Journal of Big Data

SURVEY Open Access

Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis

Adekanmi Adeyinka Adegun[1], Serestina Viriri[1*] and Jules-Raymond Tapamo[2]

*Correspondence:
viriris@ukzn.ac.za

[1] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, South Africa
[2] School of Engineering, University of KwaZulu-Natal, Durban, South Africa

Abstract

Classification and analysis of high-resolution satellite images using conventional techniques have been limited. This is due to the complex characteristics of the imagery. These images are characterized by features such as spectral signatures, complex texture and shape, spatial relationships and temporal changes. In this research, we present the performance evaluation and analysis of deep learning approaches based on Convolutional Neural Networks and vision transformer towards achieving efficient classification of remote sensing satellite images. The CNN-based models explored include ResNet, DenseNet, EfficientNet, VGG and InceptionV3. The models were evaluated on three publicly available EuroSAT, UCMerced-LandUse and NWPU-RESISC45 datasets containing categories of images. The models achieve promising results in accuracy, recall, precision and F1-score. This performance demonstrates the feasibility of Deep Learning approaches in learning the complex and in-homogeneous features of the high-resolution remote sensing images.

Keywords: Satellite images, Remote sensing images, Convolutional neural networks, Vision Transformer, Deep learning, Image classification

Introduction

In recent years, application of remote sensing images dataset has become more relevant in our day-to-day activities. Object detection and image classification from the analysis of multi-temporal high resolution remote sensing satellite imagery has become very useful in real-life applications like environmental monitoring, natural disasters and hazardous events prevention, and terrestrial biodiversity analysis [1, 2]. Analysis of remote sensing images is challenging due to the complex nature of the images [3]. Development of reliable system for the analysis of remote sensing images is therefore very important.

The manual identification and detection of objects and images from remote sensing satellite imagery is arduous and costly [3, 4]. There have been several systems created to detect objects and classify images from remote sensing imagery [4]. Over the years, there have been substantial efforts to incorporate machine learning algorithms into the development of systems for the analysis and classification of images in object detection [3]. Despite advancements in remote sensing tools and object-based image analysis

tools for analyzing high spatial and temporal resolution satellite images, the classification accuracy of complex images remains inadequate. The primary reason for this inadequacy could be due to the high variability in the spectral and spatial features of the images, which complicates the classification of heterogeneous land cover classes.

The images are prone to adversarial conditions such as cloud and solar radiance. To overcome these issues, several approaches that combine both spectral and spatial features in the classification scheme have been proposed in the past. These approaches relied on conventional methods like the Markov Random Field (MRF) model [5], Conditional Random Field (CRF) model [6], and Composite Kernel (CK) methods [7]. However, these models are limited in their ability to extract a vast number of features for supervised classification due to the time-consuming feature engineering process [8], which demands extensive knowledge for the extraction of manageable features. Additionally, classification based on hand-crafted spatial features mainly depends on low-level features, resulting in poor classification outcomes. Furthermore, these models have limited generalization capacity.

In recent times, Deep Learning, a sophisticated tool in the field of machine learning, has demonstrated its effectiveness in the realm of computer vision and subsequently, in remote sensing as well [9]. The conventional machine learning tools such as Support Vector Machine (SVM) and Random Forest (RF) [10] which are shallow-structured, have major limitations that are addressed by these advanced machine learning algorithms. Prominent deep learning models such as Deep Belief Net (DBN) [11], Stacked Auto-Encoder (SAE) [12], and deep Convolutional Neural Network (CNN) [13, 14] have shown promising results in several remote sensing applications, including segmentation, object detection [15], and classification [16]. These models are characterized by deep architecture, multi-layered interconnected channels, and a high capacity to learn features.

Despite the recent advancements in deep learning techniques and their applications in remote sensing, their effectiveness has been largely limited to the classification of high-resolution satellite and aerial imagery due to the scarcity of available datasets for model training and the need for extensive parameter tuning [8]. Recently, vision transformers have been introduced to overcome these limitations by incorporating self-attention mechanisms that enable the modeling of semantic relationships between all pairs of pixels in an image [17]. Nonetheless, the application of these transformers is computationally expensive, and their efficiency decreases exponentially with the size of the image, thereby requiring significant computational resources [18].

In this research, we present deep learning approaches based on Convolutional Neural Networks and state-of-the-art vision transformers for automatic object detection and classification of satellite imagery dataset. We perform some experimental survey and comparative analysis of the deep learning based methods. Vision transformer and CNN based models such as ResNet, DenseNet, EfficientNet, VGG and InceptionV3 have been experimented and evaluated on publicly available EUROSAT, UCMerced-LandUse and NWPU-RESISC45 datasets containing categories of object images. The models achieve promising results in accuracy, recall, precision and F1-score. The performance demonstrates the feasibility of Deep Learning methods to learn the complex and heterogeneous features of the high resolution remote sensing images.

Literature review

Characteristics of remote sensing images

Spectral, temporal, and spatial resolution are major features of remote sensing images and are important parameters to be considered during remote sensing image classification process;

1. Spectral resolution is composed of different wavelengths of electromagnetic radiation.
2. Temporal resolution is the time interval between image acquisitions.
3. Spatial resolution is the size of a pixel on the ground. These parameters play a critical role in identifying different land cover types and monitoring changes in land cover over time.

They are complex features on remote sensing images and efficient system must be able to effectively process them to achieve accurate classification of remote sensing images by focusing on spectral, temporal, and spatial resolution of the images.

There are also other types of remote sensing images based on the nature of the capturing devices. These are categorised into optical, thermal, hyper-spectral, and SAR images:

1. Optical images capture visible and near-infrared regions of the electromagnetic spectrum and are the most commonly used remote sensing data for land cover classification.
2. Thermal images capture the thermal radiation emitted by the Earth's surface and is used to detect temperature variations.
3. Hyper-spectral images capture a wide range of spectral bands with narrow bandwidths, allowing for the identification of more subtle spectral signatures.
4. SAR images use microwave radiation and can penetrate through clouds and vegetation, making them useful in detecting changes in surface features.

Approaches for classification of remote sensing images

There are different approaches for classification of remote sensing images. These include pixel-wise classification and object based classification.

1. Pixel-wise remote sensing image classification is the most commonly used method for remote sensing image classification. This method involves assigning a class label to each pixel in an image based on its spectral signature. Several classification algorithms have been developed for pixel-wise classification, including maximum likelihood, support vector machines, decision trees, and neural networks.
2. Object-based methods: These methods group adjacent pixels together into objects and classify these objects based on their spectral and contextual characteristics. Examples include OBIA (Object-Based Image Analysis) which is a specific type of remote sensing image analysis that uses image segmentation to group adjacent pixels into meaningful objects or regions, which are then classified based on their spectral and contextual features. Object-based methods offer several advantages over

pixel-based methods, including improved accuracy, reduced noise, and the ability to account for spatial context.

The choice of method depends on the specific application and the characteristics of the data being analyzed.Remote sensing image classification is an essential task in remote sensing, and various methods have been developed to improve the accuracy and efficiency of classification. The use of different types of remote sensing data and the development of new classification algorithms have enabled better monitoring and understanding of the Earth's surface.

Related works

In recent times, there have been various deep learning methods employed for the classification of remote sensing images. Bosco et al. [19] proposed a multi-granularity neural network encoding architecture based on pre-trained CNNs like InceptionV3, Inception-ReseNetV2, VGG16, and DenseNet201, with the use of activation functions and ensemble learning to extract features. The model was fine-tuned using InceptionResNetV2 and VGG16 and was evaluated on two public datasets, UCM and SIRI-WHU, as well as another dataset comprising 2112 labeled images collected through the Google Earth engine from East Africa Community Countries (EACC), categorized into nine classes. Mahdianpari et al. [20] also employed deep learning tools based on CNNs for the classification of complex wetland classes in Canada using multispectral RapidEye optical imagery. They explored seven deep convnets, including DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2 for wetland classification and mapping. The models were evaluated and compared with conventional tools, and InceptionResNetV2, ResNet50, and Xception were identified as the top three convnets, providing state-of-the-art classification accuracies for complex remote sensing scenes such as wetlands. The classification accuracies obtained using Support Vector Machine (SVM) and Random Forest (RF) were significantly inferior to those obtained using deep learning methods.

