**B. Tech Project Report**

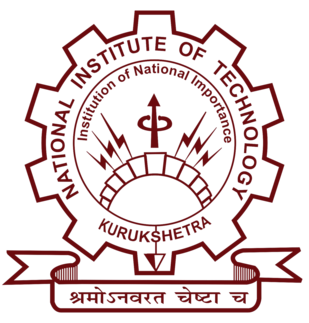
**On**

**Crop Fertilizer Recommendation and Plant Disease Detection Using ResNet**

**By**

**Aadarsh (12112120)**

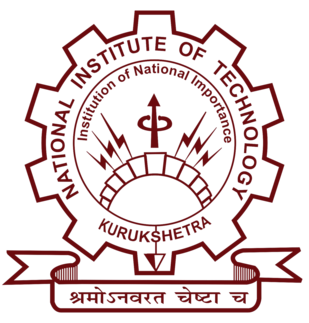
**Under the Supervision of**

**Mr. Ankit Kumar Jain (Asst. Prof.)**

**Department of Computer Engineering**

**National Institute of Technology, Kurukshetra**

**Haryana-136119, India Jan-April 2024**



CERTIFICATE

I hereby certify that the work which is being presented in this Internship project report, entitled “**Crop Fertilizer Recommendation and Plant Disease Detection Using ResNet**”, in partial fulfillment of the award of the Bachelor of Technology in Computer Engineering is an authentic record of my own work carried out during a period from January 2024 to April 2024, under the supervision of Dr. Ankit Kumar Jain, Professor, Computer Engineering Department, NIT Kurukshetra.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

Signature of candidate

Aadarsh 12112120

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of supervisor

**Mr. Ankit Kumar Jain Assistant Professor**

|  |  |  |
| --- | --- | --- |
| **Sno** | **Description** | **Pg No.** |
| 1 | Abstract | 4 |
| 2 | Motivation | 5 |
| 3 | Introduction | 6 |
| 4 | Proposed Work | 7 |
| 5 | Flow of Application | 8-9 |
| 6 | Prediction | 10-11 |
| 7 | Feature Extraction | 11-12 |
| 8 | Validation Accuracy | 13-14 |
| 9 | Validation Loss | 14-15 |
| 10 | Learning Rate Over Time | 16-17 |
| 11 | ResNet Framework | 18-21 |
| 12 | Architecture Of CNN | 22 |
| 13 | Crop Recommendation System | 23-24 |
| 14 | Fertilizer Recommendation System | 25 |
| 15 | Libraries Used | 26 |
| 16 | Comparison With Other Algorithms | 26-27 |
| 17 | Limitations | 28 |
| 18 | Advantages | 29 |
| 19 | Gantt Chart | 30 |
| 20 | References | 31 |

# ABSTRACT

Agriculture is a vital industry that sustains the livelihoods of millions globally. However, the industry faces challenges such as optimizing crop selection, ensuring appropriate fertilizer use, and effectively managing plant diseases. To address these challenges, AgriSmart, a comprehensive web application, has been developed. AgriSmart aims to assist farmers by providing tailored solutions through three primary services: the Crop Recommendation System, the Fertilizer Suggestion System, and the Disease Detection System.

The Crop Recommendation System leverages machine learning algorithms to analyze soil nutrient values (NPK), geographic data (state and city), and real-time weather information. It recommends the most suitable crops for cultivation, helping farmers make informed decisions that enhance yield and sustainability. The Fertilizer Suggestion System complements this by offering precise fertilizer recommendations based on soil nutrient content and the selected crop, ensuring balanced nutrient management for optimal growth.

The core innovation of AgriSmart is the Disease Detection System, which utilizes a deep learning model based on the ResNet (Residual Networks) architecture. This system accurately identifies plant diseases from images of leaves, providing diagnoses, and offering detailed reports that include preventive measures and treatment options. The development of this system involved extensive data collection, preprocessing, model training, and validation, making it the most effort-intensive component of the project.

AgriSmart is built with a robust technology stack: HTML, CSS, JavaScript, and Bootstrap for the frontend; Flask and Python for the backend; and libraries like NumPy, Pandas, Matplotlib, Scikit-learn, and PyTorch for machine learning and data analysis. The application is deployed on Heroku and maintained using Git for version control. By integrating these advanced technologies, AgriSmart aims to revolutionize agricultural practices, empowering farmers with the tools to improve productivity, sustainability, and crop health.

# MOTIVATION

The motivation behind AgriSmart is to address key challenges in agriculture, such as optimizing crop selection, managing soil nutrients, and detecting plant diseases. Farmers often face difficulties in making data-driven decisions due to limited access to advanced tools and expertise. AgriSmart aims to bridge this gap by providing accessible, technology-driven solutions.

The Disease Detection System is particularly critical, offering an easy-to-use tool for accurately diagnosing plant diseases using deep learning. This helps farmers take timely action, minimizing crop losses. Additionally, the Crop Recommendation and Fertilizer Suggestion Systems guide farmers in selecting suitable crops and managing soil health, improving productivity and sustainability.

AgriSmart is driven by the goal of empowering farmers with the knowledge and tools needed to enhance their agricultural practices, ultimately leading to increased yields and improved livelihoods.

# INTRODUCTION

Agriculture plays a vital role in sustaining the global population, yet it faces numerous challenges that threaten productivity and sustainability. Among these challenges are the proper selection of crops, effective soil nutrient management, and the timely detection and treatment of plant diseases.

These issues are particularly pressing in regions where farmers lack access to modern tools and data-driven insights, leading to suboptimal agricultural practices and reduced yields.

AgriSmart was developed to address these challenges by integrating advanced technologies into a single web application. The platform provides farmers with three essential services: a Crop Recommendation System, a Fertilizer Suggestion System, and a Disease Detection System.

Each of these features is designed to empower farmers with the knowledge and tools needed to make informed decisions that enhance productivity and sustainability.

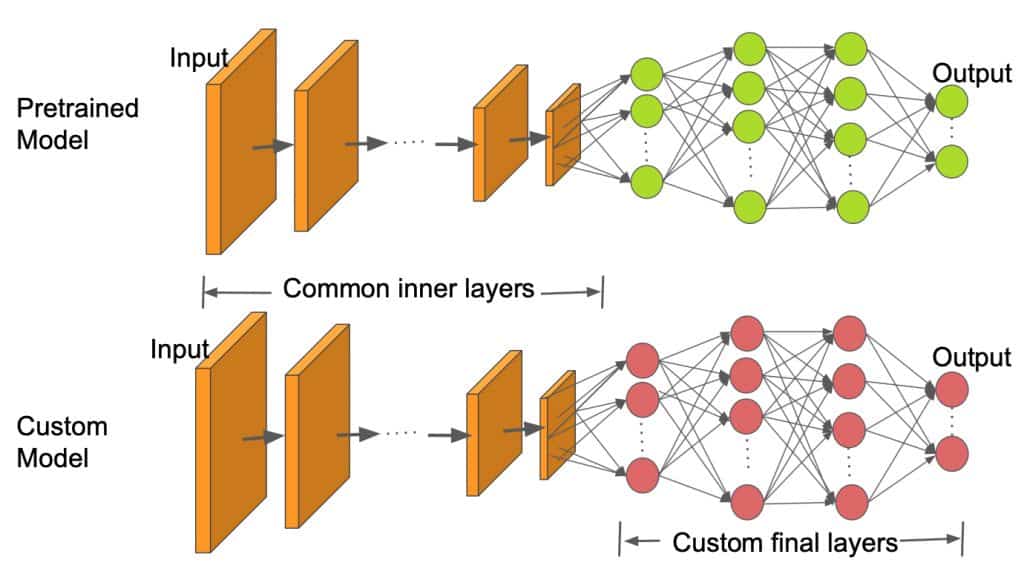
This project represents a significant step forward in the use of technology to support sustainable agriculture. By providing farmers with easy access to sophisticated tools and data, AgriSmart aims to improve crop yields, enhance resource efficiency, and promote overall agricultural resilience.

