SEMSF-Net: Explainable Squeeze–Excitation Multiscale Fusion Network for Aerial Scene and Coastal Area Recognition Using Remote Sensing Images

Muhammad John Abbas, *Member, IEEE*, Muhammad Attique Khan, *Member, IEEE*, Ameer Hamza, *Member, IEEE*, Shrooq Alsenan, Areej Alasiry, Mehrez Marzougui, Yang Li, and Yunyoung Nam, *Member, IEEE*

***Abstract*—Land use and land cover (LULC) classification has played a key role over the last decade for managing the decay of resources and mitigating the impact of population growth. It is used in several places, such as rapid urbanization, agriculture, climate change, coastal areas, and disaster recovery. The traditional remote sensing (RS) techniques encounter limitations in accurately classifying dynamic and complex ariel scenes, such as coastal areas and LULC. This article proposed a novel squeeze–excitation multiscale fusion network (SEMSF-Net) to classify LULC and the coastal regions using RS images. The proposed model is based on the squeeze-and-excitation block initially embedded with inception and dense blocks separately. These blocks are designed based on the multiscale to generate more important feature information that can later perform accurate classification. In the next phase, these blocks are fused at the network level, where bottleneck and inverted residual blocks are connected to reduce the learnable parameters and improve feature strength. The hyperparameters of this network are selected based on the several experiments utilized in the training of the proposed model. The trained SEMSF-Net architecture is further employed in the testing phase, and classification is performed. The GradCAM is also used to interpret the trained model's visual prediction. Three datasets are utilized for the experimental process: the Coastal dataset, MLRSNet, and NWPU. We obtained an improved accuracy of 94.94%, 93.7%, and 95.70% on these datasets, respectively. In addition, the macro recall rates are 79.0%, 93.0%, and 96%, respectively. Comparison with several recent techniques shows that the proposed model outperforms the selected datasets.**

***Index Terms*—Aerial scene, coastal areas, deep learning (DL), explainable artificial intelligence (AI), remote sensing (RS).**

I. Introduction

HAZARDOUSLY, between 1959 and 2019, land use change contributed 19% of total anthropogenic CO_2 emissions, which emphasizes the urgency to monitor and manage earth's resources accurately [1]. Remote sensing (RS) classification of land cover and land use (LCLU) has, thus, become an essential tool for understanding and reducing environmental challenges, assisting urban planning, disaster management, and sustainable development [2]. Modern advancement in RS technologies, such as high-resolution satellite imaging, drones, and aerial platforms, has considerably enhanced data acquisition capabilities with more comprehensive datasets for analysis [3]. These rich data have sparked interest among computer vision researchers, who have further fueled innovations in automated systems designed to interpret RS imagery with increased accuracy [4]. However, high-resolution RS data cause unusual challenges in analysis due to significant variability in land covers and environmental conditions [5].

Researchers have widely applied machine learning (ML) methods to analyze RS data [6]. Still, these methods were inefficient for grappling with high-resolution images that cause great variability between different land covers and environments [7]. Although ML methods, such as SVM [8], Single decision trees (DTs) [9], [10], boosted DTs [11], Random forests [12], KNNs [13], [14], and artificial neural networks (ANNs) [15], provide fundamental tools for analyzing and processing RS data, their dependence on handcrafted features limits them from being adaptive to dynamic and unseen patterns, which results in poor

Received 17 April 2025; revised 16 May 2025; accepted 15 June 2025. Date of publication 18 June 2025; date of current version 7 July 2025. This work was supported in part by the Princess Nourah bint Abdulrahman University Researchers Supporting Project under Grant PNURSP2025R506, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, in part by the Deanship of Research and Graduate Studies at King Khalid University through the Large Research Project under Grant RGP2/471/46, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant RS-2023-00218176. This work was also supported by the Soonchunhyang University Research Fund. *(Corresponding authors: Muhammad Attique Khan; Yunyoung Nam.)*

Muhammad John Abbas and Muhammad Attique Khan are with the Center for Artificial Intelligence, Prince Mohammad bin Fahd University, Al-Khobar 34754, Saudi Arabia (e-mail: johnabbas@ieee.org; mkhan3@pmu.edu.sa).

Ameer Hamza is with the Centre of Real Time Computer Systems, Kaunas University of Technology, 44249 Kaunas, Lithuania (e-mail: ameerhamza@ieee.org).

Shrooq Alsenan is with the Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia (e-mail: shaalsenan@pnu.edu.sa).

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

Yang Li is with the College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, China (e-mail: liyang328@shzu.edu.cn).

Yunyoung Nam is with the Department of ICR Convergence, Soonchunhyang University, Asan 31538, South Korea (e-mail: ynam@sch.ac.kr).

Digital Object Identifier 10.1109/JSTARS.2025.3580801

classification performance [16]. The handcrafted features fail to capture RS imagery's complex and multiscale nature because they are task specific and not generalizable [17]. This dependence makes the models vulnerable to changes in environmental conditions, variations in lighting, or noise in the data, which are common in high-resolution RS datasets [18]. Furthermore, traditional approaches require a lot of human effort in feature engineering, which is very time consuming and may introduce bias in the classification process [19].

To overcome these limitations, deep learning (DL) has emerged as a compelling alternative to outperform traditional ML in automatically extracting meaningful patterns from large and complex datasets [20], [21]. Artificial neural networks (ANNs), an early form of DL, were initially explored for RS image analysis [22]. The ANN can master complex relationships due to their ability to deal with nonlinear data. However, they lack spatial inductive bias, which is needed to capture hierarchal and contextual relationships. As in ANNs, each input is connected to all the neurons, so they cannot preserve spatial information and treat all the regions equally, which results in poor performance due to loss of contextual information [23]. The convolutional neural network (CNN) overcomes this inability with its specialized layers: convolutional, pooling, and fully connected for extracting hierarchical features from images in RS data [24]. Convolutional layers capture spatial features by applying filters throughout the image, identifying things like edges, textures, and shapes in RS data [25]. Pooling layers refine these by reducing spatial dimensions, making the network computationally efficient while preserving critical information [26]. However, CNNs have limitations as they require a large amount of labeled data and high computational cost to train the model from scratch. It is often unavailable in the RS domain and suffers scalability issues [27], [28]. The challenges have been overcome by pretrained CNN models, such as VGG16, VGG19 [29], AlexNet [30], ResNet [31], and EfficientNet [32], which support transfer learning to adapt prelearned weights for RS applications. Still, they have limited adaptability to the unique complexities of RS data, such as overlapping class patterns and high intraclass variability [33].

Several techniques are introduced in the literature for classifying land use and land cover (LULC) and coastal areas using RS images [34]. They used DL techniques and trained the models on well-recognized publically available datasets. Rauf et al. [35] have recently presented a novel architecture exclusively designed for RS image classification called FMANet. The developed model utilizes a tailored superresolution network intended to improve the resolution of RS images as a first step. After that, they designed a fused bottleneck self-attention mechanism for deep feature extraction. In the training process, high-resolution RS images are utilized, which are further optimized using Bayesian optimization for hyperparameter tuning. This tuning ensures that the model is well suited for diverse data characteristics. The architecture was accurately tested on three datasets: MLRSNet, Bijie Landslide, and Turkey Earthquake, and it obtained improved accuracies of 91.0, 92.8, and 99.4, respectively. Li et al. [36] presented a DL model named AMEGRF-Net, which is especially envisioned to surpass conventional architectures such as VGG16 and ResNet50 over RS tasks. This local–global feature learning methodology of AMEGRF-Net captures both spatial and semantic information from the RS imagery simultaneously. In addition, this process improves the modeling of the complex patterns in RS imagery. The second innovative aspect of this work is its receptive field expansion method, by which the network captures spatial information without significantly increasing computational overhead. Yang et al. [37] presented an efficient and lightweight satellite image classification model targeting the processing challenges of large RS images onboard satellites. They improved the working of MobileNetV3 architecture to reduce the computational and communication overhead from transmitting large amounts of image datasets to the ground stations. This improved model utilizes depth separable convolution, which reduces the number of parameters and computation cost to a large extent. An inverted residual linear structure is further added to the network that maintains better accuracy and reduces the parameters. Shah et al. [38] introduced a superresolution-based fuzzy DL architecture to classify aerial RS data related to land cover and landsliding. The presented model addresses the challenges of noise and interference in RS images while improving classification accuracy through an innovative combination of components. The architecture combines an optimistic activation function to enhance nonlinear transformations and a depth separable convolutional layer to reduce model complexity. Moreover, the inverted bottleneck block further optimizes the model by retaining the most critical spatial information and reducing redundancy. The model has a superresolution preprocessing step that improves the resolution of the input data, allowing meaningful features to be extracted. They used Bijie Earth, EuroSAT, and NWPU-RESISC45 datasets for the experimental process and obtained improved accuracy. Wang et al. [39] introduced a weakly supervised scale adaptive data augmentation network (WSADAN) to handle scene classification tasks concerning high-resolution RS images. The model deals with the problem of extracting robust multiscale features. For this challenge, the WSADAN incorporates a scale generation module that learns optimal scale parameters dynamically. This module enhances the model's ability to adapt to variations in image scale without relying on exhaustive manual tuning. In addition to the scale generation module, the WSADAN includes a fusion module that intelligently filters, and merges feature from multiple scales, further improving the robustness and accuracy of the classification process. Albarakati et al. [40] designed a self-attention fused CNN that is tailored for the classification of LCLU in RS data. The authors have addressed the strengthening of essential features in the high-dimensional RS imagery dataset, which often suffers from challenges such as class overlap, noise, and high variability. The architecture relies on two custom-built networks, namely, IBNR-65 and Densenet-64, with the former being optimized to compute efficiently with feature refinement and the latter for deep feature learning with densely connected layers. A self-attention mechanism was incorporated into the CNN framework to enhance its capacity to focus on more significant regions within the image. Shree and Meiaraj [28] evaluated two ML algorithms: SVM and RF for LULC classification for change detection based on RS data from 1993 to

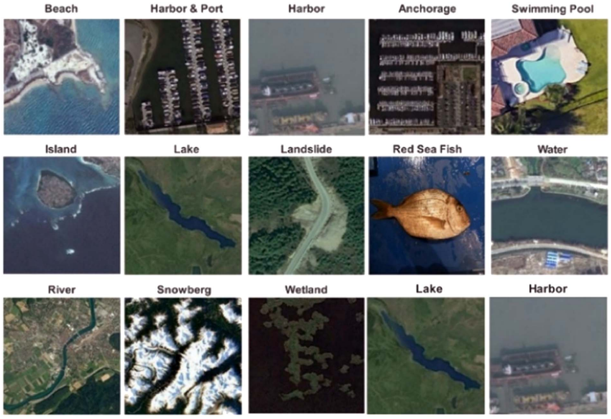

Fig. 1. Sample Coastal areas RS images collected from several publically available datasets.

