# An Integrated Parallel Inner Deep Learning Models Information Fusion With Bayesian Optimization for Land Scene Classification in Satellite Images

Ameer Hamza, Muhammad Attique Khan, *Member, IEEE*, Shams ur Rehman, Hussain Mobarak Albarakati, Roobaea Alroobaea, Abdullah M. Baqasah, Majed Alhaisoni, and Anum Masood

***Abstract*—Classification of remote scenes in satellite imagery has many applications, such as surveillance, earth observation, etc. Classifying high-resolution remote sensing images in machine learning is a big challenge nowadays. Several automated techniques based on machine learning and deep learning have been introduced in the literature; however, these techniques fail to perform for complex texture images, complex backgrounds, and small objects. In this work, we proposed a new automated technique based on the inner fusion of two deep learning models and feature selection. A new network is designed at the initial phase based on the inner-level fusion of two networks and combined weights. After that, hyperparameters have been initialized based on the Bayesian optimization (BO). Usually, the hyperparameters have been initialized through a manual approach, but that is not an efficient way of selection. After that, the designed model is trained and extracted deep features from the deeper layer. In the last step, a poor–rich controlled entropy-based feature selection technique is developed for the best feature selection. The selected features are finally classified using machine learning classifiers. We performed the experimental process of the proposed architecture on three publically available datasets: Aerial image dataset (AID), UC-Merceds, and WHU-RS19. On these datasets, we obtained the accuracy of 96.3%, 95.6%, and 97.8%, respectively. Comparison is conducted with state-of-the-art techniques and shows improved accuracy.**

***Index Terms*—Deep learning, feature selection, machine learning, models fusion, remote sensing.**

## I. Introduction

In THE widest definition, remote sensing is a data-gathering technique that does not require the investigator to have direct physical contact with the object, substance, or phenomenon being studied. The whole procedure starts with the detection of radiation using sensor technologies, which is followed by the measuring of radiation at different wavelengths. This radiation is released or reflected by distant objects and materials [1]. Because remote sensing can provide observations on a local, regional, and even global scale, it is useful for a variety of applications, including monitoring land cover and use for agricultural purposes [2], supervising forest management [3], conducting geomorphological surveys [4], and determining the dynamics of water quality [5], among others. The availability of aerial images, which allows for a more in-depth analysis of the planet's surface, has led to a significant surge in interest in earth observation [6]. In the classification of aerial scenes [7], [8], each aerial image is evaluated using semantic labeling, a core component of the field of remote sensing, to assign it a meaningful label [9]. Aerial sceneries are often quite intricate, and there aren't many visual variations across groups [10]. For instance, common land-cover types are seen throughout several different scene classes. The classification of aerial images may be challenging since several diverse spatial and structural patterns are present [11].

It is required to create a scene representation for aerial imagery before attempting to ascertain the semantic labels used in aerial scene classification. Creating a reliable scene representation has received much attention recently, and several different aerial scene classification methods have been proposed [12]. These methods may be generally divided into two groups: those that address low-level scene features and those that address medium-level scene features. The common low-level approaches include the Invariant Feature Transform, the Local Binary Pattern, the color histogram, and the GIST [13], [14], [15], [16]. The scene representation that midlevel processes create includes the low-level local feature descriptors. The methods for midlevel coding include Bag of Visual Words, Spatial Pyramid Matching, Locality-Constrained Linear Coding, Probabilistic Latent Semantic Analysis, Latent Dirichlet

Manuscript received 24 July 2023; revised 17 September 2023; accepted 8 October 2023. Date of publication 13 October 2023; date of current version 31 October 2023. This work was supported by the Deanship of Scientific Research at Umm Al-Qura University under Grant 23UQU4330028DSR002. *(Corresponding author: Anum Masood.)*

Ameer Hamza and Shams ur Rehman are with the Department of CS, HITEC University, Taxila 47080, Pakistan (e-mail: ameer.hamza@hitecuni.edu.pk; shams.rehman@hitecuni.edu.pk).

Muhammad Attique Khan is with the Department of CS, HITEC University, Taxila 47080, Pakistan, and also with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 13-5053, Lebanon (e-mail: attique.khan@ieee.org).

Hussain Mobarak Albarakati is with the Computer Engineering Department, College of Computer and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia (e-mail: hmbarakati@uqu.edu.sa).

Roobaea Alroobaea is with the Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia (e-mail: r.robai@tu.edu.sa).

Abdullah M. Baqasah is with the Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia (e-mail: a.baqasah@tu.edu.sa).

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

Anum Masood is with the Department of Physics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway (e-mail: anum.masood@ntnu.no).

Digital Object Identifier 10.1109/JSTARS.2023.3324494

Allocation, Improved Fisher Kernel, and Vector of Locally Aggregated Descriptors [17], [18], [19]. Deep convolutional neural networks (DCNNs) [20], currently dominate the classification of most aerial images. The compelling depiction of the trait served as the inspiration for these CNNs. Because CNNs can provide strong feature representations to characterize the aerial image, classification performance, particularly for high-level approaches, has greatly improved. This is especially true for sophisticated operations. High-level methodologies extract impressive representations from aerial landscapes, unlike standard low-level methods, which depend on manually created features. High-level approaches may be contrasted with traditional low-level ones [21].

A researcher has recently presented several computer vision-based methods for classifying an object using satellite images. Some worked on nonhistorical buildings using airborne and satellite imagery [22]. The researcher used and worked on developing an automatic ship detection approach and a DL method for using satellite images [23]. For example, Duarte et al. [24] suggested an approach for satellite images using a deep learning approach. In this presented method, they implemented the DCNN technique for image classification of building damages. By this method, they gained 94% accuracy. The main drawback of this presented framework was only one multiresolution network did not improve the classification accuracy compared to the used benchmark. Pritt et al. [25] presented a method for satellite image classification using deep learning. In this presented methodology, they performed object and facility recognition using high-resolution and multispectral satellite images. From this technique, they obtained 95% accuracy. The dark side of this method was the state-of-the-art object detection method, which is not well for satellite images. Gao et al. [26] a region-based deep learning approach is suggested to segment satellite images. In this presented method, they used rooftop detection by using the segmentation approach. From this method, they obtained 92% accuracy. This presented method could not avoid the speckle-like error sometimes found in the segmentation model. Rostami et al. [27] demonstrated using deep learning techniques for fire detection with Landsat-8 satellite imagery. In this presented method, they used CNN multiscale detection for AFD in the Landsat-8 dataset. Consequently, they succeeded with 95% accuracy. This presented method's limitation was detecting fires of varying sizes and shapes over challenges test shape. Yosmaoglu et al. [28] presented a road network generation using satellite images. The presented method evaluates and compares the Resnet and U-net generation models. As a result, they achieved 99% accuracy. Lim et al. [29] presented a dead pine tree detection using a deep learning method. In this presented method, they used aerial vehicle and object detection deep learning to solve the problem. As a result, they achieved 99% accuracy. Ch et al. [30] presented a method for ECDSA-based water bodies using satellite images. They employed the U-Net model to achieve data integrity by using the security feature elliptic curve electronic signature algorithm. Therefore, they obtained 94% accuracy. This technique's main flaw was extending this model into video input. Najar et al. [31] demonstrated an approach for coastal bathymetry using deep learning approaches. From this presented method, they used Sentinel-2 satellite imagery and multiple bathymetry to train the deep learning model. As a result, they achieved and predicted 50% accuracy. One limitation of this approach was the selection of data based on certain dates and the need to train on application sites. Kaur et al. [32] introduced a transfer learning-based approach for automatically detecting and tracking hurricanes using satellite imagery. In this presented method, they utilized a transfer learning-based model. Thus, they gained 95% accuracy. The limitation of this method was made more generalizable by including images and another hurricane. Zhuang et al. [33] presented a method for semantic guidance transfer-based method by using satellite images. In this presented method, the UAV-based geo localization dataset. As a result, they achieved 8% more improvement in accuracy. The limitation of this method was a lot of information would be lost when using this model. Zhang et al. [34] presented a building height extraction using satellite images. In this presented method, the researchers used a stereo-matching technique coupled with a DSM-based approach for predicting bottom elevation. As a result, they improved the accuracy as compared to other method. Ul Ain Tahir et al. [35] presented a method for wildfire detection using deep learning. In the presented method, they utilized YOLOVv5-based deep learning based model. As a result, they achieved 94% F1-score.

Hasan et al. [36] presented a novel-based resource allocation technique for 5G heterogeneous networks. In this presented work, the authors designed a new biogeography-based dynamic subcarrier allocation algorithm for minimizing the cross-tire subcarrier snooping problems in MeNB and HeND. They achieved 88.1% outage and 83.6% spectral efficiency. It was higher than the existing techniques. Ariffin et al. [37] demonstrated a modeling approach based on frequently modulated continuous radar waves for detecting landslides in Malaysia. The authors designed a radar for detecting slow-moving landslide movements in this work. They successfully achieved 20 m/s speed radar performance to detect landslide occurrences. El Asri et al. [38] presented a method for modular system based U-Net using satellite images. In this presented method, they utilized CNN based deep learning model. Therefore, they obtained 70% accuracy. The presented framework's main flaw was the use of the data augmentation method, which will improve the result.

In summary, the authors in the related works used deep learning and U-net generation models for the classification of land scenes using satellite images. Few of the authors focused on the detection of multiple objects from satellite images. Remotely sensed images play a critical role in several applications such as environmental monitoring, disaster assistance, and geological surveys. The increasing need for satellite-derived imagery has led to a substantial flow of data being acquired on a daily basis. Consequently, the database has been expanded to include a much larger quantity of remote-sensing images. However, the task of accurately and efficiently acquiring and classifying images from an unstructured database is a significant challenge. Cloud cover and atmospheric conditions may conceal certain areas of the image. Therefore, getting clear and consistent data for classification might be challenging. Landscape features are complex and there is a chance of spatial heterogeneity within a single image. Therefore, correct classification is another challenge in landscape classification using satellite images.

