LiteDenseMoE: An Explainable Lightweight Densely Connected Mixture-of-Experts Network for Aerial Scene Recognition in Low Contrast Remote Sensing Images

Muhammad John Abbas, ***Member IEEE*****, Muhammad Attique Khan,** ***Member IEEE*****, Ameer Hamza,** ***Member IEEE*****, Shrooq Alsenan, Areej Alasiry, Mehrez Marzougui, Jungpil Shin,** ***Senior Member, IEEE*****, Yunyoung Nam,** ***Member IEEE***

Abstract— Land Remote sensing image classification is crucial for understanding ongoing geographical and environmental changes. It aids in land use and land cover classification, crop and vegetation classification, change detection, and classification of coastal and aerial regions. Many advanced techniques were introduced based on some substantial modifications in the models; however, this resulted in a complex framework that is difficult to adapt. In this work, we proposed a novel Lightweight Dense Mixture of Experts (LiteDenseMoe) model for aerial and coastal regions classification using remote sensing images. The proposed model initially incorporates light, dense blocks with lightweight dense layers, as well as channel and spatial attention mechanisms. The resulting model is further fused with an Mixture of Experts block that extracted more relevant and essential features for the accurate prediction of complex aerial scenes. In the training process of the proposed model, a Hyperband Optimization technique is employed for hyperparameter initialization, rather than manual selection. After training the proposed model, classification was performed, along with output interpretation. The proposed LiteDenseMoe architecture is evaluated on three datasets and achieved an accuracy of 93.25% on MLRSNet, 92.56% on NWPU-RESISC45, and 96.54% on the EuroSAT dataset with only 0.3 million parameters. Expert allocation and their confidence per class, Expert disagreement Network, and t-SNE visualization are also observed to interpret the Moe results. Detailed Ablation studies and comparative analysis with pre-trained and SOTA models confirm the impact and efficiency of the proposed architecture for aerial and coastal regions classification.

Index Terms— Remote sensing; Aerial scene; Deep learning; Hyperparameter tuning; Mixture of Experts; Model interpretations

I. Introduction

Numerous socioeconomic and environmental applications, including urban and regional planning as well as the management and conservation of natural resources, depend on continuously updated land use and land cover data. [1, 2]. Remote sensing (RS) is defined as the science of obtaining information about the land and water bodies of the Earth from the images acquired from a distance using electromagnetic radiation emitted [3, 4]. It is beneficial as it allows us to monitor and understand the complex geological and environmental processes which cannot be tracked otherwise [5]. The history of remote sensing dates back to the early 1800s with the discovery of infrared radiation to aerial photography using balloons and airplanes; however, the term "remote sensing" was first used in the 1960s as the term "aerial photography" no longer justified the several forms of images collected using the invisible electromagnetic spectrum [4]. Over time, several evolutions have impacted the field of remote sensing, including satellite remote sensing, digital image processing, hyperspectral remote sensing, global remote sensing, and Lidars, among others [3]. These modern advancements lead to more complex and enriched remote sensing data, which can be used for more precise classification and detection of features on the Earth's surface [6].

Despite the growing capabilities of remote sensing, the analysis and classification of remote sensing data are still challenging tasks due to their high dimensionality and high spectral similarity [7, 8]. Many RS classes share overlapping visual characteristics which creates a confusion between classes like parking lots and bare lands, urban residential and industrial zones etc. Also, there is high intra-clas variability in RS data as a same land cover class shows visual variations ubder different environmental and seasonal changes. Apart from this, multi-scale feature requirements and limited labeled

Funding: This work is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R506), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00218176) and the Soonchunhyang University Research Fund. The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for Large Research Project under Grant RGP2/471/46.

Corresponding author e-mail: attique.khan@ieee.org, ynam@sch.ac.kr).

Muhammad John Abbas and Muhammad Attique Khan are with Center of AI, Prince Mohammad bin Fahd University, Al-Khobar, KSA. (johnabbas@ieee.org; attique.khan@ieee.org)

Ameer Hamza is with Centre of Real Time Computer Systems, Kaunas University of Technology, Lithuania (ameerhamza@ieee.org).

Areej Alasiry, Mehrez Marzougui are with College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia (areej.alasiry@kku.edu.sa; mhrez@kku.edu.sa)

Shrooq Alsenan is with Information Systems Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia. (shaalsenan@pnu.edu.sa).

Jungpil Shin is with School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan (jpshin@u-aizu.ac.jp).

Yunyoung Nam is with Department of ICR Convergence, Soonchunhyang University, South Korea. (ynam@sch.ac.kr)

data further increases the challenges, making it difficult for model to leann consistent class representations.

Initially, traditional machine learning (ML) algorithms such as Support Vector Machines (SVM) [9], decision trees (DT) [10], Random Forest (RF) [11] and Artificial Neural Networks (ANNs) [12] were used for remotely sensed image classification. However, these algorithms require manual feature extraction, due to which the model's performance becomes highly dependent on extracted features [13]. Moreover, their inability to deal with high-dimensional data shifts the researchers' interest towards deep learning [14]. Unlike Machine learning, deep learning algorithms support automatic feature extraction and also show good performance on complex hyperspectral remote sensing data [15-17]. Many techniques have been introduced in the literature for the classification and detection of land use and land cover from remote sensing and hyperspectral images [18-20].

K. Ali et al. [21] used a convolutional neural network (CNN) for the classification of land cover areas in semi-arid regions through Sentinel-2 satellite images. They trained the model on different areas and tested it on unseen regions to check whether the model can classify areas in similar environments. Also, they compared two types of Sentinel-2 satellite images, 4-band and 10-band, to check their ability to classify difficult land regions. Experimental findings reveal that a CNN with the proposed architecture trained on 4-band images outperformed. However, the architecture of the model can be improved as it still struggles to differentiate between highly similar spectral regions. In [22], the authors evaluated different deep learning models on benchmark datasets and compared their performance for LiDAR point cloud classification. They evaluated two types of models: projection-based (U-Net, ResNet, VGG, and DeepLab) and point-based (DGCNN, ConvPoint, PointNet, PointNet++) on four benchmarking datasets (Toronto3D, S3DIS, ModelNet40, ISPRS Vaihingen). Experimental results demonstrate that DCGNN and ConvPoint surpass other models by achieving the highest accuracy across all the datasets. However, the use of both models is limited as DCGNN is computationally heavy and ConvPoint is scale-agnostic. In [23], the authors compared three deep learning models for remotely sensed image classification. The convolutional neural network is the first model of this work that is built from scratch; however, the other models are pre-trained, such as EfficientNetB7, and fine-tuned EfficientNetB7. All these models are evaluated on the UCM Land use dataset, which comprises 21 classes, each containing 100 images. Results show that fine-tuned EfficientNetB7 achieved the highest accuracy of 88% followed by pre-trained EfficientNetB7 and CNN-FE. However, the dataset used in this study is relatively small, which limits the model's performance in terms of generalizability. M. Aljebreen et al. [24] presented a novel technique for land use land cover classification which employs River flow dynamic algorithm with deep learning (LULCC-RFDADL). It incorporates a pretrained CNN model, Dense-EfficientNet, for feature extraction; however, it also utilizes a River Flow Dynamic Algorithm for hyperparameter selection and a Multi-scale Convolutional Autoencoder (MSCAE) for classification. Moreover, they used the Seeker Optimization Algorithm for parameter optimization. Experiments are conducted on the Eurostat dataset, which contains 10 classes, each with 500 images. Experimental results show that the proposed model outperforms other deep learning models by achieving an overall accuracy of 98.12% and an average precision, recall, and F1-score of 90.7% across all the classes. The drawback of this model is its computational expense, and performance can vary with variations in data quality.

M. Fayas et al. [25] evaluated different deep learning models for accurate and efficient land cover classification using high-resolution remote sensing imagery. The study focuses on three DL models, namely Inception-v3, ResNet-50, and DenseNet-121. They used these models based on a fine-tuning process where they froze the top layers and added custom layers. All the models are evaluated on the UC-Merced_LandUse dataset, which comprises 18,000 images across 18 classes, and are also compared with State-Of-The-Art models. Experimental findings reveal that Inception-v3 surpassed all the models by obtaining an accuracy, precision, recall, and F1-score of 92%, 93%, 92% and 92%, respectively. F. S. Alsubaei et al. [26] introduced a block scrambling-based encryption technique with deep learning for remote sensing image classification. The study involves the encryption of RS images for preservation of transmission, storage, and classification of these encrypted images using deep learning techniques. Encryption is performed by dividing the image into non-overlapping blocks, which then undergo random shuffling, flipping, and rotations to make it compatible with JPEG standards. These encrypted images are then passed through DenseNet for feature extraction, whereas Artificial Gorilla Troops Optimizer (AGTO) is employed for hyperparameter optimization. The proposed model is evaluated on the UC Merced land use dataset and shows a classification precision rate of 98%. In [27], the authors presented a novel deep learning model, ResMoCNN, for the classification of Hyperspectral images by injecting morphological features into 3DCNN features via residual connections. The model incorporates a 3DCNN core for hierarchical feature extraction and a Spatial-Spectral Morphology box (SSMB) for structural and environmental feature extraction. These morphological features are extracted through four morphological operators, namely erosion, perimeter, dilation, and top-hat. These extracted features are then injected into multiple 3DCNN layers via residual connection. Before feeding it to the model, HSI data is preprocessed by PCA for dimensionality reduction. The model is trained and tested on four datasets and shows that the proposed model outperforms traditional ML and DL models by achieving an overall classification accuracy of 97.81% on the Indian Pines dataset, 99.33% on the Pavia University dataset, 98.67% on the Houston University dataset, and 99.71% on the Salinas dataset. Albarakati et al. [28] presented a novel deep learning approach based on information fusion of deep convolutional neural networks for remote sensing image classification. In this study, they implemented a Super Resolution (SR) technique to improve the contrast of images. After that, the enhanced images are passed to two separate models such as ResSANS6 and RS-IRSAN. The extracted features from both architectures are then fused by Mutual Information-Based Serial Fusion (MIBSF) and standardized by median normalization. An Arithmetic Optimization Algorithm (AOA) is employed to select optimal features while a Shallow Wide Neural Network (SWNN) is used as a classifier. Experiments are conducted on three datasets, and results show that the proposed technique achieved classification accuracy of 95.7% on RSI-CB128, 97.5% on WHU-RS19, and 92.0% on NWPU_RESISC45 dataset, respectively. Some studies focus on deep learning and design architectures for the classification of satellite images, such as SemHi [29], which is based on the SwinUNETR, custom CNN, pre-trained dense, and ResNet architectures [30, 31].

Most of the techniques proposed so far are based on either pre-trained deep learning models or custom CNN models. These techniques have some common limitations such as overfitting, high computational cost, limited generalization, and data dependency. Also, there is minimal research on the classification of aerial and coastal regions through RS data. In this study, we proposed a novel deep learning-based Mixture of Experts model named Lightweight Dense Mixture of Experts (LiteDenseMoe) for the accurate and efficient classification of aerial and coastal regions through remote-sensed imagery. The proposed model addresses the challenges in RS data through it novel architecture. The Mixture of Experts mechanism integrated in the architecture enables the model to learn specialized representations for different classes, thus overcoming intra-class variability. Also, the attention mechanisms highlight distinct spatial and spectral features, clearing the inter-class confusion. Multi-scale features are captured by dense connections, and a lightweight architecture prevents overfitting on limited data. These contributions collectively lead to more accurate and efficient classification. Our main contributions and the significant challenges that are addressed by these contributions are as follows:

- Efficient Light-weight Architecture: We proposed a lightweight dense block modified with a depth-wise separable convolutional layer to minimize the computational cost while preserving the high representational ability.
- Customized MoE: We designed a novel MoE model that consists of multiple specialized expert blocks and a routing mechanism that dynamically assigns the features to the most appropriate expert.
- Manual hyperparameter optimization is a time-consuming task, which can be replaced by an automated optimization technique such as Hyperband Optimization, which is employed in this model.
- Interpretability and Analysis: A comprehensive series of interpretability studies is performed, including some extensive expert engineer allocation analysis, confidence visualization, t-SNE feature space visualizations and GradCAM visualization to further explain the behavior and specialization of the proposed model.

