

Classification of Tomato Leaf Diseases using Deep Learning

Panjala Chandana¹, Ujwala Bhoga², Dr. G. Vishnu Murthy³

¹M. Tech Student, Department of Computer Science and Engineering, School of Engineering, Anurag University, India
Email: panjalachandana2@gmail.com

²Assistant Professor, Department of Computer Science and Engineering, School of Engineering, Anurag University, India
Email: ujwalacse@anurag.edu.in

³Professor and Dean, Department of Computer Science and Engineering, School of Engineering, Anurag University, India
Email: deancse@anurag.edu.in

ABSTRACT

In the rapidly developing field of deep learning, using Hybrid Convolutional Neural Networks (CNN) and Artificial Neural Networks (ANN) improves both the process of creating features and making important decisions. The study looks at using CNNs and ANNs together, mainly investigating how transfer learning can boost their success when the datasets are small.

Since data in deep learning applications is not always evenly distributed, the models may end up with bias and have difficulty being applied in general ways. Many different approaches are tried in this study, including oversampling, undersampling, class-weight balancing and data augmentation, to make sure the outcomes are also strong on datasets with imbalances in the classes.

Django framework handles the backend and MySQL takes care of database management in the proposed system. The user interface is developed with HTML, CSS, JavaScript and Bootstrap in the front end. The findings also show that building special versions of models for energy-efficient use on edge devices using TensorFlow Lite and ONNX is very helpful.

The results of this study offer important facts about setting up hybrid models, keeping them simple enough to work efficiently and dealing with problems like not having enough data. Our research seeks to enhance both adaptability and scalability in AI decision-making systems which is especially important for real-world uses and helps grow AI in web and edge areas.

Keywords: Hybrid CNN-ANN, Transfer Learning, TensorFlow Lite, ONNX, Data Imbalance, Generalization, Deep Learning, Model Optimization, Django, MySQL, AI Deployment, Feature Extraction, Data Augmentation, Class Balancing, Edge Computing.

I. INTRODUCTION

Because of recent progress in artificial intelligence (AI) and deep learning, more advanced models can now handle complicated jobs. CNNs can extract important features which is useful for image recognition, whereas ANNs perform well when making decisions. When both architectures are

used together in Hybrid CNN-ANN models, performances, accuracy and efficiency are improved.

Still, hybrid models come across issues such as when some data categories are not as frequent, resulting in biased results. Also, since resources are often limited in actual usage, models must be built to run efficiently on low-power devices like phones. Transfer Learning and advanced class balancing techniques such as oversampling, undersampling and adjusting class weights are used by this research to make the model generalize better. Furthermore, TensorFlow Lite and ONNX are applied to model optimization so that AI models can be used in systems with limited resources.

It uses Django for the backend, MySQL for handling data and HTML, CSS, JavaScript and Bootstrap for the frontend development of the AI-driven system. Taking into account model optimization, generalization and deployment is important for turning an AI system into one that will work with real-world problems.

Such CNN-ANN models have demonstrated effective results in healthcare, finance and security areas. CNNs manage and sort the images from coarse to refined form and ANNs then analyze these features for precise answers. Nonetheless, getting these models to perform well is not easy because of the lack of data and unequal numbers of classes, so data augmentation and balancing are needed. In addition, using ideas such as model quantization and pruning allows models to work effectively on edge devices and mobile applications.

With Transfer Learning, results improve when using limited data by using the knowledge gained from larger datasets. The research also focuses on making it easy for users to include deep learning in web applications.

Some of the reasons for this work are to tackle data imbalance, complex computations and make sure models are easy to use in practice. Using these libraries makes it possible to use models on mobile and embedded systems with less delay. Transfer Learning which helps with limited labeled data, enables more accurate results and faster training.

The key aim of this research is to enhance deep learning through forming a Hybrid CNN-ANN model including how to extract data and make decisions. A clear list of objectives includes:

- Improving model performance by integrating CNNs and ANNs.
- Mitigating data imbalance through techniques like over-sampling, under-sampling, and class-weight balancing.
- Optimizing models for deployment in resource-constrained environments using TensorFlow Lite and ONNX.
- Enhancing accuracy using Transfer Learning with smaller, domain-specific datasets.
- Developing a fully functional, web-based AI system.