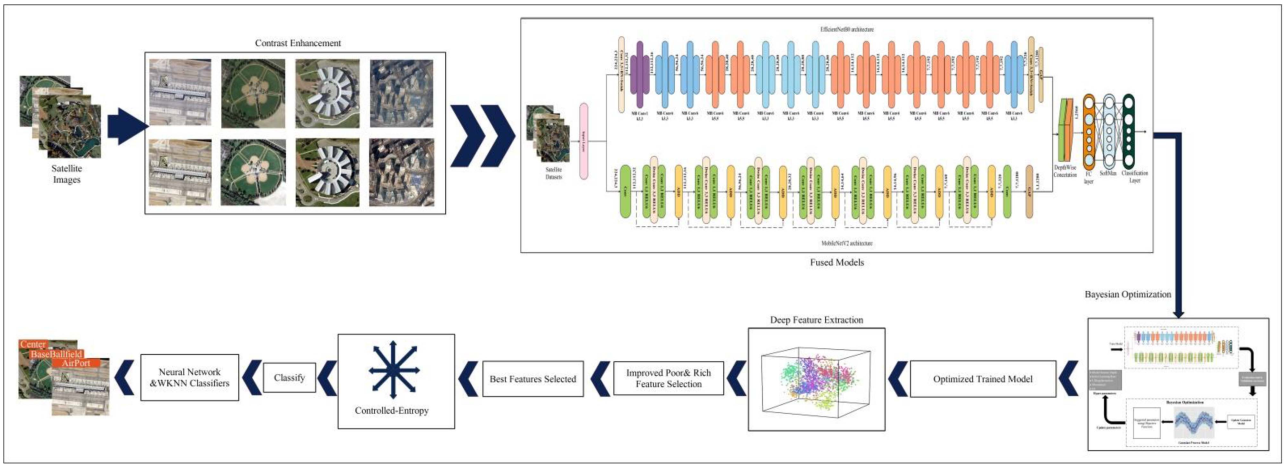


Fig. 1. Proposed methodology for the classification of land scene using satellite images.

In this work, we designed a deep learning-based internally fused models approach for classifying land scenes using satellite images.

The major contributions of the proposed framework are as follows.

1) Substitution-based approach is employed for the contrast enhancement of the satellite images.
2) Proposed a novel fused model technique based on Efficientb0 and MobileNetV2 architecture. The proposed model's hyperparameters were optimized using Bayesian optimization (BO) and trained using deep transfer learning.
3) Proposed an improved poor and rich controlled entropy optimization for best feature selection and conducted t-test analysis to measure the significance of different classifiers.

The rest of this article is organized as follows. The methodology section describes the dataset and normalization techniques, the proposed fused architecture, and the improved controlled poor and rich optimization (PRO) approach for best feature selection. The findings are explained under Section III, while Section IV presents the proposed method's conclusion.

## II. Proposed Work

This section explains the proposed landscape classification framework using a novel fused model. The proposed model was trained using BO and employed improved poor and rich controlled-entropy optimization for best feature selection, as shown in Fig. 1. This figure illustrates that the publically available datasets of satellite images were used for classification of landscape classification. In the initial step, contrast enhancement is performed by using a substitution-based approach. Following that, the two pretrained models named EfficientB0 and MobileNetv2 are internally fused for training proposes. In addition, the proposed model is fine-tuned by using deep transfer learning. Then, BO was utilized to select the optimized hyperparameters for the proposed model. The features were extracted from the trained model using newly added depth-wise activation. Furthermore, improved poor and rich controlled-entropy optimization was employed to select the best features. The optimized features are fed to neural network classifiers for the final classification. In the last phase, t-test analysis is conducted for statistical comparison of the performance of neural network classifiers.

### A. Dataset and Contrast Enhancement

In this article, we used three publically available land-use datasets for the experimental process. The selected datasets are aerial image dataset (AID) (https://captain-whu.github.io/BED4RS/), UC-Merced land use (https://captain-whu.github.io/BED4RS/), and WHU-RS19 (http://weegee.vision.ucmerced.edu/datasets/landuse.html). AID dataset is one of the largest aerial scenes datasets containing 30 aerial scene classes: forest, airport, farmland, bridge, beach, mountain, river, church, desert, dense residential, baseball field, industrial area, playground, pond, park, meadow, and to name a few. The total number of samples in this dataset is 10 000. Each aerial image has a predetermined resolution of 600 × 600 pixels to offer as much information as possible about a location. The UC-Merced land-use dataset consists of 21 land-use classes, each with 100 samples. The size of each image is 256 × 256 and manually acquired from the USGS National Map Urban Area Imagery collection for urban sites around the United States. In the WHU-RS19 dataset, 19 classes exist airport, beach, bridge, commercial area, desert, farmland, football field, forest, industrial area, meadow, mountain, park, parking lot, pond, port, railway station, residential area, river, and viaduct. This dataset's images have dimensions of 600 × 600 pixels and nearly 50 images per class. Fig. 2 presents a few images of each class of this dataset.

The images of these datasets were in low contrast and dark. These problems may lead us to misclassification. Therefore, we created a substitution-based approach for contrast enhancement by utilizing different filters. First, an adjusted filter with stretch limits is employed, and the resultant images are substituted in a sharpened filter. By sharpen filter, the intensity values of

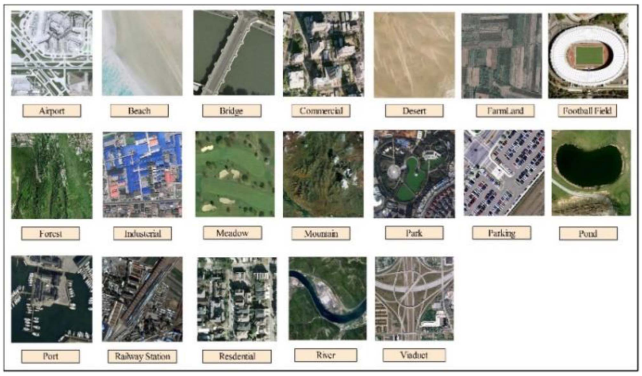


Fig. 2. Classes in the WHU-RS19 dataset.

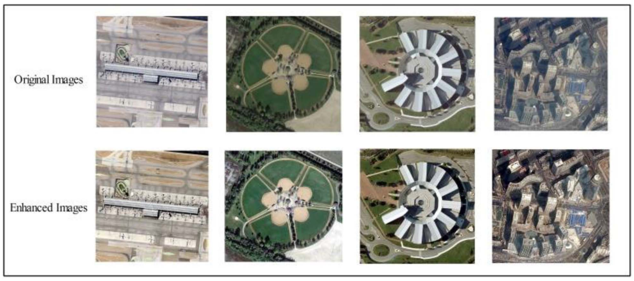


Fig. 3. Some samples of contrast enhanced of satellite datasets.

the images at the edges where different colors converge are heightened. Mathematical formula is defined as follows.

Consider that the satellite database has $k$ images $S \in \mathbb{R}^k$, where each image is represented by $f^k(v_0, h_0)$ and$(v_0, h_0) \in \mathbb{R}$. Assume that $S_L$ and $S_U$ are the specified lower and upper-restrictions on the image's intensity values before being normalized and $E_L$ and $E_H$ are the current lowest and maximum pixel values. Each pixel is measured by using the following equation:

$$g^k_{adj}\ (v_0, h_0) = (P - E_L)\left(\frac{S_U - S_L}{E_H - E_L}\right) + S_L \quad (1)$$

where $g_{adj}(v_0, h_0)$ is the resultant image, this image is further substituted in sharper filter using un-sharp masking approach. This filter is utilized to upgrade the polarity along the edges, sharpen using un-sharp mask is mathematically represented as

$$S^k\,(v_0, h_0) = g^k_{adj}\,(v_0, h_0) - \psi^k_{\text{smooth}}\,(v_0, h_0) \quad (2)$$

$$S^k_{\text{sharp}}\,(v_0, h_0) = \beta^k\,(v_0, h_0) + \alpha \times\ S^k\,(v_0, h_0) \quad (3)$$

where $\alpha$ is the scaling coefficient that determines the degree of sharpness and $\psi^k_{\text{smooth}}(v_0, h_0)$ is the smoothed variant of $\beta^k(v_0, h_0)$. $S^k_{\text{sharp}}(v_0, h_0)$ is a sharpen using the un-sharp mask filtered image. Therefore, the resultant image is mathematically defines as

$$\text{I}_{\text{out}}\,(v_0, h_0) = f\,(v_0, h_0) +\ g^k_{\text{adj}}\,(v_0, h_0) +\ S^k_{\text{sharp}}\,(v_0, h_0) \quad (4)$$

where $\text{I}_{\text{out}}(v_0, h_0)$ denotes the final contrast-enhanced image, which is presented in Fig. 3.

## B. Proposed Fused CNN Model

DCNN architectures EfficientNet-B0 and MobileNet V2 are both utilized for image classification tasks. EfficientNet-B0 is a version of the EfficientNet architecture, which was introduced by Google in 2019 [39]. EfficientNet-B0 is a variant of the EfficientNet architecture, which is known for its efficient use of computation and network capacity. EfficientNet-B0 is the smallest and most efficient version of the EfficientNet framework, differing from its larger predecessors by requiring less computing capacity and having fewer parameters. To obtain the highest accuracy in image classification tasks, the architecture of the network is based on a compound scaling technique that effectively increases the network's dimensions (depth, breadth, and resolution) [40]. Moreover, MobileNetV2 is an architecture for a network of CNN that was designed specifically to meet the needs of mobile and embedded devices [41]. In MobileNetV2, the expressive potential of the model is increased by integrating inverted residuals into its conceptual framework. Due to the design's primary focus on memory and computational efficiency, it is optimally adapted for deployment on devices with limited computing capacity, such as smartphones and tablets [42]. EfficientNetB0 and MobileNetV2 have gained recognition for their significant computational efficiency and reduced model size, while maintaining a satisfactory level of performance. In this research, the selection of both models was based on their ability to achieve an appropriate balance between computational efficiency and accuracy. The Efficient-b0 and mobileNetv2 architectures are fused into a single network to leverage both models' strengths. Efficient-b0 is utilized as a backbone network and mobileNet-v2 is added as a light -weight feature extractor. This process increased the accuracy and reduced the computation and memory usage. The fused model accepts input images up to 224 × 224 × 3 pixels in size. Fully connected, SoftMax and classification layers were removed from the Efficient-b0 and MobileNetv2 in order to add a new depth-wise layer to combine the features of global average pooling layers of both models. Following that, new fully connected layer, new softmax, and classification layers are added. The new FC layer is modified according to selected datasets. We trained the fused model by utilizing the BO in order to achieve the optimized hyperparameters. The brief explanation of BO is provided in Section III-D. After training, deep features are extracted from the depth-wise concatenation activation. The dimensions of extracted features are $N \times 2560$. The MobileNetV2 has 3.5 M parameters and EfficienetNetB0 has 3.5 M parameters. After depth-wise fusion process, resultant architecture has 6.3 M parameters instead of 8.8 M parameters. The fusion process of Efficient-b0 and MobileNetv2 architectures is presented in Fig. 4.

