

Deep learning techniques for satellite image classification

Suresh Kumar Musali¹, Rajeshwari Janthakal¹, Nuvvusetty Rajasekhar^{2,3}

¹Department of Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore, Visvesvaraya Technological University, Belagavi, India

²Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, JNTU, Hyderabad, India

³Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Jun 3, 2024

Revised Sep 19, 2024

Accepted Oct 7, 2024

Keywords:

Deep learning
Image classification
Neural networks
Remote sensing
Satellite images

ABSTRACT

Because of its wide range of uses in computer vision applications, including image retrieval, remote sensing, object recognition, scene analysis, and surveillance, image classification has attracted a lot of attention. Assigning appropriate class labels to images according to their contents is the primary objective of image classification. In the domain of remote sensing, image classification and analysis play crucial roles in both military and civil applications. Conventional methods for scene analysis and remote sensing depended on low-level representations of features, such as those of color and texture. However, recent advancements have shifted towards the use of convolutional neural networks (CNNs), which have shown promising results in remote sensing and scene classification tasks. In light of effectiveness of deep learning (DL) models, this research aims to develop four DL models by fine tuning already existing DL models-CNNs, residual neural network (ResNet), visual geometry group (VGG-19), network mobile net V2 based model and classifies satellite images of RSI-CB256 data set in to four classes namely cloudy, desert, green_area and water. For the RSI-CB256 dataset, appropriate network structures are explored in this research to get good performance. The CNN, ResNet and VGG-19 base models achieved an accuracy of 90.48, 92.68 and 91.18 respectively. While the mobile net V2 based model outperformed the other three models with 96.83% accuracy.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Suresh Kumar Musali

Department of Information Science and Engineering, Dayananda Sagar College of Engineering, Bangalore

Visvesvaraya Technological University

Belagavi-590018, Karnataka, India

Email: sureshkumar-ise@dayanandasagar.edu

1. INTRODUCTION

There hasn't been much high-resolution satellite picture classification and analysis done with traditional techniques. This is because the imagery has so many intricate details. Spectral signatures, intricate texture and shape, spatial linkages, and temporal shifts are some of the characteristics that define these images. This result demonstrates how deep learning algorithms can comprehend the complex and non-uniform features of high-resolution remote sensing images. DenseNet121 and ResNet101, two convolutional neural networks (CNN)-based architectures, outperform InceptionV3, EfficientNet, and VGG16, scoring over 90% in all assessment criteria on the EuroSAT dataset, which includes classification measures including accuracy, recall, precision, and F1-score [1]. This research paper's primary contributions can be summed up as follows:

- It offers a thorough examination of deep learning techniques for classifying satellite images.
- It suggests optimized deep learning models for satellite images interpretation.

– Visualizations of training loss and validation loss of different techniques are also presented.

In order to classify satellite photos into four distinct locations deserts, cloudy areas, green spaces, and water bodies this research paper proposes an image interpretation technique. After then, there are numerous possible uses for this interpretation. Better accuracy is provided by the suggested models, and empirical findings support this. For image classification, this paper presents the MobileNetV2 architecture, which has been shown to be more effective than other widely used models. This creative method lowers the computational burden of training the model while simultaneously improving classification accuracy. Identifying and categorizing things from satellite images is essential for numerous uses, such as land planning, ecology, military, and marine monitoring. Rich in temporal and spatial information, satellite images are used in numerous ways to address practical remote sensing issues. There are numerous difficulties in classifying satellite images. These difficulties include the distribution, quantity, quality, and availability of data. These problems make the study of satellite images more challenging. Positive and encouraging results are obtained on a challenging dataset with significant intra- and inter-class heterogeneity using the proposed technique. The RSI-CB256 dataset shows that the recommended method has good accuracy [2].

Deep learning techniques are becoming more and more successful than classic machine learning techniques. These techniques can learn from text, image, video, and audio data. This is especially true in light of recent advancements in hardware power. Architecture is predicted to have similar effects given the effectiveness and advantages of deep learning approaches in other disciplines with growing amounts of data. Instead of using broad strokes to represent textures, we narrowed down to particular in this study. In this direction, the deep convolutional neural network model was used to classify a total of 4,500 satellite photos that included clouds, deserts, green spaces, and water bodies [3].

One important application of remote sensing (RS) data-driven earth observation technology is the classification of RS images. This technology is widely used in both the military and the civilian worlds. However, because to the characteristics of RS data, such as high dimensionality and a dearth of relevant labelled examples, RS picture classification faces substantial practical and technical challenges. The advent of novel deep learning (DL) algorithms in recent years has resulted in significant progress in methods for RS image classification using DL. These advancements open up new directions for the study and advancement of RS image classification [4].

The discipline of automatic image categorization and analysis is a thriving and active area of study with great significance throughout computer vision domains. Applications for it can be found in a wide range of fields, including video processing, remote sensing, object recognition, face recognition, spatial mapping, automatic disease detection, textile image analysis, and pattern recognition. In order to facilitate accurate and effective picture categorization in various domains, researchers and practitioners are constantly investigating and developing new methods. The significance of image classification advances is broad, allowing tasks like object recognition in images, feature extraction, retrieving relevant images from large datasets, detecting diseases or anomalies, mapping geographical regions, and enhancing video analysis. The continuous progress in this field paves the way for numerous innovative applications and contributes to the development of intelligent systems that can understand and interpret visual information with greater precision [5]–[7].

