Design of Hybrid Deep Learning-Based Environmental Monitoring Using Feature Fusion Techniques on Remote Sensing Satellite Imagery

Hadi Oqaibi
Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
hoqaibi@kau.edu.sa (corresponding author)

Received: 31 October 2025 | Revised: 30 November 2025 and 19 December 2025 | Accepted: 21 December 2025

 DOI: https://doi.org/10.48084/etasr.15915

ABSTRACT

The natural environment is crucial for human development and survival, as it offers climate, land, water, biological resources, and other essential ecosystem services. Therefore, environmental monitoring is of great importance for effective resource management, environmental protection, and human survival and development. Conventional monitoring techniques are labor-intensive, costly, and time-consuming. Owing to advances in remote sensing sensor capabilities, large-scale dynamic observation can quickly capture wide spectral information at both local and global scales. In recent years, satellite remote sensing has increasingly informed the state of and pressures on biodiversity at various spatial scales. Modern Deep Learning (DL) methods, evolved from classical neural networks, have outperformed conventional machine learning models, achieving significant improvements in environmental monitoring performance. This paper presents a Hybrid Deep Learning-based Environmental Monitoring using Feature Fusion Techniques on Satellite Remote Sensing Images (HDLEM-FFTSRSI) model. Initially, the image preprocessing stage involves several steps, including resizing, normalization, dataset division, and augmentation, to improve image quality. Furthermore, the HDLEM-FFTSRSI model implements a fusion of EfficientNetV2-S, Vision Transformer (ViT), and ConvNeXt architectures for feature extraction. Finally, a hybrid of Long Short-Term Memory (LSTM) and Autoencoder (AE) models (LSTM-AE) is employed for classification. The comparative analysis demonstrates that the HDLEM-FFTSRSI method achieves a superior accuracy of 99.11% compared with other models on the EuroSat dataset.

Keywords-environmental monitoring; feature fusion; deep learning; satellite remote sensing images; computer vision

I. INTRODUCTION

The natural environment is a broad term encompassing land, climate, water, and biological resources, influencing human survival and advancement. It is intimately connected to the sustainable progress of society as well as the economy [1]. The extent of human usage of natural resources has steadily grown alongside rapid population expansion, particularly resulting in excessive exploitation, and worsening environmental conditions [2]. Generally applied ground-enabled monitoring is constrained by area and is only appropriate for point-based environmental monitoring within a limited zone [3]. This technique is labor-intensive, expensive, and time-consuming. Owing to the characteristics of wide-area and real-time observation, remote sensing sensors can swiftly gather broad-spectrum data of regional or even worldwide targets, allowing various ecological metrics to be extracted through data interpretation and retrieval [4]. Thus, remote sensing has gradually become a vital technique for ecological monitoring, specifically on a large or international level [5]. In contrast to conventional, expensive, and labor-heavy manual sampling techniques for point-based measurements in environmental monitoring, remotely sensed tools offer clear benefits due to their comprehensive reach and high spatial and temporal detail [6]. Depending on the distinct absorption features of materials, tools are built to collect information across multiple electromagnetic bands, covering Ultraviolet (UV), near-infrared, visible, thermal infrared, shortwave infrared, and microwave regions. Due to physical and engineering limits, compromises occur in the creation of remotely sensed tools for spatial, spectral, and radiometric resolution [7]. Satellite remote sensing technologies are broadly implemented in atmospheric environmental monitoring globally. It efficiently detects a continuous and dynamic state of air over a large area [8]. The advancement of remote sensing technologies has resulted in enhanced spatial resolution of images captured through remote sensing. Therefore, the

understanding of the fundamental units in an image becomes effective when image uncertainty is reduced. As a result, various object combinations in the environment have become crucial fundamental units for interpreting remotely sensed images.

Authors in [9] proposed a Multi-Branch Generative Prior (GP) Integration Network (MBGPIN) for addressing environmental monitoring. This model merges hybrid Attention Mechanisms (AMs), multiscale feature extraction, and GP learning based on pre-trained Vector-Quantized Generative Adversarial Network (VQGAN) methods. The MBGPIN dual-pathway framework comprises a GP pathway for external guidance and a spatial feature extraction pathway, which are merged through an Adaptive GP Fusion (AGPF) technique. Authors in [10] proposed a Vision Transformer Improved Super-Resolution Generative Adversarial Network (ViT-ISRGAN) approach. Authors in [11] proposed a Dual-Level Network (DLNet), combining cross- and self-AMs for enhanced multiscale feature fusion and extraction. Authors in [12] introduced the Spatial–Spectral Context-Boosted Network (SSCBNet), a groundbreaking network.