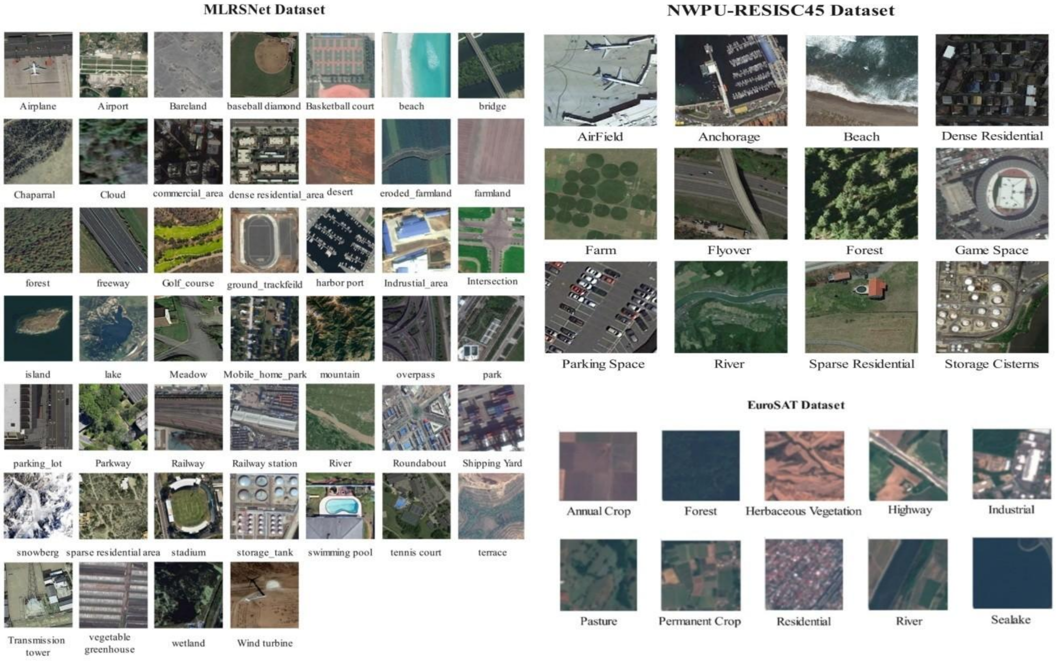

Figure 1: Sample images of the selected remote sensing datasets

II. DATASET DESCRIPTION

In this work, we utilized three datasets for the evaluation of the proposed architecture such as MLRSNet dataset [32], NWPU-RESISC45 [33], and Coastal areas combined dataset.

MLRSNet dataset: The MLRSNet dataset consists of 109161 high-resolution images divided into 46 categories, and the number of images in each category varies from 1500 to 3000. Each image has a fixed pixel size of 256×256 and pixel resolution ranging from 10m to 0.1m. Each image is tagged using pre-defined 60 class labels, and the number of labels for each image varies from 1 to 13. Sample images of the MLRSNet dataset are shown in Figure 2.

NWPU-RESISC45 dataset: The NWPU-RESISC45 dataset is composed of 10500 images separated into 12 classes such as

Airfield, Harbor, Beach, Dense residential, Farm, Overpass, Forest, Game space, Parking space, River, Sparse residential and Storage tanks. Each image has a pixel size of 256×256×3 with dpi of 96×96. Sample Images of NWPU-RESISC45 dataset are shown in the Figure 3.

Coastal Areas Combined Dataset: We acquired coastal-related classes from several publicly available datasets such as EuroSAT, MLSRNet, and SIRI-WHU to classify coastal areas. We selected 13 classes from these datasets, as shown in Figure 1. The dataset contains 13 classes: Anchorage, beach, harbor, harbor & port, island, lake, landslide, red sea fish, river, snow berg, swimming pool, water, and wetland. The size of each sample is 256×256×3, and the nature of the samples is RGB. The total number of samples in the collected dataset is 9206.

III. PROPOSED LITEDENSEMOE

In this section, we presented our proposed Lightweight Dense Mixture of Experts (LiteDenseMoe) model for aerial and coastal regions classification from remote sensing images. Convolutional Neural Network (CNN) is one of the most fundamental architectures of deep learning, which was initially introduced as LeNet by Yann LeCun [34]. It can capture hierarchical features at multiple receptive fields due to its convolutional, dense, and pooling layers. Initially proposed, CNN has some limitations such as the vanishing gradient problem and degradation with deeper architectures, which were later improved by its different variants such as DenseNet, ResNet, and Inception. Despite their improved performance, the inability of CNNs to recognize essential features leads to the introduction of attention modules (i.e., spatial and channel attention). These modules highlight the important channels and features of the image and suppress the less important ones. However, these attention-driven models were still unable to deal with diverse input data, which laid the foundation of Mixture of Experts (MoE). MoE is composed of different expert blocks and routing mechanisms that can train multiple expert blocks differently to handle diverse input images. In this work, considering the individual advantages of various deep learning models, we proposed a novel network, LiteDenseMoe, which incorporates DenseNet, channel and spatial attention blocks, and a Mixture of Experts for the classification of aerial and coastal images. A complete architecture of the proposed model is shown in Figure 2. The detailed description of this model is given below in subsections.

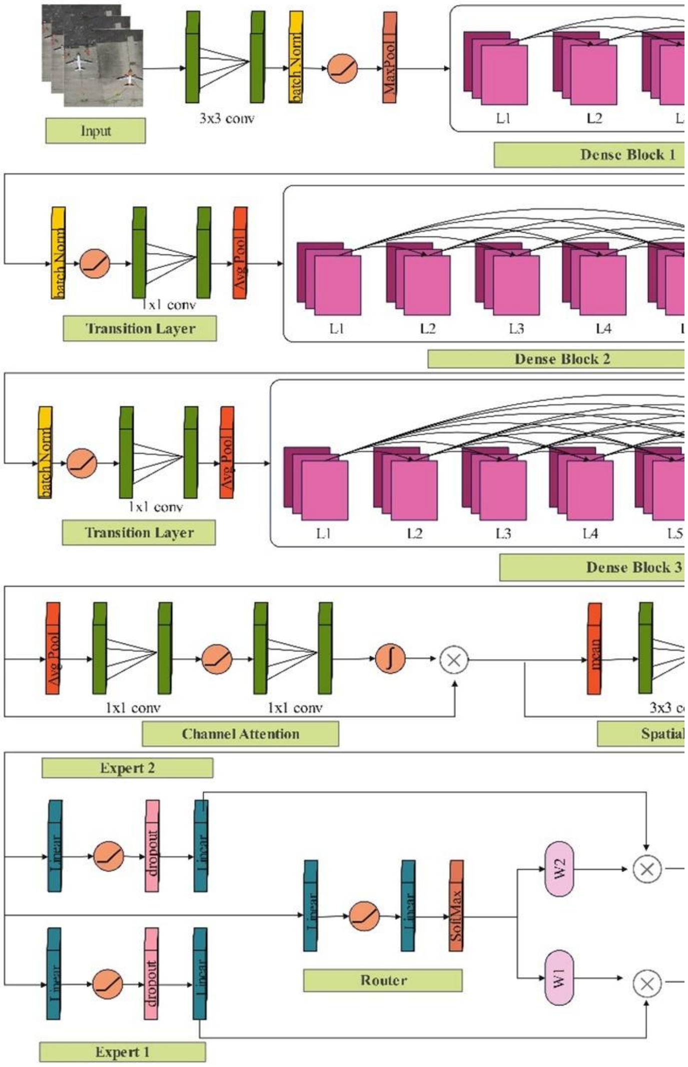

Figure 2: Proposed LiteDenseMoe for remote sensing image classification

A. Detailed Architecture

The detailed layer wise architecture detail is given under this subsection based on the mathematical formulation and visual representations of the inside modules of the proposed model.

1) Initial Layers:

The model starts with an input layer that accepts an image of size 224 × 224 × 3. The input image pixels are pass it to a 7 × 7 convolutional layer with stride of 2 × 2 and padding of 3 × 3 for initial feature extraction. The extracted features of this layer are then followed by a batch normalization layer (BNL) $\boldsymbol{\beta}$ that normalize the inputs and ReLU activation function $\boldsymbol{\sigma}$, and maxpooling layer (MPL) $\boldsymbol{\mathcal{M}_{\wp}}$ with pool size of 3 × 3 is employed. The initial layers are mathematically presented as:

$$Z = \mathcal{M}_{\wp}(\sigma\left(\beta(W_i * X + b_i)\right) \quad (1)$$

Where X denoted the input, W_i represents the convolutional weights, b_i is the bias and $*$ presented the convolutional operation.

2) First Dense Block

The first dense block is employed with four dense layers, as shown in Figure 3. In dense block, depth wise separable convolutional layer is employed instead of traditional convolutional layer for parameter efficiency. In a dense layer BNL layer followed by ReLU is attached. After that a 1 × 1 convolution is connected that followed by a BNL and ReLU activation layers. Later on, the output of this layer is passing to depth wise 3 × 3 separable convolutional layer. After that, a pointwise convolutional layer, BNL layer, and a ReLU activation is attached. Mathematically, this block is presented as follows:

$$y = \beta(\sigma\left(W_1^{(d)} * Z + b_1^{(d)}\right) \quad (2)$$

$$y' = Conv_{DW}\left(\sigma\left(\beta(y)\right)\right) \quad (3)$$

Where $Conv_{DW}$ denotes the depth wise separable convolutional layer. The final output of a dense layer is then concatenated with outputs of previous layer and pass on to the next layer, defined as:

$$Y_L = [y', Y_{L-1}, \dots . Y_0] \quad (4)$$

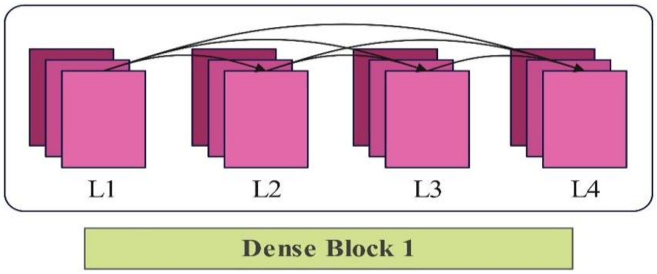

Figure 3: Proposed Dense block with four layers

3) Transition Layers Block 1

After first dense block, a transition block has been attached. The transition block is consist of BNL layer, convolutional layer, and ReLU activation. A 1 × 1 convolutional layer is added to reduce number of channels; however, an average pooling layer is added in the last to reduce spatial dimensions of the previous output, as shown in Figure 4. Mathematically, this process is formulated as:

$$T_1 = A_\rho\left(W_1^{(t)} * \left(\sigma\left(\beta(Y)\right)\right) + b_1^{(t)}\right) \quad (5)$$

Where A_ρ denoted the average pooling layer and T_1 presented the first transition layers block.

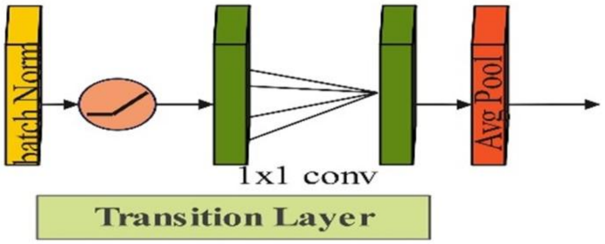

Figure 4: Proposed first transition layers block

4) *Second Dense Block*

After the first transition layer block, a second dense block has been connected. The second dense block is composed of 6 dense layers. At this time, the number of dense increases to capture representations that are more complex, as visually shown in Figure 5. All the layers in this block are connected to the next, respectively.

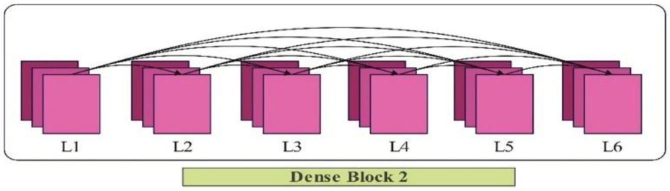

Figure 5: Proposed dense block with six layers

5) *Transition Layers Block 2*

Another transition layer is followed by this dense block to prevent the model from high complexity. The second transition layers block fallows the same phenomena as previously defined in Figure 4.

6) *Third Dense Block*

After second transition layers block, third dense block has been added with 8 dense layers. This dense block layers also consist of 1×1 convolution, BNL, ReLU activation, depth wise 3×3 separable convolutional layer, pointwise convolutional layer with BNL layer, and ReLU activation, as shown in Figure 6.

