



# A Novel Pyramidal CNN Deep Structure for Multiple Objects Detection in Remote Sensing Images

Khaled Mohammed Elgamily<sup>1</sup> · M. A. Mohamed<sup>2</sup> · Ahmed Mohamed Abou-Taleb<sup>2</sup> · Mohamed Maher Ata<sup>3,4</sup> 

Received: 13 March 2023 / Accepted: 20 November 2023 / Published online: 22 December 2023  
© Indian Society of Remote Sensing 2023

## Abstract

This article suggests a novel convolutional neural network (CNN) layering structure based on the pyramidal-shaped CNN model in the state of the art of remote sensing images. The suggested system outperforms the traditional CNN pre-trained models. Consequently, a detailed analysis of several CNN models has indeed been utilized. Furthermore, a comprehensive comparison has been acquired between the proposed Pyramidal Net model and nine different well-known pre-trained models to assess the efficacy of the developed framework. Ten distinct classes have been trained, tested, and validated from two different standardized datasets; NWPU-RESISC45 (Northwestern Polytechnical University Remote Sensing Image Scene Classification) and UC (University of California) Merced Land Use datasets. The utilized system performance has been evaluated based on several metrics: accuracy, recall, precision, IOU, and F1-score. Experimental findings demonstrate that the proposed Pyramidal Net CNN model has achieved an accuracy of 97.1%, recall: 0.96, precision: 0.96, IOU: 0.928, and F1-score: 0.96. The proposed model in comparison with other pre-trained CNN architectures has improved the classification accuracy by a percentage up to 30% taking into consideration a superior training time of 840 s for 5950 images with 10 different classes.

**Keywords** Deep learning · Satellite imagery · Object detection · Pre-trained CNN

## Introduction

Object detection in remote sensing images is an essential image-processing step for a range of applications such as industrial applications, agriculture, and military application (d'Acromont et al., 2019). The identification of land uses and objects using remote sensing images acquired by satellites is

essential for regulating and tracking life form activities (Lin & Wu, 2019). Recently, due to the massive quantity of data provided by remote sensing photographs, deep learning has been crucial in remote sensing image applications such as object segmentation, target identification, object recognition, image augmentation, and image preprocessing (Liu et al., 2022). Various deep learning convolutional neural network (CNN) models have been developed and utilized in the realm of satellite imagery; these different models' architecture can extract various deep characteristics and produce varying experimental outcomes (Ran et al., 2019). The employment of various feature extraction methods, which are capable of deep computational methods from the dataset, is largely responsible for CNN's current learning capabilities, which makes the deployment of CNN algorithms in the processes of remote sensing object detection a great performance enhancement for the system accuracy (Khan et al., 2020). To get the maximum benefit from the remote sensing images based on CNN models when utilized in the applications the article mentioned above, the applied CNN models should be developed to have the highest accuracy possible and be able to extract the very tiny object's features accurately. This

---

✉ Mohamed Maher Ata  
mmaher844@yahoo.com

<sup>1</sup> Department of Electronics and Communications Engineering, Mansoura Higher Institute for Engineering and Technology, Mansoura 35516, Egypt

<sup>2</sup> Department of Electronics and Communications Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

<sup>3</sup> School of Computational Sciences and Artificial Intelligence (CSAI), Zewail City of Science and Technology, October Gardens, 6th of October City, Giza 12578, Egypt

<sup>4</sup> Department of Communications and Electronics Engineering, MISR Higher Institute for Engineering and Technology, Mansoura 35516, Egypt

article developed a new CNN pyramidal model that focuses on improving the process of object detection from remote sensing images while taking into consideration the training time to be very small in comparison with the other existing CNN models. The suggested CNN architecture was compared with nine different pre-trained convolutional models and outperformed them. The suggested structure includes three main transitions, which are illustrated in Fig. 1: (i) gathering and preparation of datasets, (ii) development of CNN architecture, (iii) the proposed CNN model is evaluated, examined, and contrasted with several pre-trained models. This article's major contribution is:

1. Proposing a robust CNN model that is developed and employed in the state of the art of an optimized layering structure and fine-tuned hyper-parameters.
2. Utilizing the impact of traditional pre-trained CNN models on the classification process of objects in remote sensing images.
3. Comparing the performance of the suggested CNN algorithms and nine well-known pre-trained models based on standardized datasets.

The remainder of this paper is structured as follows: Section "Literature Review" summarizes the literature review. Section "Approach Preprocessing" goes through the proposed approaches in depth. Section "The Proposed CNN Model Approach" provides the suggested method's experimental outcomes. The section "Experimental Results and Discussion" brings the research to a conclusion.

## Literature Review

### Classification

Kumar et al. (2021) examined the outcome of pre-training 16 different convolutional neural network algorithms on the ImageNet database and tuned these models for the

challenge of recognizing numerous items in very high-resolution pictures. They indicated that using pre-trained algorithms would reduce the demand for vast volumes of very-high-resolution pictures.

Li et al. (2019) used four DNNs to construct comparable classification approaches in metropolitan built-up environments (CNN, SMDTR-CNN, CapsNet, and SMDTR-CapsNet). In terms of various metrics, the offered methodologies' accomplishments have been confirmed.

Liang et al. (2020) utilized a two-stream satellite imagery picture categorization system. Furthermore, the merging of CNN and GCN helps the developed system to simultaneously learn object-based spatial aspects and global-based visual characteristics. The framework acquires the appearance properties of the entire picture and the spatial dependence between items at the same time, thereby reducing visual confusion and improving feature discrimination.

Xu et al. (2021) proposed an improved classification approach for land categorization using remote sensing pictures that combine recurrent neural network (RNN) and random forest (RF). Object and pixel categorization are used for classification.

Cheng et al. (2020) described the primary issues of satellite images categorization and conducted and presented: (i) auto-encoder-based satellite images classification; (ii) CNN-based satellite image object detection methods; and (iii) methods for detecting satellite photographs using generative adversarial networks.

Dong et al. (2020) developed an approach for classifying very-high-resolution satellite pictures depending on the merging of a random forest (RF) classifier with the CNN. The fusion with the RF had a great enhancement on the task of the relevant variables selection.

Ma et al. (2021) presented the SceneNet approach for image categorization network architecture discovery based on the neural evolution of multi-objective. The system searching and architecture coding in SceneNet are accomplished by the application of an evolutionary

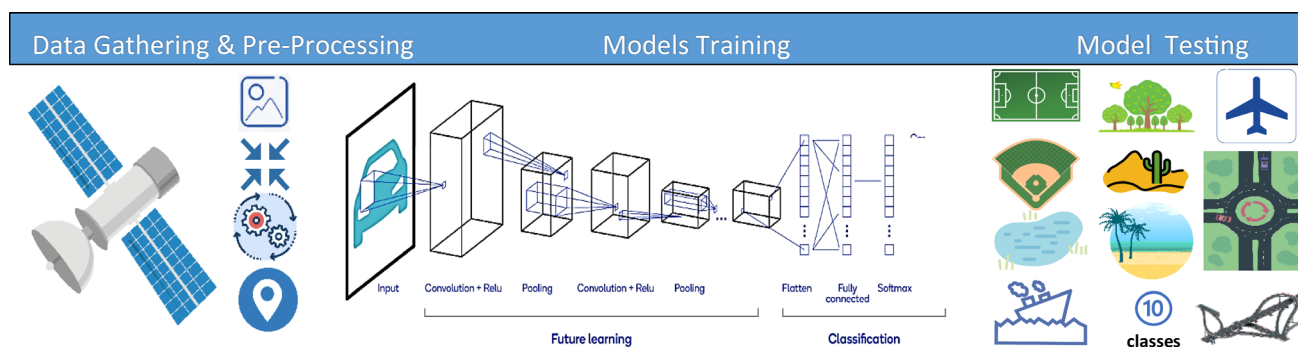


Fig. 1 Main stages of the CNN system

technique, which may construct an improvement in the hierarchical extraction of satellite image information.

Priya and Vani (2019) proposed a convolutional neural algorithm for fire detection. The algorithm is based on Inception-v3 with the transferred learning-based system that has been trained using satellite pictures for the process of classifying images into fire and non-fire images.

Unnikrishnan et al. (2019) proposed innovative deep learning designs for three different networks (AlexNet, VGG, and ConvNet) developed by hyper-tuning the model and using 2 bands of data as the input. The redesigned models using the 2-band input and a decreased layers number are trained and tested in order to categorize photographs into distinct groups.

Rohith and Kumar (2022) built a 13-layer CNN architecture for the process of classifying raw remote sensing images of the National Remote Sensing Center (NRSC) dataset.

Zhao et al. (2019) investigated the feature representation capacity of multiple classifiers from the perspective of categorization of satellite imagery. Furthermore, 4 pre-trained CNN algorithms and 3 popular databases are chosen, compared, and summarized.

Özyurt (2020) proposed feature extractors, VGG19, VGG16, Alexnet, ResNet, SqueezeNet, and GoogleNet pre-trained architectures that were employed. They acquire features from the architecture's final fully connected (FC) layers, and to produce suitable features, the article used the ReliefF approach for the process of selecting features. The convolutional neural features are then sent into the support vector machine (SVM) classification algorithm rather than the FC layers of CNN to measure the performance.

## Detection

Zalpour et al. (2020) proposed an oil tank identification framework based on oil depot detection by employing deep characteristics. First of all, oil stores are retrieved using a faster R-CNN. Second, for suitable target selection, a quick circle detection algorithm is used. For feature extraction, they coupled CNN and HOG. Finally, the SVM classifier is utilized for the process of classifying the images.

Chen et al. (2022) used the transfer learning approach in order to overcome the overfitting issue. For the purpose of finding airplanes in remote sensing photographs, the Domain Adaptation faster R-CNN (DA faster R-CNN) algorithm is suggested. The DA faster R-CNN detection technique is applied to the DOTA dataset for the detection of aircraft for the process of addressing the detection challenge resulting from the poor quality of remote sensing photos.

Darehnaei et al. (2022) suggested swarm intelligence ensemble deep transfer learning (SI-EDTL), for the

purpose of detecting various vehicles. Faster regional-based convolutional neural networks (faster R-CNN) are employed in this article. The region proposal network (RPN) is applied for extracting various regional proposals, and CNN is then employed to choose the most evocative characteristics of that region to identify objects. They utilized three different faster R-CNN as learning algorithms that trained on the ImageNet dataset, along with five transfer classification models in order to categorize the region of interest into four vehicle classes taken from the UAV dataset.

Feng et al. (2019) utilized faster R-CNN to locate vehicles in satellite pictures. They investigated the effects of the size of objects and the pooling technique on the regional proposal, and then, in order to improve the detection accuracy of multi-scale objectives, a new strategy for region recommendation was developed. Several tests are used to show the efficiency of the developed multi-scale object classification algorithm for remote sensing images.

Napiorkowska et al. (2018) utilized the FCN-VGG network to recognize three distinct items or characteristics in satellite imagery which are roadways, palm plants, and vehicles taken from Deimos-2 and Worldview-3 datasets. The outcomes indicate that the suggested strategy is successful at locating objects with various colors and forms, which conventional satellite imagery approaches such as RF or SVM cannot do.

Karnick et al. (2022) applied a Multi-Scale Swift Detection System, which is a fully convolutional network architecture, on the COWC dataset to locate cars. This detection technique employs a modified version of YOLO known as the YOLT architecture, which scans test photographs of arbitrary size using bounding boxes to find the vehicles.

Karim et al. (2019) proposed a training approach that relies on compressed and down-scaled photographs to assess the influence of automobile compression techniques and down-scaling on prediction performance.

Sharma et al. (2021) built YOLOrs, which is a novel CNN proposed for object recognition in multimodal remote sensing pictures. The utilized approach used for vehicle detection was compared with various modern techniques to prove its strengths.

Zhang et al. (2022) proposed an MFRC detection technique based on faster R-CNN for the detection of airplanes. Three steps were used in developing the suggested framework: to begin, K-means is used to combine the airplane enclosing areas while also enhancing the region detection in RPN. Second, the pooling layers of the VGG16 network are reduced from four to two in order to achieve the characteristic of small-scale airplanes. Finally, Soft-NMS is employed to improve the airplane's frame.

