IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 December 2023, accepted 30 December 2023, date of publication 2 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349285

RESEARCH ARTICLE

Land Use and Land Cover Classification Using River Formation Dynamics Algorithm With Deep Learning on Remote Sensing Images

MOHAMMED ALJEBREEN[1], HANAN ABDULLAH MENGASH[2], MOHAMMAD ALAMGEER[3], SAUD S. ALOTAIBI[4], AHMED S. SALAMA[5], AND MANAR AHMED HAMZA[6]

[1]Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3]Department of Information Systems, College of Science and Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia
[4]Department of Information Systems, College of Computing and Information Systems, Umm Al-Qura University, Mecca 24382, Saudi Arabia
[5]Department of Electrical Engineering, Faculty of Engineering and Technology, Future University in Egypt, New Cairo 11845, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia

Corresponding author: Mohammad Alamgeer (mabdul@kku.edu.sa)

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/134/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R114), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2024R459), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444). This study is partially funded by the Future University in Egypt (FUE).

ABSTRACT Currently, remote sensing images (RSIs) are often exploited in the explanation of urban and rural areas, change recognition, and other domains. As the majority of RSI is high-resolution and contains wide and varied data, proper interpretation of RSIs is most important. Land use and land cover (LULC) classification utilizing deep learning (DL) is a common and efficient manner in remote sensing and geospatial study. It is very important in land planning, environmental monitoring, mapping, and land management. But, one of the recent approaches is problems like vulnerability to noise interference, low classification accuracy, and worse generalization ability. DL approaches, mostly Convolutional Neural Networks (CNNs) revealed impressive performance in image recognition tasks, making them appropriate for LULC classification in RSIs. Therefore, this study introduces a novel Land Use and Land Cover Classification employing the River Formation Dynamics Algorithm with Deep Learning (LULCC-RFDADL) technique on RSIs. The main objective of the LULCC-RFDADL methodology is to recognize the diverse types of LC on RSIs. In the presented LULCC-RFDADL technique, the dense EfficientNet approach is applied for feature extraction. Furthermore, the hyperparameter tuning of the Dense EfficientNet method was implemented using the RFDA technique. For the classification process, the LULCC-RFDADL technique uses the Multi-Scale Convolutional Autoencoder (MSCAE) model. At last, the seeker optimization algorithm (SOA) has been exploited for the parameter choice of the MSCAE system. The achieved outcomes of the LULCC-RFDADL algorithm were examined on benchmark databases. The simulation values show the better result of the LULCC-RFDADL methods with other approaches in terms of different metrics.

INDEX TERMS Remote sensing images, land use classification, land cover, deep learning, metaheuristics.

I. INTRODUCTION

Recently, land use and land cover (LULC) classification utilizing remote sensing image (RSI) played a crucial part in different applications namely forest management (health, stand quality, resources inventory, reforestation, and harvesting), agricultural practice (riparian zone buffers, nutrient management, conservation easements, and cropping patterns), land use planning (policy regulations, suburban sprawl, growth trends, and incentives), and biological

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

resource (wetlands, fragmentation, and habitat quality) [1]. Land use refers to the goal of land cover and applies to the surface cover on the earth, including vegetation, urban structure, bare soil, water, and others; it could not define the land use, and land use might vary for lands with the similar cover form [2]. LULC evaluation is much needed for monitoring, planning, and maintaining the use of natural resources [3], [4], [5]. Indeed, LULC classification has a direct effect on water, atmospheric, and erosion of soil, but it can be indirectly related to global landscape issues. Finally, the RSI and its processing can help provide current and massive data on surface situations [6], [7].

Newly, regular works have been performed for extracting further discriminative features for RSI classification [8]. Conventional techniques can be extremely concentrated on hand-crafted features like texture and colour features. Mid-level methods are developed to create several representations to support higher-order statistical techniques [9]. In the RS domain, scene classification images are significant and challenging problems in real-time applications like geographic image retrieval, urban development from high-spatial resolution (HSR), geospatial object detection, natural hazard detection, and environmental monitoring [10], [11]. In several real-time applications of RS, scene classification is a vital phase. Newly, deep learning (DL) based techniques are performing a key role to extract the higher level features, and are developed as a leading paradigm in pattern detection and Computer Vision (CV) [12], [13]. In the DL technique, convolutional neural networks (CNNs) are considered the commonly data-driven techniques, and effective method, which is employed for finding the intricate structures and extracting the crucial data of HSR RSI [14], [15], besides the hierarchical convolutional features of hyperspectral imagery (HSI) [16]. Moreover, a sequence of DL-based scene classification techniques has been suggested [17], [18].

This study introduces a novel Land Use and Land Cover Classification employing the River Formation Dynamics Algorithm with Deep Learning (LULCC-RFDADL) method on RSIs. In the presented LULCC-RFDADL technique, a dense EfficientNet algorithm was applied for feature extraction. Additionally, the hyperparameter tuning of the Dense EfficientNet algorithm was carried out utilizing the RFDA technique. For the classification process, the LULCC-RFDADL technique uses the Multi-Scale Convolutional Autoencoder (MSCAE) model. At last, the seeker optimization algorithm (SOA) could be exploited for the parameter selection of the MSCAE approach. The accomplished solution of the LULCC-RFDADL methodology has been tested on benchmark dataset. In short, the key contributions are given as follows.

- Develop an intelligent LULCC-RFDADL technique comprising dense EfficientNet, RFDA based hyperparameter tuning, MSCAE based classification, and SOA based parameter tuning has been presented for land use classification.
- Leverages the dense EfficientNet model to accomplish robust feature extraction, taking complex patterns and information from high-resolution RSIs, contributing to the accuracy of land use and land cover classification.
- Hyperparameter optimization of the dense EfficientNet model using RFDA algorithm using cross-validation helps to boost the predictive outcome of the proposed model for unseen data.
- Utilizes MSCAE model for the classification process to understand and represent multi-scale features in RSIs, enabling a wide-ranging interpretation of land cover classes.
- Apply the SOA for parameter selection in the MSCAE system assures an effectual and optimal configuration, further enhancing the algorithm's overall performance and adaptability.

The rest of the paper can be organized as follows. Section II gives the related works and section III provides the presented method. Next, section IV gives the results analysis and section V gives the conclusion of this paper.