Authors in [13] presented a Multiscale Feature Interaction Network (MFINet) method, which incorporates a Bitemporal Feature Interaction Layer (BFIL). In addition, a Bitemporal Feature Fusion Layer (BFFL) is employed in the final encoding result. Authors in [14] developed a Global-Aware Siamese Network (GAS-Net) approach, focusing on generating global-aware features for effective change recognition by modeling relationships between foregrounds and backgrounds. This approach comprises the Foreground-Awareness Module (FAM) and Global-Attention Module (GAM), which learn context relationships and improve relational learning. Authors in [15] presented an innovative lightweight object detection technique that relies on YOLOv5s to enhance detection performance. Authors in [16] suggested an improved Faster Region-based Convolutional Neural Network (Faster R-CNN) approach that combines three key alterations to improve oil well detection accuracy. Although the presented studies are effective, they face limitations in fully capturing complementary features from multi-model architectures and in efficiently modeling spatiotemporal dependencies, revealing a clear research gap that motivates the development of a unified hybrid framework for improved environmental monitoring.

This paper presents a novel Hybrid Deep Learning-based Environmental Monitoring using Feature Fusion Techniques on Satellite Remote Sensing Images (HDLEM-FFTSRSI) model. The key contributions are listed below:

- The preprocessing stage enhances image quality and enables improved feature learning across heterogeneous data, thereby strengthening the overall performance and reliability of the environmental monitoring workflow.
- The fusion of EfficientNetV2-S, Vision Transformer (ViT), and ConvNeXt models captures complementary spatial–spectral features, improving the richness of learned representations and model robustness while also addressing the limitations of relying on a single feature extractor.
- The integration of temporal dependency modeling with latent-space reconstruction is effectively performed using the hybrid Long Short-Term Memory (LSTM) and Autoencoder (AE) (LSTM-AE) module to strengthen class discrimination, robustness against intrinsic environmental variability, and overall classification reliability.
- The novelty of the HDLEM-FFTSRSI model lies in the integration of preprocessing, EfficientNetV2-S, ViT, and ConvNeXt feature extractors with an LSTM-AE classifier, thus enhancing feature representation and classification accuracy within a unified and robust environmental monitoring framework using satellite imagery.

II. MATERIALS AND METHODS

This paper presents the HDLEM-FFTSRSI model for environmental monitoring using satellite remote sensing images, comprising three phases: preprocessing, feature extraction, and classification. Figure 1 indicates the flow of the HDLEM-FFTSRSI method.

