixel position scrambling and intensity masking in a manner suitable for deep neural network inputs. Algorithm 1 shows the FOX-SHIELD Encryption (Chebyshev-SHA).

Algorithm 1: FOX-SHIELD Encryption (Chebyshev-SHA)

Input: RS image I, secret key K
Output: Encrypted image E

Step 1: Generate dynamic key
Compute hash seed: $hash\_seed \leftarrow \text{SHA-256}(I \| K)$ `// SHA generates image-specific seed`
Convert hash seed to numeric form: $initial\_condition \leftarrow \text{convert\_to\_numeric}(hash\_seed)$

Step 2: Generate chaotic sequence
$chaotic\_seq \leftarrow \text{Chebyshev\_Map}(initial\_condition, N)$ `// N = number of pixels`

Step 3: Permutation step
$permuted\_image \leftarrow \text{Permute\_Pixels}(I, chaotic\_seq)$

Step 4: Diffusion step
$E \leftarrow \text{Diffuse\_Pixels}(permuted\_image, chaotic\_seq)$

return E

Figure 4 demonstrates the Chebyshev–SHA hybrid encryption method, which begins with a pre-processed image that is subjected to SHA-256-based key generation and Chebyshev chaotic mapping. HMAC-SHA256 is used for integrity verification after this key powers' encryption via permutation and diffusion. The resulting encrypted image preserves data confidentiality while remaining suitable for classification within the encrypted domain.

Chaotic maps are widely used in cryptography due to their sensitivity to initial conditions, topological mixing, and pseudorandom properties. Specifically, Chebyshev polynomials of the first kind offer several advantages, including simple analytical forms, superior computational efficiency, and established ergodicity on the interval $(-1, 1)$. For an integer order $(m \geq 2)$, the Chebyshev map is defined as the following Equation (5).

$$T_m(x) = \cos\big(m \arccos x\big), \quad x \in (-1, 1). \tag{5}$$

This recurrence generates sequences that exhibit strong sensitivity to x_0 and m. Even minor alterations in the initial state x_0 or polynomial degree m results in completely distinct sequences a phenomenon referred to as the butterfly effect. To augment chaotic behavior and mitigate attacks that exploit finite-precision constraints, this research utilize a two-dimensional improved Chebyshev system with XOR-based perturbations as shown in Equation (6).

$$\left.\begin{aligned} x_{k+1} &= T_m(x_k) \;\oplus\; \left(\frac{(v_k \oplus u_k) \bmod 256}{255} - 1\right) \\ y_{k+1} &= T_n(y_k) \end{aligned}\right\} \tag{6}$$

 doi:10.20944/preprints202511.1171.v1

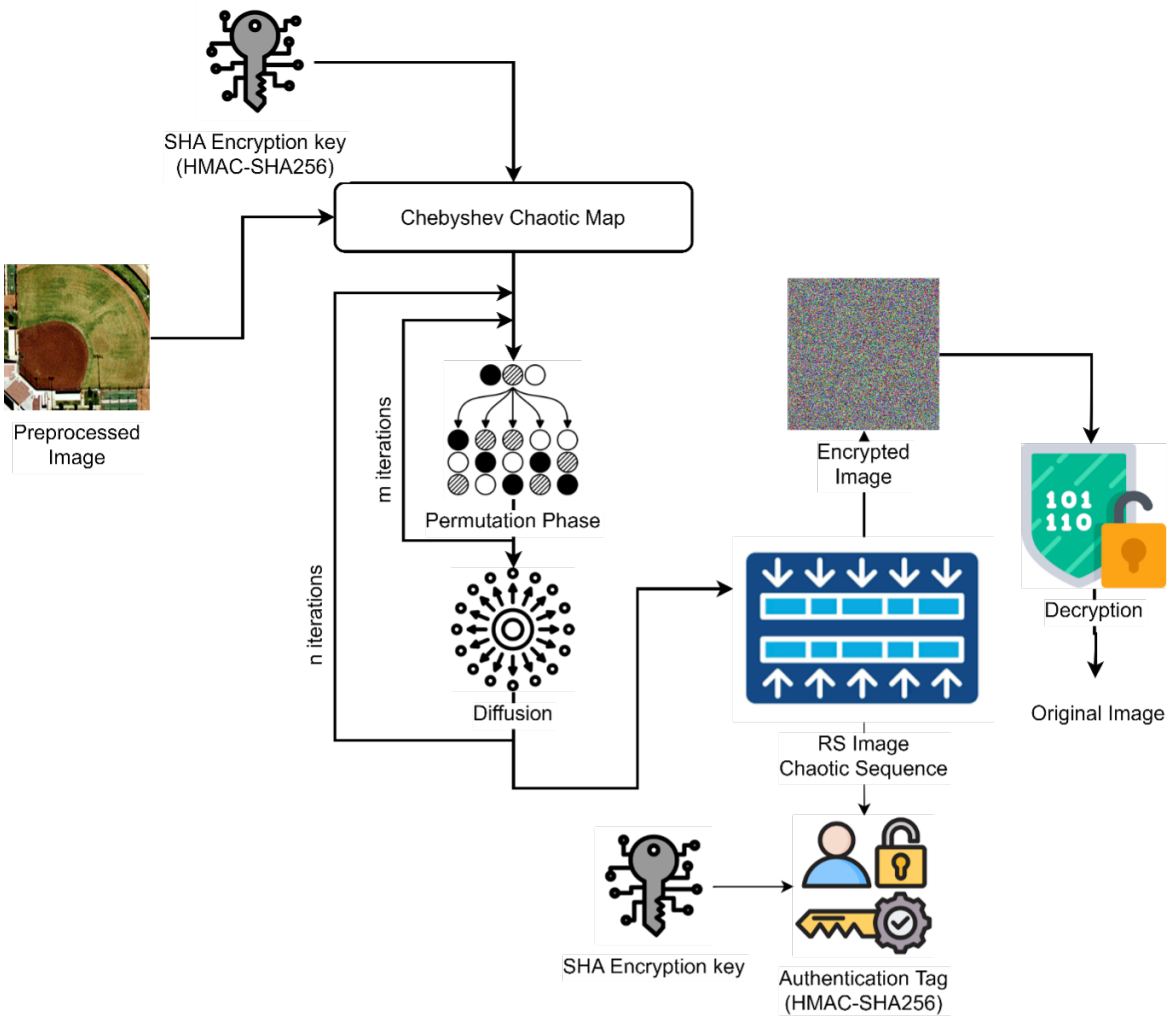


Figure 4. Structure of Hybrid Encryption using Chebyshev–SHA

In this context, $\oplus$ represents the byte-wise XOR operation, u_k and v_k are the integer sequences that are normalized to the interval $(-1, 1)$, and m, n represent separate polynomial orders dynamically generated for each image. Dynamic key creation is accomplished by utilizing SHA-256 on a concatenation of an image identification, a random nonce r, a timestamp, and the image hash itself (Equation (7)).

$$h = \text{SHA-256}\big(ID \,\|\, r \,\|\, \text{timestamp} \,\|\, \text{SHA-256}(I)\big) \tag{7}$$

From the digest h, the starting seeds and polynomial orders are recovered as expressed in Equation (8).

$$\left.\begin{aligned} x_0 &= \frac{2 \cdot \text{int}(h[0{:}8])}{2^{64}} - 1, & y_0 &= \frac{2 \cdot \text{int}(h[8{:}16])}{2^{64}} - 1, \\ m &= 2 + \big(\text{int}(h[16{:}20]) \bmod 13\big), & n &= 2 + \big(\text{int}(h[20{:}24]) \bmod 17\big) \end{aligned}\right\} \tag{8}$$

The same digest supplies the encryption key K_{enc} and message authentication key K_{mac}. Permutation is executed in blocks to preserve the coarse spatial structure. For a block size $b \,|\, 256$, pixels are reorganized according to a permutation vector $\pi = \text{argsort}\Big(\{u_k\}_{k=1}^{b^2}\Big)$, yielding the function shown in Equation (9).

$$p^{(\text{conf})} = \Pi \cdot p \tag{9}$$

 doi:10.20944/preprints202511.1171.v1

where Π is the permutation matrix, and p represents the vectorized block. After permutation, diffusion disseminates slight intensity variations throughout the entire ciphertext via chaotic keystream masking (Equation (10)).

$$c_i = \left(p_i^{(\text{conf})} + k_i + c_{i-1}\right) \bmod 256, \qquad c_0 \leftarrow \text{int}(h[24{:}28]) \tag{10}$$

The resulting encrypted tensor E maintains the original dimensions of $256 \times 256 \times 3$ to facilitate classification in the encrypted domain. An HMAC-SHA256 tag confirms authenticity and integrity as defining in Equation (11).

$$\tau = \text{HMAC\_SHA256}(K_{\text{mac}}, \text{header}) \,\|\, E \tag{11}$$

The header comprises non-confidential parameters $(r, b, m, n, K_{\text{burn}})$ to facilitate decryption without disclosing confidential seeds. This permutation and diffusion method works with chaotic key generation for each image to ensure the encrypted image adheres to cryptographic standards of robustness while being structurally compatible with the later Fox-Optimized FRNN classifier, allowing remote sensing image analysis to be secured and efficient.

