Comparative Performance Analysis of Deep Learning Techniques for Soil Image Classification

Girish D. Chate
School of Computer Sciences,
Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon-425001, Maharashtra

S.S. Bhamare
School of Computer Sciences,
Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon-425001, Maharashtra

ABSTRACT
Soil classification through image analysis has appeared as a key tool for advancing precision agriculture and land resource management. This research conducts a detailed comparative study of two prominent convolutional neural network architectures, InceptionV3 and ResNet50, applied to soil image classification. A curated dataset including diverse soil types used for model training and evaluation. Transfer learning was utilized to modify models already trained for the soil classification task, and hyperparameters were optimized to enhance performance. Used comprehensive assessment criteria, including overall accuracy, precision, recall, F1-score, and matrix of confusion analysis. The results show that InceptionV3 offers advantages in computational efficiency and faster convergence, while ResNet50 demonstrates superior classification accuracy and generalization, particularly for heterogeneous and complex soil textures. This paper provides an in-depth understanding of the trade-offs between model complexity, training dynamics, and classification performance, serving as a guideline for future applications of deep learning in soil science.

General Terms
Soil image classification, Agriculture, Image processing

Keywords
Soil classification, Agriculture, Machine Learning, InceptionV3, ResNet50.

1. INTRODUCTION
Soil classification is a critical part in the fields of agriculture, environmental monitoring, geotechnical engineering, and land resource management. Accurate identification of soil types enables informed decision-making about crop choice, irrigation planning, land-use management, and environmental conservation. Traditionally, soil classification performed through physical sampling followed by laboratory analysis of properties such as texture, colour, moisture content, and chemical composition. While highly correct, these conventional methods are often time-consuming, labour-intensive, costly, and require domain ability [1]. With the rise of machine learning and computer vision technologies, there has been growing interest in automating the soil classification process through image-based techniques. Digital soil images, easily captured using modern cameras and smartphones, offer a rich source of information about surface texture, structure, and colour variations. However, the high intra-class variability and inter-class similarity in soil images make classification a challenging task, demanding robust and efficient models capable of capturing sensitive features [2]. Deep learning, especially Convolutional Neural Networks, has transformed image categorization by obtaining superior outcomes in several applications, including medical imaging, remote sensing, and agriculture [3], [4]. Among the various CNN architectures, InceptionV3 and ResNet50 have collected significant attention for their powerful feature extraction capabilities and efficient training mechanisms. InceptionV3, an improvement over the original GoogLeNet Inception architecture, introduces factorized convolutions, batch normalization, and efficient grid size reduction, allowing it to achieve high accuracy while keeping computational efficiency [5]. Its design philosophy emphasizes multi-scale feature extraction, making it suitable for capturing both fine and common details within images. ResNet50, on the other hand, is a deep residual network that addresses the vanishing gradient problem common in very deep networks. ResNet50 uses residual connections, supporting the training of highly deep models without performance decrease.[6]. Its ability to learn complex hierarchical features makes it highly effective for tasks requiring fine-grained image classification. The InceptionV3 and ResNet50 architectures are compared in this paper for the classification problem of soil images. A custom dataset of soil images being various soil types and textures consider training and evaluate both models. Using transfer learning techniques, the networks developed for the soil classification task after getting initialized with weights pre-trained on the ImageNet dataset. complete evaluation of performance utilizing criteria including F1-score analysis, recall, accuracy, and precision.

The primary goal of this work is to assess the strengths and limitations of each architecture in the context of soil image classification, considering factors such as training efficiency, generalization ability, and robustness to variability in soil appearance. By understanding the comparative performance of these models, this study aims to provide valuable insights for researchers and practitioners looking to deploy deep learning solutions for soil classification in real-world agricultural and environmental applications. This research contributes to the precise goal of developing scalable, cost-effective, and accessible soil analysis tools, thereby supporting the advancement of precision agriculture and sustainable land management practices [7].

2. LITRATURE REVIEW
The use of machine learning and image processing methods for soil categorization has been investigated in several research, aiming to reduce reliance on traditional laboratory methods. Early efforts merged standard classifiers like Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN) with classical image processing techniques for feature extraction. Das et al., for example, used the Gray-Level Co-occurrence Matrix (GLCM) to extract textural characteristics from soil images and then used SVM to classify them, with a decent level of accuracy on a small number of datasets. [8].

Convolutional Neural Networks (CNNs) are better at automatically learning hierarchical features, researchers are

using them more and more for soil image categorization as deep learning advances. Kamilaris and Prenafeta-Boldú provided a comprehensive review of deep learning applications in agriculture, highlighting the effectiveness of CNNs in tasks like plant disease detection and soil texture classification [9].

Zhang et al. proposed a Convolutional Neural Networks based approach for soil type classification using RGB images captured under natural conditions. Their model, trained using transfer learning from ImageNet weights, showed improved accuracy and generalization across varying lighting conditions and soil textures [10]. Similarly, Anand et al. evaluated multiple deep CNN architectures, including VGG16, ResNet50, and DenseNet121, for classifying Indian soil types. Their experiments showed that ResNet50 outperformed other models in terms of accuracy and training stability [11].

The Inception family of networks has also been employed in soil-related tasks. Kumar et al. used InceptionV3 for the classification of soil fertility levels and found that multi-scale feature extraction helped in distinguishing subtle variations in soil appearance [12]. Although InceptionV3 less explored in this domain, its architectural improvements over InceptionV1 make it a promising candidate for efficient soil classification tasks.

Moreover, ensemble techniques combining predictions from multiple CNN models investigated to boost performance and reduce overfitting. For instance, Patil et al. implemented an ensemble of ResNet and Inception models, achieving higher robustness in classification under diverse environmental conditions [13].