# PROPOSED WORK

Residual Neural Network (ResNet) is a type of deep neural network architecture that uses a deep residual learning framework to address challenges in training deep neural networks. ResNets are made up of a stack of residual layers, each with a shortcut connection that allows researchers to avoid costly computing gradients. This allows the network to adapt to the residual mapping instead of the layers learning from the underlying mapping.

ResNets were introduced in 2015 by Kaiming He et al. in their paper, “Deep Residual Learning for Image Recognition”. They are a type of convolutional neural network (CNN) architecture that was designed to address the vanishing and exploding gradient problems that often occur with deeper networks.

ResNets come in various depths, such as ResNet-18, ResNet-32, and ResNet-50. ResNet-50 is a mid-sized variant that was developed by researchers at Microsoft Research Asia. It is known for its depth and efficiency in image classification tasks.



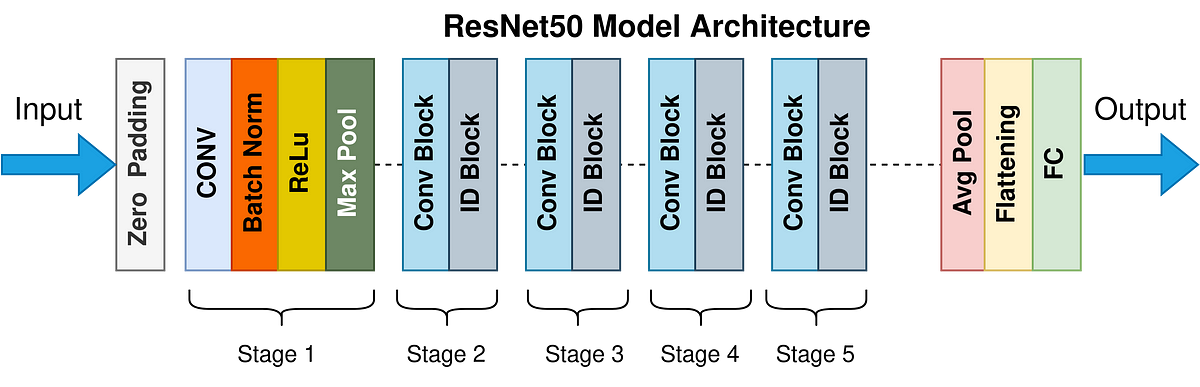
*Fig. 1 ResNet Layers*

**Flow of Application**

The AgriSmart application is designed to provide farmers with a seamless experience, ensuring they can easily access the three core services: Crop Recommendation, Fertilizer Suggestion, and Disease Detection. The user journey begins at the homepage, where users are introduced to the application’s capabilities. From here, users can navigate to the specific service they require.

**Each service follows a distinct flow:**

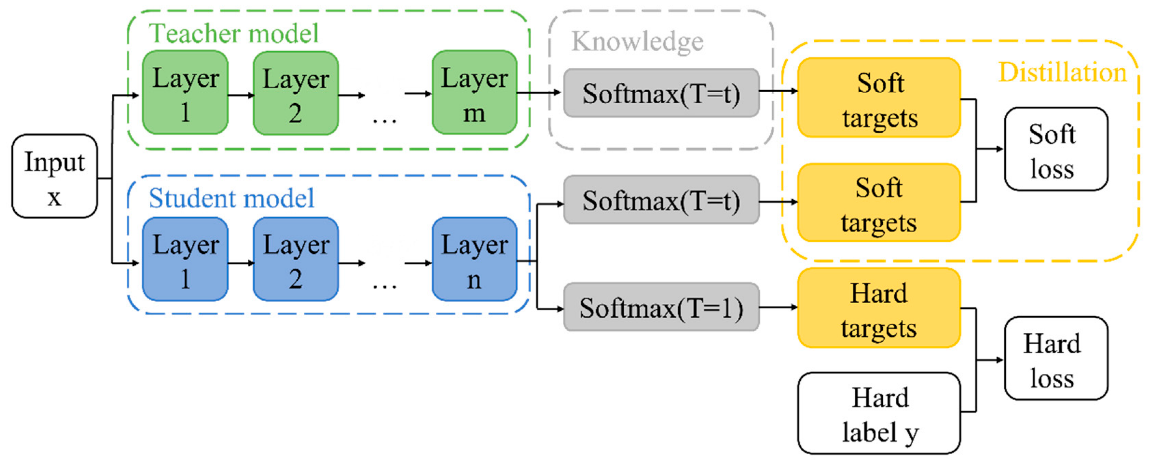
1. **User Input:** The process begins with the farmer providing necessary inputs. For the Crop Recommendation System, the farmer inputs soil nutrient values, state, and city. For the Fertilizer Suggestion System, soil nutrient content and the desired crop are required. For Disease Detection, the farmer uploads an image of the affected plant leaf.



*Fig. 2 Flow of Application*

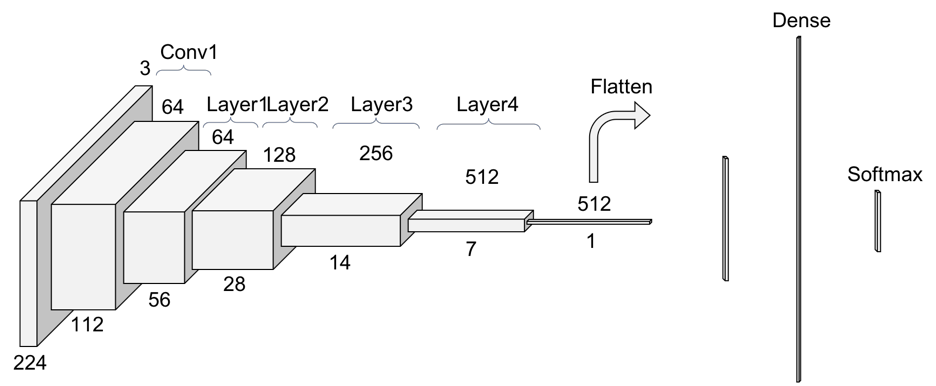
2. **Data Processing**: Once inputs are received, the application processes the data. For Crop Recommendation and Fertilizer Suggestion, this involves analyzing the soil nutrient data in conjunction with weather data retrieved from a reliable API. For Disease Detection, the system preprocesses the image, preparing it for analysis by the deep learning model.