Assigning class labels to images is the primary objective of models based on image classification. In order to do this, a training dataset a collection of images used as training samples is used to train a classification model. The training process involves learning the distinguishing features and patterns in the images that correspond to different classes. Once the model is trained, a separate test dataset is used to evaluate its performance and predict the class labels of the test images. Through the examination of the forecasts on the test dataset, the images can be arranged in a coherent and significant sequence, enabling efficient categorization and retrieval of similar images. In order to improve the performance of classification-based systems, it is crucial to select discriminative and unique features. These features should capture the distinctive characteristics of each class, enabling accurate classification. The choice of appropriate features [8] contributes significantly to improving the system's capacity to differentiate between classes and improve overall classification accuracy. Researchers continuously explore and develop novel feature extraction techniques to extract the most relevant and informative features from images, thereby enhancing the performance of image classification models.

As a subset of machine learning, deep learning [9] refers to a class of models that can efficiently represent input at different levels of abstraction by using numerous processing layers. One of the key achievements of deep learning is its remarkable success in object detection and classification tasks, primarily achieved through the use of CNNs and the parallel computing power of graphical processing units (GPUs). We specifically investigate how well-suited seven popular deep convolutional neural networks—DenseNet121, InceptionV3, VGG16, VGG19, Xception, ResNet50, and InceptionResNetV2 are for mapping wetlands in Canada. The top three convolutional neural networks are InceptionResNetV2, ResNet50, and Xception, which offer state-of-the-art classification accuracies of 96.17%, 94.81%, and 93.57%, respectively.

Support vector machine (SVM) and random forest (RF) yielded much lower classification accuracies than CNNs, at 74.89% and 76.08%, respectively. Crucially, InceptionResNetV2 consistently outperforms all other convnets, suggesting that the Inception and ResNet modules together provide a useful architecture for classifying complex remote sensing scenarios such as wetlands [10].

One of the primary difficulties arises from the fact that objects in remote sensing images can be rotated within the view, making their identification and classification more complex. Additionally, the background in remote sensing images is often characterized by a high level of complexity, further adding to the challenge of accurate classification. To gather image datasets for remote sensing research, various platforms are utilized, including aerial systems, unmanned aerial vehicles (UAVs), and satellites. These platforms capture images from different perspectives and at varying spatial resolutions, providing valuable data for evaluating and advancing the field of remote sensing [11]. The study's following sections are arranged as follows: A thorough review of the literature is provided in Section 2 and a discussion of pertinent studies concerning the classification of remote sensing images. In Section 3, the authors discuss the dataset used in this research, highlighting its significance in evaluating the proposed methods. Section 4 presents a detailed description of the models employed in the study for remote sensing image classification. Section 5 presents the results of the experiments conducted, along with a thorough analysis, discussion, and comparisons of the obtained experimental values. The paper is finally concluded in Section 6, which summarizes the main conclusions and contributions of the suggested study in the field of remote sensing image classification.

Cheng *et al.* [12] discuss the intersection of remote sensing image scene classification and deep learning, highlighting the challenges, methods, benchmarks, and opportunities in this field. They provide a comprehensive overview of the state-of-the-art techniques and datasets used in remote sensing image classification. The paper concludes by emphasizing the potential of deep learning approaches and the need for further research and collaboration to address the remaining challenges and unlock new opportunities in remote sensing image analysis. By employing an unsupervised learning algorithm called "deep belief nets," the authors of [13] demonstrate that neural networks with many layers of hidden units can learn to efficiently represent the statistical structure of high-dimensional data. A group of researchers [14] performed a state-of-the-art review on Deep learning approaches applied to remote sensing datasets. Data sets like the UC Merced Land Use from the U.S. Geological Survey were used by researchers in the past.

Deep learning is used in remote sensing environments, and the methodological approaches of deep learning have generated a lot of interest in the development of remote sensing domains. DL possesses the capacity to effectively compete and overcome the diverse obstacles encountered in the field of remote sensing. The potential of deep learning in a remote sensing context is examined in this special issue, along with the most recent advancements and improvements [15].

The authors in [16] conducted a meta-analysis and reviewed over 200 publications in the field of DL for remote sensing. Using a large dataset of 1.2 million high-resolution images from the ImageNet LSVRC-2010 contest, the researchers of [17] trained a deep convolutional neural network with the goal of classifying these images into 1000 distinct classes with top-1 and top-5 error rates of 37.5% and 17.0%, respectively. A group of researchers used the Inception architecture and achieved excellent performance at a comparatively small computational cost [18].

With CNNs, databases of RS images may be recognized and categorized. The conventional CNN method generates coarse maps for classifying images of broad areas. The CNN method based on objects can be employed to tackle this problem [19]. In this study, we investigate the relationship between a convolutional network's accuracy and depth when applied to large-scale image recognition. Our main contribution is a thorough investigation of networks with varying depth using an architecture with relatively tiny (3×3) convolution filters. We show that a significant improvement over the state-of-the-art configurations may be achieved by increasing the depth to 16–19 weight layers [20].

2. METHOD

2.1. Data description

The RSI-CB256 satellite image classification dataset is the dataset that was chosen for this research [21]. The 5,361 images in this collection are split up into 4 types and are a combination of Google Map snapshots and sensor images. Every image has a size of 256×256. Cloudy (1500), desert (1131), green area (1500), and water (1500) are the classifications assigned to the images. Sample images from the collection are shown in Figure 1. Also sample satellite images from dataset are shown in Figure 2.