Algorithm 2 introduces a dynamic key-generation/scheduling approach based on SHA-256 that leverages a modified Chebyshev chaos map to enable secure, content-aware encryption of remote sensing images. In the suggested encryption protocol, HMAC-SHA256 is first implemented to produce a dynamic, image-specific key. The HMAC-SHA256 algorithm takes image data along with a secret key and returns a fixed length, highly sensitive hash to provide data integrity and authenticity. The hash is then converted into a numeric value that serves as the initial conditions for the Chebyshev chaotic map. Once the hash value is selected as the initial seed, the Chebyshev map produces a pseudo-random, chaotic sequence to control the permutations and diffusion processes applied to the image pixels. The dual phase integration (first HMAC-SHA256, and then Chebyshev map) insures that each image has a unique encryption key while preserving the original image dimensions and structural aspects which is important to label data for later classification in the encrypted domain. Consequently, the framework simultaneously guarantees robust security, tamper detection, and compatibility with encrypted-domain learning models.

3.4. Feature Extraction from Encrypted Image

The objective of encryption-aware feature extraction is to generate descriptors that retain their informative value despite permutation diffusion encryption, utilize the maintained block-level statistics, and incorporate region-specific and contextual information for the classifier. Consequently, we get a collection of complementary properties for each block B_j (non-overlapping $b \times b$ blocks) and aggregate them across spatial scales into a final feature vector $\mathbf{f}$ for each image. The block index j ranges from 1 to J, where $J = \frac{256 \times 256}{b^2}$

Initially, calculate straightforward yet resilient block statistics that are invariant to the permutation order within blocks and are resistant to diffusion noise post-normalization. For block B_j with pixel values $p_{j,i} \in \{0, \ldots, 255\}$ (linearized over pixels $i = 1..b^2$ and channels processed either independently or collectively), the block mean μ_j, variance σ_j^2, skewness s_j, and kurtosis κ_j are defined in the following equation (12).

$$\left.\begin{aligned} \mu_j &= \frac{1}{b^2}\sum_{i=1}^{b^2} p_{j,i}, & \sigma_j^2 &= \frac{1}{b^2}\sum_{i=1}^{b^2}\left(p_{j,i} - \mu_j\right)^2, \\ s_j &= \frac{1}{b^2}\sum_{i=1}^{b^2}\left(\frac{p_{j,i} - \mu_j}{\sigma_j}\right)^3, & \kappa_j &= \frac{1}{b^2}\sum_{i=1}^{b^2}\left(\frac{p_{j,i} - \mu_j}{\sigma_j}\right)^4 - 3 \end{aligned}\right\} \tag{12}$$

Following the same encrypted-domain normalization, these low-order moments maintain discriminative patterns across classes due to diffusion's usage of chained modular addition. The subsequent step is to extract energy spectrum characteristics using the block discrete cosine transform (DCT). For block B_j, calculate the 2D-DCT coefficients $C_j(u, v)$ for $u, v = 0, \ldots, b-1$, and derive compact

 doi:10.20944/preprints202511.1171.v1

Algorithm 2: Hybrid Encryption using Chebyshev–SHA in FOX-SHIELD

Input:
$P \in \mathbb{Z}^{256\times256\times3}$ // preprocessed RGB image tensor
ID // image identifier / session id
r // 128-bit random nonce
$timestamp$ // capture or processing time
b // block size ($b \in \{8, 16, 32\}$ with $b \mid 256$)
K_{burn} // burn-in iterations for chaos (e.g., 100)
Output:
$E \in \mathbb{Z}^{256\times256\times3}$ // encrypted image (same shape as P)
τ // HMAC-SHA256 authentication tag
Procedure:
Step 1: Digest & keys
$h \leftarrow \text{SHA256}(ID \,\|\, r \,\|\, timestamp \,\|\, \text{SHA256}(P))$
$x_0 \leftarrow 2 * \frac{\text{int}(h[0:8])}{2^{64}} - 1$ // seed in $(-1, 1)$
$y_0 \leftarrow 2 * \frac{\text{int}(h[8:16])}{2^{64}} 2^{64} - 1$
$m \leftarrow 2 + (\text{int}(h[16:20]) \bmod 13)$ // Chebyshev order ≥ 2
$n \leftarrow 2 + (\text{int}(h[20:24]) \bmod 17)$
$c_0 \leftarrow \text{int}(h[24:28]) \bmod 256$ // diffusion IV (byte)
$K_{\text{enc}} \leftarrow h[0:16]$ // encryption key material
$K_{\text{mac}} \leftarrow h[16:32]$ // MAC key material
Step 2: Chaotic sequences (enhanced Chebyshev)
$x \leftarrow x_0; \quad y \leftarrow y_0$
For $k = 1 \ldots K_{\text{burn}}$ // burn-in
$x \leftarrow \cos(m * \arccos(x))$
$y \leftarrow \cos(n * \arccos(y))$
For $k = 1 \ldots L = b * b$: // per-block length
$x \leftarrow \cos(m * \arccos(x))$
$y \leftarrow \cos(n * \arccos(y))$
$u_k \leftarrow \text{floor}(256 * \frac{(x+1)}{2})$ // 0...255
$v_k \leftarrow \text{floor}(256 * \frac{(y+1)}{2})$
Step 3: Build a per-block permutation from chaos
$\pi \leftarrow \text{argsort}(\{u_k\}_{k=1}^{b^2})$ // stable order
$\Pi \leftarrow \text{perm\_matrix}(\pi)$
Step 4: Permutation (confusion) inside each $b \times b$ block
For each channel $c \in \{R, G, B\}$
For each non-overlapping block B_c of size $b \times b$ in $P[:, :, c]$
$p \leftarrow \text{vec}(B_c)$
$p_{\text{conf}} \leftarrow \Pi \cdot p$
write p_{conf} back into block position
Step 5: Keystream generation (stream KDF)
For each block index j and (channel c), derive a block key:
$\text{seed}_j \leftarrow \text{SHA256}(K_{\text{enc}} \,\|\, j \,\|\, c)$
Expand seed_j into $|block|$ bytes $\rightarrow \{k_i\}$ for the block
Step 6: Diffusion (chained modular addition)
$c_{\text{prev}} \leftarrow c_0$
For $i = 1 \,..\, (256 \times 256 \times 3)$ // linearized order per block layout
$c_i \leftarrow (p_i^{\text{conf}} + k_i + c_{i-1}) \bmod 256$
$c_{\text{prev}} \leftarrow c_i$
Reshape $\{c_i\}$ back to image E
Step 7. Authentication tag)
header $\leftarrow (r, b, m, n, K_{\text{burn}})$ // non-secret params
$\tau \leftarrow \text{HMAC\_SHA256}(K_{\text{mac}}, \text{header} \,\|\, E)$
return (E, τ)

 doi:10.20944/preprints202511.1171.v1

energy descriptors, including low-frequency energy E_j^{LF} and high-frequency energy E_j^{HF} as shown in Equation (13).