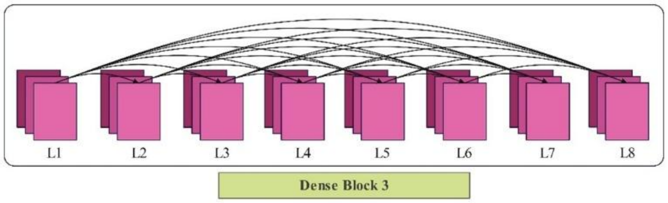

Figure 6: Proposed Dense block with eight layers

7) *Channel and Spatial Attention Module*

In the next phase, channel and spatial attention modules are employed. Channel attention module consist of an average pooling layer and pass it to 1×1 convolutional layer to reduce number of channels by a factor of 16 and learn channel wise relationships. Another 1×1 convolutional layer is employed to restore the original dimensions after feature processing. A ReLU activation function is applied between these two convolutional layers, and sigmoid activation function is employed after the second convolutional layer to scale the channel importance. This weight matrix is then multiplied with original tensor to emphasize the important channels. Mathematically, this process is defined as follows:

$$A_{ch} = \psi\left(W_2^{(CA)} * \sigma\left(W_1^{(CA)} * A_\rho(Y_3)\right)\right) \quad (6)$$

$$Y_{ch} = Y_3 \otimes A_{ch} \quad (7)$$

Where A_{ch} denoted the channel attention weight, Y_3 represent output of third dense block, ψ denotes the sigmoid activation function, Y_{ch} is the output of channel attention mechanism, and $\otimes$ represents element wise multiplication.

The output of the channel attention module is passed to the spatial attention mechanism to enhance the critical spatial regions. First of all, it aggregates channel-wise information to a spatial map and then applies a 3×3 convolution layer on it to learn spatial relationships. A sigmoid activation function is used to scale the weights, which are ultimately multiplied with the original input tensor to enhance important spatial regions. Mathematically, this process is presented as follows:

$$A_s = \psi\left(W_1^{(SA)} * A_\rho(Y_{ch})\right) \quad (8)$$

$$Y_s = A_s * Y_{ch} \quad (9)$$

Where A_s presented the spatial attention weights and Y_s denoted the spatially emphasized feature map. The channel and spatial attention module is shown in Figure 7. After this module, BNL, ReLU, average pooling layer is employed and then passed to the flatten layer.

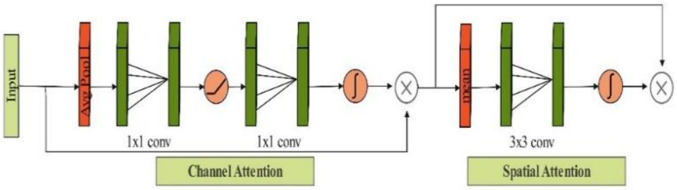

Figure 7: Proposed Channel Attention Mechanism

8) *MoE Experts and Routing Mechanism*

In this next phase, two MoE experts are connected with the output of the flattened layer. Each expert block is composed of two linear layers, a dropout layer, and a ReLU activation function. First, the linear layer projects the features into a higher-dimensional feature space to learn non-linear relationships. Then, a dropout layer with a 0.5 dropout factor is applied, which randomly discards 50% of neurons to prevent the model from overfitting. Another linear layer is utilized to map the hidden representations to the output nodes. An expert block is mathematically defined as:

$$E = \mathcal{L}_2\left(\eth, 0.5\left(\sigma\left(\mathcal{L}_1(Y_F)\right)\right)\right) \quad (10)$$

Wher Y_F denoted the output of flatten layer, $\mathcal{L}_1$ and $\mathcal{L}_2$ are linear layers, and ð denotes the dropout layer. After experts, a routing mechanism is employed to assign the weights. The routing mechanism is composed of a linear layer followed by a ReLU activation layer. Another linear layer is added after that and is followed by a SoftMax. During learning, features are passed through both expert blocks and the routing mechanism simultaneously. Initially, router assigns random but different weights to both expert blocks. These weights are refined during learning, which ultimately results in two expert blocks with other properties. Only the router is aware of expert properties, so when a test image comes, the router analyzes it and then assigns weights to both experts according to their capabilities and relevant features. The outcome of each expert block is multiplied by its corresponding weight, which is then added to generate the final output. The experts and the routing mechanism are visually presented in Figure 8. It can be represented as:

$$Y_{final} = \left((E_1 \times R_1) + (E_2 \times R_2)\right) \quad (11)$$

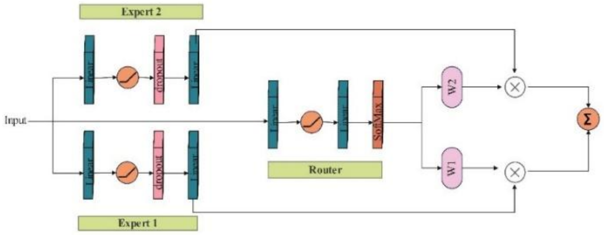

Figure 8: Proposed Mixture of Expert block for aerial scene and coastal regions classification

9) *Final Output and Auxiliary Classifier*

The final output is generated by combining the outputs of both expert blocks, which are later passed through SoftMax activation. The softmax activation converts the raw logits into probabilities, and the class with the highest probability will be output as the final prediction. However, apart from final classification, an auxiliary classifier is also introduced after the first dense block. This auxiliary classifier provides intermediate predictions, helping the model learn at its initial stages. The auxiliary classifier is composed of a 1×1 convolutional layer, a BNL layer, a ReLU layer, an average-pooling layer, a flatten layer, and a linear layer for classification output. An auxiliary classifier is shown in Figure 9. The categorical cross-entropy is utilized as a loss function, and the proposed model has only 0.3 million trainable parameters. Both final and auxiliary losses are added to the total loss. Mathematically, it can be defined as:

$$L_{total} = \lambda_1 \times L_{main} + \lambda_2 \times L_{aux} + \lambda_3 \times L_{router}$$
(12)

Where L_{main} is te final classification loss, L_{aux} is auxiliary classification loss and L_{router} is the load balancing loss to ensure balanced expert utilization. λ_1, λ_2 and λ_3 are respective weights assigned to these losses and their values are set to 0.6, 0.3 and 0.1 respectively.

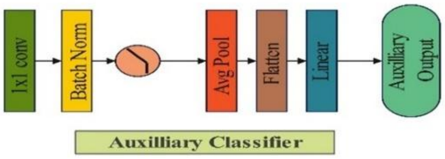

Figure 9: Proposed Auxiliary classifier for aerial scene classification

IV. HYPERPARAMETER SELECTION AND MODEL TRAINING

After designing the model, the selected datasets are divided into training and a testing set. 70% of the images from each dataset are utilized for training, and the remaining 30% of the data is employed for testing. The data is divided into a random process. Hyperband Optimization is used to select hyperparameters dynamically, rather than through manual selection.

A. *Hyperband Optimization*

Hyperparameter selection plays a vital role in model performance. Therefore, to select the most optimal hyperparameters, we applied the Hyperband Optimization technique [35], which finds the best hyperparameters while using limited resources. This technique starts by selecting the minimum budget per bracket (number of epochs) R and reduction factor η. This minimum budget and reduction factor is used to calculate total number of brackets and configurations per bracket. It then samples these numbers of configurations and divides the budget across them equally. During the first bracket, all configurations are trained using the limited budget, and then the one that performs well advances to the next bracket. After each bracket, the number of configurations is reduced by the reduction factor. Therefore, if initial configurations were 18 and the reduction factor is 3, only the top 6 performing configurations will go to the next round. In the next bracket, the budget will be divided across the remaining configurations so that each configuration will receive a larger budget for training. Again, the top-performing configurations will go to the next round. This process will continue until only one configuration is left. This configuration will be trained using the entire budget, and the selected hyperparameters will be used for the model's training. In this way, this technique selects the most optimal hyperparameters at a lower computational cost. Total number of brackets, configurations per bracket, and budget per configuration are calculated as follows:

$$s_{max} = log_{\eta}(R) \tag{12}$$

$$n = \frac{\eta^s}{s} \tag{13}$$

$$B = \frac{R}{\eta^s} \tag{14}$$

In this study, minimum budget was selected 30 and reduction factor was set to 3. The hyperparameters range and the best configuration selected by the optimization technique are shown in the Table 1.

Table 1: Hyperparameter range and selected best configuration

Hyperparameter	**Range**	**Best Configuration**
Epochs	30-100	50
Batch size	16, 32, 64,128	128
Learning rate	0.0001 – 0.01	0.001
Optimizer	Adam, SGD	Adam

B. *Training and Testing Process*

After the model design and hyperparameter selection, the next step is to train the model on the selected remote sensing datasets. The training curves are shown in Figure 10. In this figure, the training and testing plots for all datasets are visualized. The first part of this figure shows the curves for the MLRSNET dataset, indicating a stable and prosperous learning process. In the loss plot, the training and testing loss curves decline smoothly across the 50 epochs, with the testing loss declining smoothly and steadily approaching the training loss, which suggests good generalization. Both training and testing accuracy improve smoothly, as shown in the accuracy plot, with testing accuracy reaching over 90% by the end of epoch 50.

In the second part of Figure 10, the training and testing plots of the EuroSAT dataset have been illustrated. The training loss demonstrates a consistent, steady decline, while the testing loss exhibits significant fluctuations, particularly in the earlier epochs, indicating that it is sensitive to the dataset. As training continues, the training and testing loss converge and stabilize at a much lower level. The accuracy plot also reflects fluctuations at the earlier stages of training via testing accuracy. Still, by epoch 20, it has stabilized and improved, with its trajectory matching that of the training accuracy, with testing accuracy slightly below the training accuracy, which rose over 95%.

In the third part of Figure 10, the training and testing plots for the NWPU dataset have been added. Both training and testing loss exhibit a clear downward trend; however, the testing loss has more noise than the training loss. This performance indicates that the training loss exhibits greater stability compared to the testing graph. Arguments can be made that although the fluctuations in testing loss are less extreme than the previous dataset, they suggest some variance in overall model performance across validation batches. The accuracy plot produces the same results, where both training and testing accuracy exhibit an overall steady increase over the epochs; however, testing accuracy does not reach the same level of training accuracy. The testing accuracy for this dataset is above 85%, whereas the training curve reaches 90%.

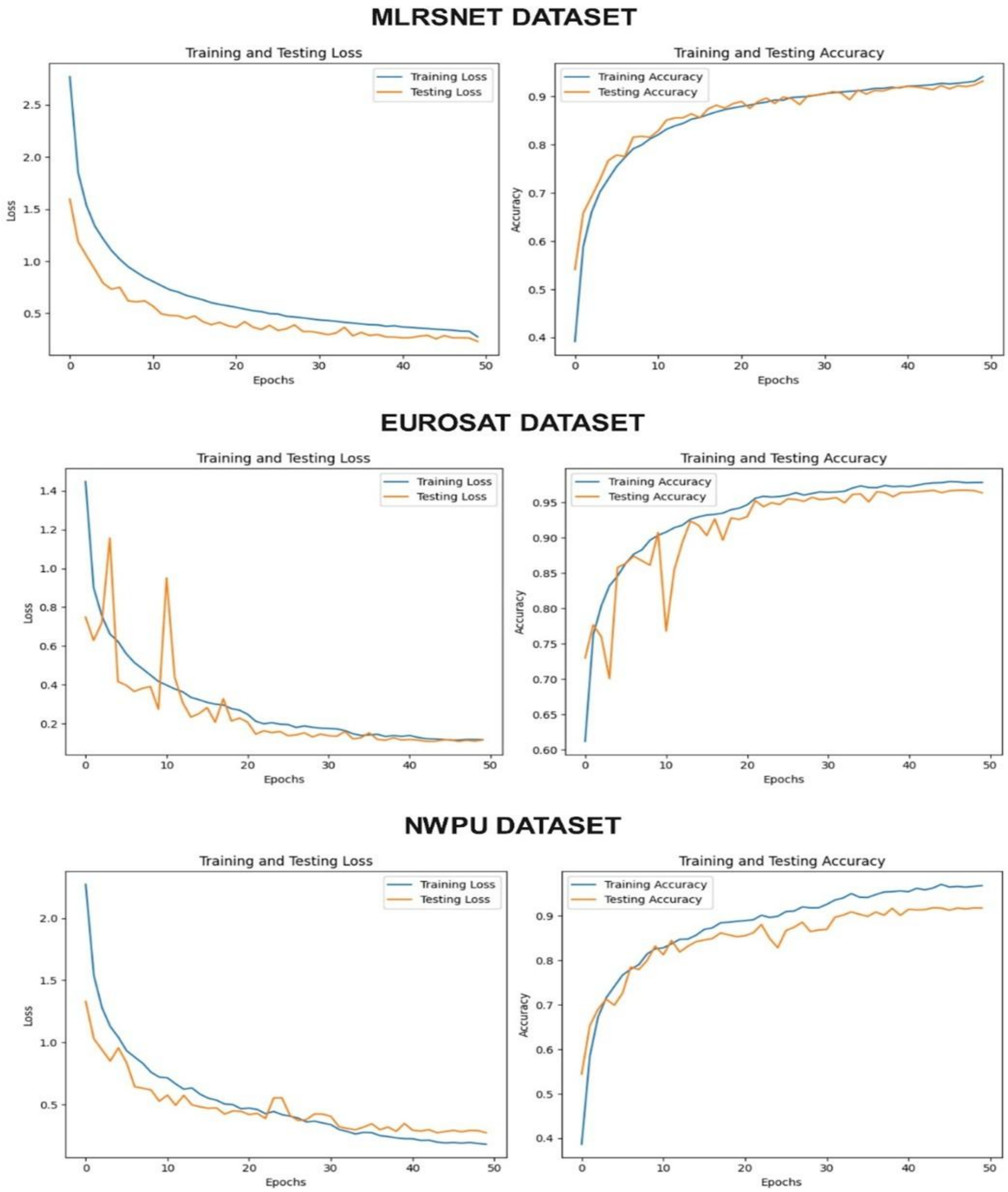

Figure 10: Training and validations plot against different datasets of this work using proposed model