2023. LULC features, such as water bodies, built-up land, barren land, agricultural land, grassland, etc., are categorized by NRSC level 1 classification, and experimental findings reveal that RF outperforms the SVM algorithm by achieving an accuracy of 0.92, which is higher than 0.81 obtained by SVM. Moreover, study shows that from 1993 to 2023, water bodies, agricultural land, and grassland decrease in area, while the built-up land, mining land, and barren land increased by area. Pushpalatha et al. [41] suggested a CNN-based approach for LULC classification using RS images obtained from linear imaging self-scanning sensor III. The objective of study was LULC classification for change detection using maps of years 2010 and 2020. Experimental results demonstrate that the proposed model obtained an overall accuracy of 94.08% and 95.30% for 2010 and 2020 data, respectively, and further analysis shows that an increase of 8.34 km^2 in built-up areas, 2.21 km^2 in agricultural land, and 3.31 km^2 in water bodies is detected, followed by a decrease of 1.49 km^2 in forest areas and 11.93 km^2 in other land sources over a period of ten years. However, the authors agreed on the point that medium resolution of data used in this study leads to high computational cost as well as limits the accuracy of model, which can be enhanced by using the high-resolution data [41].

In short, literature review highlights the most common and recent techniques used for RS classification, including traditional ML methods (SVM, KNN, and Random Forest), pretrained models (VGG16, VGG19, EfficientNet, AlexNet, and ResNet), and custom DL techniques (WSADAN, AMEGRF-Net, and FMANet), which utilize attention modules, multiscale feature fusion, and receptive field expansion, in order to improve the performance of RS image classification; however, they are still facing the issue of imbalanced datasets, important deep feature extraction, model generalizability, and overfitting. Moreover, they did not consider any work related to coastal areas' recognition, as the images of coastal areas include similar features, as shown in Fig. 1. To address these issues, we propose a novel DL architecture called squeeze–excitation multiscale fusion network (SEMSF-Net) for classifying LULC regions with improved accuracy and precision. The proposed SEMSF-Net integrates squeeze-and-excitation (SE) mechanisms throughout its structure to improve channelwise feature recalibration, thus boosting the representational power of the network. The proposed architecture is a fusion of different blocks—Residual, Bottleneck, Inverted Bottleneck, Inception, and Dense blocks—each of which has been carefully integrated to exploit the strength of each block in the process of multiscale feature extraction and hierarchical representation learning. By embedding SE blocks in these components, the SEMSF-Net gains better adaptability to spatial and contextual information, so accuracy and robustness improve for LULC classification tasks. This innovative design solves the problem of complex spatial patterns and varying scales of LULC datasets in an integrated manner, giving a holistic solution for proper and efficient classification. Our major contributions in this work are as follows:

1) Existing models are less accurate due to poor feature representation and often lack generalization. To overcome this, a new SEMSF-Net is proposed for classifying LULC and coastal areas with an improved precision rate.
2) Deep models often struggle with vanishing gradient problem and lack better hierarchal feature extraction. To address this, a residual block based on four convolutional layers is designed, where each layer has the same filter size of 3 × 3.
3) High computational cost is a serious problem while designing deep models. It is solved by designing an inverted residual, dense, and inception blocks and embedded all of them in a single block for the lesser learning parameters and better precision.
4) Prior techniques lack interpretability, which makes it difficult to trust the model's prediction. To tackle this problem, the proposed SEMSF-Net is further evaluated on explainable artificial intelligence (AI) techniques for in-depth evaluation and visual output of the trained model.

II. Dataset Description

In this work, we utilized three datasets for the evaluation of the proposed architecture: MLRSNet dataset [42], NWPU-RESISC45 [43], and coastal areas combined dataset.

A. *MLRSNet Dataset*

The MLRSNet dataset consists of 109 161 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 10 to 0.1 m. Each image is tagged using predefined 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 Fig. 2.

B. *NWPU-RESISC45 Dataset*

The NWPU-RESISC45 dataset is composed of 10 500 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

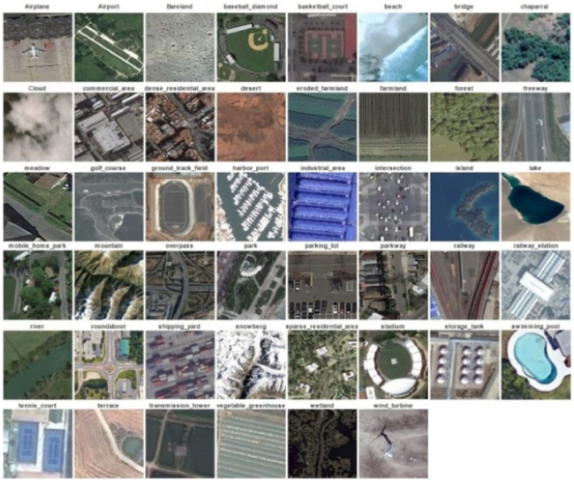

Fig. 2. Sample images of the MLRSNet dataset [42].

Fig. 3. Sample images of the NWPU-RESISC45 dataset [43].

pixel size of 256 × 256 × 3 with dpi of 96 × 96. Sample images of the NWPU-RESISC45 dataset are shown in Fig. 3.

C. 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 Fig. 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 Architecture Overview

The proposed architecture for the classification of LULC and coastal areas through DL is presented in this section. The CNN is a type of DL that first introduced by LeNet by Yann LeCun, specifically for structured data such as images and gave promising results in image classification, recognition, segmentation, and retrieval [44]. However, there were certain limitations in the initially proposed CNN, such as vanishing gradient problem, degradation, high computational cost, overfitting, exploding gradient problem, etc. [45]. Beyond its individual challenges of DL models, as it is natural that vanishing gradients, overfitting, and computational challenges are relevant problems across deep architectures, we are driven by challenges accorded to the RS image classification domain, specifically as it relates to coastal and land use scenes, which involve visual similarities among classes, e.g., water versus wetland, interclass variability associated with seasonality or change in geography, and spatial complexity related to heterogeneous landscape features. To add to these issues, most of the existing models do not generalize across multiscale patterns and struggle to learn significant salient spatial–contextual relations to high-resolution imagery. To address these domain-specific issues, we proposed an architecture named *SEMSF-Net*, as shown in Fig. 4, for RS image classification and further interpreted through the explainable AI technique.

The proposed architecture is a combination of different variants of the CNN, where each variant has its own advantage over others. It incorporates residual block to solve the vanishing gradient problem, inception blocks for multiscale feature extraction, dense blocks to promote feature reuse and gradient flow, bottleneck and inverted bottleneck blocks to balance the computational cost and SE blocks to emphasize the important regions. Each block is incorporated more than one time in the architecture which enables the hierarchal feature extraction, where initial blocks extract low-level features and later blocks extract high-level complex features, resulting in overall better feature representation. Thus, each block in the proposed architecture addresses one or more challenges in the existing models, and together they make a new model that is not only deep and accurate but also generalized and lightweight.

A. Proposed SEMSF-Net

The proposed SEMSF-Net model is presented in detail here. As shown in Fig. 4, the model starts with a 2-D convolutional layer with 64 filters, 7 × 7 filter size, stride of 2, and same padding. A 3-D input tensor of size 224 × 224 × 3 is fed to this convolutional layer, which is followed by a batch normalization layer. After that, a ReLU activation function is applied, and finally, a MaxPooling layer with a pool size of 3 × 3 and stride of 2 is applied. Then, a residual block has been added.

Originally, the residual blocks were introduced in 2015 by researchers of Microsoft [46], and the main idea of residual blocks is the integration of a shortcut connection that skips one or more layers and solves the problem of vanishing gradient and degradation [47]. In the proposed architecture, two residual blocks are added one after other, each having 64 filters. In each residual block, input tensor is fed to a 2-D convolutional layer for feature extraction followed by a batch normalization layer, which normalizes the input to the next layer. The ReLU activation function is applied to introduce nonlinearity in the model. Mathematically, it is formulated as follows:

$$c_1 = \sigma(\mathrm{BN}\,(W_1 * z) \tag{1}$$

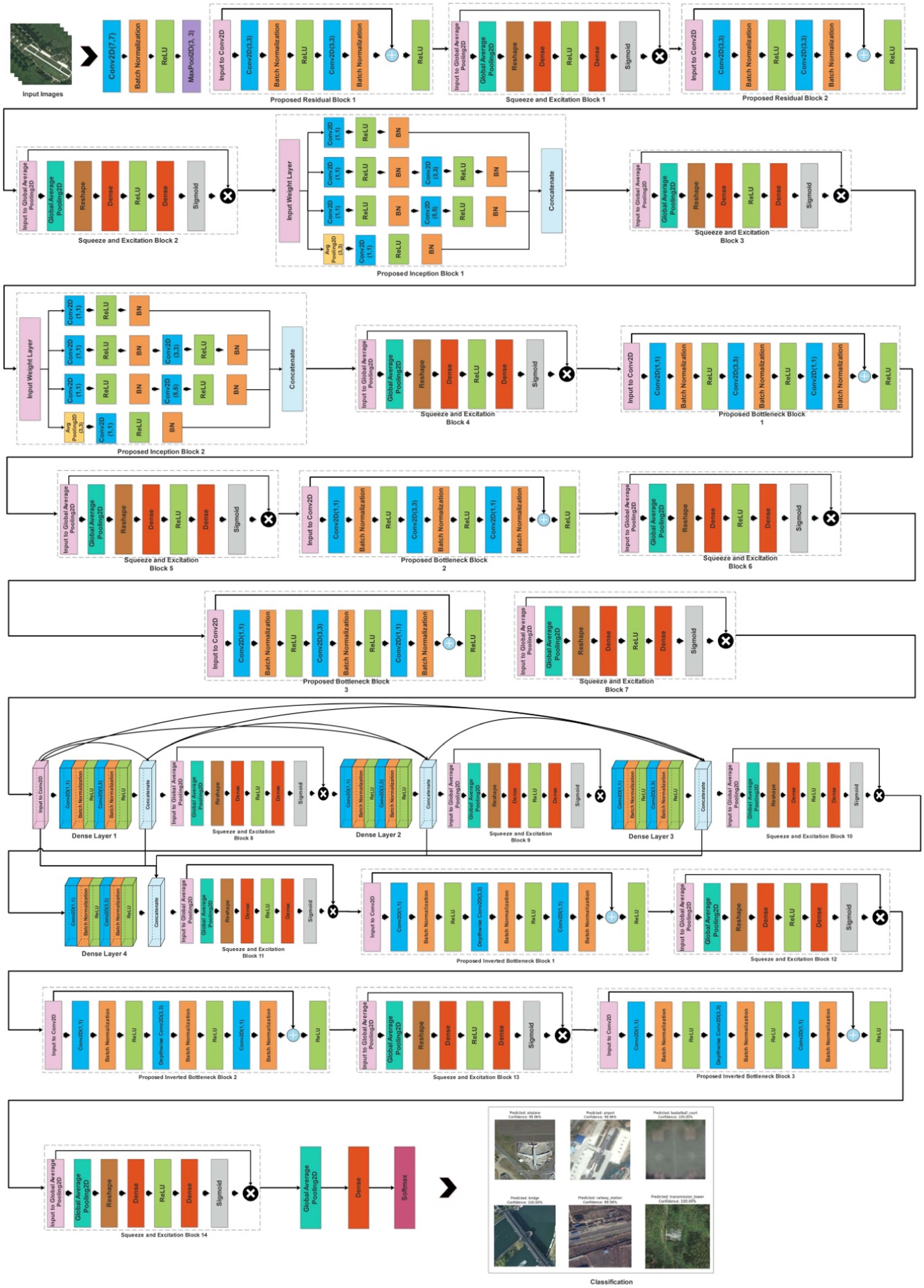
Input Images
Conv2D(7,7)
Batch Normalization
ReLU
MaxPool2D(3, 3)
Input to Conv2D
Conv2D(3,3)
Batch Normalization
ReLU
Conv2D(3,3)
Batch Normalization
ReLU
Proposed Residual Block 1
Input to Global Average Pooling2D
Global Average Pooling2D
Reshape
Dense
ReLU
Dense
Sigmoid
Squeeze and Excitation Block 1
Proposed Residual Block 2
Squeeze and Excitation Block 2
Input Weight Layer
Conv2D (1,1)
ReLU
BN
Conv2D (3,3)
Conv2D (5,5)
Avg Pooling2D (3,3)
Concatenate
Proposed Inception Block 1
Squeeze and Excitation Block 3
Proposed Inception Block 2
Squeeze and Excitation Block 4
Conv2D(1,1)
Proposed Bottleneck Block 1
Squeeze and Excitation Block 5
Proposed Bottleneck Block 2
Squeeze and Excitation Block 6
Proposed Bottleneck Block 3
Squeeze and Excitation Block 7
Dense Layer 1
Squeeze and Excitation Block 8
Dense Layer 2
Squeeze and Excitation Block 9
Dense Layer 3
Squeeze and Excitation Block 10
Dense Layer 4
Squeeze and Excitation Block 11
Depthwise Conv2D(3,3)
Proposed Inverted Bottleneck Block 1
Squeeze and Excitation Block 12
Proposed Inverted Bottleneck Block 2
Squeeze and Excitation Block 13
Proposed Inverted Bottleneck Block 3
Squeeze and Excitation Block 14
Global Average Pooling2D
Dense
Softmax
Predicted: airplane Confidence: 99.96%
Predicted: airport Confidence: 99.96%
Predicted: basketball_court Confidence: 100.00%
Predicted: bridge Confidence: 100.00%
Predicted: railway_station Confidence: 99.56%
Predicted: transmission_tower Confidence: 100.00%
Classification