3. **Model Invocation**: Each service then invokes the appropriate machine learning model. The Crop Recommendation and Fertilizer Suggestion systems rely on predictive models that assess input data to provide relevant suggestions. The Disease Detection System, however, engages the ResNet deep learning model, which performs feature extraction and disease diagnosis.



*Fig. 2.1 Invocation*

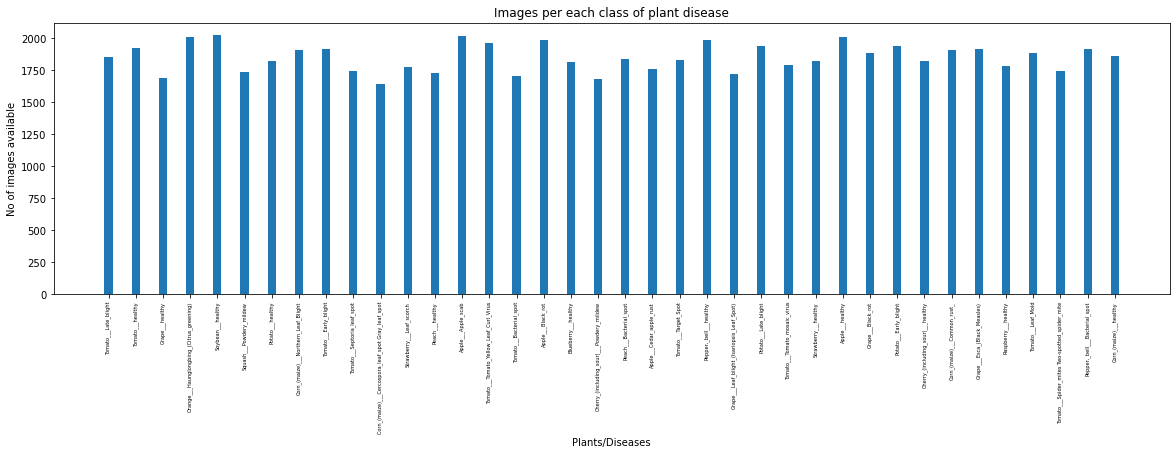
4. **Output Presentation:** Finally, the system presents the results to the user. The Crop Recommendation System suggests the best crop to cultivate; the Fertilizer Suggestion System provides a detailed fertilizer plan; and the Disease Detection System offers a comprehensive diagnosis of the plant’s health, including prevention and cure suggestions if a disease is detected.



*Fig. 2.2 Output*

**Prediction:**

Prediction is a critical component of AgriSmart, particularly within the Crop Recommendation and Fertilizer Suggestion systems. These systems are designed to provide accurate and actionable recommendations based on the analysis of input data.



*Fig. 3 Images per each class of plant disease*

1. **Crop Recommendation System:**

- The system utilizes a machine learning model trained on a large dataset that includes soil nutrient values, weather patterns, and crop yield statistics.

- The model processes the input data—NPK values, state, and city—to predict the optimal crop for cultivation.

- By integrating weather data, the system ensures that the crop suggestion is tailored to the local climate conditions, maximizing the potential yield and minimizing risk.

2. **Fertilizer Suggestion System:**

- This system predicts the appropriate fertilizer mix by analyzing the nutrient profile of the soil in conjunction with the intended crop.

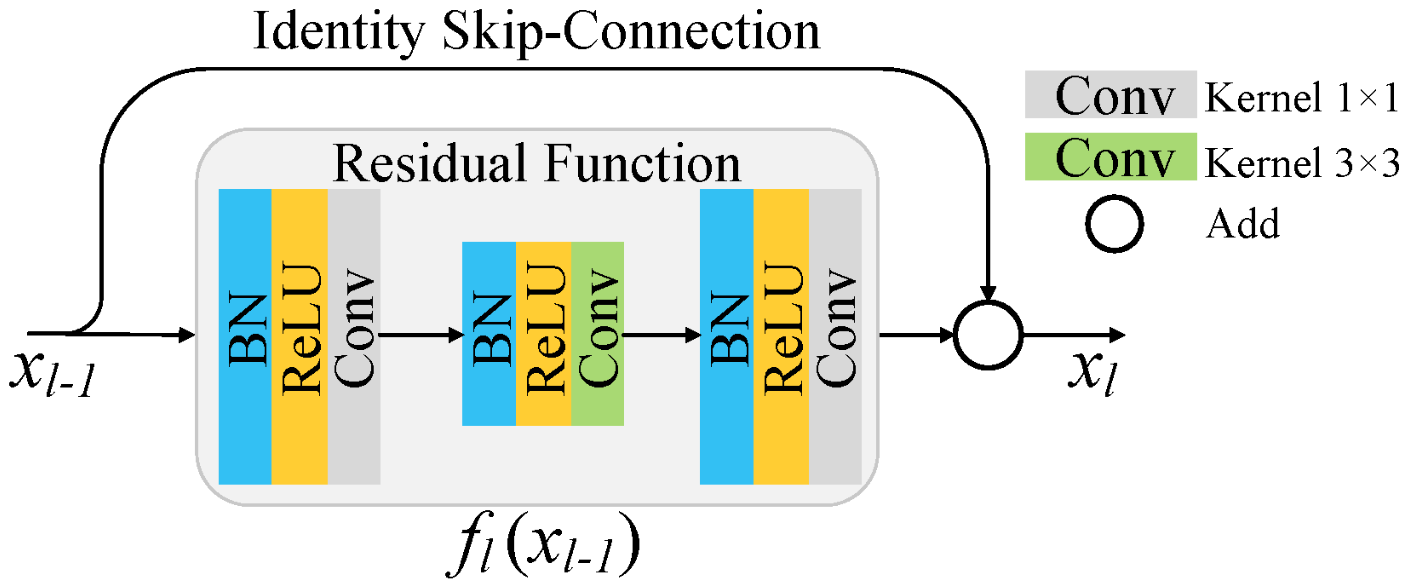
- The machine learning model is trained on data that correlates soil nutrient deficiencies with specific fertilizer types, enabling it to provide precise fertilizer recommendations.

- The output not only addresses nutrient deficiencies but also considers potential nutrient excesses, ensuring that the soil remains balanced and fertile.

Both systems employ predictive algorithms that have been fine-tuned to handle the variability in soil and weather conditions across different regions, making AgriSmart a versatile tool for farmers across diverse agricultural environments.

**Feature Extraction:**

Feature extraction is a pivotal step in the Disease Detection System of AgriSmart, where the goal is to identify and isolate the most significant aspects of the input image that indicate plant health or disease.



*Fig. 4 Residual Function*

1. **Image Preprocessing:**

- Before feature extraction, the uploaded leaf image undergoes preprocessing to enhance its quality. This may include resizing, normalization, and noise reduction to ensure the image is in the optimal format for analysis.

- Preprocessing is crucial as it standardizes the input, enabling the deep learning model to focus on the relevant features without being misled by irrelevant variations in the image.

2. **Feature Extraction Process:**

- The ResNet model is employed for feature extraction. ResNet is particularly effective because of its ability to identify intricate patterns within images through its deep network architecture.

- During this process, the model extracts features such as texture, color, and shape, which are critical indicators of plant health. These features are then used to differentiate between healthy and diseased plants.

