International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P236
Volume 70 Issue 6, 349-365, June 2022

Original Article

A Novel Approach for Satellite Image Classification using Optimized Deep Convolutional Neural Network

Bindhu J S[1], Pramod K V[2]

[1,2]*Department of Computer Applications, Cochin University of Science and Technology, Kerala, India.*

[1]bindhuscholar@gmail.com

Received: 22 April 2022 Revised: 05 June 2022 Accepted: 09 June 2022 Published: 29 June 2022

Abstract *- Recently, remote sensing images have been extensively used in different scene classification. Satellite image classification is used for different applications like land use classification, crop monitoring, forest cover mapping, and natural disaster detection. Accurate classification of scenes in satellite images is very challenging due to the complex, rich details in the images. This work presents a deep learning framework for accurately classifying scene types through improved learning. The proposed approach pre-processes the input image with an adaptive bilateral filtering approach. Then, the pre-processed input image is given as an input to the proposed Optimized Deep Convolutional Neural Network (ODCNN) for improved feature learning and classification. Here, the ODCNN framework is utilized to classify different scenes in the satellite images accurately.*

Moreover, the modified beetle swarm optimization (MBSO) algorithm is utilized for weights optimization in the ODCNN classifier. This process improves the learning of the ODCNN classifier by accurately detecting the scene in remote sensing images. The results of the presented approach are compared with various existing schemes using different performance measures. It is proved that the examined presented approach outperforms the various existing schemes in accuracy (99.75%), precision (99.16%), recall (99.523%), and F-measure (99.34%), kappa measure (0.99), and processing time (10.67 seconds).

Keywords *- Filtering, Feature learning, Optimization, Deep learning, Classification.*

1. Introduction

With the advancements in remote sensing techniques, scene classification plays a vital role in capturing photos and object recognition. Scene classification is a task in which scenes from photographs are orderly classified [1]. It mostly uses the drafts of the objects within the scene and the environmental context for classification [2]. In recent years, the main advancement in visible sensor techniques has been provided due to the attention to scene detection on various inner and outer scene images [3]. Recently, scene detection in remote sensing images has greatly increased attention [4]. Scene detection and classification is a current research topic, and it has vast applications in image recovering, land use, remote sensing, automatic disease detection, face detection, geographical imaging, etc. [5].

Satellite-based scene classification is continuously recorded using satellite imaging sensors, and each satellite image scene consists of multiple lines, and each line indicates the number of components [6]. Satellite imaging techniques have wide applications such as detecting natural disasters, crop monitoring, forest cover mapping, land cover changes, water resource application, dry land mapping, etc. [7]. Satellite images are classified into two types and they are utilized and unutilized areas. Utilized areas are further classified into residential, factories, transport, and cultivation lands [8]. Unutilized areas are further classified into forests, waterfalls, deserts, and sea areas [9]. For continuously mapping these places, image interpretation techniques need more time and field experts [10]. The GPU (Geographical Processing Unit) technique is initially performed to overcome these limitations.

GPU is lightweight with the highest performance computing technology that can find any type of computational necessities based on objective systems [11]. GPU uses RHSeg (Recursive Hierarchical Segmentation) grouping method with the help of hybrid multicore CPU (Central Processing Unit) clusters [12]. RHSeg is a methodology introduced by NASA (National Aeronautics and Space Administration) to present hierarchical classification data with multiple outcomes [13]. There are limitations, such as cost-effectiveness, time consumption, and low performance when many images are used [14]. To overcome this, AI (Artificial Intelligence) based techniques are used.

Machine Learning (ML) is an application of AI that can learn the information of remote sensing images [15]. It is mainly used in sensing socio-economic and environmental conditions in data weakened places [16]. Here ML-based algorithms such as K-Nearest neighbour (KNN) [36], ANN (Artificial Neural Network), Decision tree, and Random

Forest (RF) are used in the field of satellite imagining technique [17]. The accuracy based on land cover mapping is significantly increased compared to GPU technology [18]. At the same time, there is a lack of accuracy in scene classification [19]. The Deep Learning approach is used to classify satellite images to overcome this.

Deep Learning (DL) technique is the division of ML that uses a Convolutional Neural Network(CNN) with multiple layers [20]. CNN is the main part of extracting features in the classification field based on satellite images [21]. In the deep learning technique, the images are first divided, then the feature is extracted based on pixels, and the featured images are classified based on DL algorithms [22]. VGGNet, UNet [23], Dense U-Net+ [37], InceptionV3, along with ImageNet are some of the pre-trained models used for processing, feature extracting, and classifying satellite images [24]. In the Deep Learning technique, for the detection of satellite imaging, there are some limitations such as noise, overfitting problems, non-uniformity of the images, and low batch size compared to other models [25]. To overcome these limitations, a newly proposed methodology with automatic learning of images from the satellite is presented.

Motivation

Scene detection from remote sensing data is a tedious task. The performance of scene detection is enhanced by incorporating the deep learning approach with the effective feature learning processes. The main motivation of the deep learning approach in remote sensing images is the accurate detection of scene types through automatic feature learning. With clear input and creative algorithmic design, the deep learning approach is a promising field in scene classification.

The important contributions of the presented approach are described as follows,

- To enhance the quality of the image, pre-processing is performed with adaptive bilateral filtering.
- To develop an optimal feature learning and classification, and optimized deep convolutional neural network is performed.
- To improve the classifier performance, a modified beetle swarm optimization algorithm is utilized for weight optimization in the ODCNN framework.
- The effectiveness of the proposed methodology is validated using various performance metrics, and the comparison is done with different existing methods.

The paper organization is summarized as Section 2 analyses the current related works, the explanation about the proposed methodology is provided in section 3, Section 4 analyses the results and its corresponding discussion is also performed in this section, and the paper is concluded in Section 5.

2. Related works

Shabbir et al. [26] studied the ensemble of Transfer Learning, VGGNet, and Fine Tuning of ResNet50 to classify scenes. The pre-trained model used here was ResNet50 to network with tuning hyper parameters. For tuning, the linear corrupted scheduler rate is called vector-field. Here, five datasets were used for the analysis: SIRI-WHU, Corel-1.5k, RSSCN (Remote Sensing Scene Classification), Corel-1k, and UCM (UC Merced Land Use). The performance was based on accuracy, precision, recall, and F-1 score. The overall performance based on accuracy was nearly 92%. There were some limitations, such as the Lack of Remote Sensing for forest cover needs more advancements in distant image viewing and image classification in the field of satellite imaging. The efficiency of ResNet50 was very low while using less number of trained models.

Robinson et al. [27] investigated tree-based CNN for image classification in scene classification. Here, a tree-based CNN model and a hidden layer, Maxpool layer, ReLu activation function, fully convoluted layer, epochs, and ROI pooling layer were used for filtering, feature extracting, and segmenting of the images. The tree-based convoluted model compares with HUSTW4, CVEO (Civeo Corporation), and TreeUNet. The dataset used was NWPU_VHR_10 (geospatial object detection dataset) with 60,000 images to implement object classification. The performance was based on accuracy, F-1 Score, recall, and precision. The overall performance is based on accuracy for the Tree-based CNN model, 96.5%, CEVO model 94.8%, HUSTW4 model 95.2%, and TreeUNet model 93.4%. There was some limitation, such as a Lack of data augmentation technique, and the efficiency of this method was low compared to other models. The bulk of storage was required while using many satellite images. Here only a few pre-trained models were used for classification.

Zhang et al. [28] developed the pre-trained model called CNN-CapsNet comes under two models, namely: CNN (VGG-16 and Inceptionv3) and CapsNet (Capsule Neural Network) for scene detection. The CNN model was mainly used for feature extraction, and the extracted image was given as input to CapsNet for the final classification result. Three datasets were used: the UCM dataset with 21 images, AID (Association for India's development) dataset with 30 images, and the NWPU-RESISC45 (Northwestern Polytechnic University-Remote Sensing Image-Scene Classification) dataset with 45 images. The performance was calculated based on accuracy. The overall accuracy for UCM is 94%, AID dataset 96%, and NWPU-RESISC45 dataset 90%. The limitations they have highlighted with the Lack of merging featured maps with different pre-trained CNN models. Moreover, it was a very complex and time-consuming process

Using deep learning methods, Xu et al. [29] performed a novel approach for scene classification from remote sensing images. This method's classification method was Recurrent Neural Network (RNN) along with RF for land cover mapping using satellite images. The dataset used here was UCM with 21 classes of high-resolution images. This dataset used image clipping and filtering techniques to avoid radiometric corrections, noise, and other discrepancies. The performance was calculated based on accuracy, precision, and recall. The overall performance based on accuracy was 95%. There were some limitations, such as the datasets used here being small. This method was not suitable for large and complex image scene classification. The featured maps used here cause high overlapping. Lack of mapping forest cover, highly confused among residential and beach images due to the low number of datasets, and the error rate was very high due to improper image classification.

Cheng et al. [30] performed remote sensing classification using deep learning techniques and the role in methods, their benchmarks, and opportunities. The pre-trained model used here was auto-encoders, CNN, and GAN (Generative Adversarial Network) based model. The dataset used here was NWPU-RESISC45, UCM, and AID dataset. The performance was calculated using a confusion matrix. The overall performance-based in accuracy for the AID dataset was 96%. There were some limitations, such as when a large number of datasets were used, the performance would be very low, this method was very complex and time-consuming, the method called auto-encoders had very low accuracy due to the Lack of advancements, and the algorithms used in GAN was very complex with less compatibility.