Fig. 4. Proposed SEMSF-Net architecture for LULC and coastal area classification.

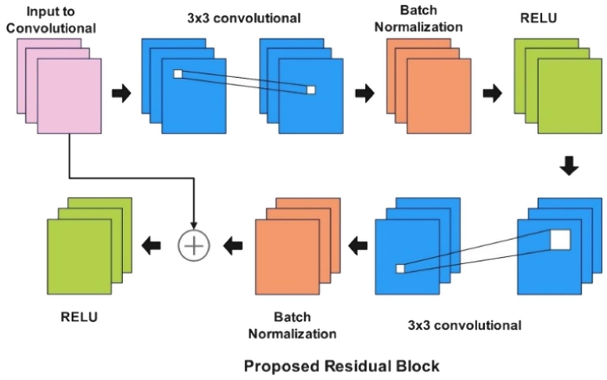

Fig. 5. Designed residual block for the proposed SEMSF-Net architecture.

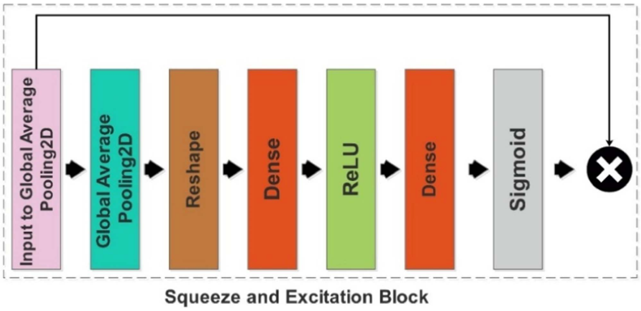
Input to Global Average Pooling2D
Global Average Pooling2D
Reshape
Dense
ReLU
Dense
Sigmoid
Squeeze and Excitation Block

Fig. 6. SE block used in the proposed SEMSF-Net architecture.

where z is an input of this block, W_1 denotes the weights of first convolutional layer, BN denotes the batch normalization layer, σ represents the ReLU activation function, $*$ represents the convolutional operation, and c_1 is the output of the first convolutional layer. After that, input is passed through the second convolutional layer with same filters again followed by batch normalization layer. Mathematically, BN is defined as follows:

$$c_2 = \text{BN}\left(W_2 * c_1\right) \tag{2}$$

where W_2 represents weights of the second convolutional layer. A shortcut connection is added to the output, and the ReLU activation function is applied to it. Moreover, a projection shortcut is used by employing the following mathematical formulation:

$$z_s = \text{BN}\left(W_p * z\right) \tag{3}$$

where z_s denotes the shortcut connection, and W_p denotes weights of the projection layer. The final output of the residual block is defined as follows:

$$Y = \sigma\left(c_2 + z_s\right). \tag{4}$$

Fig. 5 represents the visual representation of designed residual block for the proposed architecture.

After the residual block, an SE network-based block is added. The SE block was introduced by Hu et al. [48], and the main idea behind this is to recalibrate the interdependencies between channels. The SE block consists of a global average pooling layer that converts feature maps into one scalar value and two dense layers, where the first layer reduces the number of channels, and the second restores the original number of channels. The output from the residual block is first passed through the global average pooling layer to perform a squeeze operation through which a scalar value is calculated for each channel. After that, the squeezed vector is reshaped to (1, 1, C), where C is the number of channels. Two dense layers are applied to this reshaped vector in the next step. The first dense layer reduces the number of channels by 16. The ReLU activation function is applied to introduce nonlinearity, and he_normal, selected as kernel initializer, initializes the weights for a deeper network. The second dense layer restores the original number of channels, where a sigmoid function is applied, normalizing the values in the range of [0,1] to compute channelwise importance. Finally, the input tensor is multiplied by channelwise importance. Mathematically, this block is defined by the following equation:

$$S = \text{GAP}\left(Y\right) = \frac{1}{h \times w} \sum_{m=1}^{h} \sum_{n=1}^{w} Y\left(m, n, i\right) \tag{5}$$

$$E = \sigma\left(W_2''\left(\sigma'\left(W_1''S\right)\right)\right) \tag{6}$$

where h and w represent the height and width of the feature map, respectively, and i represents the index of channel. For the excitation block, σ and σ' represent the sigmoid and ReLU activation functions, respectively. Also, W_1'' and W_2'' represent the weights of dense layers. Hence,

$$Y_{\text{SE}} = Y.E. \tag{7}$$

After the implementation of the SE block, the final output of the residual block is formulated as follows:

$$Y_f = \sigma\left(c_2 + z_s\right).E. \tag{8}$$

The output Y_f enhances feature extraction and solves the problem of vanishing gradient. Visually, the SE block is shown in Fig. 6.

An optimized inception block, which includes several parallel layers, is embedded after the SE block. The inception network was initially proposed in 2014 by Google researchers in a paper titled "Going Deeper with Convolutions," and the focus behind this architecture is to increase the precision, accuracy, and speed of the model [49], [50]. An inception module consists of parallel layers with different kernel sizes, such as 1 × 1, 3 × 3, and 5 × 5, to extract features at multiple scales. In the proposed architecture, two inception blocks are added after the residual blocks, each having 64 filters and composed of four parallel branches, as shown in Fig. 7. The first branch consists of a 1 × 1 convolutional layer with the same padding to extract the localized features and reduce spatial dimensions. The ReLU activation function is applied to introduce nonlinearity, followed by a batch normalization layer to normalize the inputs to the next layer

$$z_{1\times1} = \text{BN}\left(\sigma\left(\text{conv}_{1\times1}\left(z\right)\right)\right). \tag{9}$$

The second branch is composed of a 1 × 1 convolutional layer followed by a 3 × 3 convolutional layer to extract the features at larger receptive field. The ReLU activation function followed

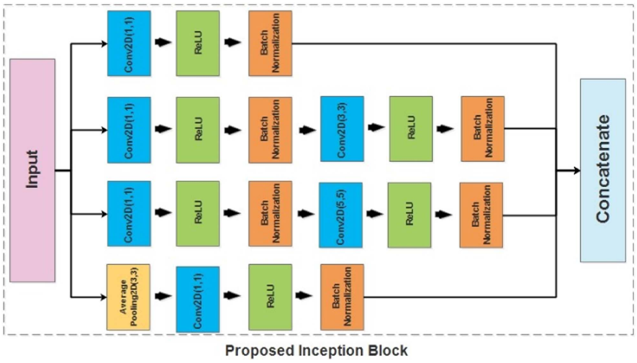

Fig. 7. Inception block embedded with the SE block.

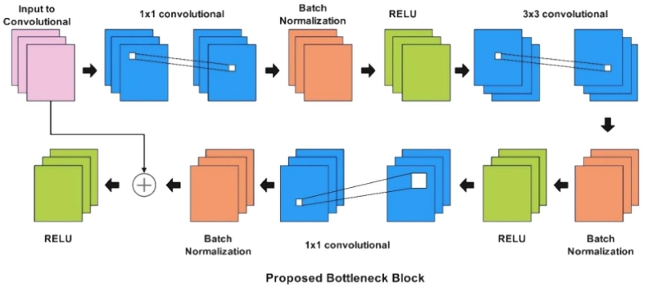

Fig. 8. Bottleneck residual block embedded with the SE block in the proposed architecture.

by the batch normalization layer is applied to both convolutional layers

$$z_{3\times 3} = \mathrm{BN}\left(\sigma\left(\mathrm{conv}_{3\times 3}\left(BN\left(\sigma\left(\mathrm{conv}_{1\times 1}\left(z\right)\right)\right)\right)\right)\right). \tag{10}$$

The third branch also consists of two convolutional layers with filter sizes of 1 × 1 and 5 × 5, respectively, to extract features at an even larger receptive field

$$z_{5\times 5} = \mathrm{BN}\left(\sigma\left(\mathrm{conv}_{5\times 5}\left(\mathrm{BN}\left(\sigma\left(\mathrm{conv}_{1\times 1}\left(z\right)\right)\right)\right)\right)\right). \tag{11}$$

The last branch consists of an average pooling layer with a pool size of 3 × 3 and same padding, followed by a 1 × 1 convolutional layer and a batch normalization layer

$$z_p = \mathrm{BN}\left(\sigma\left(\mathrm{conv}_{1\times 1}\left(\mathrm{pool}_{3\times 3}\left(z\right)\right)\right)\right). \tag{12}$$

Outputs of all branches are concatenated and fed to the SE block to enhance the feature representation

$$z_c = \mathrm{concat}\left(\left[z_{1\times 1},\ z_{3\times 3},\ z_{5\times 5},\ z_p\right]\right). \tag{13}$$

The same operations are performed again in the SE block; mathematically, it is defined as follows:

$$X = \frac{1}{h \times w}\sum_{m=1}^{h}\sum_{n=1}^{w} z_c\left(m,n,i\right) \tag{14}$$

$$X' = \sigma\left(W_2''\left(\sigma'\left(W_1''X\right)\right)\right) \tag{15}$$

$$Z_{\mathrm{SE}} = z_c.\, X'. \tag{16}$$

Hence, the final output of the inception block after passing through the SE block is formulated as follows:

$$Z_f = Z_{\mathrm{SE}}. \tag{17}$$

An addition of a bottleneck residual block [51] is essential at this stage to reduce the computational cost of the model; therefore, we designed three bottleneck blocks and integrated them with the inception excitation blocks, as shown in Fig. 8. In this block, each convolutional layer filter size is 128. However, the first has a stride of 2 to reduce spatial dimensions, while the other two blocks have a stride of 1 to enhance feature extraction. In each bottleneck block, the input tensor is fed to a 1 × 1 convolutional layer with the same padding, which reduces the number of channels by a factor of four. This compression layer reduces the spatial dimensions to reduce the computational cost, which is then followed by a batch normalization layer and a ReLU activation function. Mathematically, it is defined as follows:

$$z_1 = \sigma\left(\mathrm{BN}\left(\mathrm{conv}_{1\times 1}\left(z,\ f_{\mathrm{in}}/4\right)\right)\right). \tag{18}$$