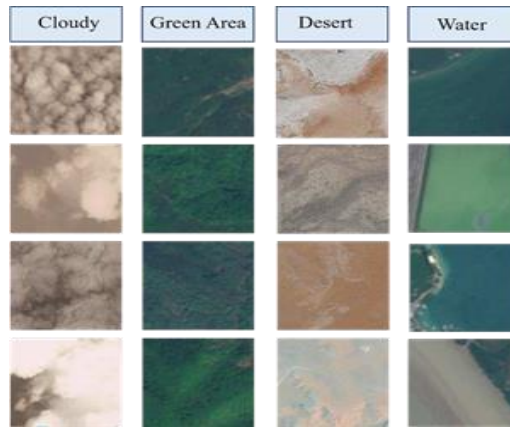


Figure 1. RSI-CB256 dataset sample images



Figure 2. RSI-CB256 dataset-samples of satellite images [2]

2.2. Data augmentation

As the dataset contains only 5,361 images it is not suitable to work with DL models. Hence this work increased the number of images in the dataset using augmentation techniques. An 80:20 ratio was used to split the data set into training and validation sets. Data augmentation provides a wider variety of training data to machine learning models, which enhances the model's performance. The input image I is mapped to dimensions of $M \times N \times C$, such that M is number of rows, N is number of columns and C is number of channels.

Rescales an input in [0,255] range to be in [0,1] range.

$RS(I(i,j))=I(i,j)/\max I(s,t)$ where i ranges from 1 to M, j ranges from 1 to N, C belongs to R,G,B.

Rotation range $R(I(i,j))=I(i*\cos -j*\sin, i*\sin + j*\cos)$

Shear range $Shu(I(i,j))=I(i+u*j,j)$

Zoom range $Zv(I(i,j))=I(i*v,j*v)$

Horizontal flip $HFw(I(i,j))=I(M-i-1,j)$

Vertical flip $VFx(I(i,j))=I(i,N-j-1)$

2.3. Proposed method

The objective of this research is to categorize satellite images. Using the satellite image as input, the suggested methodology divides the topography into four classifications. The input images are first subjected to pre-processing, and then they are classified. All experiments were conducted on hosted environment using graphical processing units as well as central processing units. GPU P100 was used. CUDA software was used to access the GPU's. Packages like NumPy [22], Pandas [23], Matplotlib [24], TensorFlow [25], Keras [26] and Sklearn [27] were used.

2.3.1. Proposed CNN based model

The proposed sequential CNN based model has 14 layers that include three 2D convolution layers, three max pooling 2D layers, four dense layers and three dropout layers. Figure 3 shows CNN architecture. 2D convolution layer is used to extract features from images. Max pooling 2D layer is used to reduce the size of the feature maps while preserving the most important features. It downscales input by using poolsize (2,2). The input data is classified or predicted using dense layers. It employs 128, 512, 256, and 4 units consecutively. In order to avoid overfitting, the dropout layers are employed last. The rate is set to 0.5. The model employed SoftMax and rectified linear unit (ReLU) as activation functions together with the Adam optimizer. Twenty epochs in all were chosen for the model's training.

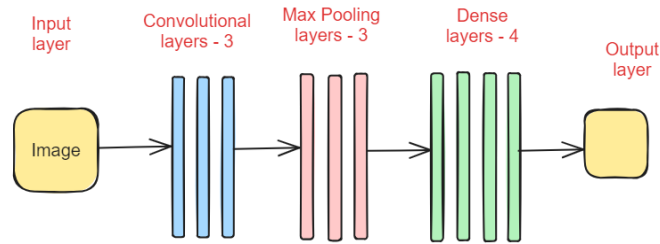


Figure 3. Proposed CNN architecture

Table 1 describes the complete summary of CNN architecture. Block refers to a set of operations applied sequentially in a particular layer or group of layers in the CNN. Operators (F) are the types of operations performed in the block. Blue coloured layer represents convolutional layers. Red coloured layer represents maxpooling layers. Green coloured layer represents dense layers or fully connected layers. Resolutions (H×W) are the dimensions of the feature maps after the operations in the block. They describe the height (H) and width (W) of the feature maps at that stage in the network. Channels (C) indicates the number of feature maps (or channels) produced by the convolutional layer. The number of channels often increases as the network goes deeper, allowing the network to learn more complex features. For instance, the first block outputs 32 channels, the second block outputs 64 channels, etc.

Table 1. CNN architecture details

Block	Operations (F)	Image resolution (H×W)	Features (C)
1	Conv (1 layer), ReLU, Maxpool (1 layer)	254×254	32
2	Conv (1), ReLU, Maxpool (1)	125×125	64
3	Conv (1), ReLU, Maxpool (1)	60×60	128

Proposed CNN algorithm:

- Step 1: proposed sequential CNN based model has 14 layers.
- Step 2: CNN uses 3 convolutional layers to extract features from images.
- Step 3: uses 3 maxpooling layers, which reduce their size and dimensionality.
- Step 4: four dense layers used to classify or predict input data.
- Step 5: three dropout layers are used to prevent overfitting.
- Step 6: finally, the output of the pooling layers is fed into a fully connected layer, which produces the final output of the network.
- Step 7: ReLU and SoftMax are used as activation functions and Adam optimizer were used in the model.

2.3.2. Proposed ResNet9 based model

The proposed ResNet9 based model has 9 layers that include convolution, residual block, max pooling layers and classifier. Figure 4 shows ResNet9 architecture. The classifier consists of an adaptive max pooling layer, a flatten layer, a dropout layer, and a linear layer. The model uses ReLU activation function and Adam optimizer. Table 2 gives the complete details of Resnet9 architecture.