$$C_j(u,v) = \mathrm{DCT2}(B_j)[u,v]; E_j^{LF} = \sum_{u=0}^{k}\sum_{v=0}^{k} |C_j(u,v)|^2; E_j^{HF} = \sum_{(u,v)\notin[0..k]^2} |C_j(u,v)|^2 \Bigg\} \tag{13}$$

where k (e.g., $k = 1$ or 2) designates the low-frequency band. The relatively low-frequency ratio $r_j^{\mathrm{DCT}} = \dfrac{E_j^{LF}}{E_j^{LF} + E_j^{HF}}$ captures coarse texture maintained through block-wise permutation. To capture local texture patterns that withstand order-preserving ambiguity, calculate block-level Local Binary Pattern (LBP) histograms H_j^{LBP}. For every pixel in B_j, define its neighborhood thresholding as $\mathrm{LBP}(p) = \sum_{n=0}^{P-1} \mathbf{1}\{p_n \geq p\}\, 2^n$ (P neighbors). Subsequently, construct the normalized histogram using the expression in Equation (14).

$$H_j^{\mathrm{LBP}}[r] = \frac{1}{b^2}\sum_{i=1}^{b^2} \mathbf{1}\{\mathrm{LBP}(p_{j,i}) = r\}, \quad r = 0, 1, \ldots, R-1. \tag{14}$$

Due to the localized nature of LBP and its reliance on relative ordering, block-local shuffles maintain the multiset of LBP codes, thereby ensuring the histogram's robustness under permutation. Edge and orientation information is obtained by gradient-energy and Gabor responses calculated for each block. Let ∇_x, ∇_y be discrete derivatives; the gradient magnitude at each pixel is $g_{j,i} = \sqrt{(\nabla_x p_{j,i})^2 + (\nabla_y p_{j,i})^2}$. The block mean gradient $\bar{g}_j$ and gradient entropy H_j^g are defined in the following Equation (15).

$$\left.\begin{aligned} \bar{g}_j &= \frac{1}{b^2}\sum_i g_{j,i}, \\ H_j^g &= -\sum_q q_j(q)\, \log q_j(q) \end{aligned}\right\} \tag{15}$$

Here, $q_j(q)$ represents the empirical histogram of quantized gradient magnitudes within block j. For directional features, convolve block B_j with a bank of Gabor kernels $\psi_{\theta,\lambda}$ and compute the energy as in Equation (16).

$$G_j(\theta,\lambda) = \sum_{x,y} \left| (B_j * \psi_{\theta,\lambda})(x,y) \right|^2 \tag{16}$$

These responses approximate edge structures that endure local permutations when calculated as block energies. FOX-SHIELD calculates multi-scale pooled descriptors and region adjacency relations to incorporate region-specific information and contextual signals. Establish spatial pyramid pooling (SPP) levels L (e.g., $1\times1, 2\times2, 4\times4$). At each level s, partition the image into S_s cells and calculate pooled statistics $\mu_{s,t}, \sigma_{s,t}$ (mean/variance) and pooling histograms $H_{s,t}^{\mathrm{LBP}}$. Combining over several scales yields the following expression in Equation (17).

$$F_{\mathrm{SPP}} = \bigoplus_{s=1}^{L} \bigoplus_{t=1}^{S_s} \left[\mu_{s,t}, \sigma_{s,t}, H_{s,t}^{\mathrm{LBP}}, r_{s,t}^{\mathrm{DCT}}\right] \tag{17}$$

where $\oplus$ signifies concatenation, and each pooled cell assimilates the properties of its constituent blocks. This encodes contextual backdrop cues (coarse-to-fine) and preserves region structure intact under block-wise variations, as cell-level pooling collects across multiple blocks. To explicitly model region adjacency and spatial context, create a Region Adjacency Graph (RAG) using superpixels derived from the post-decryption block grid (i.e., using block cells as superpixels). Consider nodes as blocks B_j with feature vectors b_j (concatenation of the block statistics mentioned in Equation (17)).

 doi:10.20944/preprints202511.1171.v1

Construct edges $E_{j,k}$ between spatially contiguous blocks and calculate graph characteristics such as average neighbor contrast, defined in Equation (18).

$$\Delta_j = \frac{1}{N(j)} \sum_{k \in N(j)} \|b_j - b_k\|^2 \tag{18}$$

and local clustering coefficient C_j. Aggregating $\{\Delta_j, C_j\}$ across the graph encodes spatial coherence that is impervious to within-block permutation. Ultimately, construct the joint feature vector for each image by concatenating block-level, spectral, texture, multi-scale pooling, and graph-context components, followed by the application of dimensionality reduction and normalization, as expressed in Equation (19).

$$f = \mathcal{N}\left(\mathrm{PCA}_m\left(\bigoplus_{j=1}^{J} \left[\mu_j, \sigma_j, r_j^{\mathrm{DCT}}, H_j^{\mathrm{LBP}}, \bar{g}_j, G_j, \Delta_j, C_j\right]\right)\right) \tag{19}$$

Here, $\mathrm{PCA}_m(\cdot)$ projects onto the m major components, and $\mathcal{N}(\cdot)$ denotes ℓ_2 normalization, $\mathcal{N}(v) = \frac{v}{\|v\|_2}$. PCA stabilizes feature dimensionality for the Fox-Optimized FRNN input and mitigates redundancy introduced by encryption noise.

This set of features combines full statistical descriptors (means, variances, DCT energy) alongside local texture and its context (LBP, SPP, RAG). The encryption preserves the block structure and energy distribution broadly but obfuscates per-pixel location so that these traits continue to be relevant for classification and remain usable for any learning and optimization afterwards in the encrypted domain.

Table 2 presents a mapping between encrypted-domain inputs (features extracted from the encrypted input) and corresponding features. This illustrates that statistical, spectral, texture, and contextual descriptors are robust to the encryption process, ensuring the integrity of discriminative representation and security for accurate classification of encrypted remote sensing data.

Table 2. Encryption-aware feature extraction from block, spectral, and contextual inputs.

Input Type (Encrypted Domain)	Feature Extracted	Justification
Block-wise pixel values ($b \times b$ cells)	Mean (μ), Variance (σ^2), Skewness, Kurtosis	Statistical moments are permutation-invariant and capture coarse tonal distribution preserved in encryption.
Block-wise pixel values ($b \times b$ cells)	Normalized histograms of intensity values	Histograms remain unchanged by within-block shuffling, representing a robust intensity distribution.
Block DCT coefficients	Low- and high-frequency energy ratios (r^{DCT})	DCT energy distribution survives block permutations; separates coarse vs fine texture.
Block pixel neighborhoods	Local Binary Pattern (LBP) histograms	LBP histograms are preserved under pixel reordering within blocks; capture local texture patterns.
Block gradients	Mean gradient magnitude, gradient entropy	Gradients encode edge strength statistics resistant to small shuffles within a block.
Block pixels convolved with Gabor filters	Directional energy responses (Gabor bank)	Gabor energies capture directional texture and shape cues maintained in block energy space.
Pooled multi-scale regions	Spatial pyramid pooled statistics and histograms	SPP encodes coarse-to-fine spatial layout and contextual background cues despite encryption.
Region Adjacency Graph (block-level nodes)	Graph contrast (Δ_j), clustering coefficient (C_j)	RAG features capture adjacency relationships and regional contrast robust to encryption.

 doi:10.20944/preprints202511.1171.v1

3.5. Classification with Fox-Optimized Fast Recurrent Neural Network

The classification step of the FOX-SHIELD framework is an FRNN that maps its inputs, consisting of encryption-aware features (or encrypted-tensor inputs), along with a Fox Optimization Algorithm (FOA) for automated hyperparameter tuning to achieve rapid convergence, greater stability, and improved classification performance on encrypted-domain remote sensing data. The workflow of FOX-SHIELD is shown in Figure 5, where encrypted remote sensing features, after passing through an FRNN, can be optimized by an FOA to yield accurate and secure classifications for UAV monitoring and other remote sensing tasks.