Zhou et al. proposed a ResNet-based architecture, ResNet-TP, which utilizes two pathways, and was tested on two scene classification datasets, UCM Land Use and NWPU-RESISC45, exhibiting significant improvements over the existing state-of-the-art methods [21]. Furthermore, Zhang et al. proposed a fully convolutional network based on DenseNet for remote sensing scene classification, which was compared with various state-of-the-art algorithms on multiple datasets, including UCM, AID, OPTIMAL-31, and NWPU-RESISC45 [22]. To classify objects and facilities into 63 different classes from the IARPA Functional Map of the World (fMoW) dataset, a deep learning system was developed by integrating satellite metadata with image features, employing an ensemble of convolutional neural networks and additional neural networks, achieving an accuracy of % and an F1 score of 0.797 [16]. In another study, Mohanty et al. employed five approaches based on U-Net and Mask R-Convolutional Neuronal Networks models for satellite imagery classification from SpaceNet dataset, using boosting algorithms, morphological filter, Conditional Random Fields, and custom losses, which were modified with training adaptations, achieving an AP of 0.937 and an AR of 0.959 [23].

Alhichri et al. [24] proposed a Deep Convolutional Neural Network (CNN) with an attention mechanism for scene classification in remote sensing. The novel approach computes a new feature map by weighting the original feature maps. The CNN, named EfficientNet-B3-Attn-2, was developed by enhancing the pre-trained EfficientNet-B3 CNN with an attention mechanism. The study demonstrated the effectiveness of the proposed approach on six remote sensing datasets including UC Merced, KSA, OPTIMAL-31, RSSCN7, WHU-RS19, and AID datasets. The results showed the system's strong performance in accurately classifying remote sensing images and scenes. Yang et al. [25] introduced a novel CNN architecture called Multi-Scale Input Spatial Pyramid Pooling Fusion Networks (MSPPF-nets) based on DenseNets for the classification of local climate zones (LCZs). The proposed system utilizes the Spatial Pyramid Pooling (SPP) layer to extract multi-scale features from various channels and fuse them through a multi-branch-input framework. Mu et al. [26] proposed a spectral-spatial classification method for hyperspectral images (HSIs) based on deep adaptive feature fusion (SSDF). The system fuses two types of HSIs features, edge features extracted by guided filter and principal component features extracted by principal component analysis, through deep adaptive fusion. The deep features are then further processed by the long short-term memory (LSTM) model for classification.

Xu et al. [27] proposed an innovative attention-based pyramid network for the classification and segmentation of remote sensing datasets. The study employed three different attention mechanisms, including attention-based multi-scale fusion, region pyramid attention, and cross-scale attention in adaptive atrous spatial pyramid pooling network. These attention mechanisms effectively fused spatial and spectral information at different and same scales, addressed geometric size diversity in large-scale remote sensing images and adapted features to diverse contents in a feature-embedded space. The attention-based modules were integrated with a spatial feature fusion pyramid network (FFPNet) and an end-to-end spatial-spectral FFPNet to establish various feature fusion pyramid frameworks. These frameworks aimed to address the spatial problem of high-resolution remote sensing images and classify hyperspectral images. The proposed system was evaluated on two high-resolution remote sensing datasets, ISPRS Vaihingen and ISPRS Potsdam, and the results demonstrated the effectiveness of the approach. Zhang et al. [28] proposed a novel method for remote sensing scene classification, the Remote Sensing Transformer (TRS), which integrates self-attention into ResNet using a Multi-Head Self-Attention layer in the bottleneck. The study utilized multiple pure Transformer encoders to improve the representation learning performance, completely depending on attention. The TRS model was tested on four public remote sensing scene datasets, namely UC-Merced, AID, NWPU-RESISC45, and OPTIMAL-31, and the results showed higher accuracy.

Some attempts have been made to use deep learning models for the analysis of thermal infrared (TIR) remote sensing images. These models also achieved promising results on thermal infrared imagery. For example, Jiang et al. [29] proposed YOLO models for extracting features from a ground based TIR remote sensing images. The research identified YOLOv5-s as the most effective algorithm with the highest mAP of person and car instances at 88.69% and fastest detection speed of 50 FPS. Masouleh et al. [30] also proposed an improved deep learning model based on encoder-decoder structure of

convolutional layers and restricted Boltzmann machine for extracting features from UAV-based thermal infrared imagery. They achieved average precision and average processing time of 0.97 and 19.73 s.

Finally, an onboard real-time object detection system for remote sensing images, named MSF-SNET, was proposed by Huang et al. [31]. The system is a lightweight one-stage detector that uses SNET as the backbone network to reduce computational complexity and the number of parameters. The system extracts three low-level features from the three stages of SNET and further extracts deep features using three convolutional layers to obtain semantic information for large-scale object detection. The deep and low-level features are fused to enhance feature representation. The system was evaluated on publicly available NWPU VHR-10 dataset and DIOR dataset. Despite the recent advancements in deep CNN-based architectures for remote sensing image analysis and classification as discussed in this section, their applications are still limited, especially in the classification of very high resolution aerial and satellite imagery. Current research has primarily focused on urban area classification using CNN-based architectures, with limited exploration of state-of-the-art classification tools such as vision transformers for complex high resolution land cover mapping of satellite imagery. Complex land cover imagery, such as forests, vegetation, crops, pastures, rivers, highways, and residential areas, pose challenges for low classification performance and insufficient object detection accuracy due to their high intra-class variance, multi-resolution, multi-spectra, and heterogeneity.

In this research, we perform robust comparative analysis and evaluation of deep learning approaches based on Convolutional Neural Networks and state-of-the-art vision transformers for automatic, data-driven and intelligence based object detection and classification of satellite imagery dataset. We investigate the capability of well-known deep CNNs in the analysis of high resolution remote sensing images. The main contributions of this study are therefore to:

1. Analyze the effectiveness of deep learning models in classifying satellite imagery with varying resolutions and spectral bands.
2. Investigate the impact of fine-tuning on improving the performance of CNN-based deep learning models for high-resolution satellite image classification.
3. Compare and contrast the performance of several widely used deep CNNs, such as DenseNet121, InceptionV3, VGG16, Efficient-Net, ResNet50, and vision transformer, on three distinct publicly available remote sensing datasets comprising of satellite images: EuroSAT, UCMerced-LandUse, and NWPU-RESISC45.

Thus, this study contributes to the use of deep learning based classification tools for complex high resolution remote sensing satellite imagery and further open up research in the application of state-of-the-art for the analysis of these images.

Methods and techniques

Various deep learning architectures have been used in the past for computer vision tasks, most especially in the analysis and classification of remote sensing images. In this section, we explore five (5) CNN-based deep learning architectures and a vision transformer based architecture.

CNN-based architectures

We discuss five (5) CNN-based deep learning architectures. They are VGG16, ResNet50, IncetionV3, EfficientNet and DenseNet121. The composition of each architecture has been discussed below:

ResNet

The ResNet system employs deep residual networks to improve the classification performance by reducing the vanishing gradient problems of deeper network through a residual process. The ResNet architecture [32] adopts residual learning to every few stacked layers. The architecture also leverages on stacking convolutional layers for learning and extracting features. The ResNet model employed in this research is composed of five blocks, with each block having the same size of convolutional layer except the first block that performs down-sampling. Basically, each block is composed of a composite function comprising batch normalization (BN), a non-linear transformation unit, rectified linear unit function (ReLU) and a Convolution layer. A skip-connection is used that bypasses the non-linear transformations with an identity function. Deep features are extracted and down-sampled using integrated pooling units of Maxpool, AdaptiveAvgPool, and AdaptiveMaxPool.

The operation is defined as:

$$y = F(x, W_i) + x \tag{1}$$

where we consider the input and output vectors of a layer as x and y respectively, then the function $F(x, W_i)$ denotes the residual mapping that needs to be learned through various convolutional layers and operators. Following this, the addition of feature maps takes place element-wise, channel by channel.