## C. Bayesian Optimization

Hyperparameters tuning is a crucial step in the training of DCNNs to achieve optimal performance. However, the search space of hyperparameters is often large, complex, and the evaluation of different hyperparameter configurations can be computationally expensive. Traditional methods, such as grid search and random search, are not well suited for this task due

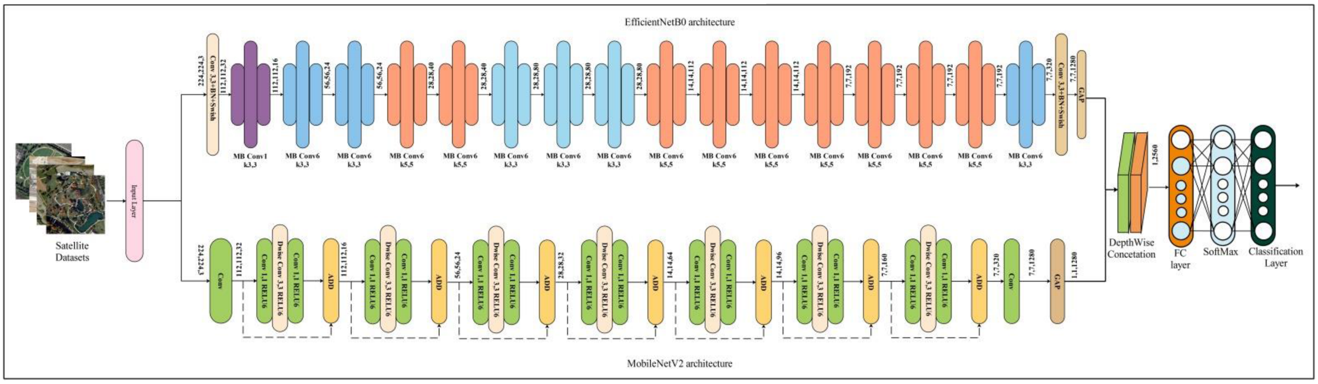


Fig. 4. Proposed fused model using depth concatenation of the classification using satellite images.

to their inefficiency and lack of ability to handle constraints and noise. BO is a powerful technique that can be used to solve this problem. It models the unknown performance of the DCNN as a function of the hyperparameters with a surrogate model, typically a Gaussian process (GP). The GP is used to model the distribution of the performance over the hyperparameters space, and the optimization algorithm is based on this distribution [43]. In each iteration, the BO algorithm chooses the next set of hyperparameters to evaluate based on the current state of the GP and an acquisition function that balances exploration and exploitation [44], [45].

GP is a type of stochastic process where the distribution of any subset of its random variables is multivariate Gaussian. This process operates under the assumption that inputs that are similar will produce similar outputs, and therefore, it uses a statistical model to represent the function. Similar to a Gaussian distribution, which is distributed by its mean and covariance, GP is defined by its mean function $\mu : d \rightarrow \mathbb{R}$ and covariance function $cov : d \times d' \rightarrow \mathbb{R}$. Which is mathematically formulated as

$$f\left(d\right) \sim GP\left(\mu\left(d\right), cov\left(d, d'\right)\right). \tag{5}$$

The function $f(d)$ for any given $d$ is instead of being a scalar, the new distribution represents $f(d)$. For simplicity, the mean function of the GP can be assumed as $\mu\ (d) = 0$. For covariance function cov, the exponential function is selected which is mathematically defined as

$$\text{cov}\ \left(d_i, d_j\right) = \exp\left(\frac{1}{2}||d_i - d_j||^2\right) \tag{6}$$

where $d_i$ and $d_j$ denote the $i$th, $j$th samples, respectively. The closer $d_i$ and $d_j$ are to each other, the more likely the value of some parameter will approach $1$. Conversely, as the separation between $d_i$ and $d_j$ increases, the value of the parameter tends to approach $0$. This relationship highlights the correlation and mutual influence between the samples, which intensifies, as the samples are closer together, and weakens as they move further apart.

The procedure for ascertaining the posterior distribution of $f(d)$ is as follows.

Initially, sample $s$ observations as training set $T_{1:s} = \{d_n, f_n\}_{n=1}^{s}$, $f_n = f(d_n)$. Assume that the values $f$ are derived according to multivariate normal distribution $f \sim N(0, \boldsymbol{\tau})$, where

$$\boldsymbol{\tau} = \begin{bmatrix} \text{cov}\left(d_1, d_1\right) & \text{cov}\left(d_1, d_2\right) & \cdots\cdots & \text{cov}\left(d_1, d_s\right) \\ \text{cov}\left(d_2, d_1\right) & \text{cov}\left(d_2, d_2\right) & \cdots\cdots & \text{cov}\left(d_2, d_s\right) \\ \vdots & \vdots & & \vdots \\ \text{cov}\left(d_s, d_1\right) & \text{cov}\left(d_s, d_2\right) & \cdots\cdots & \text{cov}\left(d_s, d_s\right) \end{bmatrix}. \tag{7}$$

Every value of vector $\boldsymbol{\tau}$ is determined by using (6). The degree of approximation between the two samples is calculated by function $f$ and without taking the noise effect the diagonal values of cov $(d_i, d_i) = 1$.

Based on function $f$, calculate the function value of new sample point $d_{s+1}$ using $f_{s+1} = f(x_{s+1})$. Based on the GP assumption, it can be stated that the combination of the function values of $f_{1:s}$ in the training set and the value of $f_{s+1}$ follows a normal distribution with $s + 1$ dimensions described as

$$\begin{bmatrix} f_{1:s} \\ f_{s+1} \end{bmatrix} \sim N\left(0, \begin{bmatrix} \boldsymbol{\tau} & \text{cov} \\ \text{cov}^T & \text{cov}\left(d_{s+1}, d_{s+1}\right) \end{bmatrix}\right) \tag{8}$$

where

$f_{1:s} = [f_1, f_2, f_3 \ldots f_s]^T$, $\text{cov} = [\text{cov}(d_{t+1}, d_1)\text{cov}(d_{t+1}, d_2) \ldots \text{cov}(d_{s+1}, d_s)]$ In addition, $f_{s+1}$ adheres to a normal distribution with a single dimension, meaning that according to the characteristics of a joint Gaussian distribution

$$\mu_{s+1}\left(d_{t+1}\right) = \text{cov}^T\ \tau^{\text{inv}}\ f_{1:s} \tag{9a}$$

$$\sigma_{s+1}^2\left(d_{t+1}\right) = -\ \text{cov}^T\ \tau^{-1}\text{cov}^T + \text{cov}\left(d_{s+1}, d_{s+1}\right). \tag{9b}$$

Once the posterior distribution of the objective function is established, BO employs an acquisition function ($\varphi$) to find the maximum of the function $f$. Typically, a high value of the acquisition function is assumed to correspond to a high value of the objective function $f$. As a result, maximizing the acquisition function is considered the same as maximizing the function $f$. Hence, the objective function is defined as

$$d^+ = \underset{d \in A}{\text{argmax}}\ \varphi\left(d|D\right). \tag{10}$$

The employed acquisition function is expected improvement (EI). The EI method calculates the expected level of improvement that can be achieved while investigating the area around the current suitable value. The current ideal value could be a local optimum, and the algorithm will need to look for the best value in other regions of the domain if the actual improvement of the function value is less than the predicted value after the process has run. The difference between the function value at the sample point and the present optimal value is used to compute improvement (I). The improvement is regarded as 0 if the function value at the sample point is less than the existing optimum value

$$I\left(d\right)=\max\left\{0,f_{s+1}\left(d\right)-f\left(d\right)\right\}. \tag{11}$$

In accordance with the EI optimization strategy, the objective is to maximize EI with respect to the current optimum value (f)

$$arg\max\ E\ \left[I\left(d\right)\right]=arg\max\ E(\max\left\{0,f_{s+1}\left(d\right)-f\left(d^{+}\right)\right\} \tag{12}$$

When $f_{s+1}(d)-f(d^{+})\geq 0$, the distribution $f_{s+1}(d)$ of follows a normal distribution with the mean $\mu(d)$, and the standard deviation, $\sigma^2(d)$. Consequently, the distribution of the random variable $I$ is also a normal distribution, with the mean $\mu(d)-f(d^{+})$ and standard deviation both being equal to $\sigma^2(d)$. The probability density function of $I$ is

$$f\left(I\right)=\frac{1}{\sqrt{2\pi\sigma\left(d\right)}}\ \exp\left(-\frac{\left(\mu\left(d\right)-f\left(d^{+}\right)-I\right)^{2}}{2\sigma^{2}\left(d\right)}\right),\ I\geq 0. \tag{13}$$

The function EI is used to compute the expected value of the degree of improvement that can be derived by analyzing the neighborhood surrounding the current optimal value. If the increase in function value during algorithm execution is less than the expected value, then the current optimal value point may represent a local optimal solution. In such situations, the algorithm will continue to seek for the optimal value point in other domain locations. The definition of EI is as follows:

$$\begin{aligned}E\left(I\right)\int_{\infty}^{\infty}If\left(I\right)\ dI&=\int_{I=0}^{I=\infty}I\frac{1}{\sqrt{2\pi\sigma\left(d\right)}}\\&\quad\exp\left(-\frac{\left(\mu\left(d\right)-f\left(d^{+}\right)-I\right)^{2}}{2\sigma^{2}\left(d\right)}\right)dI\\&=\sigma\left(d\right)\left[\omega\emptyset\left(\omega\right)+\emptyset\left(\omega\right)\right]\end{aligned} \tag{14}$$

where

$$\omega=\frac{\mu\left(d\right)-f\left(d^{+}\right)}{\sigma\left(d\right)}. \tag{15}$$

The expectation of improvement (I) is represented by (14), which is the definition of the EI function. In the final step, the employed stopping condition of BO has two factors; MaxTime is the first in which the BO optimization procedure will have a time limit of 50 400 s, which is equivalent to 14 h. The optimization process will end upon reaching the allotted time limit, regardless of whether it has converged to the optimal solution or not and the second stopping condition was when it completed its initial 30 function evaluations. In this research, we employed the BO to fine-tune the proposed model hyperparameters. The considered hyperparameters for tuning are listed in Table I.

TABLE I
SELECTED HYPERPARAMETERS AND ITS RANGES FOR OPTIMIZATION USING BAYESIAN OPTIMIZATION

| HyperParameters | Ranges |
|---|---|
| Learning Rate | [0.001 ,1] |
| Section Depth | [1, 3] |
| Momentum | [0.5, 0.95] |
| L2Regularization | $[1e^{-7}, 1e^{-2}]$ |
| Dropout | [0.0, 0.5] |
| Activation type | RELU, Leaky RELU |

## D. Improved Feature Selection Technique

Feature selection in deep learning is a challenging task that requires specific techniques to handle the high dimensionality and nonlinearity of deep neural networks. The use of regularization, auto-encoder-based methods, and metaheuristic optimization methods are effective strategies that can improve the accuracy and effectiveness of deep learning algorithms. Selecting the most suitable feature selection approach is crucial in order to get optimum results, since it should be based on the distinctive attributes of the given situation. In feature selection, the aim is to identify a subset of relevant features from a large number of input features that can contribute to the prediction accuracy of the model [46]. The achieved feature vector from the proposed model was high in dimension, which can lead to increased computational cost and longer training time. Therefore, we employed PRO controlled entropy for feature selection. The proposed technique can reduce the computational cost and improve the training efficiency, while still maintaining high accuracy. The original PRO algorithm is based on a real-world social phenomenon that can provide a viable solution for complex optimization problems [47]. Wealth is a widely used concept in various fields, particularly in economics. Its definition varies based on the attitude and implementation of the context. It is a measure of the economic status of individuals, and its quality and quantity are defined within the economic categories. The aspiration to become wealthy is a universal human desire, and people are naturally driven by financial pursuits to satisfy their needs and desires. Although there are numerous ways to acquire wealth, seeking insights from the experience and knowledge of the wealthiest individuals globally seems to be the most effective approach. Sociologically, people in a society are classified into two financial classes: the rich, whose wealth level exceeds the average, and the poor, whose wealth level is below the average. Members of both classes strive to improve their economic conditions through diverse means. However, they share a common tendency to observe each other's behavior and attempt to enhance their position by emulating or influencing the other. Therefore, the PRO algorithm's fundamental concept is to apply two strategies.