- The extracted features are fed into the classification layers of the model, where they are analyzed to determine the presence of disease.

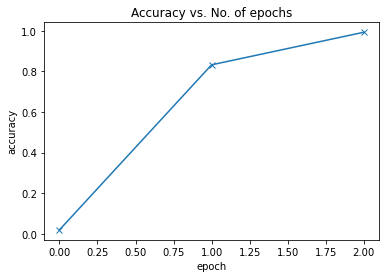
3. **Importance of Feature Extraction:**

- Feature extraction is essential because it condenses the image data into a set of representative attributes that are easier to analyze. This not only improves the accuracy of the disease diagnosis but also accelerates the processing time, making the system responsive and efficient.

- By focusing on the most significant features, the model can more accurately distinguish between different diseases, even when the visual differences are subtle.

**Validation Accuracy**

Validation accuracy is a critical metric used during the training of the deep learning model to assess its performance on unseen data. In the context of AgriSmart's Disease Detection System, validation accuracy represents the proportion of correctly classified plant leaf images in the validation set.

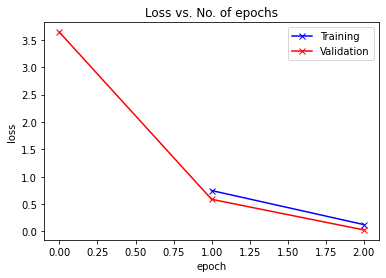


*Fig. 5.1 accuracy VS epoch*

1. **Monitoring Model Performance:**
   * During the training process, the model's performance is evaluated at regular intervals using a separate validation set, which is distinct from the training data. This helps in assessing how well the model generalizes to new, unseen data.
   * High validation accuracy indicates that the model is correctly identifying plant diseases in images it has not been trained on, suggesting that it has learned relevant features rather than just memorizing the training data.
2. **Trends in Validation Accuracy:**
   * Initially, as the model begins learning, the validation accuracy tends to increase rapidly. However, as training progresses, improvements in accuracy might slow down and eventually plateau, indicating that the model has reached its learning capacity for the given data and architecture.
   * If validation accuracy starts to decline while training accuracy continues to increase, it may suggest that the model is overfitting, meaning it is becoming too specialized to the training data and losing its ability to generalize.

**Validation Loss**

Validation loss is another crucial metric used to evaluate the model’s performance, complementing validation accuracy. While accuracy measures the proportion of correct predictions, validation loss quantifies the difference between the predicted outputs and the actual outputs in the validation set.

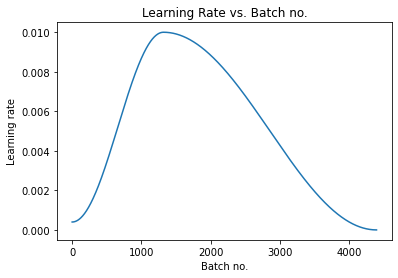


*Fig. 5.2 loss VS epoch*

1. **Understanding Loss:**
   * Loss functions, such as cross-entropy loss for classification tasks, are used to penalize incorrect predictions. The goal during training is to minimize this loss function.
   * Validation loss provides insight into how well the model's predictions align with the true labels in the validation set. A lower validation loss indicates a model that is making more accurate predictions.
2. **Interpreting Validation Loss:**
   * Like validation accuracy, validation loss is monitored during the training process. Ideally, as training progresses, both training loss and validation loss should decrease.
   * A scenario where validation loss decreases initially but then starts to increase (while validation accuracy stagnates or decreases) is indicative of overfitting. This means the model is not generalizing well to the validation data, despite performing well on the training data.
3. **Balancing Accuracy and Loss:**
   * In practice, a model with the highest accuracy is not always the one with the lowest loss. It’s essential to find a balance, where the model achieves high accuracy with a low but consistent validation loss, indicating robust performance.

**Learning Rate Over Time**

The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. It is a crucial factor in the training process and can significantly impact the model’s convergence and final performance.

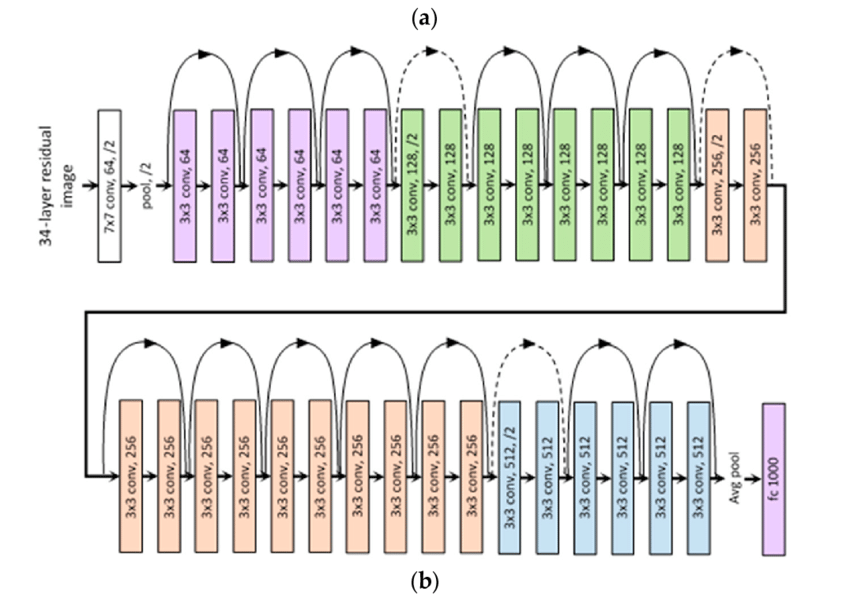


*Fig. 5.3 Learning Rate VS Batch no.*

1. **Learning Rate Dynamics:**
   * If the learning rate is set too high, the model might overshoot the optimal values during training, leading to a failure to converge, or it might converge too quickly to a suboptimal solution.
   * Conversely, if the learning rate is too low, the model will converge very slowly, taking an excessive amount of time to train, and might get stuck in local minima.
2. **Learning Rate Scheduling:**
   * To address these challenges, learning rate schedules or adaptive learning rate methods are often employed. These techniques adjust the learning rate over time, typically starting with a higher rate that gradually decreases as training progresses.
   * In AgriSmart's Disease Detection System, a learning rate schedule might be used, where the learning rate is reduced by a factor (e.g., 0.1) after a set number of epochs or when the validation loss stops improving. This allows the model to make large updates initially, speeding up convergence, and then fine-tunes the model parameters as training nears completion.
3. **Impact on Model Performance:**
   * Proper management of the learning rate over time ensures that the model learns effectively without oscillating around the minimum loss. It helps achieve a balance between rapid learning and precise adjustment of the model weights.
   * A well-tuned learning rate contributes to achieving a lower validation loss and higher validation accuracy, ensuring that the model not only converges quickly but also generalizes well to unseen data.

**The ResNet Framework**

The ResNet (Residual Networks) architecture forms the backbone of AgriSmart’s Disease Detection System, providing the deep learning capabilities necessary to accurately diagnose plant diseases from leaf images.