Composition of the blocks is further described in the Table 1 where n is the number of the block with the same composition, F are the operators and the resolutions which represent the sizes are denoted with H and W. These are further described in Fig. 1 showing the detailed layout diagram of the ResNet50 architecture.

The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 44,611,648
2. Total trainable parameters: 2,216,832
3. Total non-trainable parameters: 42,394,816

Table 1 The summary Of ResNet50 layered architecture. Architecture

Block	Units (n)	Operators (F)	Resolutions ($H \times W$)	Channels
Block A	1	Conv, ReLu, MaxPool, BatchNorm	112 × 112	64
Block B	15	Conv, ReLu, BatchNorm	56 × 56, 28 × 28, 14 × 14, 7 × 7	256, . . . , 2048
Block C	10	Conv, BatchNorm	56 × 56, 28 × 28, 14 × 14, 7 × 7	64, 128, ..., 2048
Block D	12	Conv, Relu, BatchNorm, Adaptive-MaxPool, AdaptAvgPool	7 × 7	2048
Block E	1	Linear, Relu, BatchNorm	7 × 7	512
Block F	1	Linear	7 × 7	10

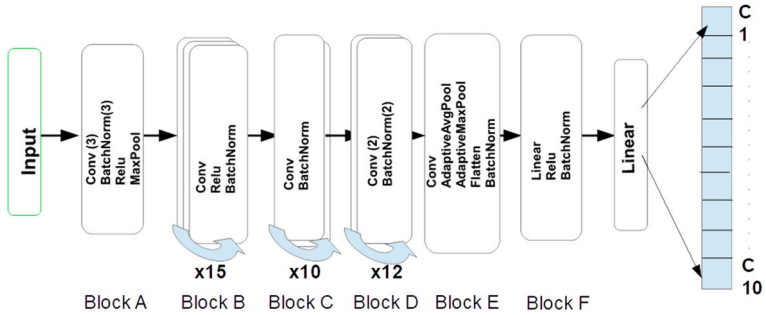

Fig. 1 The basic architectural diagram for Resnet50 model design

VGG16

VGG architecture [33] improves on the basic ConvNet architecture by steadily increasing the depth of the network through the addition of convolutional layers. As a result, the model becomes significantly more accurate. The input images are passed through a stack of convolutional (convs) layers of various sizes. This is followed by non-linearity ReLu activation function, batch normalization unit and pooling units such as average pooling, maximum pooling, adaptive average pooling, and adaptive maximum pooling. The pooling layers are utilized to preserve the spatial resolution after the convolution process and are executed over a 2×2 pixel window.VGG16 model is composed of a set of blocks as shown in Fig. 2. The basic ConvNet applies Eq. (2) for features extraction:

$$F_i = ReLU(W \times F_{i-1} + b_i) \tag{2}$$

where F_i refers to the feature map of the current layer, F_{i-1} refers to the feature map of the previous layer, W is the filter kernel, and b_i is the bias added to the feature map of each layer. The rectified linear unit (ReLU) activation function is defined as:

$$U(y) = \max(0, y) = \begin{cases} y \text{ if } y \geq 0 \\ 0 \text{ if } y < 0 \end{cases}$$

where y is the resulting feature map.

The composition of the blocks is further described in the Table 2 where n is the number of blocks with the same composition, F are the operators and the resolutions which represent the sizes are denoted with H and W. The detailed layout diagram of VGG16 architecture is further described in Fig. 2.

The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 11,117,632
2. Total trainable parameters: 532,480
3. Total non-trainable parameters: 10,585,152

EfficientNet

EfficientNet architecture [34] performs compound Scaling of ConvNets by scaling all three dimensions—depth (number of layers), width (number of channels) or image

Table 2 The summary Of VGG16 layered architecture. Architecture

Block	Units (*n*)	Operators (*F*)	Resolutions ($H \times W$)	Channels (*C*)
Block A	2	Conv, ReLu, MaxPool	224 × 224	64
Block B	112	Conv, ReLu, BatchNorm	112 × 112	128
Block C	1	Conv, ReLu, BatchNorm, AvgPool	56 × 56	256
Block D	1	Conv, ReLu, BatchNorm, AvgPool	56 × 56	256
Block E	1	Conv, ReLu, AvgPool, AdaptiveAvg-Pool, AdaptiveMaxpool	28 × 28	512
Block F	1	Linear, Relu, BatchNorm	28 × 28	512

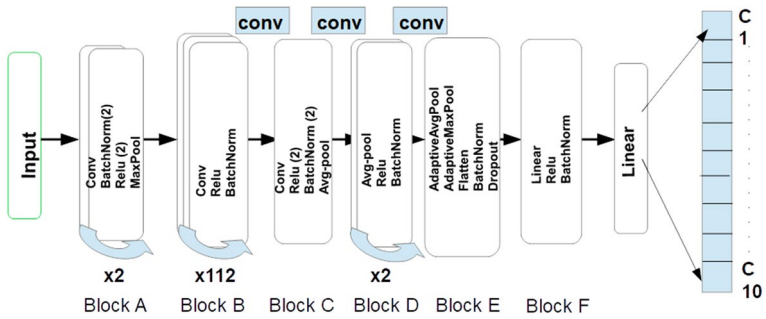

Fig. 2 The basic architectural diagram for VGG16 model design

resolution (image size)and still maintains a balance between the three dimensions of the network by scaling up ConvNets. Each block in EfficientNet comprises batch normalization (BN), followed by a Swish activation function, pooling and Convolution layers. Deep features from the network are extracted and down-sampled using integrated pooling units. The architecture is composed of six main blocks.

The composition of the blocks is further described in the Table 3 where *n* is the number of blocks with the same composition, F are the operators and the resolutions which represent the sizes are denoted with H and W. These are further described in Fig. 3 showing the detailed layout diagram of the EfficientNet architecture. The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 5,329,532
2. Total trainable parameters: 1,364,000
3. Total non-trainable parameters: 3,965,532

Inceptionv3

The inception model utilizes multiple Inception layers to achieve a reliable classification performance. Inception Modules allow for more efficient computation and deeper networks through a dimensionality reduction with stacked 1×1 convolutions [35]. The Inception-v3 model is designed for multiscale approach, as it increases both the width and depth of the network. This ensures that the vanishing gradient problems is minimized while also creating deeper architectures for efficient classification of the remote sensing satellite

Table 3 The summary of EfficientNet layered architecture. Architecture

Block	Units (*n*)	Operators (*F*)	Resolutions($H \times W$)	Channels
Block A	12	Conv, BatchNorm, Swish, AdaptiveAvg-Pool	112×112, 56×56, 28×28, 14×14, 7×7	32,96,144,
Block B	4	Conv, BatchNorm, Swish, AdaptiveAvg-Pool	112×112, 56×56, 28×28, 14×14, 7×7	96,144,240
Block C	4	Conv, BatchNorm, Swish	112×112, 56×56, 28×28, 14×14, 7×7	96,144,240
Block D	16	Conv, Swish	1×1	8,4,6
Block E	16	Conv, Sigmoid	1×1	32,96,144
Block F	16	Conv, BatchNorm	112×112, 56×56, 28×28, 14×14, 7×7	24,40,80
Block G	1	Linear, Relu, BatchNorm	7×7	512

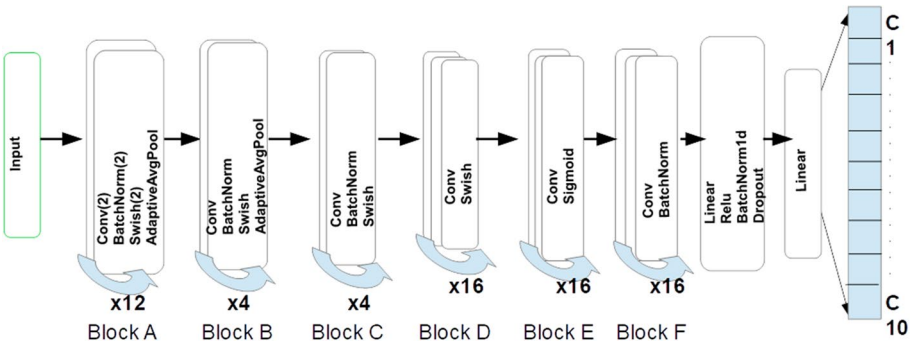

Fig. 3 The basic architectural diagram for EfficientNet model design

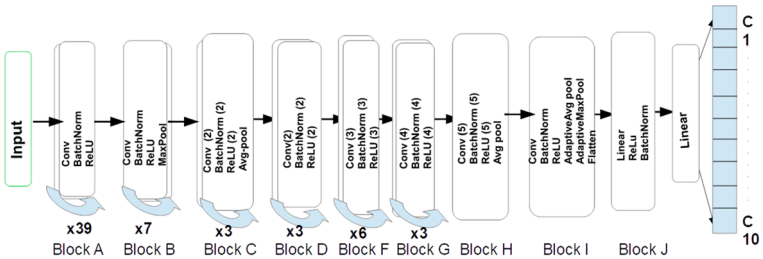

Fig. 4 The basic architectural diagram for InceptionV3 model design

images. The deep features from the network are extracted and down-sampled using the integrated pooling system. Figure 4 shows a basic architecture diagram of Inception-v3.