Anwer et al. [31] investigated the compact deep color features for remote sensing classification. Here deep CNN with normalization, pooling, and Fully Connected (FC) layers were used. The datasets used here were as follows: UCM with 21 images, WHU-RS19 with 19 images, RSSCN7 with 7 images, AID with 30 aerial images, and NWPU-RESISIC45 with 45 images. The ImageNet dataset was used for extracting features. The performance was calculated using compact color feature fusion, standard RGB (Red, Green, Blue) deep features, and RS scene classification datasets. The overall accuracy for UCM, WHU-RS19, AID and NWPU-RESISC45 were 96%, 98%, 94%, and 87%, respectively. There were limitations such as a Lack of integration of texture features with color features, and only a few colors can be classified using this method, and need more advancements in multi-color deep feature fusion for object detection.

Liechang et al. [32] studied the remote sensing image scene classification based on the fusion method. The pre-trained models used here were VGG16, MobileNet, and ResNet50. The datasets used here were: UCM from national map urban area imagery collection of 21 images, WHU-RS19 from satellite images using Google Earth, and NWPU-RESISC45 from Google Earth with 31,000 images. The performance was based on the accuracy of the front side, middle side, and back side fusion methods. The overall accuracy is based on the front side at 91%, the middle side at 90%, and the back side at 92%. Here, the fusion method is too complex and needs more special architectures; this method was more expensive and needed more computation and storage. The training model used in the middle side fusion method was inefficient and Lacked enough libraries for further improvement.

Yue et al. [33] developed the remote sensing classification based on a high-order graph convolutional network. The pre-trained model used here was the CNN model based DenseNet and MobileNetV2. It is mainly used for extracting the feature images. The features were gained through GAP (Global Average Pooling and H-GCN (High Order Graph Convoluted Network) to cover the semantic features of images. The datasets used were UCM, RSSCN7, AID, and NWPU-RESISC45. The SGD (Stochastic Gradient Descent) method was used for optimizing the dataset images. The overall accuracy was nearly about 97%. Here also were some limitations, such as a Lack of key components within the featured images, automatic selection of different structures of images, a need for more improvement in image classification, and a time-consuming process. The comparison of different existing techniques is provided in table 1.

Table 1. Comparison of different existing techniques

Author &Reference	Techniques	Datasets	Merits	Demerits
Shabbir et al. [26]	ResNet50, VGGNet.	UCM, RSSCN SIRI-WHU, Corel-1k and Corel-1.5k.	Noise-free, low power consumption.	Lack of forest cover, low efficiency, Lack of combining natural image with RS image.
Robinson et al. [27]	Tree-based CNN model.	NWPU_VHR_10 dataset	Fast output, low energy consumption.	Lack of data augmentation technique, low
Zhang et al. [28]	CNN (VGG-16 and Inceptionv3) and CapsNet.	UCM, AID dataset, NWPU-RESISC45	High compatibility, high efficiency, and superior output without data augmentation	Lack of merging featured maps with different pre-trained CNN models, highly
Xu et al. [29]	RNN along with RF.	UCM dataset	Without feature extraction, weather prediction can be done easily, with a simple method, and with high efficiency.	Datasets used here were small. This method was not suitable for large and complex image scene classification; the error rate was very high
Cheng et al. [30]	Auto-encoders, CNN, GAN.	NWPU-RESISC45, UCM, and AID dataset.	High efficient, fast output requires less storage.	The algorithms used in GAN were very complex with less compatibility. Auto-encoders have very low
Anwer et al. [31]	Deep CNN with normalization, pooling, and FC layers.	UCM, WHU-RS19, RSSCN7, AID and NWPU-RESISIC45.	High dimensionality with compact output, highly compatible, simple method.	Need more advancements like using multi-color deep feature fusion for object detection, Lack of integration of texture features with color features, and only a few colors can be classified.
Liechang et al. [32]	VGG16, MobileNet and ResNet50	UCM, WHU-RS19, NWPU-RESISIC45.	Low error rate, the fusion method used here shows high performance due to efficient training sets and high efficiency.	Too complex a process and needed more special architectures; this method was more expensive and needed more computation and storage.
Yue et al. [33]	DenseNet and MobileNetV2.	UCM, AID, RSSCN7 and NWPU-RESISC45 datasets.	Fast output, the presence of attention module helps discriminate the images based on CNN models, Simple method.	Lack of key components within the featured images, Lack of automatic selection.

The existing studies introduce various approaches for scene classification in remote sensing images. Most approaches have used artificial intelligence (AI) for scene classification but fail in accurate scene classification. Moreover, the process is very complicated and takes much time for scene classification. An effective deep learning framework is presented for accurate scene classification through an optimal feature learning process to overcome these issues. The proposed architecture has introduced ODCNN, a metaheuristic, and a deep CNN combination.

CNN provides an accurate detection rate for all kinds of the classification process. Therefore, deep CNN is integrated with the proposed framework. Further, the processing time of the proposed classification technique is improved by combining a metaheuristic optimization algorithm along with the proposed DCNN. This combination reduces the processing time by selecting the optimal weight parameter required for efficient classification.

3. Proposed methodology

The proposed framework accurately detects scene types in remote sensing images using ODCNN. At first, the pre-processing is done over the input image with an adaptive bilateral filtering technique and given as an input to a proposed optimized deep convolutional neural network. The ODCNN framework accurately calculates different scenes in the satellite images. Moreover, the modified beetle swarm optimization (MBSO) algorithm is utilized for the weights optimization of the ODCNN classifier. This process improves the performance of the ODCNN classifier by accurately detecting the scene in remote sensing images. The schematic architecture of the proposed scene detection methodology is depicted in figure 1.

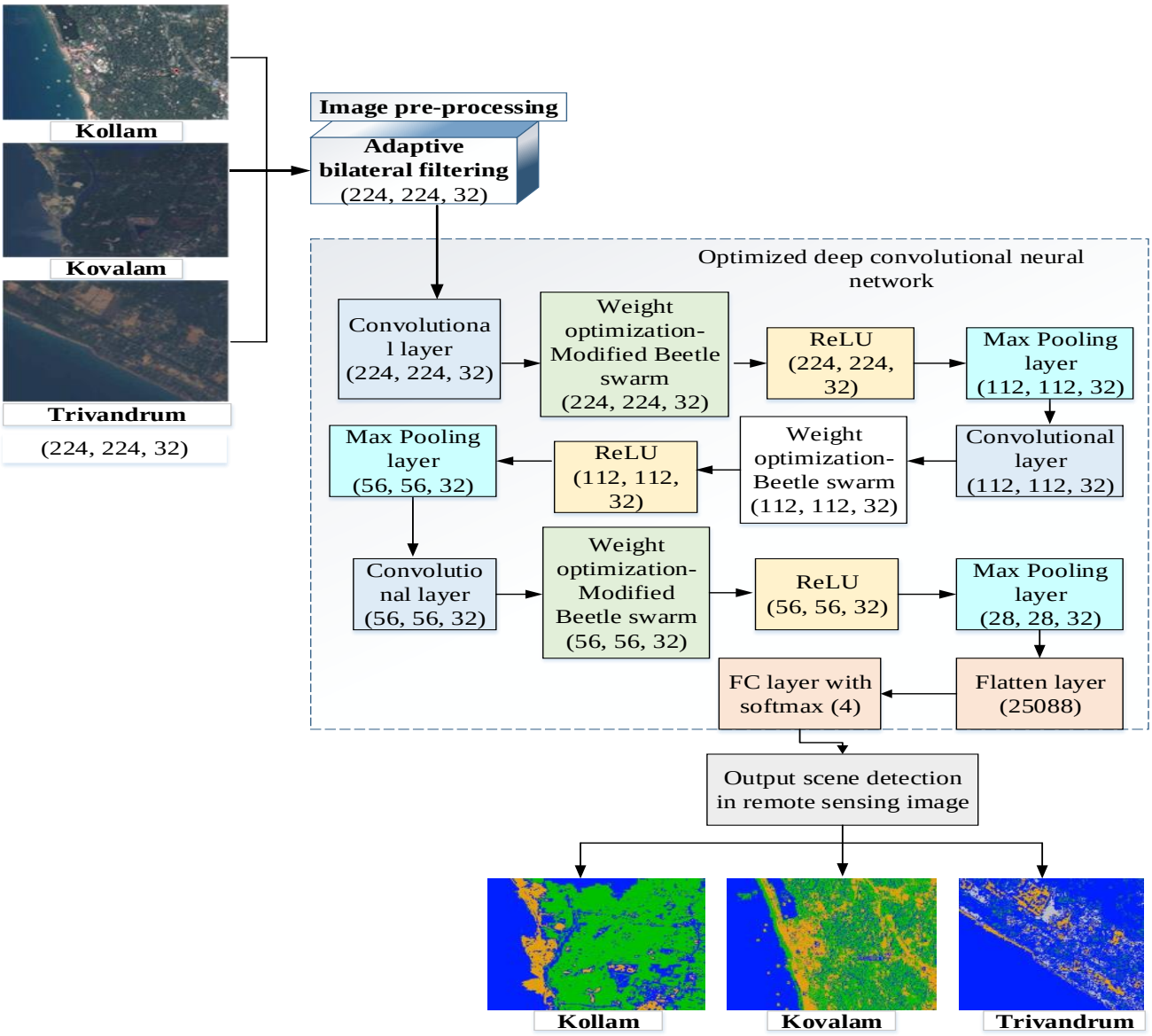

Fig. 1 Schematic diagram of the proposed methodology

3.1 Pre-processing: Adaptive bilateral filter

The input image is initially pre-processed with an adaptive bilateral filtering approach. Bilateral filtering is an edge-preserving and noise-eliminating effective smoothing filter in images. The proposed adaptive bilateral filtering not only smoothens the image and also sharpens the image edges. The data points are replaced by the weighting points to achieve the accuracy of an image. The adaptive filtering is described in equation (1)

$$AB_F = \frac{1}{2}\left(k_1 \sigma_n \sqrt{2 \log Q} + k_2 \frac{\sigma_n^2}{\sigma_s} \right) \quad (1)$$

In equation (1), a standard deviation of the noisy and noiseless images is denoted as σ_n representing the total number of pixels. AB_F represents the adaptive bilateral filtering, and the constant parameters are denoted as k_1 and k_2 respectively.