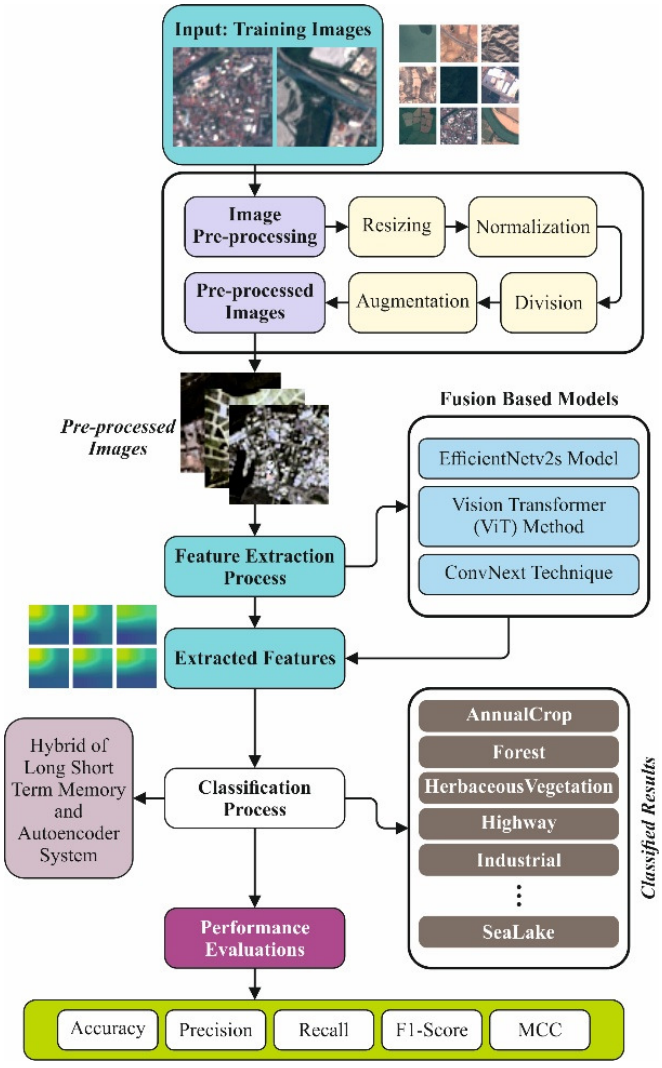

Fig. 1. Overall workflow of the HDLEM-FFTSRSI model.

A. Stage I: Image Preprocessing

Primarily, the image preprocessing phase comprises various steps, such as resizing, normalization, dataset division, and augmentation, to enhance image quality [17]. The preprocessing stage is tailored to improve the ability of the network to adjust independently more efficiently. The stage contains the three techniques discussed below.

1) Image Resizing

The initial phase includes resizing each image in the dataset to uniform dimensions. Processing images at a uniform and smaller size accelerates the training procedure and decreases computation time in comparison with handling images of large and diverse dimensions. The resizing can be achieved utilizing the resizing function from the OpenCV (cv2) library, guaranteeing that every image is resized to identical dimensions for consistency.

2) Scaling

After resizing, all images are passed through a function, mask_split_threshold, which scales the pixel values of the image to a range of (0,1). It guarantees that the input data are normalized, making it simpler for the model to learn patterns without being affected by variations in the raw value range. This process accelerates the model's convergence during training and enhances overall performance.

3) Image Contrast Enhancement

This step is applied to adjust the target image such that its features are more easily perceived by automated image analysis methods. It allows optimization of saturation, brightness, sharpness, and contrast. Such a module is called an image enhancement module, which offers appropriate techniques for the essential procedure of image adjustment.

4) Splitting

Supervised machine learning determines mapping relationships between the input (independent variable) and the output (dependent variable) in order to forecast unseen outputs. The predictive performance of the model is evaluated using unseen data that were not included during training. Data splitting is performed to assess whether the model suffers from overfitting.

5) Augmentation

Deep learning methods largely depend on the availability of a large quantity of data. Data augmentation is an effective solution to address the relative insufficiency of data compared with the large number of parameters in neural network models. It involves generating new samples while preserving the characteristics of the original data. Common augmentation techniques include geometric transformations, such as rotation, scaling, shifting, and mirroring, as well as photometric transformations. In this work, flipping, rescaling, and rotation are applied to increase dataset size.

B. Stage II: Fusion of Feature Extractors

Subsequently, the HDLEM-FFTSRSI model implements a fusion of EfficientNetV2-S, ViT, and ConvNeXt methods for feature extraction. These methods are chosen for their capability to effectively integrate complementary features, capturing both global and local representations, thus improving feature richness and generalization compared with single-model approaches.

1) EfficientNetV2-S Method

The EfficientNetV2-S method effectively converges with a stable training procedure and dynamic regularization adaptation [18]. It supports advanced learning and deals with memory consumption by resizing input images to 384 × 384 pixels. The adjustment of input image resolution enables improved precision, enhanced feature learning, and reduced overfitting. This architecture has proven proficient in addressing challenges in image analysis, with applications including automatic detection. Compared with EfficientNetV1, this framework demonstrates superior parameter efficiency and Floating-Point Operations (FLOPs) performance. FLOPs, which measure model complexity, show that EfficientNetV2 outperforms EfficientNetV1 due to improved computational efficiency. EfficientNetV1, which relies on larger image sizes, faces high memory consumption, requiring small batch sizes and consequently slowing training speed. To address this limitation, the traditional flattened layer was replaced with Global Average Pooling (GAP) to reduce parameters and speed up calculations. In addition, a dropout layer with a rate of 0.2 was presented to improve model robustness and generalization. Generally, these changes contribute to increased efficiency and classification accuracy in the fine-tuned model.