The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 23,897,056
2. Total trainable parameters: 2,145,920
3. Total non-trainable parameters: 21,751,136

Table 4 Basic composition of the functional operators in each block that constitute the DenseNet121 Architecture

Block	Units (*n*)	Operators (*F*)	Resolutions ($H \times W$)	Channels
Block A	2	Conv, ReLu, MaxPool, BatchNorm	112×112	64
Block B	112	Conv, ReLu, BatchNorm	56×56, 28×28,14×14, 7×7	128,32
Block C	1	Conv, ReLu, BatchNorm	56×56	32
Block D	1	Conv, Relu, MaxPool, BatchNorm, AvgPool	56×56	128
Block E	2	Conv, AvgPool, BatchNorm, Relu	28×28	256
Block F	1	Conv, BatchNorm, AdaptiveMaxPool, Linear, Relu, BatchNorm	7×7	–
Block G	1	Linear, Relu, BatchNorm	7×7	512

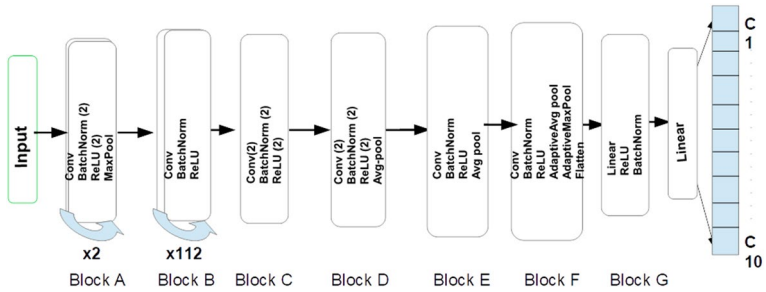

Fig. 5 The basic architectural diagram for DenseNet121 model design

DenseNet

DenseNets, as proposed in [36], aim to leverage feature reuse in the network to obtain compact and highly parameter efficient models that are easy to train. The network consists of a series of dense blocks, and to enhance computational efficiency within each block, a 1×1 convolution layer is introduced before every 3×3 convolution layer. This reduces the number of input feature maps, which are typically more than the output feature maps.

The process of concatenation and the dense blocks are described by the following equation:

$$y = C_n([x_0, x_1, \ldots, x_{n-1}]) \tag{3}$$

where $x_0 \ldots x_{n-1}$ refers to the concatenation of input feature maps from the convolutional operators C_n. The composition of the blocks is further described in the Table 4 where n is the number of blocks with the same composition, F are the operators and the resolutions which represent the sizes are denoted with H and W. These are further described in Fig. 5 showing the detailed layout diagram of the DenseNet121 architecture.

The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 8,012,672
2. Total trainable parameters: 1,142,464
3. Total non-trainable parameters: 6,870,208

Vision Transformer-based architecture

Vision transformers employ multi-head attention mechanism, which provides both local and global context for effective extraction of multi-scale, multi-resolution and high-level spatial features. The dense feature maps generated are then up-sampled and concatenated using the global average pooling system. The method leverages on both local and global attention, along with global average pooling, for efficient learning and extraction of the complex features in remote sensing satellite images. The entire system is composed of processes such as flattening, tokenization, position embedding, and classification as shown in Fig. 6. Specifically, the input image is divided into fixed-size patches, flattened and linearly embedded, added to position embedding, and sent into the Transformer encoder.

The composition of the parameters in this model are enumerated below:

1. Total number of parameters: 4,166,151
2. Total trainable parameters: 4,166,151
3. Total non-trainable parameters: 0

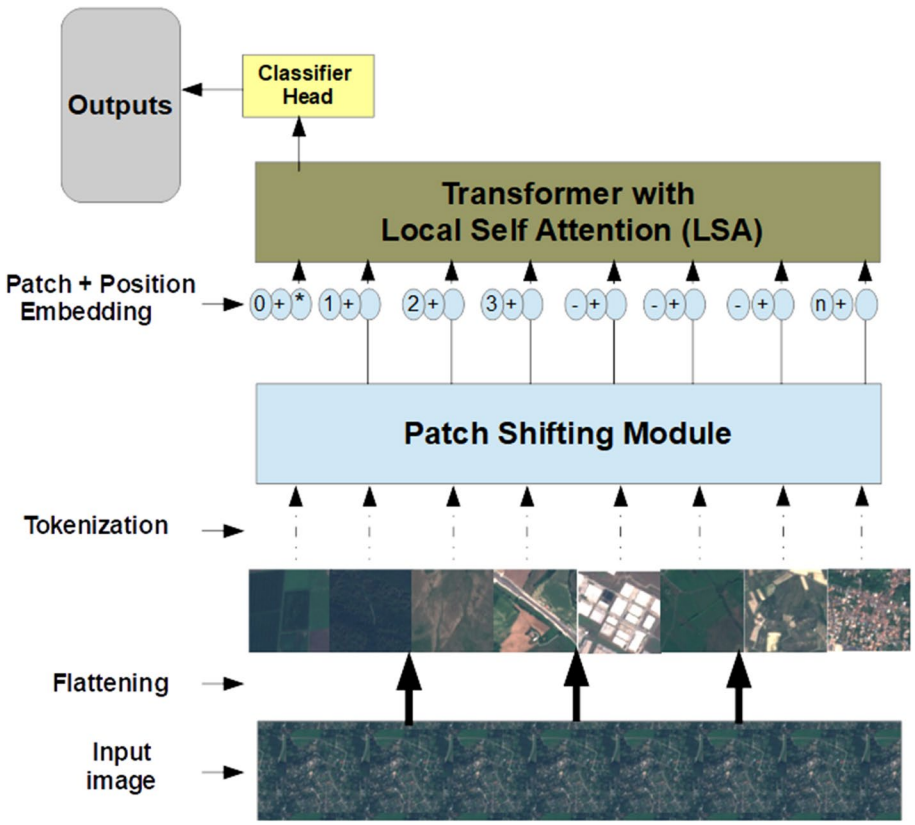

Fig. 6 The basic layout diagram for Vision Transformer design

Experiments and results

In this section, we conducted a series of experiments to assess the performance of the models. The results of the evaluations are presented and elaborated below, and are compared with other similar approaches.

Datasets

EuroSAT

The EuroSAT benchmarking datasets [37] utilized in this study consist of 27,000 labeled images across ten classes. Each class comprises 2,000–3,000 images, with a size of 224×224 pixels. It contains classes, including annual crops, forest, herbaceous vegetation, highway, industrial, pasture, permanent crops, residential, river, and sea lakes. Figure 7 shows some sample images from the dataset.

UCMerced-LandUse

The UC Merced Land-Use dataset [38] comprises 2100 aerial scene satellite images, categorized into 21 land use scene classes, with each class containing 100 images of dimensions 256×256 pixels.

NWPU-RESISC45

The NWPU-RESISC45 dataset [39] consists of 31,500 remote sensing satellite images categorized into 45 scene classes. Each class comprises 700 images of dimensions 256×256 pixels.