3.2 Scene classification in satellite images using optimized deep CNN

The ODCNN framework operates feature extraction and classification decision in images. The developed ODCNN framework comprises a convolutional layer, max-pooling layer, and the FC layer. The structure of the optimized convolutional neural network is depicted in figure 2.

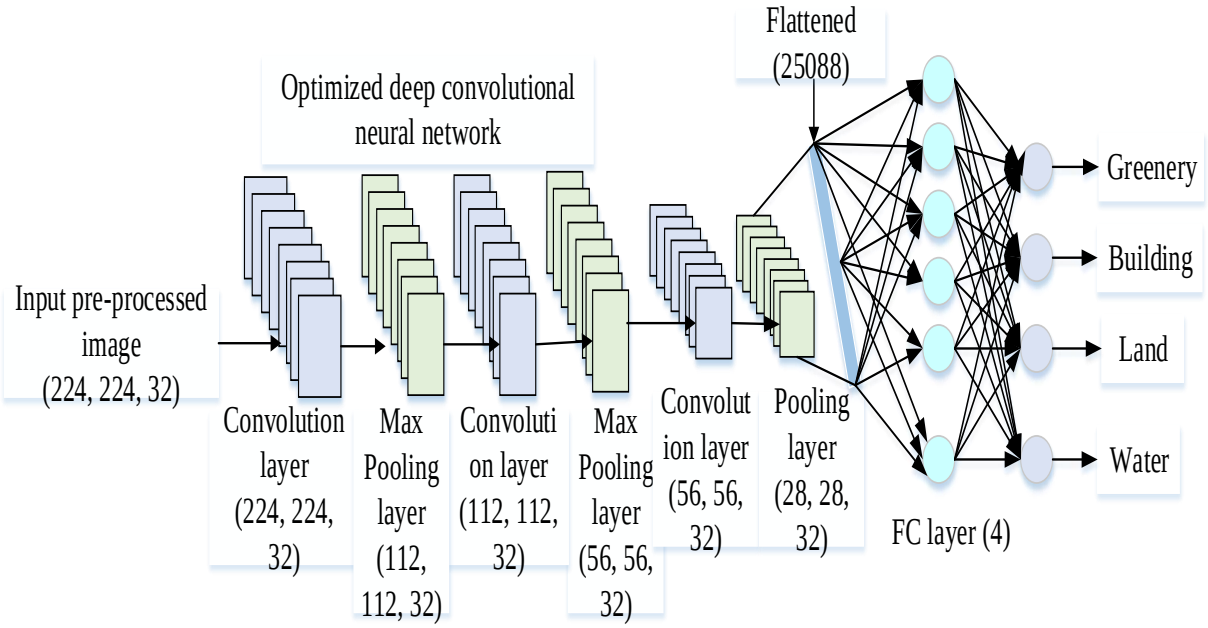

Fig. 2 Optimized CNN structure

Figure 2 shows the structure of the optimized convolutional neural network. This framework performs feature learning using a convolutional and max-pooling network. The FC layer performs the scene type classification. These layers are described in the subsequent sub-sections.

3.2.1. Convolutional layer

The pre-processed image is given as an input to the DCNN. The convolutional layer extracts the features from the pre-processed image. The convolutional layer performs the convolution operation, which performs the linear multiplication of weights with the input data. The operation of the convolutional layer is described in the subsequent condition (2),

$$\bar{Z} = \bar{Y} * W_m \quad (2)$$

Here $\bar{Z}$ the convolutional layer output, $\bar{Y}$ the input data, and W_m the weight matrix. The deep framework initially applies the convolution operation on the images to directly extract the features. This feature extraction increases the ability of scene detection in remote sensing images. This framework considers images' distributed information learned for further processing. Here, the optimized weight data is updated by the modified beetle swarm optimization approach. The initial hidden layer is the convolutional layer, comprised of 32 feature maps. Here, the rectifier function has been incorporated for attaining the linear outputs.

3.2.2. Weight optimization

At first, the arbitrary weights are initialized, and the optimal weights are obtained according to the beetle search

behavior. The initial arbitrary weight vectors are described in the subsequent condition (3),

$$\vec{B} = \frac{R\left(\overline{D},1\right)}{\left\|R\left(\overline{D},1\right)\right\|} \tag{3}$$

Here, R represents the random function, $\overline{D}$ the dimension of search space, and $\|\cdot\|$ the norm function. Then, the coordinates of spatial search with right and left sides are described as follows,

$$y_{right}^{m} = y^{m} + \left(\bar{d}^{m} \cdot \vec{B}\right) \tag{4}$$

$$y_{left}^{m} = y^{m} - \left(\bar{d}^{m} \cdot \vec{B}\right) \tag{5}$$

Here, $\vec{B}$ represents the arbitrarily initialized weights, $\bar{d}^{m}$ the feature space dimension, and y^{m} the feature data. Afterward, the initial optimized weight factor is computed in the iterative condition (6),

$$W_m = \left(\frac{M_I - C_I}{M_I}\right)\left(W_{\max} - W_{\min}\right) + W_{\min} \tag{6}$$

Here, $W_{\max}$ represents the maximum weight value, $W_{\max}$ represents the minimum weight value, M_I represents the maximum iteration, W_m represents the weight factor and M_I represents the current iteration. The weight updated maximum iteration is expressed as per the subsequent condition (7),

$$y^{m+1} = y^{m} + \tilde{\varepsilon} \times \vec{B} \times W_m \times Sign\left(\bar{f}\left(y_{right}^{m}\right) - \bar{f}\left(y_{Left}^{m}\right)\right) \tag{7}$$

Here, $\tilde{\varepsilon}$ represents the step size of the beetle equivalent to 0.95. The local optima issue is solved using the step size value of 0.95. The updated weight factor is represented as W_m, sign function is represented as $Sign$, $\bar{f}\left(y_{right}^{m}\right)$ represents the generated right side weight data and $\bar{f}\left(y_{right}^{m}\right)$ represents the generated left side weight data. The convolutional layer initially extracts the low level of features, and the remaining layers extract the higher level of features for learning.

3.2.3. Rectified linear unit (ReLU)

After the convolution operation, ReLU is adapted as an activation function due to its linearity. This activation function is important for transforming the convolution output to the next layer. In this layer, the non-zero elements are removed from the data. It is utilized for providing outputs in linear form. The output data is the same if it is positive, and the value is zero if it is negative. It is expressed by the subsequent condition (8),

$$\mathrm{Re}\,Lu(Z) = Max(0, \bar{Y}) \tag{8}$$

This layer function is performed to attain a rectified feature map.

3.2.4. Max pooling layer

This layer is one of the feature extraction layers, and this layer performs the pooling operation and chooses the maximum values from the feature map. Hence, the output of this layer will be the prominent features of the previous layer. It is expressed in the condition (9),

$$O_{pooling} = \max\left(F_m\right), \quad m = every\ block \tag{9}$$

Here, $O_{pooling}$ the max-pooling layer output and F_m the feature value with block size m .are represented. This layer decreases the size of features by selecting the maximum units in each feature map.

3.2.5. Fully connected layer

The output attained in the max-pooling layer is given as an input to the FC layer. The features from the pooling layer are flattened and given as an input to the FC layer. In this, the two-dimensional data is converted into a one-dimensional vector.

Softmax: The FC layer uses the softmax activation function to get the probabilities of input pixels for assigning in a particular class. It results from the output labeling by detecting the data class in remote sensing images. Various types of scene labeling in remote sensing images are expressed as,

$$\bar{h}(y) = \begin{bmatrix} p\left(z^{d} = 1 | y, \theta\right) \\ p\left(z^{d} = 2 | y, \theta\right) \\ p\left(z^{d} = 3 | y, \theta\right) \\ \vdots \\ p\left(z^{d} = T | y, \theta\right) \end{bmatrix} \tag{10}$$

Here, $p\left(z^{d} = T | y, \theta\right)$ represents the probability of output class with label T, $p\left(z^{d} = T | y, \theta\right)$ represents the scene type in images. This layer makes the final decision in scene detection of remote sensing images. In this layer, the number of neurons is equivalent to the type of classes that inferred. The pseudo code of the proposed approach is provided in algorithm 1.

Algorithm 1. Pseudo code of proposed methodology

Input:

Input image $(\bar{I})$ with size 224×224, number of iterations $(i_{\max})$, learning rate=0.001, kernel size 3×3 and batch size 32.

Output:

Predicted scene types in input remote sensing image $(\bar{I})$

Begin

Pre-process the input image $(\bar{I})$ // adaptive bilateral filtering

For j=1 to M pixels do

For k=1 to N pixels do

Compute equation (1) to attain filtered image.

Fed the image samples to DCNN for feature learning and classification

For image pixels $i = 1$ to P_k do // feature extraction

For layers (K): $1 \to k-1$ do // here feature extraction layers K (Convolutional, Max pooling, Convolutional, Max pooling, Convolutional, Max pooling).

