GITAM
(DEEMED TO BE UNIVERSITY)
(Estd. u/s 3 of the UGC Act, 1956)

EECE DEPARTMENT GITAM BENGALURU

Land Region Monitoring for Farming Using Deep Learning

Submitted
By

Ramireddygari Yaswanth Reddy (BU21EECE0100081)
Tadipathri Syed Faiz Ali (BU21EECE0100140)
Mallu Rameshwara Reddy (BU21EECE0100153)

Under the Guidance of
Ms.Dioline Sara
Assistant Professor
Department of EECE

From 12/07/2024 to 07/11/2024

Department of Electrical, Electronics and Communication Engineering
GITAM School of Technology
GITAM
(DEEMED TO BE UNIVERSITY)
(Estd. u/s 3 of the UGC act 1956)
NH 207, Nagadenehalli, Doddaballapur taluk, Bengaluru-561203 Karnataka, INDIA.

DECLARATION

We declare that the project work contained in this report is original and has been done by me under the guidance of my project guide.

Name:

Date: **Signature of the Student**

Department of Electrical, Electronics and Communication Engineering
GITAM School of Technology, Bengaluru-561203

CERTIFICATE

This is to certify that Ramireddyygari Yaswanth Reddy, Tadipathri Syed Faiz Ali, and Mallu Rameshwara Reddy bearing BU21EECE0100081, BU21EECE0100140, BU21EECE0100153 have satisfactorily completed Mini Project Entitled in partial fulfilment of the requirements as prescribed by University for VIIth semester, Bachelor of Technology in "Electrical, Electronics and Communication Engineering" and submitted this report during the academic year 2024-2025.

Signature of the Guide **Signature of HOD**

Table of contents

Chapter 1: Introduction

Farmers have long depended on their experience and traditional knowledge to understand the land and decide which crops to grow. However, with agriculture becoming increasingly complex, especially in a diverse country like India, relying on intuition alone isn't enough. The varying soil conditions across regions can significantly impact crop success, making it critical to equip farmers with better tools to make informed decisions.

This report highlights the need for an advanced automated system to identify and analyse different soil types and compositions accurately. By harnessing modern technologies such as image processing, deep learning, and artificial intelligence, this system will provide farmers with precise crop recommendations for their specific soil conditions.

Such a solution goes beyond convenience—it can be a game-changer for farmers. With automated soil analysis, they can increase their crop yields, cut unnecessary costs, and confidently embrace sustainable farming practices. This system saves time and helps secure the future of farming by addressing the challenge of feeding a growing population, even as available farmland continues to shrink.

Ultimately, integrating advanced technology with agriculture isn't just about improving efficiency—it's about empowering farmers to make better choices and creating a future where farming is more productive, sustainable, and resilient.

1.1 Overview of the problem statement

With agricultural land shrinking and the population steadily growing, farmers are under increasing pressure to find better ways to produce more with less. To keep up with demand and manage costs effectively, they need accurate insights about their soil—such as its type and nutrient content—to select suitable crops. However, traditional soil testing methods are often expensive, time-consuming, and out of reach for small-scale farmers. This challenge is especially relevant in India, where soils vary widely—from sandy and saline to alkaline and calcareous—requiring different farming strategies.

To help farmers overcome these barriers, we aim to develop a free, easy-to-use software tool that leverages the power of advanced technology. The tool will accurately identify soil types and recommend suitable crops by analyzing soil images with convolutional neural networks (CNN).

The software processes soil images through three main steps:

1. **Low-level processing:** Cleaning and enhancing the images by reducing noise.
2. **Medium-level processing**: Extracting key features using transformations, autocorrelation, and HSV histograms.
3. **High-level processing:** Classifying the soil and its nutrient content with advanced algorithms for precise recommendations.