II. RELATED WORKS

In [19], the authors suggested an interpretable DL model for LULC classification in RS utilizing the Shapley additive explanations (SHAPs) algorithm. It uses a compact-CNN approach for classifying the satellite images and after giving the outcomes to a SHAP deep explainer thus, enhancing the classification performances. Rajesh et al. [20] developed a novel approach to classify LISS IV satellite images by applying the DL technique. DL technique is to automatically extract several features with the absence of user intervention. The accuracy of classification using DL was quite enhanced by comprising object-related segmentation. The object-based deep feature learning technique through CNNs can be utilized for accurately classifying the RSI.

In [21], an innovative CD model based on CNN is developed not only to overcome the above-mentioned challenges but also to dramatically enhance the accuracy level. The recommended CNN-CD network includes 3 parallel channels namely, 1st and 2nd channels, correspondingly, extracting deep features under the original 1st and 2nd image and 3rd channel attention on the removal of modified deep features depends upon staking and distinguishing deep features. Tong et al. [22] introduced a method to use a deep framework attained at labelled LC databases for classifying unlabeled HRRS images. The fundamental notion can depend on DNN for giving the contextual data included in various kinds of land covers and suggest sample selection and a pseudo-labelling method for enhancing the transferability of deep frameworks. More accurately, the DCNNs have a primary pre-training with a good-interpreted land cover database, specified as the source information.

Ekim and Sertel [23] overcome this issue by using three various DNN Ensemble (DNNE) techniques and making a comparison outcome for the LCLU classification tasks. DNNE facilitates the development of the solution of DNNs

by making sure a variety of methods, can be incorporated. Jeyavathana [24] recommended a new concept for classifying the Landsat-8 multi-spectral RSIs. DL and ML approaches act as highly significant tools for classifying these multi-spectral images into different land cover types like forest, cropland, water, cloud, plantation, and urban area. In this study, LSTM of RNN can be utilized for achieving higher level classification accuracy and resolving the memory problems that could happen in an internal state. The developed effort demonstrated that; it resolved the gradient issue of RNNs.

In [25], a hybrid hot encoding VGG19 DL algorithm was suggested. The TL technique is employed for transferring the trained data, trained by the RestNet-50 algorithm to the presented HGVGG-19 approach. The aerial and satellite images are together from different sources and classification depends on features. Afterwards, for the classification of the images utilizing the MLR classifier, the class prediction has been expected through DT. Alhassan et al. [26] introduced a method for LULC mapping from multi-spectral satellite images utilizing the DL approach. The words map generation and satellite image classification, while utilized interchangeably have exact meanings in the area of RS. This method of LULC mapping is attained by employing DNNs pre-training on the fine-tuned target database and ImageNet large-scale visual identification competition databases.

Olimov et al. [27] introduced a simple algorithm named ''Deep Clean Before Training Net'' (DCBT-Net) that depends on cleaning wrongly labelled data points utilizing the data from eigenvalues of the Laplacian matrix attained from similarities among the data instances. The cleaned information was trained by DCNN to obtain remarkable outcomes. Olimov et al. [28] present an efficient and accurate DCNN method (AEDCN-Net) that relies on detailed pre-processing steps and a resourceful method. The AEDCN-Net utilizes bottleneck, atrous, and asymmetric convolutional-based residual skip connections from the encoder path that decrease the amount of trainable parameters and FLOPs for learning feature representation with the large receptive domain. Olimov et al. [29] developed a robust, efficient, and fast network (REF-Net) which fuses the encoder and decoder paths. Particularly, the contraction path exploits the combination of dilated and asymmetric convolutional layers with bottleneck layers and skip connections, whereas the decoder path assists from the nearest neighbor interpolation technique that demands no trainable parameters for restoring the new image dimensional.

III. THE PROPOSED MODEL

In this study, we have considered developing an automated LULCC-RFDADL algorithm on RSIs. The LULCC-RFDADL method has aimed to identify the different types of land cover on RSIs. In the proposed LULCC-RFDADL technique, several stages of operations are involved namely dense EfficientNet feature extractor, RFDA-related hyperparameter tuning, MSCAE classification, and SOA-based parameter optimization. Fig. 1 represents the overall flow of the LULCC-RFDADL methodology.

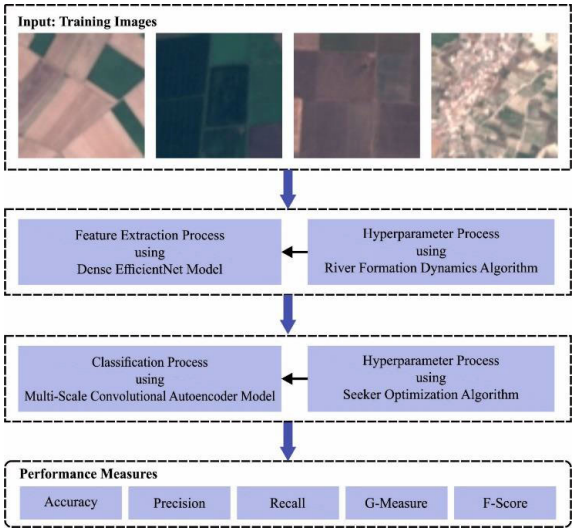

FIGURE 1. Overall flow of the LULCC-RFDADL method.

A. OPTIMAL FEATURE EXTRACTOR

To develop a suitable set of feature vectors, the dense EfficientNet technique is applied. An innovative dense CNN technique was projected in the work that is a mix of pre-training EfficientNetB0 with dense layers [30]. EfficientB0 contains 230 layers and 7 MBConv blocks. Every layer in this framework exploits the outcome feature mapping (FM) of prior levels as the input FM. The dense block model has collected convolutional layers of a similar dimensional as the input FM in EfficientNet. Dense block proceeds benefit of the previous convolutional layer outcome FM for generating additional FMs with any convolution kernels. This CNN technique retrieved 150×150 improved MRI data. The dense EfficientNet takes a different dense and dropout layer. The dense layer can be a vital layer that gives all outcomes in the prior layer for all its neurons, all neurons give one outcome to the following layers. The dropout layer has been employed for minimizing the thinning or size of the network in training and avoiding overfitting. At last, the researchers employed a dense layer containing 4 FC neurons along with the Softmax output layer to categorize as well as compute the likelihood score in every class.