For j=1 to M pixels do

For k=1 to N pixels do

Obtain $\bar{Z} = \bar{Y} * W_m$ //Convolution operation

End

End

For $\bar{Y}$, initialize random population

Compute equations from (3) to (7)// Weight optimization

End

For $\bar{Z}$,ReLu function $\mathrm{Re}Lu(Z) = Max(0, \bar{Y})$

End

For j=1 to M pixels do

For k=1 to N pixels do

Obtain

$$O_{pooling} = \max(F_m), \quad m = every\ block$$

End

End

Repeat feature learning N=3// convolution to max pooling function

Attain the edge feature map $\bar{F}_m$

For features $\bar{F}_m$ isolate the features based each scene in the data// data flatten

Compute equation (10)//final decision with fully connected layer

For j=1 to M pixels do

$t\arg et = 0$

For k=1 to N pixels do

$t\arg et = tmp++$

End for

End

End

Return predicted scene labels in remote sensing images.

End

In algorithm 1, the pseudo-code of the presented approach is provided. The accurate detection of scene types such as greenery, bare land, water body, and the building is obtained by the optimal feature learning and classification using the optimized deep convolutional neural network.

4. Results and Discussion

The experimental analysis using the proposed ODCNN-based scene classification from remote sensing data is examined in this section. The presented scene type classification is implemented in the PYTHON platform, Anaconda – Spyder (tensorflow) environment. The performance achieved by the proposed framework is compared with the various existing methods in terms of accuracy, F-measure, Precision, and recall. The four different classes that are classified using this presented methodology are Class 1 (Greenery), Class 2 (water bodies), Class 3 (bare land), and Class 4 (buildings). The input remote sensing images of three different places, Kollam, Kovalam, and Trivandrum in India, are depicted in figure 3. The classified images of the Kollam, Kovalam, and Trivandrum regions in India are depicted in figure 4.

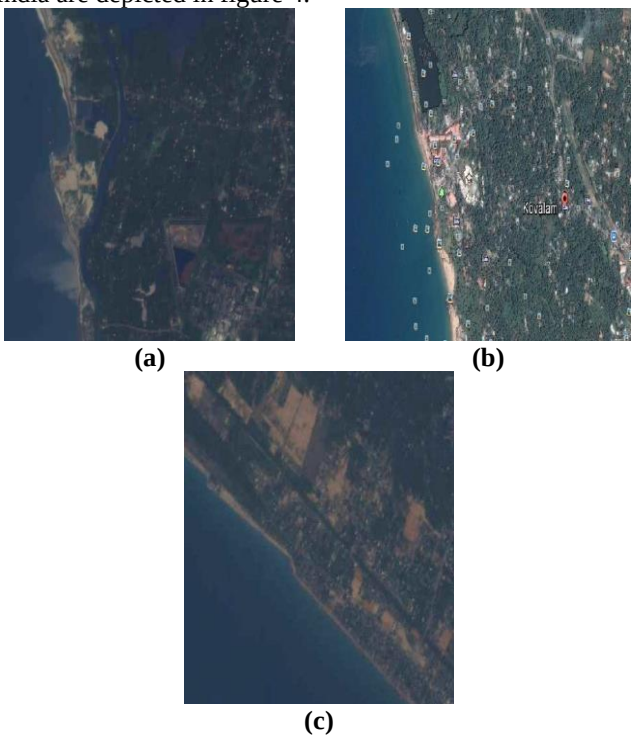

Fig. 3 Input remote sensing images (a) Kollam, (b) Kovalam, and (c) Trivandrum

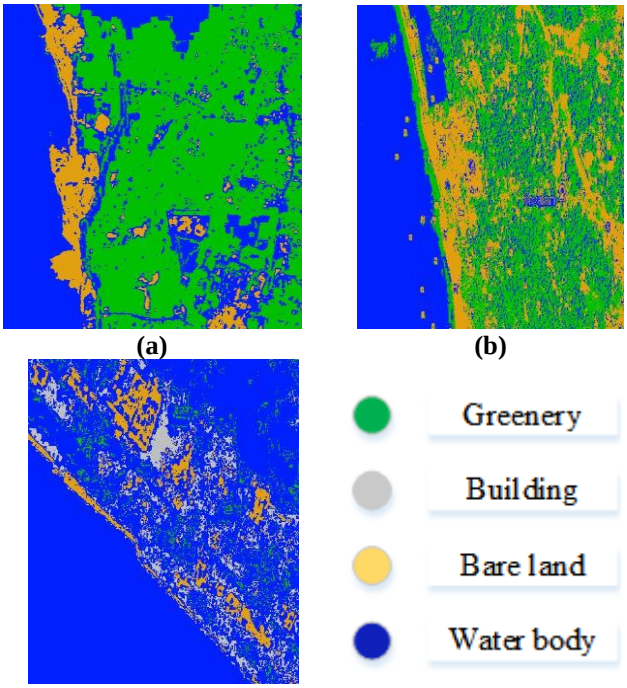

Fig. 4 Classified remote sensing images (a) Kollam, (b) Kovalam, and (c) Trivandrum

4.1. Learning data analysis

The deep learning framework performs the classification on unsupervised data, and this performs the improved learning. The hyper parameter in the CNN design is comprised of the number of filters, kernel size, and learning rate. Filter size is based on the amount of information taken in the convolution step as features. The chosen filter size is 32, kernel size is 3×3, and the learning rate is 0.001. The hyper parameter selection enhances the performance of classification. The activation and loss function is an important function used in the deep learning framework, and it enables the network for capable of learning on layer-wise mapping. The input size of the image is $224\times224\times1$. Scene classification is the process of learning and finding the scene types in the remote sensing images. The categorical cross-entropy is used as a loss function for multiple scene classification. The multi-class outputs are obtained with the classification model. The hyperstimulation parameters used for deep learning classification are provided in table 2.

Table 2. Learning parameters

Parameters	Specification
Learning rate	0.001
Kernel or filter size	3×3
Input image size	224×224
Batch size	32
Epochs	100
Activation function	Softmax
Padding	Same
Pooling size	2×2

The optimal hyper learning parameter's selection enhances the presented approach's performance and achieves improved results than the existing schemes. The kernel size used in the convolution layer is 3×3, and the kernel size used in the pooling layer is 2×2. The batch size describes the amount of data taken for iteration processing. Here, to attain better training stability, the chosen batch size is 32. The smaller kernel size decreases the computational cost and time. The optimal choice of the kernel size is 3×3. The kernel size 1×1 is utilized only for decreasing the feature size, and it is not preferred due to capturing features in a 1-pixel feature map and eliminating important features.

Moreover, 2×2, 4×4 size is not considered due to its even size, and it causes distortions among the layers. The maximum weight value in the optimization approach is 8.811, and the minimum weight value is 0.60. Here, the weight value is increased to improve the global search time. The optimal weight value is 0.9, and the optimal result is attained at the 100^{th} iteration.

4.2. Performance metrics

Different performance metrics are analyzed, and a comparison is done between the proposed and the different existing approaches. The accuracy, F-measure, Precision, and recall performances are evaluated, which are defined in the subsequent sub-sections.

4.2.1. Accuracy

This performance measure is the proportion of accurately predicted classes to the total number of classes. It computes the accurate prediction of classes, and it is evaluated by the subsequent condition (11),

$$\overline{A}_y = \frac{t_{(positive)} + t_{(negative)}}{t_{(positive)} + f_{(positive)} + f_{(negative)} + t_{(negative)}} \tag{11}$$

Here, $\overline{A}_y$ represents the accuracy, $t_{(negative)}$ represents the true negative, $f_{(positive)}$ represents the false positive, $t_{(positive)}$ represents the true positive, and $f_{(negative)}$ represents the false negative.

4.2.2. Precision

This metric is utilized to predict the relevant positive class among all the positive classes. This measure is expressed in the condition (12),

$$\overline{P}_n = \frac{t_{(positive)}}{t_{(positive)} + f_{(positive)}} \quad (12)$$

Here, $\overline{P}_n$ represents the precision metric.

4.2.3. Recall

This metric is utilized to compute the proportion of correctly predicted classes among the total data under a particular category. It is computed as per the condition (13),

$$\overline{R_L} = \frac{t_{(positive)}}{t_{(positive)} + f_{(negative)}} \quad (13)$$

Here, $\overline{R_L}$ represents the recall performance.

4.2.4. Kappa statistics

This performance evaluates the probability level among the classified data. It is computed by the subsequent condition (14),

$$\overline{K}_a = \frac{\overline{a}_0 - \overline{a}_v}{1 - \overline{a}_v} \quad (14)$$

Here, $\overline{K}_a$ represents the kappa statistical metric, $\overline{a}_0$ and $\overline{a}_v$ evaluations are expressed as,

$$\overline{a}_0 = \frac{[t_{(negative)} + t_{(positive)}]}{N} \quad (15)$$

$$\overline{a}_v = \frac{[(t_{(negative)} + f_{(negative)}) * (t_{(negative)} + f_{(positive)}) + (f_{(positive)} + t_{(positive)}) * (f_{(negative)} + t_{(positive)})]}{N^2} \quad (16)$$

Here, $\overline{a}_0$ represents the observed accuracy, N the total number of data, and the accuracy probability.

4.2.5. F-measure

The F-measure performance is estimated by using both precisions and recall performance measures. This F-measure performance is expressed in the condition (17),

$$\overline{F}_M = 2 \times \frac{(\overline{R_L} * \overline{P}_n)}{(\overline{R_L} + \overline{P}_n)} \quad (17)$$

Here, $\overline{F}_M$ represents the F-measure performance, $\overline{R_L}$ the recall measure, and the precision measure.