This article has been accepted for publication in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. This is the author's version which has not been fully e
content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2025.3645113

V. Experimental Setup

In this section, the experimental setup has been discussed. The hyperparameters, such as mini-batch size, learning rate, optimizer, and epochs, are dynamically selected using the hyperband optimization. The static hyperparameters are learning decay is 0.2, and min LR is 0.00001. The performance of the proposed model is evaluated using traditional metrics such as accuracy, precision, recall, F1-score, and confusion matrix. To interpret the proposed model decisions, we visualize the expert allocation for all the classes and the confidence of the expert in their predictions. Disagreement between the expert predictions is analyzed to understand the learning behavior of the proposed model. Lastly, T-SNE visualization of the feature space is performed to visualize the classification abilities of the proposed model. All experiments were conducted using Python 3.10 and the PyTorch library on a Desktop Computer Equipped with 128GB RAM, an NVidia 20 GB RTX A4500 graphics card, and a 512 GB SSD Drive.

A. Results on MLRSNet dataset

The classification results achieved using the proposed LiteDenseMoE model on the MLRSNet dataset have been presented in Table 2. In this table, the proposed model achieved an overall classification accuracy of 93.25% which demonstrates the capacity of the architecture to handle complex aerial imagery. The complex aerial imagery contains both high intra-class variability and inter-class similarity. After closely inspecting the confusion matrix as shown in Figure 11, it is evident that the LiteDenseMoE has high performance for several visually complex and fine-grained classes, including swimming_pool, where the F1score is 0.9941, shipping_yard (0.9906), and vegetable_greenhouse (0.9759), respectively. The high scores in these classes further reiterate the model's demonstrated robustness at distinguishing detailed structural patterns, which holds importance in remote sensing tasks. Similarly, classes with strong visual features, such as airplane, cloud, and island, also performed well in terms of F1-score, exceeding 0.96.

On the other hand, some categories showed relatively lower performance, such as railway station, Park, and overpass. These classes indicate difficulty in differentiating due to similar patterns. The misclassifications related to these classes can be explained by shared features with neighboring classes, such as railway, and in cases where the adjacent classes may have more relevance to the context spatially. The precision and recall across most classes also indicate that the LiteDenseMoE model has a good balance of false positives and false negatives. This directional balance is significant for remote sensing applications, especially as a class imbalance. This class imbalance shows subtle differences between classes, particularly for rare or confusing class types. Moreover, it can lead to a detection bias that many remote sensing models exhibit. The ability of the model to sustain high recall for rare or confusing classes, such as snowberg (recall 0.9717) and tennis_court (recall 0.9533), reaffirms the adaptiveness of the proposed architecture.

Table 2: Classification report of proposed architecture using MLRSNet dataset

Class Label	**Precision**	**Recall**	**F1-Score**	**Support**
airplane	0.9706	0.9594	0.9650	517
airport	0.9131	0.8920	0.9024	648
bareland	0.9303	0.9078	0.9189	412
Baseball-diamond	0.9850	0.9641	0.9744	613
Basketball-court	0.9302	0.8822	0.9055	891
beach	0.9530	0.9695	0.9612	753
bridge	0.9120	0.8930	0.9024	766
chaparral	0.9633	0.9658	0.9646	761
cloud	0.9724	0.9760	0.9742	541
Commercial-area	0.8897	0.9342	0.9114	760
Dense-residential-area	0.9577	0.9879	0.9726	825
Desert	0.9706	0.9693	0.9699	749
Eroded-farmland	0.8886	0.9120	0.9001	761
farmland	0.9519	0.9648	0.9583	739
Forest	0.9559	0.9180	0.9366	732
Freeway	0.9338	0.9531	0.9433	725
Golf-course	0.9471	0.9585	0.9528	747
Ground-track-field	0.9207	0.9232	0.9219	742
Harbor-port	0.9779	0.9580	0.9679	786
Industrial-area	0.9418	0.9239	0.9328	631
intersection	0.9479	0.9090	0.9280	780
island	0.9764	0.9713	0.9738	766
lake	0.9842	0.9120	0.9467	750
meadow	0.9289	0.9218	0.9254	780
Mobile-home-park	0.9666	0.9693	0.9679	716
Mountain	0.8960	0.8716	0.8836	771
Overpass	0.8682	0.8487	0.8584	714
Park	0.8126	0.9140	0.8603	465
Parking-lot	0.9763	0.9723	0.9743	721
Parkway	0.9266	0.9064	0.9164	780
Railway	0.8400	0.8422	0.8411	773
Railway-station	0.7628	0.7548	0.7588	673
River	0.9418	0.9051	0.9231	769
roundabout	0.8818	0.9040	0.8928	594
Shipping-yard	0.9946	0.9867	0.9906	751
Snowberg	0.9140	0.9717	0.9420	777
Sparse-residential-area	0.9719	0.9488	0.9602	547
Stadium	0.9069	0.9019	0.9044	724
Storage-tank	0.9723	0.9409	0.9563	745
Swimming-pool	0.9949	0.9933	0.9941	595
Tennis-court	0.8952	0.9533	0.9234	771
Terrace	0.9327	0.9537	0.9431	712
Transmission-tower	0.9508	0.9716	0.9611	775
Vegetable-greenhouse	0.9681	0.9838	0.9759	803
Wetland	0.8493	0.9043	0.8759	773
Wind-turbine	0.9871	0.9776	0.9823	625
Accuracy			0.9325	32749

Macro Average	0.9329	0.9327	0.9325	32749
Weighted Average	0.9332	0.9325	0.9326	32749

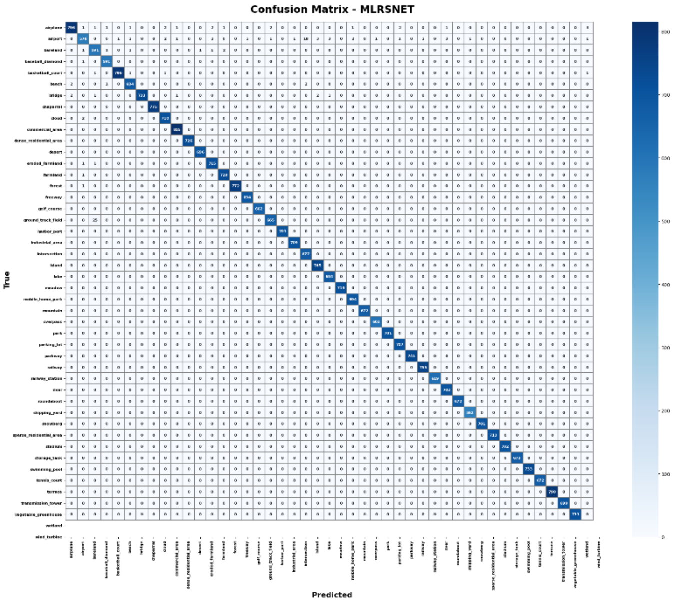

Figure 11: Confusion matrix of proposed architecture for MLRSNet dataset

B. Results on NWPU-RESISC45 Dataset

The classification results of the proposed LiteDenseMoE model on the NWPU-RESISC45 dataset have been presented in Table 3. The proposed model achieved an overall accuracy of 92.56%, as shown in this table. The macro average F1-score is 91.88% and the weighted average was 91.57%, respectively, which indicates that LiteDenseMoE has strong classification performance. From the confusion matrix in Figure 12, it is clear that LiteDenseMoE performs exceptionally well on classes that are visually distinct from one another, such as Forest (F1-score: 0.9746), Parking Space (0.9515), and Dense Residential (0.9548), where the precision and recall for several of these classes were each above 95%. Based on these values, it is noted that the proposed LiteDenseMoE framework effectively learns fine-grained relevant details of scenes, as well as differentiable spatial patterns. Classes such as Anchorage, Beach, and Farm also performed well, with F1-scores greater than 0.93 and a close to 1.0 recall value. However, some classes with lower performances, such as River (F1-score: 0.8109) and Sparse Residential (0.9104), exhibited higher misclassification rates at their respective accuracies, likely due to their high visual similarities with other natural or urban-based classes.

Table 3: Classification report of proposed architecture for NWPU dataset

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Airfield	0.8773	0.8853	0.8813	436
Anchorage	0.9352	0.9484	0.9417	213
Beach	0.9378	0.9378	0.9378	209
Dense Residential	0.9596	0.9500	0.9548	200
Farm	0.9282	0.9502	0.9391	422
Flyover	0.8905	0.9179	0.9040	195
Forest	0.9697	0.9796	0.9746	196
Game Space	0.8825	0.8949	0.8886	428
Parking Space	0.9600	0.9432	0.9515	229
River	0.7990	0.8232	0.8109	198
Sparse Residential	0.9543	0.8704	0.9104	216
Storage Cisterns	0.9594	0.9043	0.9310	209
Accuracy			**0.9256**	**3151**
Macro Avg	**0.9211**	**0.9171**	**0.9188**	**3151**
Weighted Avg	**0.9164**	**0.9156**	**0.9157**	**3151**

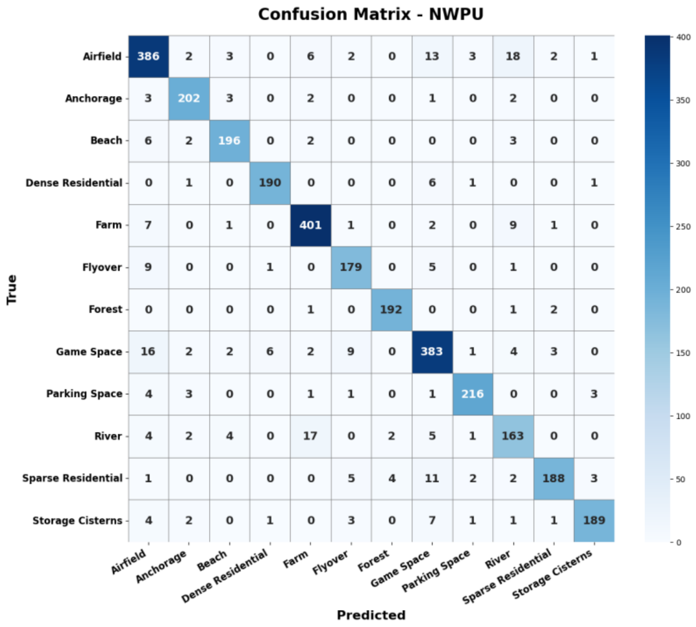

Figure 12: Confusion matrix of proposed architecture for NWPU dataset

C. Results on EuroSAT Dataset

In Table 4, the classification results of the proposed LiteDenseMoE model on the EuroSAT dataset have been presented. From this table, it is observed that the overall accuracy achieved by the model is 96.54% and macro and weighted F1-scores are 96.49% and 96.54%, respectively. For the detailed observation, the confusion matrix is shown in Figure 13. This figure indicates that the LiteDenseMoE performs exceptionally well in components such as the Residential class, achieving an F1-Score value of 0.9917, Sea Lake is 0.9929, and the Forest class is 0.9885, respectively. These classes all present sharp visual clarity and distinguishable texture patterns present in remote sensing imagery. For the more visually ambiguous classes, such as Herbaceous Vegetation (F1-score: 0.9461) and Annual Crop (0.9478), the model also provides a strong classification performance. There is a minor confusion between the Annual

Crop and Herbaceous Vegetation classes. Confusion between those specific classes is widespread due to their seasonal and spectral aspects. However, the recall and precision of Herbaceous Vegetation are close in numerical outcomes, representing acceptable output values. Moreover, the model obtains strong recall across all classes, limiting the amount of false negatives, indicating class-specific instances are far less likely to be missed entirely. This recall rate is essential for use case usability when considering real-world remote sensing problems, such as monitoring agricultural land or urban planning.