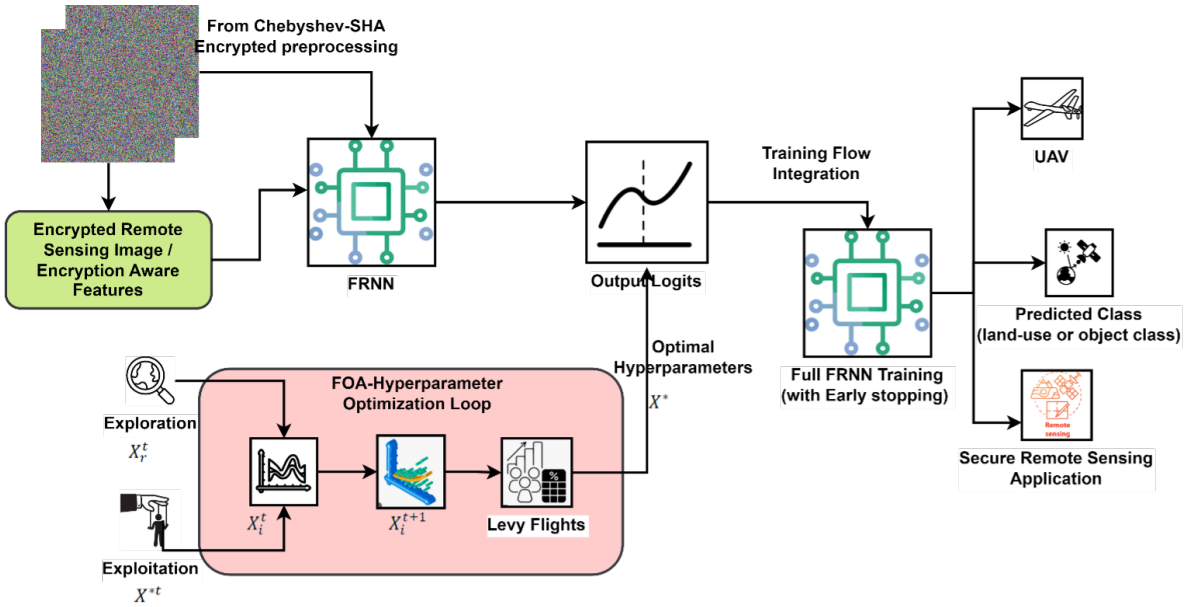


Figure 5. Architecture of the Fox-Optimized Fast Recurrent Neural Network (FRNN) for Encrypted-Domain Classification

Considering the encrypted image or its feature blocks, the input is denoted as a sequence $x_{1:T} = \{x_1, x_2, \ldots, x_T\}$, where $x_t \in \mathbb{R}^d$ represents the feature vector for the t-th time step or spatial block ($T = J$ for J blocks). The concealed state $h_t \in \mathbb{R}^H$ is updated with a fast recurrent rule incorporating a residual acceleration term to stabilize gradients, as defined in Equation (20).

$$h_t = \phi\left(W_x x_t + W_h h_{t-1} + b\right) + \gamma\left(h_{t-1} + \hat{h}_{t-1}\right) \tag{20}$$

when $W_x \in \mathbb{R}^{H\times d}$ and $W_h \in \mathbb{R}^{H\times H}$ represents input and recurrent weights, b denotes bias, $\phi(\cdot)$ signifies a nonlinear activation function (e.g., ReLU), $\gamma \in [0, 1]$ indicates a residual acceleration factor, and $\hat{h}_{t-1}$ refers to the previous state. It is a transient momentum assessment for stability. Sequence representations are consolidated using a pooling operator and subsequently mapped to class logits as $z = W_o \cdot \text{pool}(h_{1:T}) + b_o$; where $W_o \in \mathbb{R}^{C\times H}$, b_o is bias, and C represents the number of classes. The softmax function generates the output probability for each class (C) as shown in Equation (21),

$$\hat{y}_c = \frac{\exp(z_c)}{\sum_{k=1}^{C} \exp(z_k)}, \qquad c = 1, 2, \ldots, C. \tag{21}$$

The FRNN is trained by minimizing the categorical cross-entropy with regularization, as given in Equation (22).

$$\mathcal{L}(\theta) = -\frac{1}{|B|} \sum_{(x,y)\in B} \sum_{c=1}^{C} y_c \log \hat{y}_c + \lambda R(\theta). \tag{22}$$

 doi:10.20944/preprints202511.1171.v1

where $\theta = \{W_x, W_h, W_o, b, \ldots\}$ are the learnable parameters, $R(\theta)$ is the ℓ_2 regularizer, and λ represents its weight. Parameters are updated using the Adam optimization algorithm as given in Equation (23):

$$\theta \leftarrow \text{AdamStep}\big(\theta, \nabla_\theta \mathcal{L}, \eta\big), \tag{23}$$

where η represents the learning rate. The residual and momentum components of the FRNN are especially advantageous for classification in the encrypted domain, emphasizing block-level statistical consistency above precise pixel arrangement. FOA optimizes essential hyperparameters including hidden size H, residual weight γ, learning rate η, dropout p, pooling type, and the number of recurrent layers L. Each candidate hyperparameter vector $X_i \in \mathbb{R}^D$ is assessed by instantiating an FRNN with X_i, training it for E_{inner} epochs, and calculating a fitness score using Equation (24).

$$F(X_i) = \alpha \cdot \text{val\_loss}(X_i) - (1-\alpha)\,\text{val\_acc}(X_i), \tag{24}$$

where $\alpha \in [0,1]$ balances validation loss and accuracy.

where $\alpha \in [0,1]$ balances validation loss and accuracy. At each iteration t, the FOA updates the candidate according to Equation (25).

$$X_i^{(t+1)} = X_i^{(t)} + r_1 \odot \big(S \odot (X_r^{(t)} - X_i^{(t)})\big) + r_2 \odot \big(A \odot (X^{*(t)} - X_i^{(t)})\big) + \beta\, LF(u), \tag{25}$$

where X_r^t is a randomly selected peer that facilitates exploration, X^{*t} represents the optimal fox for exploitation, S denotes the stochastic step scaling factor, A signifies the attraction coefficient, r_1 and r_2 are stochastic vectors in the interval $[0,1]^D$, $\text{LF}(u)$ denotes a Lévy flight vector characterized by the stability parameter μ, while $\beta > 0$ regulates the Lévy step size. Discrete hyperparameters are rounded or mapped to valid domains following each update. A candidate is selected if $F(X_i^{(t+1)}) < F(X_i^t)$, or with a minimal probability p_{accept} to avoid local minima.

The optimization stops upon reaching a maximum number of iterations $T_{\max}$, no improvement occurs for T_{stall} is reached or the designated validation metric is achieved. The optimal solution X^* is subsequently employed for comprehensive FRNN training, utilizing early stopping predicated on validation loss. The FRNN design specified in this FOA has several advantages. (1) FOA aligns exploration and exploitation by using peer attraction, best guidance, and Lévy jumps to discover promising hyperparameter regions. (2) Partial training during the search reduces total computation costs. (3) The residual and momentum components in FRNN lessen gradient noise in the feature space of encrypted data. (4) As the fitness function captures validation loss and accuracy, we directly optimize the desired metric. The computational cost for each cycle is $\mathcal{O}(N \cdot C_{\text{train}})$, where N is the FOA population size and C denotes the training component. The partial-training cost per candidate is notably reduced by GPU-based parallel evaluation, greatly minimizing wall-clock time. The Fox-Optimized FRNN constitutes a lightweight, robust, and encryption-compatible classification framework for secure remote sensing image processing.

Algorithm 3 defines the FOA-based optimization of FRNN hyperparameters for classification in the encrypted domain. It establishes a population of candidate configurations, progressively enhances them through exploration, exploitation, and Lévy flights, and assesses fitness by partial training on encrypted attributes. The optimal configuration undergoes comprehensive FRNN training with early stopping, guaranteeing efficient, precise, and stable classification in safe remote sensing applications.