For the optimum choice of the hyperparameters related to the Dense-EfficientNet architecture, the RFDA is employed. The principles of RFD stimulate the process of riverbed development. A group of drops positioned at the beginning will be exposed to the gravitational force that attracts these drops toward the centre of the Earth [31]. Therefore, the drops will suffer distribution throughout the environment looking for the lowest point in the sea. The RFD algorithm is described in the following. There is a task of the soil counts to all the nodes. Once the drops move, they can erode the path or deposit the accepted sediment (increasing the altitude of the node). Drops will move one until they arrive at the

target, or it can travel the maximal node counts. The overall node counts in an environment constitute the abovementioned maximal node counts. Eqs. (1) to (3) expresses the probability $Pk(i, j)$ which drops k that reside in i^{th} nodes would select the j^{th} nodes:

$$P_k(i,j) = \begin{cases} \dfrac{gradient\ (ij)}{total}, & for\ j \in v_k\ (i)\ , \\ \dfrac{|gradient\ (ij)|}{total}, & for\ j \in U_k\ (i)\ , \\ \dfrac{\delta}{total}, & for\ j \in F_k\ (i)\ , \end{cases} \tag{1}$$

where

$$gradient\ (i,j) = \frac{attitude(i) - attitude\ (j)}{distance\ (i,j)}, \tag{2}$$

$$\begin{aligned} total = (&\sum_{l \in V_k(i)} \text{gradient}\ (i,l)) \\ &+ \left(\sum_{l \in U_k(i)} \frac{\omega}{|gradient\ (i,l)|} \right) \\ &+ \left(\sum_{l \in F_k(i)} \delta \right) \end{aligned} \tag{3}$$

where $Uk(i)$ represents the neighbouring node-set, which can be a negative gradient (node j altitude is superior to node i), $Vk(i)$ denotes the adjacent node-set which can be a positive gradient (node i altitude can be greater than node j), and $Fk(i)$ shows neighbor with flat gradient. ω and δ coefficient has fixed value.

When the drops are completed moving, the implementation of a process of erosion on the travelled path over the minimization of nodes' altitude depends upon the gradient to the consecutive nodes. Based on Eq. (4), the amount of erosion for all the pairs of i and j nodes is reliant on the amount of D drops used, the amount of nodes in graph N, along with the particular erosion coefficient E.

$$\begin{aligned} \forall i, j \in Path_k, altitude(i) = altitude(i) - &\frac{B}{(N-1)\ .D} \\ &.gradient(i,j). \end{aligned} \tag{4}$$

Now, $Pathk$ indicates the drop k's traversed path. Moreover, once the drop stops, these deposits a fraction of the sediment accepted and end up vanishing for a residual part of the iterative model. Meanwhile, it reduces the probability of transition near blind alleys, which might lead to lessening the bad path.

There is the addition of specific and least sediment amount to each node on each iteration completion. This is for the avoidance of situations where each altitude was closer to 0 since it might lead to negligible gradient and ruination of the path formed:

$$\forall i \in G \wedge i \neq goal, altitude(i) = altitude(i) + \frac{erosionProduced}{N-1}. \tag{5}$$

where G indicates the node set of graphs used, the *goal* represents the goal node, and *erosion produced* refers to the amount of erosion produced in the existing iteration,

$$\sum_{path_k} El\ (N-1)\ .D.gradient\ (i,j)\ , \forall k \in drops. \tag{6}$$

The process repeats until arrival at the final condition and this represents the drops that are moving along a similar path. Maximum iteration was defined for the reduction of computation time and condition to validate if the former n loops made some developments in the performance.

B. IMAGE CLASSIFICATION

In this work, the MSCAE algorithm was executed for the classification of images. The infrastructure of the convolutional auto-encoder (CAE) is acquired by exchanging the matrix functions in it with the convolution function [32]. Generally, the features can be extracted utilizing convolution kernels of similar dimensions in the CAE. These works proved the efficiency of the multiscale convolution network. The presented MSCAE utilizes 3 convolutional paths for extracting features of input data. This is appropriate to observe that the 3 convolution paths employ 1D convolutional with distinct convolution kernel sizes, to create complete utilization of the local and global data of the input data for further efficient extraction features. Fig. 2 displays the architecture of MSCAE.

At the encoded phase, the 3 pathways carry out parallel convolutional as well as pooling tasks under the input data. Let $E^i_{n,m}$ signifies the n^{th} channel of convolution layer data from the m^{th} encoded block from the $i^{th}(i = 1, 2, 3)$ pathway and N^e_m denotes the count of convolution layer channels from the m^{th} encoded block. Afterwards, the 1D convolutional function was written as:

$$Z^{i,e}_{k,m+1} = f_r \left(\sum_{n=1}^{N^e_m} E^i_{n,m} * w^{i,e}_{k,n,m} + b^{i,e}_m \right) \tag{7}$$

whereas $*$ denotes the 1D convolutional function. $w^{i,e}_{k,n,m}$ refers to the weighted k^{th} convolution kernel of the convolution layer from the m^{th} encoded block. $b^{i,e}_m$ denotes bias. f_r implements the ReLU activation function. $Z^{i,e}_{k,m+1}$ stands for the k^{th} channel data of convolution layer function outcome from the m^{th} encoded block of the i^{th} pathway.

Afterwards executing a convolutional function, a downsampling function was carried out on the convolutional outcome employing maximal pooling for reducing the data size. The pooling outcome $I^i_{k,m+1}$ for k^{th} channel data of pooling layer from the m^{th} encoded block of i^{th} pathway is formulated as:

$$E^i_{k,m+1} = P\left(Z^{i,e}_{k,m+1}, p_{m'} s_m\right) \tag{8}$$

In which, P implies the maximal pooling function, p_m denotes the dimensional of the pooling window from the m^{th} convolutional block, and s_m represents the stride.