After that, a second convolutional layer with a 3 × 3 filter size is applied for feature extraction, which is followed by the batch normalization layer and the ReLU activation function

$$z_2 = \sigma\left(\mathrm{BN}\left(\mathrm{conv}_{3\times 3},\ \left(z_1,\ f_{\mathrm{in}}/4\right)\right)\right). \tag{19}$$

Another 1 × 1 convolutional layer is applied to restore the feature dimensions. Following this, a batch normalization layer is employed to normalize the inputs to the next layer

$$z_3 = \mathrm{BN}\left(\mathrm{conv}_{1\times 1}\left(z_2,\ f_{\mathrm{out}}\right)\right). \tag{20}$$

Finally, a residual connection is added to the output of the last convolutional layer for proper gradient flow

$$z_s = \mathrm{BN}(\mathrm{conv}_{1\times 1}\left(z,\ f_{\mathrm{out}},\ s\right) \tag{21}$$

$$z_s = z \tag{22}$$

$$z_r = \sigma\left(z_s + z_3\right). \tag{23}$$

The output of this block z_r is passed to the SE block for enhanced feature representation. Mathematically, it is defined as follows:

$$X_{\mathrm{SE}} = z_r.\, X' \tag{24}$$

where X' represents weights calculated by the SE block. Hence, the final output of the residual bottleneck block after passing through the SE block is as follows:

$$X_f = X_{\mathrm{SE}}. \tag{25}$$

We added dense blocks connected to inverted bottleneck residual blocks to feed this network forward. The dense blocks are derived from the DenseNet architecture introduced in 2017 by Huang et al. [52]. In dense blocks, each layer is connected directly to all the subsequent layers, preserving the feedforward nature. Unlike ResNet, features are combined by concatenation, which solves the problem of vanishing gradient at less computational cost and increases the system's efficiency [52], [53]. In the proposed architecture, a dense block is incorporated after the bottleneck residual block to enhance parameter efficiency

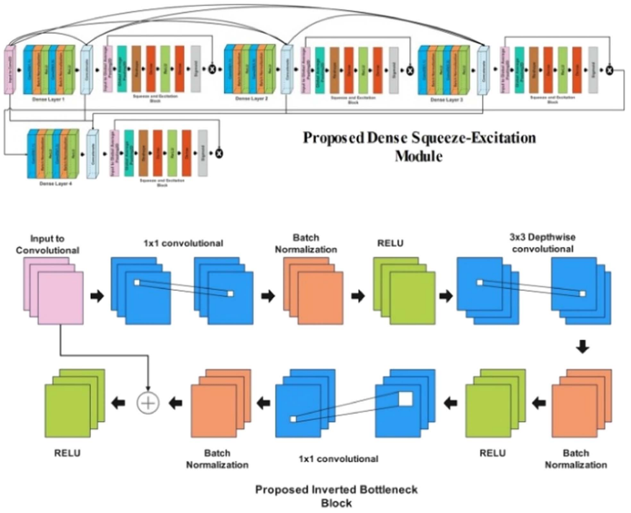
Dense Layer 1
Squeeze and Excitation Block
Dense Layer 2
Squeeze and Excitation Block
Dense Layer 3
Squeeze and Excitation Block
Dense Layer 4
Squeeze and Excitation Block
Proposed Dense Squeeze-Excitation Module
Input to Convolutional
1x1 convolutional
Batch Normalization
RELU
3x3 Depthwise convolutional
Batch Normalization
RELU
1x1 convolutional
Batch Normalization
RELU
Proposed Inverted Bottleneck Block

Fig. 9. Dense block embedded with SE and inverted bottleneck for the proposed architecture.

by encouraging feature reuse through dense connections, as shown in Fig. 9. The designed dense block comprises four layers, each consisting of a 1 × 1 convolutional layer, a 3 × 3 convolutional layer, a concatenation operation, and an SE block. Input tensor first passed through a 1 × 1 convolutional layer, in which the growth factor, originally 32, was multiplied by a factor of four. This intermediate expansion ensures enhanced feature representation while maintaining the computational cost. The output of the first convolutional block is passed to a 3 × 3 convolutional layer for feature extraction. The growth rate of this layer is the same as the original. After that, a concatenation layer is applied, which merges the features of the current layer with subsequent layers as follows:

$$z_{j+1} = \text{concat}\left(z_0,\ z_1,\ \ldots, z_j,\ f_3\left(f_1\left(z_j\right)\right)\right). \tag{26}$$

Finally, an SE block is applied, which enhances the feature representation; hence, the total number of features after the dense block is defined by the following equation:

$$F_{\text{out}} = F_0 + L.m \tag{27}$$

where L denotes the number of layers, m is the growth factor, and F is the number of extracted features on this block.

Moreover, three inverted bottleneck blocks are incorporated, where first block has a stride of 2, while the other two blocks have stride of 1 to ensure the effective feature extraction while maintaining computational cost. Input tensor first passed through the 1 × 1 convolutional layer, in which the number of filters, originally 256, expanded by the factor of four. This intermediate expansion offers rich feature representation while maintaining computational cost. After that, a depthwise separable 3 × 3 convolutional layer is employed for feature extraction. The depthwise convolutional layer is much better in terms of parameters as compared with the standard convolutional layer. Again, a batch normalization layer and an activation layer are applied as follows:

$$z'' = \sigma\left(\text{BN}\left(\text{DWconv}_{3\times 3}\left(z'\right)\right)\right). \tag{28}$$

TABLE I
HYPERPARAMETERS SELECTED BY VARIOUS EXPERIMENTS OF THE PROPOSED ARCHITECTURE

Hyperparameters	Values
Epochs	25
Learning Rate	0.001
Optimizer	Adam
Mini Batch Size	32
Dropout	0.5
Momentum	0.688

A 1 × 1 convolutional layer is applied to restore the original number of channels and make the network computationally efficient. It is followed by a batch normalization layer to normalize the inputs to the next layer. Finally, an SE block is applied to enhance the feature representation, as shown in Fig. 9.

After the third inverted bottleneck block, a global average pooling layer is incorporated to convert the feature map to 1-D. That 1-D feature vector is passed to the dense layer, and finally, a SoftMax activation is added for classification. The loss is calculated using the categorical cross entropy. The proposed architecture contains a total of 2.64 million parameters with 6.86 GFLOPS, which makes it computationally inexpensive while having enough power to catch complex patterns from the data.

B. Training of the Proposed Architecture

In this section, we added a detailed explanation about the training of the proposed model on the selected datasets. As we used three datasets, three different models will be obtained in the output. In the training process of the proposed SEMSF-Net architecture, we opted several hyperparameters based on random search and different experiments. The best hyperparameters that are selected in this work are based on the obtained training accuracy. Table I presents the selected hyperparameters of this work. The categorical crossentropy is used as the loss function for training, whereas 70% of the data from each dataset are used for the training.

C. Testing of the Proposed Architecture

The testing process of the proposed architecture is presented in this section. In the testing phase, the testing image set of the selected datasets was passed to the trained model which outputs the classification result. The numerical results are presented in detail in Section IV. Moreover, the trained models are also visually tested through an explainable AI technique named GradCAM. The visual illustration of the proposed testing process is shown in Fig. 10. This figure shows that the test image is passed to the trained model, where features are extracted and matched for the final prediction.

IV. EXPERIMENTAL SETUP

Three publicly available datasets are utilized for the experimental process of the proposed classification challenge. Datasets

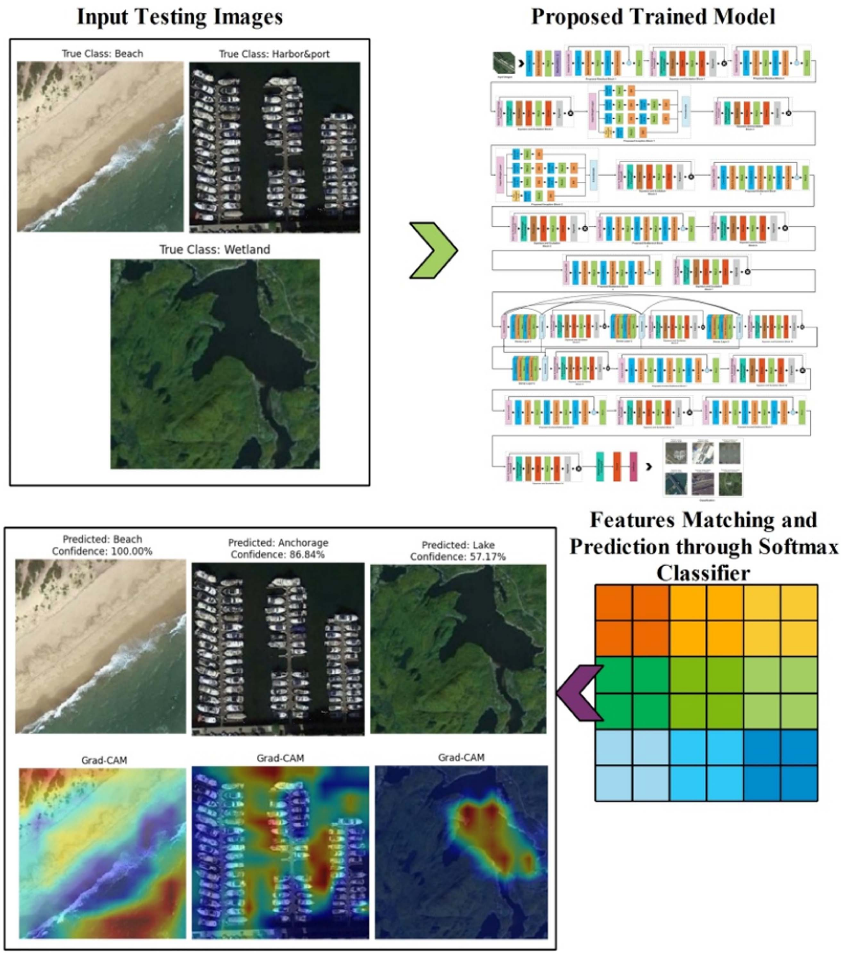

Fig. 10. Testing process of the proposed deep architecture for coastal and LULC area classification.

are presented under Section II. Each dataset is split into a ratio of 70:30, from which 70% of the data are used for training purposes and the remaining 30% for testing purposes. Through detailed ablation studies, Adam is selected as an optimizer with a learning rate of 0.001. In contrast, the number of epochs is 25, and the batch size is 32, which is also selected through detailed ablation studies. During the testing phase, a SoftMax classifier is used for classification, and GradCAM is implemented for model interpretability and transparency. Accuracy, precision, recall, and F1-score are evaluation metrics. Experiments are conducted using Python programming language with its framework, i.e., Tensorflow/keras. Simulation environment includes a workstation with 24-GB RAM and a 12-GB Graphics Card RTX 4090.