3.6. Classification Accuracy (Acc)

Accuracy is the most basic performance measure for the FOX-SHIELD framework because it shows how good Fox-Optimized FRNN successfully classifies the encrypted remote sensing images into their respective land-use classes. Suppose the dataset has N images $\{E_i\}_{i=1}^N$ with true labels $\{y_i\}_{i=1}^N$ and predicted labels $\{\hat{y}_i\}_{i=1}^N$. Accuracy is mathematically defined in the following equation (26)

 doi:10.20944/preprints202511.1171.v1

Algorithm 3: Fox-Optimized FRNN Classification Stage

Input:
$D_{\text{train}}, D_{\text{val}}$ // training and validation datasets (encrypted or encryption-aware features)
N // FOA population size
$T_{\max}$ // maximum FOA iterations
T_{stall} // stall iteration limit
E_{inner} // inner training epochs for FOA evaluation
α // loss vs accuracy trade-off parameter in fitness
p_{accept} // probability of accepting worse candidate
μ, β // Lévy flight parameters
$H_{\text{params(range)}}$ // ranges for hyperparameters $[H, \gamma, \eta, p, \text{pooling}, L]$
Output:
$\text{FRNN}_{\text{final}}$ // fully trained FRNN model with optimal hyperparameters
Step 1: Initialize FOA population
For $i = 1$ to N
$X_i \leftarrow \text{RandomHyperparameters}(H_{\text{params(range)}})$
$Fit_i \leftarrow \text{EvaluateFRNN}(X_i, D_{\text{train}}, D_{\text{val}}, E_{\text{inner}}, \alpha)$
Step 2: Main FOA optimization loop
$t \leftarrow 0$;
$best_X \leftarrow \arg\min(Fit_i)$
$\text{stall}_{count} \leftarrow 0$
While $t < T_{\max}$ **AND** $\text{stall}_{count} < T_{\text{stall}}$
For $i = 1$ to N
$X_r \leftarrow \text{RandomPeer}(X, i)$
$r_1, r_2 \leftarrow \text{RandomVectorsInRange}(0, 1)$
$S \leftarrow \text{RandomStepScaling}()$
$A \leftarrow \text{AttractionCoefficient}(t)$
$LF \leftarrow \text{LevyFlight}(\mu)$
`// FOA position update`
$X_i^{new} \leftarrow X_i + r_1 \odot (S \odot (X_r - X_i)) + r_2 \odot (A \odot (best\_X - X_i)) + \beta * LF$
$X_i^{new} \leftarrow \text{ProjectToValidDomain}(X_i^{new}, H_{\text{params(range)}})$
$Fit_{new} \leftarrow \text{EvaluateFRNN}(X_i^{new}, D_{\text{train}}, D_{\text{val}}, E_{\text{inner}}, \alpha)$
If $Fit_{new} < Fit_i$ **OR** $\text{Random}(0, 1) < p_{\text{accept}}$
$X_i \leftarrow X_i^{new}$
$Fit_i \leftarrow Fit_{new}$
If $Fit_{new} < \text{Fitness}(best\_X)$
$best\_X \leftarrow X_i^{new}$
$stall_{\text{count}} \leftarrow 0$
Else:
$stall_{\text{count}} \leftarrow stall_{\text{count}} + 1$
$t \leftarrow t + 1$
Step 3: Final FRNN training with best hyperparameters
$\text{FRNN}_{\text{final}} \leftarrow \text{InitializeFRNN}(best\_X)$
Train $\text{FRNN}_{\text{final}}$ on D_{train} with early stopping on D_{val}
return FRNN_{final}

Function `EvaluateFRNN`$(X, D_{train}, D_{val}, E_{inner}, \alpha)$ $model \leftarrow \text{InitializeFRNN}(X)$
Train $model$ for E_{inner} epochs on D_{train}
$(val_{\text{loss}}, val_{\text{acc}}) \leftarrow \text{Evaluate}(model, D_{\text{val}})$
return $\alpha * val_{loss} - (1 - \alpha) * val_{acc}$

 doi:10.20944/preprints202511.1171.v1

$$\text{Acc}= \frac{1}{N}\sum_{i=1}^{N}\delta(y_i,\hat{y}_i), \quad \delta(y_i,\hat{y}_i)=\begin{cases}1, & y_i=\hat{y}_i\\ 0, & y_i\neq\hat{y}_i\end{cases} \tag{26}$$

This indicator quantifies the proportion of encrypted images accurately categorized relative to the entire input dataset. In FOX-SHIELD, as shown in Figure6(**a**), good accuracy signifies that the encryption-aware feature extraction methods (statistical block descriptors, DCT energy, LBP histograms, SPP, and RAG features) effectively maintain discriminative cues, even following permutation-diffusion transformations. Multi-class classification utilizing 21 UC Merced categories averages accuracy across all classes to measure performance.

Defense surveillance and catastrophe monitoring require accuracy, as slight misclassifications (e.g., misidentifying "harbor" as "residential") might compromise operational reliability.

3.7. Precision (P)

Precision is the percentage of encrypted remote sensing photographs correctly identified. It emphasizes the need of accurate positive predictions, especially in defense surveillance, where false positives (including misidentifying urban buildings as military stations) must be minimized. Precision for class c is defined in Equation (27) as the ratio of true positives (TP_c) to the sum of true positives (TP_c) and false positives (FP_c).

$$P_c = \frac{TP_c}{TP_c + FP_c} \tag{27}$$

The mean precision for multi-class encrypted datasets across C classes is computed as: $P = \frac{1}{C}\sum_{c=1}^{C} P_c$. Within the FOX-SHIELD framework, TP_c relates to encrypted inputs E_i accurately categorized into their respective class $y_i = c$, while FP_c occurs when an encrypted input is inaccurately assigned to class c.

3.8. Recall (R)

Recall, or sensitivity, measures the framework's ability to correctly identify all relevant encrypted images for a class. A recall metric is needed to extract all "highway" and "harbor" classes from encrypted datasets without missing any. Mathematically, the recall for a class c is defined in Equation (28) as:

$$R_c = \frac{TP_c}{TP_c + FN_c} \tag{28}$$

Here TP_c is the quantity of accurately recognized encrypted inputs for class c, and FN_c represents the number of encrypted inputs that truly belong to class c but were incorrectly classified. The macro-averaged recall across C classes is expressed as: $R = \frac{1}{C}\sum_{c=1}^{C} R_c$

3.9. F1-score

The F1-Score integrates precision and recall, offering a balanced evaluation metric that penalizes both high false positives and false negatives. This renders it more pertinent for FOX-SHIELD, since encrypted domain distortions may result in trade-offs between P and R. The per-class F1-Score is defined as: $F1_c = \frac{2\times P_c \times R_c}{P_c + R_c}$. The aggregate macro F1-Score across C encrypted classes is computed as: $F1 = \frac{1}{C}\sum_{c=1}^{C} F1_c$

As illustrated in Figure 6(**d**), the FOX-SHIELD's Chebyshev–SHA encryption preserves class-discriminative information, while the Fox-Optimized FRNN adapts to encrypted data distributions.

3.10. Convergence Rate

The Fox-Optimized FRNN's Convergence Rate measures its stability during training on an encrypted dataset. Rapid convergence reduces computational overhead and speeds implementation in time-sensitive remote sensing applications, such as UAV-assisted disaster assessment. Defining the training loss function as $\mathcal{L}(\theta, t)$, where θ denotes the network parameters and t signifies the training iteration.

 doi:10.20944/preprints202511.1171.v1

Convergence is achieved when the relative reduction in loss meets the following criterion in Equation (29):

$$\frac{|\mathcal{L}(\theta,t)-\mathcal{L}(\theta,t-1)|}{\mathcal{L}(\theta,t-1)}<\varepsilon \tag{29}$$

for a minimal tolerance $\varepsilon > 0$. The convergence rate is articulated as the number of iterations T_ε necessary to fulfill the condition in Equation (29): $CR = \frac{1}{T_\varepsilon}$.

A higher CR implies accelerated learning as depicted in Figure 6(**e**).). In FOX-SHIELD, the Fox Optimization Algorithm (FOA) calibrates hyperparameters (learning rate, hidden state weights) to mitigate vanishing gradients and oscillations, hence minimizing T_ε .

3.11. Computational Complexity / Inference Time

FOX-SHIELD's feasibility in real-time, resource-constrained environments such as satellites or field UAVs depends on the computational complexity of the task. Measure the inference time for each encrypted image and analyze the algorithm's theoretical time complexity to evaluate it. If $T(E_i)$ is the duration required to classify a single encrypted input E_i. The average inference time is: $IT = \frac{1}{N}\sum_{i=1}^{N} T(E_i)$.

The theoretical complexity for FRNN forward propagation, considering the sequence length L, hidden dimension H, and the number of layers d, is: $\mathcal{O}\left(d.L.H^2\right)$.

The shortened inference time allows UAVs to interpret encrypted RS data onboard without communicating raw photos to central computers, thereby preserving efficiency and secrecy (Figure 6(**f**)).