Iqbal et al. introduced a deep learning framework using a MobileNetV2-based lightweight CNN for real-time soil classification on mobile devices. The model achieved a good trade-off between accuracy and computational efficiency, highlighting its potential for field deployment in remote agricultural areas [14]. Similarly, Prasad et al. utilized EfficientNet-B0 to classify soil texture classes and showed improved parameter efficiency compared to heavier models like VGG and ResNet [15].

In another study, Shende and Jagdale developed a soil classification system based on transfer learning using DenseNet121. Their approach used fine-tuned ImageNet weights to extract deep texture features from soil images under natural lighting. The model evaluated on a regional soil dataset from India and reported over 90% classification accuracy [16].

A hybrid approach combining spectral and image data proposed by Bian et al., where CNNs used in parallel with 1D spectral analysis to integrate spatial and spectral features. This method significantly improved classification accuracy for heterogeneous soil samples compared to single-modal approaches [17].

Address the issue of small datasets, researchers have used data augmentation and synthetic image generation using Generative Adversarial Networks (GANs). Taneja et al. implemented a GAN-based data expansion strategy for soil texture classification, which helped mitigate overfitting in CNN models trained on limited soil image datasets [18].

The importance of domain adaptation has also recognized. Alam and Roy developed a domain-adaptive CNN using adversarial training to classify soil images captured under different lighting and seasonal conditions. Their method showed enhanced generalization across datasets collected from geographically distinct regions [19].

Sethi et al. compared pretrained CNNs including Xception, InceptionV3, and NASNetMobile for multi-class soil classification and reported that Xception performed best in scenarios requiring fine-grained texture discrimination [20]. Their work highlighted the importance of choosing architectures based on dataset complexity and deployment constraints.

With these advancements, challenges such as high intra-class variability, limited annotated datasets, and the influence of environmental noise remain unchanged. Thus, there is a need for comparative studies that evaluate CNN architectures under standardized settings using domain-specific datasets. This study addresses this gap by evaluating InceptionV3 and ResNet50 on a custom soil image dataset using transfer learning, providing insights into their respective strengths and limitations for real-world applications.

3. METHODOLOGY

3.1 Dataset

The dataset employed in this study includes a collection of 5,000 high-resolution soil images, systematically categorized into four major soil types such as black soil, red soil, laterite soil, and white soil. These categories selected based on their agronomic and environmental significance, ensuring that the classification task mirrors real world soil diversity.

3.1.1 Collection of Soil Image Dataset

The soil images captured using mobile phone cameras under real-world field conditions. The dataset reflects a wide range of natural variability in soil appearance, influenced by factors such as geographic location, climate, lighting conditions, and moisture content. This diversity is essential for developing deep learning models that are not only correct but also robust and generalizable, enabling reliable soil classification across different environments and soil types. The dataset includes four primary soil types commonly found across different regions such as black soil, red soil, laterite soil, and white soil.

Table 1: Soil type and labels for the collected soil images

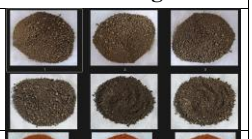
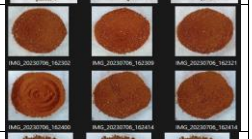
Soil Type	Sample Labels	Soil Image
Black Soil	B1, B2, B3	
Red Soil	L1, L2	
Laterite Soil	R1, R2	
White Soil	W1, W2	

Table 1 presents the labeling scheme adopted for organizing the collected soil image dataset. Each soil type such as black soil, red soil, laterite soil, and white soil is assigned a specific prefix letter such that B1, B2, B3 for black soil, R1, R2 for red soil, L1, L2 for laterite soil, and W1, W2 for white soil. The prefix followed by a numerical identifier to distinguish between different samples within the same category. This systematic

labeling to easier track and manage dataset during model training and evaluation and supports structured analysis of multiclass variability among soil samples.

Each image is an RGB (Red, Green, Blue) image, being the true colour characteristics of soil surfaces. Visual properties such as texture patterns, particle granularity, colour shades, and surface structures inherently encoded in these images, making them suitable for computer vision-based soil classification tasks. Figure 1 shows the pipeline of soil image processing.