Zhu et al. (2020) proposed an innovative satellite images object recognition technique that employs a fusion-based feature reinforcement component (FB-FRC) to enhance object feature discrimination. Two fusion techniques are proposed in detail: (i) a hard fusing approach and (ii) a soft fusing technique.

Cui et al. (2021) collected global and local characteristics simultaneously, to introduce a detection mechanism that uses the dual-channel deep learning (DCDL) method. In order to construct local mining and residual calculations on the image in the first stage, they used a multiscale convolution residual network. Secondly, the local concentration approach is used to limit the information by assigning weight factors to local attributes. Lastly, 2-layer convolution is applied to achieve deep feature mining in order to detect three separated classes from the NWPU-RESISC45 dataset.

## Segmentation

Diakogiannis et al. (2020) innovated a powerful deep learning modeling approach for the segmentation of high-definition satellite photographs introduced ResUNet-a; their deep learning architecture is built on the encoder/decoder concept, with typical convolutions substituted with ResNet modules.

Pang and Gao (2022) introduced the MAGC-Net neural network algorithm for pixel-level classification of ocean satellite imagery pictures, which is built on a multi-head attention technique supervised by Conv-LSTM. The outcome demonstrates that the suggested three Conv-LSTM layers that analyze deep features in this network utilize the multi-head attention technique to fully exploit the number of hosts, substantially decreasing features and fusion of features enhancing.

Table 1 summarizes the most recent research articles that discuss the problem of recognizing multiple objects in remote sensing images.

The literature review represents that remote sensing applications can be categorized into different categories based on multiple ways:

- (1) Application (classification, detection, and segmentation).
- (2) Acquired image quality (high-resolution images, low-resolution images).
- (3) Objects (multiple object detection, single object detection).
- (4) Image capturing distance from the ground (satellite imagery, drone imagery)

In this article, we are utilizing low-resolution multiple-object satellite images for classification purposes.

## Approach Preprocessing

### Image Acquisition

Images are collected from two datasets, which are summarized in Table 2: (a) Northwestern Polytechnical University (NWPU) published the NWPU-RESISC45 dataset, which is released for Remote Sensing Image Scene Classification (RESISC). This data collection comprises 31,500 photographs divided into 45 environment classes, every with 700 images (Cheng et al., 2017). Dataset samples are illustrated in Fig. 2. (b) The UC Merced Land Use dataset contains 21 land-use types represented by aerial photographs ( $256 \times 256$  dimensions in RGB). The classes consist of 100 photographs each (Yang & Newsam, 2010). Figure 3 illustrates the samples from the dataset.

### Image Preprocessing

Image preprocessing is a large field of study that contains many fields including image resizing and image augmentation, which plays a huge role in deep learning applications and especially in object detection tasks (Kodali & Dhanekula, 2021; Marastoni et al., 2021). The proposed preprocessing steps are summarized in Fig. 4.

### Image Resizing

Image resizing is an important image preprocessing step, in which dimensions of input images are resized in order to be more suitable for the CNN architecture that the system is dealing with. Furthermore, it can enhance the overall accuracy and processing time (Kodali & Dhanekula, 2021; Pathak & Raju, 2022). This article has applied image resizing for all the input images depending on the pre-defined input size of the involved deep learning pre-trained model based on the Python resizing function, which has been used in recent articles (Vyas et al., 2022).

### Image Augmentation

Image augmentation is an approach applied to expand the volume of information by adding slightly changed replicas of either current data or newly produced synthetic data from the already existing data. It functions as a regularizer and assists in preventing overfitting while developing a deep learning algorithm (Chlap et al., 2021; Khalifa et al., 2022). In this article, data augmentation has been involved by adding an

**Table 1** A complete literature review of recent research articles in state of the art of remote sensing image classification

References	Application	Technique	Strength	Challenges
Kumar et al. (2021)	Classification	16 Pre-trained CNN models	Parameters tuning	No new architecture
Li et al. (2019)	Classification	SMDTR-CNN	New layering structure (Capsule)	Large training time
Liang et al. (2020)	Classification	CNN-GCN	Features fusion	Complexity
Xu et al. (2021)	Classification	RNN-RF	Classification technique	Small dataset size
Cheng et al. (2020)	Classification	Published classification techniques	Comprehensive comparison of classification methods	No new architecture
Dong et al. (2020)	Classification	CNN-RF	Fusion of CNN and RF for feature extraction and classification	A large number of epochs
Ma et al. (2021)	Classification	SceneNet	Multi-objective optimization	Very Large training time
Priya and Vani (2019)	Classification	Inception-v3	High accuracy	Single class and small dataset
Unnikrishnan et al. (2019)	Classification	AlexNet, ConvNet, VGG	Hyper-tuning the pre-trained networks	Lack of performance indicators
Rohith and Kumar (2022)	Classification	Thirteen-layer deep CNN model	Layering structure	Low accuracy
Zhao et al. (2019)	Classification	Pre-trained CNN models	Comparison of feature representation methods	No new architecture
Özyurt (2020)	Feature selection	CNN—Relief—SVM	High accuracy	Complexity and time
Zalpour et al. (2020)	Detection	Improved faster R-CNN with SVM	Small processing time	Single class
Chen et al. (2022)	Detection	DA faster R-CNN	Fast iteration time and low brightness domain adaptation	Single object detection
Darehnaei et al. (2022)	Detection	SI-EDTL	Combined 3 pre-trained models with 5 classifiers	Complexity and processing time
Feng et al. (2019)	Detection	Modified RPN	Layering structure	Many iterations and lack of indicators
Napiorkowska et al. (2018)	Detection	VGG-8	High accuracy	Single object detection
Karnick et al., (2022)	Detection	YOLT	Detect objects at several scales	Complexity
Karim et al. (2019)	Detection	RCNN	Impact of image rescaling on detection	No new architecture
Sharma et al. (2021)	Detection	YOLOrs	Detect objects at several scales	Complexity and processing time
Zhang et al. (2022)	Detection	Faster RNN	Boundary box optimization	Single object detection
Zhu et al. (2020)	Detection	FB-FRC	Features fusion strategy	Complexity
Cui et al. (2021)	Recognition	DCDL	Mining global and local features simultaneously	Recognition is for a single class
Diakogiannis et al. (2020)	Semantic segmentation	ResUNet-a	Multiple object segmentation	Only used with high-resolution images
Pang and Gao (2022)	Segmentation	MAGCNet—ConvLSTM	Features extraction technique	Single object segmentation

**Table 2** The dataset details

Dataset	No. of classes	No. of images per class	Image size
NWPU-RESISC45	45	700	256×256×3
UC Merced	21	100	256×256×3

extra version of data to increase both the validation and training input data amount by adding some changes to the data which was rotation with an angle between zero-degree and 180-degree, vertical flip, horizontal flip and zoom in with a 0.1 percent of the overall input image dimensions.





Fig. 2 NWPU-RESISC45 dataset samples

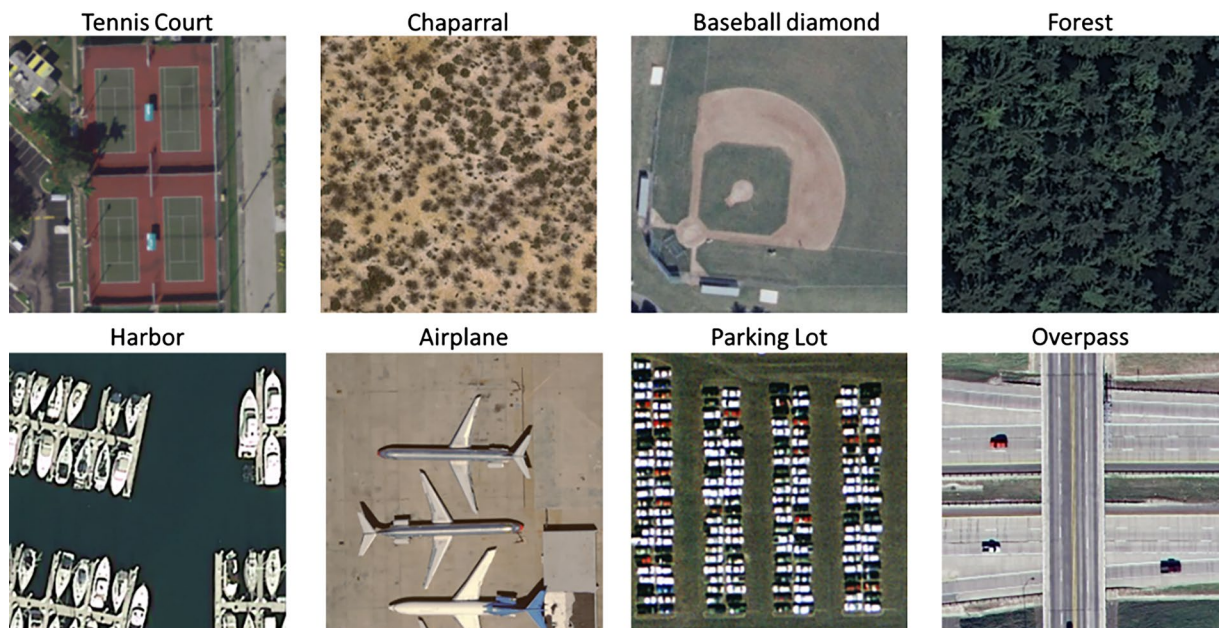
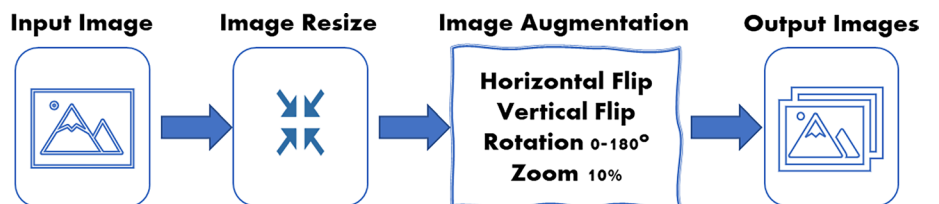


Fig. 3 UC Merced dataset samples

Fig. 4 Preprocessing flow chart



## Model Optimization

Optimizers are techniques that adjust the deep learning algorithm's properties including learning rate (Lr) and weights to increase the accuracy of the system. Optimizers are critical in decreasing the loss caused by the training phase (Manickam et al., 2021). This article proposed the Adam optimization technique starting with random weights and with a 0.00025 starting learning rate. Furthermore, the learning rate value was reduced by a factor of 0.25 using a learning rate reduction method, which is done if accuracy stays constant for multiple consecutive epochs with a max. of four overall reductions with a min. learning rate of  $1 \times 10^{-6}$  (Kingma & Ba, 1412).

## Pre-trained CNN Algorithms

CNN has already made incredible progress, mostly in image processing techniques, and has rekindled academics' interest in ANNs. Numerous research papers have been done in order to improve CNN's ability to complete tasks. CNN advancement may be divided into several categories, such as optimizations, regularization, deep learning architectures, and design improvements (Lei et al., 2020). This area of the site tracks advancements among the most prevalent convolutional networks. This article applied multiple pre-trained deep learning techniques in order to compare their accuracy with the model innovated within this article including:

- The Vgg16 algorithm is a deep learning network composed of thirteen convolutional combined with three fully linked layers. It is divided into 41 pieces, including the SoftMax layer, the Max pool, the fully connected layer, the Relu layer, and the Dropout layer (Ye et al., 2021). The VGG16 input image default dimensions are (224,224,3).
- The Vgg19 algorithm is a deep learning network composed of sixteen convolutional combined with three fully linked layers. It is divided into 41 parts, including the Max pool, the fully linked layer, the Relu layer, the Dropout layer, and the SoftMax layer (Li et al., 2020). The VGG19 input image default dimensions are (224,224,3).
- AlexNet was created using deep learning methods. This design reduced the number of failures in computer image classification. Five convolutional layers, three pooling layers, and three fully connected layers are the main layers of AlexNet (Dhillon & Verma, 2020). AlexNet default picture size of the input image is (224,224,3). The equation summarizes the model represented in Eq. (1).

$$A(M) = I(M) + X(M) \quad (1)$$

where the output target  $I(M)$  and the summed companion targets  $X(M)$  are each independently calculated in Eqs. (2) and (3).

$$I(M) \equiv \|\|m(o)\|\|2 + L(M, m(o)) \quad (2)$$

$$X(M) \equiv \Sigma N - 1n \\ = 1[\|\|m(N)\|\|2 + l(M, m(n)) - r \quad (3)$$

- MobileNet is composed of depth-separable convolution layers. The depth-wise convolution and point-wise convolution make up every depth-wise separable convolutional layer. If point-wise convolutions and depth-wise convolutions are computed individually, a MobileNet has 28 layers. A basic MobileNet contains 4.2 million parameters (Hou et al., 2020). MobileNet default picture size of the input image is (224,224,3). The depth-wise convolution is represented in Eq. (4).