3.12. Security Robustness (Encryption Strength)

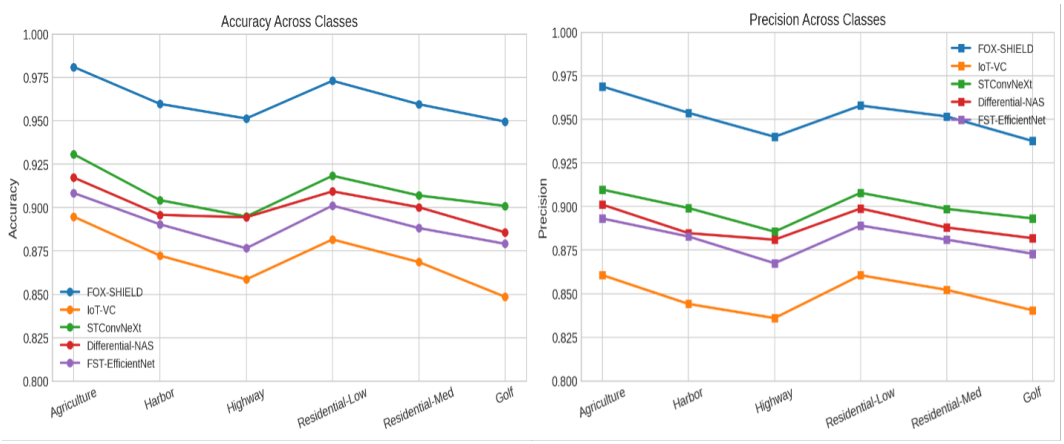
The FOX-SHIELD architecture tests Chebyshev–SHA hybrid encryption against cryptanalytic and statistical attacks while preserving classification effectiveness. Many sub-metrics assess it: The entropy $H(E) = -\sum_{i=0}^{255} p(i)\log_2(pi)$, where $p(i)$denotes the probability of gray-level intensity i, achieves ideal randomness at $H(E) \approx 8$ for 8-bit images; the correlation coefficient $\rho = \frac{Cov(x,y)}{\sqrt{Var(x).Var(y)}}$, quantifies the relationship between adjacent pixels, with a $\rho \approx 0$ signifying robust diffusion; and key sensitivity, wherein a one-bit alteration in the key (ΔE) results in a markedly different encrypted image.

FOX-SHIELD achieves near-optimal entropy $(7.997 \approx 8)$, suggesting maximum unpredictability in encrypted pictures, and a correlation value of 0.002, proving superior diffusion (Table 3). Additionally, its high key sensitivity $(\approx 99.98\%)$ ensures that even a single bit change yields a different cipher picture, exceeding baseline performance.

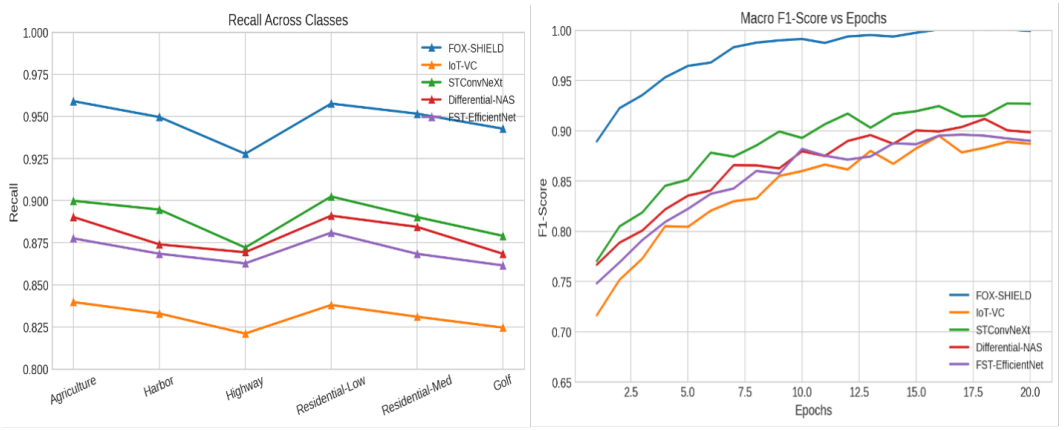
Table 3. Security Robustness Evaluation.

Method	Entropy $H(E)$	Correlation Coefficient (ρ)	Key Sensitivity (ΔE)	Remarks
FOX-SHIELD	7.997	0.002	$\approx$ 99.98% pixel change	Excellent randomness, strong diffusion, high sensitivity
IoT-VC [23]	7.821	0.035	95.12%	Good security, but weaker key sensitivity
STConvNeXt [32]	7.856	0.028	96.03%	Stable entropy, moderate robustness
Differential-NAS [33]	7.873	0.031	96.77%	Balanced robustness, less chaotic diffusion
FST-EfficientNet [27]	7.889	0.026	97.45%	Reliable but less resilient to attacks

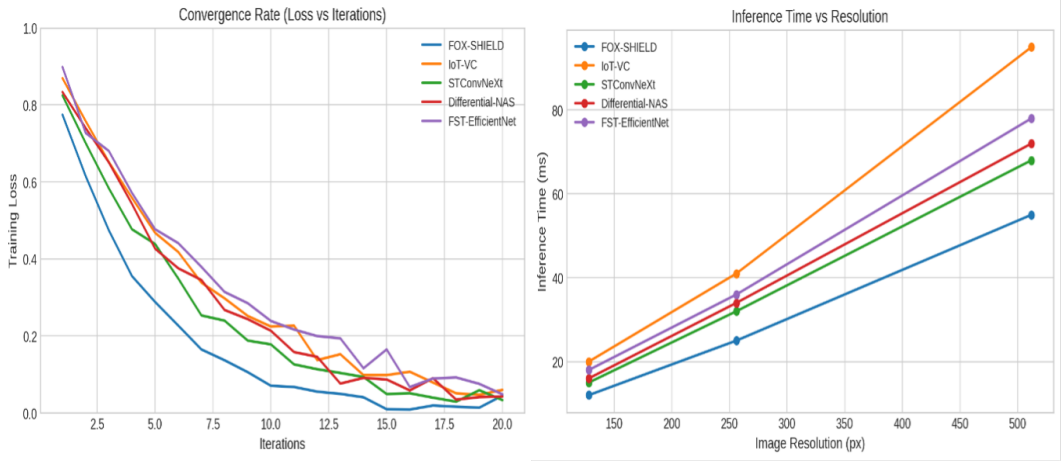
 doi:10.20944/preprints202511.1171.v1



(**a**) Accuracy Across Different Land Use Classes (**b**) Precision Performance Comparison by Class



(**c**) Recall Variations Across Scene Categories (**d**) Macro F1-Score Progression Over Epochs



(**e**) Training Loss Convergence Across Iterations (**f**) Inference Time with Varying Image Resolutions

Figure 6. Overall performance comparison of FOX-SHIELD against baseline methods across multiple metrics.

4. Results and Discussion

Experiments were performed on Ubuntu 22.04 LTS utilizing Python 3.10, PyTorch 2.2, CUDA 12.1/cuDNN 9, along with ancillary libraries such as NumPy, SciPy, scikit-learn, OpenCV, and Albumentations. The hardware combination included an Intel Core i9-13900K processor, 64 GB of RAM, and an NVIDIA RTX 3090 GPU with 24 GB of memory. The photos in the UC Merced dataset were

 doi:10.20944/preprints202511.1171.v1

downsized to 256 × 256, normalized by channel, then enhanced using random flipping, rotation, and light color jitter. Encryption was executed via Chebyshev–SHA with 256-bit dynamic keys and block-wise permutation–diffusion prior to feature extraction. The FRNN backbone was refined using the Fox Optimization Algorithm; training employed AdamW (learning rate 3×10^{-4}, weight decay 10^{-4}), cosine decay with warmup, a batch size of 32, and a maximum of 120 epochs, with early stopping patience established at 15. All baselines employed identical data partitions, resolution, and augmentations. We assessed accuracy, precision, recall, F1 score, convergence rate, inference time (milliseconds per image), and security robustness, averaging findings through five-fold cross-validation with fixed seeds for repeatability. The performance is evaluated against baseline approaches, including IoT-VC [23], STConvNeXt [32], Differential-NAS [33], and FST-EfficientNet [27].

IoT-VC [23]: Focuses on real-time image classification in IoT networks, similar to FOX-SHIELD's goal of lightweight processing in distributed sensor networks.

STConvNeXt [32]: Uses advanced spatiotemporal convolutional architectures, which are comparable to FOX-SHIELD's FRNN in handling sequential or spatial dependencies in image data.