4.3. Performance analysis

Performance achieved by proposed ODCNN-based classification using remote sensing images and the comparative analysis between the proposed and existing methods in terms of different performance metrics is discussed in this section. The confusion matrix attained for the Kollam image is depicted in figure 5.

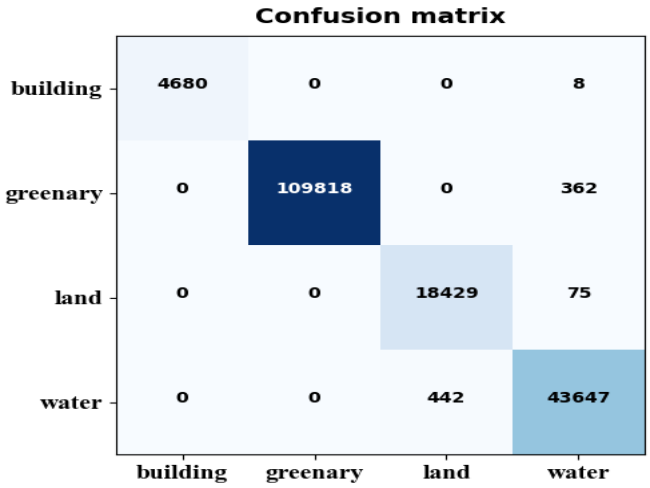

Fig. 5 Confusion matrix (Kollam)

In figure 5, the generated confusion matrix of the Kollam region image. Here, the correctly predicted data samples in the building class are 4680 samples, greenery class is 109818 data samples, land class is 18429 data samples, and water class is 43647 data samples. Each performance evaluation is based on the confusion matrix's obtained false positive, true positive, false negative, and true negative values. Moreover, the confusion matrix of the Kovalam region remote sensing image is depicted in figure 6.

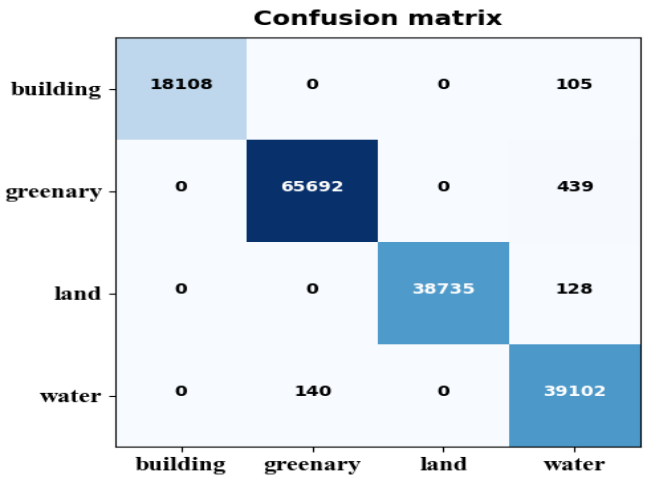

Fig. 6 Confusion matrix (Kovalam)

In figure 6, the confusion matrix of the Kovalam region remote sensing image. Here, the accurately detected data samples in building class are 18108, accurately detected data samples in greenery class are 65692, the amount of accurately predicted data in land class is 38735, and the accurately predicted data in water class is 39102. Furthermore, the confusion matrix of the Trivandrum region is depicted in figure 7.

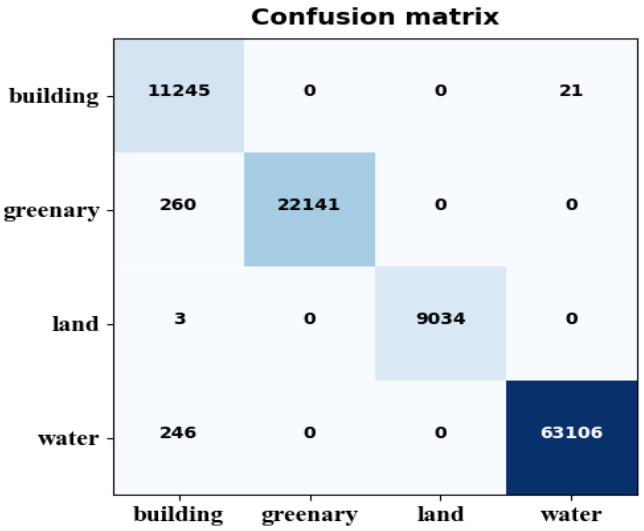

Fig. 7 Confusion matrix (Trivandrum)

In figure 7, the confusion matrix of the Trivandrum region is depicted. Here, the predicted data in the building class is 11245, the amount of correctly predicted data in the greenery class is 22141, the amount of correctly predicted data land class is 9034, and the number of accurately detected data samples is 63106. The overall performance of the presented approach for different performance metrics is provided in table 3.

Table 3. Overall performance of the presented approach

Performance metrics	Performance value (Kollam)	Performance value (Kovalam)	Performance value (Trivandrum)
Accuracy (%)	99.75	99.75	99.75
Precision (%)	99.16	99.52	98.90
Recall (%)	99.52	99.51	99.55
F1-score (%)	99.34	99.519	99.22
Kappa	0.9927	0.988	0.9895
Processing time	10.67 seconds		

In table 3, the performance of the presented approach for varying performance metrics is given. The processing time taken to detect the scene in the satellite image is 10.67 seconds. The performance value of the Kollam region image in terms of accuracy, precision, recall, F1-score, and kappa is 99.75%, 99.16%, 99.52%, 99.34%, 0.9927 respectively. Similarly, the Kovalam region image in terms of accuracy, precision, recall, F1-score, and kappa is 99.75%, 99.52%, 99.51%, 99.519%, 0.988, respectively. Moreover, the performance of Trivandrum image region is 99.75%, 98.90%, 99.55%, 99.22% and 0.9895 respectively. It proved that the optimal selection of parameters in deep learning enhances the output performance. Moreover, the performance evaluation of each predicted scene class is provided in table 4.

Table 4. Class-wise performance of remote sensing image (Kollam)

Performance metrics	Scene types			
	Building	Greenery	Land	Water
Accuracy (%)	99.995	99.79	99.70	99.5
Precision (%)	100	100	97.65	98.99
Recall (%)	99.82	99.67	99.59	98.99
F1-score (%)	99.34	99.519	99.22	98.997
Kappa	0.99	0.99	0.98	0.98

In table 4, the performance of four classes is given. It is the analysis of the performance of each class in different performance measures for the Kollam region. The accuracy performance value of building, greenery, land, and water scenes is 99.995%, 99.79%, 99.70%, and 99.5%, respectively. The precision performance value of building, greenery, land, and water scenes is 100%, 100%, 97.65%, and 98.99%, respectively. The recall performance value of

building, greenery, land, and water scenes is 99.82%, 99.67%, 99.59%, and 98.99%, respectively. The F1-score performance value of building, greenery, land, and water scenes is 99.34%, 99.519%, 99.22%, and 98.997%, respectively. The kappa performance value of building, greenery, land, and water scenes is 0.99, 0.99, 0.98, and 0.98, respectively. It indicates the improvement of the proposed method. Similarly, the class-wise performance comparison on the Kovalam image is provided in table 5.

Table 5. Performance of remote sensing image for 4 different classes (Kovalam)

Performance metrics	Scene types			
	Building	Greenery	Land	Water
Accuracy (%)	99.93	99.64	99.92	99.5
Precision (%)	100	99.78	100	98.31
Recall (%)	99.42	99.33	99.67	99.64
F1-score (%)	99.34	99.519	99.22	98.97
Kappa	99.71	99.56	99.83	98.97

Table 5 shows the class-wise performance comparison on the Kovalam image. The presented approach attains improved performance in each class for the Kovalam region. The accuracy performance value of building, greenery, land, and water scenes is 99.93%, 99.64%, 99.92%, and 99.5%, respectively. The precision performance value of building, greenery, land, and water scenes is 100%, 99.78%, 100%, and 98.31%, respectively. The recall performance value of building, greenery, land, and water scenes is 99.42%, 99.33%, 99.67%, and 99.64%, respectively. The F1-score performance value of building, greenery, land, and water scenes is 99.34%, 99.519%, 99.22%, and 98.997%, respectively. The kappa performance value of building, greenery, land, and water scene is 0.9971, 0.9956, 0.9983, and 0.9897, respectively. It indicates the improvement of the proposed method. Similarly, the class-wise performance comparison on the Trivandrum image is provided in table 6.

Table 6. Performance achieved by 4 different classes for image (Trivandrum)

Performance metrics	Scene types			
	Building	Greenery	Land	Water
Accuracy (%)	99.50	99.75	99.99	99.74
Precision (%)	95.66	100	100	99.96
Recall (%)	99.82	99.67	99.59	98.99
F1-score (%)	99.81	98.83	99.96	99.611
Kappa	97.69	99.41	99.98	99.78

Table 6 shows the class-wise performance comparison on the Trivandrum image. The presented approach shows better performance for each class. The accuracy performance value of building, greenery, land, and water scenes is 99.5%, 99.75%, 99.99%, and 99.74%, respectively. The precision performance value of building, greenery, land, and water scenes is 95.66%, 100%, 100%, and 99.96%, respectively. The recall performance value of building, greenery, land, and water scenes is 99.82%, 99.67%, 99.59%, and 98.99%, respectively. The F1-score performance value of building, greenery, land, and water scenes is 99.81%, 98.83%, 99.96%, and 99.611%, respectively. The kappa performance value of building, greenery, land, and water scene is 0.9769, 0.9941, 0.9998, and 0.9978, respectively. It indicates the improvement of the proposed method. Moreover, the comparative results for each compared and proposed approach in terms of overall accuracy are depicted in figure 8.