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,j-1,m} \quad (4)$$

where  $K$  is the depth-wise convolution kernel,  $F$  is the feature map input, and  $G$  is the created feature map.

- ResNet is a more complex design with 152 layers than any other known architecture. It is made up of several residual blocks. The ResNet default input dimensions are (224,224,3). Equations (5–7) represent the ResNet model (Sarwinda et al., 2021).

$$S_{1+n}^j = A_c(S_{1 \rightarrow n}^j, j_{1 \rightarrow n}) S_I^j \quad n \geq I \quad (5)$$

$$S_{1+n}^j = A_a(S_{1+n}^j) \quad (6)$$

$$A_c(S_{1 \rightarrow n}^j, j_{1 \rightarrow n}) = S_{1+n}^j - S_I^j \quad (7)$$

where  $A_c(S_{1 \rightarrow n}^j, j_{1 \rightarrow n})$  is a converted signal, and  $S_I^j$  is the  $I$ -th layer input.  $A_c(S_{1 \rightarrow n}^j, j_{1 \rightarrow n})$  and  $S_{1+n}^j$  are the input of the next layer after the activation function  $A_a$  is applied.

- DenseNet Traditional  $n$ -layer deep network function  $n$  connections, one between every level and the layer after it. DenseNet contains  $n(n+1)/2$  interconnection since each layer links to all the layers in some kind of feed-forward way. All previous layers' local features are utilized as inputs to every layer, while its local features are employed as inputs in all following layers. DenseNet default input dimensions are (224,224,3) (Zhai et al., 2020). DenseNet improved the system by combining all of the image features successively rather than summarizing the resulting feature maps among all preceding layers, as shown in Eq. (7). where the layer index is denoted by  $n$ , while the nonlinear operations are denoted by  $G$  and the feature of the  $n$ th layer is represented by  $f$ .

$$f_n = G_n([f_0, f_1, f_2, \dots, f_{n-1}]) \quad (8)$$

- LeNet is the first publicly available CNNs to get widespread recognition for their effectiveness on tasks involving computer vision. LeNet is composed of two parts: (a) a convolution encoder with 2 convolution layers and (b) a dense network with three FC layers. LeNet default input dimensions are (32,32,3). Equation (8) shows the process of estimating the output in the LeNet model (Bouti et al., 2020).where  $y_n$  is the output layer,  $x_m$  is the vector of the input, and  $\phi_{nm}$  is the vector of the weights.

$$y_n = \sum (x_m - \phi_{nm})^2 \tag{9}$$

- Xception is a 71-layer convolutional neural model used in deep learning applications. The input is routed through the entering layers, the mid-layers, which is performed 8 times, and lastly the output layers (Jie et al., 2020). Xception model default input dimensions are (299,299,3).

- Inception-V3 is composed of asymmetric and symmetric construction elements, containing dense layers, pooling layers, sequences, dropouts, and fully connected layers. Because of the inception modules inside its structure, it has a complicated architecture (Kumthekar & Reddy, 2021). The inception model default input dimensions are (299,299,3).

### The Proposed CNN Model Approach

To achieve a deep learning algorithm based on CNN with enhanced performance and a small amount of error or loss, some parameters have to be chosen accurately such as optimization algorithm, layering structuring, filter sizes, batch size, activation function, learning rate, and a number of filters for a given dataset. The utilized CNN model was built using several max. pooling, convolutional layers, batch normalization, and dense layers. Combining, pairing, and

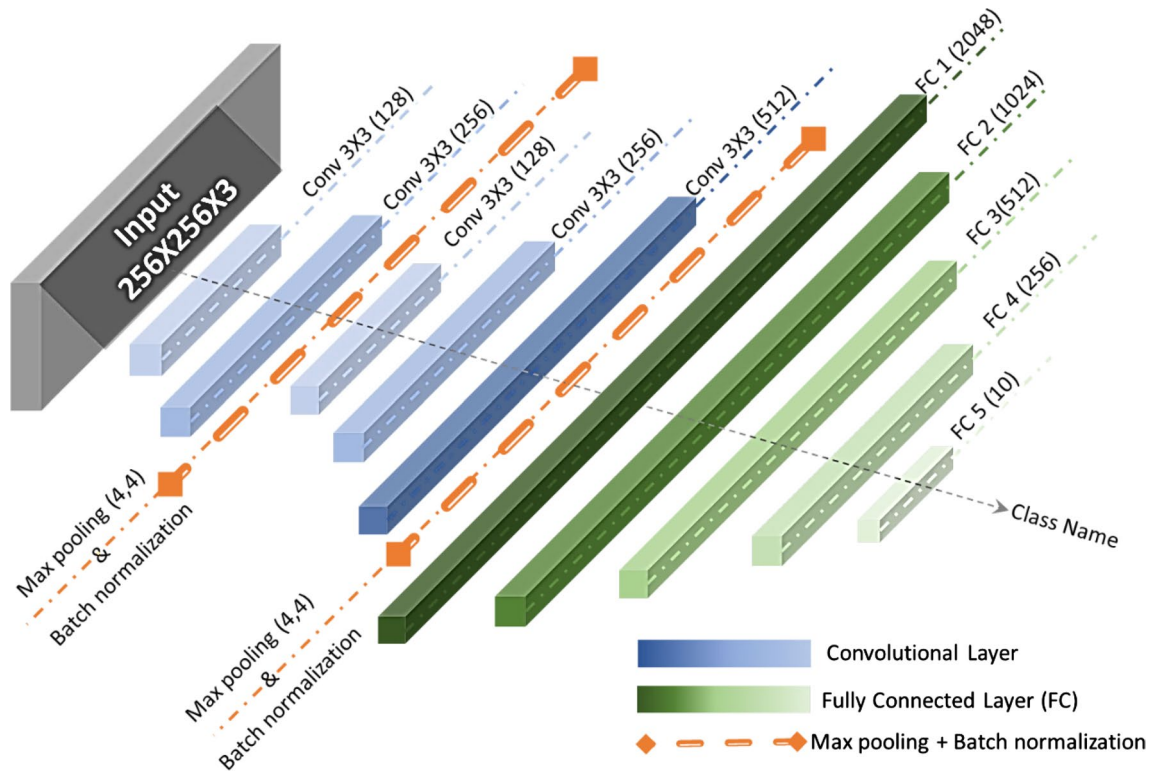


Fig. 5 The full layout of the proposed CNN architecture



Fig. 6 The summary structure of the proposed CNN architecture



layering are employed to construct an appropriate method that outperforms the known pre-trained models. Figures 5 and 6 illustrate the full layout of the suggested CNN algorithm. The proposed CNN structure starts with an input layer with an input image dimension of (128,128,3); after the input layer, there are five convolutional layers combined with 2 max-pooling and 2 batch normalization layers; all the convolution layers are implemented with (same) padding and ReLU activation function as follows:

- (1) Convolutional layer with a kernel size of  $3 \times 3$  and 128 filters implemented with (same) padding and ReLU activation function.
- (2) Convolutional layer with a kernel size of  $3 \times 3$  and 256 filters implemented with (same) padding and ReLU activation function.
- (3) Max. pooling layer with a kernel size of  $4 \times 4$ .
- (4) Batch normalization.
- (5) Convolutional layer with a kernel size of  $3 \times 3$  and 128 filters implemented with (same) padding and ReLU activation function.
- (6) Convolutional layer with a kernel size of  $3 \times 3$  and 256 filters implemented with (same) padding and ReLU activation function.
- (7) Convolutional layer with a kernel size of  $3 \times 3$  and 512 filters implemented with (same) padding and ReLU activation function.
- (8) Max-pooling layer with a kernel size of  $4 \times 4$ .
- (9) Batch normalization.
- (10) Flattened layer with size equal to 2048 units.
- (11) The fully connected part of the suggested model consists of three dense layers using the Relu activation function of 1024, 512, and 256 units in the same order.
- (12) Batch normalization.
- (13) A dense layer of 10 units and SoftMax activation function acts as the output layer or the classifier.

The rest of the parameters are included in the results Section “Parameters”. The number of nodes in the SoftMax layer matches the number of classes that the suggested method is capable of supporting. The loss of all the models in this article is measured based on the sparse categorical cross-entropy, and the activation function used is Adam. The suggested CNN model is based on the pyramid shape as shown in Fig. 5 and aims to enhance the performance of object classification in satellite imagery images to outperform the traditional pre-trained deep learning algorithms concentrating on accuracy and training processing time. The proposed method is named Pyramidal Net due to its pyramidal shape. The modification in the proposed model can be referred to the unique and optimized layering structure, which includes

the selection of the number of filters, kernel size, and optimization technique. The encoding decoding part of the number of filters improves the feature extracting purposes, which increase the overall performance of the system.

## Experimental Results and Discussion

### Experiments Dataset Description

The firstly employed dataset includes 10 different classes from the NWPU-RESISC45 dataset, which are airplane, baseball court, desert, beach, overpass, roundabout, forest, stadium, harbor, and lake. Each class of the ten classes consists of 700 different images. Furthermore, in order to ensure the experimental findings, the UC Merced Land Use dataset has been utilized. The secondly employed database consists of 10 different classes from the UC Merced Land Use dataset, which are airplane, baseball court, chaparral, beach, overpass, parking lot, forest, tennis court, harbor, and agricultural. Each class of the ten classes consists of 100 different images. Those 10 classes in both datasets are chosen randomly to evaluate the model’s performance metrics and compare the proposed model with the pre-trained model’s performance. Both datasets are divided into 70% training set with 4900 images, 15% validation set with 1050 images, and 15% for testing with 1050 images distributed equally for the 10 classes.

### Performance Metrics

Object detection from remote sensing image’ operational efficiency is measured by assessing the suitable accuracy, processing time, and complexity degree. Researchers could evaluate how parameter changes impact the model’s performance during the training process by exploring deep learning approaches. True-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN) measurements are calculated for the measurements (Bouguettaya et al., 2022; Singh et al., 2022). As a result, the following evaluation metrics have been calculated:

1. Accuracy is measured by the number of instances that were correctly detected. Accuracy is calculated by dividing the total number of correctly classified objects by the total number of classifications performed by the algorithm.

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (10)$$

2. Precision is calculated as the total number of true positive classifications divided by all of the algorithm’s positive classifications.

**Table 3** Predefined parameters are assigned to the proposed system

Parameter	Value
Batch size	50
Starting learning rate	$25 \times 10^{-5}$
Min. learning rate	$1 \times 10^{-6}$
Epochs	30
Optimizer	Adam
Number of classes	10
Train, validation, and test percentage	70%, 15%, and 15%

**Table 4** Augmentation techniques

Augmentation technique	Amount
Rotation	0–180
Horizontal flip	–
Vertical flip	–
Zoom-in	0.1

$$\text{precision} = \frac{TP}{TP + FP} \quad (11)$$

3. The proportion of true positively categorized samples to all positively classed samples serves as a measure of recall.

$$\text{recall} = \frac{TP}{FN + TP} \quad (12)$$

4. The F1 score is the weighted mean of precision and recall.

$$F1 - \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (13)$$

5. Intersection Over Union (IOU) is the measure of similarity. Furthermore, it can also be called Jaccard, and it is equal to the proportion between the total of real positive categories and all of the negative classifications.

$$IOU = \frac{TP}{TP + FP + FN} \quad (14)$$

## Parameters

All the results were computed based on the parameters described in Table 3. Furthermore, as we discussed earlier in the article image augmentation has been utilized in all experiments in order to overcome the overfitting problem. Table 4 describes the applied augmentation techniques, which have been done on the input data. The number of epochs is fixed to be 30 epochs for all the utilized models to minimize the training time.