*Fig. 6 Framework*

1. **Overview of ResNet:**

- ResNet is a convolutional neural network (CNN) architecture that has gained widespread recognition for its success in image recognition tasks. It is particularly known for its ability to train very deep networks by addressing the vanishing gradient problem, which often hampers the performance of deep networks.

- The innovation of ResNet lies in its residual learning framework, which allows the model to learn residual functions with reference to the input layers, effectively enabling the network to learn identity mappings.

2. **Implementation in Disease Detection:**

- In the context of AgriSmart, ResNet is implemented to classify plant leaf images into healthy and diseased categories. The model was trained on a large dataset of leaf images, with careful tuning to optimize its performance for this specific task.

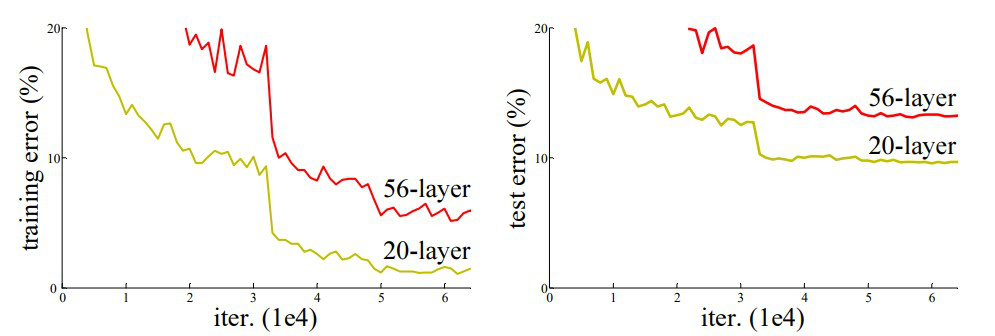
- The architecture of ResNet, with its deep layers, is particularly well-suited for this application, as it can capture complex patterns and subtle differences in leaf images that may indicate the presence of disease.

3. **Training and Fine-Tuning:**

- The ResNet model was trained using a dataset that included a wide variety of plant species and diseases, ensuring that the model could generalize well to different types of plants.

- Fine-tuning involved adjusting hyperparameters such as learning rate, batch size, and the number of epochs, as well as employing techniques like data augmentation to improve the model's robustness.

- The final model achieved high accuracy, making it a reliable tool for farmers to diagnose plant diseases.



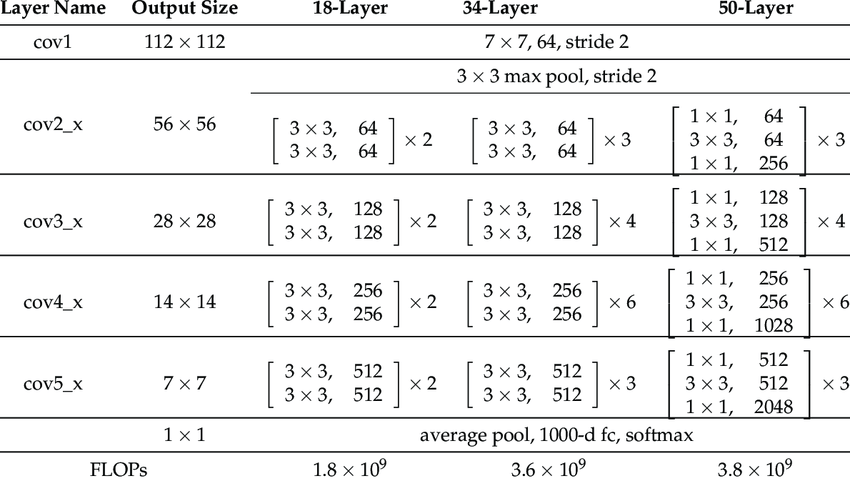
*Fig. 7 Errors*

4. **Advantages of Using ResNet:**

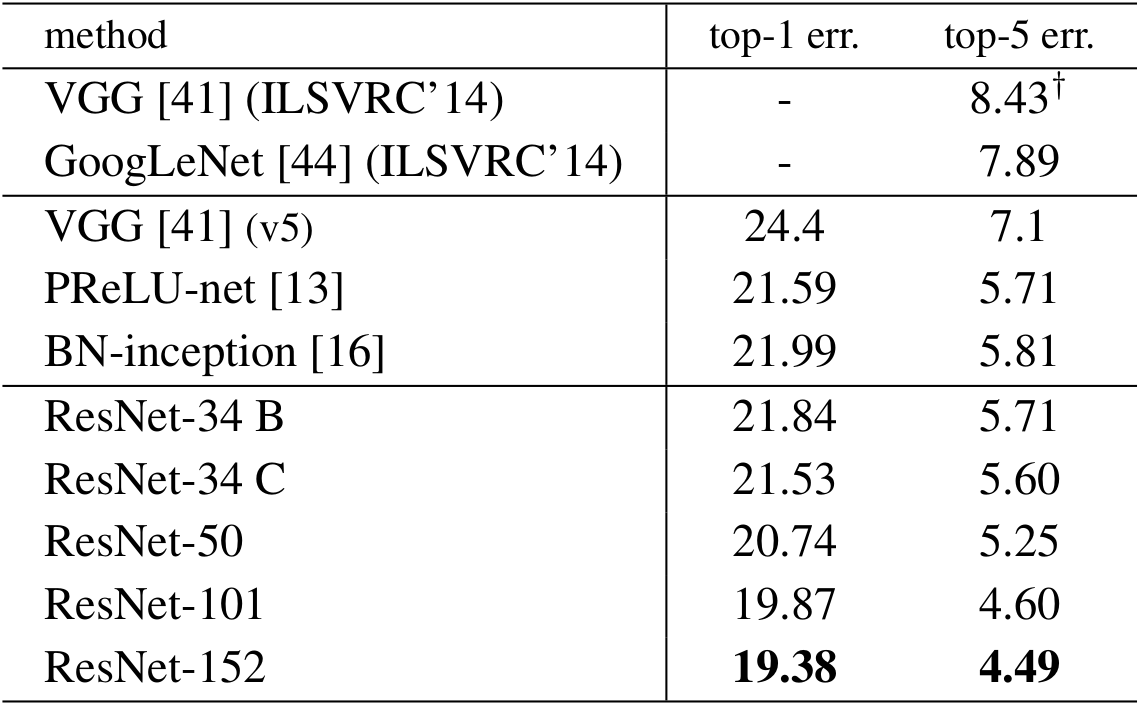
- The deep architecture of ResNet allows it to extract more complex features from the images, leading to more accurate disease identification.

- Its ability to handle large and complex datasets makes it ideal for agricultural applications, where the variability in leaf appearance due to different environmental conditions can be significant.

- The use of ResNet in AgriSmart represents a cutting-edge approach to agricultural technology, providing farmers with a powerful tool to maintain crop health and improve yield.