2) Vision Transformer Technique

The ViT is inspired by the transformer architecture originally designed for machine translation tasks [19]. Unlike conventional Convolutional Neural Networks (CNNs), this model is specially tailored for image classification and ignores the decoding element normally utilized in transformers for sequence generation. An essential technology of ViT is its withdrawal from convolutional processes, instead using Multi-Layer Perceptrons (MLPs) for local feature learning and Multi-Head Self-Attention (MSA) mechanisms to capture global dependencies. This structure enables the method to effectively learn and classify images using self-attention across all patches. The original image is divided into learnable patches, flattened into feature vectors, and enhanced with learnable positional embeddings before being fed to the transformer encoder, as shown in (1). Here, the notation [·] denotes the vector's concatenation. The token x_{class} , representing the class embedding, serves as the feature vector y that encodes the entire image.

$$z_0 = [x_{class}; x_p^1 E; \ldots; x_p^N E] + E_{pos},$$
$$E \in \mathbb{R}^{(P^2 C)\times D},\ E_{pos} \in \mathbb{R}^{(N+1)\times D} \quad (1)$$

The incorporation of location embeddings and patch linear projections forms vectors in $\mathbb{R}^{(N+1)\times D}$, which are then forwarded to the transformer encoder. The next latent feature is calculated according to the preceding layer utilizing (2). Here, the function LN(·) signifies layer normalization, whereas MSA(·) represents multi-head self-attention. Then, the next layer processes $\tilde{z}$ through LN followed by an MLP with a residual link, as shown in (3). Finally, the output prediction is

obtained from the final representation of the class embedding x_{class}, as expressed in (4).

$$\tilde{z}_l = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1} \quad (2)$$

$$z_l = \mathrm{MLP}(\mathrm{LN}(\tilde{z}_l)) + \tilde{z}_l \quad (3)$$

$$y = \mathrm{LN}(z_l^0) \quad (4)$$

Within the MSA mechanism, the process includes learning the value (v), query (q), and key (k) vectors, as presented in (5). This computation is accelerated through multiplication by U_{qkv} . The significance of the features, characterized as attention weights A, are derived using a dot-product softmax operation between the key k and query q, as shown in (6). The weights A are then applied to calculate the self-attention over the value v utilizing (7). This procedure mimics gaining information in information recovery, whereas the query is the input when searching, the key serves as features or identifiers, and the value comprises the information.

$$[q, k, v] = zU_{qkv}, U_{qkv} \in \mathbb{R}^{D\times 3D_h} \quad (5)$$

$$A = \mathrm{softmax}\left(\frac{qk^T}{\sqrt{D_h}}\right),\ A \in \mathbb{R}^{N\times N} \quad (6)$$

$$\mathrm{SA}(z) = Av \quad (7)$$

$$\mathrm{MSA}(z) = [\mathrm{SA}_1(z); \mathrm{SA}_2(z); \ldots;\ \mathrm{SA}_k(z)]U_{msa},$$
$$U_{msa} \in \mathbb{R}^{kD_h\times D} \quad (8)$$

To improve the capability of the ViT to detect unidentified classes, Transfer Learning (TL) is applied by utilizing ViT as a feature extractor during classifier training. The resulting feature scores are then passed to the classification stage described in the following section.

3) ConvNeXt Model

The ConvNeXt model is a streamlined and contemporary adaptation of the famous ResNet architecture [20], where each stage begins with downsampling followed by repeated residual blocks. The network concludes with a classification head. Every stage starts with a downsampling block that minimizes spatial resolution and increases the spectral size. Initially, downsampling reduces spatial size by a factor of four, followed by two-fold reductions in later stages, producing images with lower spatial but higher spectral resolution. Then, a normalization stage is implemented, succeeded by the use of the dense layer to get the final outputs. In multi-class classification tasks, the softmax activation function is commonly employed to estimate class probabilities. However, in this study involving binary classification, the sigmoid activation function is used instead of softmax.