1) The poor population endeavors to improve their status and reduce the class gap by learning from the rich.
2) The rich population aims to widen the class gap by observing and gaining wealth from the poor.

The PRO algorithm involves the generation of an initial population by a random process using a uniform distribution technique. This technique selects values within specified upper and lower boundaries for each parameter. The original population is thereafter assessed according to the objective function and afterward arranged in ascending order depending on the outcomes. The PRO algorithm primarily consists of two distinct subpopulations, that represent the rich and the poor, respectively. The main population is mathematically defined as

$$P_{\text{main}}^{f} = P_{\text{rich}}^{f} \; + P_{\text{poor}}^{f} \tag{16}$$

where $P_{\text{main}}^{f}$ , $P_{\text{rich}}^{f}$, and $P_{\text{poor}}^{f}$ denoted the main population, rich population, and poor population size of features $f$, respectively. Following that, the main population is sorted in ascending order. The better-position population is considered as rich population and the remaining are considered as poor population of feature. The equation is defined as

$$f_1 < f_2 < f_3 < f_4 \ldots f_r < f_{r+1} < f_{r+2} \ldots f_N \tag{17}$$

where $f_1, f_2, f_3, f_4,$ and $f_r$ represented the rich population and $f_{r+1}, f_{r+2},$ and $f_N$ denoted the poor population. The primary population comprises two subpopulations: the poor and the rich. At each iteration of the algorithm, a defined mechanism must be employed to alter the position of every member of both subpopulations

The change in position of each feature of rich population by using the following equation:

$$\overrightarrow{V_{r,k}^{\text{new}}} = \overrightarrow{V_{r,k}^{\text{old}}} + \propto \left[\overrightarrow{V_{r,k}^{\text{old}}} - \overrightarrow{V_{p,\text{best}}^{\text{old}}}\right] \tag{18}$$

where $\overrightarrow{V_{r,k}^{\text{new}}}$ denotes new *k*th position value of rich population, $\overrightarrow{V_{r,k}^{\text{old}}}$ represents the present *k*th position value of rich population, $\propto$ is the parameter that represents the class gap, and $\overrightarrow{V_{p,\text{best}}^{\text{old}}}$ denotes the present position of best member of the poor population. The value of $V$ considered as a vector of all variables. Actually, each member of rich population widens the gap with every member of the poor population. Therefore, $\overrightarrow{V_{p,\text{best}}^{\text{old}}}$ is the best member of poor population. When the distance of rich population member increases from the $\overrightarrow{V_{p,\text{best}}^{\text{old}}}$, its distance increases from all the members of poor population. Actually, the poor population gets poorer when the distance between poor and rich gets higher. The distance that each member of the rich population should maintain from the poor population is determined by a random value, $\propto$ which falls between 0 and 1. The arbitrary nature of $\propto$ creates an internal competition within the rich population.

In every alteration of PRO, change in position of each feature of poor population by using the following equation:

$$\overrightarrow{V_{p,k}^{\text{new}}} = \overrightarrow{V_{p,k}^{\text{old}}} \; + \left[\propto (\text{pattern}) - \overrightarrow{V_{p,k}^{\text{old}}}\right] \tag{19}$$

where $\overrightarrow{V_{p,k}^{\text{new}}}$ represents the new $k\text{th}$ position of poor population, $\overrightarrow{V_{p,k}^{\text{old}}}$ denotes the current value of $k\text{th}$ position of poor population, $\propto$ is a random parameter, which presents the pattern improvement and pattern of getting rich. The pattern value mathematically formulated as

$$\text{Pattern} = \frac{\overrightarrow{V_{r,\text{best}}^{\text{old}}} + \overrightarrow{V_{r,\text{avg}}^{\text{old}}} + +\overrightarrow{V_{r,\text{worst}}^{\text{old}}}}{3} \tag{20}$$

where $\overrightarrow{V_{r,\text{best}}^{\text{old}}}$ represents the best member positions of the rich population, $\overrightarrow{V_{r,\text{avg}}^{\text{old}}}$ represents the average position member of the rich population while $\overrightarrow{V_{r,\text{worst}}^{\text{old}}}$ denotes the worst position member of the rich population.

In the realm of economics, certain factors have the potential to positively or negatively affect the overall economic climate. Examples of these factors include sudden fluctuations in the price of gold, oil, or petrochemicals, as well as significant changes in exchange rates, stock interest rates, or banking interests. Such factors can lead to abrupt alterations in the situation of certain individuals within a given society. Due to the inherent difficulty and sometimes impossibility of predicting these factors, they are utilized as a form of mutation in the algorithm. In this algorithm, we employed Gaussian mutation process. In Gaussian mutation, a small random value is added to each variable in an individual's solution vector, drawn from a Gaussian (normal) distribution with mean zero. The Gaussian distribution is a probability distribution that is symmetric around the mean, with most values close to the mean and progressively fewer values further away from the mean and the scale and shrink parameters determine the standard deviation of the distribution. At the first generation, the standard deviation is determined by the scale parameter. The initial population range is defined as a vector $V$ with rows and columns, the standard deviation for each coordinate $i$ of the parent vector is determined by scale × $(V(i,2) - V(i,1))$, and the reduction in standard deviation as generation's progress is determined by the shrink parameter. The standard deviation for coordinate $i$ of the parent vector at the $k\text{th}$ generation, represented as $\sigma_{i,k}$, is determined by utilizing a recursive formula

$$\sigma_{i,k} = \sigma_{i,k-1} \; \left(1 - \text{Sh}\frac{k}{\text{generations}}\right) \tag{21}$$

where $Sh$ denotes the shrink, the default value of shrink and scale is set. For generating new population after every iteration. After each iteration of the PRO algorithm, fitness is calculated by employing KNN. This function returns the cost value and cost is measured by using the following equation:

$$\text{Error} = 1 - \text{Accuracy}. \tag{22}$$

The cost function of KNN is mathematically formulated in the following equation:

$$\text{cost} = \alpha \times \text{Error} + \beta \times \left(\frac{\text{No of selected features}}{\text{Max of features}}\right) \tag{23}$$

where the default values of $\alpha$ 0.99 and $\beta$ are 0.01. There exist four distinct populations. These include the original populations of both the rich and the poor, as well as the updated populations of poor and rich. An objective function is used to assess each of these four populations, which are then merged into a composite population based on their ascending order of values. Prior to the creation of this composite population, the poor and rich subpopulations are separated by a predefined number. The purpose of merging the poor and rich populations at the end of each iteration is to account for the possibility that a member of the poor population may have gained enough wealth to replace a member of the rich population, and vice versa. It is worth noting that the top-performing member is always the first one in the rich population. Based on this, the original PRO selected features of dimension $\times \widehat{S}_i$ where $i \in \{N \times 1267,\ N \times 773,\ N \times 1220\}$. These feature vectors are obtained for three selected satellite datasets. After that, an improved version is designed based on the entropy calculation after each iteration.

*Entropy-controlled Selection:* Consider (21); the Entropy is computed after each iteration, removing the uncertainty among them. Entropy is computed as follows:

$$Entr\ (k) = -\sum_i h_i \log_2 h_i. \tag{24}$$

Based on the entropy value, the (21) is updated as follows:

$$\sigma_{i,k} = \sigma_{i,k-1}\left(1 - Sh\frac{\text{Entr}}{\text{generations}}\right). \tag{25}$$

This equation's values (features) are returned and passed to the fitness function that checks the fitness after each iteration. In addition, the cost of each iteration is computed after each iteration. In the end, the final feature vectors are obtained of dimensions $N \times 1060,\ N \times 642,\ N \times 1004$, respectively, for all three datasets. The selected features are fed into neural network classifiers for the final classification.

## III. Results and Analysis

In this section, detailed experimental results of the proposed framework are described. The experiments are conducted on three datasets, and a complete description of each dataset is given under the Dataset and Contrast Enhancement section. Each dataset is divided into a 50:50 ratio. This indicates that 50% of images are utilized to train the proposed model and other 50% images are opted for testing. All the experiments were conducted using the 10-fold cross-validation because 10-fold cross-validation is widely favored due to its ability to achieve an appropriate balance between variance, which pertains to the generalization of the performance estimate, and computational cost. In our case, we had N×2560 features the smaller value of *k* was not performed well and after 10 values of *k*, the performance of models was consistent. The utilized static hyperparameters during training of the proposed model are epochs, minibatch size and optimizer having values are 300, 18, and stochastic gradient decent with momentum, respectively. Furthermore, the initial learning rate, section depth, momentum, L2Regularization, dropout, and activation type are defined with their ranges and optimized by using BO. Multiple neural network classifiers and KNN are employed for the classification task, including narrow neural network, medium neural network, wide neural network, bi-layered neural network, and weighted KNN. The performance evaluation parameters are precision, recall, accuracy, error, false negative rate, f1-score, and time. All the experiments were conducted on MATLAB R2023a executing on MSI's leopard series with Intel core i7 processor, 16 GB RAM, 512 SSD with 1TB HDD integrated disk, and 4 GB NVIDIA RTX graphics card.

### A. AID Dataset Results

In this section, the AID dataset's results are provided. The deep features of the proposed fused architecture model are extracted in the first step. The enhanced data set was used to train this model using BO and deep transfer learning. Table II shows the classification accuracy of this model, which obtained a 95.7% score from the wide neural network classifier. The precision, recall, error, FNR, and F1-score are 95.58%, 95.53%, 4.3%, 4.47%, and 95.55%. The medium neural network has the shortest execution time of 32.21 (s) and the longest execution time of 106.23 (s) in this phase experiment, which records the classification computational time for each classifier. The best features were selected in the next phase utilizing PRO. According to Table III, selected features are passed to the classifiers. Wide neural network classifier achieved a maximum accuracy of 95.6%. The wide neural network recall rate is 95.37%, the accuracy rate is 95.45%, the error rate is 4.4%, the FNR is 4.63, and the F1-score is 95.54%. Each classifier's processing times are further recorded.

The results of the third step, which involves controlled Entropy are performed, are shown in Table IV. The wide neural network classifier has an accuracy of 96.3%, higher than the previous two steps (see Tables II and III). Furthermore, recall and precision has 96.13 and 96.0%, respectively. A confusion matrix is shown in Fig. 5 and may be used to verify the performance of a wide neural network. The controlled entropy approach significantly improves accuracy in comparison to the previous two experiments performed on this dataset. It is also noticed that time decreases after the entropy phase.