Performance metrics

Typically, the performance of deep learning models for image classification is assessed using standard metrics such as Accuracy, Recall, Precision, and F1-score. These metrics are defined as follows:

Accuracy measures the proportion of correct predictions out of the total number of cases examined. It is calculated using the equation:

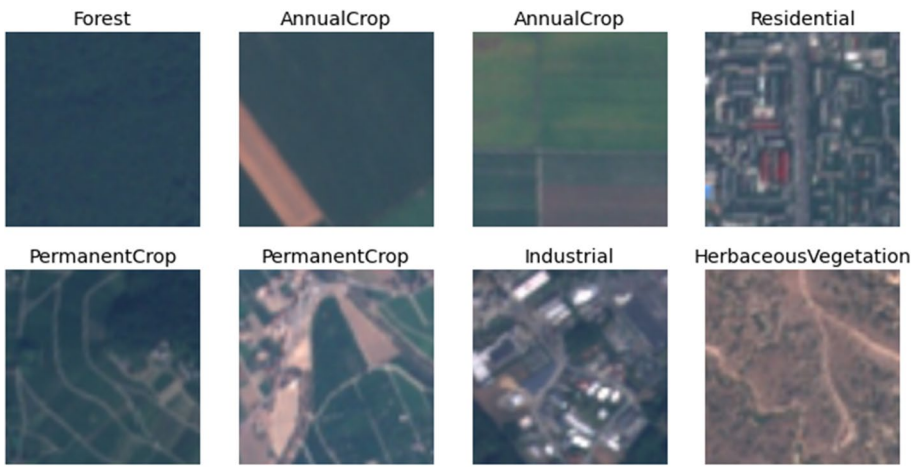

Fig. 7 Sample images from EuroSAT dataset

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

F1-score is the weighted average of Precision and Recall, providing a better assessment of incorrectly classified cases. It is defined as:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{TP}{TP + 1/2(FP + FN)} \tag{5}$$

Recall measures the proportion of truly positive cases out of all actual positive cases. It is also known as sensitivity and is calculated using the equation:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Precision measures the proportion of true positive cases out of all predicted positive cases. It is calculated using the equation:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Confusion Matrix provides a detailed analysis of the results by reporting the number of false positives, false negatives, true positives, and true negatives. It shows the combinations of predicted and true classes for a test dataset, using variables such as *FP* (false positive), *FN* (false negative), *TP* (true positive), and *TN* (true negative).

Results and discussion

In this study, we conducted several experiments to assess the performance of five CNN-based deep learning frameworks and the vision transformer, using these standard metrics on three publicly available datasets: EuroSAT, UCMerced-LandUse, and NWPU-RESISC45. The results of these experiments are presented and discussed in detail in the following sections.

Models performance on EuroSAT dataset

The results from the evaluation of all the models on EuroSAT dataset as represented in Table 5, Figs. 8, 9 and 10 clearly show detailed comparison of the performance of the model. By examining the result as represented in Table 5, using classification metrics such as accuracy, recall, precision and F1-score, two of the CNN based architectures; DenseNet121 and ResNet101 perform excellently with more 90% score in all the evaluation metrics. The vision transformer also performs at par with these two models.

Table 5 Performance analysis (%) and comparison of the deep learning methods on EuroSAT Dataset

Methods	Accuracy	Precision	Recall	F1Score
DenseNet121	98	98	98	98
ResNet101	98	98	98	98
InceptionV3	75	75	75	75
EfficientNet	65	66	65	65
VGG16	79	79	79	79

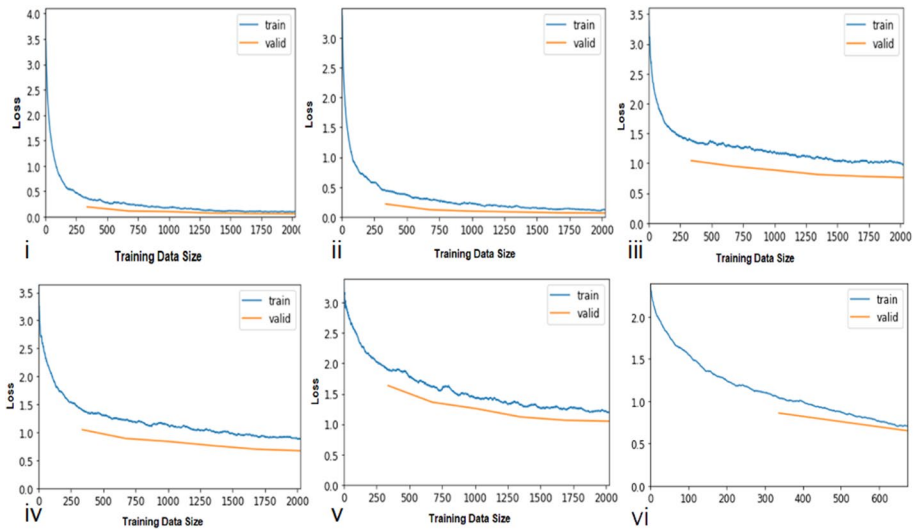

Fig. 8 The figure shows the training loss curve diagrams of deep learning models on EuroSAT dataset: (i) represents the training loss curve for DenseNet121 model; (ii) represents the training loss curve for ResNet101 model; (iii) represents the training loss curve for InceptionV3 model; (iv) represents the training loss curve for VGG16 model; (v) represents the training loss curve for EfficientNetV1 model; (vi) represents the training loss curve for Vision Transformer

However, three other CNN based architectures; InceptionV3, EfficientNet, VGG16 only produce average performance. These three models seem to require larger training dataset to improve their performance. This can also be inferred from the loss curves in Fig. 8. The loss curves for DenseNet121, ResNet101 and vision transformer show that these models achieve very low score of less than 5% and also display stable loss throughout the training process.

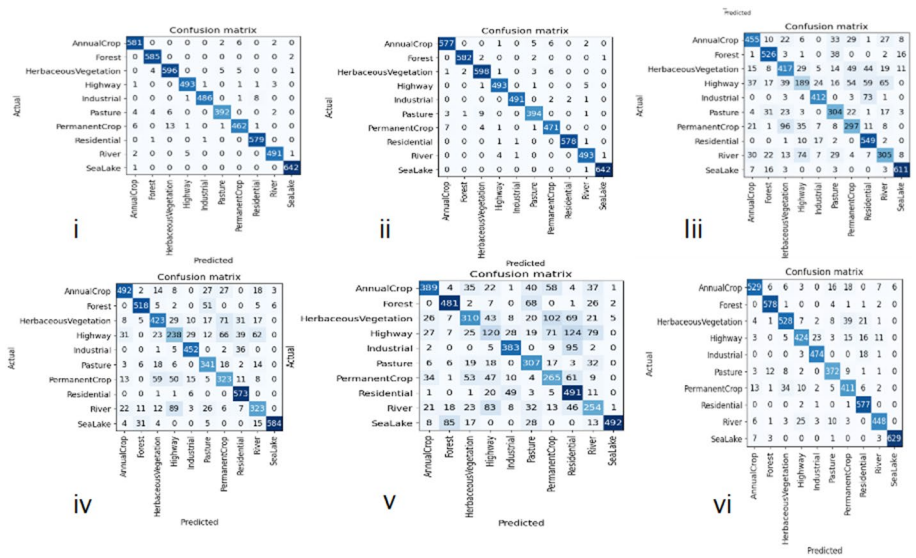

Fig. 9 The figure shows the confusion matrix diagrams of deep learning models on EuroSAT dataset: (i) represents the confusion matrix for DenseNet121 model; (ii) represents the confusion matrix for ResNet101 model; (iii) represents the confusion matrix for InceptionV3 model; (iv) represents the confusion matrix for VGG16 model; (v) represents the confusion matrix for EfficientNetV1 model; (vi) represents the confusion matrix for Vision Transformer

Fig. 10 The figure shows testing sample classification output diagrams of deep learning models on EuroSAT dataset: (i) represents the Classification output for DenseNet121 model; (ii) represents the Classification output for ResNet101 model; (iii) represents the Classification output for InceptionV3 model; (iv) represents the Classification output for VGG16 model; (v) represents the Classification output for EfficientNetV1 model; (vi) represents the Classification output for Vision Transformer

Figures 9 and 10 display the classification output performance of the model when tested on test samples. The confusion matrixes in Fig. 9 for DenseNet121 and ResNet101 show that most of the classes are correctly classified while misclassification rates are higher in InceptionV3, EfficientNet, and VGG16. The vision transformer also performs at par with DenseNet121 and ResNet101 in this case. The results from the confusion matrixes also collaborates the results as discussed earlier. This is also seen from the classification output from Fig. 10. DenseNet121 and ResNet50 models predict and detect the 12 test sample images correctly without missing anyone.