C. Stage III: Hybrid Classification Process

Finally, the hybrid LSTM-AE method is employed for the classification process. The hybrid LSTM-AE is used for efficiently capturing temporal dependencies and reconstructing latent features, thus enhancing class separability and robustness compared to conventional classifiers. The LSTM-AE method, which combines the capabilities of LSTM [21] and AE [22], establishes stronger performance in identifying faults, whereas dynamic and periodic changes are often affected by intrinsic anomalies. By reconstructing and learning time-series data, state transitions are successfully captured, permitting subtle anomalies to be identified. The model framework includes encoding, which compresses input time-series data into a lower-dimensional representation, and decoding, which reconstructs the input from these features. LSTM, an advanced Recurrent Neural Network (RNN), leverages historical data for time-series prediction while better handling long-term dependencies through its gating mechanism. This model combines LSTM with an AE architecture, where the encoder of LSTM learns the Hidden Layer (HL) at every time step of the input sequence, and the decoder rebuilds the sequence using the current HL and the preceding time-step reconstruction. Given a time series $X = [x_1, x_2, \ldots, x_t]$, where every sub-sequence of length T, $[x_1, x_2, \ldots, x_T]$, is extracted from X, and $x_t \in \mathbb{R}^m$ represents an m-dimensional input feature vector at time-step t, the LSTM encoder learns the consistent HL hE_i at each time step $i \in \{1,2, \ldots, n\}$. These HLs are then arranged into a 2D array, where the first dimension corresponds to sample count and the second dimension represents time steps. Standard time-series data are thus converted into 2D arrays. The HL at time step t_i denoted hE_i, is propagated through the network, and the final HL hE_n serves as the initial state for the LSTM decoder.

$$y(i) = w \cdot h_i^D + b \quad (9)$$

where z_t is used as input to compute hD_{t-1} in the decoder layer to sequentially reconstruct $y(i)$. The learnable parameter b denotes the bias term. This procedure continues until the complete time-series data has been recreated. x^t denotes the input vector at time t, and y^t is the corresponding output at time t.

$$\varepsilon_{loss} = \|X' - X\|^2 \quad (10)$$

III. EVALUATION METRICS AND PERFORMANCE ASSESSMENT

The performance of the HDLEM-FFTSRSI model was evaluated on the EuroSat dataset [23]. The method was implemented in Python 3.6.5 on a system with an Intel i5-8600K CPU, 4 GB GPU, 16 GB RAM, 250 GB SSD, and 1 TB HDD, using a learning rate of 0.01, ReLU activation, 50 epochs, 0.5 dropout, and a batch size of 5. The dataset used in this study contains 27,500 remote sensing images across ten land use and land cover classes: Highway, Annual Crop, Forest, Herbaceous Vegetation, Industrial, Pasture, Permanent Crop, River, Residential, and Sea/Lake. Figure 2 shows the confusion matrices created by the HDLEM-FFTSRSI approach for 80:20 and 70:30 Training Phase–Testing Phase (TRPHE/TSPHE) splits. The results indicate that the HDLEM-FFTSRSI method effectively detects and classifies each class.

Table I presents the classifier results of the HDLEM-FFTSRSI approach under 80:20 TRPHE/TSPHE. With 80% TRPHE, the HDLEM-FFTSRSI approach achieved an average accuracy ($accur_y$) of 99.11%, precision ($preci_n$) of 95.58%, recall ($recal_l$) of 95.51%, F1-score ($F1_{Score}$) of 95.54%, and Matthews Correlation Coefficient (MCC) of 95.05%. For 20% TSPHE, the model achieved an average $accur_y$ of 99.05%, $preci_n$ of 95.26%, $recal_l$ of 95.17%, $F1_{Score}$ of 95.21%, and MCC of 94.69%.