### B. UC-Merced Land-Use Results

In the initial phase, UC-Merced Land-use dataset results are described. Deep features are extracted from the proposed fused architecture model and trained using BO and deep transfer learning on enhanced datasets. Table V presents the classification results of the UC-Merced Land-use dataset. The wide neural network achieved a higher accuracy of 96.4% in this table. The precision, recall, error, FNR, and f1-score having values are 96.4%, 96.3%, 3.6%, 3.7%, and 96.3%, respectively. The wide neural network classifier has achieved higher accuracy than all the listed classifiers in Table V. Furthermore, the computation is also recorded for all the classifiers. The shortest execution time is 15.96 (s), and the longest execution time has been recorded for the bi-layered neural network classifier, which is 23.09 (s).

In the next phase, the best features are selected by opting for PRO. The optimized features are passed to a neural network classifier for classification. Table VI illustrates the improved PRO results on the UC-Merced Land-use dataset. The wide neural network achieved a higher accuracy of 96.5% from this experiment. Wide neural networks outperformed the rest of the classifiers. The precision rate is 96.5%, the recall rate is 96.4%, the error rate is 3.5%, FNR is 3.6%, and the f1-score is 96.4%.

TABLE II
PROPOSED FUSED ARCHITECTURE OF EFFICIENTB0 AND MOBILENETV2 MODEL FUSION RESULTS ON THE AID DATASET

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time(s) | F1-Score |
|---|---|---|---|---|---|---|---|
| NNN | 89.90 | 87.17 | 90.2 | 9.8 | 12.83 | 50.12 | 88.51 |
| MNN | 94.98 | 95.08 | 95.2 | 4.8 | 4.92 | 32.21 | 95.03 |
| **WNN** | **95.58** | **95.53** | **95.7** | **4.3** | **4.47** | **40.43** | **95.55** |
| BNN | 86.41 | 86.93 | 86.8 | 13.2 | 13.07 | 106.23 | 86.67 |
| WKNN | 93.63 | 91.98 | 92.2 | 7.8 | 8.14 | 92.90 | 92.72 |

Bold entities presents the highest values in the tables.

TABLE III
PROPOSED IMPROVED POOR AND RICH OPTIMIZATION RESULTS ON AID DATASET

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time(s) | F1-Score |
|---|---|---|---|---|---|---|---|
| NNN | 88.27 | 88.05 | 88.4 | 11.6 | 11.95 | 31.05 | 88.16 |
| MNN | 94.58 | 94.54 | 94.7 | 5.3 | 5.46 | 18.54 | 94.52 |
| **WNN** | **95.45** | **95.37** | **95.6** | **4.4** | **4.63** | **22.09** | **95.41** |
| BNN | 85.72 | 85.59 | 86.0 | 14.0 | 14.41 | 43.62 | 85.64 |
| WKNN | 93.07 | 91.46 | 91.7 | 8.3 | 8.64 | 51.64 | 92.23 |

Bold entities presents the highest values in the tables.

TABLE IV
PROPOSED CONTROLLED ENTROPY RESULTS ON AID DATASET

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time(s) | F1-Score |
|---|---|---|---|---|---|---|---|
| NNN | 90.4 | 90.5 | 90.8 | 9.2 | 9.5 | 17.13 | 90.42 |
| MNN | 94.8 | 94.7 | 95.1 | 4.9 | 5.2 | 10.43 | 94.86 |
| **WNN** | **96.1** | **96.0** | **96.3** | **3.7** | **4.0** | **13.15** | **96.06** |
| BNN | 87.6 | 87.6 | 88.0 | 12.0 | 12.4 | 29.01 | 87.64 |
| WKNN | 94.8 | 93.5 | 93.8 | 6.2 | 6.5 | 27.15 | 94.13 |

Bold entities presents the highest values in the tables.

TABLE V
PROPOSED FUSED ARCHITECTURE MODEL RESULTS ON UC-MERCED LANDUSE DATASET

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| NNN | 92.8 | 92.6 | 92.7 | 7.3 | 7.4 | 20.13 | 92.6 |
| MNN | 95.9 | 95.8 | 95.9 | 4.1 | 4.2 | 15.96 | 95.8 |
| **WNN** | **96.4** | **96.3** | **96.4** | **3.6** | **3.7** | **20.74** | **96.3** |
| BNN | 90.9 | 90.7 | 90.8 | 9.2 | 9.3 | 23.09 | 90.7 |
| WKNN | 88.7 | 80.9 | 81.0 | 19.0 | 19.1 | 23.05 | 84.6 |

Bold entities presents the highest values in the tables.

TABLE VI
PROPOSED IMPROVED POOR AND RICH OPTIMIZATION RESULTS ON UC-MERCED LAND-USE DATASET

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| NNN | 91.1 | 91.0 | 91.0 | 9 | 9 | 13.43 | 91.0 |
| MNN | 95.0 | 95.5 | 95.5 | 4.5 | 4.5 | 9.37 | 95.2 |
| **WNN** | **96.5** | **96.4** | **96.5** | **3.5** | **3.6** | **12.09** | **96.4** |
| BNN | 87.8 | 87.6 | 87.6 | 12.4 | 12.4 | 14.9 | 87.7 |
| WKNN | 89.3 | 86.7 | 86.8 | 13.2 | 13.3 | 17.35 | 88.0 |

Bold entities presents the highest values in the tables.

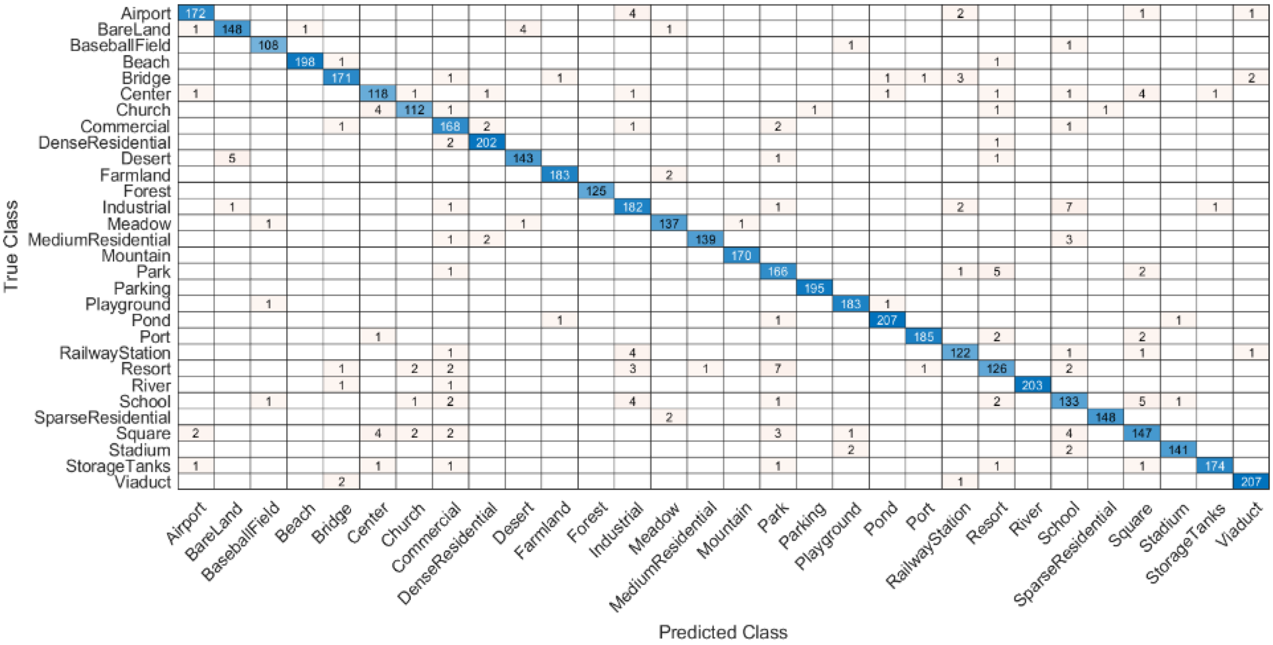


Fig. 5. Confusion matrix of a wide neural network of controlled entropy process on AID dataset.

TABLE VII
Proposed Controlled Entropy Based Results on UC-Merced Land-Use Dataset

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| **NNN** | 91.5 | 91.3 | 91.3 | 8.7 | 8.7 | 3.90 | 91.4 |
| **MNN** | **90.9** | **95.6** | **95.6** | **4.4** | **4.4** | **2.77** | **93.7** |
| **WNN** | 97.4 | 97.3 | 93.3 | 6.7 | 2.7 | 3.42 | 97.3 |
| **BNN** | 89.1 | 93.1 | 93.1 | 6.9 | 6.9 | 3.33 | 93.4 |
| **WKNN** | 93.7 | 93.1 | 93.1 | 6.9 | 6.9 | 3.31 | 93.4 |

Bold entities presents the highest values in the tables.

TABLE VIII
Proposed Fused Architecture Model Results on WHU-RS19 Dataset

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| **NNN** | 88.7 | 88.3 | 88.2 | 11.8 | 11.7 | 13.7 | 88.4 |
| **MNN** | 92.1 | 91.9 | 91.8 | 8.2 | 8.1 | 12.7 | 91.9 |
| **WNN** | **93.2** | **93.5** | **92.8** | **7.2** | **6.5** | **16.0** | **93.3** |
| **BNN** | 81.4 | 81.2 | 81.2 | 18.8 | 18.8 | 24.0 | 81.2 |
| **WKNN** | 84.6 | 72.1 | 72.4 | 27.6 | 27.9 | 17.5 | 77.8 |

Bold entities presents the highest values in the tables.

These values are also calculated from the other classifiers. The computation time is noted for all the classifiers, and it is observed that the medium neural network classifier required a lesser time of 9.37 (s).

In contrast, weighted KNN takes the longest time, which is 17.35 (s). The final step employs a controlled entropy approach on best features. Table VII shows the controlled entropy results on the UC-Merced Land-use dataset. In this table, wide neural network gained the highest accuracy of 95.6%. The precision, recall, error, FNR, and f1-score values are 90.9%, 95.6%, 4.4%, 4.4%, and 93.7%. A confusion matrix presented in Fig. 6, can be utilized to verify the performance of a wide neural network classifier. This experiment shows that the computation time is reduced from the 1 and 2, described in Tables V and VI.