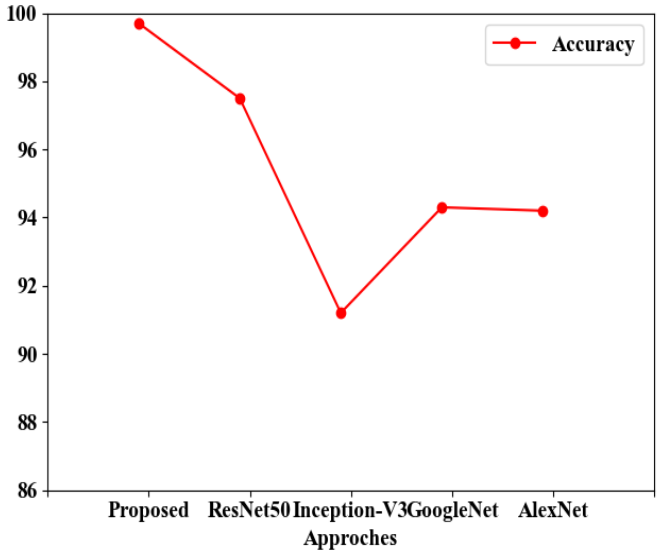

Fig. 8 Performance analysis of overall accuracy

In figure 8, the achieved accuracy of proposed and compared techniques. Here, the proposed performance is compared with existing ResNet50, Inception_V3, GoogLeNet, and AlexNet [34]. The Inception_V3 framework used the residual connections in learning. The GoogLeNet framework used deep learning-based feature extraction to achieve better scene classification. AlexNet provides deep learning-based scene classification using different dataset images. Here, the accuracy comparison among the proposed and existing frameworks is carried out for the UCM dataset. The comparison analysis of processing time is mentioned in table 7.

Table 7. Comparison of processing time and accuracy

Techniques	Processing time (seconds)	Accuracy (%)
Tree based CNN	176	96.5
CEVO	360	94.8
HUSTW4	325	95.2
TreeUNet	400	93.4
Proposed	**10.67**	**99.75**

In table 7, the performance comparison on processing time is depicted. A fair comparison is not possible when the proposed approach uses private datasets and not public datasets. Here, the processing time of the presented approach is 10.67 seconds, which is much lesser than the existing tree-based CNN (176 seconds), CEVO (360 seconds), HUSTW4 (325 seconds), and TreeUNet (400 seconds) [35] approaches. Tree-based CNN approach used pixel-based segmentation before feature extraction to attain the enhanced performance. CEVO approach used an attention-based semantic segmentation to enhance the scene image classification. This approach attains the important features on different scales. HUSTW4 approach is used for the high-resolution image segmentation.

Moreover, the encoder and decoder-based approach is incorporated for scene representation. TreeUNet framework is used for semantic segmentation in complex remote sensing images. Figure 9 depicts the processing time comparison.

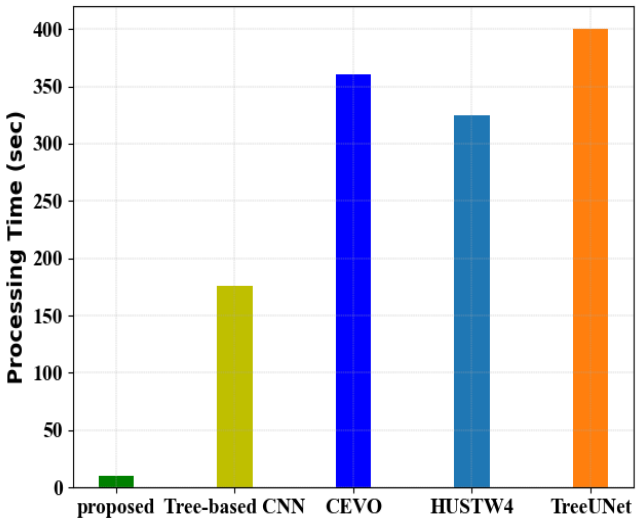

Fig. 9 Comparison analysis of processing time

The processing time obtained for the proposed architecture is much reduced than the existing approaches, which are illustrated in figure 9. Moreover, the F-measure performance of four scene classes like greenery, water bodies, land, and the building is illustrated in figure 10.

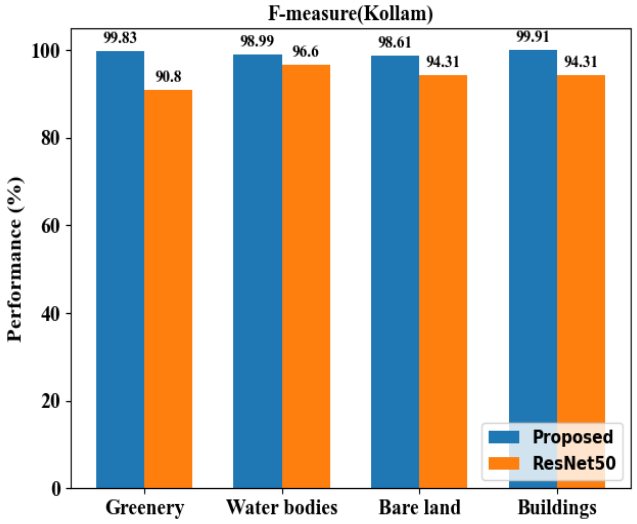

Fig. 10 F-measure comparison (Kollam)

In figure 10, the F-measure comparison for different scene types is illustrated. Here, the presented approach performance is compared with the existing ResNet50 framework [34]. The ResNet50 performs the scene classification based on the mid-level feature extraction and transfer learning. The obtained F-measure of the greenery region is 99.11%, the water body region is 99.78%, the bare land region is 99.98%, and the building is 97.69%. Figure 10 displays the F-measure value of the presented approach is much higher than the existing ResNet50 framework. Furthermore, the F-measure comparison for Kovalam is depicted in figure 11.

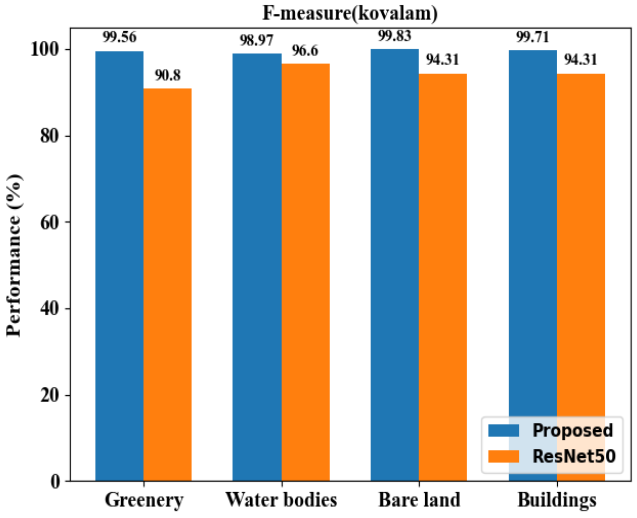

Fig. 11 F-measure comparison for the Kovalam area

Figure 11 illustrates the F-measure comparison of the Kovalam image. Here, the performance of the proposed framework is compared with the existing ResNet50. The

obtained F-measure of greenery class is 99.56%, water body class is 98.97%, bare land class is 99.83%, and the building class is 99.71%. It proved that the attained F-measure value of the presented architecture is much higher than the existing ResNet50 framework. Similarly, the F-measure comparison with the Trivandrum image is depicted in figure 12.

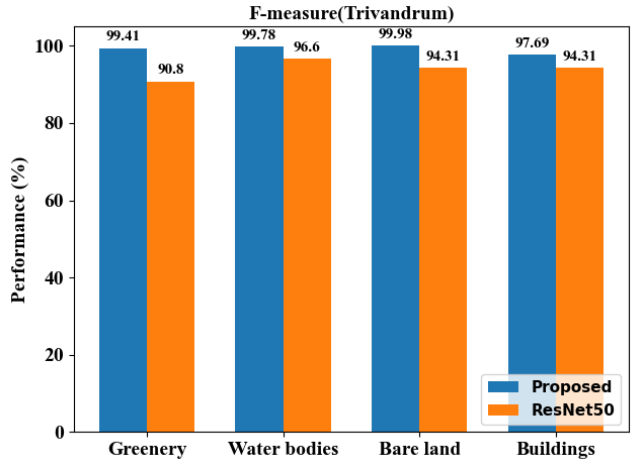

Fig. 12 Performance comparison on F-measure (Trivandrum)

Figure 12 illustrates the performance comparison on F-measure for the Trivandrum image. The performance of the presented approach in terms of accuracy is improved than the existing framework for each class. Similarly, the recall comparison of the Kovalam image is illustrated in figure 13.

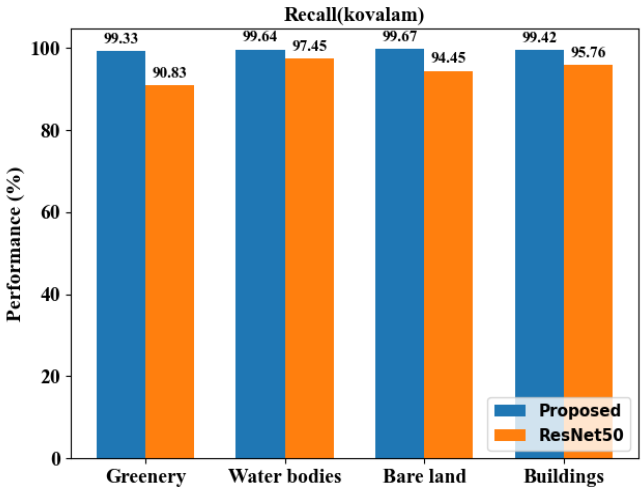

Fig. 13 Comparison of Recall (Kovalam)

Figure 13 illustrates the recall performance of the Kovalam image with four classes. Here, the performance of the presented approach is compared with the current ResNet50 framework. The attained recall value of the greenery class is 99.33%, water body class is 99.64%, bare land is 99.67%, and the building class is 99.42%, respectively. The attained recall performance is improved than the existing ResNet50 framework. Moreover, the performance comparison on recall (Kollam) is depicted in figure 14.