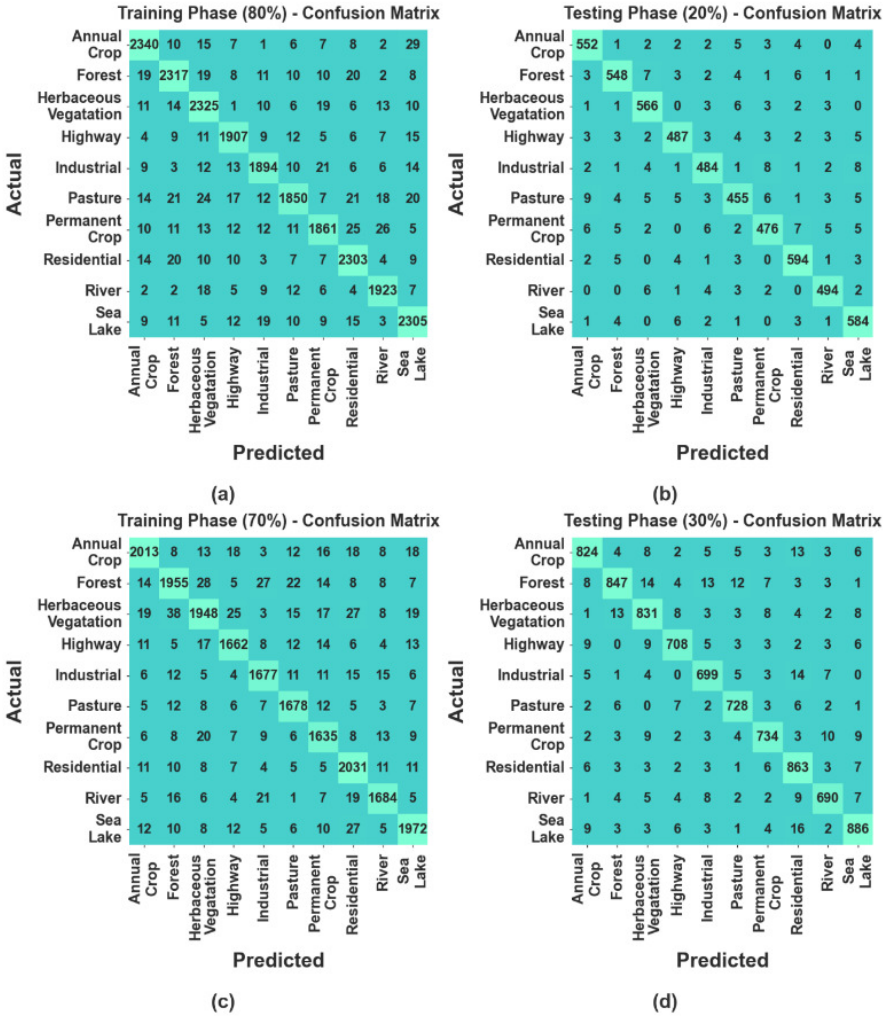

Fig. 2. Confusion matrices of the HDLEM-FFTSRSI model under different TRPHE/TSPHE splits: (a) TRPHE 80%, (b) TSPHE 20%, (c) TRPHE 70%, (d) TSPHE 30%.

TABLE I. CLASSIFIER PERFORMANCE OF HDLEM-FFTSRSI WITH 80:20 TRPHE/TSPHE

Class labels	$accur_y$	$preci_n$	$recal_l$	$F1_{Score}$	MCC
TRPHE (80%)	99.11	95.58	95.51	95.54	95.05
TSPHE (20%)	99.05	95.26	95.17	95.21	94.69

a. Average values (%).

Table II presents the classifier results under a 70:30 split. With 70% TRPHE, the HDLEM-FFTSRSI model achieved an average $accur_y$ of 98.97%, $preci_n$ of 94.84%, $recal_l$ of 94.87%, $F1_{Score}$ of 94.85%, and MCC of 94.28%. For 30% TSPHE, the model achieved an average $accur_y$ of 98.93%, $preci_n$ of 94.70%, $recal_l$ of 94.68%, $F1_{Score}$ of 94.68%, and MCC of 94.09%. Table III provides a comparative analysis of the HDLEM-FFTSRSI model with recent approaches [24, 25]. The results show that HDLEM-FFTSRSI achieved higher $accur_y$, $preci_n$, $recal_l$, and $F1_{Score}$ of 99.11%, 95.58%, 95.51%, and 95.54%, respectively. Other models, including Shallow CNN, GoogleNet, GeoSystemNet, ResNet50, SMS-EMOA, MobileNet, and RSSC-DBOEDL, show lower performance.

TABLE II. CLASSIFIER PERFORMANCE OF HDLEM-FFTSRSI WITH 70:30 TRPHE/TSPHE

Class labels	$accur_y$	$preci_n$	$recal_l$	$F1_{Score}$	MCC
TRPHE (80%)	98.97	94.84	94.87	94.85	94.28
TSPHE (20%)	98.93	94.70	94.68	94.68	94.09

a. Average values (%).

TABLE III. COMPARATIVE PERFORMANCE OF HDLEM-FFTSRSI WITH RECENT METHODS

Model	$accur_y$ (%)	$preci_n$ (%)	$recal_l$ (%)	$F1_{Score}$ (%)
Shallow CNN [24]	98.00	90.14	92.95	91.73
GoogleNet [24]	95.23	93.83	90.12	92.11
GeoSystemNet [24]	89.79	89.39	90.66	90.67
ResNet50 [24]	96.12	90.54	89.23	89.06
SMS-EMOA [25]	73.02	94.78	89.96	90.44
MobileNet [25]	87.52	94.78	91.54	93.01
RSSC-DBOEDL [25]	95.07	93.76	93.91	89.60
HDLEM-FFTSRSI [proposed]	99.11	95.58	95.51	95.54

Table IV presents the computational efficiency of the HDLEM-FFTSRSI approach [26]. The CaffeNet and fine-tuned MobileNetV2 required high computational resources and memory. In contrast, the HDLEM-FFTSRSI model achieved superior efficiency with the lowest FLOPs of 0.92 G, minimal GPU usage of 673 MB, and the fastest inference time of 1 ms, making it highly suitable for real-time or resource-constrained applications.