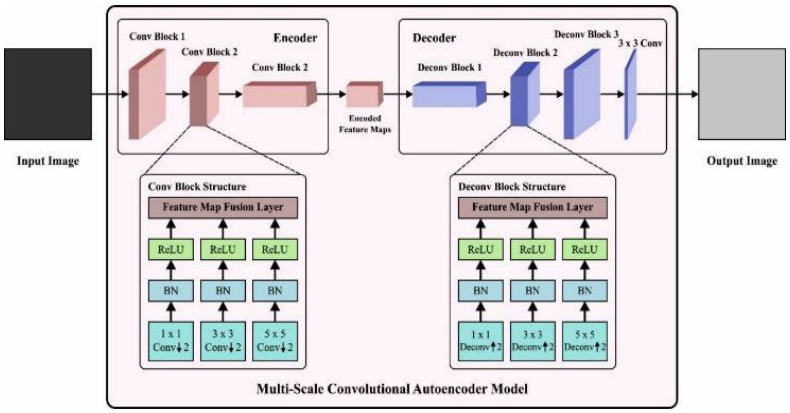

FIGURE 2. Architecture of MSCAE.

Afterwards, the input data can be removed by L-encoded blocks of features, the Flatten layer was employed to change the higher-level representations into 1D data that can be the inputs to FC block for HI extraction. Afterwards, the HI extraction is formulated as:

$$\hat{y} = \sigma\left(W_{fe}X_e + b_{fe}\right) \tag{9}$$

In which, $\hat{y}$ stands for the HI extraction, b_{fe} and w_{fe} demonstrate the encoding partial bias and weight, correspondingly, and σ refers to the Sigmoid activation process.

At the decoded phase, there exist 3 parallel paths. The decoded section in the FC block carries out a dimension expansion function on the HI extraction, and the outcomes accomplished are depicted as:

$$X_d = f_r\left(w_{fd}\hat{y} + b_{fd}\right) \tag{10}$$

In which, X_d indicates the outcome of the decoded part in the FC block, b_{fd} and w_{fd} represent the bias and weight of the decoded part in the FC block, correspondingly, and f_r demonstrates the ReLU activation function.

C. PARAMETER TUNING USING SOA

Lastly, the SOA has been employed to select the parameter values of the MSCAE algorithm. SOA is a SI optimizer system derived from the cooperative behaviours of humans [33]. Consider the group as a search population and the location of the searcher as the solution candidate to optimizer problems. Later, this defines the search way by mimicking performances, namely proactivity, egoism, and altruism. Also, it defines the step length by mimicking the intellectual property of human searches for completing the coordinate iteration. Eqs. (11) to (13) provide the mathematical expression of the simulating behaviors:

$$\vec{d}_{i,ego}(t) = sign\left(\vec{p}_{i,best} - \vec{x}_i(t)\right) \tag{11}$$

$$\vec{d}_{i,alt}(t) = sign\left(\vec{g}_{i,best} - \vec{x}_i(t)\right) \tag{12}$$

$$\vec{d}_{i,pro}(t) = \begin{cases} \vec{d}_i(t-1), & f\left(\vec{x}_i(t)\right) > f\left(\vec{x}_i(t-1)\right) \\ -\vec{d}_i(t-1), & f\left(\vec{x}_i(t)\right) \le f\left(\vec{x}_i(t-1)\right) \end{cases} \tag{13}$$

where $\vec{d}_{i,ego}$ indicates the egoistic direction, $\vec{d}_{i,alt}$ shows the altruistic direction and $\vec{d}_{i,pro}$ represents the pro-activeness direction that can be searched by the i^{th} seekers once the iterative algebra is t.$\vec{g}_{ibest}$ means that the entire population of searchers has found the optimum location, $\vec{p}$ shows that the i^{th} searchers have found the optimum location. $f\left(\vec{x}_i(t)\right)$ means the fitness value that can be searched by the i^{th} searchers once the iterative algebra is t. Likewise, $f\left(\vec{x}_i(t-1)\right)$ implies the FF once the iterated algebra refers to $t-1$. The stochastic weight averaging technique determines the last search direction assuming the abovementioned 3 directions:

$$\vec{d}_i(t) = sign\left(\omega_0\vec{d}_{pro} + \varphi_1\vec{d}_{ego} + \varphi_2\vec{d}_{alt}\right) \tag{14}$$

$$\omega_0 = \omega_{0\max} - \frac{t\left(\omega_{0\max} - \omega_{0\min}\right)}{t_{\max}} \tag{15}$$

where ω_0 denotes the inertia weighted; $sign(\cdot)$ indicates the symbolic function; φ_1 and φ_2 are randomly generated values with mutual independence that follow zero and one distribution; and $\omega_{0\,\max}$ and $\omega_{0\,\min}$ imply the maximal and minimal inertia weight, correspondingly.

The membership function (MF) of step length was formulated by the Gaussian function whose properties are reliable with the characteristics of the human mind.

$$u(\alpha) = e^{-\frac{\alpha^2}{2\delta^2}} \tag{16}$$

In Eq. (16), u refers to the membership degree whose value is lower than 0.0111, the step length α is the fuzzy reasoning input that lies outside $[-3\delta, 3\delta]$.

$$u_i = u_{\max} - \frac{s - I_i}{s - 1}\left(u_{\max} - u_{\min}\right) \tag{17}$$

In Eq. (17), $u_{\max}$ refers to the maximal degree of membership, which is equivalent to or closer to 1. Moreover, $u_{\min}$ denotes the minimal degree, which is 0.0111. s shows the level of an entire population, and I_i represents the serial number of i^{th} seekers after arranging the fitness in ascending sequence. For improving the arbitrariness of every size and local search model's capability, the step length α_{ij} can be given as,

$$\alpha_{ij} = \tau_i\sqrt{-ln\left(u_{ij}\right)} \tag{18}$$

FIGURE 3. Sample images.