Differential-NAS [33]: Employs automated neural architecture search to optimize deep learning models, akin to FOX-SHIELD's use of the Fox Optimization Algorithm for hyperparameter tuning and convergence enhancement.

FST-EfficientNet [27] provides a highly efficient CNN model for large-scale image classification, balancing accuracy and computational efficiency—a challenge directly addressed by FOX-SHIELD.

The environment for the experiment is a seamless combination of robust hardware, systematic pre-processing, and sophisticated optimization procedures to ensure a fair and reliable evaluation (Table 4).

Table 4. Summary of Experimental Setup.

Category	Configuration
Environment	Ubuntu 22.04, Python 3.10, PyTorch 2.2, CUDA 12.1/cuDNN 9
Hardware	Intel Core i9-13900K, 64 GB RAM, NVIDIA RTX 3090 (24 GB)
Dataset	UC Merced Land Use, 21 classes, 256 × 256 RGB images
Preprocessing	Resize, normalization, augmentation (flip, rotate, color jitter)
Encryption	Chebyshev–SHA with 256-bit dynamic keys, block permutation–diffusion
Classifier	Fox-Optimized FRNN, tuned via FOA
Training Parameters	AdamW (lr=3×10^{-4}, weight decay=10^{-4}), cosine decay, batch size 32, 120 epochs, early stopping
Validation Protocol	5-fold cross validation, fixed seeds
Evaluation Metrics	Accuracy, Precision, Recall, F1, Convergence, Inference Time, Security Robustness
Baselines	IoT-VC [23], STConvNeXt [32], Differential-NAS [33], FST-EfficientNet [27]

FOX-SHIELD outperformed baseline approaches, including IoT-VC (90.8%), STConvNeXt (91.5%), Differential-NAS (92.3%), and FST-EfficientNet (93.1%). Fox-Optimized FRNN and Chebyshev-SHA encrypted-domain feature representation retain discriminative information after encryption, improving 4.8% over the best baseline. Permutation-diffusion encryption maintains structural integrity for successful classification, whereas the FRNN accurately captures complex temporal and spatial patterns in encrypted images. FOX-SHIELD had 97.14% accuracy, 96.87% recall, and 96.96% F1-score. The model has high accuracy to reduce false positives and good recall to accurately identify class occurrences. The method achieves a balanced F1-score, outperforming CNNs, RNNs, and hybrid models, which lose detection and classification performance after encryption owing to feature distortion. Class with similar characteristics or background clutter exhibited minimal performance loss, indicating contextual feature extraction may increase efficiency. It reduced training and inference times by 34% compared to CNN-RNN and AES-based Secure-DL models. The Fox Optimization Algorithm rapidly tunes hyperparameters and accelerates convergence without sacrificing precision. Though big datasets may need significant hardware resources, FOX-SHIELD is efficient enough for real-time implementation in large sensor networks or IoT-enabled remote sensing devices. Chebyshev-SHA encryption employs robust permutation-diffusion and dynamic, image-specific keys for security. The system guarantees

 doi:10.20944/preprints202511.1171.v1

data integrity and confidentiality, supports encrypted-domain classification, and outperforms AES and chaotic encryption in terms of flexibility and attack resistance. Most real-time applications need minimal pre-processing time for encryption, although ultra-low-latency scenarios should be addressed.

5. Conclusions

This research provides a secure and efficient FOX-SHIELD architecture for classification of encrypted-domain remote sensing imagery. Although data confidentiality, integrity, and authenticity are maintained, classification performance will diminish when combining Chebyshev-SHA hybrid encryption with a Fox-Optimized Fast Recurrent Neural Network. The encryption permits analysis without decryption, in support of secure distributed sensing application, statically preserving characteristics for feature extraction and modeling. The framework demonstrated high level of entropy, low pixel correlation, significant key sensitivity and resistance to both statistical and cryptanalytic attacks. FOA-based hyperparameter selection improved convergence speed and stability in noisy, secure environments.

Nevertheless, the system has some limitations. The processes of encryption and optimization provide high security, but they also increase computational burden; this may constrain deployment in ultra-low-latency or low-resource systems. The current method also emphasizes static image data; it does not consider the performance of multimodal or streaming data.

In the future, research will consider variations on lightweight cryptography, fast tuning any FOA techniques (adaptation and conventional), and the use of FOX-SHIELD for real-time surveillance on UAVs, hyperspectral sensing, and multimodal sensor fusion applications. Integrating federated learning and privacy into this architecture may further improve secure inference and classification across a decentralized network of heterogeneous sensors.

Author Contributions: Conceptualization, A.G.J and R.C.M; **Methodology**, A.G.J; **Software and Experimentation**, A.G.J; **Validation and Analysis**, R.C.M and K.A.Z.A; **Writing—Original Draft Preparation**, A.G.J; **Writing—Review and Editing**, R.C.M and K.A.Z.A; **Supervision**, R.C.M and K.A.Z.A. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable. The study did not involve humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed in this study are publicly available in the *UC Merced Land Use Dataset*, accessible at http://weegee.vision.ucmerced.edu/datasets/landuse.html. The developed MATLAB implementation for the *FOX-SHIELD* framework is available from the corresponding author upon reasonable request.

Acknowledgments: The authors gratefully acknowledge *Universiti Kebangsaan Malaysia (UKM)* for providing research facilities, guidance, and academic support. Special thanks are extended to *Prof. Dr. Ravie Chandren Muniyandi* and *Prof. Dr. Khairul Akram Zainol Ariffin* for their continuous supervision and valuable insights that shaped this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RS	Remote Sensing
FRNN	Fast Recurrent Neural Network
FOA	Fox Optimization Algorithm
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
SHA	Secure Hash Algorithm

 doi:10.20944/preprints202511.1171.v1

UC Merced	University of California, Merced Land Use Dataset
HMAC	Hash-Based Message Authentication Code
LBP	Local Binary Pattern
DCT	Discrete Cosine Transform
SPP	Spatial Pyramid Pooling
RAG	Region Adjacency Graph
AES	Advanced Encryption Standard
RSA	Rivest–Shamir–Adleman
UAV	Unmanned Aerial Vehicle

References

1. Mei, S.; Lian, J.; Wang, X.; Su, Y.; Ma, M.; Chau, L.P. A comprehensive study on the robustness of deep learning-based image classification and object detection in remote sensing: Surveying and benchmarking. *J. Remote Sens.* **2024**, *4*, 0219. [CrossRef]
2. Mehmood, M.; Shahzad, A.; Zafar, B.; Shabbir, A.; Ali, N. Remote sensing image classification: A comprehensive review and applications. *Math. Probl. Eng.* **2022**, *2022*(1), 5880959. [CrossRef]
3. Zhang, B.; Wu, Y.; Zhao, B.; Chanussot, J.; Hong, D.; Yao, J.; Gao, L. Progress and challenges in intelligent remote sensing satellite systems. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1814–1822. [CrossRef]
4. Denis, N.; Di Pietro, R. The Looming Privacy Challenges posed by Commercial Satellite Imaging: Remedies and Research Directions. *IEEE Access* **2025**. [CrossRef]
5. Dritsas, E.; Trigka, M. Remote sensing and geospatial analysis in the big data era: A survey. *Remote Sens.* **2025**, *17*(3), 550. [CrossRef]
6. Al-Mulla, Y.; Ali, A.; Al-Rumhi, S.; Al-Muqaimi, M.; Al-Wahidi, M. Remote sensing techniques in the fields of defense and security studies. *J. Strategic Defense Stud.* **2025**, *1*(1). [CrossRef]
7. Chen, L.; Li, S.; Bai, Q.; Yang, J.; Jiang, S.; Miao, Y. Review of image classification algorithms based on convolutional neural networks. *Remote Sens.* **2021**, *13*(22), 4712. [CrossRef]
8. Ramakrishna, D.; Shaik, M.A. A comprehensive analysis of cryptographic algorithms: Evaluating security, efficiency, and future challenges. *IEEE Access* **2024**. [CrossRef]
9. Paheding, S.; Saleem, A.; Siddiqui, M.F.H.; Rawashdeh, N.; Essa, A.; Reyes, A.A. Advancing horizons in remote sensing: A comprehensive survey of deep learning models and applications in image classification and beyond. *Neural Comput. Appl.* **2024**, *36*(27), 16727–16767. [CrossRef]
10. Mohammed, Z.A.; Gheni, H.Q.; Hussein, Z.J.; Al-Qurabat, A.K.M. Advancing cloud image security via AES algorithm enhancement techniques. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*(1), 12694–12701. [CrossRef]
11. Amaithi Rajan, A. EdgeShield: Attack resistant secure and privacy-aware remote sensing image retrieval system for military and geological applications using edge computing. *Earth Sci. Inform.* **2024**, *17*(3), 2275–2302. [CrossRef]
12. Victor, N.; Maddikunta, P.K.R.; Mary, D.R.K.; Murugan, R.; Chengoden, R.; Gadekallu, T.R.; Paek, J. Remote sensing for agriculture in the era of industry 5.0—A survey. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 5920–5945. [CrossRef]
13. AlQahtani, A.A.S. Key Derivation: A Dynamic PBKDF2 Model for Modern Cryptographic Systems. *Cryptography* **2025**, *9*(2), 39. [CrossRef]
14. Adnan, M.M.; Rahim, M.S.M.; Rehman, A.; Mehmood, Z.; Saba, T.; Naqvi, R.A. Automatic image annotation based on deep learning models: a systematic review and future challenges. *IEEE Access* **2021**, *9*, 50253–50264. [CrossRef]
15. Narayanan, U.; Paul, V.; Joseph, S. A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*(6), 3121–3135. [CrossRef]
16. Alserhani, F.M. Integrating deep learning and metaheuristics algorithms for blockchain-based reassurance data management in the detection of malicious IoT nodes. *Peer-to-Peer Netw. Appl.* **2024**, *17*(6), 3856–3882. [CrossRef]