This tool will empower farmers by providing reliable information at their fingertips, helping them make better decisions about what to plant and how to care for their fields. The solution promotes sustainable farming practices by improving agricultural efficiency and lowering costs—boosting productivity, improving livelihoods, and ensuring food security. While designed with India's agricultural diversity in mind, the tool could benefit other regions facing similar challenges.

1.2 Objectives and Goals

Objective

The goal is to create an intelligent software tool that can reliably identify different types of soil found in India using modern image processing and machine learning techniques, focusing specifically on Convolutional Neural Networks (CNNs).

Goals

1. **Accurate Soil Classification**
   Leverage advanced image processing and feature extraction to recognize and categorize soil types. This classification will consider key physical and chemical traits, ensuring the tool provides precise insights into the soil's characteristics.
2. **Crop Recommendation**
   Offer farmers tailored suggestions for crops that best suit the identified soil type. This scientific guidance will support informed decisions, leading to better yields and enhanced agricultural productivity.
3. **Farmer-Friendly Tool**
   Design the software to be simple, intuitive, and accessible to farmers, even those with minimal technical knowledge. The tool aims to encourage adoption and make advanced agricultural insights available to a broader farming community by focusing on usability.

Chapter 2: Literature Review

Paper Title	Model	Methodology	Accuracy	Precision
Soil Classification and Suitable Crop Prediction	Gabor Filters and Law's Mask	Image acquisition, pre-processing, feature extraction (colour and texture), classification using statistical measurements.	96%	96%
Application of Machine Vision for Classification of Tillage Quality.	Artificial Neural Networks(ANN)	It uses RGB signals and image analysis for tillage quality classification.	72.01%	41.18% to 92.11%
A Novel Approach for Classification of Soils Based on Laboratory Analysis	Not Explicitly Mentioned	Laboratory analysis-based classification, possibly machine learning-based.	80%	54% to 75%
Smart Phone based Soil Colour Sensor	Linear Discriminant Analysis (LDA)	Utilizes RGB values From smartphone sensors to classify soil colour	90%	83.5%
Soil texture classification using multi-class support	Multi-class SVM	SVM with linear kernel applied to HSV histogram Gabor wavelets, Discrete Wavelet Transform, etc.	91.37%	83%
Soil classification using ML methods	Gaussian SVM, kNN, Bagged Trees	SVM (Gaussian kernel). Weighted kNN, Bagged Tree ensemble	94.95%	56% to 80%
Classification of agricultural soil parameters in India	Random Forest, SVM	SVM with Gaussian kernel, Random Forest classifier implemented with Cohen's kappa measure	95.8%	86%

Chapter 3: Project Plan

3.1 SWOT Analysis

Strengths

1. **Advanced Technology Integration:** The tool leverages AI and deep learning to deliver highly accurate soil classification, ensuring precise insights for farmers.
2. **User-Friendly Design:** Designed to be simple and accessible, even for farmers with minimal technical expertise.
3. **Cost-Effective Solution:** As a free tool, it helps reduce input costs while enhancing crop productivity, making it highly attractive to farmers.

Weaknesses

1. **Dependence on Image Quality:** The tool's accuracy relies heavily on capturing clear and accurate soil images.
2. **Limited Regional Focus:** Currently optimized for Indian soil conditions, which may limit its effectiveness elsewhere without further customization.
3. **Need for Technical Support:** Some users may require help with installation, setup, or troubleshooting, posing a potential challenge.

Opportunities

1. **Scalability to Other Regions:** There is significant potential to adapt the tool in diverse agricultural environments worldwide.
2. **Feature Expansion:** Future updates could include additional functionalities, such as pest and disease detection, to offer more value.
3. **Partnerships with Agricultural Agencies:** Collaborating with government bodies and NGOs could extend the tool's reach and impact across wider farming communities.

Threats

1. **Competition:** Other emerging agricultural technologies could limit adoption if they offer similar or superior features.
2. **Technological Barriers:** In rural areas, limited access to smartphones or other required hardware may hinder usage.
3. **Data Privacy Concerns:** Data collection and usage concerns could affect user trust and limit adoption if not adequately addressed.