In Eq. (18), α_{ij} denotes the step length from the following movement, the parameter τ_i represents the Gaussian MF and is formulated using Eq. (9) and (10), and u_{ij} shows the step length membership that can be searched by i^{th} seekers at j^{th} size and is directed to distribute arbitrarily:

$$\vec{\tau}_i = w_1 \cdot abs\left(\vec{g}_{best} - \vec{x}_{rand}\right) \tag{19}$$

$$w_1 = \frac{t_{\max} - t}{t_{\max}} \tag{20}$$

where $\vec{g}_{best}$ refers to the location of optimum seekers in the present population, $\vec{x}_{rand}$ shows the random position from the search range of the existing population. Next, w represents the weight that varies with the iteration counts for improving the accuracy of the search process, and $t_{\max}$ indicates the maximal iterative algebra. The degree was connected to the fitness of the individual searcher to resolve the uniform degree of membership. After attaining the direction and step length, the updated location of the seeker can be given as

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t)\, d_{ij}(t) \tag{21}$$

The SOA approach grows an FF for attaining greater classifier outcomes. It resolves a positive integer for signifying the better solution for candidate performances. During this work, the minimized classifier rate of errors can be regarded as FF, as provided in Eq. (22).

$$\begin{aligned} fitness\,(x_i) &= ClassifierErrorRate\,(x_i) \\ &= \frac{No.of\ misclassified\ instances}{Total\ No.of\ instances} * 100 \end{aligned} \tag{22}$$

IV. RESULTS AND DISCUSSION

The proposed method can be examined utilizing the Python 3.8.5 tool on 16GB RAM, PC i5-8600k, 1TB HDD, GeForce 1050Ti 4GB, and 250GB SSD. The parameter settings are specified namely dropout: 0.5, rate of learning: 0.01, batch size: 5, epoch count: 50, and activation: ReLU. In this section, the LULC classifier results of the LULCC-RFDADL algorithm are examined with the EuroSat Database [34] comprising 10 classes. Each class holds a total of 500 samples. Fig. 3 exhibits the sample images.

Fig. 4 displays the classifier solution of the LULCC-RFDADL algorithm on the test database. Figs. 4a-4b show the confusion matrices achieved by the LULCC-RFDADL system on 70% of TRS and 30% of TSS. The result stated that the LULCC-RFDADL algorithm has detected and classified all

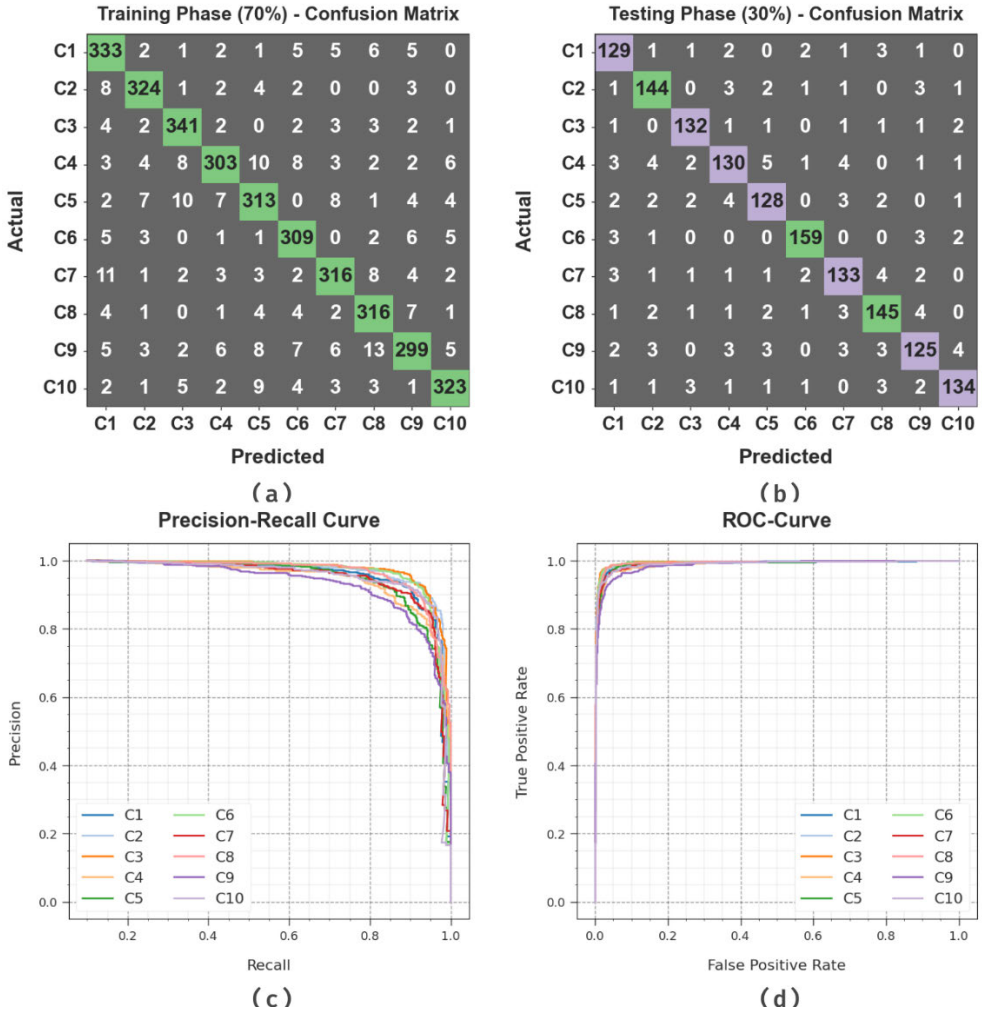

FIGURE 4. Performances of (a-b) 70% of TRS/30% of TSS, (c) PR_curve, and (d) ROC_curve.

10 classes correctly. Likewise, Fig. 4c exhibits the PR study of the LULCC-RFDADL system. The outcomes showed that the LULCC-RFDADL algorithm has gained greater PR solution on 10 classes. However, Fig. 4d represents the ROC outcome of the LULCC-RFDADL approach. The result revealed that the LULCC-RFDADL algorithm can be a proficient solution with higher ROC values with 10 classes.

In Table 1 and Fig. 5, the LULC classifier analysis of the LULCC-RFDADL system is portrayed. The outcomes inferred that the LULCC-RFDADL methodology properly recognizes ten classes. On 70% of the TRS, the LULCC-RFDADL technique accomplishes average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 98.15%, 90.79%, 90.79%, 90.76%, and 90.78% respectively. Also, on 30% of TSS, the LULCC-RFDADL approach attains average $accu_y$, $prec_n$, $reca_l$, F_{score}, and $G_{measure}$ of 98.12%, 90.54%, 90.56%, 90.54%, and 90.55% correspondingly.