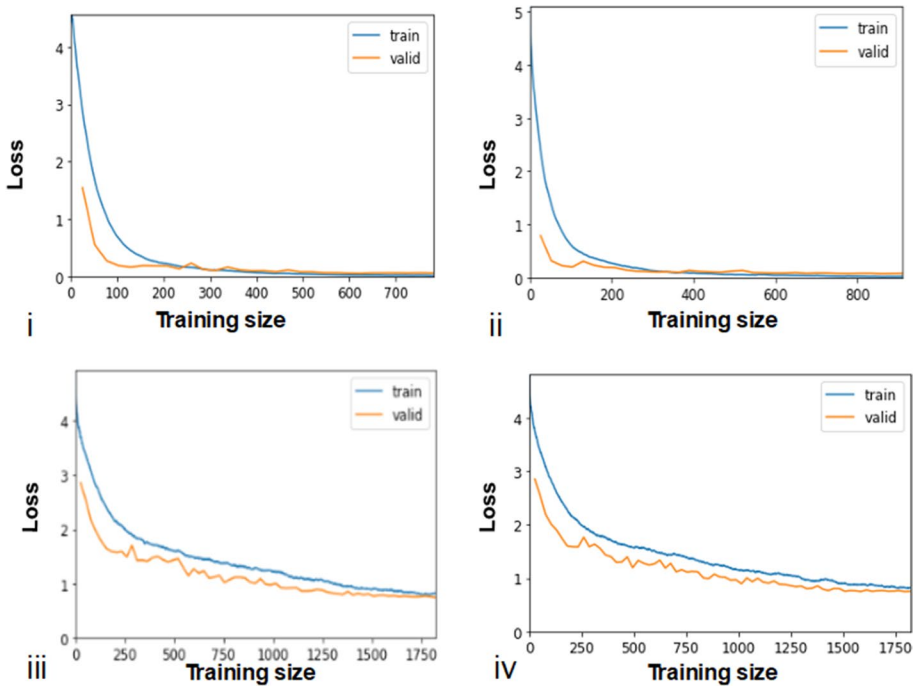

Fig. 11 The figure shows the training loss curve diagrams of deep learning models on UCMerced-LandUse dataset: (i) represents the training loss curve for DenseNet121 model; (ii) represents the training loss curve for ResNet101 model; (iii) represents the training loss curve for InceptionV3 model; (iv) represents the training loss curve for VGG16 model

Models performance comparison on UCMerced-LandUse dataset

Table 6, Figs. 11, 12, and 13 present the evaluation results of DenseNet121, ResNet101, InceptionV3, and VGG16 models on the UCMerced dataset. The results, analyzed using classification metrics such as accuracy, recall, precision, and F1-score, reveal that DenseNet121 and ResNet101, both CNN-based architectures, perform exceptionally well on UCMerced dataset, achieving scores of more than 90% in all evaluation metrics. The performance of InceptionV3, and VGG16 has also improved on UCMerced dataset due to the size of the training dataset. This inference is supported by the loss curves in Fig. 11, which depict that DenseNet121, and ResNet101, achieve lower scores of less than 2%. Figure 12 and Fig. 13 demonstrate the classification output performance of the models when tested on test samples. The confusion matrixes in Figure 12 for DenseNet121 and ResNet101 demonstrate that most of the classes are correctly classified, with higher misclassification rates observed in InceptionV3 and VGG16. These results are further supported by the classification output shown in Fig. 13, where DenseNet121 and ResNet101 accurately predict and detect all 12 test sample images, without missing any.

Models performance comparison on NWPU-RESISC45 dataset

Table 7, Figs. 14, 15, and 16 present the evaluation results of various models on the NWPU-RESISC45 dataset, providing a detailed performance comparison. The results, analyzed using classification metrics such as accuracy, recall, precision, and F1-score, reveal that DenseNet121 and ResNet101, both CNN-based architectures, perform

Table 6 Performance analysis (%) and comparison of the deep learning methods on UCMerced-LandUse dataset

Methods	Accuracy	Precision	Recall	F1Score
DenseNet121	98	98	98	98
ResNet101	98	98	98	98
InceptionV3	74	74	73	72
VGG16	74	74	74	73

exceptionally well, achieving scores of more than 90% in all evaluation metrics. However, InceptionV3 and VGG16, exhibit only average performance, suggesting that they may require larger training datasets to improve their results. This inference is supported by the loss curves in Fig. 14, which depict that DenseNet121, and ResNet101 achieve low scores of less than 2% and exhibit stable loss during the training process, indicating that they do not over-fit and can generalize well. Figures 15 and 16 demonstrate the classification output performance of the models when tested on test samples. The confusion matrixes in Fig. 15 for DenseNet121 and ResNet101 demonstrate that most of the classes are correctly classified, with higher misclassification rates observed in InceptionV3, and VGG16. These results align with those discussed earlier and are further supported by the classification output shown in Fig. 16, where DenseNet121 and ResNet101 accurately predict and detect all 12 test sample images, without missing any.

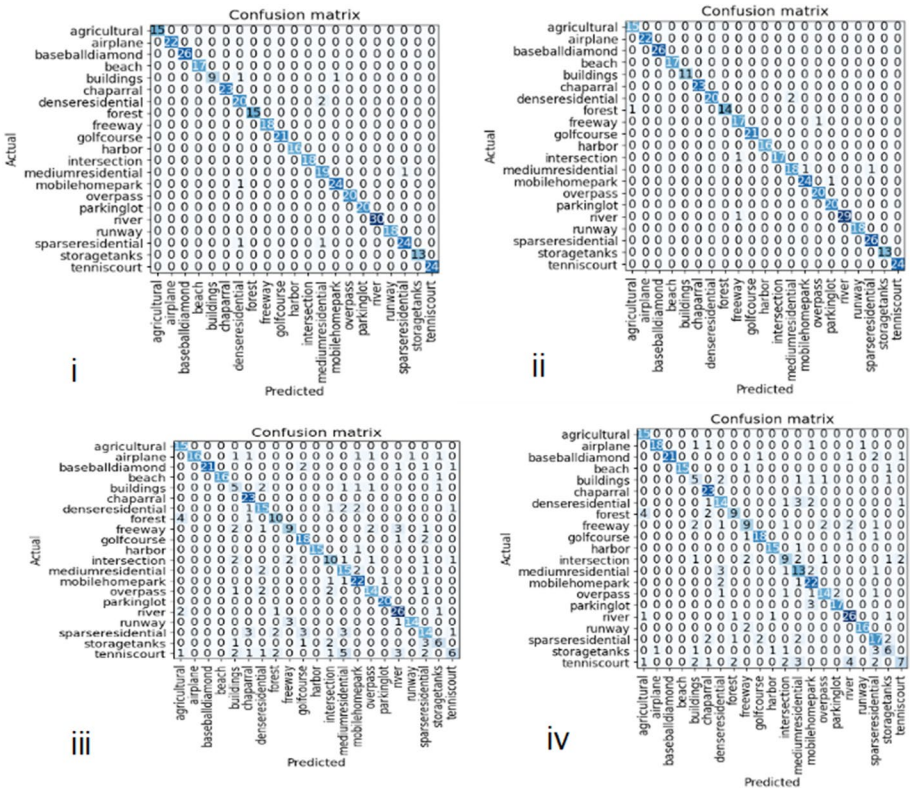

Fig. 12 The figure shows the confusion matrix diagrams of deep learning models on UCMerced-LandUse dataset: (i) represents the confusion matrix for DenseNet121 model; (ii) represents the confusion matrix for ResNet101 model; (iii) represents the confusion matrix for InceptionV3 model; (iv) represents the confusion matrix for VGG16 model