Table 4: Classification report of proposed LiteDenseMoE architecture for EuroSAT dataset

Class	Precision	Recall	F1-Score	Support
Annual Crop	0.9667	0.9296	0.9478	938
Forest	0.9885	0.9885	0.9885	873
Herbaceous Vegetation	0.9399	0.9524	0.9461	903
Highway	0.9599	0.9624	0.9612	772
Industrial	0.9728	0.9795	0.9761	730
Pasture	0.9509	0.9556	0.9532	608
Permanent Crop	0.9239	0.9492	0.9363	767
Residential	0.9912	0.9923	0.9917	904
River	0.9612	0.9485	0.9548	757
Sea Lake	0.9918	0.9941	0.9929	848
Accuracy			**0.9654**	**8100**
Macro Avg	**0.9647**	**0.9652**	**0.9649**	**8100**
Weighted Avg	**0.9656**	**0.9654**	**0.9654**	**8100**

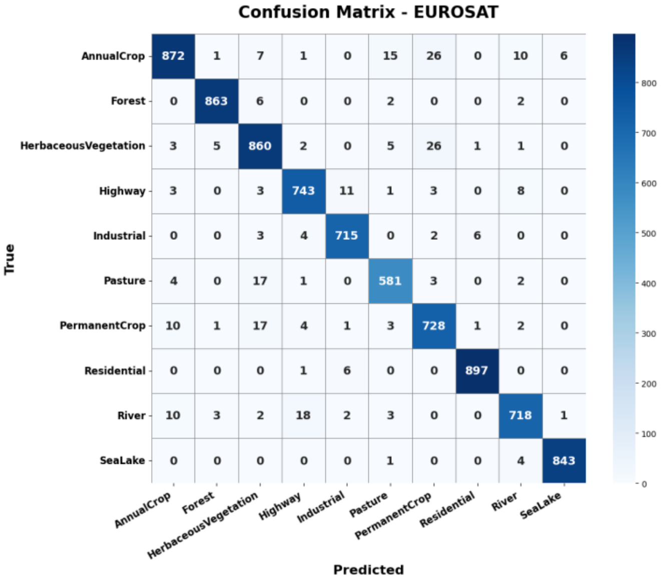

Figure 13: Confusion matrix of proposed LiteDenseMoE model using EuroSAT dataset

VI. Model Interpretation

A. Study 1

This section presents the allocation of experts for each class, which means which expert block is more suitable to handle the respective class. In Figure 14, the blue color represents Expert 1, while the orange color represents Expert 2. Each bar represents a respective class, and the majority color in that bar shows the preferred expert for that class. In the EuroSAT dataset, Expert 1 is selected for the Sea-lake, Pasture, and Forest classes, indicating that this expert is particularly efficient at extracting these types of features. For all the other classes, Expert 2 is the major choice. In the NWPU dataset, Expert 1 is allocated to 4 out of 12 classes, and Expert 2 is selected for 6 out of 12 classes. The remaining two classes demonstrate a 50-50 preference for both experts. For the MLRSNet dataset, Expert 1 is preferred for almost 14 classes out of 46. For all the other classes, Expert 2 is the primary choice. Overall, it is observed that Expert 2 is primarily selected for most classes, while Expert 1 is preferred for only a few classes.

B. Study 2

This section shows each expert's confidence level for the respective class. In Figure 15, the color chart represents the intensity of confidence level for each expert, where blue color shows the highest confidence level and light-yellow color shows the lowest confidence level. The right column represents Expert 2 while the left column represents Expert 1 in each plot. For the MLRSNet dataset, most instances in the right column display different shades of blue, indicating that Expert 2 is confident in its predictions for the respective classes. Only a few cases show a yellow color, which represents low confidence for those classes. In the left column, the primary color is light yellow, indicating that Expert 1 is not very confident in its prediction. In the NWPU dataset, the trade-off between blue and yellow colors is almost the same for both columns, indicating that Expert 1 is confident in its predictions for 50% of instances. At the same time, Expert 2 is confident for the remaining half. In the EuroSAT dataset, Expert 2 shows higher confidence for 8 out of 10 classes. In comparison, Expert 1 shows higher confidence for almost six classes, which means that for some instances, both experts are confident in their predictions.

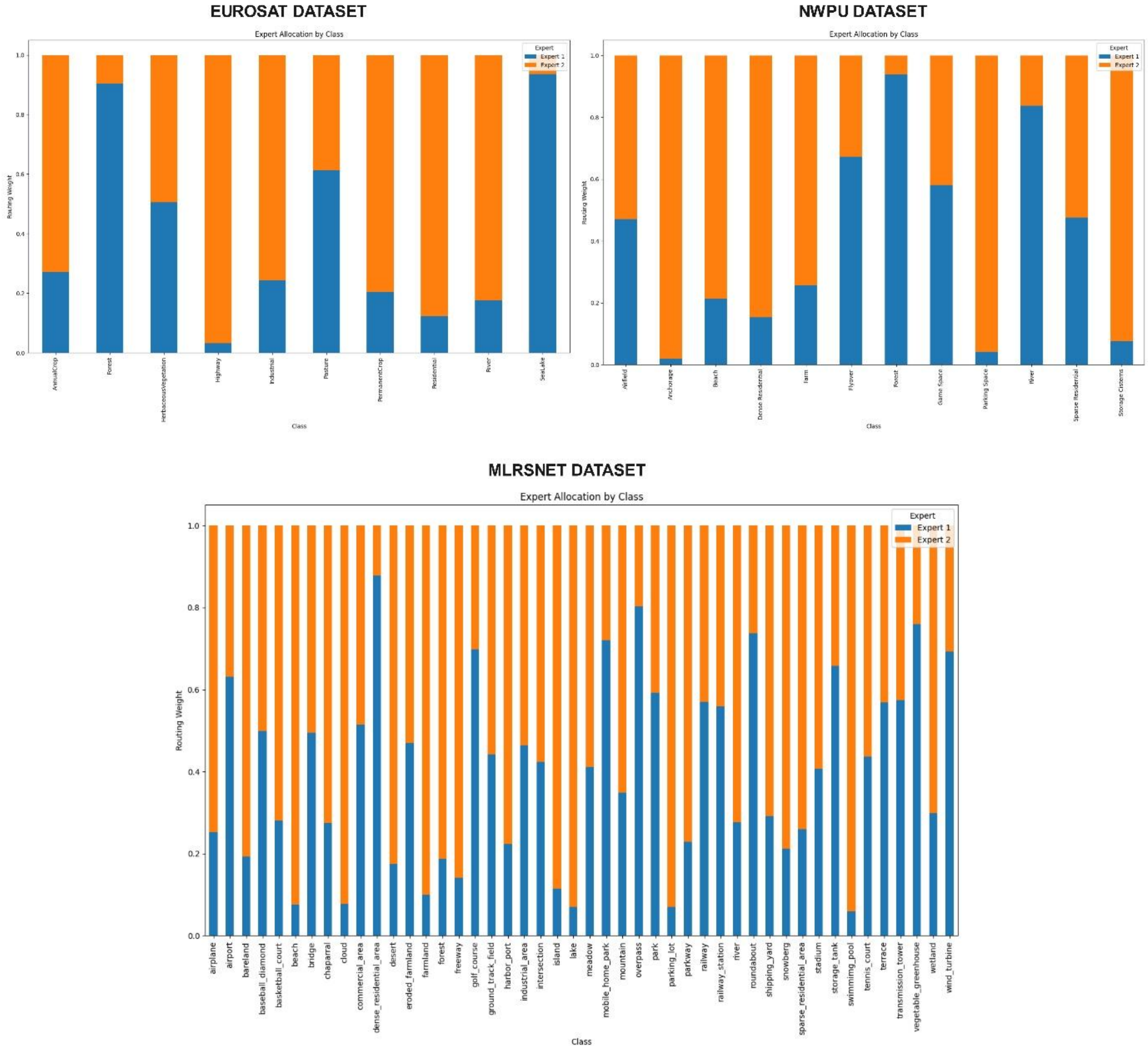

Figure 14: Expert allocation per class for all three datasets

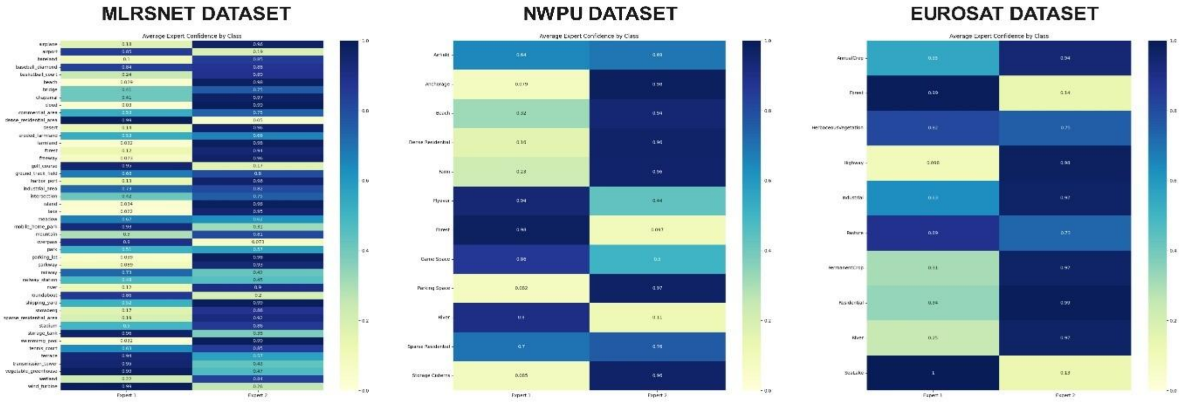

Figure 15: Expert confidence per class for all three datasets

C. Study 3

The t-SNE visualization of feature space and expert specialization in that feature space has been presented in Figure 16. The left plot in this figure shows the t-SNE visualization for each dataset, whereas the cluster of different colors denotes the dataset classes. Closely filled distinctive clusters show that the model can effectively differentiate among classes, whereas mixed points between clusters show that the model is confused among those classes. The right plot

This article has been accepted for publication in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. This is the author's version which has not been fully e
content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2025.3645113

shows the allocation of experts for that class. Here, expert one is represented by red and expert two is represented by green. If the same color shows a whole cluster, it means that the model can differentiate this class among others and assigns the same expert for that entire class. If a cluster shows both (red and green) colors, then the model misclassified some instances of that class. In this case, allocates different experts and considers them different classes.