Fig. 6 demonstrates the TR_accu_y and VL_accu_y of the LULCC-RFDADL system. The TR_accu_y can be well defined by the calculation of the LULCC-RFDADL approach on the database of TR but the VL_accu_y has been calculated by assessing the solution with a separate TS database. The obtained solutions exhibit that TR_accu_y and VL_accu_y increase with enrichment in epochs. Therefore, the solution of the LULCC-RFDADL methodology gets enhancement in the dataset of TR and TS with a rise of epoch count.

In Fig. 7, the *TR_loss* and *VR_loss* curves of the LULCC-RFDADL methodology can be revealed. The *TR_loss* exhibits the error among the original and predicted values with the database of TR. The *VR_loss* shows the evaluation of the outcome of the LULCC-RFDADL approach with individual validating data. The attained outcome shows the *TR_loss* and *VR_loss* diminish with increasing epochs. This exposed the increased effectiveness of the LULCC-RFDADL system and its ability to create correct classifications. The diminished result of *TR_loss* and *VR_loss* reveals the higher performance of the LULCC-RFDADL methodology on capturing designs and correlations.

TABLE 1. LULC classifier outcome of LULCC-RFDADL model at 70% of TRS/30% of TSS.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	$G_{Measure}$
TRS (70%)					
C1	97.97	88.33	92.50	90.37	90.39
C2	98.74	93.10	94.19	93.64	93.64
C3	98.63	92.16	94.72	93.42	93.43
C4	97.94	92.10	86.82	89.38	89.42
C5	97.63	88.67	87.92	88.29	88.29
C6	98.37	90.09	93.07	91.56	91.57
C7	98.11	91.33	89.77	90.54	90.55
C8	98.23	89.27	92.94	91.07	91.08
C9	97.46	89.79	84.46	87.05	87.09
C10	98.46	93.08	91.50	92.29	92.29
Average	98.15	90.79	90.79	90.76	90.78
TSS (30%)					
C1	98.13	88.36	92.14	90.21	90.23
C2	98.20	90.57	92.31	91.43	91.43
C3	98.80	92.96	94.29	93.62	93.62
C4	97.53	89.04	86.09	87.54	87.55
C5	97.93	89.51	88.89	89.20	89.20
C6	98.87	95.21	94.64	94.93	94.93
C7	97.93	89.26	89.86	89.56	89.56
C8	97.93	90.06	90.62	90.34	90.34
C9	97.47	88.03	85.62	86.81	86.81
C10	98.40	92.41	91.16	91.78	91.78
Average	98.12	90.54	90.56	90.54	90.55

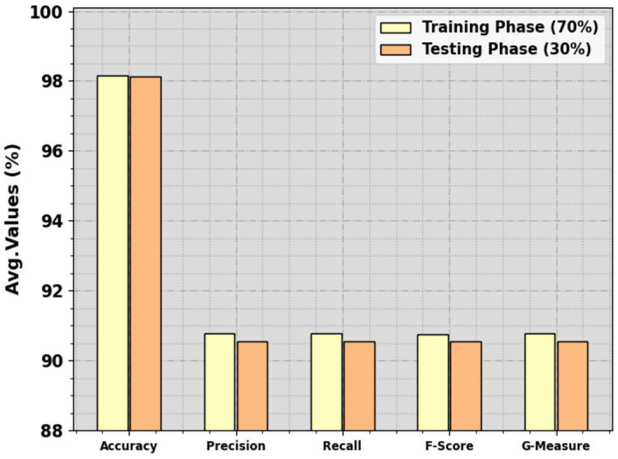

FIGURE 5. Average outcome of LULCC-RFDADL methodology on 70% of TRS/30% of TSS.

FIGURE 6. $Accu_y$ curve of the LULCC-RFDADL approach.

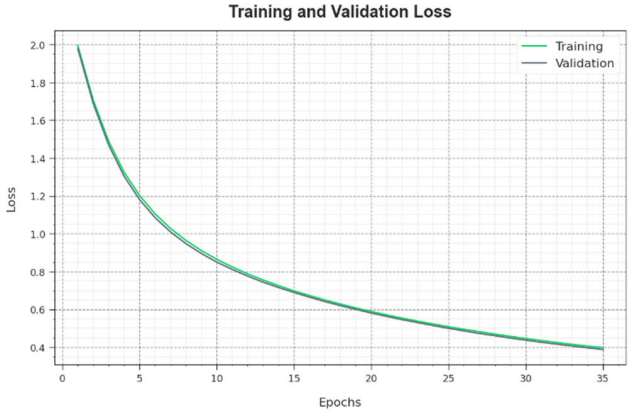

FIGURE 7. Loss curve of the LULCC-RFDADL system.

TABLE 2. Comparative outcome of LULCC-RFDADL algorithm with recent DL approaches [13], [35], [36].

Model	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
Shallow CNN	87.96	85.40	81.64	84.39
GoogleNet	96.02	85.90	68.78	80.39
DenseNetl21	96.64	71.12	55.20	77.72
Inception V3	96.86	86.80	74.97	81.93
ResNet50	96.43	82.87	56.84	82.94
VGGl6	96.65	83.61	60.27	82.44
GeoSystemNet	94.65	89.32	68.28	88.38
LULCC-RFDADL	98.15	90.79	90.79	90.76

The LULC outcomes of the LULCC-RFDADL system can be compared with recent DL models in Table 2 and Fig. 8 [13], [35], [36]. The experimental values defined that the Shallow CNN model reached lower performance whereas the GeoSystemNet model achieved slightly boosted results. At the same time, the GoogleNet, DenseNetl21, Inception V3, ResNet50, and VGGl6 models obtained certainly boosted performance. Nevertheless, the LULCC-RFDADL methodology gets supreme outcomes with a higher $accu_y$ of 98.15%, $prec_n$ of 90.79%, $reca_l$ of 90.79%, and F_{score} of 90.76%.