Summary

Important factors identified from the research

In choosing the best convolutional neural network (CNN) for remote sensing images classification, some factors have been identified from the experiments carried out. These include size of dataset, complexity of model complexity and availability of computational resources. These factors have varying effects on the performance of all the models evaluated on the three remote sensing dataset used for experiments: 1. Size of the dataset: To achieve improved performance of the models, larger datasets are required which in turn require deeper and more complex models to capture the variety of features. When the dataset is small, complex model with deeper networks tend to experience over-fitting. 2. Model Complexity: It has been established in this research that a deeper CNN with more convolutional layers such as DenseNet121 and ResNet101 achieve better performance than the shallower CNN. This accounts for the good performance of DenseNet121 and ResNet101. Datasets with complex features require a deeper CNN with more convolutional layers for effective analysis. 3. Computational resources: Deeper and more complex models require more computational resources for training and inference. Therefore,

Fig. 13 The figure shows testing sample classification output diagrams of deep learning models on UCMerced-LandUse dataset: (i) represents the Classification output for DenseNet121 model; (ii) represents the Classification output for ResNet101 model; (iii) represents the Classification output for InceptionV3 model; (iv) represents the Classification output for VGG16 model

Table 7 Performance analysis (%) and comparison of the deep learning methods on NWPU-RESISC45 Dataset

Methods	Accuracy	Precision	Recall	F1Score
DenseNet121	98	98	98	98
ResNet101	98	98	98	98
InceptionV3	68	71	70	70
VGG16	70	71	70	69

the choice of CNN should also consider the available computational resources, such as the CPU, GPU, and memory. In this research GPU systems have been employed.

It has been established in this research that pre-trained deeper CNN models with more convolutional layers such as DenseNet121 and ResNet101 achieve better performance in the analysis of remote sensing images on the three benchmark datasets examined in this research. The vision transformer also performs at the same level with them but require more computational resources. In the future, more works will be done to establish the appropriate database size and the complexity level of models required for efficient analysis of remote sensing images. We propose the combination of attention mechanisms based models such as vision transformer with CNN based deep learning models. Larger datasets require deeper and more complex models to capture the variety of features. Consideration should also be put on the complexity of the dataset as different remote sensing classes have varying levels of complexity. A class with many complex features requires a deeper CNN with more convolutional layers for efficient classification.

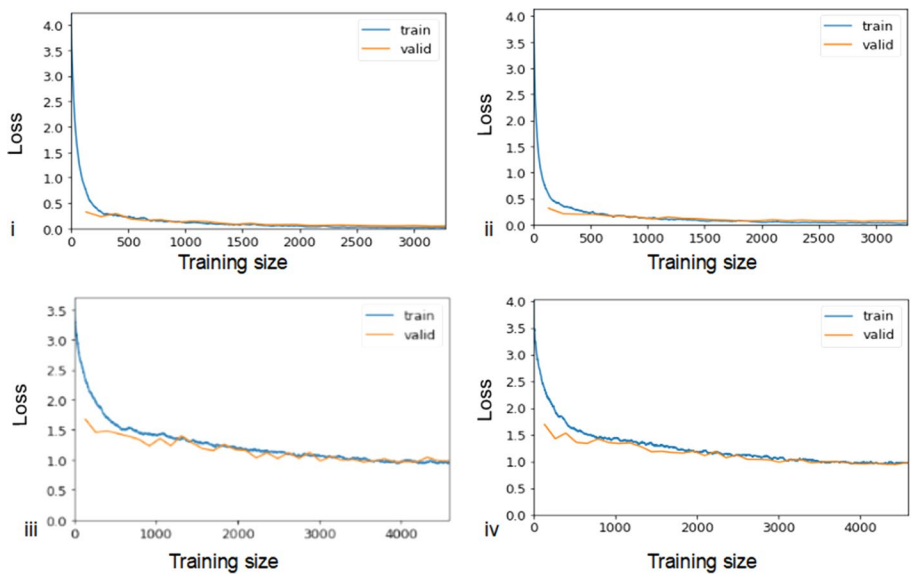

Fig. 14 The figure shows the training loss curve diagrams of deep learning models on NWPU-RESISC45 dataset: (i) represents the training loss curve for DenseNet121 model; (ii) represents the training loss curve for ResNet101 model; (iii) represents the training loss curve for InceptionV3 model; (iv) represents the training loss curve for VGG16 model

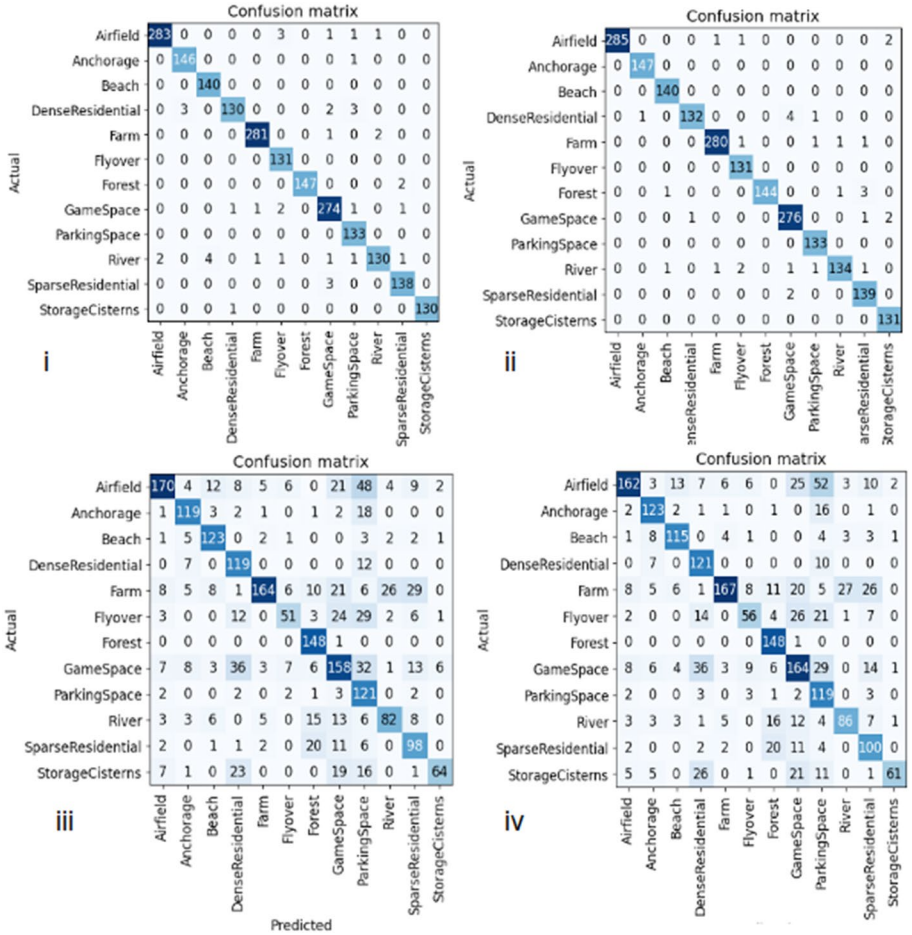

Fig. 15 The figure shows the confusion matrix diagrams of deep learning models on NWPU-RESISC45 dataset: (i) represents the confusion matrix for DenseNet121 model; (ii) represents the confusion matrix for ResNet101 model; (iii) represents the confusion matrix for InceptionV3 model; (iv) represents the confusion matrix for VGG16 model

Evaluation of visual results on some important challenges for remote sensing image classification

Challenging complex features in remote sensing images include spectral signatures, texture, shape, spatial relationships and temporal changes. These features affect the visual results of the models when evaluated. In this research, InceptionV3 and VGG16 models are unable to classify some images correctly due to their shape and texture that look similar. For example, these models misclassified sealake as Annual crop, river as forest and forest as pasture due to similar textures when evaluated on EuroSAT dataset as shown in Fig. 10, They also misclassified intersection as freeway, freeway as river, buildings as overpass and golfcourse as sparse residential due to shapes and spatial relationships on UCMerced-LandUse dataset as shown in Fig. 13.