A. Proposed Architecture Results

This section presents the proposed architecture results in terms of numerical values, confusion matrices, and graphs. The results are separately presented for each dataset, whereas the detailed discussion and ablation studies are performed in the following:

1) MLRSNET Dataset Results: This section presents the proposed architecture results for the MLRSNET dataset. Table II presents the obtained average accuracy of 93.07% on this dataset. Moreover, the precision rate of 0.9381 emphasizes that it minimizes false positives to the maximum extent. Its recall rate is 0.9275, suggesting that the model captures the most relevant positive cases. An F1-score of 0.9321 reflects a robust balance between precision and recall. The above metrics collectively highlight the model's robust performance. It delivers precise and reliable predictions over the MLRSNET dataset.

Table II shows the classwise performance metrics for the MLRSNet dataset, showing the model's effectiveness in classifying RS images with different categories. High precision, recall, and F1-scores are seen for most classes, such as airplane, beach, cloud, and dense residential area, which means that the model can correctly predict these categories with fewer errors. However, some classes, such as railway stations, overpasses, and parks, tend to have relatively lower F1-scores due to interclass similarities or even fewer features that distinguish such classes. Macro average scores (94% precision, 93% recall, and 93% F1-score) confirm that the model was balanced on all classes, while micro average and weighted average metrics showed excellent overall accuracy on this dataset. Although the model faces some

TABLE II
PROPOSED CLASSIFICATION RESULTS OF THE MLRSNET DATASET USING THE PROPOSED ARCHITECTURE

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Airplane	97	97	97	503
Airport	92	88	90	673
Bareland	88	95	92	463
baseball_diamond	97	99	98	617
basketball_court	87	92	89	910
Beach	99	98	98	780
Bridge	89	94	92	711
Chaparral	97	97	97	771
Cloud	98	99	99	536
commercial_area	92	91	92	725
dense_residential_area	96	98	97	867
Desert	97	97	97	763
eroded_farmland	89	87	88	726
Farmland	94	95	94	746
Forest	94	95	95	731
Freeway	96	94	95	753
golf_course	95	97	96	780
ground_track_field	96	91	94	754
harbor_port	97	96	97	766
industrial_area	98	88	93	699
Intersection	95	91	93	733
Island	98	98	98	717
Lake	97	97	97	743
Meadow	97	82	89	773
mobile_home_park	97	98	97	753
Mountain	91	85	88	741
Overpass	81	89	85	769
Park	90	76	82	509
parking_lot	96	98	97	726
parkway	90	93	92	742
railway	84	77	81	743
railway_station	74	79	76	653
river	92	93	93	769
roundabout	92	85	89	599
shipping_yard	99	99	99	760
snowberg	94	97	96	768
sparse_residential_area	93	95	94	544
stadium	93	90	91	701
storage_tank	95	96	96	745
swimming_pool	99	100	99	591
tennis_court	92	92	92	759
terrace	95	91	93	732
transmission_tower	98	93	96	752
vegetable_greenhouse	98	97	98	773
wetland	95	81	87	758
wind_turbine	99	100	99	609
–	–	–	–	–
micro average	94%	93%	93%	32 736
macro average	94%	93%	93%	32 736
weighted average	94%	93%	93%	32 736
samples average	93%	93%	93%	32 736

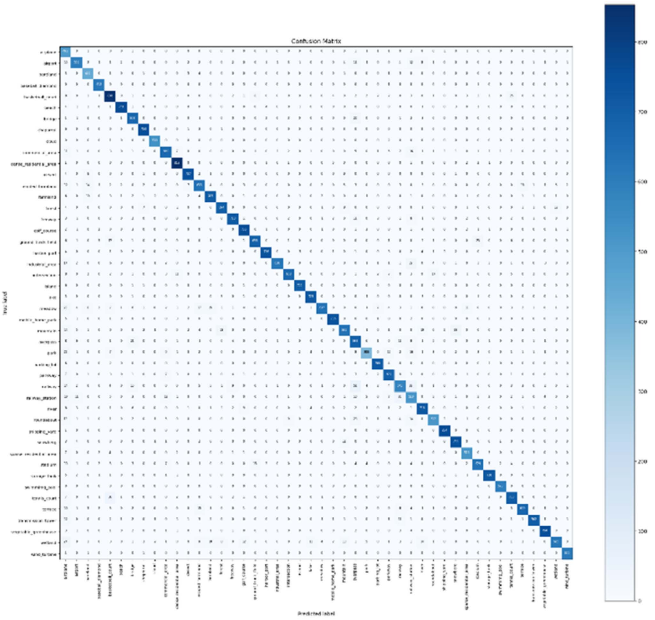

Fig. 11. Confusion matrix of the MLRSNet dataset using the proposed architecture.

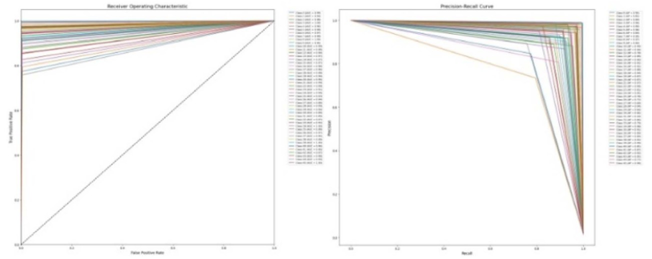

Fig. 12. ROC and PRC curves of the MLRSNet dataset using the proposed architecture.

challenges in a few classes, it obtains robust classification results and can be used for complex and heterogeneous RS data.

Fig. 11 also depicts the confusion matrix of this dataset. The confusion matrix visually depicts the model's classification performance across all classes. It shows strong diagonal dominance, meaning that most categories are predicted accurately. Misclassifications are sparse, showing that the model can effectively handle diverse RS categories with minimal confusion between similar classes.

To further analyze the performance of the proposed architecture on this dataset, we plotted ROC curves. Fig. 12 illustrates the ROC curves of the dataset obtained by the proposed architecture. The left ROC curve depicts the tradeoff between the true and false positive rates at different thresholds, with curves closer to the top-left indicating better performance. The precision–recall curve emphasizes the relationship between precision and recall, which is useful for imbalanced datasets, where curves closer to the top-right indicate superior performance. In this figure, all the classes are plotted based on their precision and recall values.

TABLE III
Ablation Study 1 on the MLRSNet Dataset for the Evaluation of the Proposed DL Model

	Epochs															
10	✓															
15		✓														
20			✓													
25				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Optimizers																
ADAM	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓
SGDM						✓										
POP							✓									
NADAM								✓								
Batch Size																
16									✓							
32	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓	✓
64											✓					
128												✓				
Learning Rate																
0.1													✓			
0.01														✓		
0.001	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
0.0001																✓
Accuracy	88	89	91	**93**	80	90	90	**93**	89	**93**	87	89	67	87	**93**	89

Bold denotes the most highest accuracy.

a) Ablation studies: For the in-depth analysis of the proposed model for this dataset, we performed some ablation studies. These ablation studies systematically evaluate the effect of several hyperparameters, including epochs, optimizers, batch size, and learning rate, on model accuracy. All the results of this ablation study are noted in Table III. This table shows that 25 epochs consistently produce the highest accuracy of 93%, thus being a preferred number of epochs for convergence. In the case of optimizers, ADAM performed best, making high accuracy across configurations, whereas other alternatives, such as SGDM, POP, and NADAM, did not significantly contribute to achieving high accuracy. For batch size, the highest is the 32 instances that facilitate multiple repetitions with high accuracy, and more than that, such as 64 and 128, cause underperformance. For a learning rate, the rate of 0.001 facilitates the best, with 93% of best accuracy, whereas a tremendous rate (0.1) and minimal rates (0.0001) bring suboptimal performance. This study shows that fine-tuning hyperparameters enhance performance, and it is essential to balance factors like training iterations, choices of optimization algorithms, and data batch handling to achieve robust model results.

TABLE IV
Ablation Study 2 for the MLRSNet Dataset to Analyze the Performance of Baseline Models

Variants	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Baseline	92.74%	93.51%	92.27%	92.85%
Baseline + SE	93.07%	93.81%	92.75%	93.21%

In the second ablation study, we compare the baseline model's performance with and without including the SE module across four metrics: accuracy, precision, recall, and F1-score (see results in Table IV). The baseline model achieves an accuracy of 92.74%, precision of 93.51%, recall of 92.27%, and F1-score of 92.85% . Adding the SE module improves every metric by a small extent, and accuracy increases to 93.07%, precision up to 93.81%, and recall up to 92.75% while increasing the F1-score to 93.21% . These indicate that the SE module improves the model's capability to target relevant features, contributing to marginal but consistent gain in all evaluation metrics. Moreover, we compared the proposed CNN architecture with pretrained models, which shows that our model improves accuracy with the least parameter count and is the most efficient. The proposed architecture achieves a maximum accuracy of 93.07% with a minimum number of parameters, i.e., 2.64 million parameters (see results in Table V).

2) NWPU Dataset Results: Table VI presents the results of the NWPU dataset that achieved overall improved performance of the model compared to all the key metrics of evaluation. The model has obtained an accuracy of 95.70%, expressing its ability to classify most cases with correctness. It has also achieved a precision of 96.04%, stating that the positive predictions are solidly reliable, and the recall of 95.41%, which portrays its efficiency in identifying those relevant instances. The F1-score of 95.72% confirms the balanced performance between precision

TABLE V
COMPARATIVE ANALYSIS WITH PRETRAINED MODELS FOR THE MLRSNET DATASET

Model Name	Accuracy	Parameters (Million)	Model size (MB)	Model Layers
Alexnet [30]	61.32	60	240	8
VGG16 [29]	72.44	138	528	16
VGG19 [29]	67.10	143.7	550	19
GoogleNet [54]	84.36	6.8	27	22
ResNet50 [31]	85.68	25.6	98	50
ResNet101 [31]	86.05	44.5	170	101
Proposed	**93.07**	**2.64**	**10**	**211**

TABLE VI
PROPOSED CLASSIFICATION RESULTS OF THE NWPU DATASET USING THE PROPOSED ARCHITECTURE

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Airfield	94	92	93	422
Anchorage	100	97	98	224
Beach	97	99	98	202
Dense Residential	94	99	96	194
Farm	96	96	96	428
Flyover	95	98	97	189
Forest	98	99	98	207
Game Space	97	96	96	440
Parking Space	97	96	97	195
River	90	89	90	207
Sparse Residential	98	93	95	210
Storage Cisterns	100	94	97	218
–	–	–	–	–
Micro Avg	96%	95%	96%	3136
Macro Avg	96%	96%	96%	3136
Weighted Avg	96%	95%	96%	3136
Samples Avg	95%	95%	95%	3136

Bold denotes the most highest accuracy.

and recall. These metrics highlight the model's robustness and reliability for classification tasks on the NWPU dataset.

The classwise performance metrics that highlight the model's effectiveness across various classes are also computed in this table. The precision, recall, and F1-scores for most classes are remarkably high, while some classes, such as Anchorage and Storage Cisterns, perform with perfect precision at 100% and high F1-scores of 98% and 97%, respectively. Other classes, such as Beach, Forest, and Parking Space, have a remarkable balance in their metrics. On the other hand, the class River scored relatively less (90% precision, 89% recall, and 90% F1-score). This may denote difficulty with differentiating features for this particular class. A micro, macro, and weighted average value of 96% could be found for precision, recall, and F1-score, denoting consistency across all classes. A confusion matrix is also presented for this dataset in Fig. 13. A strong diagonal on the confusion matrix signifies that the model's prediction is quite accurate for most classes. There are off-diagonal cells representing minor misclassifications, and such occurrences are very noticeable, especially for classes similar to "River." In summary, the confusion matrix further ascertains that the model is accurate with strong robustness performance on the diversified categories.