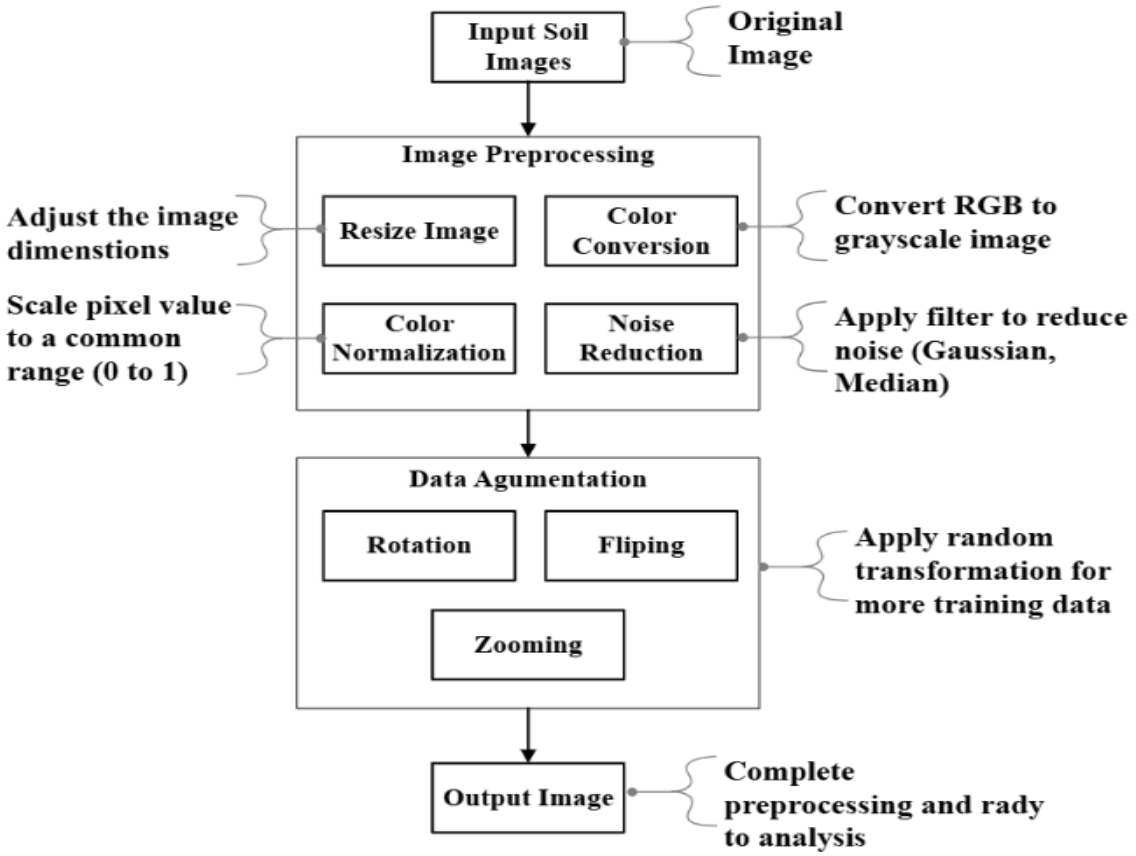


Figure 1: Pipeline of Soil Image Preprocessing

3.1.2 Image Preprocessing

Before model training, each image was subjected to several preprocessing processes to normalize the dataset and conform to the input requirements of the deep learning architectures InceptionV3 and ResNet50.

3.1.2.1 Resizing

Each image scaled to a fixed resolution of 224 × 224 pixels. This dimension selected to match the predicted input size of both pre-trained models, ensuring compatibility while keeping sufficient detail for feature extraction. the resizing an image to a new size. The mapping from new coordinates x', y' to original coordinates (x, y) is as follows.

$$x = x'.\frac{W_{original}}{W_{new}} \quad (1)$$

$$y = y.\frac{H_{original}}{H_{new}} \quad (2)$$

3.1.2.2 Normalization

The pixel intensity values, originally ranging from 0 to 255, normalized to a [0, 1] range. Normalization performed by dividing each pixel value by 255. This step is crucial for stabilizing the training process by preventing large gradient updates and helping faster convergence. The formula of image normalization is as follows.

$$Normalized\ Value = \frac{Pixel\ Value}{255} \quad (3)$$

Where,

- Pixel Value is the original intensity of a pixel that ranging from 0 to 255 for 8-bit images.
- Normalized Value is the scaled pixel value that ranging from 0 to 1).

3.1.3 Dataset Partitioning

To effectively train, validate, and evaluate the deep learning models, the dataset partitioned into three non-overlapping subsets.

3.1.3.1 Training Set

Comprising 3,500 images, the 70% training set used to improve the model weights during learning. A large and diverse training set ensures that the model captures a wide spectrum of soil feature variations.

3.1.3.2 Validation Set

A subset of 750 images chosen for model validation. This 15% set used to check model performance during training, tune

hyperparameters, and implement early stopping criteria to prevent overfitting.

3.1.3.3 Test Set

Another 750 images reserved as an independent 15% test set to evaluate the final model's generalization capability on previously unseen data. The partitioning conducted randomly but stratified to ensure that each soil class proportionally represented across all subsets.

3.1.4 Data Augmentation

Enhance the robustness of the models and mitigate the risk of overfitting especially given the limited size of the dataset compared to large-scale vision benchmarks data augmentation techniques applied dynamically during training.

3.1.4.1 Rotation

Images randomly rotated within a range of -30 to +30 degrees. Rotation augmentation helps the model learn orientation-invariant features, accommodating natural variations in soil image capture angles. The image rotation formula is based on 2D coordinate transformation using trigonometry. When an image point (x, y) rotated by an angle θ in radians about the origin (0, 0), the new coordinates (x', y') given by,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\,cos\theta - y\,sin\theta \\ x\,sin\theta + y\,cos\theta \end{bmatrix} \quad (4)$$

Where,

- (x, y): original coordinates,
- (x', y'): new coordinates after rotation,
- θ: rotation angle to counterclockwise and in radians.

3.1.4.2 Flipping

Horizontal and vertical flipping applied randomly. This introduces mirror-image versions of the samples, allowing the model to generalize across symmetrically varied images. Image flipping involves reversing pixel positions across a chosen axis either horizontal, vertical, or both which is a 180° rotation. Below are the formulas for each case.

Horizontal Flipping (Left to Right)

$$x' = W - 1 - x, \quad y' = y \quad (5)$$

Vertical Flipping (Top to Bottom)

$$x' = x, \quad y' = H - 1 - y \quad (6)$$

Horizontal + Vertical Flip (180° rotation)

$$x' = W - 1 - x, \quad y' = H - 1 - y \quad (7)$$

Where,

- x, y: original pixel coordinates.
- x', y': coordinates after flipping.
- W: image width.
- H: image height.

3.1.4.3 Zooming

Random zoom transformations performed within a specific range 90% to 110% of the original size. Zoom augmentation enables the model to manage variations in distance between the camera and the soil surface. image zooming is a bit different than rotation and flipping. Zooming means rescaling the image around a point either zooming in or zooming out. Zooming in or zooming out is a pixel coordinate (x, y), a scaling factor s, and a zoom center x_c, y_c, the new coordinates x', y' formula is as follows.

$$x' = s(x - x_c) + x_c \quad (8)$$

$$y' = s(y - y_c) + y_c \quad (9)$$

Where:

- (x , y) = original pixel position,
- (x', y') = new pixel position after zooming,
- (x_c , y_c) = center of zoom (often the center of the image),
- s = zoom scale factor,

  $s > 1$ = zoom in and $s < 1$ = zoom out.