Finally, in Table 3 and Fig. 9, a comparative computation time (CT) result of the LULCC-RFDADL system can be given. The obtained outcome exhibits that the LULCC-RFDADL technique shows promising performance with a least CT of 0.89s. Conversely, the shallow CNN, GoogleNet, DenseNet121, InceptionV3, ResNet50, VGG16, and GeoSystemNet approaches achieved worse solutions with increased CT of 2.05s, 3.18s, 3.10s, 2.17s, 2s, 2.20s, and 2.18s respectively. Therefore, the LULCC-RFDADL

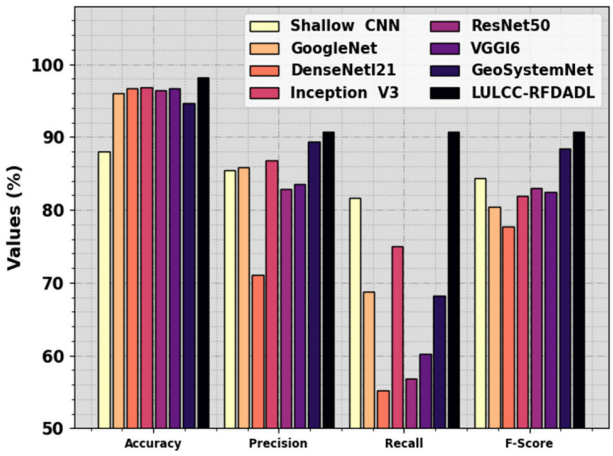

FIGURE 8. Comparative outcome of LULCC-RFDADL algorithm with recent DL approaches.

TABLE 3. CT analysis of the LULCC-RFDADL model with recent DL methods [13], [35], [36].

Model	Computational Time (sec)
Shallow CNN	2.05
GoogleNet	3.18
DenseNet121	3.10
Inception V3	2.17
ResNet50	2.00
VGG16	2.20
GeoSystemNet	2.18
LULCC-RFDADL	0.89

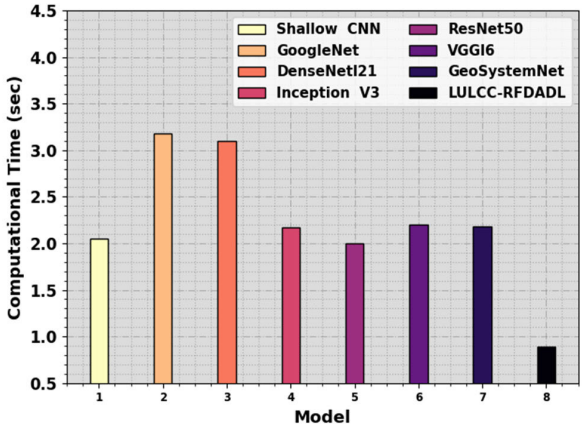

FIGURE 9. CT analysis of the LULCC-RFDADL method with recent DL models.

technique can be used for the automated LULC classification process.

V. CONCLUSION

In this study, we have designed an automated LULCC-RFDADL algorithm on RSIs. The aim of the LULCC-RFDADL system can be to identify the different types of land cover on RSIs. In the presented LULCC-RFDADL method, several stages of operations have been included namely dense EfficientNet feature extractor, RFDA-based hyperparameter tuning, MSCAE classification, and SOA-related parameter optimizer. The development of RFDA and SOA supports optimally choosing the hyperparameters related to the dense EfficientNet and MSCAE models respectively. The achieved outcomes of the LULCC-RFDADL algorithm were examined under the EuroSAT database. The simulation outcomes show the promising result of the LULCC-RFDADL algorithm with other methods with respect to diverse metrics. The LULCC-RFDADL model improves classification accuracy, making it valuable for applications in environmental monitoring, urban planning, and natural resource management.

The limitations of the LULCC-RFDADL algorithm include probable sensitivity to differences in data quality and characteristics, as well as the requirement for large and diverse training datasets to assure robust performance across various geographical regions. In addition, the computational complexity associated with the integration of RFDA and dense EfficientNet for hyperparameter tuning may pose challenges for real-time applications or resource-constrained environments. Future work focus on relating RSIs with other geospatial data sources like geographical information systems (GIS) and weather information, for enhancing the accuracy and contextual understanding of land cover classification. Moreover, future work is to expand the database to contain RSIs from various geographical regions and distinct seasons are enhance the abilities of the model for generalizing across distinct landscapes and weather conditions.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/134/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R114), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2024R459), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444). This study is partially funded by the Future University in Egypt (FUE).

REFERENCES

[1] O. Aidantausta and P. Asman, "Land use/land cover classification from satellite remote sensing images over urban areas in Sweden: An investigative multiclass, multimodal and spectral transformation, deep learning semantic image segmentation study," M.S. thesis, 2023.

[2] M. Digra, R. Dhir, and N. Sharma, "Land use land cover classification of remote sensing images based on the deep learning approaches: A statistical analysis and review," *Arabian J. Geosci.*, vol. 15, no. 10, p. 1003, May 2022.

[3] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (ExViT) for land use and land cover classification: A multimodal deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023.