## Selection of Optimization Parameters

This section delves into the reasoning behind the article optimization parameter selections and investigates their impact on the accuracy of the proposed deep learning model for the classification of objects from remote sensing imagery. Choosing the Adam optimizer was one of the most important decisions made within the article. In comparison with the other tested optimization algorithms, Adam frequently converges faster and produces competitive performance with minimum hyper-parameter adjustment. The Adam optimizer was chosen based on theoretical reasons and empirical research. A series of tests with varying learning rates and batch sizes have been done to determine how the Adam optimizer affected the accuracy of the proposed object classification model. The proposed model was trained with initial learning rates that ranged from 0.001 to  $1 \times 10^{-7}$ . We discovered that learning rates much above the optimal range resulted in unstable convergence and overfitting, whereas extremely low learning rates delayed the convergence period without appreciably improving accuracy. A starting learning rate of 0.00025 produced the optimal compromise in convergence speed and accuracy. The batch size utilized during training is another important component in optimization. Larger batch sizes frequently result in faster convergence, but they also raise the possibility of exceeding the ideal solution or becoming stranded in inefficient local minima. The proposed model tested batch sizes of 16, 32, 50, and 64. While bigger batch sizes accelerated convergence, they also showed evidence of overfitting on occasion. A batch size of 50 provided a good mix between convergence speed and generalization.

## Experiment

In this article, a novel CNN model has been utilized and compared with nine different pre-trained models with respect to several performance indicators (accuracy, recall, precision, IOU, and F1-score) for the process of object detection on remote sensing images. The experiment can be categorized into three main stages starting with image resizing and augmentation, then the training of nine pre-trained convolutional deep learning algorithms that are VGG16, VGG19, AlexNet, DenseNet201, ResNet152V2, LeNet5, MobileNet, Xception, and InceptionV3 plus the proposed method, and finally the testing and comparison. The image input size of each CNN algorithm is represented in Table 5. The article has applied the default values of each model provided by the keras library, and for the proposed model we tested an input size of  $224 \times 224 \times 3$ , did not make any changes in performance and also increased the training time while one of the objectives of the proposed model is to increase the accuracy with a small training time in comparison with other

**Table 5** Input image size of the proposed CNN algorithms

Algorithm	Input size
VGG 16	(224,224,3)
VGG 19	(224,224,3)
AlexNet	(224,224,3)
DenseNet 201	(224,224,3)
ResNet 50	(224,224,3)
MobileNet	(224,224,3)
LeNet5	(32,32,3)
Inception V3	(299,299,3)
Xception	(299,299,3)
Proposed model	(128,128,3)

models. Algorithm 1 introduces the full processes of the three stages of the object detection process for the proposed model.

### Algorithm 1 The Proposed Object Detection Technique

**Input:** 7000 remote sensing images (700 from each class).

**Output:** class name classification.

1: **Import** all required libraries.

2: **Import** input images  $x(n,n) \rightarrow X$  and define the 10 classes name  $\rightarrow Y$ .

3: **For** all  $x(n,n)$  in  $X$  **do**

4:     Resize  $x(n,n) \rightarrow x(m,m)$

5:      $X.append(x(m,m))$

# (following table 5 values)

6: **End for**

7: **Split**  $X'$  training  $\rightarrow Tr$ , validation  $\rightarrow V$ , and testing  $\rightarrow Ts$

# (following table 3 values)

8: **For** all  $x(m,m)$  in  $Tr$  &  $V$  **do**

9:     Augmentation

#( following table 4 values)

10:     Data1  $\leftarrow$  Rotation (0:180)

11:     Data2  $\leftarrow$  Horizontal flip

12:     Data3  $\leftarrow$  Vertical flip

13:     Data4  $\leftarrow$  Zoom in (0.1)

14:      $Tr' \leftarrow$  Add ( $Tr$ , Data1, Data2, Data3, Data4)

15: **End for**

16: **Initialize** learning rate reduction technique  $\rightarrow Min\_Lr = 1 \times 10^{-6}$ .

17: **Initialize** parameters  $\rightarrow$  number\_of\_epochs= 30, batch\_size= 50.

18: **Initialize** Optimizer  $\rightarrow$  optimizer='adam', learning\_rate =  $25 \times 10^{-5}$ .

19: **Build** the CNN models structure (VGG16, VGG19, AlexNet, DenseNet, LeNet, ResNet, MobileNet, Inception, Xception).

20: **Build** the Proposed Model (Pyramidal Model)

#( following Fig. 5 Structure)

16:     Model.add  $\rightarrow$  Layers ( Conv2D, Maxpool, Batch Normalization, Flateen, Denese, Softmax)

31: **For** all the 10 models **do**

32:     **Train Model**  $\leftarrow Tr', V$

33:     **Test Model**  $\leftarrow Ts$

34:     **Plot**  $\rightarrow$  accuracy and loss curves.

35:     **Calculate**  $\rightarrow$  Precision, Recall, IOU, and F1-score.

36:     **Plot**  $\rightarrow$  confusion matrix

37:     **Save Model weights**

38: **End For**

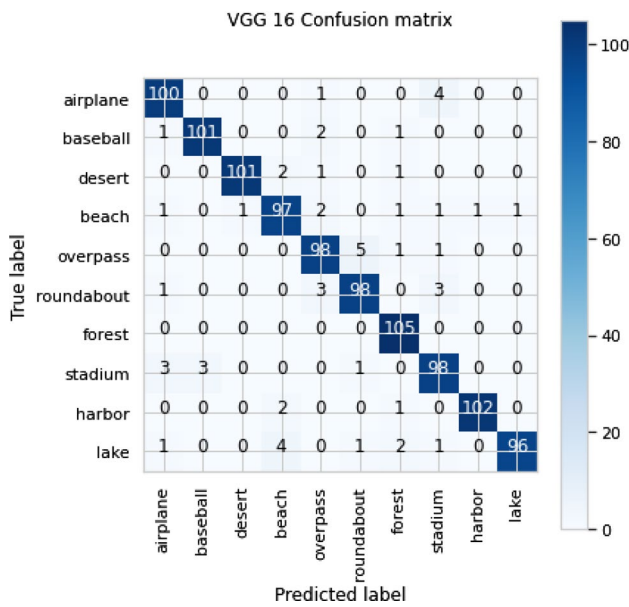
Table 6 introduces a complete comparison study between the proposed model and the most recent pre-trained deep structures with respect to test loss, test accuracy, train loss, train accuracy, validation loss, and validation accuracy taking into consideration the same number of epochs. Moreover, both confusion matrices as shown in Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 and accuracy and loss curves in Figs. 17, 18, 19, 20, 21, 22, 23, 24, 25 and 26 have been utilized and demonstrated. Accordingly, the proposed methodology has the best performance among all the tested pre-trained CNN models. Even if some pre-trained algorithms score a little close to the proposed algorithm as Xception, InceptionV3, and VGG16 CNN models, the proposed model training time was the smallest among the other pre-trained models.

**Table 6** Comparison between different pre-trained CNN algorithms and the proposed model on the NWPU-RESISC45 dataset

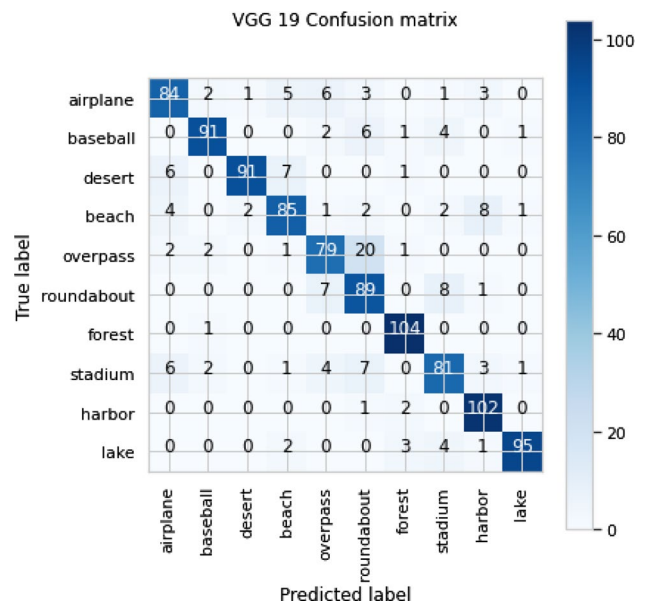
CNN model	Test loss	Test accuracy	Train loss	Train accuracy	Validation loss	Validation accuracy	Epochs
VGG 16	0.188	0.949	0.073	0.975	0.250	0.935	30
VGG 19	0.407	0.858	0.350	0.881	0.440	0.852	30
AlexNet	0.259	0.908	0.200	0.934	0.298	0.912	30
MobileNet	0.558	0.810	0.488	0.825	0.486	0.834	30
ResNet 152V2	0.415	0.865	0.363	0.886	0.378	0.876	30
DenseNet 201	0.264	0.923	0.125	0.959	0.241	0.929	30
LeNet	1.061	0.648	0.793	0.732	0.834	0.699	30
Xception	0.168	0.948	0.060	0.980	0.214	0.950	30
Inception	0.225	0.937	0.137	0.961	0.214	0.940	30
Proposed model	0.112	0.963	0.023	0.995	0.109	0.971	30

CNN model	Precision	Recall	F1	IOU	Image size	Training time (S)	Lr reduction epochs
VGG 16	0.95	0.95	0.95	0.902	(224,224,3)	2303.5	16,23,27,28
VGG 19	0.86	0.86	0.86	0.751	(224,224,3)	2360.1	5,19,23,26
AlexNet	0.91	0.91	0.91	0.831	(224,224,3)	2365.5	6,17,21,25
MobileNet	0.82	0.81	0.81	0.680	(224,224,3)	1995.6	13,18,19,22
ResNet 152V2	0.87	0.86	0.87	0.762	(224,224,3)	3454.5	6,13,15,20
DenseNet 201	0.92	0.92	0.92	0.857	(224,224,3)	2288.9	4,13,16,20
LeNet	0.70	0.65	0.65	0.479	(32,32,3)	87.1	14,17,18,19
Xception	0.95	0.95	0.95	0.900	(299,299,3)	4749.9	5,11,18,24
Inception	0.94	0.94	0.94	0.882	(299,299,3)	3904.2	6,20,23,25
Proposed model	0.96	0.96	0.96	0.928	(128,128,3)	844.3	13,20,27,29



**Fig. 7** VGG16 confusion matrix



**Fig. 8** VGG19 confusion matrix

In order to ensure the experimental result findings, both the proposed pyramidal CNN algorithm and the pre-trained algorithms have been applied to both the UC Merced Land Use datasets. The experimental results showed that the

suggested pyramidal model outperforms all other CNN models that have been evaluated. Furthermore, the pyramidal model has a great performance dealing with small-sized