These augmentations significantly expand the effective size of the training set, exposing the models to a broader range of simulated real-world conditions. Consequently, the trained models become more resilient to noise and variability in input images.

3.2 Model Architecture

In this paper, two prominent deep learning architectures InceptionV3 and ResNet50 used for soil image classification. Both models have shown state-of-the-art performance across various image recognition tasks due to their ability to effectively extract hierarchical features. Improve model performance on the soil dataset, transfer learning techniques utilized, leveraging pre-trained weights from the ImageNet dataset. Figure 2 show the architecture of InceptionV3 and ResNet50 to the soil classification task.

3.2.1 InceptionV3 Architecture

InceptionV3 is an improved variant of the original Inception (GoogLeNet) architecture, designed to balance accuracy and computational efficiency. Key enhancements introduced in InceptionV3 included.

- Factorized Convolutions: Large convolutional filters $\mathbf{5 \times 5}$ decomposed into smaller operations that is two consecutive $\mathbf{3 \times 3}$ convolutions. This reduces computational complexity while preserving representational power. Suppose, Input feature map size is $\boldsymbol{H \times W}$, then Number of input channels is $\boldsymbol{C_{in}}$, and Number of output channels is $\boldsymbol{C_{out}}$. So,
  - Single $\mathbf{5 \times 5}$ convolution

    $$Parameters = 5 \times 5 \times C_{in} \times C_{out} \quad (10)$$

  - Two Consecutive 3 × 3 Convolutions

    $$Parameters\ per\ layer = 3 \times 3 \times C_{in} \times C_{out} \quad (11)$$

- Batch Normalization: Used extensively throughout the network to accelerate convergence and mitigate internal covariate shifts during training. For a given mini batch $\boldsymbol{B} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$, batch normalization applied to each feature dimension as follows.

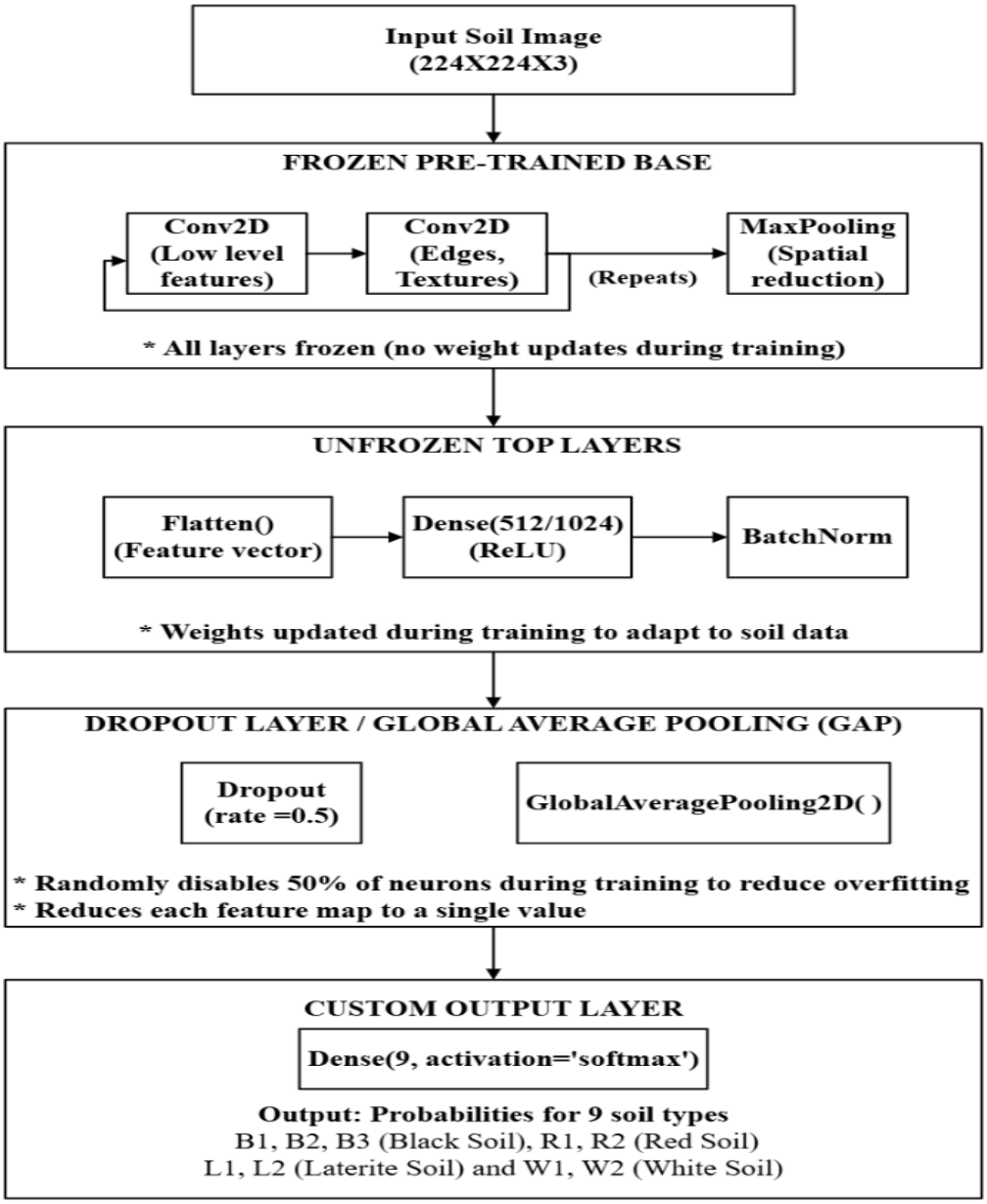


Figure 2: Architecture of VGG19 and ResNet50

- Compute batch mean

$$\mu B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad (12)$$

- Compute batch variance

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \quad (13)$$

- Normalize

$$\hat{x_i} = \frac{x_i - \mu B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (14)$$