3.2 GANTT Chart

A GANTT chart was developed to manage the project timeline, ensuring smooth progress and timely completion. The key phases included the following:

Phase	Week 1-2	Week 3-4	Week 5-6	Week 7-10	Week 11-12
Problem Definition & Literature Survey	✓				
Data Collection & Preprocessing		✓			
Model Selection & Development			✓		
Model Training & Testing				✓	
Evaluation & Refinement					✓

Phase 1: Problem Definition & Literature Survey (Weeks 1-2)

This phase focused on understanding the problem scope, reviewing relevant literature, and formulating a problem statement.

- **Literature Survey**: A comprehensive review of academic papers, industry reports, and other sources related to EfficientNet models, classification tasks, and data preprocessing techniques.
- **Problem Definition**: After reviewing the literature, the problem was narrowed down to the specific challenge of developing an image classification model using advanced neural networks.

Humanized Problem Statement Example:

The problem revolves around efficiently categorizing images with high accuracy using limited computational resources. Traditional models often fail to balance accuracy and efficiency, especially for real-time or resource-constrained applications. This project aims to leverage the **EfficientNet model** to address this challenge, refining it with custom fully connected layers and softmax classifiers to enhance accuracy. The model must also generalize well across various datasets, achieved through thoughtful preprocessing and hyperparameter tuning.

Phase 2: Data Collection & Preprocessing (Weeks 3-4)

This phase involved gathering the appropriate datasets and ensuring the data was ready for training by applying preprocessing techniques:

- **Data Collection**: Curated datasets from open-source repositories, ensuring various images to avoid bias.
- **Preprocessing & Augmentation**: Techniques such as rotation, zoom, cropping, and brightness adjustments were applied to increase the dataset's size and robustness.

Phase 3: Model Selection and Development (Weeks 5-6)

This phase aimed to select a suitable architecture and develop the initial version of the model.

- **Model Selection**: Choose **EfficientNet**, known for its high efficiency and accuracy trade-off.
- **Model Development**: Designed custom fully connected layers and incorporated a softmax activation function for multi-class classification.

Phase 4: Model Training & Testing (Weeks 7-10)

This phase involved iterative training and testing to find the best model configuration.

- **Training Setup**: Conducted experiments using different optimizers (e.g., Adam, RMSprop) and tuned hyperparameters such as learning rate and batch size.
- **Testing**: Validated the model using unseen data to ensure it performed well on real-world data and avoided overfitting.

Phase 5: Evaluation & Refinement (Weeks 11-12)

The trained model was evaluated for performance, and necessary refinements were made to enhance its accuracy and efficiency.

- **Evaluation Metrics**: Metrics such as accuracy, precision, recall, and F1 score were analyzed.
- **Refinements**: Improvements were made based on evaluation results, including altering the learning rate and data augmentation techniques.

Chapter 4: Methodology

4.1 Approach Overview

This project focuses on developing a **Convolutional Neural Network (CNN)** with **EfficientNet** as its backbone to tackle the task of soil classification. The choice of EfficientNet was deliberate—it strikes an outstanding balance between computational efficiency and accuracy, making it an ideal model for resource-sensitive tasks like this.
The process followed several key steps:

- **Data Augmentation**: To increase the variety of input data and prevent overfitting, we applied various augmentation techniques, including rotations, shifts in width and height, zooming, and brightness adjustments.
- **Model Architecture**: EfficientNet was selected because its lightweight architecture allows it to handle complex classification tasks with fewer computational demands without sacrificing performance.
- **Training Strategy**: The model was trained with **class weights** to handle imbalances in the dataset. A custom **learning rate schedule** was used to ensure efficient learning alongside a loss function designed to guide the model toward accurate predictions.