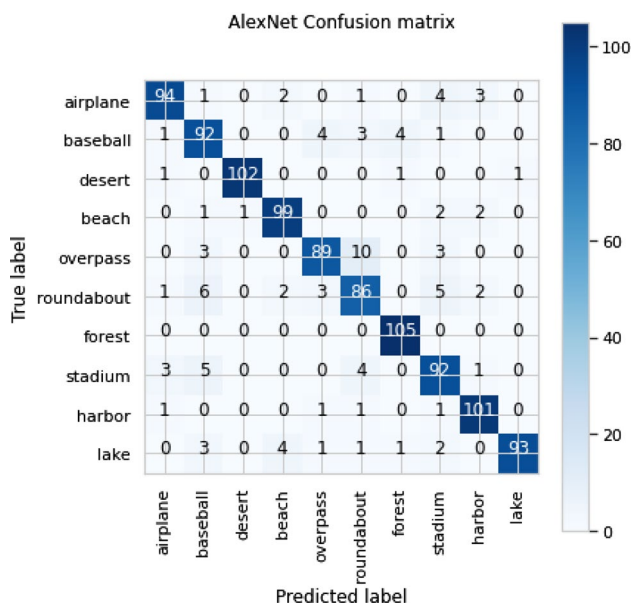


Fig. 9 AlexNet confusion matrix

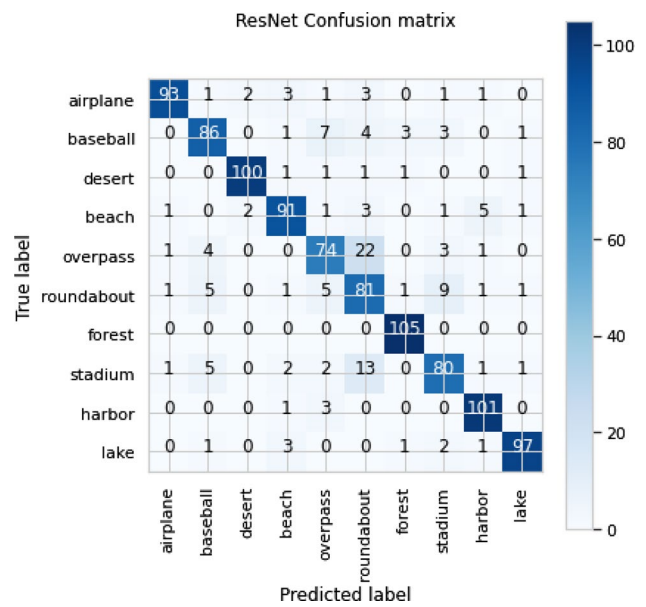


Fig. 11 ResNet152V2 confusion matrix

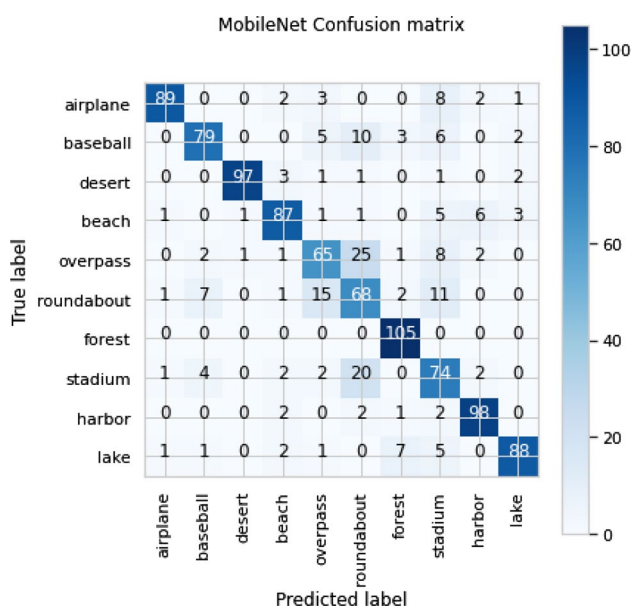


Fig. 10 MobileNet confusion matrix

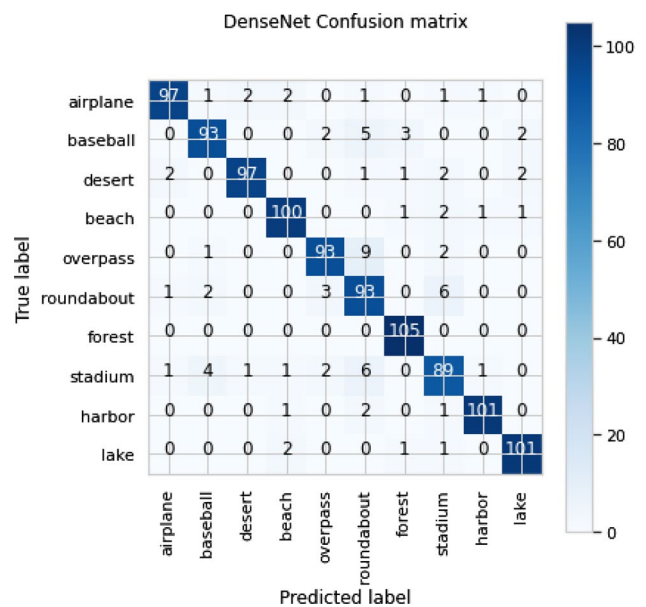


Fig. 12 DenseNet201 confusion matrix

datasets unlike most of the pre-trained models which have an overfitting problem as it is represented in Table 7.

In order to guarantee that the proposed model is size invariant and able to classify objects with different sizes, an additional experiment that categorizes the accuracy in terms of the size of the output classes has been done, by

taking three classes from NWPU-RESISC45 dataset, small size object (airplanes), medium size object (stadium) and large size object (desert) each with 700 images. These 2100 images are then split into a 70% training set, a 15% validation set, and a 15% test set. In this experiment, to avoid overfitting all the convolutional layers filters count of the proposed model are divided by two. Tables 8 and 9 show the performance metrics of the proposed model over this experiment. Finally, Fig. 27 shows the accuracy and loss curves of

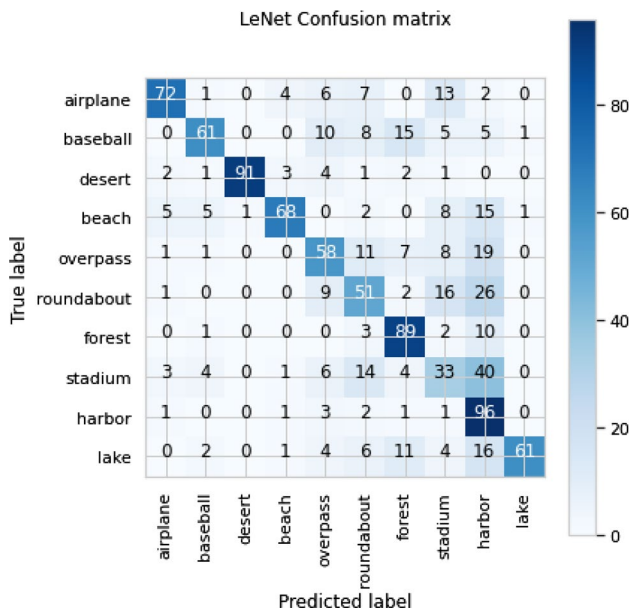


Fig. 13 LeNet confusion matrix

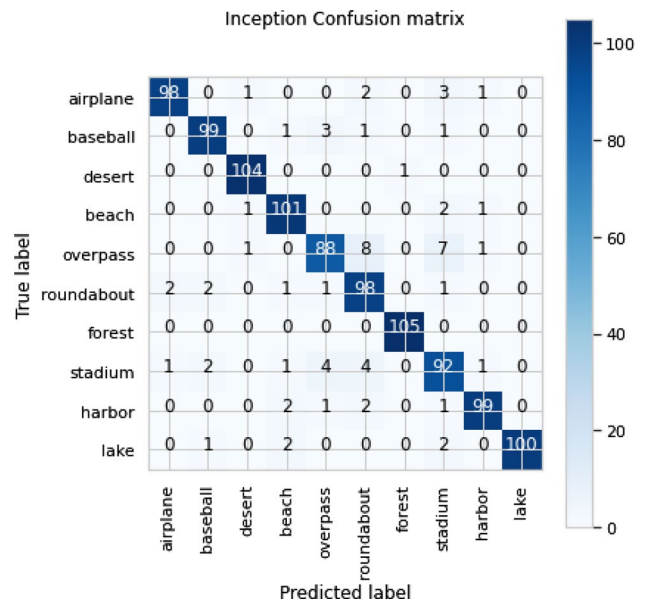


Fig. 15 Inception confusion matrix

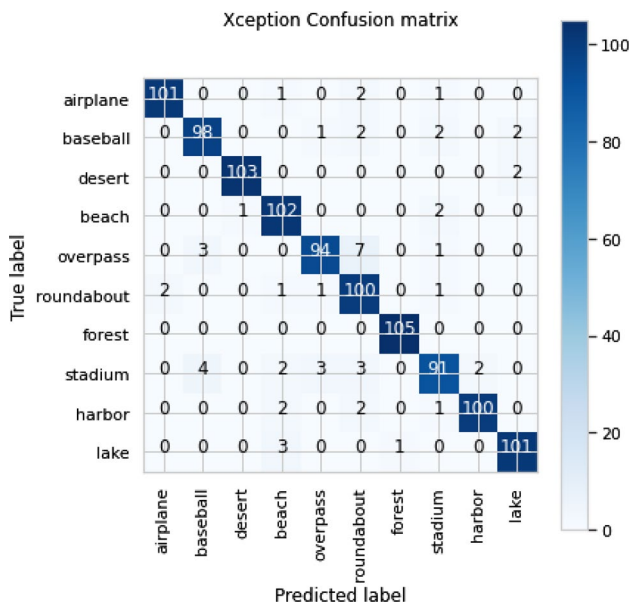


Fig. 14 Xception confusion matrix

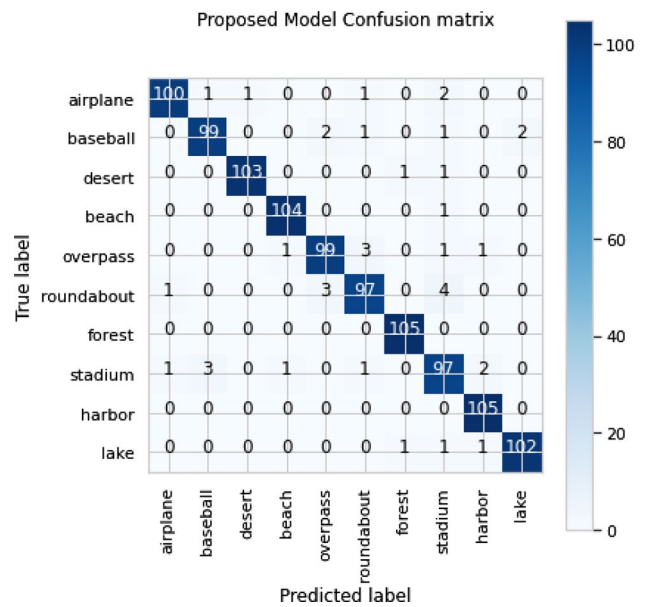


Fig. 16 Proposed model confusion matrix

the experiment, while Fig. 28 shows the confusion matrix of the three classes.

According to Figs. 27 and 28, as well as the performance metrics illustrated in Tables 8 and 9, the proposed model can be considered as a size-invariant model, because of the high accuracy the proposed model had in this experiment. Furthermore, in order to ensure that the proposed model is illumination-invariant, the article examined the proposed model performance on a binary dataset that contains 700

images with clouds and 700 clear images (without clouds) mixed equally from all of the other nine classes used in the NWPU-RESISC45 dataset; these 1400 images are then split to 70% training set, 15% validation set, and 15% test set. Brightness augmentation with a brightness range between [0.2–2] was used during dataset preparation to strengthen the model's ability to handle shifting lighting conditions. This augmentation method adds to the model's robustness in varied illumination situations. In this experiment to avoid