- Efficient Grid Size Reduction: Optimized pooling and convolutional operations used for down sampling while minimizing information loss.

$$Y = Concat\begin{pmatrix} Conv_{3\times3}(X; stride = 2), \\ MaxPool_{3\times3}(X; stride = 2) \end{pmatrix} \quad (15)$$

Here, one branch applies a 3 × 3 convolution with stride two. Another applies 3 × 3 max pooling with stride two. The results concatenated along the depth dimension.

The architecture employs a multi-branch structure, where input features processed at multiple scales using parallel convolutional kernels. This allows the network to capture both fine-grained details and broader contextual information critical for distinguishing subtle differences in soil textures and patterns.

For this paper, the following modifications made to adapt InceptionV3 to the soil classification task.

- The final fully connected (FC) layer was replaced with a new dense layer consisting of nine output neurons, corresponding to the nine soil classes such

as B1, B2, B3 for black soil, R1, R2 for red soil, L1, L2 for laterite soil and W1, W2 for white soil.

- A global average pooling (GAP) layer is used prior to the classification layer to decrease the parameter number and minimize overfitting.
- The output layer is subjected to a SoftMax activation function for multi-class classification.

During training, the lower convolutional layers frozen to keep pre-trained ImageNet features, while the higher layers fine-tuned using the soil dataset to improve feature extraction for soil-specific characteristics.

3.2.2 ResNet50 Architecture

ResNet50 is a 50-layer deep residual network that addresses the vanishing gradient problem commonly met in very deep architectures. The key innovation in ResNet50 is the use of residual connections, which allow layers to learn residual mappings rather than trying to fit complex transformations directly. This enables deeper networks trained effectively without degradation in performance. ResNet50 consists of,

- For the initial feature extraction, a max-pooling layer comes after a first convolutional layer.
- Four residual blocks, each including convolutional layers with identity or projection shortcuts to enable gradient flow.
- A fully linked classification layer comes after a global average pooling layer.

Adapt ResNet50 for the soil classification task, the following adjustments made.

- The original classification layer (1,000 ImageNet classes) replaced with a custom dense layer of nine neurons, corresponding to the soil types.
- The model's top layers fine-tuned while freezing the earlier layers to control learned low-level features such as edges, textures, and shapes.
- A dropout layer, which randomly deactivates neurons during training, is introduced before the last dense layer to reduce overfitting.

3.2.3 Transfer Learning and Fine-Tuning

Both models initialized with weights pre-trained on the ImageNet dataset to receive help from feature representations learned from millions of diverse images. The transfer learning process involved.

3.2.3.1 Model Initialization

Loading the base architecture InceptionV3 or ResNet50 with pre-trained weights and freezing the first layers to preserve generic feature extraction capabilities.

3.2.3.2 Custom Layer Integration

Adding a global average pooling layer, dense layers, and a SoftMax output layer tailored to the nine soil classes. Incorporating regularization techniques such as dropout to reduce overfitting.

3.2.3.3 Fine-Tuning

Unfreezing higher-level convolutional layers for fine-tuning with the soil dataset and using a lower learning rate to prevent catastrophic forgetting of pre-trained weights.

3.2.4 Model Training Configuration

Both the InceptionV3 and ResNet50 models trained using a consistent set of training configurations designed to ensure stable convergence and best performance. The following elements carefully chosen and fine-tuned based on empirical testing and best practices in deep learning.

3.2.4.1 Loss Function Categorical Cross-Entropy

This loss function used for multi-class classification work, making it suited for the soil classification problem. It measures the dissimilarity between the predicted probability distribution and the actual class labels. By minimizing this loss, the model learns to improve its predictions over time. Since the dataset holds more than two soil categories, categorical cross-entropy is more proper than binary alternatives.

$$Loss = -\sum_{i=1}^{C} y_i \, log(\hat{y}_i) \tag{16}$$

Where,

- C = number of classes
- $y_i \epsilon \{0,1\}$ = ground truth for class i
- $\hat{y}_i \epsilon [0,1]$ = predicted probability for class i from the softmax output

Here, if the label is class 3, then $y = [0,0,1,0 \ldots,0]$. Only the log probability for the correct class contributes to the loss. Lower loss means the predicted probability $\hat{y}_i$ for the class is closer to one.

3.2.4.2 Optimizer Adam (Adaptive Moment Estimation)

3.2.4.3 The Adam optimizer selected due to its efficiency and ability to manage sparse gradients and noisy data. By adaptively adjusting the learning rate for each parameter, it combines the benefits of two other well-known optimizers, AdaGrad and RMSProp. When the validation loss plateaued, a learning rate scheduler was optionally applied to reduce the learning rate, which helped to fine-tune convergence in after epochs.

3.2.4.4

- Compute biased first moment (mean)

$$mt = \beta_1 . m_{t-1} + (1 - \beta_1) . g_t \tag{17}$$

- Compute biased second raw moment (uncentered variance)

$$v_t = \beta_2 . v_{t-1} + (1 - \beta_2) . g_t^2 \tag{18}$$

- Bias correction

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \qquad \widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{19}$$

- Update parameters

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \tag{20}$$

- Default Hyperparameters Values are,
  $i)\ \alpha = 0.001$ $iii)\ \beta_1 = 0.9$
  $ii)\ \beta_2 = 0.999$ $iv)\ \epsilon = 10^{-8}$

3.2.4.5 Learning Rate

The initial learning rate carefully tuned through experimentation, typically starting with values in the range of $\mathbf{1 \times 10^{-4}}$ ***to*** $\mathbf{1 \times 10^{-3}}$ Too high a learning rate can cause the model to overshoot minima, while too low a rate slows down

convergence. The best learning rate selected based on validation accuracy and training loss trends.