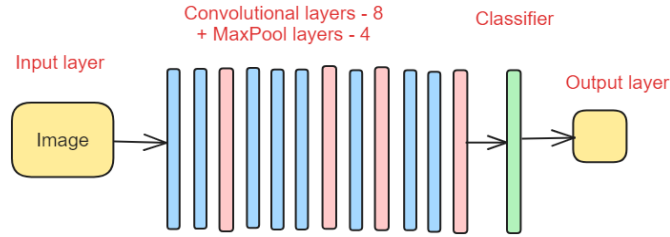


Figure 4. Proposed ResNet9 architecture

Table 2. ResNet9 architecture details

Block	Operations (F)	Image resolution (H×W)	Features (C)
1	Conv (2), ReLU, Maxpool (1)	112×112	64
2	Conv (3), ReLU, Maxpool (1)	56×56	128
3	Conv (1), ReLU, Maxpool (1)	28×28	256
4	Conv (2), ReLU, Maxpool (1)	14×14	512

Proposed ResNet9 algorithm:

- Step 1: proposed ResNet9 based model has 9 layers.
- Step 2: ResNet9 uses 8 convolutional layers to extract features from images.
- Step 3: uses 4 maxpooling layers, which reduce their size and dimensionality
- Step 4: one classifier layer used to classify or predict input data.
- Step 6: finally, the output of the pooling layers is fed into a fully connected layer, which produces the final output of the network.
- Step 7: ReLU is used as activation functions and Adam optimizer were used in the model.

2.3.3. Proposed VGG_19 based model

The proposed sequential VGG_19 based model has a total of 26 layers such that 16 are 2D convolution layers, five max pool layers, one flatten layer, one input layer and three dense layers. Figure 5 shows VGG_19 architecture. SoftMax was used as the activation function, Adam optimizer and 20 epochs. The model uses the pre-trained weights from the ImageNet dataset. Table 3 describes the complete summary of VGG_19 architecture.

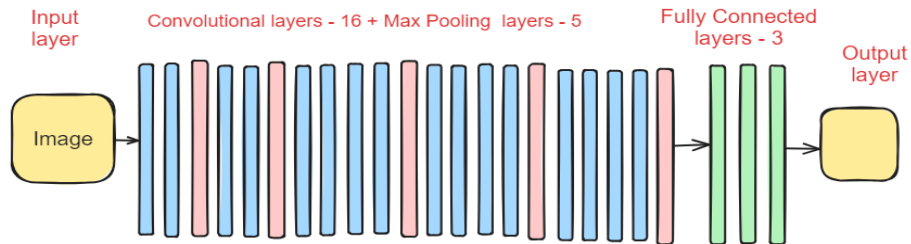


Figure 5. Proposed VGG_19 architecture

Table 3. VGG_19 architecture details

Block	Operations (F)	Image resolution (H×W)	Features (C)
1	Conv (2), ReLU, Maxpool (1)	224×224	64
2	Conv (2), ReLU, Maxpool (1)	112×112	128
3	Conv (4), ReLU, Maxpool (1)	56×56	256
4	Conv (4), ReLU, Maxpool (1)	28×28	512
5	Conv (4), ReLU, Maxpool (1)	14×14	512

Proposed VGG_19 algorithm:

- Step 1: proposed VGG-19 based model has 26 layers.
- Step 2: VGG-19 uses 16 convolutional layers to extract features from images.
- Step 3: uses 5 maxpooling layers, which reduce their size and dimensionality.

- Step 4: three dense layers used to classify or predict input data.
- Step 5: finally, the output of the pooling layers is fed into a fully connected layer, which produces the final output of the network.
- Step 6: SoftMax was used as activation functions and Adam optimizer were used in the model.

2.3.4. Proposed MobileNetV2 based model

The proposed MobileNetV2 based model is a sequential model that contains a total of five layers. It contains two dense layers, one global average pooling layer, one input and one functional layer. Figure 6 shows MobileNetV2 architecture. We add the MobileNetV2 base model without the top layer. ReLu and SoftMax activation functions are used. Adam optimizer is used for training the model. Table 4 describes the complete summary of MobileNetV2 architecture.

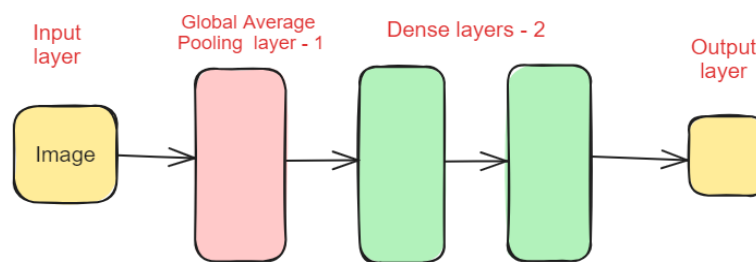


Figure 6. Proposed MobileNetV2 architecture

Table 4. MobilenetV2 architecture details

Block	Operations (F)	Image resolution (H×W)	Features (C)
1	ReLU, global Avgpool (1)	None, none	1280

Proposed MobileNetV2 algorithm:

- Step 1: proposed MobileNetV2 based model has 5 layers.
- Step 2: MobileNetV2 uses 1 global average pooling layer.
- Step 3: two dense layers used to classify or predict input data.
- Step 4: ReLU and SoftMax are used as activation functions and Adam optimizer were used in the model.