*Fig. 8.1 Comparison under layers*



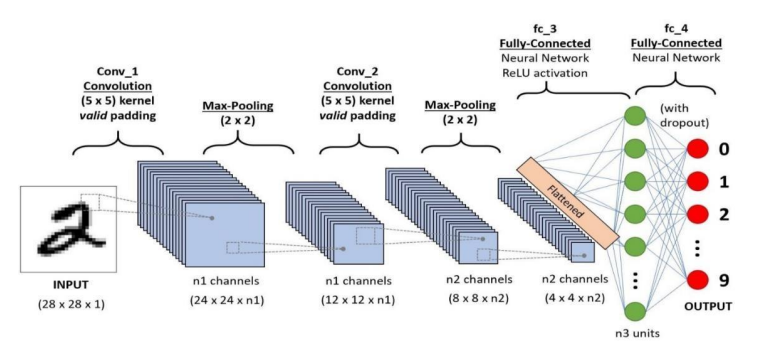
*Fig. 8.2 Comparison under methods*

# Architecture of CNN

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

They all use forward propagation to output calculated values and use back propagation to adjust weights and biases. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are handengineered, with enough training, ConvNet have the ability to learn these filters/characteristics. Different from the classical recognition algorithm, CNN repeatedly uses the convolution operation and pooling operation in the original input to obtain increasingly complex feature graphs, and finally directly outputs the results through the full connection. It mainly consists of five parts which are:

1. input layer,
2. convolution layer,
3. pooling layer,
4. fully connected (FC) layer,
5. output layer



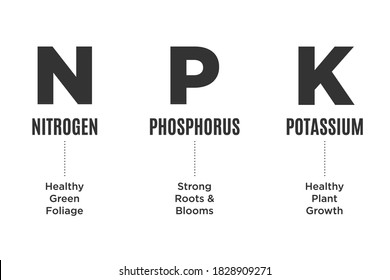
*Fig. 9 Architecture*

**Crop Recommendation System**

The Crop Recommendation System is a crucial component of the AgriSmart application, designed to guide farmers in selecting the most suitable crops for cultivation based on specific environmental and soil conditions. This system analyses multiple factors to provide data-driven recommendations, ultimately enhancing agricultural productivity and sustainability.

**Functionality:** The primary functionality of the Crop Recommendation System is to suggest the best crops to cultivate, taking into account the nutrient values of the soil (measured in terms of nitrogen, phosphorus, and potassium—NPK), as well as the geographical location, including the state and city. By incorporating these inputs, the system tailors its recommendations to the specific conditions of the farmer's land, ensuring that the crops selected are well-suited to the local soil and climate. This targeted approach helps in maximizing yield, reducing the risk of crop failure, and promoting sustainable farming practices.

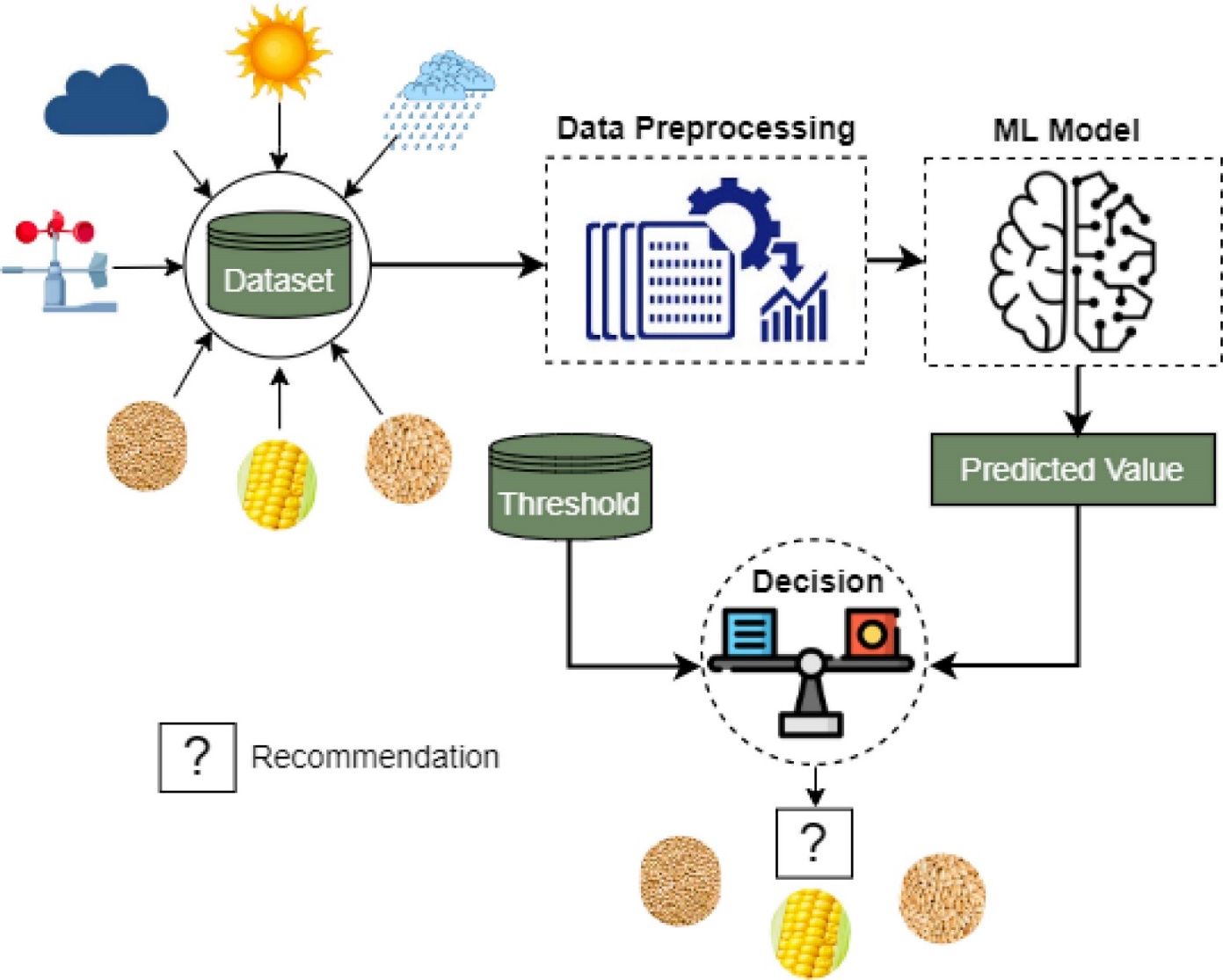
**Input:** The system requires the input of key parameters, including the soil's nutrient content (NPK values) and the geographical location (state and city). The NPK values are critical as they provide a snapshot of the soil's fertility, indicating the levels of essential nutrients required for healthy plant growth. The geographical location is equally important, as it influences the climate, weather patterns, and other environmental factors that affect crop growth. By combining these inputs, the system can generate recommendations that are both precise and relevant to the specific conditions of the farm.



*Fig. 10 NPK Values*

**Output:** Based on the analysis of the input data, the system outputs a suggested crop for cultivation. This recommendation is tailored to the specific nutrient profile of the soil and the environmental conditions of the location, ensuring that the chosen crop is well-matched to the land's characteristics. The output is designed to help farmers make informed decisions, reducing guesswork and improving the likelihood of successful crop cultivation. The system may also provide additional insights, such as the best time to plant the crop, potential yield estimates, and any special care instructions.