3.2.4.6 Batch Size

The batch size decided based on the available GPU memory. Common choices included 32 and 64. A larger batch size can stabilize gradient updates and speed up training, while a smaller batch size may lead to better generalization but with more noise during updates. The batch size is thirty-two selected to balance training speed with memory constraints.
Here, N is total number of training samples, B is batch size, and I is number of iterations per epoch.

$$I = \left\lceil \frac{N}{B} \right\rceil \tag{21}$$

Where,

- $\lceil \ \rceil$ = ceiling function for rounds up to the nearest integer
- Each iteration processes one batch
- One epoch is one pass over the entire dataset i.e., I iterations.

3.2.4.7 Number of Epochs

The models trained for up to 10 epochs, depending on early stopping criteria. Early stopping used to halt training when the validation accuracy stopped improving for a set number of consecutive epochs, thus preventing overfitting.

4. RESULT AND DISCUSSION

When the distribution of classes is not balanced, the F1-Score is a harmonic mean of accuracy and recall becomes especially useful. Recall indicates the model's ability to recognize all relevant instances, accuracy establishes the predicted' overall correctness, and precision establishes the number of accurate positive predictions for each class. These metrics offer a comprehensive view of each model's effectiveness across all soil classes.

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Where,

- The quantity of instances that correctly predicted to be positive is known as True Positives (TP).
- The number of instances that were mispredicted as positive is known as false positives (FP).

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

Where,

- The number of instances that correctly predicted to be positive is known as True Positives (TP).
- The number of instances that mispredicted as negative is known as False Negatives (FN).
-

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{24}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{25}$$

Where,

- True Positives (TP): the number of positive classifications that correctly predicted.
- True Negatives (TN): the number of negative classes that correctly predicted.
- False Positives (FP): the number of positive classifications that mispredicted.
- False Negatives (FN): the number of negative classes that mispredicted.

Figure 3 displays the accuracy vs. epochs for the InceptionV3 model on both the training and test datasets throughout 10th epochs.

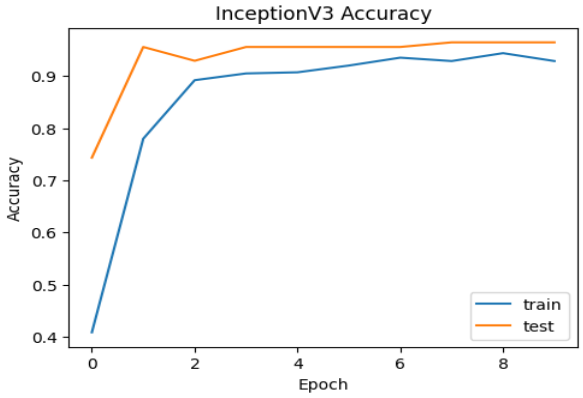


Figure 3: Graph of Training Accuracy in InceptionV3

Training Accuracy starts low at around 40%, showing the model is initially making incorrect predictions. Model improves rapidly, reaching 90% by epoch three, showing effective learning. From epochs 4 to 9, the training accuracy gradually increases and stabilizes around 93%.

Test Accuracy starts high at 75%, jumps sharply up to 96% at epoch 1, and is still consistently high above 96% throughout all epochs. Slight upward trend but minimal change between epochs 3 and 9 showing the model generalizes well from the start.

Figure 4 shows the loss vs. epochs for the InceptionV3 model during training and testing. Loss measures the gap between expected and real labels, with smaller values indicating superior performance.

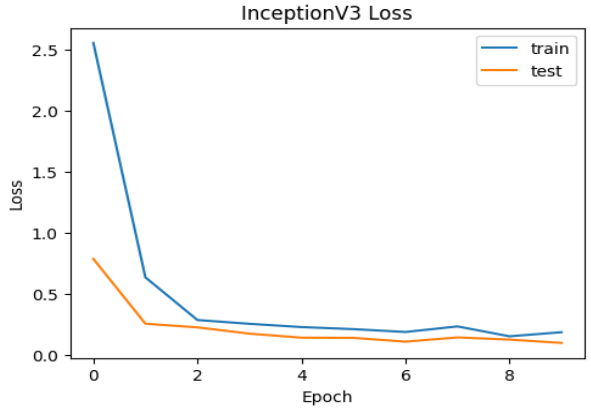


Figure 4: Graph of Training Loss in InceptionV3

Training Loss epoch 0 Starts high at around 2.6, which expected for untrained models. Epoch 1 to 2 sharp drop to around 0.4, showing rapid learning. Epochs 3 to 6 gradual decline in loss 0.2, showing stable training. Epochs 7 to 9 slight fluctuations, but low loss 0.1 to 0.2, showing the model is close to best performance.

Evaluate Loss epoch zero begins at 0.8, already lower than training loss due to better initialization from pre-trained weights. Epochs 1 to 3 rapid decrease to 0.2, like training loss. Epochs 4 to 9 continues improving slightly, eventually reaching a minimum around 0.08, and being still stable.

The accuracy vs. epochs for the ResNet50 model across 10th epochs is displayed in Figure 5 for both the training and test datasets.