2.3.5. Accuracy

The following four evaluations are used for classification estimates [17]:

- I. True positive (TP): they are positive class labels that have been correctly predicted by the model.
- II. True negative (TN): they are negative class labels that have been correctly predicted by the model.
- III. False positive (FP): they are positive class labels that have been incorrectly predicted by the model.
- IV. False negative (FN): they are negative class labels that have been incorrectly predicted by the model.

Accuracy is expressed as the ratio of correct predictions in the model to all predictions:

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

3. RESULTS AND DISCUSSION

3.1. CNN based model

Table 5 shows the training loss, validation loss and validation accuracy of the model designed using CNN. There were 20 epochs in total. The model's highest accuracy was 90.48%. As the number of epochs increased from 1 to 20, the model's loss decreased. Figure 7 shows the training loss and validation loss as a function of the number of epochs. It showed consistent loss over the course of the training. Figure 8 shows the accuracy and epoch visualization of the model.

Table 5. Results of the CNN based model

Epoch.no.	Train loss	Validation loss	Accuracy
1	81.52	54.12	75.84
2	55.03	36.02	86.07
3	41.99	60.07	84.30
4	41.04	32.74	88.36
5	42.74	34.83	85.71
6	38.71	34.55	83.77
7	39.07	31.72	90.12
8	33.45	37.43	82.54
9	34.79	30.01	90.48
10	33.49	25.62	89.24
11	31.06	26.96	90.83
12	30.71	30.18	87.83
13	32.07	27.77	88.89
14	33.11	42.45	86.24
15	32.64	24.40	90.65
16	28.46	28.92	89.42
17	29.81	21.99	90.83
18	27.21	25.59	89.77
19	34.32	29.84	90.65
20	32.03	28.35	90.48

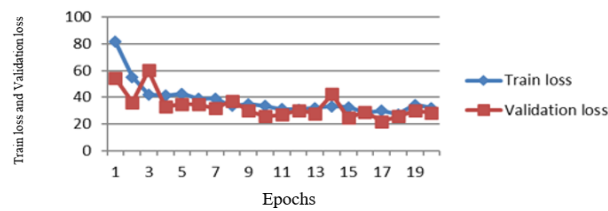


Figure 7. Visualization of training loss and validation loss vs no. of epochs

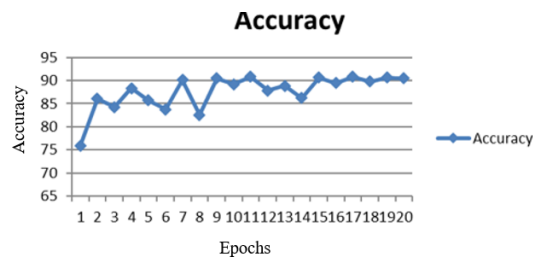


Figure 8. Visualization of accuracy and epochs of CNN based model

3.2. ResNet9 based model

Table 6 shows the training loss, validation loss and validation accuracy of the model designed using ResNet9. The number of epochs was set to 20. The model's highest accuracy was 92.68%. As the number of epochs increased from 1 to 20, the model's loss decreased. Figure 9 shows the relationship between training loss and validation loss and epoch count graphically. It showed consistent loss over the course of the training. Figure 10 shows the accuracy and epochs of the ResNet9-based model visualized.

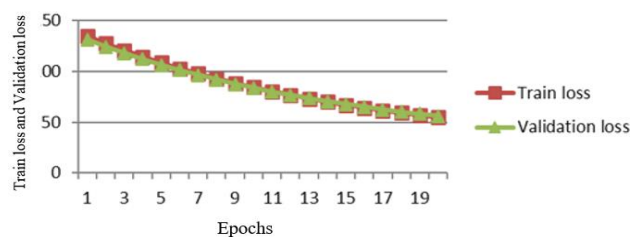


Figure 9. Visualization of training loss and validation loss vs no. of epochs

Table 6. Results of the ResNet9 based model

Epoch no	Train loss	Validation loss	Accuracy
1	135.06	131.30	76.19
2	127.57	124.35	79.28
3	120.75	117.97	80.34
4	114.39	112.01	81.66
5	108.47	106.45	83.42
6	102.94	101.27	84.66
7	97.77	96.42	86.07
8	92.95	91.91	86.77
9	88.45	87.69	87.21
10	84.25	83.76	88.18
11	80.33	80.09	88.71
12	76.66	76.65	89.15
13	73.25	73.46	89.77
14	70.06	70.47	90.74
15	67.08	67.67	90.92
16	64.29	65.05	91.18
17	61.67	62.61	91.45
18	59.23	60.30	92.06
19	56.93	58.15	92.50
20	54.77	56.12	92.68

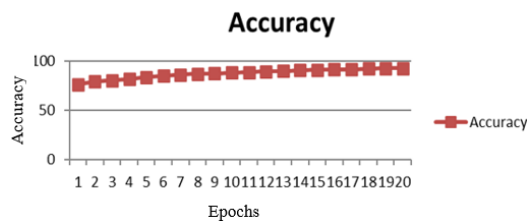


Figure 10. Visualization of accuracy and epochs of ResNet9 based model

3.3. VGG-19 based model

Table 7 shows the training loss, validation loss and validation accuracy of the model designed using VGG-19. The number of epochs was set to 20. The model achieved a maximum accuracy of 91.18 %. The model loss decreased as the number of epochs increased from 1 to 20. Figure 11, graphically depicts of training loss and validation loss VS No. of epochs. It showed consistent loss over the course of training. visualization of accuracy and epochs of model can be seen in Figure 12.