Models sensitivity towards outliers

The sensitivity of a remote sensing image classification algorithm towards outliers can significantly affect its accuracy. Outliers are data points that deviate significantly from

Fig. 16 The figure shows testing sample classification output diagrams of deep learning models on NWPU-RESISC45 dataset: (i) represents the Classification output for DenseNet121 model; (ii) represents the Classification output for ResNet101 model; (iii) represents the Classification output for InceptionV3 model; (iv) represents the Classification output for VGG16 model

the majority of the data and can arise due to various reasons such as noise, measurement errors, or physical phenomena. Our analysis shows that Densenet121 and Resnet101 models are moderately sensitive to outliers. The overall accuracy of the classifiers on the dataset is 98%, while the overall accuracy on the outlier-contaminated dataset drops to 95%. This indicates that the presence of outliers in the dataset can lead to a reduction in classification accuracy. Furthermore, most of the sample images tested using Desnet121 and Resnet101 models are correctly classified. For the VGG16 and InceptionV3, the models exhibit greater sensitivity to outliers. The overall accuracy of the classifiers on the dataset is 68%, while the overall accuracy on the outlier-contaminated dataset drops to 60%. Some of the sample images tested (3 out of 10) using these models are incorrectly classified.

Conclusion

In this research, we have performed experimental analysis and comparison of some deep learning based architectures on three publicly available remote sensing satellite images, EuroSAT, UCMerced-LandUse and NWPU-RESISC45 datasets. The research shows that the models generally perform well with better optimization on high resolution remote sensing satellite. Experiments show that models that are based on deeper CNN with more convolutional layers are able to efficiently overcome the challenges such as heterogeneous appearance of remote sensing satellite images.

Acknowledgements
Not Applicable.

Author Contributions
A.A. conducted the experiments, coding, and manuscript draft writing. S.V. defined the research problem, validated the results, and proofread the manuscript. J.T. validated the research problem, suggested the techniques, and proofread the manuscript. All authors read and approved the final manuscript.

Funding
Not Applicable.

Availability of data and materials
All data used is publicly available.

Declarations

Ethics approval and consent to participate
Yes, consent is granted.

Consent for publication
Yes, consent is granted for publication.

Conflict of interest
The authors declare that they have no competing interests.

Received: 6 July 2022 Accepted: 17 May 2023
Published online: 02 June 2023

References

1. Phiri D, Simwanda M, Salekin S, Nyirenda VR, Murayama Y, Ranagalage M. Sentinel-2 data for land cover/use mapping: a review. Remote Sens. 2020;12(14):2291.
2. Van Westen CJ. Remote sensing for natural disaster management. Int Archiv Photogrammetry Remote Sens. 2000;33(B7/4; PART 7):1609–17.
3. Cheng G, Han J, Guo L, Qian X, Zhou P, Yao X, Xintao H. Object detection in remote sensing imagery using a discriminatively trained mixture model. ISPRS J Photogramm Remote Sens. 2013;85:32–43.
4. Timberlynn W. Deep convolutional neural networks for remote sensing investigation of looting of the archeological site of Al-Lisht, Egypt. PhD dissertation, University of Southern California; 2018.
5. Jackson Q, Landgrebe DA. Adaptive Bayesian contextual classification based on Markov random fields. IEEE Trans Geosci Remote Sens. 2002;40(11):2454–63.
6. Zhong P, Wang R. Learning conditional random fields for classification of hyperspectral images. IEEE Trans Image Process. 2010;19(7):1890–907.
7. Camps-Valls G, Gomez-Chova L, Muñoz-Marí J, Vila-Francés J, Calpe-Maravilla J. Composite kernels for hyperspectral image classification. IEEE Geosci Remote Sens Lett. 2006;3(1):93–7.
8. Sarker IH. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci. 2021;2(6):1–20.
9. Yann LC, Yoshua B, Geoffrey H. Deep learning. Nature. 2015;521(7553):436–44.
10. Ball JE, Anderson DT, Chan CS. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. J Appl Remote Sens. 2017;11(4): 042609.
11. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527–54.
12. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res. 2010;11:12.
13. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 2012 (p. 25).
14. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015 (pp. 1–9).

15. Chen X, Xiang S, Liu C-L, Pan C-H. Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci Remote Sens Lett. 2014;11(10):1797–801.
16. Pritt M, Chern G. Satellite image classification with deep learning. In 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). IEEE. 2017 (pp. 1–7).
17. Yang Y, Licheng J, Xu L, Fang L, Shuyuan Y, Zhixi F, Xu T. Transformers meet visual learning understanding: a comprehensive review. (2022). arXiv preprint arXiv:2203.12944.
18. Lin T, Wang Y, Liu X, Qiu X. A survey of transformers. 2021. arXiv preprint arXiv:2106.04554
19. Bosco JM, Wang G, Hategekimana Y. Learning multi-granularity neural network encoding image classification using DCNNs for Easter Africa Community Countries. IEEE Access. 2021;9:146703–18.
20. Mahdianpari M, Salehi B, Rezaee M, Mohammadimanesh F, Zhang Y. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sens. 2018;10(7):1119.
21. Zhou Z, Zheng Y, Ye H, Pu J, Sun G. Satellite image scene classification via convnet with context aggregation. In Pacific Rim Conference on Multimedia. Springer, Cham. 2018 (pp. 329–339).
22. Zhang J, Chaoquan L, Li X, Kim H-J, Wang J. A full convolutional network based on DenseNet for remote sensing scene classification. Math Biosci Eng. 2019;16(5):3345–67.
23. Mohanty SP, Czakon J, Kaczmarek KA, Pyskir A, Tarasiewicz P, Kunwar S, Rohrbach J, et al. Deep learning for understanding satellite imagery: an experimental survey. Front Artif Intell 2020;85.
24. Alhichri H, Alswayed AS, Bazi Y, Ammour N, Alajlan NA. Classification of remote sensing images using EfficientNet-B3 CNN model with attention. IEEE Access. 2021;9:14078–94.
25. Yang R, Zhang Y, Zhao P, Ji Z, Deng W. MSPPF-nets: a deep learning architecture for remote sensing image classification. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE. 2019 (pp. 3045–3048).
26. Mu C, Liu Y, Liu Y. Hyperspectral image spectral–spatial classification method based on deep adaptive feature fusion. Remote Sens. 2021;13(4):746.
27. Xu Q, Yuan X, Ouyang C, Zeng Y. Attention-based pyramid network for segmentation and classification of high-resolution and hyperspectral remote sensing images. Remote Sens. 2020;12(21):3501.
28. Zhang J, Zhao H, Li J. TRS: transformers for remote sensing scene classification. Remote Sens. 2021;13(20):4143.
29. Jiang C, Ren H, Ye X, Zhu J, Zeng H, Nan Y, Sun M, Ren X, Huo H. Object detection from UAV thermal infrared images and videos using YOLO models. Int J Appl Earth Obs Geoinf. 2022;112: 102912.
30. Masouleh MK, Shah-Hosseini R. Development and evaluation of a deep learning model for real-time ground vehicle semantic segmentation from UAV-based thermal infrared imagery. ISPRS J Photogrammetry and Remote Sens 2019;155:172–186.
31. Huyan L, Bai Y, Li Y, Jiang D, Zhang Y, Zhou Q, Wei J, Liu J, Zhang Y, Cui T. A lightweight object detection framework for remote sensing images. Remote Sens. 2021;13(4):683.
32. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 (pp. 770–778).
33. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
34. Tan M, Quoc L. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning. PMLR. 2019 (pp. 6105–14).
35. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015 (pp. 1–9).
36. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2017 (pp. 4700–4708).
37. Helber P, Bischke B, Dengel A, Borth D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE J Sel Top Appl Earth Observ Remote Sens. 2019;12(7):2217–26.
38. Yang Y, Newsam S. 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; 2010 (pp. 270–279).
39. Cheng G, Han J, Xiaoqiang L. Remote sensing image scene classification: benchmark and state of the art. Proc IEEE. 2017;105(10):1865–83.
40. Kang J, Fernandez-Beltran R, Duan P, Liu S, Plaza AJ. Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast. IEEE Trans Geosci Remote Sens. 2020;59(3):2598–610.

Publisher's Note