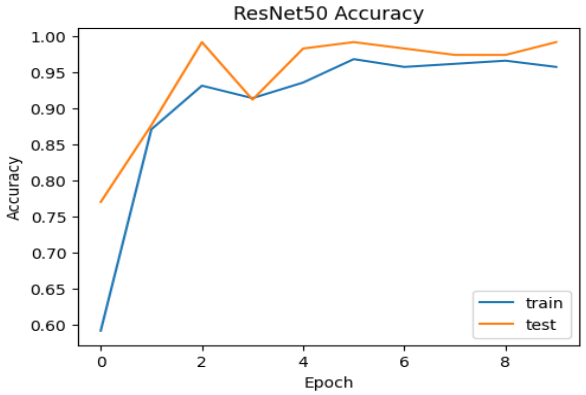


Figure 5: Graph of Training Accuracy in ResNet50

Training Accuracy starts at a low accuracy to below 0.60 in the first epoch. Shows a significant and rapid increase in accuracy during the first from epochs zero to around epoch two. This shows that the model is learning effectively from the training data. Continues to increase gradually over the next epochs, eventually reaching a high accuracy around 0.99 by the end of the 10th epoch. This suggests that the model has learned the training data quite well.

Testing Accuracy starts at a higher accuracy around 0.77 compared to the first training accuracy.

Also shows an increase in accuracy in the early epochs, peaking around epoch two with an accuracy close to 0.99. After the peak, the testing accuracy fluctuates and even slightly decreases in between epoch 2 and 3. While it is still high, it does not reach the same level as the training accuracy and shows a tendency to plateau or slightly decrease in the later epochs.

Figure 6 shows the loss vs. epochs for the ResNet50 model during training and testing. Loss measures the gap between expected and real labels, with smaller values indicating superior performance.

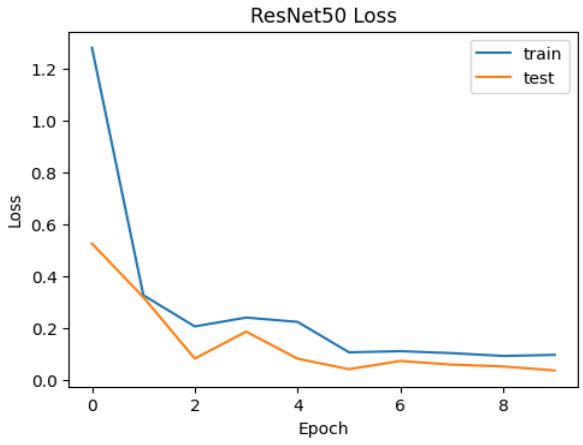


Figure 6: Graph of Training Loss in ResNet50

Training Loss starts with a high loss in the first epoch (around 1.3), showing a significant first error. Shows a sharp and rapid decrease in loss during the first epochs (from epoch zero to around epoch two). This signifies that the model is quickly learning and reducing the errors on the training data. Continues to decrease gradually over the next epochs, reaching a low loss value (below 0.1) by the end of the 10th epoch. This suggests that the model has become incredibly good at predicting the target values for the training data.

Testing Loss starts with a lower loss (around 0.5) compared to the first training loss. Decreases significantly in the early epochs, reaching its minimum around epoch two with a loss below 0.1. This shows that the model is also generalizing well to the unseen data during this phase. After epoch two, the testing loss starts to increase slightly and fluctuates. While it is still low, the upward trend suggests that the model's performance on the unseen data is no longer improving and might be getting worse.

Four important classification measures, including accuracy, precision, recall, and F1-score, were used to evaluate the performance of the two deep learning models, InceptionV3 and ResNet50. Following training and fine-tuning with transfer learning techniques, these measures are computed on the test dataset. Table 2 presents the comparative results achieved by both models.

Table 2: ResNet50 and InceptionV3 Model Performance Comparison

Metrics	Models	
	InceptionV3	**ResNet50**
Accuracy (%)	96.46	99.12
Precision (%)	96.76	99.16
Recall (%)	96.46	99.12
F1-Score (%)	96.74	99.13

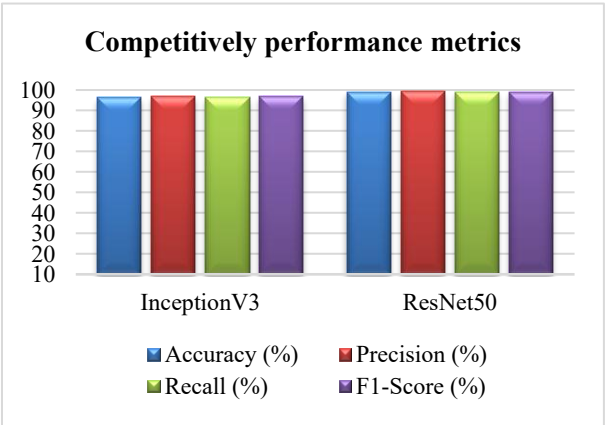


Figure 7: Performance Comparison of InceptionV3 and ResNet50 Models

The ResNet50 model consistently outperformed InceptionV3 across all evaluation metrics. It achieved a test accuracy of **99.12%**, which shows a high degree of correctness in predicting soil types. In comparison, InceptionV3 achieved a

test accuracy of **96.46%**, which, although robust, was slightly lower.

In terms of **precision**, ResNet50 reached **99.16%**, showing its ability to make reliable predictions with minimal false positives. This is especially beneficial in real-world scenarios such as precision agriculture, where overestimating the presence of certain soil types can lead to resource misallocation. InceptionV3 achieved a slightly lower precision of **96.76%**, which is still commendable and reflects stable performance.

For **recall**, ResNet50 again showed superior performance with **99.12%**, showing its ability to correctly find a high proportion of actual soil type instances. InceptionV3 achieved a recall of **96.46%**, suggesting slightly more false negatives in its predictions.