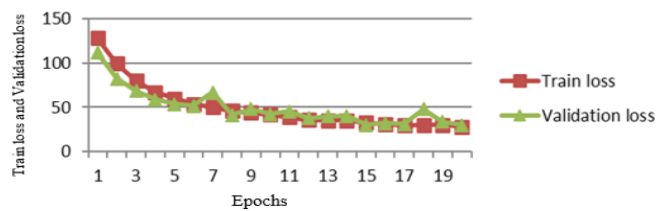


Figure 11. Visualization of training loss and validation loss Vs no. of epochs

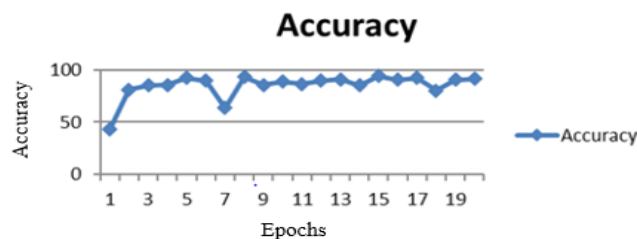


Figure 12. Visualization of accuracy and epochs of VGG based model

Table 7. Results of the VGG model

Epoch no	Train loss	Validation loss	Accuracy
1	128.53	111.75	43.21
2	99.30	82.53	81.13
3	80.04	68.72	85.19
4	67.30	58.87	85.54
5	59.89	53.72	92.24
6	53.64	51.54	89.77
7	50.35	67.13	63.49
8	46.19	41.56	93.30
9	44.15	48.23	85.36
10	41.94	42.35	88.89
11	39.11	45.35	86.24
12	35.87	37.94	89.77
13	35.25	40.10	90.83
14	35.32	40.25	85.01
15	32.48	30.17	94.00
16	31.35	32.23	90.65
17	29.86	31.38	92.42
18	29.91	48.11	80.25
19	30.02	33.47	90.65
20	27.88	30.10	91.18

3.4. MobileNetV2 based model

Table 8 shows the training loss, validation loss and validation accuracy of the model designed using MobileNetV2. The number of epochs was set to 20. The model achieved a maximum accuracy of 96.83%. The model loss decreased as the number of epochs increased from 1 to 20. Figure 13, graphically depicts of training loss and validation loss VS no. of epochs. Over the course of the training, there is no consistent loss. Visualization of accuracy and epochs of MobileNetV2 based model can be seen in Figure 14.

Table 8. Results of the MobileNet V2 based model

Epoch no	Train loss	Validation loss	Accuracy
1	26.14	11.52	97.00
2	9.25	6.49	98.24
3	7.02	9.74	96.47
4	5.92	9.98	96.65
5	5.66	6.44	98.24
6	6.29	16.66	94.18
7	5.36	13.82	95.77
8	5.76	3.59	98.77
9	4.57	6.46	97.71
10	3.58	6.61	97.71
11	3.21	9.87	96.83
12	4.57	6.45	97.53
13	4.32	4.27	98.41
14	4.67	6.88	97.35
15	3.29	11.04	96.30
16	2.80	3.96	98.06
17	3.30	13.25	95.24
18	3.71	1.86	99.12
19	3.45	2.48	99.12
20	2.32	13.02	96.83

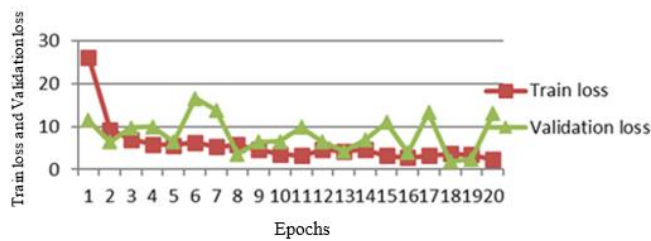


Figure 13. Visualization of training loss and validation loss VS No. of epochs

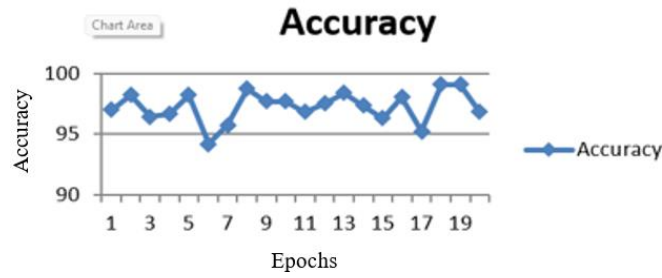


Figure 14. Visualization of accuracy and epochs of MobileNetV2 based model

3.5. Comparison of four models

Table 9 depicts the accuracies achieved by all the four models at each epoch. The corresponding graph is depicted by Figure 15. It shows the final accuracies achieved by the proposed models. In the experiments conducted by this research, the CNN based model achieved 90.48%, ResNet based model achieved 92.68%, VGG-19 based model achieved 91.18% and finally MobileNetV2 based model achieved 96.83% accuracy. Clearly, MobileNet V2 based model outperforms the other three deep learning models in the experiments conducted by this research. Since these three models show consistent loss during the training process, they will perform better with bigger training datasets. Model complexity and dataset size plays an important role in choosing best deep learning model for remote sensing images classification. A bar-graph in Figure 16 is projected for showing the accuracy achieved by the four models.