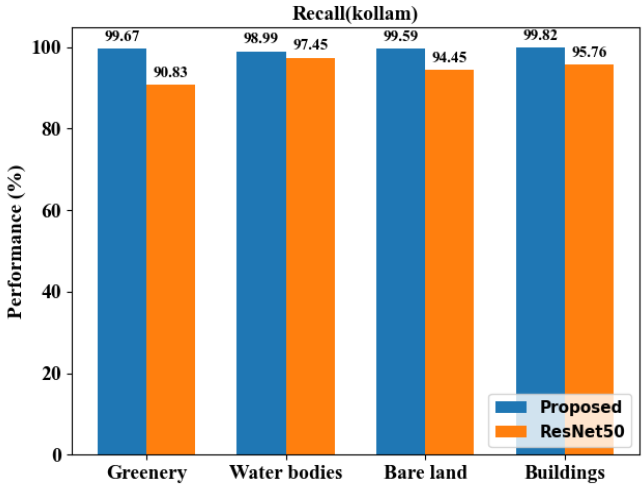

Fig. 14 Comparison of Recall (Kollam)

Figure 14 illustrates the recall comparison of the Kollam image. The attained recall of greenery class is 99.67%, water body class is 98.99%, bare land class is 99.59%, and the building class is 99.82%. It proved that the attained recall value of the presented methodology is much higher than the existing ResNet50 framework. Furthermore, the comparison of recall for Trivandrum is depicted in figure 15.

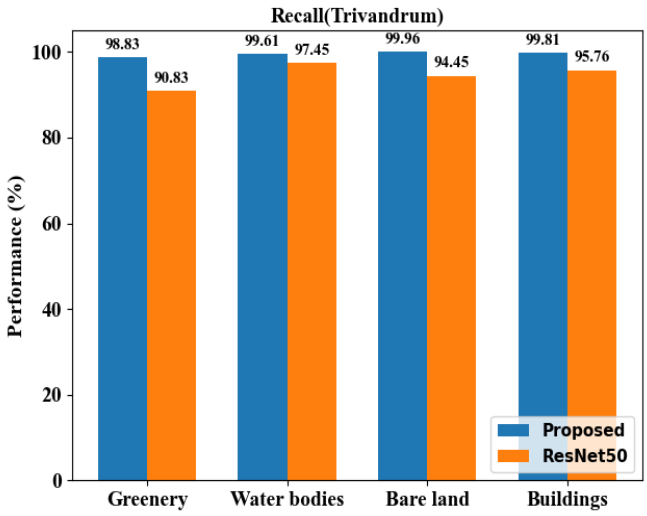

Fig. 15 Comparison of Recall (Trivandrum)

Figure 15 displays the recall comparison of the Trivandrum RS image. Here, the presented approach performance is compared with the existing ResNet50 framework. The attained recall of the greenery class is 98.93%, the water body class is 99.61%, the bare land class is 99.96%, and the building class is 99.59%. Figure 15 displays the attained recall value of the presented methodology, which is much higher than the existing ResNet50 framework. Similarly, the precision comparison with the Trivandrum image is depicted in figure 16.

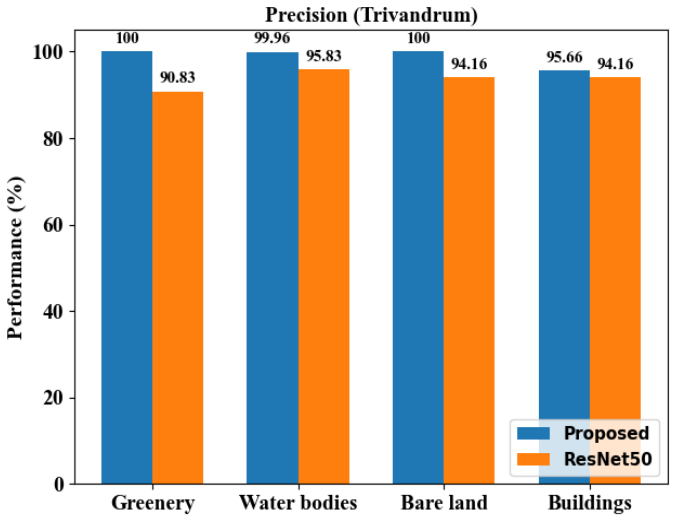

Fig. 16 Comparison of precision (Trivandrum)

In figure 16, the precision comparison for different scene types is demonstrated. Here, the presented approach performance is compared with the traditional ResNet50 framework. The obtained F-measure of the greenery region is 100%, the water body region is 99.96%, the bare land region is 100%, and the building is 95.66%. Figure 16 displays the precision value of the presented approach is much higher than the existing ResNet50 framework. Furthermore, the comparison of precision for Kovalam is depicted in figure 17.

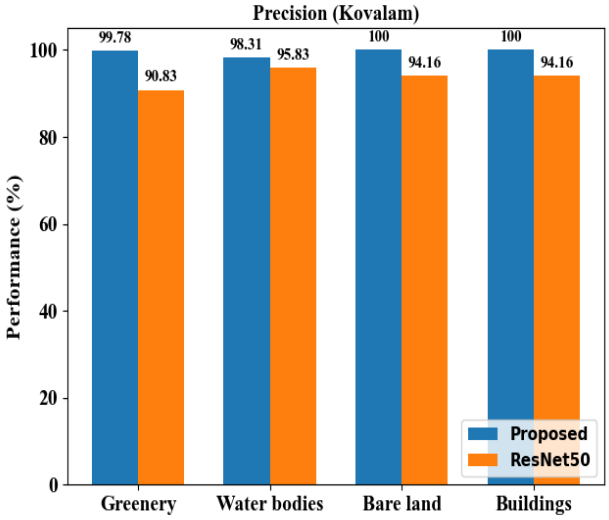

Fig. 17 Comparison of precision (Kovalam)

Figure 17 illustrates the precision comparison of the Kovalam image. The obtained F-measure of greenery class is 99.78%, water body class is 98.31%, bare land class is 100%, and the building class is 100%. It proved that the attained F-measure value of the presented methodology is much higher than the existing ResNet50 framework. Similarly, the precision comparison with the Kollam image is depicted in figure 18.

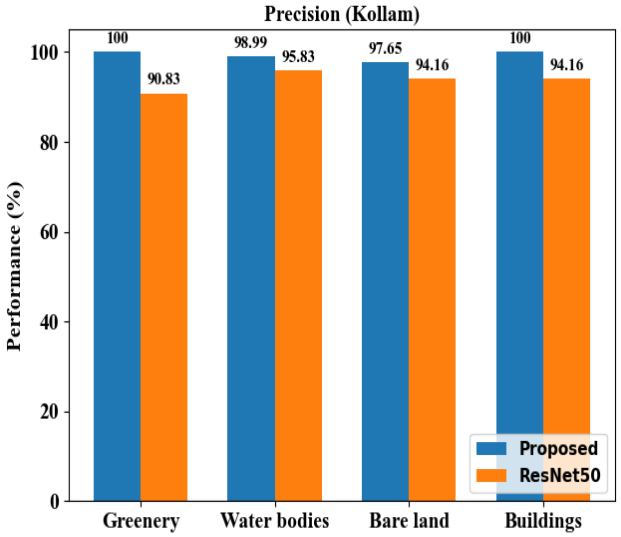

Fig. 18 Comparison of precision (Kollam)

In figure 18, the precision comparison for different scene types is demonstrated. The obtained F-measure of the greenery region is 100%, the water body region is 99.96%, the bare land region is 100%, and the building is 95.66%. It demonstrates that the precision value of the presented approach is much higher than the existing ResNet50 framework.

5. Conclusion

This paper accurately detected a scene in remote sensing images using optimal feature learning and classification. The detected scenes are greenery, water body, land, and building regions in the Kollam, Kovalam, and Trivandrum region images. The result achieved by the proposed approach during experimentation is compared with various existing approaches. The performance metrics and values achieved by the proposed method are accuracy (99.75%), precision (99.16%), recall (99.523%), F-measure (99.34%), and kappa measure (0.99) and processing time (10.67 seconds). The proposed deep learning framework outperforms the different compared techniques. The effectiveness of the proposed approach is validated with the private remote sensing images collected from Kollam, Kovalam, and Trivandrum regions.

Conflicts of Interest

We (Author and Co-author) declared that we have no conflict of interest/Competing interest.

Funding Statement

We (The author and Co-author) have not received any funding for doing this work.

References

[1] Lv, Y.; Zhang, X.; Xiong, W.; Cui, Y.; Cai, M. An End-to-End Local-Global-Fusion Feature Extraction Network for Remote Sensing Image Scene Classification. Remote Sens. 11 (2019) 3006. Https://Doi.Org/10.3390/Rs11243006

[2] Bechtel, B.; Alexander, P.J.; Böhner, J.; Ching, J.; Conrad, O.; Feddema, J.; Mills, G.; See, L.; Stewart, I. Mapping Local Climate Zones for a Worldwide Database of the Form and Function of Cities. ISPRS Int. Jgeo-Inf, 4 (2015) 199-219. Https://Doi.Org/10.3390/Ijgi4010199

[3] Lu Cui, Yonghua Zhao, Jianchao Liu, Huanyuan Wang, Ling Han, Juan Li, Zenghui Sun, Vegetation Coverage Prediction for the Qinling Mountains Using the CA–Markov Model, ISPRS International Journal of Geo-Information, 10.3390/Ijgi10100679. 10 (679) (2021).