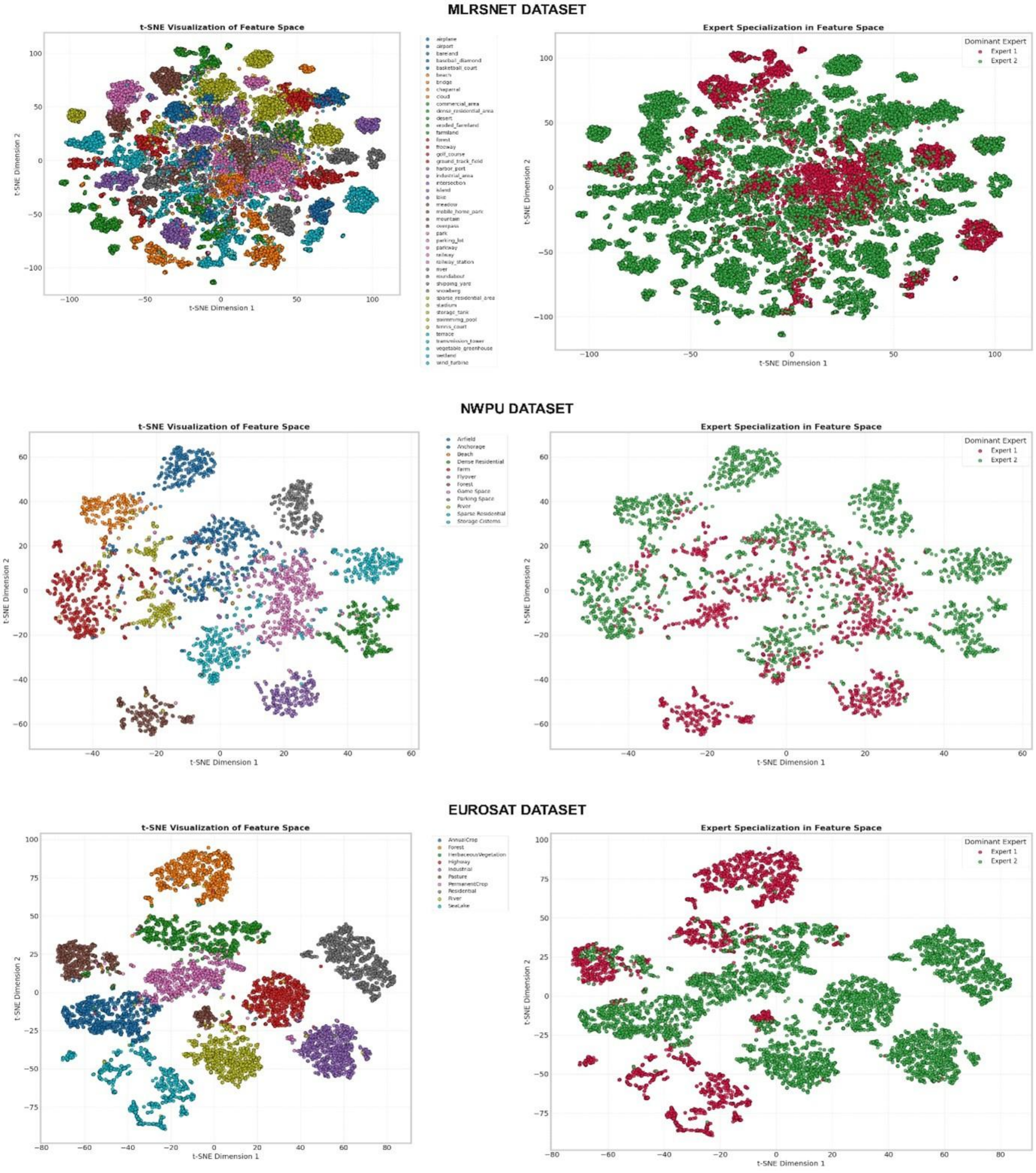

Figure 16: t-SNE visualization of feature space for all three datasets

VII. Ablation Studies

A. Experiment 1

In Table 5, an ablation study has been conducted by employing different model configurations in the architecture and evaluated on three selected datasets to verify the model's robustness and effectiveness. Using the MLRSNet dataset, it is observed that the proposed LiteDenseMoE uses both types of attention modules and achieves better accuracy than all ablation variants, reaching an accuracy of 93.25%. The proposed LiteDenseMoE, with only a spatial attention module, achieves an accuracy of 89.08%, while with only channel attention, it has an estimated accuracy of 90.09%. It is observed that the spatial and channel attention modules guide the model to focus on portions of the aerial imagery. This imagery was important in space to discriminate feature-map dependent spatial-based feature cues that our model processes. On the EuroSAT dataset, the proposed LiteDenseMoE model achieved an accuracy of 96.54%. The accuracies of 90.60% and 89.79% achieved without considering spatial attention and channel attention. However, the accuracy dropped to 86.83% with no attention modules, still demonstrating the importance of accurately emphasizing certain spatial and spectral features that distinguish fine gain classifications. These results are also shown for the NWPU dataset. The proposed architecture achieved an accuracy of 92.56%.

In comparison, the variant that only used the spatial attention module had an accuracy of 87.03%, and the other variant that only included a channel attention module had a similar accuracy of 88.16%. Again, removing both attention modules from the proposed model reduced the performance to 84.70%. This data reaffirms that spatial attention does allow the model to focus on features of interest that are more salient within that region. Moreover, the channel attention enabled the model to discover and strengthen inter-channel dependencies, which are crucial in complex coastal and land scene regions with noisy visual ambiguity.

Table 5: Ablation study one of proposed architecture for all three datasets

Architecture Configuration	Accuracy	Precision	Recall	F1-Score
MLRSNET Dataset				
With Spatial Attention	89.08	90.10	89.08	89.06
With Channel Attention	90.09	90.34	90.34	90.32
Without Spatial and Channel Attention	82.61	84.97	83.61	82.61
Proposed	**93.25%**	**93.32**	**93.25**	**92.26**
EUROSAT Dataset				
With Spatial Attention	90.60%	90.61%	90.60%	90.60%
With Channel Attention	89.79%	89.80%	89.79%	89.79%
Without Spatial and Channel Attention	86.83%	86.84%	86.83%	86.83%
Proposed	**96.54%**	**96.56%**	**96.54%**	**96.54%**
NWPU Dataset				
With Spatial Attention	87.03	87.11	87.03	87.02
With Channel Attention	88.16	88.17	88.16	88.13
Without Spatial and Channel Attention	84.70	84.73	84.70	84.69
Proposed	**92.56**	**91.64**	**91.56**	**91.57**

B. Experiment 2

This section provides a comparative analysis of the proposed LiteDenseMoE with pre-trained models. We selected top-performing pre-trained models such as AlexNet, VGG16, VGG19, GoogLeNet, ResNet50, and ResNet101 and evaluated them on all three selected datasets, as presented in Table 7. The LiteDenseMoE achieves the highest classification accuracy in all datasets, such as 93.25% on MLRSNet, 92.56% on NWPU, and 96.54% on EuroSAT, respectively. The LiteDenseMoE is also a remarkably lightweight design with only 0.3M parameters and 1.27 MB model size. The other deep models, such as VGG16, VGG19, and deep ResNet101, are on a completely different scale in terms of memory and complexity. Also, these models achieved less accuracy than LiteDenseMoE using large-scale remote sensing datasets. In the MLRSNet dataset, ResNet101 achieves an accuracy of 86.41% with 170 MB, while the LiteDenseMoE model achieves almost 7% higher accuracy with only 1.27 MB. The same pattern occurs for the NWPU and EuroSAT datasets. If we extend this comparison to relatively lightweight models like GoogLeNet, which has 6.8 million parameters and is 23 MB in size, the model's accuracy is approximately 9% less than the proposed model. This performance is a tribute not only to the parameter efficiency of LiteDenseMoE but to the architectural creativity of LiteDenseMoE and its complete integration of a number of design principles, such as dense connectivity paired with a Mixture of Experts mechanism that effectively increases representational capacity without increasing computation.

Table 6: Comparative analysis of proposed architecture with pre-trained models

Models	Accuracy	Parameters	Model Size
MLRSNET Dataset			
Alexnet	61.42	60	240 MB
VGG16	72.14	138	528 MB
VGG19	67.34	143.7	548 MB
GoogleNet	84.47	6.8	23 MB
ResNet50	85.25	25.6	102 MB
ResNet101	86.41	44.5	170 MB
Proposed Model	**93.25**	**0.3**	**1.27 MB**
NWPU Dataset			
AlexNet	71.00	60	240 MB
VGG16	74.94	138	528 MB
VGG19	74.97	143.7	548 MB
GoogLeNet	83.54	6.8	23 MB
ResNet50	88.95	25.6	102 MB
ResNet101	89.05	44.5	170 MB
Proposed Model	**92.56%**	**0.3**	**1.27 MB**
EUROSTAT Dataset			
AlexNet	80.00	60	240 MB
VGG16	83.00	138	528 MB
VGG19	83.59	143.7	548 MB
GoogLeNet	87.54	6.8	23 MB
ResNet50	88.95	25.6	102 MB
ResNet101	90.00	44.5	170 MB
Proposed Model	**96.54%**	**0.3**	**1.27 MB**

C. Experiment 3

In Figure 17, the stability of the proposed model is evaluated against the noise. The heatmap shows the effect of noise intensity on classification accuracy across the three datasets, MLRSNet, NWPHU, and EuroSAT. There is an overall decrease in accuracy as the intensity of noise increased from 0.1% to 2.0%. The overall downward trend in accuracy also indicates the sensitivity of the models to input noise. EuroSAT has the highest overall resilience across the datasets, with consistently high accuracy even at the higher noise levels (89% at 2.0%) and even higher accuracy of 94% at the lowest noise level. MLRSNet and NWPHU exhibited similar trends, with accuracy decreasing from 92% to 87% for MLRSNet and from 92% to 86% for NWPHU as the noise level increased. These results indicate that while there is some resilience to low levels of noise for all models, higher levels of noise result in more prominent degradations in performance. Therefore, the performance of the models is negatively and weakly correlated. The EuroSAT dataset is more resistant to noise than the other two datasets in all noise intervals. Additionally, these results reveal the importance of the proposed model's performance and its noise robustness in real-world applications.

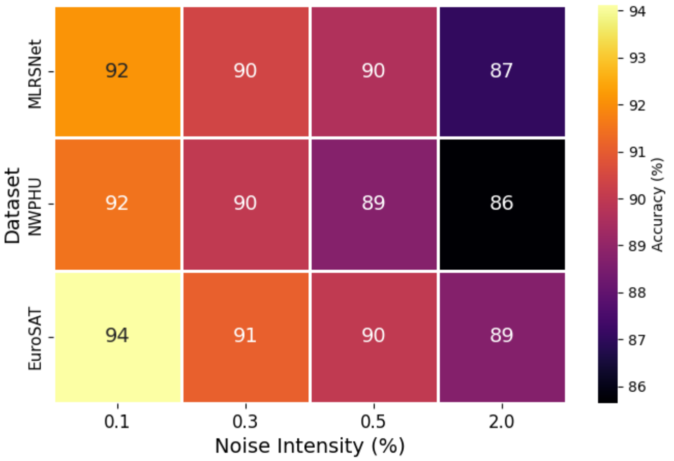

Figure 17: Heatmap to evaluate the model stability against the noisy data

Experiment 4: A critical design decision in this architecture was determining the optimal number of experts. While the conventional wisdom suggests that an increase in the number of experts will result in improved performance, our ablation study reveals a different picture. We evaluate the performance of our model with 2, 3, 4, and 6 experts across all three datasets, and the results are shown in the Table. The table shows that the proposed model achieved the highest accuracies of 93.25%, 92.56%, and 96.54% on the MLRSNet, NWPU, and EuroSAT datasets, respectively, with only two experts. The number of parameters in this configuration is also the least (30M). However, as the number of experts increases from 2, the model's performance starts to diminish, along with an increase in computational complexity. A drop in accuracy of almost 2% and nearly double the computational overhead suggest that an optimal accuracy-efficiency trade-off is possible with only two expert blocks. The reason behind these results is the overspecialization of experts, where each expert becomes overly specialized in a small subset of the training data, thereby reducing their ability to generalize effectively. Additionally, with an increased number of experts, routing mechanisms must make fine-grained decisions, which increases decision complexity and leads to suboptimal routing, ultimately degrading overall performance. These results validate our decision to select just two results for this architecture.

No. of Experts	MLRSNet Accuracy %	NWPU Accuracy %	EuroSAT Accuracy %	Parameters
2	**93.25**	**92.56**	**96.54**	**0.30M**
3	92.87	91.98	96.12	0.42M
4	92.45	91.45	95.68	0.54M
6	91.82	90.87	95.23	0.78M

Experiment 5: This study validates our choice of channel-spatial attention integrated in the architecture. Unlike natural images, RS images contain distinct spatial and spectral properties that need special attention. RS images, particularly those obtained from multispectral sensors, contain information across different wavelengths. Also in RGB representations, these channels encode essential details on land cover types. The channel attention enables the model to dynamically recalibrate channel-wise features, allowing it to learn specific spectral information for each scene type. On the other hand, Ariel scenes contain more spatially localized objects that are critical for classification. Spatial attention learns to focus on discriminative spatial regions while suppressing background information. The sequential integration of channel attention, followed by spatial attention, is based on the "what-then-where" principle, where channel attention identifies the critical feature maps, and spatial attention determines where in the image these features are most relevant. To justify this choice, we compared the proposed model with alternative attention mechanisms on the MLRSNet dataset, as shown in the table. The table shows that the proposed channel-spatial attention combination (CBAM-style) achieves superior accuracy while maintaining the computational complexity. The SE-Net with channel attention only achieves 90.34% accuracy, while spatial attention achieves 89.08% accuracy, highlighting the need for complementary attention modules. Self-attention shows good performance in terms of accuracy, but it is more computationally expensive than the proposed module. Thus, this study confirms that channel-spatial attention is the most optimal choice for this architecture.