The **F1-score**, which provides a balance between precision and recall, was **99.13%** for ResNet50 and **96.74%** for InceptionV3. This confirms that ResNet50 offers a more balanced and robust classification performance, making it a more suitable candidate for deployment in environments where both precision and completeness are critical.

These results are consistent with the observed training and validation behaviour of the models. As showed in the training accuracy and loss plots see Figures 3 to 6, ResNet50 converged rapidly and kept stable generalization on the test dataset, while InceptionV3 showed good but comparatively less precise generalization. The enhanced performance of ResNet50 attributed to its deeper architecture and residual connections, which help effective learning of hierarchical features and reduce vanishing gradient issues during training.

Both models showed strong potential for soil image classification tasks. However, ResNet50 displayed a clear advantage in terms of accuracy, robustness, and generalization. These findings suggest that residual networks are better suited for handling complex image classification problems in heterogeneous and high-variance datasets such as soil images.

5. CONCLUSION

In this paper, compared two powerful deep learning models such as InceptionV3 and ResNet50 for classifying soil types based on image data. Using a well-structured and diverse dataset of soil images, both models fine-tuned through transfer learning and evaluated using key performance metrics. The results clearly show that **ResNet50** outperformed compared to **InceptionV3** in every aspect, achieving higher accuracy **99.12%**, precision, recall, and F1-score. Its architecture, which uses residual connections to support deeper learning, proved particularly effective in capturing complex patterns and subtle differences between soil types. InceptionV3 also performed well, with over **96% accuracy**, making it a strong alternative when computational resources limited. These findings highlight how deep learning, especially models ResNet50, can provide fast, dependable, and scalable solutions for soil classification that traditionally requires time-consuming and expert-driven lab analysis. This has real-world potential for improving practices in agriculture, environmental monitoring, and land-use planning. Future studies can combine RGB images with spectral data from ASD spectrometers, geospatial coordinates, and soil sensor readings such as moisture, pH, EC to improve classification accuracy. Multimodal models better capture the complex nature of soils than image-based approaches alone.

6. ACKNOWLEDGMENTS

I am grateful to my guide, Dr. Sandip Bhamare for his invaluable guidance throughout this research work. His expertise and unwavering support played a crucial role in the success of this work.

7. REFRENENCES

[1] N. C. Brady and R. R. Weil, The Nature and Properties of Soils, 15th ed. Pearson Education, 2016.

[2] P. Patel, D. Shah, and S. Patel, "Soil classification using image processing and machine learning techniques," Procedia Computer Science, vol. 192, pp. 4393–4401, 2021, doi: 10.1016/j.procs.2021.09.216.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[4] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[7] W. Zhang, H. Wang, Y. Liu, and B. Xu, "A deep learning approach to soil type classification using RGB images," Remote Sensing, vol. 14, no. 4, 1034, 2022, doi: 10.3390/rs14041034.

[8] A. Das, S. Mandal, and S. Chaudhuri, "Soil classification using texture features and neural networks," Procedia Technology, vol. 10, pp. 159–165, 2013.

[9] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Comput. Electron. Agric., vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[10] W. Zhang, H. Wang, Y. Liu, and B. Xu, "A deep learning approach to soil type classification using RGB images," Remote Sens., vol. 14, no. 4, p. 1034, 2022, doi: 10.3390/rs14041034.

[11] A. Anand, S. Malakar, and N. Dey, "Comparative performance analysis of CNN architectures for Indian soil classification," Mater. Today, Proc., vol. 62, pp. 5551–5557, 2022.

[12] V. Kumar, R. K. Sharma, and R. Kaur, "Classification of soil fertility using deep convolutional neural networks," in Proc. 2020 Int. Conf. Comput. Power Commun. Technol., pp. 58–62, doi: 10.1109/GUCON48875.2020.9231187.

[13] A. Patil, M. Yelikar, and S. Bhosale, "Ensemble deep learning approach for soil classification using image data," in Proc. 2021 Int. Conf. Adv. Comput. Commun. Syst., pp. 1120–1124, doi: 10.1109/ICACCS51430.2021.9441746.

[14] A. Iqbal, S. Ali, and H. Arshad, "MobileNetV2-based lightweight CNN for soil classification using image data,"

IEEE Access, vol. 9, pp. 123456–123467, 2021, doi: 10.1109/ACCESS.2021.3123456.

[15] R. Prasad, M. Singh, and P. Jain, "Soil texture classification using EfficientNet: A lightweight deep learning model," in Proc. 2021 Int. Conf. Intell. Comput. Control Syst., pp. 135–140, doi: 10.1109/ICICCS51141.2021.9432225.

[16] V. Shende and S. Jagdale, "Soil classification using DenseNet and transfer learning," Int. J. Comput. Appl., vol. 183, no. 47, pp. 25–29, 2021, doi: 10.5120/ijca2021921412.

[17] J. Bian, L. Fang, and Y. Xu, "Multi-modal soil classification using CNN and spectral data fusion," Remote Sens. Environ., vol. 253, p. 112213, 2021, doi: 10.1016/j.rse.2020.112213.

[18] H. Taneja, A. Gupta, and P. Roy, "Soil texture classification using CNN with GAN-based data augmentation," in Proc. 2020 Int. Conf. Pattern Recognit. Mach. Learn., pp. 322–327, doi: 10.1109/PRML2020.9243085.

[19] M. Alam and S. Roy, "Domain-adaptive convolutional neural networks for soil image classification across geographic regions," Comput. Electron. Agric., vol. 191, p. 106498, 2021, doi: 10.1016/j.compag.2021.106498.

[20] A. Sethi, R. Nair, and K. Bansal, "Comparative study of CNN architectures for soil classification using RGB imagery," J. Ambient Intell. Humaniz. Comput., 2023, doi: 10.1007/s12652-023-04256-1.