TABLE IV. COMPUTATIONAL EFFICIENCY COMPARISON OF HDLEM-FFTSRSI WITH RECENT METHODS

Method	FLOPs (G)	GPU (MB)	Inference time (ms)
LCNN-BFF	24.60	2,908	8.40
GoogLeNet	1.50	4,312	3.30
CaffeNet	715.00	3,122	6.98
VGG-VD-16	15.50	5,951	4.16
Fine-tune MobileNetV2	334.00	2,703	7.89
SE-MDPMNet	3.27	3,234	7.12
Contourlet CNN	2.10	5,614	2.48
HDLEM-FFTSRSI (proposed)	0.92	673	1.00

IV. CONCLUSION

This paper has presented the Hybrid Deep Learning-based Environmental Monitoring using Feature Fusion Techniques on Satellite Remote Sensing Images (HDLEM-FFTSRSI) model for effective environmental monitoring using remote sensing images. The proposed model mainly focuses on the design of a fusion-based feature extraction process through the integration of three Deep Learning (DL) models, namely EfficientNetV2-S, Vision Transformer (ViT), and ConvNeXt. The novelty of the HDLEM-FFTSRSI model lies in the integration of preprocessing, feature fusion, and a Long Short-Term Memory–Autoencoder (LSTM-AE) classifier, thus enhancing feature representation and classification accuracy. This integration provides a unified and robust framework for environmental monitoring using satellite imagery. The comparison analysis of the HDLEM-FFTSRSI method demonstrates a superior accuracy value of 99.11% compared with other models on the EuroSat dataset. Therefore, the proposed model is found to be an efficient and scalable tool for automated environmental monitoring. Future work may focus on the design of hyperparameter-tuned optimization algorithms to further improve the performance of the DL models.

DATA AVAILABILITY

The data that support the findings of this study are openly available from the Kaggle repository [23].

REFERENCES

[1] G. M. Lovett *et al.*, "Who needs environmental monitoring?," *Frontiers in Ecology and the Environment*, vol. 5, no. 5, pp. 253–260, June 2007, https://doi.org/10.1890/1540-9295(2007)5%255B253:WNEM%255D2.0.CO;2.

[2] D. B. Olawade, O. Z. Wada, A. O. Ige, B. I. Egbewole, A. Olojo, and B. I. Oladapo, "Artificial intelligence in environmental monitoring: Advancements, challenges, and future directions," *Hygiene and Environmental Health Advances*, vol. 12, Dec. 2024, Art. no. 100114, https://doi.org/10.1016/j.heha.2024.100114.

[3] H. Jafarbiglu and A. Pourreza, "A comprehensive review of remote sensing platforms, sensors, and applications in nut crops," *Computers and Electronics in Agriculture*, vol. 197, June 2022, Art. no. 106844, https://doi.org/10.1016/j.compag.2022.106844.

[4] R. Wang *et al.*, "Remote Sensing Application in Ecological Restoration Monitoring: A Systematic Review," *Remote Sensing*, vol. 16, no. 12, June 2024, Art. no. 2204, https://doi.org/10.3390/rs16122204.

[5] P. K. Rai, V. N. Mishra, and K. N. P. Raju, "Methodology and Applications of Remote Sensing and GIS in Environmental Mapping and Monitoring," *National Geographical Journal of India*, vol. 64, no. 1–2, pp. 266–276, June 2018.

[6] P. R. Maheshmalkar, S. B. Sayyad, and S. V. Kshirsagar, "SAR Remote Sensing for Environmental Monitoring," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 9, no. 5, pp. 46–51, July 2021.

[7] M. Gatcha, F. Messelmi, and S. Saadi, "An Anisotropic Diffusion Adaptive Filter for Image Denoising and Restoration Applied on Satellite Remote Sensing Images: A Case Study," *Engineering, Technology & Applied Science Research*, vol. 12, no. 6, pp. 9715–9719, Dec. 2022, https://doi.org/10.48084/etasr.5363.