[4] Zou, F., Xiao, W., Ji, W., He, K., Yang, Z., Song, J., .& Li, K. Arbitrary-Oriented Object Detection Via Dense Feature Fusion and Attention Model for Remote Sensing Super-Resolution Image. Neural Computing and Applications, 32(18) (2020) 14549-14562.

[5] Long, Y., Xia, G. S., Li, S., Yang, W., Yang, M. Y., Zhu, X. X., & Li, D., Dirs: on Creating Benchmark Datasets for Remote Sensing Image Interpretation, (2020). Arxiv Preprint Arxiv:2006.12485.

[6] Georgiosovakoglou, Thomas K. Alexandridis, Jan G.P.W Clevers, Ioannis Z. Gitas., Downscaling of MODIS Leaf Area Index Using Landsat Vegetation Index. Geocarto International 0:0, (2020) 1-24.

[7] Lanfa Liu, Ana-Maria Olteanu-Raimond, Laurence Jolivet, Arnaud-Le Bris, Linda See., A Data Fusion- Based Framework to Integrate Multi-Source VGI in an Authoritative Land Use Database. International Journal of Digital Earth, 14(4) (2021) 480-509.

[8] El-Hamid, H.T.A., Caiyong, W. &Yongting, Z. Geospatial Analysis of Land Use Driving Force in Coal Mining Area: Case Study in Ningdong, China. Geojournal, 86 (2021) 605–620. Https://Doi.Org/10.1007/S10708-019-10078-2

[9] Liu, J., Kuang, W., Zhang, Z. Et Al. Spatiotemporal Characteristics, Patterns, and Causes of Land-Use Changes in China Since the Late 1980s. J. Geogr. Sci, 24 (2014) 195–210. Https://Doi.Org/10.1007/S11442-014-1082-6

[10] Bey, A.; Sánchez-Pausdíaz, A.; Maniatis, D.; Marchi, G.; Mollicone, D.; Ricci, S.; Bastin, J.-F.; Moore, R.; Federici, S.; Rezende, M.; Patriarca, C.; Turia, R.; Gamoga, G.; Abe, H.; Kaidong, E.; Miceli, G. Collect Earth: Land Use and Land Cover Assessment Through Augmented Visual Interpretation. Remote Sens, 8 (2016) 807. Https://Doi.Org/10.3390/Rs8100807

[11] Thoman, P., Dichev, K., Heller, T. Et Al. A Taxonomy of Task-Based Parallel Programming Technologies for High-Performance Computing. J Supercomput, 74 (2018) 1422–1434. Https://Doi.Org/10.1007/S11227-018-2238-4

[12] Hossam, M.A., Ebied, H.M., Abdel-Aziz, M.H. Et Al. Accelerated Hyperspectral Image Recursive Hierarchical Segmentation Using Gpus, Multicore Cpus, and Hybrid CPU/GPU Cluster. J Real-Time Image Proc , 14 (2018) 413–432. Https://Doi.Org/10.1007/S11554-014-0464-4

[13] Chen, Y., Chen, Q., & Jing, C., Multi-Resolution Segmentation Parameters Optimization and Evaluation for VHR Remote Sensing Image Based on Mean NSQI and Discrepancy Measure. Journal of Spatial Science, 66(2) (2021)253-278.

[14] Chen, C., Gong, W., Chen, Y., & Li, W., Object Detection in Remote Sensing Images Based on A Scene-Contextual Feature Pyramid Network. Remote Sensing, 11(3) (2019) 339.

[15] Wang, L., Yan, J., Mu, L., & Huang, L., Knowledge Discovery From Remote Sensing Images: A Review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(5) (2020) E1371.

[16] Zahra Assarkhaniki, Soheilsabri, Abbas Rajabifard., Using Open Data to Detect the Structure and Pattern of Informal Settlements: an Outset to Support Inclusive Sdgs' Achievement. Big Earth Data, 5(4) (2021) 497-526.

[17] Shah, K., Patel, H., Sanghvi, D. Et Al. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. Augment Hum Res, 5 (2020) 12. Https://Doi.Org/10.1007/S41133-020- 00032-0

[18] Hu, Y.; Zhang, Q.; Zhang, Y.; Yan, H. A Deep Convolution Neural Network Method for Land Cover Mapping: A Case Study of Qinhuangdao, China. Remote Sens, 10 (2018) 2053. Https://Doi.Org/10.3390/Rs10122053

[19] Zahraassarkhaniki, Soheilsabri, Abbasrajabifard., Using Open Data to Detect the Structure and Pattern of Informal Settlements: an Outset to Support Inclusive Sdgs' Achievement. Big Earth Data, 5(4) (2021) 497-526.

[20] Sebastiãorogério Da Silva Neto, Thomástabosa Oliveira, Igor Vitorteixeira, Samuel Benjamin Aguiar De Oliveira, Vanderson Souza Sampaio, Theo Lynn, Patricia Takako Endo and Rhoel Ramos Dinglasan Machine Learning and Deep Learning Techniques to Support Clinical Diagnosis of Arboviral Diseases: A Systematic Review, Journal: PLOS Neglected Tropical Diseases, 16(1) (2022) E0010061. DOI: 10.1371/Journal.Pntd.0010061.

[21] Avenash, R., & Viswanath, P., Semantic Segmentation of Satellite Images Using A Modified CNN with Hard-Swish Activation Function. in VISIGRAPP (4: VISAPP), (2019) 413-420.

[22] Neupane, B., Horanont, T., & Aryal, J., Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis. Remote Sensing, 13(4) (2021) 808.

[23] Khamparia, A., Gupta, D., De Albuquerque, V.H.C., Et Al. Internet of Health Things-Driven Deep Learning System for Detection and Classification of Cervical Cells Using Transfer Learning. J Supercomput, 76 (2020) 8590–8608. Https://Doi.Org/10.1007/S11227-020-03159-4

[24] Yu, X., Tang, L., Wu, X. Et Al. Nondestructive Freshness Discriminating of Shrimp Using Visible/Near-Infrared Hyperspectral Imaging Technique and Deep Learning Algorithm. Food Anal. Methods, 11 (2018) 768–780. Https://Doi.Org/10.1007/S12161-017-1050-8

[25] Chen, Y., Ming, D. &Lv, X. Superpixel Based Land Cover Classification of VHR Satellite Image Combining Multi-Scale CNN and Scale Parameter Estimation. Earth Sci Inform, 12 (2019) 341–363.Https://Doi.Org/10.1007/S12145-019-00383-2.

[26] Shabbir, Amsa, Nouman Ali, Jameel Ahmed, Bushra Zafar, Aqsa Rasheed, Muhammad Sajid, Afzal Ahmed, and Saadathanif Dar, Satellite and Scene Image Classification Based on Transfer Learning and Fine-Tuning of Resnet50, Mathematical Problems in Engineering, 2021 (2021).

[27] Robinson, Y. Harold, Et Al. Tree-Based Convolutional Neural Networks for Object Classification in Segmented Satellite Images, the International Journal of High-Performance Computing Applications, (2020) 1094342020945026.

[28] Zhang, W.; Tang, P.; Zhao, L. Remote Sensing Image Scene Classification Using CNN-Capsnet. Remote Sens, 11 (2019) 494. Https://Doi.Org/10.3390/Rs11050494

[29] Xiaoweixu, Yinrong Chen, Junfeng Zhang, Yu Chen, Prathik Anandhan&Adhiyaman Manickam., A Novel Approach for Scene Classification From Remote Sensing Images Using Deep Learning Methods, European Journal of Remote Sensing, 54(2) (2021) 383-395. DOI: 10.1080/22797254.2020.1790995

[30] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13 (2020) 3735–3756.

[31] Anwer, R.M., Khan, F.S. &Laaksonen, J. Compact Deep Color Features for Remote Sensing Scene Classification. Neural Process Lett, 53 (2021) 1523–1544. Https://Doi.Org/10.1007/S11063-021-10463-4

[32] Yin, Liancheng, Peiyi Yang, Keming Mao, and Qian Liu, Remote Sensing Image Scene Classification Based on Fusion Method, Journal of Sensors, 2021 (2021).

[33] Yuegao, Jun Shi, Jun Li &Ruoyu Wang ., Remote Sensing Scene Classification Based on High-Order Graph Convolutional Network, European Journal of Remote Sensing, 54 (2021) 141-155. DOI: 10.1080/22797254.2020.1868273

[34] Shabbir, A., Ali, N., Ahmed, J., Zafar, B., Rasheed, A., Sajid, M., & Dar, S. H., Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning Ofresnet 50. Mathematical Problems in Engineering, (2021). Https://Doi.Org/10.1155/2021/5843816

[35] Robinson, Y. H., Vimal, S., Khari, M., Hernández, F. C. L., & Crespo, R. G., Tree-Based Convolutional Neural Networks for Object Classification in Segmented Satellite Images. The International Journal of High Performancecomputingapplications, (2020).

[36] Vinothini, Vengateshkumar., Design of KNN Classified Segmentation of Hyper Spectral Imaging, SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), (2019) 67-74.

[37] Keerti Maithil , Tasneem Bano Rehman., Urban Remote Sensing Image Segmentation Using Dense U-Net+. SSRG International Journal of Computer Science and Engineering, 9(3) (2022) 21-28.