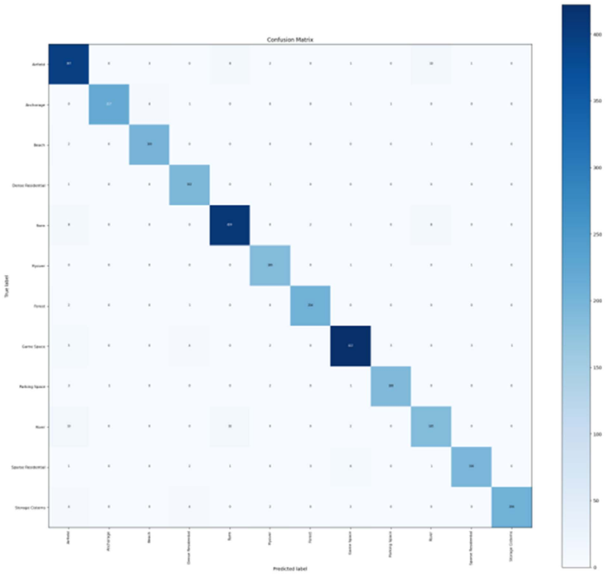

Fig. 13. Confusion matrix of the NWPU dataset using the proposed CNN model.

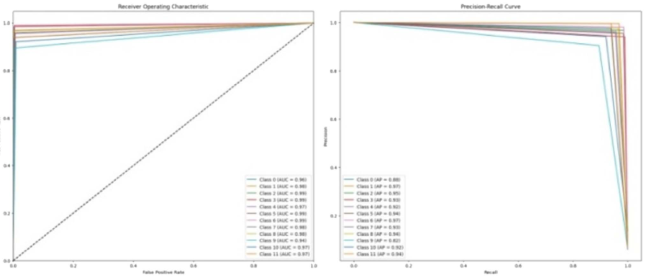

Fig. 14. ROC and PRC curve of the NWPU dataset.

Moreover, the ROC and PR curves for multiclass classification performance are shown in Fig. 14. The AUC values of the ROC curves are all high and close to 1 for all classes, which means that true positive rates are excellent, while false positive rates are low across the classes. Similarly, the PR curve shows relatively high precision and recall, and most classes show good performances, as illustrated by the steep curves near the upper right region.

TABLE VII
ABLATION STUDY 1 ON THE NWPU DATASET FOR THE EVALUATION OF THE PROPOSED DL MODEL

Epochs																
20	✓															
40		✓														
60			✓													
80				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Optimizers																
ADAM	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓
SGDM						✓										
POP							✓									
NADAM								✓								
Batch Size																
16									✓							
32	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓	✓
64											✓					
128												✓				
Learning Rate																
0.1													✓			
0.01														✓		
0.001	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
0.0001																✓
Accuracy	92	91	93	**95**	92	91	93	**95**	93	**95**	91	94	93	92	**95**	92

Bold denotes the most highest accuracy.

These metrics confirm a strong discriminative ability for the model with good reliability in both balanced and imbalanced scenarios.

a) Ablation studies: We performed several ablation studies and comparisons with pretrained models for the in-depth evaluation of the proposed CNN model. In the first ablation study, the performance of a proposed model under varying hyperparameter settings, such as epochs, optimizers, batch size, and learning rates, is done (see Table VII). The accuracy values, thus, depict the impact of those configurations on the model's effectiveness. The performance improves by increasing the epochs (from 20 to 80), with a highest consistent result at 80 epochs. Among optimizers, the ADAM consistently shows robustness across configurations. While others, such as SGDM, POP, and NADAM, contribute moderately. Batch size variations indicate that 32 is the best performance, although results can be competitive with sizes of 16 and 64. The choice of learning rates presents 0.001 as a stable and efficient rate since it achieves high accuracy in most settings; on the other hand, others, such as 0.1 and 0.0001, present a high variability. This analysis highlights the role of hyperparameter tuning, which should be used to achieve an optimal model configuration with 95% accuracy.

The second ablation study compares the baseline level with the addition of SE blocks, and the performance is computed (see Table VIII). The baseline model resulted in an accuracy of 94.96%, a precision rate of 95.33%, a recall rate of 94.67%, and

TABLE VIII:
ABLATION STUDY 2 FOR NWPU DATASET, WHERE BASE MODELS ARE COMPARED

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Baseline	94.96	95.33	94.67	94.97
Baseline + SE	95.70	96.04	95.41	95.72

TABLE IX
COMPARATIVE ANALYSIS WITH PRETRAINED MODELS FOR THE NWPU DATASET

Model Name	Accuracy	Parameters (Million)	Model size (MB)	Model Layers
Alexnet [30]	84.66	60	240	8
VGG16 [29]	86.25	138	528	16
VGG19 [29]	85.92	143.7	550	19
GoogleNet [54]	83.41	6.8	27	22
ResNet50 [31]	90.14	25.6	98	50
ResNet101 [31]	91.78	44.5	170	101
Proposed	**95.70**	**2.64**	**10**	**211**

an F1-score of 94.97% . The addition of the SE block further improved all the metrics, such as accuracy of 95.70%, precision rate of 96.04%, recall rate of 95.41%, and the F1-score value of 95.72% . This result shows that SE blocks improve feature representation and enhance the model's overall performance in classification tasks.

Finally, a comparative analysis with standard pretrained models was conducted using a proposed accuracy and parameter efficiency model. A comparison is performed in Table IX. From this table, it is noted that the proposed model gained much accuracy of 95.70% while having only 2.64 million parameters, making it the most accurate and lightweight compared to benchmarked architectures. Also, this highlights its exceptional balance between performance and computational efficiency, making it a more practical choice for deployment in real-world applications, especially in environments with limited computational resources.

TABLE X
CLASSIFICATION RESULTS OF THE COASTAL DATASET USING THE PROPOSED ARCHITECTURE

Class	Precision	Recall	F1-Score	Support
Anchorage	0.76	0.93	0.84	210
Beach	0.97	0.97	0.97	585
Harbor	0.58	0.47	0.52	60
Harbor & port	0.95	0.74	0.83	376
Island	0.95	0.99	0.97	375
Lake	0.97	0.99	0.98	1275
Landslide	0.98	0.90	0.94	231
Red Sea Fish	1.00	1.00	1.00	315
River	0.94	0.95	0.95	960
Snowberg	0.96	0.98	0.97	384
Swimmimg_pool	1.00	0.98	0.99	301
Water	0.53	0.42	0.47	60
Wetland	0.92	0.89	0.91	392
Micro Avg	0.95	0.95	0.95	5524
Macro Avg	0.81	0.79	0.80	5524
Weighted Avg	0.95	0.95	0.95	5524
Samples Avg	0.95	0.95	0.95	5524

Bold denotes the most highest accuracy.

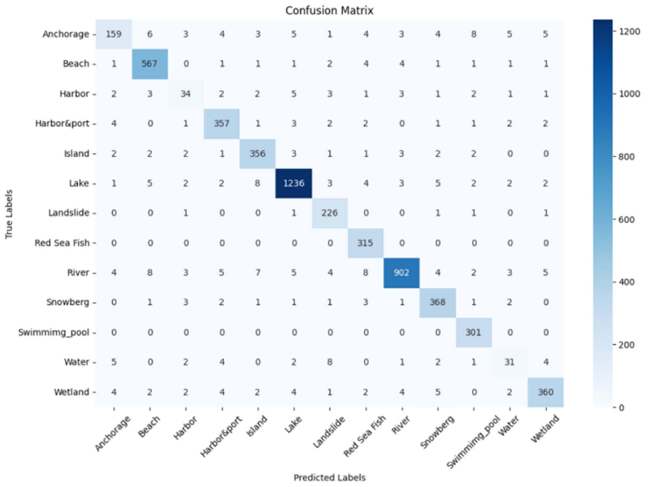

Fig. 15. Confusion matrix of the Coastal dataset using the proposed architecture.

3) Coastal Dataset Results: The coastal dataset results for the proposed model are presented in Table X. Overall, the proposed model obtained 94.94% accuracy on this dataset. The precision rate of this dataset is 95.36, the recall rate is 94.81, and the F1-score value is 94.94% . The classwise results are also presented in Table X. This table shows that the precision rates of Anchorage, Harbor, and Water classes are 0.76, 0.58, and 0.53, respectively, which are not good compared to other classes presented in this dataset. This dataset's micro average precision rate is 0.95, the macro average precision rate is 0.81, the weighted average precision rate is 0.95, and the sample average value is 0.95. Similarly, the micro average recall rate is 0.95, the macro average recall rate is 0.79, and the weighted average recall rate is 0.95. Overall, it is observed that the recall rate and precision rates are above 90% for this dataset using the proposed architecture. Fig. 15 also illustrates the confusion matrix of this dataset that shows each class's improved correct prediction rate.

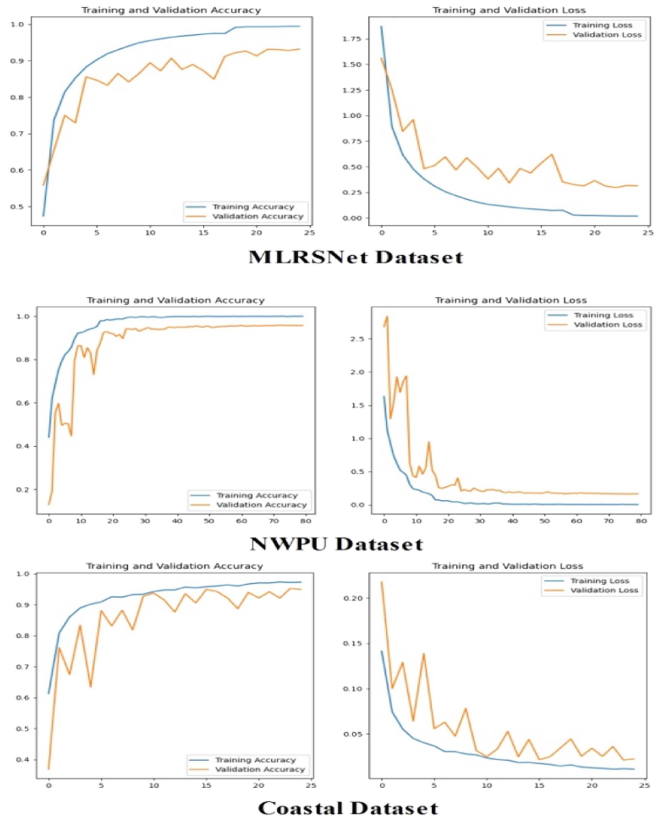

Fig. 16. Training plots of the proposed architecture on selected datasets.