Table 9. Depicting the accuracies of the proposed models at every epoch

Epoch no	CNN	RESNET9	VGG-19	Mobile Net V2
1	75.84	76.19	43.21	97.00
2	86.07	79.28	81.13	98.24
3	84.30	80.34	85.19	96.47
4	88.36	81.66	85.54	96.65
5	85.71	83.42	92.24	98.24
6	83.77	84.66	89.77	94.18
7	90.12	86.07	63.49	95.77
8	82.54	86.77	93.30	98.77
9	90.48	87.21	85.36	97.71
10	89.24	88.18	88.89	97.71
11	90.83	88.71	86.24	96.83
12	87.83	89.15	89.77	97.53
13	88.89	89.77	90.83	98.41
14	86.24	90.74	85.01	97.35
15	90.65	90.92	94.00	96.30
16	89.42	91.18	90.65	98.06
17	90.83	91.45	92.42	95.24
18	89.77	92.06	80.25	99.12
19	90.65	92.50	90.65	99.12
20	90.48	92.68	91.18	96.83

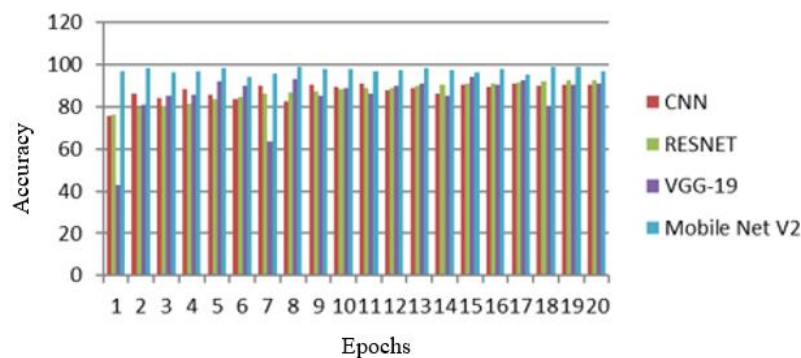


Figure 15. Accuracy visualization of four models at each epoch

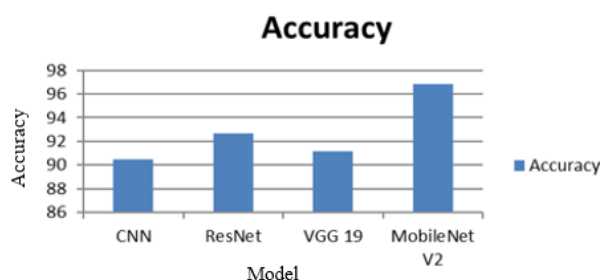


Figure 16. Bar graph depicting accuracy of proposed models

3.6. Discussions

The automatic interpretation of satellite images is presented by the proposed methodology. Work involving manual interpretation can be quite taxing. To further add to the difficulty of the endeavor, domain expertise is needed. Four distinct classes can be created from the images using the suggested method. The data clearly show that MobileNetV2 offers the greatest performance.

The fact that the full-training technique outperformed fine-tuning in all circumstances in terms of classification results is among the research's most intriguing findings. Previous research on the classification of very high resolution pictures shown that fine-tuning outperformed full-training. A branch of machine learning called deep learning is able to represent and analyze data at various levels of abstraction. It has produced quite impressive outcomes in the geoscience and satellite imaging fields. This research evaluates several deep learning models for satellite image interpretation and classification. These models consist of VGG19, CNN, ResNet9, and MobileNetV2. With MobileNetV2, accuracy is at its peak. Both VGG19 and ResNet9, a non-scalable network, produce incredibly encouraging outcomes. CNN, on the other hand, was unable to provide extremely positive outcomes. Convolution-based neural networks, such as ResNet and MobileNet, are used to extract features from images and classify them.

ResNet performs better when handling vanishing gradient issues. This is because it makes it easier for gradients to pass across intricate neural networks, which makes it possible to train deeper networks and improve convergence qualities. MobileNet provides the greatest results, with an accuracy of 96.83. MobileNets employ depth-wise separable convolutions to reduce resource requirements and build lighter networks. MobileNets employ separable convolutions for the subsequent layers after using standard convolutions for the first layer.

4. CONCLUSION

This paper developed four fine tuned deep learning models using CNN, ResNet9, VGG_19 and MobileNetV2 respectively for satellite image classifiers. The performance of all the models was satisfactory. The paper analyzed the accuracy, loss of these models with respect to number of epochs. The accuracy of MobileNet V2 based model stood highest among its peers with 96.83% accuracy and lower computational cost. It also has a lightweight architecture with fewer parameters when compared with other models. On the other hand, CNN, Resnet9 and VGG_19 models achieved 90.48, 92.68 and 91.18 accuracy respectively. All models were trained with an epoch count of 20 beyond which there may be over fitting in the models. In future the works aims to increase the number of epochs as well as images in the data set for a better analysis. Further, performance on satellite images from remote sensing can be enhanced by using larger convolutional layers in deeper CNN models.

ACKNOWLEDGEMENTS

The authors express their gratitude and acknowledge the support provided by Dayananda Sagar College of Engineering.





REFERENCES

- [1] A. A. Adegun, S. Viriri, and J. R. Tapamo, "Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis," *Journal of Big Data*, vol. 10, no. 1, p. 93, Jun. 2023, doi: 10.1186/s40537-023-00772-x.
- [2] S. Tehsin, S. Kausar, A. Jameel, M. Humayun, and D. K. Almfarreh, "Satellite image categorization using scalable deep learning," *Applied Sciences (Switzerland)*, vol. 13, no. 8, p. 5108, Apr. 2023, doi: 10.3390/app13085108.
- [3] F. DIKER and İ. ERKAN, "Classification of satellite images with deep convolutional neural networks and its effect on architecture," *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*, vol. 23, pp. 31–41, Dec. 2022, doi: 10.18038/estubtda.1165890.