### C. *WHU-RS19 Results*

In this experiment, the result of the WHU-RS19 has been presented. Deep features were extracted from the proposed fused architecture of the efficientnetb0 and mobilenetv2 model in the first step. The proposed model was trained through BO and deep transfer learning. Table VIII illustrates the classification results of this model. The wide neural network classifier gained the

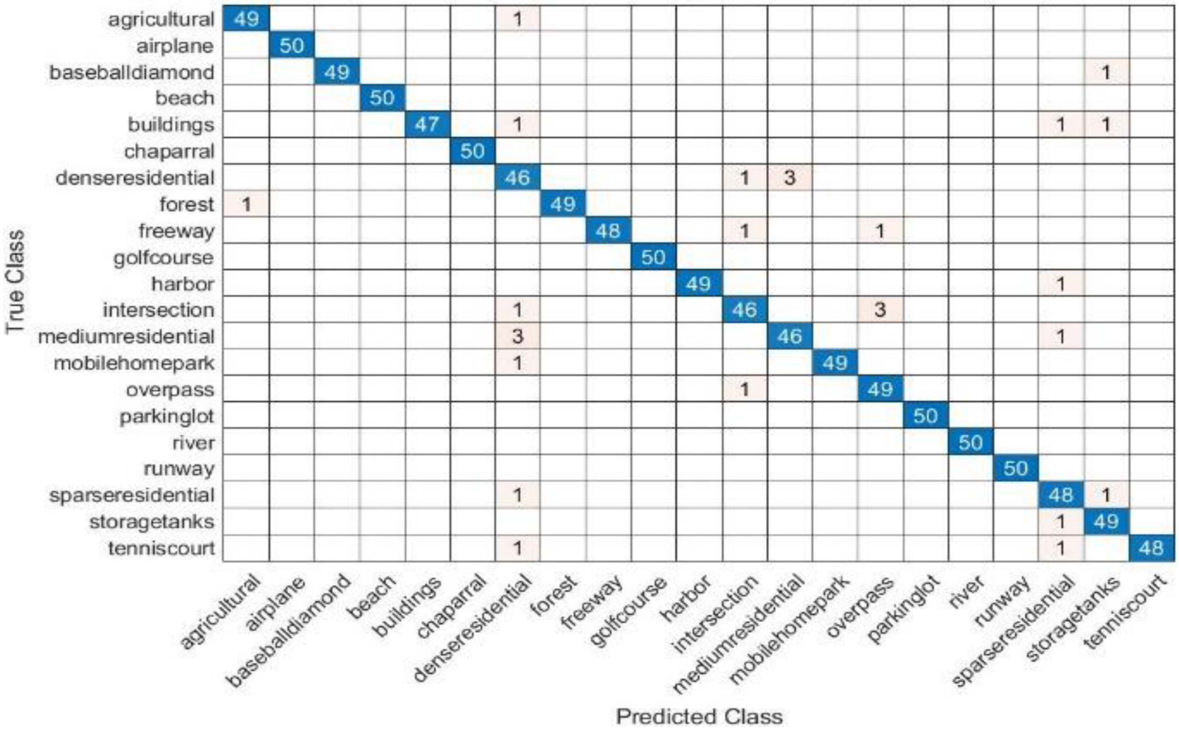


Fig. 6. Confusion, matrix of controlled entropy technique on medium neural network classifier, using UC-Merced land-use dataset.

TABLE IX
Proposed Improved Poor and Rich Optimization Results on WHU-RS19 Dataset

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| **NNN** | 85.6 | 85.5 | 85.6 | 14.4 | 14.5 | 14.98 | 85.5 |
| **MNN** | 92.6 | 92.1 | 92.2 | 7.8 | 7.9 | 11.57 | 92.4 |
| **WNN** | **93.2** | **92.9** | **93.0** | **7.00** | **7.1** | **13.01** | **93.0** |
| **BNN** | 78.5 | 77.6 | 77.8 | 22.2 | 22.4 | 27.42 | 78.0 |
| **WKNN** | 84.1 | 75.17 | 75.4 | 24.6 | 24.83 | 14.04 | 74.38 |

Bold entities presents the highest values in the tables.

highest accuracy from all the other classifiers in this table. The highest accuracy is 92.8%. The precision, recall, error, FNR, and f1-score values are 93.2%, 93.5%, 7.2, 6.5%, and 93.3%. These statistics are calculated for all the other classifiers. It is observed that the medium neural network classifier executes faster than the listed classifiers. The executing time of this classifier is 12.7 (s), although the longest execution time is 17.5 (s). The extracted features are optimized in the next step by employing improved PRO. Following that, the optimized features are passed to the neural network classifier. The results of improved poor and rich feature selection on selected features are presented in Table IX. This table shows that the wide neural network outperformed all the other neural network classifiers. It achieved an accuracy of 93.0%. The precision rate is 93.2%, the recall rate is 92.9%, the error rate is 7.0%, the FNR rate is 7.1%, and the f1-score is 93.0%. The computation time is recorded for all the listed classifiers; it is noted that the medium neural network takes less time, which is 11.57 (s).

Table X shows controlled entropy-based results on the WHU-RS19 dataset in the final step. In this table, the maximum 97.8% accuracy has been noted from the medium neural network classifier, and it takes 2.80 (s) for execution, which is the shortest time from all the listed classifier's computation time and maximum execution time of 127.7 (s) has been recorded from bi-layered neural network classifier. The precision, recall, error, FNR, and f1-score have 97.7%, 97.8%, 2.2%, 2.2%, and 97.7% values. This numerical Analysis is also conducted for all the other neural kernels. After applying Entropy, it was observed that the accuracy was significantly improved. Moreover, it was clearly observed that computation is reduced from the previous experiments, shown in Tables VIII and IX. Fig. 7 presents the confusion matrix of the medium neural network classifier, which further verifies computed values.

### *D. Discussion*

*1) T-Test-Based Analysis:* The *t*-test is a statistical test that helps determine whether the means of two groups or samples differ significantly. The performance of the two classifiers can be compared using a *t*-test analysis. In this work, we performed

TABLE X
Proposed Controlled Entropy Based Results on WHU-RS19 Dataset

| Classifier | Precision | Recall | Accuracy | Error | FNR | Time | F1-Score |
|---|---|---|---|---|---|---|---|
| **NNN** | 92.5 | 92.3 | 92.4 | 7.6 | 7.7 | 3.54 | 92.4 |
| **MNN** | **97.7** | **97.8** | **97.8** | **2.2** | **2.2** | **2.80** | **97.7** |
| **WNN** | 92.8 | 97.8 | 97.8 | 2.2 | 2.2 | 3.33 | 95.2 |
| **BNN** | 86.5 | 86.1 | 86.0 | 14 | 13.9 | 12.47 | 86.3 |
| **WKNN** | 93.3 | 91.4 | 91.6 | 8.4 | 8.6 | 3.99 | 92.3 |

Bold entities presents the highest values in the tables.

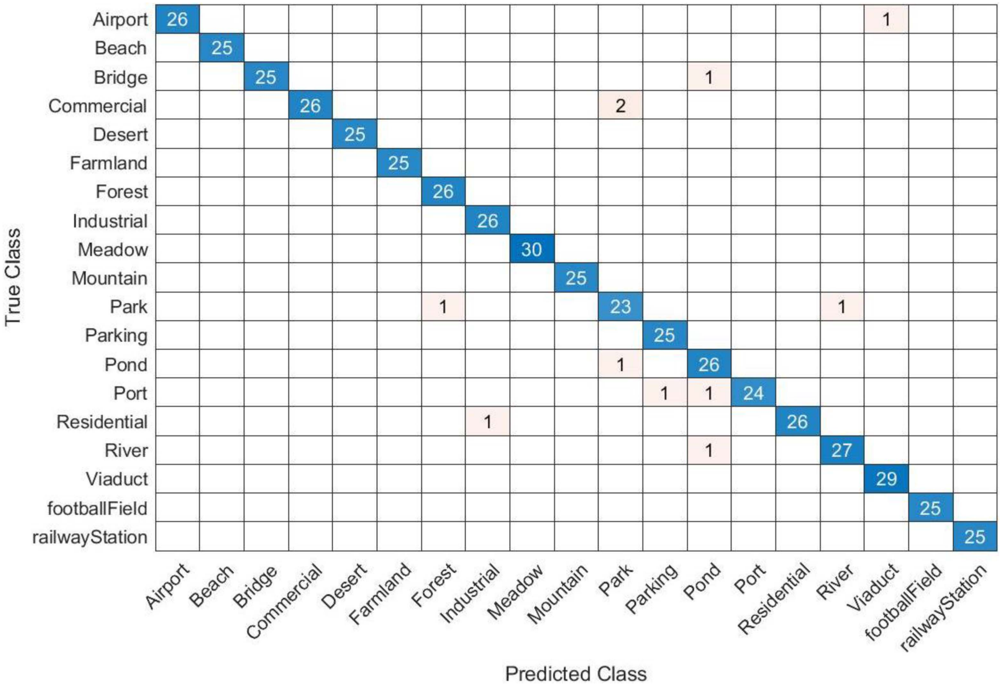


Fig. 7. Confusion matrix of controlled entropy technique on medium neural network classifier using WHU-RS19 dataset.

a $t$-test for the selected datasets. Two classifiers from all the selected datasets have been selected based on the highest and second-highest accuracies, as shown in Table XI. Initially, we selected two hypotheses named the null hypothesis $(H_0)$ and alternative hypothesis $(H_1)$, the $H_0$ supposed that there is no significant difference in the classifier's performance, whereas $H_1$ assumed that there is a significant difference. In the first step, we calculate the difference among the classifier accuracies for each process using (26). The value of $N = 3$, which denotes the process of the proposed framework. After that, we computed the mean $(\mu)$ value of differences $(\partial)$ for all the selected datasets by using (27)

$$\partial = (C_1 - C_2) \tag{26}$$

$$\text{mean}\,(\mu) = \frac{1}{N}\sum_{i=1}^{N}(\partial) \tag{27}$$

where $C_1, C_2$ presented the wide neural network classifier (highest value classifier) and medium neural network classifier (second highest classifier), respectively, $N$ denotes the total number of processes in the framework. The values of $\mu$ for selected datasets are $0.86,\ 1.26$, and $0.6$. In the next phase, we calculate the standard deviation using the following equation:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(\partial - \mu)^2}{N-1}}. \tag{28}$$

Standard deviation $\sigma$ having values are $0.351,\ 0.92$, and $0.52$, respectively; the $t$-test values are calculated by using the $t$-selection formula. The $t$-selection is mathematically formulated as

$$T_{sel} = \frac{\sqrt{N} \times \mu}{\sigma} \tag{29}$$

where values of $T_{sel}$ are $4.26,\ 2.36,$ and $1.96$. The degree of freedom is computed as $N - 1$. We set the significance level to 95% on 0.05. The $t$-distribution table range is $[-4.303, +4.303]$ based on significance level and degree of freedom. The values of $T_{sel}$ Lies between a critical range of $t$-distribution range. Hence,