[8] Y. Slimani and R. Hedjam, "A Hybrid Metaheuristic and Deep Learning Approach for Change Detection in Remote Sensing Data," *Engineering, Technology & Applied Science Research*, vol. 12, no. 5, pp. 9351–9356, Oct. 2022, https://doi.org/10.48084/etasr.5246.

[9] F. Safarov, U. Khojamuratova, M. Komoliddin, F. Bolikulov, S. Muksimova, and Y.-I. Cho, "MBGPIN: Multi-Branch Generative Prior Integration Network for Super-Resolution Satellite Imagery," *Remote Sensing*, vol. 17, no. 5, Feb. 2025, Art. no. 805, https://doi.org/10.3390/rs17050805.

[10] Y. Yang, H. Zhao, X. Huangfu, Z. Li, and P. Wang, "ViT-ISRGAN: A High-Quality Super-Resolution Reconstruction Method for Multispectral Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 3973–3988, 2025, https://doi.org/10.1109/JSTARS.2025.3527226.

[11] W. Meng, L. Shan, S. Ma, D. Liu, and B. Hu, "DLNet: A Dual-Level Network with Self- and Cross-Attention for High-Resolution Remote Sensing Segmentation," *Remote Sensing*, vol. 17, no. 7, Mar. 2025, Art. no. 1119, https://doi.org/10.3390/rs17071119.

[12] X. Li *et al.*, "A Spectral–Spatial Context-Boosted Network for Semantic Segmentation of Remote Sensing Images," *Remote Sensing*, vol. 16, no. 7, Mar. 2024, Art. no. 1214, https://doi.org/10.3390/rs16071214.

[13] W. Ren, Z. Wang, M. Xia, and H. Lin, "MFINet: Multi-Scale Feature Interaction Network for Change Detection of High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 16, no. 7, Apr. 2024, Art. no. 1269, https://doi.org/10.3390/rs16071269.

[14] R. Zhang, H. Zhang, X. Ning, X. Huang, J. Wang, and W. Cui, "Global-aware siamese network for change detection on remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 199, pp. 61–72, May 2023, https://doi.org/10.1016/j.isprsjprs.2023.04.001.

[15] J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and Lightweight: An Improved YOLOv5 Object Detector for Remote Sensing Images," *Remote Sensing*, vol. 15, no. 20, Oct. 2023, Art. no. 4974, https://doi.org/10.3390/rs15204974.

[16] Z. Wang *et al.*, "Optimized faster R-CNN for oil wells detection from high-resolution remote sensing images," *International Journal of Remote Sensing*, vol. 44, no. 22, pp. 6897–6928, Nov. 2023, https://doi.org/10.1080/01431161.2023.2275322.

[17] V. S. Padmavathy and R. Priya, "Image contrast enhancement techniques-a survey," *International Journal of Engineering & Technology*, vol. 7, no. 2.33, pp. 466–469, June 2018, https://doi.org/10.14419/ijet.v7i2.33.14811.

[18] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, Online, 2021, pp. 10096–10106.

[19] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, June 03, 2021, https://doi.org/10.48550/arXiv.2010.11929.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022, pp. 11966–11976, https://doi.org/10.1109/CVPR52688.2022.01167.

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006, https://doi.org/10.1126/science.1127647.

[23] "EuroSat Dataset." Kaggle. [Online]. Available: https://www.kaggle.com/datasets/apollo2506/eurosat-dataset.

[24] S. A. Yamashkin, A. A. Yamashkin, V. V. Zanozin, M. M. Radovanovic, and A. N. Barmin, "Improving the Efficiency of Deep Learning Methods in Remote Sensing Data Analysis: Geosystem Approach," *IEEE Access*, vol. 8, pp. 179516–179529, 2020, https://doi.org/10.1109/ACCESS.2020.3028030.

[25] M. Alamgeer, A. Al Mazroa, S. S. Alotaibi, M. H. Alanazi, M. Alonazi, and A. S. Salama, "Improving remote sensing scene classification using dung Beetle optimization with enhanced deep learning approach," *Heliyon*, vol. 10, no. 18, Sept. 2024, Art. no. e37154, https://doi.org/10.1016/j.heliyon.2024.e37154.

[26] C. Shi, X. Zhang, J. Sun, and L. Wang, "Remote Sensing Scene Image Classification Based on Self-Compensating Convolution Neural Network," *Remote Sensing*, vol. 14, no. 3, Jan. 2022, Art. no. 545, https://doi.org/10.3390/rs14030545.