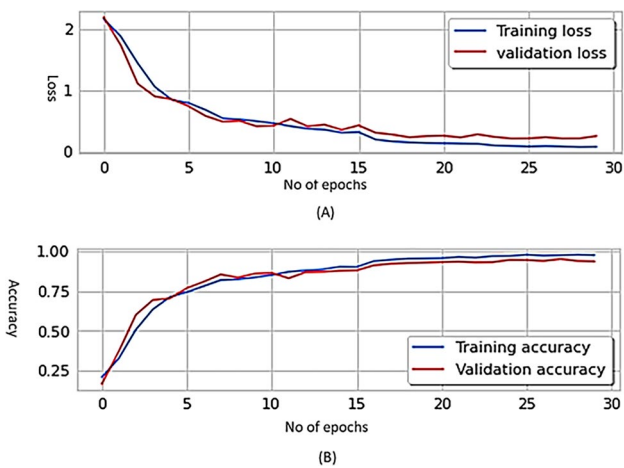


Fig. 17 **A** VGG16 loss curve, **B** VGG16 accuracy curve

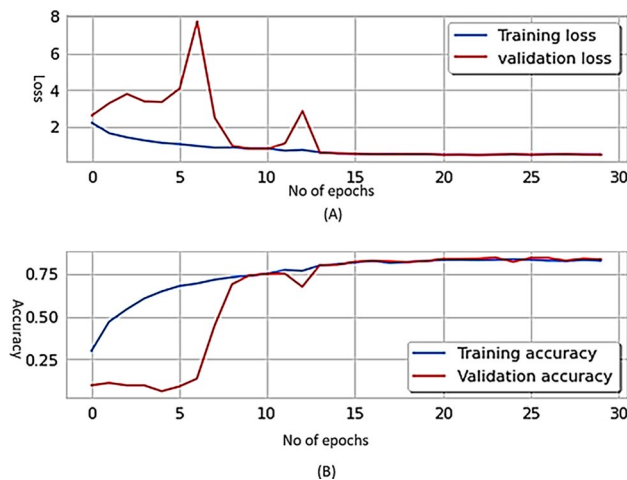


Fig. 20 **A** MobileNet loss curve, **B** MobileNet accuracy curve

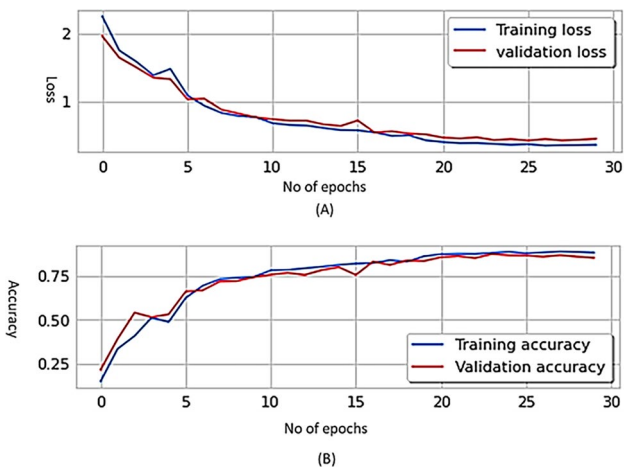


Fig. 18 **A** VGG19 loss curve, **B** VGG19 accuracy curve

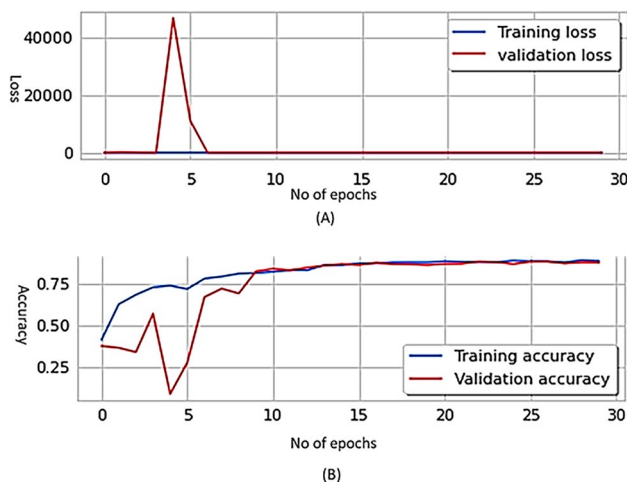


Fig. 21 **A** ResNet152V2 loss curve, **B** ResNet152V2 accuracy curve

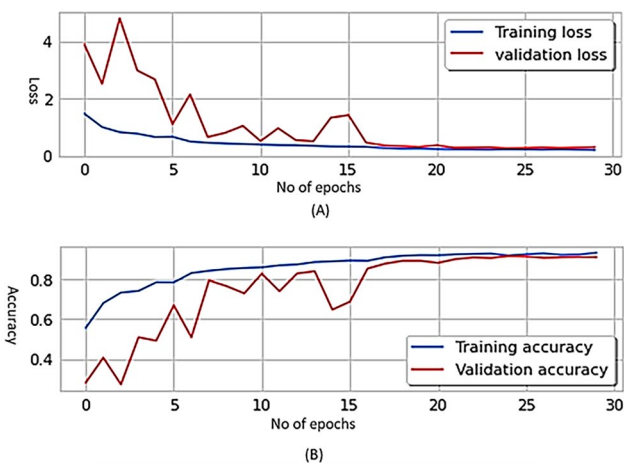


Fig. 19 **A** AlexNet loss curve, **B** AlexNet accuracy curve

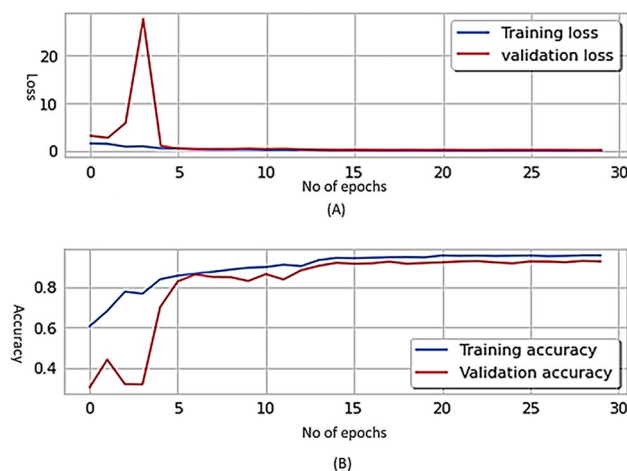


Fig. 22 **A** DenseNet201 loss curve, **B** DenseNet201 accuracy curve

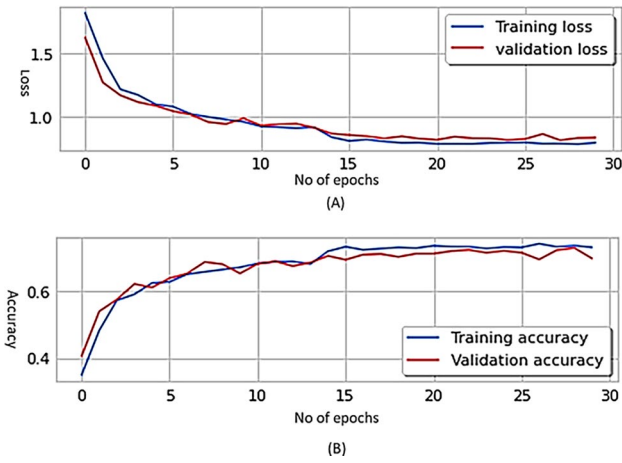


Fig. 23 **A** LeNet loss curve, **B** LeNet accuracy curve

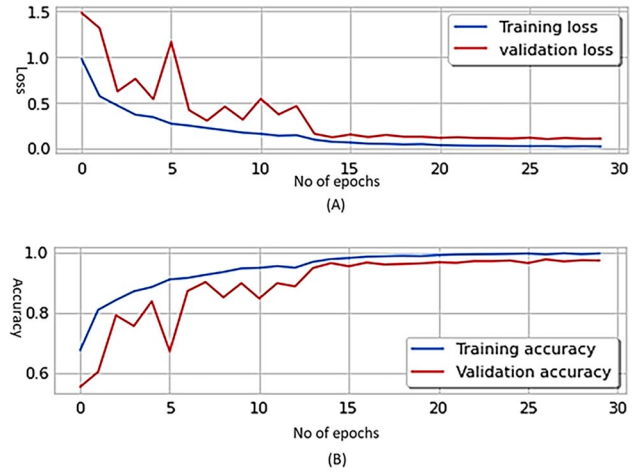


Fig. 26 **A** Proposed model loss curve, **B** proposed model accuracy curve

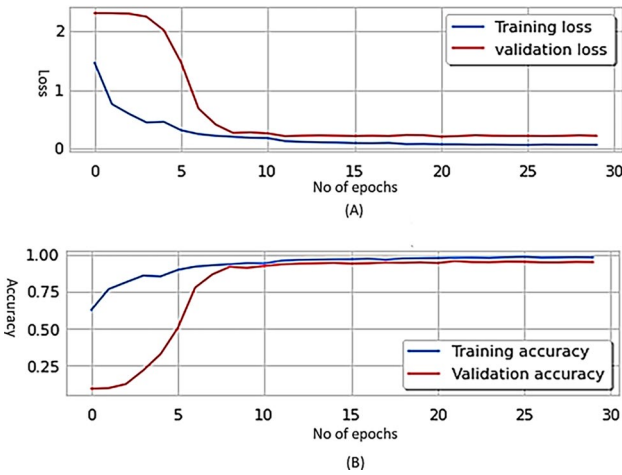


Fig. 24 **A** Xception loss curve, **B** Xception accuracy curve

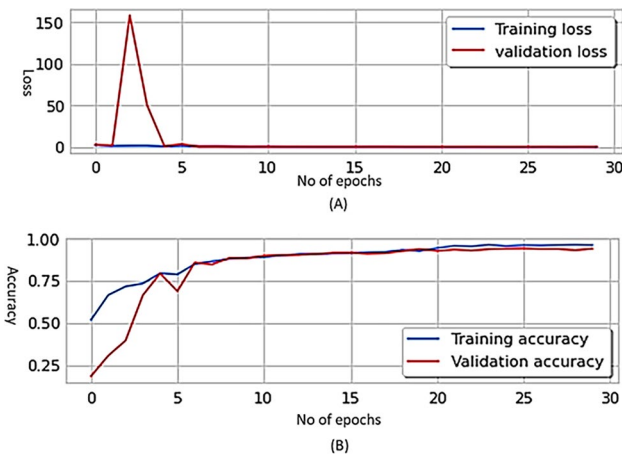


Fig. 25 **A** Inception loss curve, **B** inception accuracy curve

overfitting all the convolutional layers filters count of the proposed model are divided by two. Figure 29 shows the accuracy and loss curves of the experiment, while Fig. 30 shows the confusion matrix of the two classes. Finally, Table 10 shows the performance metrics of the proposed model over this experiment.

According to Figs. 29 and 30, as well as the performance metrics illustrated in Table 10, the proposed model can be considered as an illumination invariant model, because of the high accuracy the proposed model had this experiment.

Finally, to guarantee that the proposed model has a positive effect on such remote sensing images classification performance the article compared the proposed method with some recently published research articles in the remote sensing images classification field, which is represented in Table 11.

### Aspects Contributed to Accuracy Improvement

In order to enhance the performance of the proposed model, the article contributed with different parameters including the number of filters, layering structure, kernel size, and optimization techniques. The unique and optimized layering structure in the suggested model, which includes the choice of the number of filters, kernel size, and the overall hierarchical representation enhanced features extraction, which boosts the system's overall performance. Furthermore, a learning rate reduction technique has been done to improve convergence and model generalization. In addition, the proposed model applied data augmentation, which increased the proposed model's exposure to various data fluctuations, which increased the model's robustness. Later Adam was chosen as the optimization



**Table 7** Comparison between different pre-trained CNN algorithms and the proposed model on the UC Merced Land Use dataset

CNN model	Test loss	Test accuracy	Train loss	Train accuracy	Validation loss	Validation accuracy	Epochs
VGG 16	1.142	0.667	0.920	0.673	1.145	0.596	30
VGG 19	1.278	0.567	0.962	0.633	1.146	0.556	30
AlexNet	0.741	0.787	0.345	0.889	0.413	0.894	30
MobileNet	3.293	0.100	1.315	0.557	3.429	0.100	11
ResNet 152V2	0.890	0.707	0.476	0.857	0.551	0.815	30
DenseNet 201	0.765	0.833	0.089	0.979	0.242	0.940	30
LeNet	1.485	0.460	1.251	0.566	1.272	0.573	30
Xception	2.305	0.100	0.075	0.971	2.314	0.080	11
Inception	5.926	0.113	0.308	0.926	6.204	0.127	11
Proposed model	0.226	0.940	0.036	0.996	0.197	0.947	30

CNN model	Precision	Recall	F1	IOU	Image size	Training time (S)	Lr reduction epochs
VGG 16	0.61	0.67	0.63	0.500	(224, 224, 3)	453.6	5,10,16,19
VGG 19	0.50	0.57	0.50	0.395	(224, 224, 3)	421.0	3,4,17,19
AlexNet	0.82	0.79	0.79	0.648	(224, 224, 3)	285.4	6,12,13,14
MobileNet	0.01	0.10	0.02	0.053	(224, 224, 3)	124.0	7,11
ResNet 152V2	0.74	0.71	0.70	0.546	(224, 224, 3)	502.2	8,12,14,16
DenseNet 201	0.86	0.83	0.83	0.714	(224,224,3)	382.3	4,14,18,19
LeNet	0.49	0.46	0.44	0.299	(32, 32, 3)	14.1	6,8,10,11
Xception	0.01	0.10	0.02	0.053	(299, 299, 3)	266.1	4,11
Inception	0.02	0.11	0.03	0.060	(299, 299, 3)	205.1	6
Proposed Model	0.95	0.94	0.94	0.887	(128, 128, 3)	123.2	10,15,17,19

**Table 8** The proposed model performance metrics of the three classes dataset individually

Class	Precision	Recall	F1-Score
Airplane	0.98	0.90	0.94
Stadium	0.90	0.99	0.94
Desert	0.98	0.95	0.97

algorithm, which sped up convergence and improved the use of gradient data.