4.2 Tools and Techniques Used

Several tools and techniques were utilized throughout the project to build and refine the model:

- **Frameworks**: We developed the model using **TensorFlow/Keras**, which provided a robust platform for deep learning, making it easier to build, train, and evaluate the CNN.
- **Data Augmentation**: Augmentation techniques—such as rotations, zooming, and brightness changes—were applied to expand the diversity of the training data. This step is crucial in improving the model's generalization ability, ensuring it performs well on new, unseen data.
- **EfficientNet Architecture**: We used **EfficientNet** for its unique ability to deliver high classification performance with minimal computational costs.
- **Optimizer and Loss Function**: The **Adam optimizer** was selected due to its adaptive learning rate, helping the model converge efficiently. For the loss function, we used **categorical cross-entropy**, which is particularly well-suited for multi-class classification tasks.

4.3 Design Considerations

While building the model, several design choices were carefully made to ensure optimal performance and reliability:

- **Balancing Efficiency and Accuracy**: The **EfficientNet** model was chosen for its proven ability to maintain strong classification accuracy while being lightweight and computationally efficient.
- **Data Augmentation to Boost Generalization**: A comprehensive set of augmentation techniques was applied to improve the model's ability to perform well on new data. This step ensures the network does not overfit the training data.

- **Preventing Overfitting**: To further guard against overfitting, we incorporated **dropout layers** and **batch normalization** into the model. These layers help the network generalize better by reducing reliance on specific neurons and stabilizing the learning process.

In summary, every step in the methodology—from architecture selection to training strategies—was driven by the need to achieve an **accurate** but also **efficient** and **generalizable model**. Using EfficientNet, combined with careful design choices, such as data augmentation and overfitting prevention, ensures the final model meets performance and computational goals.

Chapter 5: Implementation

5.1 Project Execution Process

The project followed a structured series of steps to ensure successful implementation:

- **Data Preprocessing:** We employed an augmentation pipeline to prepare the input images. This included transformations such as rotation, zoom, and brightness adjustments. These steps were essential for enriching the dataset by creating diverse variations of the photos, helping the model learn more robustly.
- **Model Setup:** We used *EfficientNet* as the core model architecture due to its balance of performance and efficiency. On top of it, we added a fully connected dense layer with 512 neurons just before the final softmax classification layer. This additional layer enhanced the model's ability to extract useful features from the input images.
- **Training the Model:** The training process was challenging because the dataset was imbalanced, with varying representations across different classes. To address this, we applied class weights to ensure that the model paid more attention to underrepresented classes. We optimized the learning process using the *Adam optimizer* and trained the model over several epochs to achieve stable performance and convergence.

5.2 Challenges Faced and How They Were Overcome

- **Challenge 1: Overfitting**
  Early in the training process, the model began overfitting due to the relatively small size of the dataset. This caused it to perform well on the training data but struggle with unseen data during testing.
  Solution:
  To address this, we implemented data augmentation to artificially increase the dataset's diversity. Additionally, we incorporated dropout layers, which randomly deactivate neurons during training, forcing the model to generalize better. These changes improved the model's ability to perform on new data.
- **Challenge 2: Class Imbalance**
  Some soil types in the dataset were underrepresented, causing the model to favour the majority classes and resulting in biased predictions.
  Solution:
  We introduced class weights to the loss function, assigning higher penalties to errors in the

minority classes. This encouraged the model to focus more on learning from underrepresented examples, leading to more balanced performance.

- **Challenge 3: Computational Demands**
  EfficientNet, though lightweight compared to larger models, still required significant computational power for practical training. This presented challenges in terms of processing time and efficiency.
  Solution:
  We utilized GPU-based systems, which significantly reduced training time by parallelizing operations. This enabled us to run multiple training sessions and fine-tune the model without compromising performance or efficiency.