Attention Mechanism	Accuracy	Precision	Recall	F1-score
SE-Net (channel only)	90.34%	89.94%	90.05%	88.90%
Spatial only	89.08%	87.54%	89.98%	89.00%
Self-Attention	91.45%	90.45%	89.78%	91.00%
CBAM (proposed)	**93.25%**	**93.32**	**93.25**	**92.26**

VIII. Cross-Dataset Evaluation

To evaluate the generalization ability of the proposed LiteDenseMoE beyond the same train-test data splits, we conducted a comprehensive cross-dataset evaluation where models were trained on one dataset and tested on another. The table presents the results for all six possible combinations across our three benchmark datasets. The table shows that when the model was trained on MLRSNet and tested on NWPU, the accuracy drops from 92.56% (same-dataset performance) to 84.23%. A similar trend is observed in all the combinations. The cross-dataset performance varies based on domain gap differences.

The performance drop observed on MLRSNet->EuroSAT is the lowest (7.09%), which is explained by MLRSNet's diverse spatial resolution range (0.1m-10m), including training samples similar to EuroSAT's 10m resolution. On the other hand, the most significant performance gap was observed in EuroSAT-MLRSNet (14.33%), which can be attributed to the lower spatial resolution (64x64 vs 256x256) and the limited class diversity of the EuroSAT dataset. When the model is trained on 10 classes and tested on 46 classes, it shows a greater drop than when it is trained on 46 classes and tested on 10 classes. However, despite these performance gaps, the proposed model still outperforms baseline deep learning models, such as ResNet-50 and DenseNet-121. This strong cross-dataset generalization validates that the proposed model learns fundamental transferable characteristics of RS data rather than overfitting to dataset-specific artifacts.

Train Dataset	Test Dataset	Accuracy	F1-score
MLRSNet	NWPU	84.23%	83.67%
MLRSNet	EuroSAT	89.45%	89.12%
NWPU	MLRSNet	81.34%	80.89%
NWPU	EuroSAT	88.67%	88.34%
EuroSAT	MLRSNet	78.92%	78.45%
EuroSAT	NWPU	82.56%	82.12%

IX. Statistical Significance Analysis

To ensure the statistical significance of the performance shown by LiteDenseMoE, we conducted a statistical analysis that is shown in table . We performed a 5-fold cross-validation on all three datasets, and for each fold, we trained the LiteDenseMoE from scratch using the same hyperparameters. Table shows the mean accuracies and standard deviations for all three datasets. Small std Dev of 0.43, 0.51 and 0.31 represents the robustness and stability of model across each fold. The 95% confidence intervals further provide evidence of the model's strong performance. The narrow ranges showcased the reliable and robust behavior of model under different training/testing splits.

Dataset	Mean Accuracy	Std Dev	95% CI
MLRSNet	93.25%	±0.43%	[92.82, 93.68]
NWPU	92.56%	±0.51%	[92.05, 93.07]
EuroSAT	96.54%	±0.31%	[96.23, 96.85]

X. Comparison with SOTA Models

A comprehensive comparison between the proposed model and SOTA models has been presented in Table 8. This table illustrates that the proposed model achieved the highest performance against the state-of-the-art methods. In NWPU dataset, the authors [36] employed pre-trained models with global optimal structural loss (GOSL) and they achieved 90.30% highest accuracy. In [37], the authors designed a DBOW feature unsupervised learning method, and they obtained 82.10% accuracy. Authors in [38]and [39] implemented DELF+ VLAD and IBNR-65+DenseNet64 models, and both studies achieved 85.70% and 91.70% accuracies, respectively. Similarly, On EuroSAT and MLSRNet datasets, the authors of [36], [40], [41], [41], and [42] employed pre-trained models and they obtain 88.68, 85.23, 87.52, 88.51, and 82.59% accuracies, respectively. In [43] and [44], the authors employed customized CNNs such as FMANet and AMEGRF-Net, but they gained 91.00% and 91.51% of accuracy on these selected datasets. Our proposed model achieved improved accuracy of 92.56, 96.54, and 93.25% on NWPU, EuroSAT, and MLSRNet datasets, respectively.

Table 7: Comparative analysis of proposed architecture with SOTA models

Architecture	Accuracy
NWPU Dataset	
Pretained models + GOSL [36]	90.30
DBOW feature based [37]	82.10
DELF + VLAD [38]	85.70
IBNR-65 + Densenet-64 [39]	91.70
Proposed	**92.56**
EUROSAT Dataset	
Global Optimal structured loss [36]	88.68
EfficientNet [40]	85.23
MobileNetV2 [41]	87.52
InceptionV1 [45]	88.51
Proposed	**96.54**
MLRSNet Dataset	
FMANet [43]	91.0
AMEGRF-Net [44]	91.51
MobileNetV3 + Channel Attention + Spatial pyramid pooling [42]	82.59
Proposed	**93.25%**

XI. GradCam Explainable AI (XAI) Results

The Grad-CAM visualizations in Figure 18 demonstrate that the model achieves strong alignment between prediction and relevant image regions in most cases. Correct classifications, such as airfield, dense residential, forest, and wind turbine, show clear attention to distinctive features like aircraft, housing blocks, vegetation, and turbine structures, confirming the reliability of the network's learned representations. However, some misclassifications reveal important limitations. The farm image classified as forest highlights the difficulty of separating large-scale vegetation patterns, while the flyover classified as game space indicates confusion caused by structurally complex layouts. Similarly, the beach image misclassified as cloud reflects the challenge of low-texture surfaces where discriminative cues are minimal. Overall, the results emphasize that while the model demonstrates high accuracy and interpretable feature utilization in many cases, it remains sensitive to texture similarity and structural overlap across certain categories.

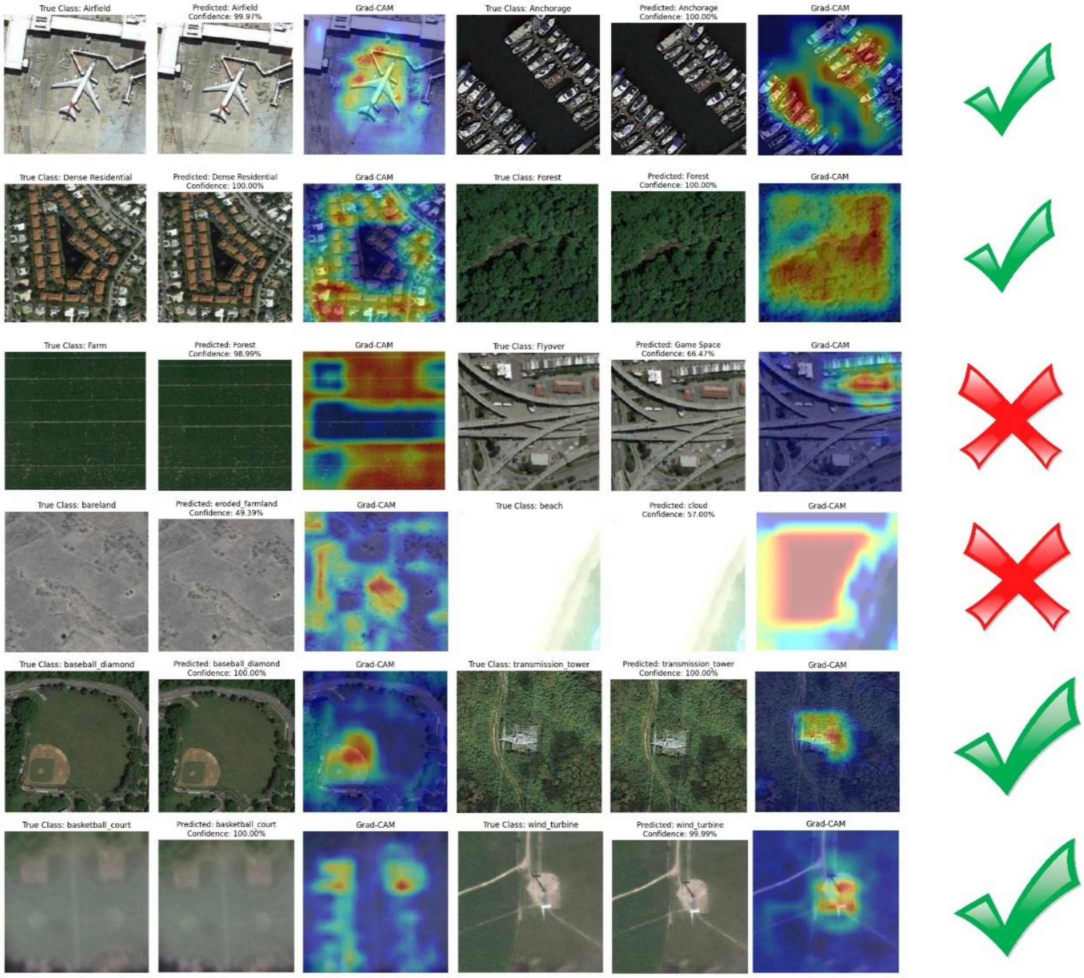

Figure 18: Explainable AI (XAI) GradCAM results

XII. Conclusion

This paper presents a novel deep learning architecture named Lightweight Dense Mixture of Experts (LiteDenseMoE) for aerial and coastal regions classification using remote sensing images. The proposed model benefits from the depth-wise separable convolutional block, incorporating both channel attention and spatial attention modules. Moreover, a two-expert MoE block functioning with an intelligent routing mechanism has been connected. Hence, the proposed model extracted the most important information of an image using the current mechanism. Hyperparameters of the proposed model during the training process are initialized through the Hyperband optimization algorithm, which improved the training efficiency and scalability. The model was systematically and rigorously evaluated using three publicly available benchmark datasets, such as MLRSNet, NWPU-RESISC45, and EuroSAT, and obtained improved accuracies of 93.25, 92.56, and 96.54% respectively, with a compact model size of 0.3 million parameters.

Comprehensive ablation studies demonstrated the impact of each component of the proposed model that contributes to the classification performance. The interpretability analysis highlighted the different expert behaviours, the confidence of the experts, and described the type of features for MoE representation. GradCAM visualization further interpreted model predictions. Even with promising results, there are still some limitations.

- The model performance could be sensitive to noise in lower-quality remote sensing data
- The performance could vary in diverse geographies and sensor modalities, as currently these datasets do not have this challenge.
- GradCAM-based interpretation shows exceptional classification abilities of the model; however, wrong predictions for a few images suggest there is room for improvement.
- It is also noted that the reliance on two expert blocks could limit the scalability into more complex scenes, which require more specialization.

Future work could involve extending the LiteDenseMoE framework to support multi-modal and multi-temporal datasets. Moreover, we will apply the scalability analysis to measure the computational performance.

CONFLICT OF INTEREST

All authors declared no conflict of interest.

DATASET AVAILABILITY

The datasets of this work are publically available for the research purposes.

XI. REFERENCES

[1] M. Li, S. Zang, B. Zhang, S. Li, and C. Wu, "A review of remote sensing image classification techniques: The role of spatio-contextual information," *European Journal of Remote Sensing,* vol. 47, no. 1, pp. 389-411, 2014.

[2] F. Gao, X. Jin, X. Zhou, J. Dong, and Q. Du, "MSFMamba: Multi-scale feature fusion state space model for multi-source remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing,* 2025.

[3] J. B. C. a. R. H. Wynne, *Introduction to remote sensing, fifth edition*.

[4] L. Zhang, M. Kong, C. Jing, and X. Xing, "CLPM: A Hybrid Network with Cross-Space Learning and Perception-Driven Mechanism for Long-Tailed Remote Sensing Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2025.

[5] A. Helen, "Seeing Nature from Above: Using Remote Sensing Data for Nature-based Decision Making."

[6] X. Zhang, Y. n. Zhou, and J. Luo, "Deep learning for processing and analysis of remote sensing big data: A technical review," *Big Earth Data,* vol. 6, no. 4, pp. 527-560, 2022.