TABLE XI
COMPREHENSIVE COMPARISON WITH EXISTING TECHNIQUES

| Ref | Year | Methodology | Dataset | Accuracy |
|---|---|---|---|---|
| **[48]** | 2023 | Hybrid deep learning networks and whale optimization for optimal guidance | AID | 89.58% |
| **[49]** | 2023 | Few-shot scene classification using metric learning and local descriptors | UC Merced, WHU-RS19 | 77.76%, 82.06% |
| **[50]** | 2023 | few-shot remote sensing classification using deep features | UC Merced, WHU-RS19, AID | 86.06%, 94.51%, 86.17% |
| **[51]** | 2023 | Dynamical scalable transformer methods for remote sensing image classification and segmentation | UC Merced, WHU-RS19 | 93.44%, 96.16% |
| **[52]** | 2022 | Graph based network for few-shot remote scene image classification | UC Merced, WHU-RS19 | 81.66%, 82.37% |
|  |  | Proposed Methodology | AID, UC Merced, WHU-RS19 | 96.3%, 95.6%, 97.8% |

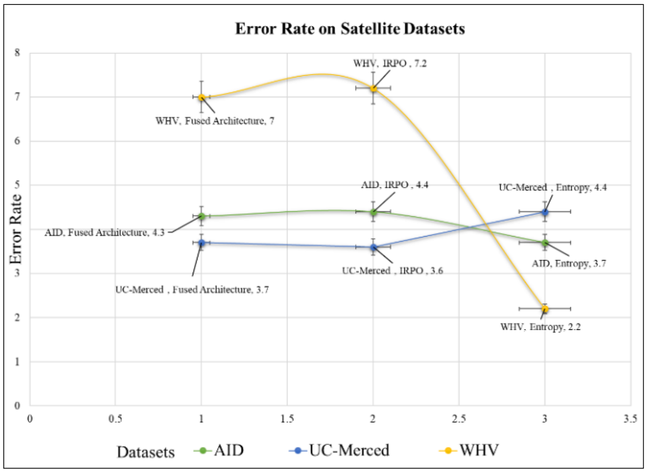


Fig. 8. Error rate graph measured on satellite datasets.

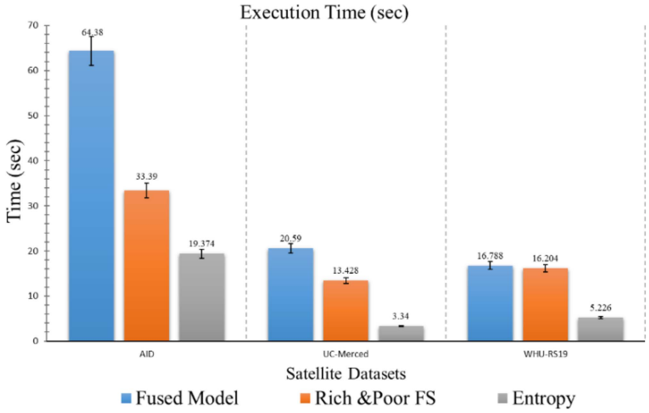


Fig. 9. Computational time-based graph measured on satellite datasets.

TABLE XII
SELECTED CLASSIFIERS FROM ALL THE DATASETS FOR T-TEST ANALYSIS

| AID Dataset | | | |
|---|---|---|---|
| Classifier | Proposed Fused Model | Feature Selection | Controlled Entropy |
| WNN | 95.7 | 95.6 | 96.3 |
| MNN | 95.2 | 94.7 | 95.1 |
| UC Merced Land use Dataset | | | |
| WNN | 96.4 | 96.5 | 93.3 |
| MNN | 95.9 | 95.5 | 95.6 |
| WHU-RS19 Dataset | | | |
| WNN | 92.8 | 93.0 | 97.8 |
| MNN | 91.8 | 92.2 | 97.8 |

$H_0$ is accepted, it indicates no significant difference between the classifiers performance.

*2) Graphical Results:* Fig. 8 illustrates the error rate of selected datasets corresponding to their methods. The graphs show that the WHV dataset has a lower % error rate of 2.2% when controlled Entropy is applied and 7.2% and 7.0% when proposed fused architecture and PRO is employed. The AID dataset shows that the smallest error rate is achieved by employing Entropy, which is 3.7%. In addition, the maximum error rate is noted when improved PRO (IRPO) is applied. In the UC-Merced dataset, a 3.6% error rate has been noted when improved PRO is utilized, which is the lowest error rate from the other methods. The entire graph shows that the error is gradually reduced when a controlled entropy process is employed, which is a strength of this experiment.

*3) Comparison With SOTA:* A Comprehensive comparison with existing techniques has been presented in Table XII. It can be observed that the proposed method outclasses the rest of the listed advanced methods. The highest accuracy achieved on the AID dataset is 89.58% by Vinaykumar et al. [48], whereas the proposed methodology achieved 96.3%. Similarly, on UC Merced and WHU-RS19 datasets, the achieved higher accuracies using the proposed methodology are 95.6% and 97.8%, compared to [49], [50], [51], and [52] by different methods.

## IV. CONCLUSION

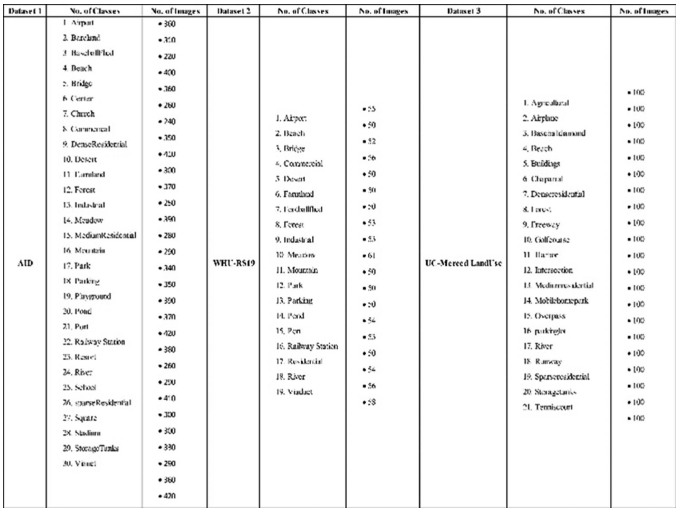
This article proposes new deep learning models, inner information fusion, and optimal feature selection-based architecture to classify land scene images. The proposed architecture includes contrast enhancement, model creation, hyperparameter optimizations, feature selection, and classification. Contrast enhancement is performed initially, and a deep learning model is designed. The purpose of enhancement is to increase the quality of low-contrast images and then better learning of a designed model. After that, the hyperparameters have been initialized based on BO instead of manual assignment. The manual assignment is inefficient, and sometimes, this process reduces the learning performance. After that, features are selected based on the poor-rich controlled entropy technique and classified using machine learning classifiers. Three publically available datasets have been employed for the experimental process and obtained the accuracy of 96.3%, 95.6%, and 97.8%, respectively. Comparison with the recent techniques shows an overall improvement in accuracy and less computational time. Overall, we conclude with the following points.

1) Fusion of inner layers based on deep learning models improved accuracy and lessened overall parameters.
2) Initialization of hyperparameters using BO improved the accuracy and learning performance.
3) Selection of best features using the poor-rich controlled entropy technique reduced the computational time and maintained the accuracy.

The limitation of this work was the training time increased after the internal fusion of EfficientNet B0 and MobileNetV2 architecture. Moreover, the designed architecture has a large amount of pooling activations due to the fusion of both models, which reduces the useful information from the data. These limitations will be considered as future work.

*Data Availability:* The selected datasets are AID (https://captain-whu.github.io/BED4RS/), UC-Merced land use (https://captain-whu.github.io/BED4RS/), and WHU-RS19 (http://weegee.vision.ucmerced.edu/datasets/landuse.html).

## Appendix



| Dataset 1 | No. of Classes | No. of Images | Dataset 2 | No. of Classes | No. of Images | Dataset 3 | No. of Classes | No. of Images |
|---|---|---|---|---|---|---|---|---|
| AID | 1. Airport<br>2. Bareland<br>3. BaseballField<br>4. Beach<br>5. Bridge<br>6. Center<br>7. Church<br>8. Commercial<br>9. DenseResidential<br>10. Desert<br>11. Farmland<br>12. Forest<br>13. Industrial<br>14. Meadow<br>15. MediumResidential<br>16. Mountain<br>17. Park<br>18. Parking<br>19. Playground<br>20. Pond<br>21. Port<br>22. Railway Station<br>23. Resort<br>24. River<br>25. School<br>26. sparseResidential<br>27. Square<br>28. Stadium<br>29. StorageTanks<br>30. Viaduct | • 360<br>• 310<br>• 220<br>• 400<br>• 360<br>• 260<br>• 240<br>• 350<br>• 410<br>• 300<br>• 370<br>• 250<br>• 390<br>• 280<br>• 290<br>• 340<br>• 350<br>• 390<br>• 370<br>• 420<br>• 380<br>• 260<br>• 290<br>• 410<br>• 300<br>• 300<br>• 330<br>• 290<br>• 360<br>• 420 | WHU-RS19 | 1. Airport<br>2. Beach<br>3. Bridge<br>4. Commercial<br>5. Desert<br>6. Farmland<br>7. footballfield<br>8. Forest<br>9. Industrial<br>10. Meadow<br>11. Mountain<br>12. Park<br>13. Parking<br>14. Pond<br>15. Port<br>16. Railway Station<br>17. Residential<br>18. River<br>19. Viaduct | • 55<br>• 50<br>• 52<br>• 56<br>• 50<br>• 50<br>• 50<br>• 53<br>• 53<br>• 61<br>• 50<br>• 50<br>• 50<br>• 54<br>• 53<br>• 50<br>• 54<br>• 56<br>• 58 | UC-Merced LandUse | 1. Agricultural<br>2. Airplane<br>3. Baseballdiamond<br>4. Beach<br>5. Buildings<br>6. Chaparral<br>7. Denseresidential<br>8. Forest<br>9. Freeway<br>10. Golfcourse<br>11. Harbor<br>12. Intersection<br>13. Mediumresidential<br>14. Mobilehomepark<br>15. Overpass<br>16. parkinglot<br>17. River<br>18. Runway<br>19. Sparseresidential<br>20. Storagetanks<br>21. Tenniscourt | • 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100<br>• 100 |

## References

[1] T. D. Acharya, I. T. Yang, and D. H. Lee, "Land cover classification using a KOMPSAT-3A multi-spectral satellite image," *Appl. Sci.*, vol. 6, no. 11, 2016, Art. no. 371.

[2] Q.-B. Zhou, Q.-Y. Yu, L. Jia, W.-B. Wu, and H.-J. Tang, "Perspective of Chinese GF-1 high-resolution satellite data in agricultural remote sensing monitoring," *J. Integrative Agriculture*, vol. 16, no. 2, pp. 242–251, 2017.

[3] N. Ibharim, M. Mustapha, T. Lihan, and A. Mazlan, "Mapping mangrove changes in the Matang Mangrove Forest using multi temporal satellite imageries," *Ocean Coastal Manage.*, vol. 114, pp. 64–76, 2015.

[4] C. Jacqueminet et al., "Land cover mapping using aerial and VHR satellite images for distributed hydrological modelling of periurban catchments: Application to the Yzeron catchment (Lyon, France)," *J. Hydrol.*, vol. 485, pp. 68–83, 2013.