TABLE XI
COMPARATIVE ANALYSIS WITH SOTA FOR THE MLRSNET DATASET

MLRSNET Dataset Comparison			
Architecture	**Accuracy**	**mF1**	**Params**
FMA-Net [35]	91.0		11.3M
AMEGRF-Net [36]	91.51		14.8M
Yang et al. [37]	82.59		7.79M
VGG16 [55]	–	68.01	134M
VGG16 + SSM [55]	–	72.44	–
VGG16 + SRBM [55]	–	71.26	–
VGG16 + SR-NET [55]	–	73.80	31M
VGG19 [55]	–	67.10	140M
VGG19 + SSM [55]	–	72.91	–
VGG19 + SRBM [55]	–	70.12	–
VGG19 + SR-Net [55]	–	73.33	36M
ResNet50 [55]	–	85.68	24M
ResNet50 + SSM [55]	–	86.58	–
ResNet50 + SRBM [55]	–	86.07	–
ResNet50 + SR-Net [55]	–	87.21	42M
ResNet101 [55]	–	86.05	43M
ResNet101 + SSM [55]	–	86.92	–
ResNet101 + SRBM [55]	–	87.71	–
ResNet101 + SR-Net [55]	–	87.55	61M
DenseNet201 [55]	–	86.17	18M
DenseNet201 + SSM [55]	–	86.56	–
DenseNet201 + SRBM [55]	–	86.26	–
DenseNet201 + SR-Net [55]	–	87.36	39M
Proposed	**93.07**	**93.21**	**2.64M**

NWPU Dataset		
Architecture	**Accuracy**	**Params**
Khan et al. [56]	93.3	5.7M
EAM [57]	93.04	–
WSADAN-ResNet50 [39]	92.63	–
BestC [58]	95.28	43.78M
Albarakati et al. [40]	91.7	18.6M
Proposed	**95.70**	**2.64M**

Bold denotes the most highest accuracy.

B. Discussion and Comparison With State of the Art

This section includes a brief discussion based on the proposed architecture's graphical and tabular form. Tables II–X present detailed results for each dataset of this article. Moreover, the obtained accuracy and precision values are supported through confusion matrices, as seen in Figs. 11, 13, and 15. Moreover, precision–recall ROC plots are also added to validate the proposed model performance.

The detailed architecture of the proposed SEMSF-Net, which was trained on three datasets separately, is presented. The training and loss curves are shown in Fig. 16. In this figure, the graphs of the MLRSNet dataset depicted the model's training and validation performance over 25 epochs. The left graph increases in both the training and validation accuracy. On the other hand, the right graph shows a smooth decrease for both training and validation loss, confirming the convergence of the model. Although the validation loss varies slightly, its overall decreasing pattern indicates that the prediction capacity is improving. The graphs of the NWPU dataset show that the model's training and validation accuracy (left) and loss (right) have been done over 80 epochs. Training accuracy is steadily increasing to nearly 97%, while validation accuracy is stabilizing at a high value after some initial fluctuations, which means that it has learned effectively and generalized well. Similarly, the coastal area dataset training curves show the validation and training accuracy of over 90% and below 94.6% . Hence, the overall training of the proposed model seems smooth on the selected datasets without any overfitting.

1) Comparison With State of the Art: The comparative analysis evaluates the proposed architecture against state-of-the-art (SOTA) models in terms of accuracy, mean F1-score (mF1), and parameter efficiency (see Table XI). The proposed model outperforms all existing architectures with an accuracy of 93.07% and an mF1-score of 93.21% . In addition, the proposed model is highly parameter efficient with just 2.64M parameters, which is much smaller than other architectures, such as ResNet101 with SR-Net (61M) and DenseNet201 with SR-Net (39M). Notably, SOTA models, such as AMEGRF-Net accuracy of 91.51%, parameters 14.8M, FMA-Net accuracy of 91.0%, and parameters 11.3M, are a bit higher in parameters and lesser in terms of accuracy than the proposed architecture.

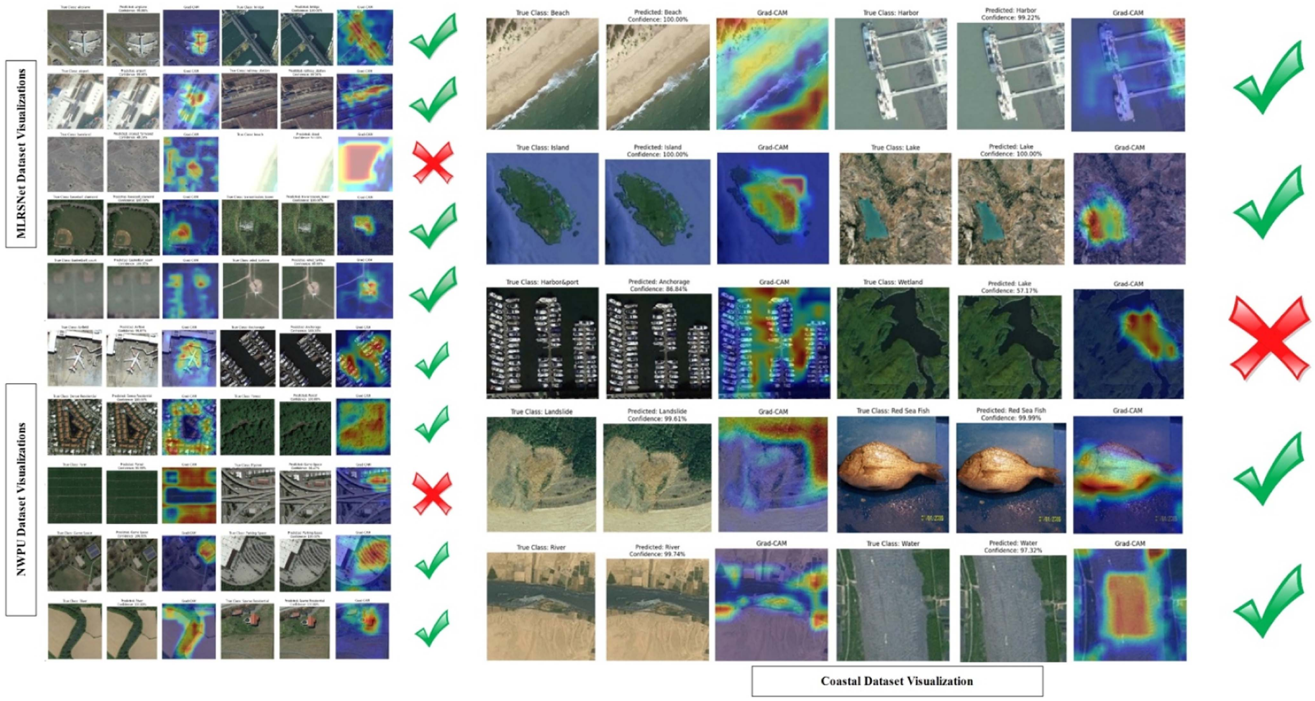

Fig. 17. GradCAM visualization of the proposed SEMSF-net model on the selected dataset.

Similarly, for the NWPU dataset, it shows that the proposed model improved the accuracy and less required trainable parameters. The existing architectures, such as in [5], achieve 93.3% accuracy with 5.7 million parameters, while EAM [6] achieves 93.04% . WSADAN-ResNet50 [7] delivers 92.63% accuracy, and Pal and Mather [9] obtain 91.7% with 18.6 million parameters. BestC [8], with an impressive accuracy of 95.28%, is very parameter intensive and has 43.78 million parameters, which makes it less efficient. The proposed model surpasses these approaches with a maximum accuracy of 95.70% and also has only 2.64 million parameters. This indeed shows a wonderful balance of accuracy and computational efficiency that makes the proposed model more suitable for resource-constrained environments. Moreover, it makes the applicability more effective toward real-world practice.

2) GradCAM Visualization of the Proposed SEMSF-Net: Finally, to analyze the training of the proposed model on selected datasets, we performed interpretation through GradCAM visualization. GradCAM highlights different areas of the image with different colors, and each color shows the importance of that region in the overall classification. The red color shows that these areas are most critical and influential. The model's decision heavily relies on these regions, whereas orange or yellow colors show that these regions have a moderate influence on model's output, while the blue color shows little to no influence. Overall, the image shows the visual outcome of a model's classifying performance, including Grad-CAM visualizations for interpretability. Correctly classified examples are marked with green ticks, showing accurate predictions that include "airplane," "bridge," and "baseball diamond." The Grad-CAM heatmaps depict regions contributing to the predictions: they align well with target objects. However, an incorrect classification is also obtained and circled by a red cross, such as the "Beach" class, which shows that the model could not successfully localize or interpret the dominant characteristic features and, thus, produced heatmaps on the wrong characteristics.

In the samples of the NWPU dataset, each row is a class from Airfield to Anchorage, Dense Residential, Forest, and more, which are represented by the predicted labels and confidence scores accompanying the Grad-CAM heatmaps. Most predictions align well with the actual class, as the green checkmarks indicate; thus, the model is precise. However, there are a few misclassifications, marked by red crosses, which indicate areas for improvement. For the coastal region dataset, the Harbor & Port class is wrongly predicted by the proposed model. In contrast, the rest of the images are correctly predicted and generated heatmap on the correct region. Hence, the results emphasize the model's strength in feature localization and classification accuracy, but the occasional errors give insights into further refinement.

V. Conclusion

This article proposes a novel SEMSF-Net to classify LULC and the coastal regions using the RS images. In the proposed network, several blocks are designed based on multiscale such as dense, inception, inverted, and bottleneck residual embedded with the SE module to extract an RS image's more in-depth feature information. The proposed architecture is trained on three datasets: Coastal dataset, MLRSNet, and NWPU, with an improved accuracy of 94.94%, 93.7%, and 95.70%, respectively. Based on the detailed ablation studies and comparisons, we conclude the following points:

1) The datasets used in the work are imbalanced or have fewer training images; however, the proposed network is especially designed for imbalanced and smaller image datasets. With the current datasets, there is no need for augmentation; the network training performance is not to be degraded.
2) The SE module increased the entire network's learning capacity when embedded with dense, inception, and bottleneck residual blocks.
3) The use of SE with these blocks increased the training/testing accuracy and reduced the learnable parameters that make this network more suitable for real-time applications.
4) GradCAM-based interpretation of the proposed shows the correct prediction with high accuracy and precision rate; however, for the few images, a wrong prediction occurred.

The limitation of the proposed framework is the model produced some misclassification for class with small differences between classes, such as "Harbor" and "Harbor & Port." There is a slight performance drop for noisy or underrepresented classes in the imbalanced datasets, suggesting room for improvement in handling data imbalance issues, and the proposed model components do limit long-range dependencies and do not have the similar modeling features of a transformer architecture. Therefore, we will explore various architecture designs with vision transformers and also various modifications in data augmentation techniques to improve classification performance and generalizability in the future.

Acknowledgment

Dataset availability: The datasets of this work are publically available for the research purposes. The links of datasets used in this study are as follows.

1) MLRSNet: A Multilabel High Spatial Resolution RS Dataset for Semantic Scene Understanding. [Online]. Available: https://github.com/cugbrs/MLRSNet

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

References

[1] M. S. Khan, S. Ullah, T. Sun, A. U. Rehman, and L. Chen, "Land-use/land-cover changes and its contribution to urban heat Island: A case study of Islamabad, Pakistan," *Sustainability*, vol. 12, no. 9, 2020, Art. no. 3861.

[2] M. A. Moharram and D. M. Sundaram, "Dimensionality reduction strategies for land use land cover classification based on airborne hyperspectral imagery: A survey," *Environ. Sci. Pollut. Res.*, vol. 30, no. 3, pp. 5580–5602, 2023.

[3] B. Bansod, R. Singh, R. Thakur, and G. Singhal, "A comparision between satellite based and drone based remote sensing technology to achieve sustainable development: A review," *J. Agriculture Environ. Int. Develop.*, vol. 111, no. 2, pp. 383–407, 2017.

[4] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.

[5] M. C. Hansen and T. R. Loveland, "A review of large area monitoring of land cover change using Landsat data," *Remote Sens. Environ.*, vol. 122, pp. 66–74, 2012.