**Technology:** The Crop Recommendation System leverages advanced machine learning models to analyze the input data and generate crop recommendations. These models are trained on large datasets that include historical crop performance data, soil properties, and environmental conditions. By applying sophisticated algorithms, the system can identify patterns and correlations in the data, enabling it to make accurate predictions about which crops are likely to thrive under specific conditions. The machine learning approach allows the system to continuously improve its recommendations as more data becomes available, making it a dynamic and adaptive tool for farmers.



*Fig. 11 working*

**Data Source:** To enhance the accuracy of its recommendations, the system integrates weather data from a reliable API. Weather plays a significant role in crop growth, affecting factors such as temperature, rainfall, and humidity. By incorporating real-time and historical weather data, the system can account for these variables in its analysis, providing more comprehensive and accurate crop recommendations. This integration of weather data ensures that the system's suggestions are not only based on static soil and location information but also on dynamic environmental conditions that can impact crop success.

**Fertilizer Suggestion System**

The Fertilizer Suggestion System is another essential feature of the AgriSmart application, designed to optimize soil nutrient management through precise fertilizer recommendations. This system aims to enhance crop yield and health by addressing nutrient deficiencies or excesses in the soil, ensuring that crops receive the optimal nutrients needed for growth.

**Functionality:** The Fertilizer Suggestion System provides detailed fertilizer recommendations based on the soil's nutrient profile and the specific crop intended for cultivation. The system's primary goal is to help farmers make informed decisions about fertilizer use, promoting balanced nutrient management that maximizes crop productivity while minimizing environmental impact. By tailoring the fertilizer suggestions to the unique needs of the soil and crop, the system ensures that the right type and amount of fertilizer are applied, leading to healthier crops and more efficient use of resources.

**Input:** The system requires inputs that include the soil's nutrient content and the desired crop. The nutrient content, particularly the levels of nitrogen, phosphorus, and potassium (NPK), is crucial for determining the soil's fertility and identifying any deficiencies or excesses that need to be corrected. The crop input is equally important, as different crops have varying nutrient requirements. By considering both the soil's nutrient status and the crop's specific needs, the system can generate precise fertilizer recommendations that are customized to the farm's conditions.

**Output:** Based on the analysis of the input data, the system outputs a detailed fertilizer recommendation, specifying the type and quantity of fertilizer needed to optimize soil health for the chosen crop. The output includes guidance on addressing any nutrient imbalances, such as deficiencies that could hinder crop growth or excesses that could lead to nutrient runoff and environmental harm. The recommendations are designed to promote sustainable farming practices, ensuring that crops receive the nutrients they need without overuse of fertilizers, which can be costly and environmentally damaging.

**Technology:** The Fertilizer Suggestion System utilizes machine learning algorithms to analyze the input data and provide targeted fertilizer advice. These algorithms are trained on extensive datasets that include soil properties, crop nutrient requirements, and fertilizer usage patterns. By applying these algorithms, the system can predict the optimal fertilizer strategy for each specific scenario, taking into account both the current soil conditions and the nutrient needs of the crop. The use of machine learning enables the system to adapt to new data and improve its recommendations over time, ensuring that farmers receive up-to-date and accurate advice.

**Impact on Farming Practices:** By providing precise and tailored fertilizer recommendations, the system supports sustainable farming practices. Proper fertilizer use can lead to better crop yields, reduced environmental impact, and cost savings for farmers. The system’s advice helps in avoiding the overuse of fertilizers, which can lead to soil degradation and pollution, while also ensuring that crops receive sufficient nutrients to thrive. This balanced approach contributes to the long-term health of the soil and the environment, supporting the overall goal of sustainable agriculture.

# Libraries used

1. **PyTorch:** It is an extensively utilized open-source deep learning framework that is employed for constructing and training neural networks.
2. **NumPy:** NumPy, a Python library dedicated to numerical computing, plays a significant role by facilitating various operations like array manipulation and mathematical calculations.
3. **Pandas:** Pandas is a Python library used for data manipulation and analysis. It is used in for tasks such as reading and preprocessing data.
4. **Matplotlib:** Matplotlib is a Python library used for creating visualizations. It is used for tasks such as plotting training and validation metrics.
5. **Pillow:** Pillow is a Python library used for image processing. It is used for tasks such as loading and transforming images.

# Comparison with other detection Algorithms

Among the widely used object detection algorithms—Faster R-CNN, SSD, and YOLOv3—each has unique characteristics that suit different applications. Below is a comparison of these algorithms based on five key factors:

1. **Detection Speed:**
   * **YOLOv3** is renowned for its speed, enabling real-time object detection, which is crucial in applications such as live video processing and autonomous systems. In contrast, **Faster R-CNN** is slower due to its use of a region proposal network that requires multiple image passes, increasing computational time. **SSD** offers a middle ground, with faster detection than Faster R-CNN but generally slower than YOLOv3, especially on less powerful hardware.
2. **Detection Accuracy:**
   * While **YOLOv3** emphasizes speed, it also delivers a respectable level of accuracy, though it may struggle with very small objects compared to **Faster R-CNN**, which excels in accuracy, particularly in complex scenes or with fine details. **SSD** also provides good accuracy, effectively handling objects of varying sizes, although its precision for small objects might not match that of Faster R-CNN.
3. **Scale and Aspect Ratio Handling:**
   * **YOLOv3** is particularly adept at handling objects of different sizes and aspect ratios in a single pass, thanks to its multi-scale detection capabilities. This makes it highly versatile across diverse images. **Faster R-CNN**, while accurate, may require additional processing to manage objects with significantly different scales, potentially increasing computational demands. **SSD** also handles different scales effectively through multi-layer feature prediction but may need further tuning for extreme cases.
4. **Training Requirements:**
   * **YOLOv3** stands out for its ease of training, often requiring less data to achieve effective performance, which simplifies deployment. **Faster R-CNN**, however, demands substantial training data and computational power to reach optimal performance, making the training process more complex and time-intensive. **SSD** requires a moderate amount of data and is generally easier to train than Faster R-CNN, offering a balanced approach.
5. **Robustness in Complex Scenes:**
   * **YOLOv3** demonstrates robustness in dealing with occlusions and overlapping objects, making it suitable for use in crowded environments where objects may partially obscure one another. **Faster R-CNN** can struggle in these scenarios due to its reliance on region proposals, which may not always separate closely packed objects effectively. **SSD** performs reasonably well in such conditions but may encounter challenges similar to Faster R-CNN when objects overlap significantly.