- [4] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, Nov. 2018, doi: 10.1002/widm.1264.
- [5] D. S. Shakya, "Analysis of artificial intelligence based image classification techniques," *Journal of Innovative Image Processing*, vol. 2, no. 1, pp. 44–54, Mar. 2020, doi: 10.36548/jiip.2020.1.005.
- [6] A. Shabbir *et al.*, "Detection of glaucoma using retinal fundus images: a comprehensive review," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, pp. 2033–2076, 2021, doi: 10.3934/MBE.2021106.
- [7] A. Rasheed *et al.*, "Fabric defect detection using computer vision techniques: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–24, Nov. 2020, doi: 10.1155/2020/8189403.
- [8] J. Srinivas, A. Abdul, M. Qyser, and B. E. Reddy, "Classification of textures based on circular and elliptical weighted symmetric texture matrix," vol. 7, pp. 593–600, 2018, doi: Classification of Textures based on Circular and Elliptical Weighted Symmetric Texture Matrix.
- [9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [10] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, no. 7, p. 1119, Jul. 2018, doi: 10.3390/rs10071119.
- [11] B. Petrovska, T. Atanasova-Pacemska, R. Corizzo, P. Mignone, P. Lameski, and E. Zdravevski, "Aerial scene classification through fine-tuning with adaptive learning rates and label smoothing," *Applied Sciences (Switzerland)*, vol. 10, no. 17, p. 5792, Aug. 2020, doi: 10.3390/app10175792.
- [12] G. Cheng, X. Xie, J. Han, L. Guo, and G. S. Xia, "Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.
- [13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.
- [14] A. Abdollahi, B. Pradhan, N. Shukla, S. Chakraborty, and A. Alamri, "Deep learning approaches applied to remote sensing datasets for road extraction: A state-of-the-art review," *Remote Sensing*, vol. 12, no. 9, p. 1444, May 2020, doi: 10.3390/RS12091444.
- [15] G. Manogaran, H. Quadrat-Ullah, and B. S. Rawal Kshatriya, "Introduction to the special issue on deep learning for remote sensing environments," *European Journal of Remote Sensing*, vol. 53, no. sup1, pp. 1–3, Jun. 2020, doi: 10.1080/22797254.2020.1777802.
- [16] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, Jun. 2019, doi: 10.1016/j.isprsjprs.2019.04.015.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 4278–4284, 2017, doi: 10.1609/aaai.v31i1.11231.
- [19] N. Supriya and M. S. N. Murthy, "A comparative study of the CNN based models used for remote sensing image classification," *International Journal of Electrical and Electronics Research*, vol. 11, no. 3, pp. 646–651, Jul. 2023, doi: 10.37391/ijeer.110301.
- [20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [21] Mahmoud Reda, "Satellite Image Classification." [Online]. Available: <https://www.kaggle.com/datasets/mahmouredda55/satellite-image-classification>
- [22] T. Oliphant and J. k. Millma, "A guide to NumPy," *Trelgol Publishing*, 2006, doi: DOI:10.1109/MCSE.2007.58.
- [23] L. A. Snider and S. E. Swedo, "PANDAS: current status and directions for research," *Molecular Psychiatry*, vol. 9, no. 10, pp. 900–907, 2004, doi: 10.1038/sj.mp.4001542.
- [24] P. Barrett, J. D. Hunter, and P. Greenfield, "matplotlib--a portable python plotting package," in *Astronomical data analysis software and systems XIV*, 2005, pp. 91–95.
- [25] TensorFlow Developers, "TensorFlow," *Zenodo*, May 2022, doi: 10.5281/zenodo.6574269..
- [26] N. Ketkar, "Introduction to keras," in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 97–111. doi: 10.1007/978-1-4842-2766-4_7.
- [27] B. Komer, J. Bergstra, and C. Eliasmith, "Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn," in *Proceedings of the 13th Python in Science Conference*, 2014, pp. 32–37. doi: 10.25080/majora-14bd3278-006.

BIOGRAPHIES OF AUTHORS






Suresh Kumar Musali     holds a M.Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi. He currently serves as an Assistant Professor in Dayananda Sagar College of Engineering, Bangalore. He has 16 years of experience in teaching. He has published around 10 papers in Journals and Conferences. He can be contacted at email: sureshkumar-ise@dayanandasagar.edu.



Dr. Rajeshwari Janthakal    working as Professor and Head of Information Science and Engineering department, Dayananda Sagar College of Engineering, Bangalore. She earned her Ph. D degree from Visvesvaraya Technological University, Belagavi. She has 21 years of experience. She has published around 30 papers in Journals and Conferences. She has also published two patents and received a Research grant from Indian Council of Medical Research. She can be contacted at email: rajeshwarij-ise@dayanandasagar.edu.



Dr. Nuvvusetty Rajasekhar    currently working as Professor in the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad. He received doctoral degree in the year 2015 from Acharya Nagarjuna University, Guntur and M.Tech. in Computer Science & Engineering (2006) from Bharath University, Chennai. He graduated in Mechanical Engineering from S.V. University, Tirupati. He is currently, India. He has a total of 18 plus years' experience in teaching. He is actively involved in research and some of his research interests include Network Security, Data Mining, Machine Learning and Artificial Intelligence. He has published more than 30 research papers in international journals and conferences and has received best paper award in international conference at IEEE Conference at Agadir, Morocco in the year 2016. He is also serving as reviewer and editorial board member for various reputed peer reviewed journals and technical committee member for various international conferences. He can be contacted at email: rajasekhar531@gmail.com.