## Challenges

In this section, the main potential limitations and challenges that the proposed model may encounter have been highlighted:

1. Limited data set variation: One of the limitations the proposed model faces is the lack of a dataset that con-

tains variations in seasonal conditions, lighting, and environment.

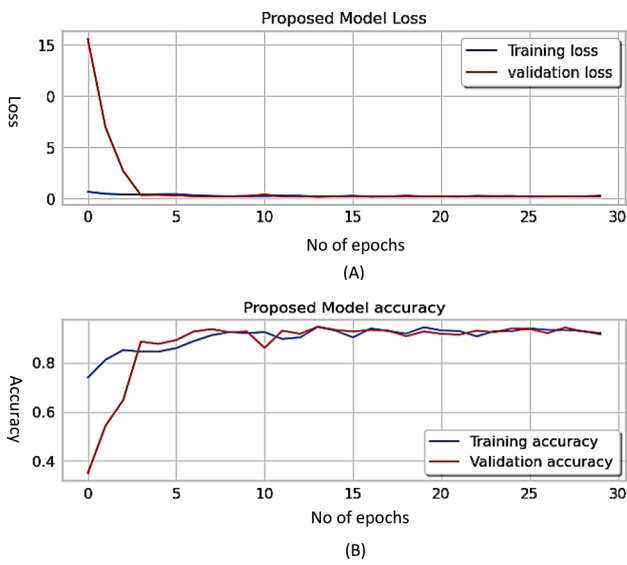
2. Errors in labeling and annotation: The accuracy of the proposed model is largely dependent on the caliber of the labels applied to the training data. The performance of the model as a whole could be affected by inherent flaws in labeling or annotation.
3. The ability to transfer to other domains: Although our model is designed for object classification in satellite pictures, it may need additional tuning or adaptations in order to be used in other domains or datasets.

## Areas of Improvement

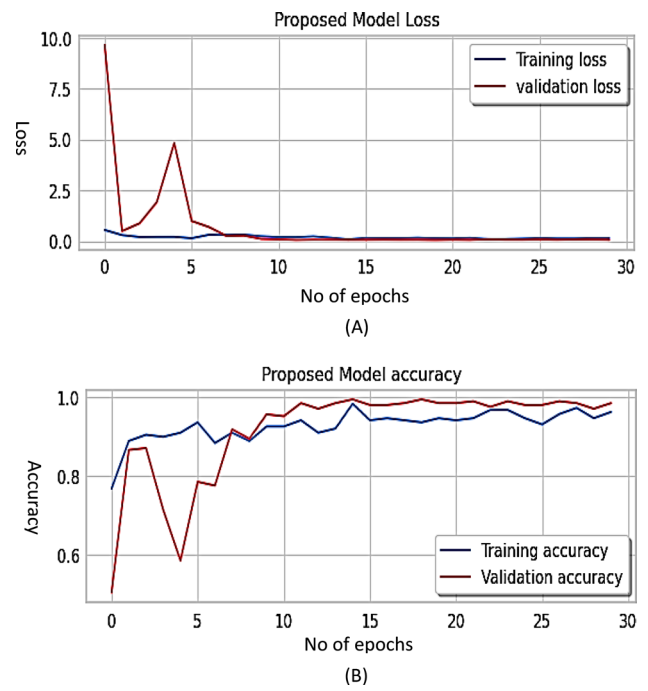
1. Enhance data augmentation: utilize more diverse and complex techniques of data augmentation to make the model more resistant to variations in satellite photographs, such as changes in illumination, weather, and seasonal circumstances.

**Table 9** The overall performance of the proposed model of the three classes dataset

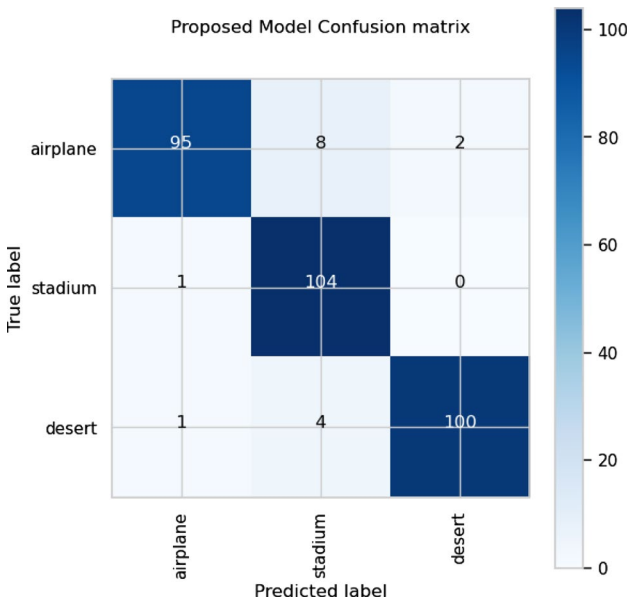
CNN model	Test loss	Test accuracy	Train loss	Train accuracy	Validation loss	Validation accuracy	Epochs
Proposed model	0.14	0.95	0.2467	0.9156	0.2118	0.9206	30



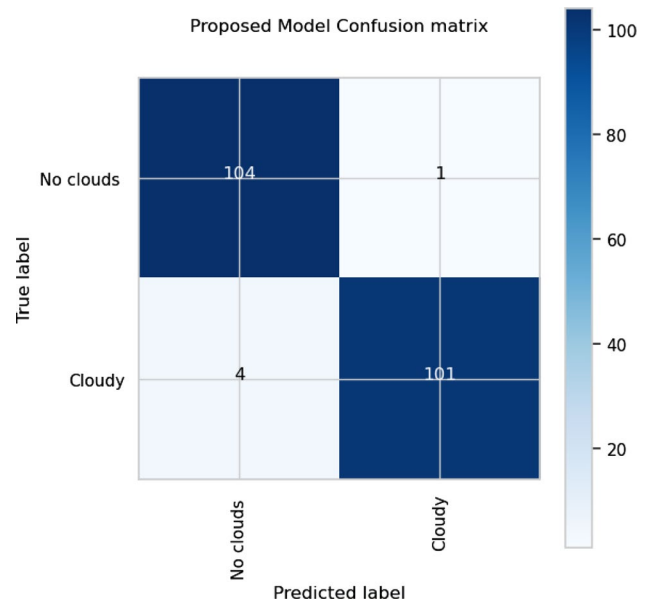
**Fig. 27** **A** Proposed model loss curve of the three classes, **B** proposed model accuracy curve of the three classes



**Fig. 29** **A** Proposed model loss curve of a two-class dataset, **B** proposed model accuracy curve of a two-class dataset



**Fig. 28** Proposed model confusion matrix of the three classes



**Fig. 30** Proposed model confusion matrix of two-class dataset

2. Preprocessing: Before incorporating satellite photographs into models, create preprocessing methods that improve noise reduction, feature extraction, and overall data quality.

3. Object localization: Increase the model's capacity to precisely localize and specify object boundaries inside the satellite pictures in addition to classifying objects.
4. Ensemble Models: Explore ensemble learning techniques by mixing different models or model variants to improve the overall classification accuracy.

**Table 10** The overall performance of the proposed model of a two-class dataset

CNN model	Test loss	Test accuracy	Train loss	Train accuracy	Validation loss	Validation accuracy	Epochs
Proposed model	0.086	0.976	0.1489	0.96	0.068	0.986	30

**Table 11** Comparison between the proposed method and related work for remote sensing images classification accuracy

Model	Year	Accuracy (%)	Training time	Complexity
SMDTR-CNN (Cheng et al., 2020)	2019	95	High	High
RNN-RF (Ma et al., 2021)	2021	87	Moderate	Low
SceneNet (Rohith & Kumar, 2022)	2021	95.2	High	High
Modified 13 layers CNN structure (Zhang et al., 2022)	2022	88.9	High	Low
Multi-label land cover CNN (Stivaktakis et al., 2019)	2019	82.29	Moderate	Low
MSMatch (Gómez & Meoni, 2021)	2021	90.71	Moderate	Moderate
Global–local dual-branch structure model (Xu et al., 2022)	2022	94.46	Moderate	High
TFADDN (Xu et al., 2020)	2020	90.86	Moderate	Moderate
MIDC-Net CS (Bi et al., 2020)	2020	87.99	Moderate	Moderate
mmsCNN–HMM combined model with stacking ensemble (Cheng & Lei, 2022)	2022	95.5	High	High
Two-speed deep-learning ensemble (Horry et al., 2023)	2023	64	Low	Low
Proposed model	2022	97.1	Moderate	Moderate

## Conclusion

This article introduces a robust CNN model structure for object detection from remote sensing images. The article focused on building an optimized CNN model with a novel structure and suitable hyper-parameters to be able to classify remote sensing images with a performance that exceeds the pre-trained models' performances with the smallest possible training time. The article evaluates the effectiveness of the presented models based on both the NWPU-RESISC45 and UC Merced Land Use dataset. All findings have concluded that the proposed pyramidal CNN model structure has the highest detection accuracy with a very small training time in comparison with the well-known pre-trained CNN models, and it can be utilized efficiently for object detection processes from remote sensing images with an accuracy reaching 97.1%. The proposed model had a great performance dealing with different size object classes and different illumination datasets. Furthermore, the pyramidal model showed a great performance dealing with small datasets, unlike the traditional pre-trained datasets. Future work may include utilizing more optimization algorithms, deep learning models, and classes. In addition, improving the proposed CNN structure could be our future work of interest.

**Acknowledgements** The authors declare that there is no conflict of interest regarding the manuscript.

**Author Contributions** We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

**Funding** There is no funding to declare regarding this manuscript.

**Data Availability** The data that support the findings of this study are available within this article.

## Declarations

**Conflict of interest** No potential competing interest was reported by the authors.

## References

- Bi, Q., Qin, K., Li, Z., Zhang, H., Xu, K., & Xia, G.-S. (2020). A multiple-instance densely-connected convnet for aerial scene classification. *IEEE Transactions on Image Processing*, 29, 4911–4926. <https://doi.org/10.1109/TIP.2020.2975718>
- Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2022). Deep learning techniques to classify agricultural crops through UAV imagery: A review. *Neural Computing and Applications*, 34, 9511–9536. <https://doi.org/10.1007/s00521-022-07104-9>
- Bouti, A., Mahraz, M. A., Riffi, J., & Tairi, H. (2020). A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network. *Soft Computing*, 24(9), 6721–6733. <https://doi.org/10.1007/s00500-019-04307-6>
- Chen, J., Sun, J., Li, Y., & Hou, C. (2022). Object detection in remote sensing images based on deep transfer learning. *Multimedia Tools*