[5] M. Halabisky, L. M. Moskal, A. Gillespie, and M. Hannam, "Reconstructing semi-arid wetland surface water dynamics through spectral mixture analysis of a time series of Landsat satellite images (1984–2011)," *Remote Sens. Environ.*, vol. 177, pp. 171–183, 2016.

[6] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2183–2195, Apr. 2018.

[7] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[8] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[9] X. Zheng, Y. Yuan, and X. Lu, "Dimensionality reduction by spatial–spectral preservation in selected bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5185–5197, Sep. 2017.

[10] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat: A learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2015, pp. 1–10.

[11] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[12] L. Wang, W. Song, and P. Liu, "Link the remote sensing Big Data to the image features via wavelet transformation," *Cluster Comput.*, vol. 19, no. 2, pp. 793–810, 2016.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[15] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[16] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[17] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE 9th Int. Conf. Comput. Vis.*, 2003, vol. 2, pp. 1470–1477.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[21] Y. Jia et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[22] H. Yazdi, S. Sad Berenji, F. Ludwig, and S. Moazen, "Deep learning in historical architecture remote sensing: Automated historical courtyard house recognition in Yazd, Iran," *Heritage*, vol. 5, no. 4, pp. 3066–3080, 2022.

[23] K. Patel, C. Bhatt, and P. L. Mazzeo, "Deep learning-based automatic detection of ships: An experimental study using satellite images," *J. Imag.*, vol. 8, no. 7, 2022, Art. no. 182.

[24] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Satellite image classification of building damages using airborne and satellite image samples in a deep learning approach," *ISPRS Ann. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 4, no. 2, pp. 89–96, 2018.

[25] M. Pritt and G. Chern, "Satellite image classification with deep learning," in *Proc. IEEE Appl. Imagery Pattern Recognit. Workshop*, 2017, pp. 1–7.

[26] K. Gao et al., "A region-based deep learning approach to instant segmentation of aerial orthoimagery for building rooftop detection," Geomatica, vol. 75, pp. 148–164.

[27] A. Rostami, R. Shah-Hosseini, S. Asgari, A. Zarei, M. Aghdami-Nia, and S. Homayouni, "Active fire detection from LandSat-8 imagery using deep multiple kernel learning," *Remote Sens.*, vol. 14, no. 4, 2022, Art. no. 992.

[28] S. Yosmaoglu and S. Nieland, "Road network generation with conditional generative adversarial networks from aerial images," *Riga, Latvia*, Dec. 2021.

[29] W. Lim, K. Choi, W. Cho, B. Chang, and D. W. Ko, "Efficient dead pine tree detecting method in the forest damaged by pine wood nematode (Bursaphelenchus xylophilus) through utilizing unmanned aerial vehicles and deep learning-based object detection techniques," *Forest Sci. Technol.*, vol. 18, no. 1, pp. 36–43, 2022.

[30] A. Ch, R. Ch, S. Gadamsetty, C. Iwendi, T. R. Gadekallu, and I. B. Dhaou, "ECDSA-based water bodies prediction from satellite images with UNet," *Water*, vol. 14, no. 14, 2022, Art. no. 2234.

[31] M. A. Najar et al., "Coastal bathymetry estimation from sentinel-2 satellite imagery: Comparing deep learning and physics-based approaches," *Remote Sens.*, vol. 14, no. 5, 2022, Art. no. 1196.

[32] S. Kaur et al., "Transfer learning-based automatic hurricane damage detection using satellite images," *Electronics*, vol. 11, no. 9, 2022, Art. no. 1448.

[33] J. Zhuang, X. Chen, M. Dai, W. Lan, Y. Cai, and E. Zheng, "A semantic guidance and transformer-based matching method for UAVs and satellite images for UAV geo-localization," *IEEE Access*, vol. 10, pp. 34277–34287, 2022.

[34] C. Zhang, Y. Cui, Z. Zhu, S. Jiang, and W. Jiang, "Building height extraction from GF-7 satellite images based on roof contour constrained stereo matching," *Remote Sens.*, vol. 14, no. 7, 2022, Art. no. 1566.

[35] H. Ul Ain Tahir, A. Waqar, S. Khalid, and S. M. Usman, "Wildfire detection in aerial images using deep learning," in *Proc. 2nd Int. Conf. Digit. Futures Transformative Technol.*, 2022, pp. 1–7.

[36] M. K. Hasan, S. Islam, T. R. Gadekallu, A. F. Ismail, S. Amanlou, and S. N. H. S. Abdullah, "Novel EBBDSA based resource allocation technique for interference mitigation in 5G heterogeneous network," *Comput. Commun.*, vol. 209, pp. 320–330, 2023.

[37] N. F. M. Ariffin, F. N. M. Isa, A. F. Ismail, and M. K. Hasan, "Frequency modulated continuous wave radar modeling for landslide detection in Malaysia," in *Proc. Adv. Comput. Commun. Eng. Technol.*, 2016, pp. 1153–1161.

[38] S. A. El Asri, S. El Adib, I. Negabi, and N. Raissouni, "A modular system based on U-net for automatic building extraction from very high-resolution satellite images," in *Proc. E3S Web Conf.*, 2022, vol. 351, Art. no. 01071.

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

[40] J. Liu, M. Wang, L. Bao, and X. Li, "EfficientNet based recognition of maize diseases by leaf image classification," *J. Phys.: Conf. Ser.*, vol. 1693, no. 1, 2020, Art. no. 012148.

[41] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 2–8.

[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[43] R. Garnett, *Bayesian Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[44] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 24, pp. 5–7.

[45] T. Rusch, P. Mair, and K. Hornik, "Structure-based hyperparameter selection with Bayesian optimization in multidimensional scaling," *Statist. Comput.*, vol. 33, no. 1, 2023, Art. no. 28.

[46] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021.

[47] S. H. S. Moosavi and V. K. Bardsiri, "PRO algorithm: A new human-based and multi populations algorithm," *Eng. Appl. Artif. Intell.*, vol. 86, pp. 165–181, 2019.

[48] V. Vinaykumar, J. A. Babu, and J. Frnda, "Optimal guidance whale optimization algorithm and hybrid deep learning networks for land use land cover classification," *EURASIP J. Adv. Signal Process.*, vol. 2023, no. 1, 2023, Art. no. 13.

[49] Z. Yuan, C. Tang, A. Yang, W. Huang, and W. Chen, "Few-shot remote sensing image scene classification based on metric learning and local descriptors," *Remote Sens.*, vol. 15, no. 3, 2023, Art. no. 831.

[50] S. Yang, H. Wang, H. Gao, and L. Zhang, "Few-shot remote sensing scene classification based on multi subband deep feature fusion," *Math. Biosciences Eng.*, vol. 20, no. 7, pp. 12889–12907, 2023.

[51] F. Wang, J. Ji, and Y. Wang, "DSViT: Dynamically scalable vision transformer for remote sensing image segmentation and classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5441–5452, 2023.

[52] Z. Yuan, W. Huang, C. Tang, A. Yang, and X. Luo, "Graph-based embedding smoothing network for few-shot scene classification of remote sensing images," *Remote Sens.*, vol. 14, no. 5, 2022, Art. no. 1161.

**Ameer Hamza** is working toward the Ph.D. degree in computer vision with HITEC University, Taxila, Pakistan.

His major interest include object detection and recognition, video surveillance, medical, and agriculture using deep learning and machine learning. He has published four impact factor papers to date.

**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 2021, respectively.

He is currently an Assistant Professor with Department of Computer Science, HITEC University, Taxila, Pakistan. He has above 280 publications that have more than 10000+ citations and an impact factor of 850+ with h-index 61 and i-Index 165. His primary research interests include medical imaging, COVID-19, MRI analysis, video surveillance, human gait recognition, and agriculture plants using deep learning.

Dr. Khan is reviewer of several reputed journals such as IEEE TRANSACTION ON INDUSTRIAL INFORMATICS, IEEE TRANSACTION OF NEURAL NETWORKS, *Pattern Recognition Letters*, *Multimedia Tools and Application*, *Computers and Electronics in Agriculture*, *IET Image Processing*, *Biomedical Signal processing Control*, *IET Computer Vision*, *Eurasipe Journal of Image and Video Processing*, IEEE ACCESS, *MDPI Sensors*, *MDPI Electronics*, *MDPI Applied Sciences*, *MDPI Diagnostics*, and *MDPI Cancers*.

**Shams ur Rehman** received the master's degree in computer science in 2023 from HITEC University, Taxila, Pakistan, where he is currently a research associate.

He has published one paper in MDPI Diagnostics and currently submitted several papers. His research interest includes medical imaging, remote sensing, and action recognition.

**Hussain Mobarak Albarakati** is with Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia.

He is a Senior Professor of the university where teaching courses related to AI and embedded systems. In addition, he is a senior AI researcher related to remote sensing and medical. He published more than 50 research articles and also a reviewer for several good journals.

**Roobaea Alroobaea** received the bachelor's degree (Hons.) in computer science from the King Abdul-Aziz University, Saudi Arabia, in 2008, and the master's degree in information systems and the Ph.D. degree in computer science from the University of East Anglia, Norwich, U.K., in 2012 and 2016, respectively.

He is currently an Associate Professor with the College of Computers and Information Technology, Taif University, Ta'if, Saudi Arabia. His research interests include human–computer interaction, software engineering, cloud computing, the Internet of Things, artificial intelligence, and machine learning.

**Abdullah M. Baqasah** is with College of Computers and Information Technology, Taif University, Ta'if, Saudi Arabia, Saudi Arabia. He has published more than 40 papers in several reputed journals related to knowledge management and AI. He is also a senior contributor of AI in the same university for the last 2 years.

**Majed Alhaisoni** is currently a Professor of Computer Science with the University of Ha'il Kingdom of Saudi Arabia. He published more than 50 high impact factor papers from last 3 years. His research interest includes artificial intelligence and optimization.

Mr. Alhaisoni is also reviewer of many journals such as *Multimedia Systems*, *Multimedia Tools and Applications*, IEEE TRANSACTION OF PATTEN ANALYSIS AND MACHINE INTELLIGENCE, and few others.

**Anum Masood** received the B.Sc. and M.Sc. degrees in computer science from the COMSATS University Islamabad, Islamabad, Pakistan, in 2012 and 2014, respectively, and the Ph.D. degree in computer science and engineering from the Shanghai Jiao Tong University, Shanghai, China in 2019.

She worked as a Lecturer with the Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan, from 2014 to 2020. She is currently a postdoctoral researcher with the Norwegian

University of Science and Technology, Norway, and affiliated with PET Centre, St. Olav's Hospital, Trondheim, Norway. She also worked as visiting researcher with Institute of Neuroscience and Medicine, Forschungszentrum Jülich, Institute for Cardiogenetics, University of Luebeck, Germany and Liverpool John Moores University, Liverpool, U.K. Her research interests include medical image analysis, automated cancer detection, machine learning, and image processing.