[4] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," *Remote Sens.*, vol. 12, no. 15, p. 2495, Aug. 2020.
[5] S. Puttinaovarat, K. Khaimook, and P. Horkaew, "Land use and land cover classification from satellite images based on ensemble machine learning and crowdsourcing data verification," *Int. J. Cartogr.*, pp. 1–21, Mar. 2023.
[6] R. Saini and S. Rawat, "Land use land cover classification in remote sensing using machine learning techniques," in *Proc. 1st Int. Conf. Innov. High Speed Commun. Signal Process. (IHCSP)*, Mar. 2023, pp. 99–104.
[7] A. Alem and S. Kumar, "Deep learning methods for land cover and land use classification in remote sensing: A review," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim., Trends Future Directions (ICRITO)*, Jun. 2020, pp. 903–908.
[8] K. Sanjar, O. Bekhzod, J. Kim, A. Paul, and J. Kim, "Missing data imputation for geolocation-based price prediction using KNN–MCF method," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 227, Apr. 2020.
[9] B. Olimov, S. Karshiev, E. Jang, S. Din, A. Paul, and J. Kim, "Weight initialization based-rectified linear unit activation function to improve the performance of a convolutional neural network model," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 22, p. e6143, Nov. 2021.
[10] B. Olimov, B. Subramanian, R. A. A. Ugli, J.-S. Kim, and J. Kim, "Consecutive multiscale feature learning-based image classification model," *Sci. Rep.*, vol. 13, no. 1, p. 3595, Mar. 2023.
[11] A. A. U. Rakhmonov, B. Subramanian, B. Olimov, and J. Kim, "Extensive knowledge distillation model: An end-to-end effective anomaly detection model for real-time industrial applications," *IEEE Access*, vol. 11, pp. 69750–69761, 2023.
[12] A. A. Darem, A. A. Alhashmi, A. M. Almadani, A. K. Alanazi, and G. A. Sutantra, "Development of a map for land use and land cover classification of the northern border region using remote sensing and GIS," *Egyptian J. Remote Sens. Space Sci.*, vol. 26, no. 2, pp. 341–350, Aug. 2023.
[13] R. Naushad, T. Kaur, and E. Ghaderpour, "Deep transfer learning for land use and land cover classification: A comparative study," *Sensors*, vol. 21, no. 23, p. 8083, Dec. 2021.
[14] S. Talukdar, P. Singha, S. Mahato, Shahfahad, S. Pal, Y.-A. Liou, and A. Rahman, "Land-use land-cover classification by machine learning classifiers for satellite observations—A review," *Remote Sens.*, vol. 12, no. 7, p. 1135, Apr. 2020.
[15] J. Wang, M. Bretz, M. A. A. Dewan, and M. A. Delavar, "Machine learning in modelling land-use and land cover-change (LULCC): Current status, challenges and prospects," *Sci. Total Environ.*, vol. 822, May 2022, Art. no. 153559.
[16] Y. Liu, Z. Yan, J. Tan, and Y. Li, "Multi-purpose oriented single nighttime image haze removal based on unified variational retinex model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 4, pp. 1643–1657, Apr. 2023.
[17] Y. Liu, Z. Yan, S. Chen, T. Ye, W. Ren, and E. Chen, "NightHazeFormer: Single nighttime haze removal using prior query transformer," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 4119–4128.
[18] Z. Jin, H. Feng, Z. Xu, and Y. Chen, "Nighttime image dehazing by render," *J. Imag.*, vol. 9, no. 8, p. 153, Jul. 2023.
[19] A. Temenos, N. Temenos, M. Kaselimi, A. Doulamis, and N. Doulamis, "Interpretable deep learning framework for land use and land cover classification in remote sensing using SHAP," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.
[20] S. Rajesh, T. G. Nisia, S. Arivazhagan, and R. Abisekaraj, "Land cover/land use mapping of LISS IV imagery using object-based convolutional neural network with deep features," *J. Indian Soc. Remote Sens.*, vol. 48, no. 1, pp. 145–154, Jan. 2020.
[21] S. T. Seydi, M. Hasanlou, and M. Amani, "A new end-to-end multi-dimensional CNN framework for land cover/land use change detection in multi-source remote sensing datasets," *Remote Sens.*, vol. 12, no. 12, p. 2010, Jun. 2020.
[22] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111322.
[23] B. Ekim and E. Sertel, "Deep neural network ensembles for remote sensing land cover and land use classification," *Int. J. Digit. Earth*, vol. 14, no. 12, pp. 1868–1881, Dec. 2021.
[24] R. B. Jeyavathana, "Land use and land cover classification using Landsat-8 multispectral remote sensing images and long short-term memory-recurrent neural network," *AIP Conf.*, vol. 2452, no. 1, Nov. 2022, Art. no. 070001.
[25] J. Jagannathan and C. Divya, "Deep learning for the prediction and classification of land use and land cover changes using deep convolutional neural network," *Ecol. Informat.*, vol. 65, Nov. 2021, Art. no. 101412.
[26] V. Alhassan, C. Henry, S. Ramanna, and C. Storie, "A deep learning framework for land-use/land-cover mapping and analysis using multispectral satellite imagery," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 8529–8544, Jun. 2020.
[27] B. Olimov, J. Kim, and A. Paul, "DCBT-Net: Training deep convolutional neural networks with extremely noisy labels," *IEEE Access*, vol. 8, pp. 220482–220495, 2020.
[28] B. Olimov, S.-J. Koh, and J. Kim, "AEDCN-Net: Accurate and efficient deep convolutional neural network model for medical image segmentation," *IEEE Access*, vol. 9, pp. 154194–154203, 2021.
[29] B. Olimov, J. Kim, and A. Paul, "REF-Net: Robust, efficient, and fast network for semantic segmentation applications using devices with limited computational resources," *IEEE Access*, vol. 9, pp. 15084–15098, 2021.
[30] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, "Brain tumor classification using dense efficient-net," *Axioms*, vol. 11, no. 1, p. 34, Jan. 2022.
[31] S. S. Ahmad, R. Rani, I. Wattar, M. Sharma, S. Sharma, R. Nair, and B. Tiwari, "Hybrid recommender system for mental illness detection in social media using deep learning techniques," *Comput. Intell. Neurosci.*, vol. 2023, pp. 1–14, Jul. 2023.
[32] Z. Ye, Q. Zhang, S. Shao, T. Niu, and Y. Zhao, "Rolling bearing health indicator extraction and RUL prediction based on multi-scale convolutional autoencoder," *Appl. Sci.*, vol. 12, no. 11, p. 5747, Jun. 2022.
[33] S. Zhong, C. Xu, D. Sun, L. Duan, and J.-A. Duan, "Optimization of coupling efficiency in butterfly optical communication laser based on chaotic adaptive seeker optimization algorithm," *Micromachines*, vol. 14, no. 7, p. 1417, Jul. 2023.
[34] Accessed: Jul. 15, 2023. [Online]. Available: https://www.kaggle.com/datasets/apollo2506/eurosat-dataset
[35] C. Zhang, Y. Chen, X. Yang, S. Gao, F. Li, A. Kong, D. Zu, and L. Sun, "Improved remote sensing image classification based on multi-scale feature fusion," *Remote Sens.*, vol. 12, no. 2, p. 213, Jan. 2020.
[36] A. Stateczny, S. M. Bolugallu, P. B. Divakarachari, K. Ganesan, and J. R. Muthu, "Multiplicative long short-term memory with improved mayfly optimization for LULC classification," *Remote Sens.*, vol. 14, no. 19, p. 4837, Sep. 2022, doi: 10.3390/rs14194837.