- and Applications, 81(9), 12093–12109. <https://doi.org/10.1007/s11042-021-10833-3>
- Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10), 1865–1883. <https://doi.org/10.1109/JPROC.2017.2675998>
- Cheng, G., Xie, X., Han, J., Guo, L., & Xia, G.-S. (2020). 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, 3735–3756. <https://doi.org/10.1109/JSTARS.2020.3005403>
- Cheng, X., & Lei, H. (2022). Remote sensing scene image classification based on mmsCNN–HMM with stacking ensemble model. *Remote Sensing*, 14(17), 4423. <https://doi.org/10.3390/rs14174423>
- Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5), 545–563. <https://doi.org/10.1111/1754-9485.13261>
- Cui, X., Zou, C., & Wang, Z. (2021). Remote sensing image recognition based on dual-channel deep learning network. *Multimedia Tools and Applications*, 80(18), 27683–27699. <https://doi.org/10.1007/s11042-021-11079-5>
- d’Acremont, A., Fablet, R., Baussard, A., & Quin, G. (2019). CNN-based target recognition and identification for infrared imaging in defense systems. *Sensors*, 19(9), 1–16. <https://doi.org/10.3390/s19092040>
- Darehnaei, Z. G., Shokouhifar, M., Yazdanjoui, H., & Rastegar Fatemi, S. M. J. (2022). SI-EDTL: Swarm intelligence ensemble deep transfer learning for multiple vehicle detection in UAV images. *Concurrency and Computation Practice and Experience*, 34(5), e6726. <https://doi.org/10.1002/cpe.6726>
- Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: A review of models, methodologies, and applications to object detection. *Progress in Artificial Intelligence*, 9(2), 85–112. <https://doi.org/10.1007/s13748-019-00203-0>
- Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162, 94–114.
- Dong, L., Du, H., Mao, F., Han, N., Li, X., Zhou, G., Zheng, J., Zhang, M., Xing, L., & Liu, T. (2020). Very high-resolution remote sensing imagery classification using a fusion of random forest and deep learning technique—Subtropical area for example. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 113–128. <https://doi.org/10.1109/JSTARS.2019.2953234>
- Feng, Y., Wang, L., & Zhang, M. (2019). A multi-scale target detection method for optical remote sensing images. *Multimedia Tools and Applications*, 78(7), 8751–8766. <https://doi.org/10.1007/s11042-018-6325-6>
- Gómez, P., & Meoni, G. (2021). MSMATCH: Semisupervised multispectral scene classification with few labels. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 11643–11654. <https://doi.org/10.48550/arXiv.2103.10368>
- Horry, M. J., Chakraborty, S., Pradhan, B., Shulka, N., & Almazroui, M. (2023). Two-speed deep-learning ensemble for classification of incremental land-cover satellite image patches. *Earth Systems and Environment*, 7(2), 525–540. <https://doi.org/10.1007/s41748-023-00343-3>
- Hou, D., Miao, Z., Xing, H., & Wu, H. (2020). Exploiting low dimensional features from the MobileNets for remote sensing image retrieval. *Earth Science Informatics*, 13(4), 1437–1443. <https://doi.org/10.1007/s12145-020-00484-3>
- Jie, B. X., Zulkifley, M. A., & Mohamed, N. A. (2020). Remote sensing approach to oil palm plantations detection using xception. In *2020 11th IEEE control and system graduate research colloquium (ICSGRC)* (pp. 38–42). <https://doi.org/10.1109/ICSGRC49013.2020.9232547>
- Karim, S., Zhang, Y., Yin, S., Laghari, A. A., & Brohi, A. A. (2019). Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery. *Multimedia Tools and Applications*, 78(22), 32565–32583. <https://doi.org/10.1007/s11042-019-08033-x>
- Karnick, S., Ghalib, M. R., Shankar, A., Khapre, S., & Tayubi, I. (2022). A novel method for vehicle detection in high-resolution aerial remote sensing images using YOLT approach. *Multimedia Tools and Applications*, 81, 23551–23566. <https://doi.org/10.1007/s11042-022-12613-9>
- Khalifa, N. E., Loey, M., & Mirjalili, S. (2022). A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review*, 55, 2351–2377. <https://doi.org/10.1007/s10462-021-10066-4>
- Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Kingma, D. P., & Ba, J. L. (2015). *Published as a Conference Paper at the 3rd International Conference for Learning Representations, San Diego*. arXiv preprint <https://doi.org/10.48550/arXiv.1412.6980>
- Kodali, R. K., & Dhanekula, R. Face mask detection using deep learning. In *2021 international conference on computer communication and informatics (ICCCI)* (pp. 1–5). <https://doi.org/10.1109/ICCCI50826.2021.9402670>
- Kumar, A., Abhishek, K., Kumar Singh, A., Nerurkar, P., Chandane, M., Bhirud, S., Patel, D., & Busnel, Y. (2021). Multilabel classification of remote sensed satellite imagery. *Transactions on Emerging Telecommunications Technologies*, 32(7), e3988. <https://doi.org/10.1002/ett.3988>
- Kumthekar, A., & Reddy, G. R. (2021). An integrated deep learning framework of U-Net and inception module for cloud detection of remote sensing images. *Arabian Journal of Geosciences*, 14(18), 1–13. <https://doi.org/10.1007/s12517-021-08259-w>
- Lei, J., Luo, X., Fang, L., Wang, M., & Gu, Y. (2020). Region-enhanced convolutional neural network for object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(8), 5693–5702. <https://doi.org/10.1109/TGRS.2020.2968802>
- Li, W., Liu, H., Wang, Y., Li, Z., Jia, Y., & Gui, G. (2019). Deep learning-based classification methods for remote sensing images in urban built-up areas. *IEEE Access*, 7, 36274–36284. <https://doi.org/10.1109/ACCESS.2019.293127>
- Li, W., Wang, Z., Wang, Y., Wu, J., Wang, J., Jia, Y., & Gui, G. (2020). Classification of high-spatial-resolution remote sensing scenes method using transfer learning and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 1986–1995. <https://doi.org/10.1109/JSTARS.2020.2988477>
- Liang, J., Deng, Y., & Zeng, D. (2020). A deep neural network combined CNN and GCN for remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 4325–4338. <https://doi.org/10.1109/JSTARS.2020.3011333>
- Lin, Y., & Wu, L. (2019). Improved abrasive image segmentation method based on bit-plane and morphological reconstruction. *Multimedia Tools and Applications*, 78(20), 29197–29210. <https://doi.org/10.1007/s11042-018-6687-9>
- Liu, S., Zhang, L., Lu, H., & He, Y. (2022). Center-boundary dual attention for oriented object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60(5603914), 1–14. <https://doi.org/10.1109/TGRS.2021.3069056>



- Ma, A., Wan, Y., Zhong, Y., Wang, J., & Zhang, L. (2021). SceneNet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search. *ISPRS Journal of Photogrammetry and Remote Sensing*, 172, 171–188. <https://doi.org/10.1016/j.isprsjprs.2020.11.025>
- Manickam, A., Jiang, J., Zhou, Y., Sagar, A., Soundrapandiyam, R., & Samuel, R. D. J. (2021). Automated pneumonia detection on chest X-ray images: A deep learning approach with different optimizers and transfer learning architectures. *Measurement*, 184, 109953. <https://doi.org/10.1016/j.measurement.2021.109953>
- Marastoni, N., Giacobazzi, R., & Preda, M. D. (2021). Data augmentation and transfer learning to classify malware images in a deep learning context. *Journal of Computer Virology and Hacking Techniques*, 17(4), 279–297. <https://doi.org/10.1007/s11416-021-00381-3>
- Napiorkowska, M., Petit, D., & Marti, P. (2018). Three applications of deep learning algorithms for object detection in satellite imagery. In *IGARSS 2018—2018 IEEE international geoscience and remote sensing symposium* (pp. 4839–4842). <https://doi.org/10.1109/IGARSS.2018.8518102>
- Özyurt, F. (2020). Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. *The Journal of Supercomputing*, 76(11), 8413–8431. <https://doi.org/10.1007/s11227-019-03106-y>
- Pang, S., & Gao, L. (2022). Multihead attention mechanism guided ConvLSTM for pixel-level segmentation of ocean remote sensing images. *Multimedia Tools and Applications*, 81, 24627–24643. <https://doi.org/10.1016/j.isprsjprs.2020.01.013>
- Pathak, D., & Raju, U. S. (2022). Content-based image retrieval for super-resolution images using feature fusion: Deep learning and hand crafted. *Concurrency and Computation: Practice and Experience*, 34(22), e6851. <https://doi.org/10.1002/cpe.6851>
- Peng, S., Sun, S., & Yao, Y.-D. (2021). A survey of modulation classification using deep learning: Signal representation and data preprocessing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 7020–7038. <https://doi.org/10.1109/TNNLS.2021.3085433>
- Priya, R. S. & Vani, K. (2019). Deep learning based forest fire classification and detection in satellite images. In *2019 11th international conference on advanced computing (ICoAC)* (pp. 61–65). <https://doi.org/10.1109/ICoAC48765.2019.246817>
- Ran, Q., Xu, X., Zhao, S., Li, W., & Du, Q. (2019). Remote sensing images super-resolution with deep convolution networks. *Multimedia Tools and Applications*, 79(13), 8985–9001. <https://doi.org/10.1007/s11042-018-7091-1>
- Rohith, G., & Kumar, L. S. (2022). Design of deep convolution neural networks for categorical signature classification of raw panchromatic satellite images. *Multimedia Tools and Applications*, 81, 28367–28404. <https://doi.org/10.1007/s11042-022-12928-7>
- Sarwinda, D., Paradisa, R. H., Bustamam, A., & Anggia, P. (2021). Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. *Procedia Computer Science*, 179, 423–431. <https://doi.org/10.1016/j.procs.2021.01.025>
- Sharma, M., Dhanaraj, M., Karnam, S., Chachlakis, D. G., Ptucha, R., Markopoulos, P. P., & Saber, E. (2021). YOLOrs: Object detection in multimodal remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 1497–1508. <https://doi.org/10.1109/JSTARS.2020.3041316>
- Singh, N., Tewari, V. K., Biswas, P. K., Dhruw, L. K., Pareek, C. M., & Singh, H. D. (2022). Semantic segmentation of in-field cotton bolls from the sky using deep convolutional neural networks. *Smart Agricultural Technology*, 2, 100045. <https://doi.org/10.1016/j.atech.2022.100045>
- Stivaktakis, R., Tsagkatakis, G., & Tsakalides, P. (2019). Deep learning for multilabel land cover scene categorization using data augmentation. *IEEE Geoscience and Remote Sensing Letters*, 16(7), 1031–1035. <https://doi.org/10.1109/LGRS.2019.2893306>
- Unnikrishnan, A., Sowmya, V., & Soman, K. P. (2019). Deep learning architectures for land cover classification using red and near-infrared satellite images. *Multimedia Tools and Applications*, 78(13), 18379–18394. <https://doi.org/10.1007/s11042-019-7179-2>
- Vyas, T., Yadav, R., Solanki, C., Darji, R., Desai, S., & Tanwar, S. (2022). Deep learning-based scheme to diagnose Parkinson's disease. *Expert Systems*, 39(3), e12739. <https://doi.org/10.1111/exsy.12739>
- Xu, K., Huang, H., & Deng, P. (2022). Remote sensing image scene classification based on global-local dual-branch structure model. *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5. <https://doi.org/10.1109/JSTARS.2021.3126082>
- Xu, K., Huang, H., Deng, P., & Shi, G. (2020). Two-stream feature aggregation deep neural network for scene classification of remote sensing images. *Information Sciences*, 539, 250–268. <https://doi.org/10.1016/j.ins.2020.06.011>
- Xu, X., Chen, Y., Zhang, J., Chen, Y., Anandhan, P., & Manickam, A. (2021). A novel approach for scene classification from remote sensing images using deep learning methods. *European Journal of Remote Sensing*, 54(sup. 2), 383–395. <https://doi.org/10.1080/22797254.2020.1790995>
- Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems* (pp. 270–279). <https://doi.org/10.1145/1869790.1869829>
- Ye, M., Ruiwen, N., Chang, Z., He, G., Tianli, H., Shijun, L., Yu, S., Tong, Z., & Ying, G. (2021). A lightweight model of VGG-16 for remote sensing image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 6916–6922. <https://doi.org/10.1109/JSTARS.2021.3090085>
- Zalpour, M., Akbarizadeh, G., & Alaei-Sheini, N. (2020). A new approach for oil tank detection using deep learning features with control false alarm rate in high-resolution satellite imagery. *International Journal of Remote Sensing*, 41(6), 2239–2262. <https://doi.org/10.1080/01431161.2019.1685720>
- Zhai, S., Shang, D., Wang, S., & Dong, S. (2020). DF-SSD: An improved SSD object detection algorithm based on densenet and feature fusion. *IEEE Access*, 8, 24344–24357. <https://doi.org/10.1109/ACCESS.2020.2971026>
- Zhang, Y., Song, C., & Zhang, D. (2022). Small-scale aircraft detection in remote sensing images based on faster-RCNN. *Multimedia Tools and Applications*, 81(13), 18091–18103. <https://doi.org/10.1007/s11042-022-12609-5>
- Zhao, L., Zhang, W., & Tang, P. (2019). Analysis of the inter-dataset representation ability of deep features for high spatial resolution remote sensing image scene classification. *Multimedia Tools and Applications*, 78(8), 9667–9689. <https://doi.org/10.1007/s11042-018-6548-6>
- Zhu, D., Xia, S., Zhao, J., Zhou, Y., Niu, Q., Yao, R., & Chen, Y. (2020). Fusion based feature reinforcement component for remote sensing image object detection. *Multimedia Tools and Applications*, 79(47), 34973–34992. <https://doi.org/10.1007/s11042-020-08876-9>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.