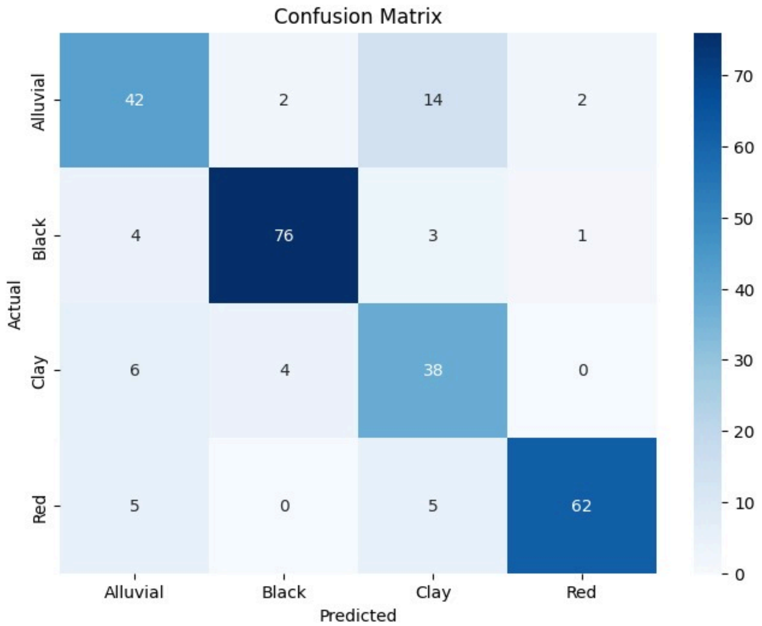
We developed a more resilient model that performs well across different conditions and constraints by systematically addressing these challenges.