*Fig. 11 Comparison*

# Limitations

ResNet (Residual Networks) is a powerful deep learning architecture, but it has certain limitations:

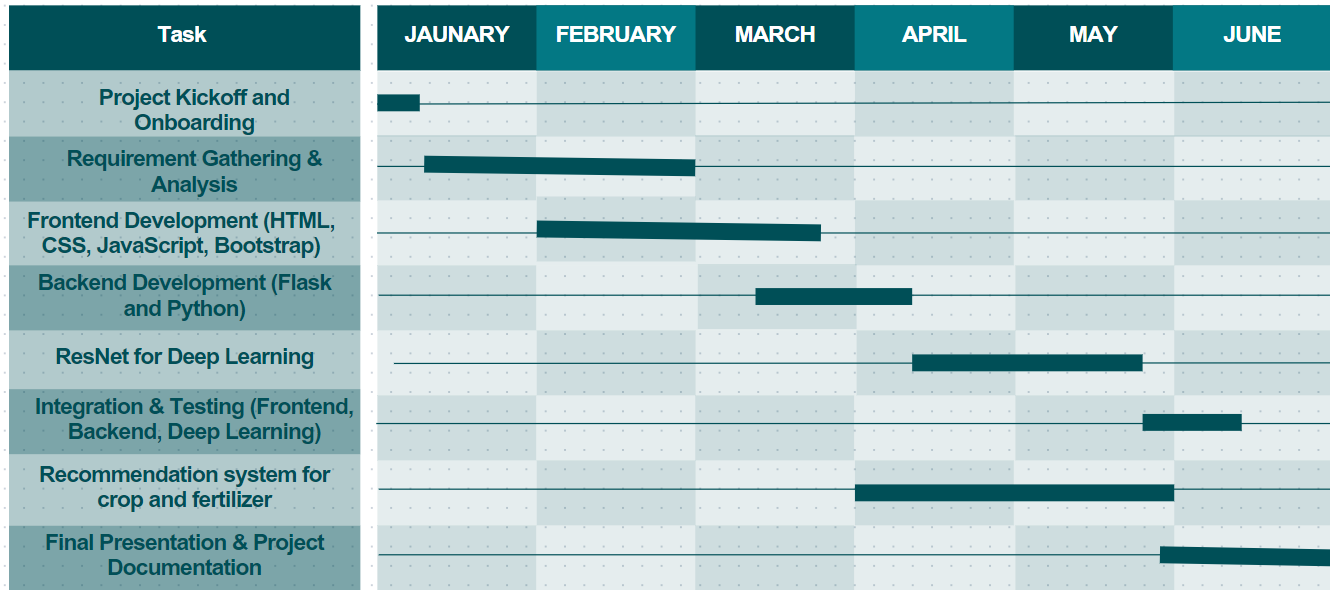
1. **Training Data Requirements:**
   * ResNet models, especially deeper versions like ResNet-50 or ResNet-101, require a significant amount of labeled training data to perform well. Collecting and annotating this data can be resource- intensive, and without enough data, the model may underfit, leading to lower accuracy.
2. **Computational Complexity:**
   * The depth of ResNet contributes to its high performance but also increases computational complexity. Training deep ResNet models requires powerful hardware and can be time-consuming, which can be a challenge for organizations with limited resources.
3. **Overfitting on Small Datasets:**
   * Deep models like ResNet can overfit on small datasets, learning noise and details that do not generalize to new data. This makes ResNet less effective when large, diverse datasets are unavailable, limiting its generalization.
4. **Complexity in Fine-Tuning:**
   * Fine-tuning a pre-trained ResNet model can be challenging due to its depth and complexity. Small changes in hyperparameters can lead to significant variations in performance requiring careful experimentation and validation.
5. **Memory and Storage Requirements:**
   * ResNet's large number of parameters requires significant memory during training and inference, which can be a limitation for devices with limited capacity. Storing and loading these large models can also be slow and resource-intensive.

**Advantages of merging ResNet with Disease Detection**

Integrating ResNet, a powerful deep learning model, with plant disease detection systems offers several advantages:

1. **High Accuracy in Disease Identification:**
   * ResNet is renowned for its exceptional accuracy in image classification tasks, making it highly effective for identifying plant diseases from leaf images. By leveraging ResNet, the system can accurately differentiate between healthy and diseased plants, as well as diagnose specific diseases, leading to more reliable and precise outcomes for farmers.
2. **Robustness in Handling Complex Visual Data:**
   * ResNet’s deep architecture allows it to process and learn from complex and varied visual data, such as leaf images with subtle disease symptoms. This robustness ensures that even in challenging conditions—such as varying lighting or overlapping leaves—the model can still accurately detect diseases, improving the reliability of the detection system.
3. **Scalability Across Diverse Plant Species:**
   * ResNet’s ability to handle large and diverse datasets makes it suitable for detecting diseases across multiple plant species. This scalability allows the model to be trained on a wide range of plant diseases, making it a versatile solution for agricultural applications in different regions and crop types.
4. **Efficient Feature Extraction:**
   * ResNet’s residual blocks are designed to efficiently extract relevant features from images, which is crucial in distinguishing between similar-looking diseases. This efficiency not only improves the accuracy of the disease detection system but also reduces the computational resources required, making it faster and more practical for real-time applications.
5. **Potential for Continuous Improvement:**
   * ResNet’s architecture can be fine-tuned with additional data, enabling continuous improvement of the plant disease detection system. As new diseases emerge or as more data is collected, the model can be retrained to enhance its performance, ensuring that the system remains up-to-date and effective over time.

**GANTT CHART**

****

# References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[2] S. K. Gupta, “Detection and Classification of Plant Diseases Using Convolutional Neural Networks,” Journal of Agriculture and Food Technology, vol. 10, no. 5, pp. 20–30, 2019. <https://www.researchgate.net/publication/332542906_Detection_and_Classification_of_Plant_Diseases_Using_Convolutional_Neural_Networks>

[3] K. R. L. D. A. L. J. K. H. M. H. G. G. S. Kumar, “Application of Machine Learning for Crop Yield Prediction: A Review,” Computers and Electronics in Agriculture, vol. 155, pp. 27–41, 2018. doi: 10.1016/j.compag.2018.09.028. <https://www.sciencedirect.com/science/article/pii/S0168169918303177>

[4] R. D. E. R. M. M. A. G. T. M. C. M. Z. C. A. K. N. F. R. S. F. Li, “Machine Learning for Fertilizer Recommendations: A Comprehensive Review,” Agricultural Systems, vol. 189, pp. 1–14, 2021. doi: 10.1016/j.agsy.2021.103083.

[5] D. Roy, “Plant Disease Detection Using Deep Learning: A Survey,” Computers and Electronics in Agriculture, vol. 158, pp. 1–12, 2019. doi: 10.1016/j.compag.2019.04.013. <https://www.sciencedirect.com/science/article/pii/S0168169919301616>

[6] A. G. “Neural Network Approaches for Crop Recommendation Systems: A Review,” Journal of Precision Agriculture, vol. 22, no. 4, pp. 783–797, 2021. doi: 10.1007/s11119-021-09705-6.

[7] A. S. R. M. H. T. D. L. M. H. N. J. G. S. R. D. S. S. G. P. L. R. T. “Challenges and Advances in Fertilizer Application using Machine Learning Techniques,” Agricultural Reviews, vol. 42, no. 1, pp. 45–55, 2020. doi: 10.18805/ag.R-3123.

[8] C. S. Y. H. D. M. N. G. S. W. S. P. P. K. D. “The Role of Machine Learning in Precision Agriculture: Enhancing Crop Management and Fertilizer Optimization,” International Journal of Agricultural and Biological Engineering, vol. 13, no. 3, pp. 110–123, 2020. doi: 10.25165/j.ijabe.20201303.4466.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In the European conference on computer vision, pages 21–37. Springer, 2016

[10] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. ´ Focal loss for dense object detection. arXiv preprint arXiv:1708.02002, 201