[7] A. Y. A. Abdelmajeed and R. Juszczak, "Challenges and limitations of remote sensing applications in northern peatlands: present and future prospects," *Remote Sensing,* vol. 16, no. 3, p. 591, 2024.

[8] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine,* vol. 1, no. 2, pp. 6-36, 2013.

[9] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS journal of photogrammetry and remote sensing,* vol. 66, no. 3, pp. 247-259, 2011.

[10] M. Pal and P. M. Mather, "Decision tree based classification of remotely sensed data," in *22nd Asian conference on remote Sensing*, 2001, vol. 5, p. 9.

[11] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS journal of photogrammetry and remote sensing,* vol. 114, pp. 24-31, 2016.

[12] N. A. Mahmon and N. Ya'acob, "A review on classification of satellite image using Artificial Neural Network (ANN)," in *2014 IEEE 5th Control and system graduate research colloquium*, 2014: IEEE, pp. 153-157.

[13] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE,* vol. 104, no. 11, pp. 2207-2219, 2016.

[14] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE transactions on geoscience and remote sensing,* vol. 57, no. 9, pp. 6690-6709, 2019.

[15] C. Shi, M. Ding, and L. Wang, "Re-Parameterized Feature Aggregation Convolutional Neural Network for Remote Sensing Scene Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2025.

[16] Z. Chen, H. Yang, Q. Liu, Y. Liu, M. Zhu, and X. Liang, "Deep Learning for Hyperspectral Image Classification: A Critical Evaluation via Mutation Testing," *Remote Sensing,* vol. 16, no. 24, p. 4695, 2024.

[17] S. Rubab *et al.*, "BNResNet: Batch Normalization Inspired Deep Bottleneck Residual Architecture for Aerial Scene Recognition in Low Contrast Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2025.

[18] F. Ma, Y. Feng, F. Zhang, and Y. Zhou, "Cloud adversarial example generation for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing,* 2025.

[19] Y. He *et al.*, "SemiBaCon: Semi-Supervised Balanced Contrastive Learning for Multi-Modal Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing,* 2025.

[20] L. Bin *et al.*, "Multi-sensor remote sensing and advanced image processing for integrated assessment of geological structure and environmental dynamics," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2025.

[21] K. Ali and B. A. Johnson, "Land-use and land-cover classification in semi-arid areas from medium-resolution remote-sensing imagery: A deep learning approach," *Sensors,* vol. 22, no. 22, p. 8750, 2022.

[22] A. Diab, R. Kashef, and A. Shaker, "Deep learning for LiDAR point cloud classification in remote sensing," *Sensors,* vol. 22, no. 20, p. 7868, 2022.

[23] A. Alem and S. Kumar, "Deep learning models performance evaluations for remote sensed image classification," *Ieee Access,* vol. 10, pp. 111784-111793, 2022.

[24] M. Aljebreen, H. A. Mengash, M. Alamgeer, S. S. Alotaibi, A. S. Salama, and M. A. Hamza, "Land use and land cover classification using river formation dynamics algorithm with deep learning on remote sensing images," *IEEE Access,* vol. 12, pp. 11147-11156, 2024.

[25] M. Fayaz, J. Nam, L. M. Dang, H.-K. Song, and H. Moon, "Land-cover classification using deep learning with high-resolution remote-sensing imagery," *Applied Sciences,* vol. 14, no. 5, p. 1844, 2024.

[26] F. S. Alsubaei, A. A. Alneil, A. Mohamed, and A. Mustafa Hilal, "Block-scrambling-based encryption with deep-learning-driven remote sensing image classification," *Remote Sensing,* vol. 15, no. 4, p. 1022, 2023.

[27] M. Esmaeili, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and Q. Li, "ResMorCNN model: hyperspectral images classification using residual-injection morphological features and 3DCNN layers," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 17, pp. 219-243, 2023.

[28] H. M. Albarakati *et al.*, "A Unified Super-Resolution Framework of Remote Sensing Satellite Images Classification based on Information Fusion of Novel Deep Convolutional Neural Network Architectures," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2024.

[29] X. Li, J. Li, J. Jiang, X. Pan, and X. Huang, "Spatio-temporal-text fusion for hierarchical multi-label crop classification based on time-series remote sensing imagery," *International Journal of Applied Earth Observation and Geoinformation,* vol. 139, p. 104471, 2025.

[30] K. VanExel, S. Sherchan, and S. Liu, "Optimizing Deep Learning Models for Climate-Related Natural Disaster Detection from UAV Images and Remote Sensing Data," *Journal of Imaging,* vol. 11, no. 2, p. 32, 2025.

[31] Z. Li, D. Li, Y. Yan, Y. Zhang, and J. Wu, "MFFD: Multilayer Feature Fusion and Decision Network for Remote Sensing Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2025.

[32] *MLRSNet: A Multi-label High Spatial Resolution Remote Sensing Dataset for Semantic Scene Understanding*. [Online]. Available: https://github.com/cugbrs/MLRSNet

[33] *NWPU-RESISC45* [Online]. Available: https://figshare.com/articles/dataset/NWPU-RESISC45_Dataset_with_12_classes/16674166?file=30871912

[34] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1, no. 4, pp. 541-551, 1989.

[35] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research,* vol. 18, no. 185, pp. 1-52, 2018.

[36] P. Liu, G. Gou, X. Shan, D. Tao, and Q. Zhou, "Global optimal structured embedding learning for remote sensing image retrieval," *Sensors,* vol. 20, no. 1, p. 291, 2020.

[37] X. Tang, X. Zhang, F. Liu, and L. Jiao, "Unsupervised deep feature learning for remote sensing image retrieval," *Remote Sensing,* vol. 10, no. 8, p. 1243, 2018.

[38] R. Imbriaco, C. Sebastian, E. Bondarev, and P. H. de With, "Aggregated deep local features for remote sensing image retrieval," *Remote Sensing,* vol. 11, no. 5, p. 493, 2019.

[39] H. M. Albarakati *et al.*, "A novel deep learning architecture for agriculture land cover and land use classification from remote sensing images based on network-level fusion of self-attention architecture," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2024.

[40] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019: PMLR, pp. 6105-6114.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.

[42] X. Yang *et al.*, "An Efficient Lightweight Satellite Image Classification Model with Improved MobileNetV3," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),* 2024: IEEE, pp. 1-6.

[43] F. Rauf *et al.*, "FMANet: Super Resolution Inverted Bottleneck Fused Self-Attention Architecture for Remote Sensing Satellite Image Recognition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* 2024.

[44] Z. Li, J. Hu, K. Wu, J. Miao, and J. Wu, "Adjacent-Atrous Mechanism for Expanding Global Receptive Fields: An End-to-End Network for Multi-Attribute Scene Analysis in Remote Sensing Imagery," *IEEE Transactions on Geoscience and Remote Sensing,* 2024.

[45] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

Authors Biography

Muhammad John Abbas received the bachelor’s degree in 2025 from HITEC University, Rawalpindi, Pakistan. Currently, he is research associate at Prince Mohammad bin Fahd Univeristy, KSA under the Center of AI. He is a highly skilled data scientist and machine learning expert with a passion for remote sensing and biomedical engineering. With a strong background in computer science and mathematics, he has extensive experience in developing and deploying complex models for a variety of applications. John major expertise includes a variety of machine learning techniques such as supervised and unsupervised learning, deep learning and computer vision.

Muhammad Attique Khan (Member IEEE) received the master's and Ph.D. degrees in human activity recognition for application of video surveillance and skin lesion classification using deep learning from COMSATS University Islamabad, Islamabad, Pakistan, in 2018 and 2022, respectively. He is currently an Assistant Professor with AI Department, Prince Mohammad Bin Fahd, Al-Khobar, Saudi Arabia. His primary research focus in recent years is medical imaging, COVID-19, MRI analysis, video surveillance, human gait recognition, and agriculture plants using deep learning. He has above 350 publications that have more than 16 000+ citations and an impact factor of 1050+ with h-index 74 and i-index 230. He is the Reviewer of several reputed journals, such as the IEEE Transaction on Industrial Informatics, IEEE Transaction of Neural Networks, Pattern Recognition Letters, Multimedia Tools and Application, Computers and Electronics in Agriculture, IET Image Processing, Biomedical Signal Processing Control, IET Computer Vision, EURASIP Journal of Image and Video Processing, IEEE Access, MDPI Sensors, MDPI Electronics, MDPI Applied Sciences, MDPI Diagnostics, and MDPI Cancers.

Ameer Hamza is currently working toward the Ph.D. degree in computer science with KTU University, Kaunas, Lithuania. His major interests include object detection and recognition, video surveillance, medical, and agriculture using deep learning and machine learning. He has published 20 impact factor papers to date.

Shrooq Alsenan received the Ph.D. degree in information systems’ sciences from King Saud University, Riyadh, Saudi Arabia.,She is an academic and a researcher of artificial intelligence, and currently directs the AI Center with Princess Nourah bint Abdulrahman University, Riyadh, Saudia Arabia. She has received a prestigious postdoctoral fellowship with CSAIL and Jameel Clinic, MIT. Her research expertise spans AI in healthcare, remote sensing, bioinformatics, and hyperspectral images.

Mehrez Marzougui was born in Kasserine, Tunisia, in 1972. He received the B.Sc. degree from the University of Tunis, Tunis, Tunisia, in 1996, and the M.Sc. and Ph.D. degrees from the University of Monastir, Monastir, Tunisia, in 1998 and 2005, respectively, all in electronics. From 2001 to 2005, he was a Research Assistant with Electronics and Micro-Electronics Laboratory. From 2006 to 2012, he was an Assistant Professor with Electronics Department, University of Monastir. Since 2013, he has been an Assistant Professor with Engineering Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia. He is the author of more than 30 articles. His research interests include hardware/software cosimulation, image processing, and multiprocessor systems on chips.

Areej Alasiry received the B.Sc. degree in information systems from King Khalid University, Abha, Saudi Arabia, and the M.Sc. degree (Hons.) in advanced information systems and the Ph.D. degree in computer science and information systems from Birkbeck College, University of London, U.K., in 2010 and 2015, respectively. She is currently an Assistant Professor at the College of Computer Science, King Khalid University. She also holds the position of the College Vice Dean for Graduate Studies and Scientific Research. Her main research interests include machine learning and data science.

Jungpil Shin (Senior Member, IEEE) received the B.Sc. degree in computer science and statistics and the M.Sc. degree in computer science from Pusan National University, South Korea, in 1990 and 1994, respectively, and the Ph.D. degree in computer science and communication engineering from Kyushu University, Japan, in 1999, under a scholarship from the Japanese Government (MEXT). He was an Associate Professor, a Senior Associate Professor, and a Full Professor with the School of Computer Science and Engineering, The University of Aizu, Japan, in 1999, 2004, and 2019, respectively. He has co-authored more than 420 published papers for widely cited journals and conferences. He was included among the top 2% of scientists worldwide edition of Stanford University/Elsevier, in 2024. His research interests include pattern recognition, image processing, computer vision, machine learning, human-computer interaction, non-touch interfaces, human gesture recognition, automatic control, Parkinson’s disease diagnosis, ADHD diagnosis, user authentication, machine intelligence, bioinformatics, handwriting analysis, recognition, and synthesis. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He serves as an Editorial Board Member for Scientific Reports. He served as the general chair, the program chair, and a committee member for numerous international conferences. He serves as an Editor for IEEE journals, Springer, Sage, Taylor and Francis, Sensors (MDPI), Electronics (MDPI), and Tech Science. He serves as a reviewer for several major IEEE and SCI journals.

Yunyoung Nam (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 2001, 2003, and 2007, respectively. He was a Senior Researcher with the Center of Excellence in Ubiquitous System, Stony Brook University, Stony Brook, NY, USA, from 2007 to 2010, where he was a Postdoctoral Researcher, from 2009 to 2013. He was a Research Professor with Ajou University, from 2010 to 2011. He was a Postdoctoral Fellow with Worcester Polytechnic Institute, Worcester, MA, USA, from 2013 to 2014. He was the Director of the ICT Convergence Rehabilitation Engineering Research Center, Soonchunhyang University, from 2017 to 2020. He has been the Director of the ICT Convergence Research Center, Soonchunhyang University, since 2020, where he is currently an Assistant Professor with the Department of Computer Science and Engineering. His research interests include multimedia database, ubiquitous computing, image processing, pattern recognition, context-awareness, conflict resolution, wearable computing, intelligent video surveillance, cloud computing, biomedical signal processing, rehabilitation, and healthcare systems.