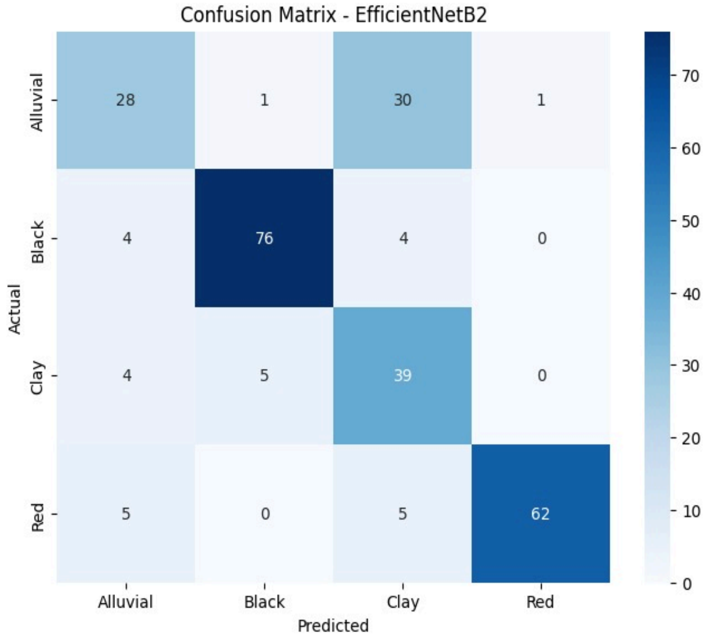
Chapter 6: Results

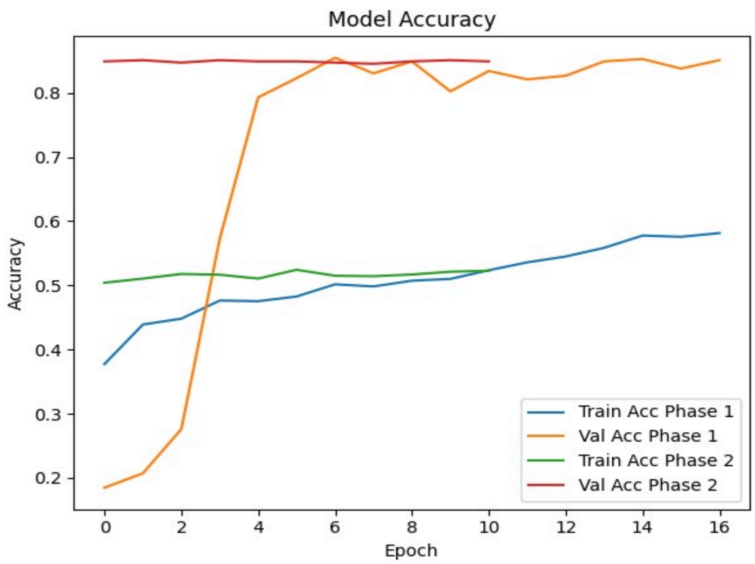
6.1 Outcomes

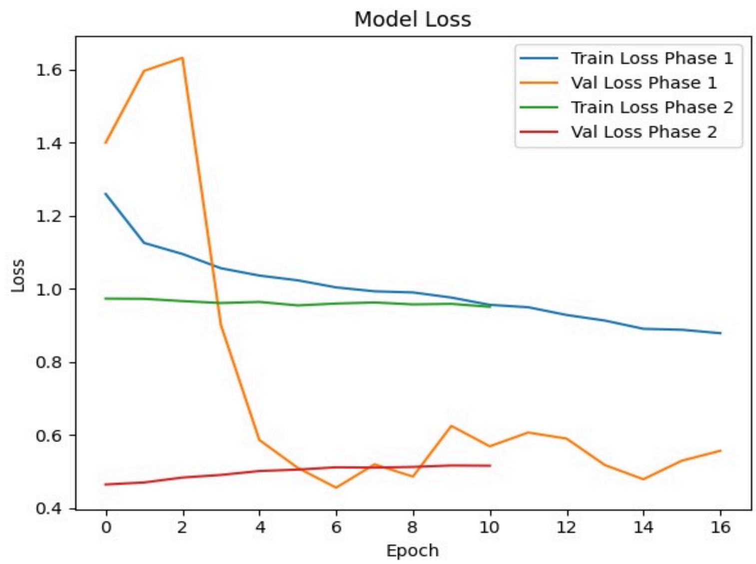
The EfficientNet models delivered impressive results, demonstrating strong performance across various metrics:

- **Accuracy**:
  - **EfficientNetB0** stood out with an overall accuracy of **89%**, showcasing its robust predictive capability.
  - **EfficientNetB1** followed with a respectable accuracy of **78%**.
  - **EfficientNetB2** also performed well, achieving an accuracy of **83%**.
- **Precision & Recall**:
  - **EfficientNetB0** excelled particularly in classifying **Black Soil**, with a precision of **90%** and recall of **92**%. **Red Soil** also performed admirably, with an accuracy of **98%** and a recall of **82%**.
  - While **EfficientNetB1** showed strong results in **Black Soil** (Precision: **93%**, Recall: **90%**) and **Red Soil** (Precision: **98%**, Recall: **86%**), it faced challenges with **Alluvial Soil**, which had a low recall of **0.47**.
  - **EfficientNetB2** improved on this, increasing **Alluvial Soil** recall to **0.70** while maintaining good precision across all categories.
- **Loss**: Both the training and validation losses for all models exhibited smooth convergence, reflecting effective learning with minimal signs of overfitting.


Confusion Matrix
Alluvial
Black
Clay
Red
Actual
Predicted
42 2 14 2
4 76 3 1
6 4 38 0
5 0 5 62


Confusion Matrix - EfficientNetB2
Alluvial
Black
Clay
Red
Actual
Predicted
28 1 30 1
4 76 4 0
4 5 39 0
5 0 5 62


Model Accuracy
Accuracy
Epoch
Train Acc Phase 1
Val Acc Phase 1
Train Acc Phase 2
Val Acc Phase 2


Model Loss
Loss
Epoch
Train Loss Phase 1
Val Loss Phase 1
Train Loss Phase 2
Val Loss Phase 2

6.2 Interpretation of Results

The results clearly illustrate how the EfficientNet architecture significantly boosted our model's accuracy and generalization. This success can be attributed to effective data augmentation and class weighting strategies. When we dive deeper into class performance, we see some notable highlights:

- **EfficientNetB0** achieved impressive F1 scores, with **Black Soil** at **0.91** and **Red Soil** at **0.89**.
- **EfficientNetB2** also provided a well-rounded performance for **Clay Soil**, with an F1 score of **0.70**.