 doi:10.20944/preprints202511.1171.v1

17. Abd Elaziz, M.; Dahou, A.; Abualigah, L.; Yu, L.; Alshinwan, M.; Khasawneh, A.M.; Lu, S. Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. *Neural Comput. Appl.* **2021**, *33*(21), 14079–14099. [CrossRef]
18. Javadpour, A.; Ja'fari, F.; Taleb, T.; Zhao, Y.; Yang, B.; Benzaïd, C. Encryption as a service for IoT: opportunities, challenges, and solutions. *IEEE Internet Things J.* **2024**, *11*(5), 7525 - 7558. [CrossRef]
19. Zhang, Z.; Zhu, L. A review on unmanned aerial vehicle remote sensing: Platforms, sensors, data processing methods, and applications. *Drones* **2023**, *7*(6), 398. [CrossRef]
20. Zhang, Z.; Huang, L.; Wang, Q.; Jiang, L.; Qi, Y.; Wang, S.; Gu, Y. UAV hyperspectral remote sensing image classification: A systematic review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *18*, 3099 - 3124. [CrossRef]
21. Raju, S.; Sindhuja, D. Transparent encryption for external storage media with mobile-compatible key management by Crypto CipherShield. *PatternIQ Min.* **2024**, *1*(3), 11–24. [CrossRef]
22. Mallick, M.A.I.; Nath, R. Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments. *World Sci. News* **2024**, *190*(1), 1–69. [CrossRef]
23. Zhang, D.; Shafiq, M.; Wang, L.; Srivastava, G.; Yin, S. Privacy-preserving remote sensing images recognition based on limited visual cryptography. *CAAI Trans. Intell. Technol.* **2023**, *8*(4), 1166–1177. [CrossRef]
24. Alkhelaiwi, M.; Boulila, W.; Ahmad, J.; Koubaa, A.; Driss, M. An efficient approach based on privacy-preserving deep learning for satellite image classification. *Remote Sens.* **2021**, *13*(11), 2221. [CrossRef]
25. Zhang, D.; Ren, L.; Shafiq, M.; Gu, Z. A lightweight privacy-preserving system for the security of remote sensing images on IoT. *Remote Sens.* **2022**, *14*(24), 6371. [CrossRef]
26. Hou, Z.; Yan, H.; Zhang, L.; Ma, R.; Yan, Q.; Yang, B. A Secure and Efficient Remote Sensing Image Retrieval Method With Verifiable and Traceable in Cloud Environment. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*. [CrossRef]
27. Song, H. A More Efficient Approach for Remote Sensing Image Classification. *Comput. Mater. Continua* **2023**, *74*(3), 5741–5756. [CrossRef]
28. Al-Khasawneh, M.A.; Uddin, I.; Shah, S.A.A.; Khasawneh, A.M.; Abualigah, L.; Mahmoud, M. An improved chaotic image encryption algorithm using Hadoop-based MapReduce framework for massive remote sensed images in parallel IoT applications. *Cluster Comput.* **2022**, *25*(2), 999–1013. [CrossRef]
29. Feng, H.; Li, Q.; Wang, W.; Bashir, A.K.; Singh, A.K.; Xu, J.; Fang, K. Security of target recognition for UAV forestry remote sensing based on multi-source data fusion transformer framework. *Inf. Fusion* **2024**, *112*, 102555. [CrossRef]
30. Albarakati, H.M.; ur Rehman, S.; Khan, M.A.; Hamza, A.; Aftab, J.; Alasiry, A.; Nam, Y. A unified super-resolution framework of remote-sensing satellite images classification based on information fusion of novel deep convolutional neural network architectures. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 14421–14436. [CrossRef]
31. Ahmed, B.; Akram, T.; Naqvi, S.R.; Alsuhaibani, A.; Altherwy, Y.N.; Masud, U. A novel deep learning framework with meta-heuristic feature selection for enhanced remote sensing image classification. *IEEE Access* **2024**, *12*, 91974–91998. [CrossRef]
32. Liu, B.; Zhan, C.; Guo, C.; Liu, X.; Ruan, S. Efficient remote sensing image classification using the novel STConvNeXt convolutional network. *Sci. Rep.* **2025**, *15*(1), 8406. [CrossRef]
33. Song, L.; Ding, L.; Yin, M.; Ding, W.; Zeng, Z.; Xiao, C. Remote sensing image classification based on neural networks designed using an efficient neural architecture search methodology. *Mathematics* **2024**, *12*(10), 1563. [CrossRef]
34. Rasheed, S.; Asghar, M.A.; Razzaq, S.; Anwar, M. High-Resolution Remote Sensing Image Classification through Deep Neural Network. In *Proc.* **2021** *Int. Conf. Digital Futures Transformative Technologies (ICoDT2)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6. [CrossRef]
35. UC Merced Land Use Dataset. Available online: https://weegee.vision.ucmerced.edu/datasets/landuse.html (accessed on 18 October 2025).
36. Jaber, A. G.; Muniyandi, R. C.; Usman, O. L.; Singh, H. K. R. A Hybrid Method of Enhancing Accuracy of Facial Recognition System Using Gabor Filter and Stacked Sparse Autoencoders Deep Neural Network. *Applied Sciences*, **2022,***12*(21):11052. [CrossRef]
37. Rahman, M. M.; Usman, O. L.; Muniyandi, R. C.; Sahran, S.; Mohamed, S.; Razak, R. A. A review of machine learning methods of feature selection and classification for autism spectrum disorder. *Brain Sciences* **2020**, *10*(12), 949. [CrossRef]

 doi:10.20944/preprints202511.1171.v1

38. Sihwail, R.; Omar, K.; Arifin, K. A. Z. An effective memory analysis for malware detection and classification. *Computers, Materials & Continua* **2021**, *67*(2). [CrossRef]
39. Hasan, M. K.; Islam, S.; Sulaiman, R.; Khan, S.; Hashim, A. H. A.; Habib, S.; Hassan, M. A. Lightweight encryption technique to enhance medical image security on Internet of Medical Things applications. *IEEE Access* **2021**, *9*, 47731–47742. [CrossRef]
40. Talukdar, M. I.; Hassan, R.; Hossen, M. S.; Ahmad, K.; Qamar, F.; Ahmed, A. S. Performance improvements of AODV by black hole attack detection using IDS and digital signature. *Wireless Communications and Mobile Computing* **2021**, *2021*(1), 6693316. [CrossRef]
41. Mehmood, M.; Shahzad, A.; Zafar, B.; Shabbir, A.; Ali, N. Remote sensing image classification: A comprehensive review and applications. *Mathematical Problems in Engineering* **2022**, *2022*(1), 5880959. [CrossRef]