[6] M. K. Bhatti et al., "A novel approach for high-resolution coastal areas and land use recognition from remote sensing images based on multimodal network-level fusion of SRAN3 and lightweight four encoders ViT," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 6844–6858, 2025.

[7] J. Holloway and K. Mengersen, "Statistical machine learning methods and remote sensing for sustainable development goals: A review," *Remote Sens.*, vol. 10, no. 9, 2018, Art. no. 1365.

[8] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.

[9] M. Pal and P. M. Mather, "Decision tree based classification of remotely sensed data," in *Proc. 22nd Asian Conf. Remote Sens.*, 2001, vol. 5, Art. no. 9.

[10] M. Pal and P. M. Mather, "An assessment of the effectiveness of decision tree methods for land cover classification," *Remote Sens. Environ.*, vol. 86, no. 4, pp. 554–565, 2003.

[11] J. G. Ghatkar, R. K. Singh, and P. Shanmugam, "Classification of algal bloom species from remote sensing data using an extreme gradient boosted decision tree model," *Int. J. Remote Sens.*, vol. 40, no. 24, pp. 9412–9438, 2019.

[12] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogrammetry Remote Sens.*, vol. 114, pp. 24–31, 2016.

[13] L. Samaniego, A. Bárdossy, and K. Schulz, "Supervised classification of remotely sensed imagery using a modified $ k $-NN technique," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2112–2125, Jul. 2008.

[14] X. Chao and Y. Li, "Semisupervised few-shot remote sensing image classification based on KNN distance entropy," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8798–8805, 2022.

[15] N. A. Mahmon and N. Ya'acob, "A review on classification of satellite image using artificial neural network (ANN)," in *Proc. IEEE 5th Control Syst. Graduate Res. Colloq.*, 2014, pp. 153–157.

[16] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 11, pp. 2207–2219, Nov. 2016.

[17] M. J. Cracknell and A. M. Reading, "Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information," *Comput. Geosci.*, vol. 63, pp. 22–33, 2014.

[18] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.

[19] W. Li et al., "Uncertainties analysis of collapse susceptibility prediction based on remote sensing and GIS: Influences of different data-based models and connections between collapses and environmental factors," *Remote Sens.*, vol. 12, no. 24, 2020, Art. no. 4134.

[20] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Mag.*, vol. 35, no. 4, pp. 105–120, 2014.

[21] H. Ni, H. Guan, X. Tong, and J. Chanussot, "Conditional Gaussian enhanced dense correlation matching for cross-category land cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5604515.

[22] G. Farias et al., "Automatic feature extraction in large fusion databases by using deep learning approach," *Fusion Eng. Des.*, vol. 112, pp. 979–983, 2016.

[23] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *Int. J. Remote Sens.*, vol. 29, no. 3, pp. 617–663, 2008.

[24] A.-M. Tousch, S. Herbin, and J.-Y. Audibert, "Semantic hierarchies for image annotation: A survey," *Pattern Recognit.*, vol. 45, no. 1, pp. 333–345, 2012.

[25] K. Zhou, D. Ming, X. Lv, J. Fang, and M. Wang, "CNN-based land cover classification combining stratified segmentation and fusion of point cloud and very high-spatial resolution remote sensing image data," *Remote Sens.*, vol. 11, no. 17, 2019, Art. no. 2065.

[26] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, 2023, Art. no. 151.

[27] A. Zafar et al., "A comparison of pooling methods for convolutional neural networks," *Appl. Sci.*, vol. 12, no. 17, 2022, Art. no. 8643.

[28] N. G. Shree and C. Meiaraj, "Land use and land cover change detection by machine learning classifiers (SVM and RF) using satellite remote sensing observations for Cuddalore Taluk, Tamil Nadu, India," *J. Earth Syst. Sci.*, vol. 134, no. 1, pp. 1–16, 2025.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Jun. 27, 2025.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[32] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[33] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community," *J. Appl. Remote Sens.*, vol. 11, no. 4, 2017, Art. no. 042609.

[34] M. Peng et al., "Optimizing cover mapping in coastal areas using swin transformer-based multi-sensor remote sensing satellite data fusion," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, early access, Jun. 27, 2025, doi: 10.1109/JSTARS.2025.3541107.

[35] F. Rauf et al., "FMANet: Super resolution inverted bottleneck fused self-attention architecture for remote sensing satellite image recognition," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 18622–18634, 2024.

[36] 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 Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5523119.

[37] X. Yang et al., "An efficient lightweight satellite image classification model with improved MobileNetV3," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2024, pp. 1–6.

[38] S. M. A. H. Shah, M. Q. Khan, A. Rizwan, S. U. Jan, N. A. Samee, and M. M. Jamjoom, "Computer-aided diagnosis of Alzheimer's disease and neurocognitive disorders with multimodal Bi-Vision Transformer (BiViT)," *Pattern Anal. Appl.*, vol. 27, no. 3, 2024, Art. no. 76.

[39] L. Wang, K. Qi, C. Yang, and H. Wu, "Weakly supervised scale adaptation data augmentation for scene classification of high-resolution remote sensing images," *Nat. Remote Sens. Bull.*, vol. 27, no. 12, pp. 2815–2830, 2024.

[40] 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 J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 6338–6353, 2024.

[41] V. Pushpalatha, P. Mallikarjuna, H. Mahendra, S. R. Subramoniam, and S. Mallikarjunaswamy, "Land use and land cover classification for change detection studies using convolutional neural network," *Appl. Comput. Geosci.*, vol. 25, 2025, Art. no. 100227.

[42] *MLRSNet: A Multi-label High Spatial Resolution Remote Sensing Dataset for Semantic Scene Understanding*. Accessed: Jun. 27, 2025. [Online]. Available: https://github.com/cugbrs/MLRSNet

[43] NWPU-RESISC45. Accessed: Jun. 27, 2025. [Online]. Available: https://figshare.com/articles/dataset/NWPU-RESISC45_Dataset_with_12_classes/16674166?file=30871912

[44] D. Bhatt et al., "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, 2021, Art. no. 2470.

[45] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, pp. 1–74, 2021.

[46] N. Aggarwal, B. Saini, and S. Gupta, "Role of artificial intelligence techniques and neuroimaging modalities in detection of Parkinson's disease: A systematic review," *Cogn. Comput.*, vol. 16, no. 4, pp. 2078–2115, 2024.

[47] Z. Feng, "An Overview of ResNet Architecture and Its Variants." Accessed: Jun. 27, 2025. [Online]. Available: https://builtin.com/artificial-intelligence/resnet-architecture

[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[49] A. W. Salehi et al., "A study of CNN and transfer learning in medical imaging: Advantages, challenges, future scope," *Sustainability*, vol. 15, no. 7, 2023, Art. no. 5930.

[50] R. Alake, "Deep learning: Understanding the inception module." Accessed: Jun. 27, 2025. [Online]. Available: https://towardsdatascience.com/deep-learning-understand-the-inception-module-56146866e652

[51] N. S. Choudhary, "A comprehensive guide to understanding and implementing bottleneck residual blocks." Accessed: Jun. 27, 2025. [Online]. Available: https://medium.com/@neetu.sigger/a-comprehensive-guide-to-understanding-and-implementing-bottleneck-residual-blocks-6b420706f66b

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[53] N. Naheed, M. Shaheen, S. A. Khan, M. Alawairdhi, and M. A. Khan, "Importance of features selection, attributes selection, challenges and future directions for medical imaging data: A review," *Comput. Model. Eng. Sci.*, vol. 125, no. 1, pp. 314–344, 2020.

[54] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[55] X. Tan, Z. Xiao, J. Zhu, Q. Wan, K. Wang, and D. Li, "Transformer-driven semantic relation inference for multilabel classification of high-resolution remote sensing images," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1884–1901, 2022.

[56] J. A. Khan et al., "Design of super resolution and fuzzy deep learning architecture for the classification of land cover and landsliding using aerial remote sensing data," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 337–351, 2025.

[57] C. Sitaula, S. KC, and J. Aryal, "Enhanced multi-level features for very high resolution remote sensing scene classification," Jun. 27, 2025, *arXiv:2305.00679*.

[58] W. Hu, C. Lan, T. Chen, S. Liu, L. Yin, and L. Wang, "Scene classification of remote sensing image based on multi-path reconfigurable neural network," *Land*, vol. 13, no. 10, 2024, Art. no. 1718.

Muhammad John Abbas (Member, IEEE) received the bachelor's degree in computer science from HITEC University, Rawalpindi, Pakistan, in 2025.

He is currently a Research Associate with the Center for Artificial Intelligence, Prince Mohammad bin Fahd University, Al-Khobar, Saudi Arabia. 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. His research interests include 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 the Center for Artificial Intelligence, Prince Mohammad bin Fahd University, Al-Khobar, Saudi Arabia. He has authored or coauthored more than 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 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS OF NEURAL NETWORKS, *Pattern Recognition Letters*, *Multimedia Tools and Applications*, *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*. His primary research interests include medical imaging, COVID-19, magnetic resonance imaging analysis, video surveillance, human gait recognition, and agriculture plants using deep learning.

Ameer Hamza (Member, IEEE) is working toward the Ph.D. degree in computer science with the Kaunas University of Technology, Kaunas, Lithuania.

He has authored or coauthored 20 impact factor papers to date. His major research interests include object detection and recognition, video surveillance, medical, and agriculture using deep learning and machine learning.

Shrooq Alsenan received the Ph.D. degree in information system sciences from King Saud University, Riyadh, Saudi Arabia, in 2022.

She is an Academic and a Researcher of Artificial Intelligence (AI) and currently directs the AI Center, Princess Nourah bint Abdulrahman University, Riyadh. She has received a prestigious postdoctoral fellowship with the Computer Science and Artificial Intelligence Laboratory and Jameel Clinic, Massachusetts Institute of Technology, Cambridge, MA, USA. Her research interests include AI in health care, remote sensing, bioinformatics, and hyperspectral images.

Areej Alasiry received the B.Sc. degree in information systems from King Khalid University, Abha, Saudi Arabia, in 2008, 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, London, U.K., in 2010 and 2015, respectively.

She is currently an Assistant Professor with the College of Computer Science, King Khalid University, where she is also the College Vice Dean for Graduate Studies and Scientific Research. Her main research interests include machine learning and data science.

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 engineering.

From 2001 to 2005, he was a Research Assistant with the Electronics and Microelectronics Laboratory, University of Monastir, where he was also an Assistant Professor with Electronics Department from 2006 to 2012. Since 2013, he has been an Assistant Professor with the Department of Engineering, 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.

Yang Li received the M.S. degree in electrical engineering from the Dalian University of Technology, Dalian, China, in 2016.

He is currently an Associate Professor with the College of Mechanical and Electrical Engineering, Shihezi University, Shihezi, China. His research interests include pattern recognition, few-shot learning, image processing, and data quality assessment.

Prof. Li is an Associate Editor for several journals, such as *Precision Agriculture*, *Plant Methods*, and *Data Technologies and Applications*.

Yunyoung Nam (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, Suwon, South Korea, in 2001, 2003, and 2007, respectively.

From 2007 to 2010, he was a Senior Researcher with the Center of Excellence in Ubiquitous System, Stony Brook University, Stony Brook, NY, USA, where he was also 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. From 2017 to 2020, he was the Director of the ICT Convergence Rehabilitation Engineering Research Center, Soonchunhyang University, Asan, South Korea, where he has been the Director of the ICT Convergence Research Center since 2020 and 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.