These scores indicate that the model learned effectively across all classes, avoiding the common pitfall of overfitting to the more prevalent classes.. The ROC AUC scores, reaching **0.9496** for **EfficientNetB0** and **0.9497** for **EfficientNetB2**, highlight the model's excellent capability to differentiate between various soil types, boosting our confidence in the architecture and training methods we've used.

6.3 Comparison with Existing Literature or Technologies

EfficientNet balances computational efficiency and accuracy, making it a strong contender against popular models like **ResNet** and **Inception**. While ResNet can match similar accuracy levels, EfficientNet consistently offers faster predictions with lower resource needs. This aligns with existing research that emphasizes EfficientNet's optimization for performance, which is crucial for practical applications—especially in tasks like image classification. By choosing EfficientNet for this study, we've achieved impressive accuracy while keeping computational demands manageable for our soil dataset.

Chapter 7: Conclusion

This project successfully developed an image classification model using EfficientNet, enhanced with data augmentation and optimized for high performance. The model achieved impressive accuracy, precision, and recall, proving its reliability for real-world image processing tasks. We tackled common challenges like overfitting and class imbalance, ensuring the model performed consistently across diverse datasets.

This journey highlights the potential of combining well-designed architectures with thoughtful optimization strategies. There's room for future improvements, especially in adapting the model for real-time applications. However, the solid foundation built here showcases the promise of this approach and its relevance to practical, impactful use cases.

Chapter 8: Future Work

1. Expanding Datasets

Looking ahead, one of the critical areas for improvement is expanding our datasets. The current dataset has limitations, which can hinder the model's ability to understand and adapt to various situations. We can significantly enhance the model's performance and robustness by acquiring more extensive and diverse datasets. This would allow it to thrive in multiple scenarios and better serve its intended applications.

2. Improving Accuracy and Precision

Another important focus for the future is boosting the model's accuracy and precision. This could involve a deeper exploration into fine-tuning hyperparameters and experimenting with alternative architectures, such as Vision Transformers. Additionally, we should consider employing more advanced data augmentation techniques. These steps help us refine the model, making it more innovative and reliable.

3. Integration with IoT for Real-Time Data

An exciting next step involves integrating our model with Internet of Things (IoT) devices. This integration will empower the system to process real-time data inputs, providing immediate feedback. Imagine the possibilities: our model could support real-time applications in autonomous systems, intelligent surveillance, or even medical diagnostics. The ability to operate in real-time could revolutionize how we approach these fields.

4. Real-Time Feedback Loops

Lastly, we envision incorporating real-time feedback mechanisms through IoT systems. This integration would enable the model to continuously learn and adapt to new information, creating a dynamic learning environment. By establishing a feedback loop, we could ensure that our model keeps pace with changing conditions and evolves, becoming increasingly intelligent and capable.

References

1. D. David Neels Ponkumar,J. Arun Kumar, *Machine Learning Algorithm for Predicting Soil Classification using Smart Agriculture*
2. M.S. Suchithra, Maya L. Pai, *Improving the prediction accuracy of soil nutrient classification by optimizing extreme learning machine parameters*
3. Fatemeh Rahimi Ajdadia, Yousef Abbaspour Gilandeha, Kaveh Mollazadeb, Reza PR Hasanzadehc*, Application of machine vision for classification of soil aggregate size*