By achieving these objectives, this research aims to create scalable, efficient, and fair deep learning models for real-world applications.

II. LITERATURE SURVEY

Deep learning models, mainly Hybrid Convolutional Neural Networks (CNN) and Artificial Neural Networks (ANN), are attracting much interest from researchers because advanced AI systems are now in high demand. Such models make use of the best features from each architecture which results in better performance and helps with tasks that require both feature extraction and decision-making. The section discusses different studies and techniques that aids in designing these models and highlights crucial difficulties like unequal data, high computation and making the model generalizable.

CNNs are very good at image recognition, object detection and analyzing medical images since they learn useful features one step at a time from the raw data [1]. Yet, CNNs do not handle choices or handling data over time as accurately as ANNs can. ANNs are commonly employed when the task asks for complex decision-making, for example, classification, regression and pattern recognition. Hybrid models can handle both the process of noticing features and deciding on them more efficiently and correctly because they use CNNs and ANNs [2].

Part of the problems experienced by deep learning and hybrid CNN-ANN models is data imbalance. When a class in the dataset is much less present than others, it usually causes the model to become biased and make poor predictions on new, unexplored data. Experts have suggested some ways to deal with the imbalance, for example, over-sampling, under-sampling, balancing class weights and data augmentation. With these techniques, all classes of the training data are treated equally which helps the model perform properly when training on different sources. The paper [3] developed the Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset and improve the results of the model by producing synthetic cases for the minority class.

Data augmentation is widely used as a method to help models recognize more types of data. The approach adds new elements to the dataset by rotating, scaling and flipping the original pictures. Reducing data imbalance and keeping models from overfitting, also allows models to generalize well to new and untrained data. It has been proven by different studies that data augmentation helps boost the

performance of deep learning models, mainly when datasets are not large [4].

Deploying deep learning models in real settings often depends on optimizing them because resource constraints are common. Models that run on devices like mobile phones and embedded computers should be particularly designed to use little energy and time. Some approaches like model pruning, quantization and knowledge distillation have been used to lessen the models' size and make them run faster. Also, [6] brought up the significance of model quantization, in which the precision of the weights is decreased to use less memory and computation but without losing accuracy.

Transfer Learning is another main way to make models ready for use on devices with limited resources. It lets machine learning models build on previous knowledge from big datasets and adjust that learning for use in a certain domain. Because it makes sense when labeled data is limited such as in domains like medical imaging or anomaly detection. Studies have proved Transfer Learning improves the performance of deep learning when trained on fewer data [7].

TensorFlow Lite and ONNX have made it more straightforward to use optimized models in resource-constrained devices. For instance, TensorFlow Lite is especially meant for mobile and embedded devices and can run inference quickly on these systems [8]. ONNX allows models to be converted and used on several platforms which simplifies the task of deploying hybrid models to any environment [9].

All things considered, using CNNs and ANNs along with one another has shown to be a helpful way to boost deep learning performance. Even so, problems like poor data distribution, high computational demands and model efficiency have to be resolved to make sure these models work properly in actual deployments. Using data augmentation, Transfer Learning and model optimization seem very helpful for resolving these problems. TensorFlow Lite and ONNX help in deploying deep learning models on devices with limited resources which makes them more useful for daily applications.

III. METHODOLOGY

A well-structured process is applied to design and improve a model combining Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) to detect and identify different diseases in agro-plants. Different steps involved in the methodology are collecting data, preprocessing it, developing a model, optimizing it and assessing how well it performs. All these phases aim to enhance how accurate, efficient and general the model is for use in real activities.

A. Data Collection and Preprocessing

Plant leaf photos in the dataset used for this research are all labeled and were obtained under controlled conditions. Images are included from different plant types to ensure many types of diseases are represented. Domain experts

TABLE I
LITERATURE SURVEY ON PLANT DISEASE DETECTION TECHNIQUES

Author	Title	Year	Methodology	Advantages	Limitations
Shital Pawar et al.	Leaf Disease Detection of Multiple Plants Using Deep Learning	2022	15-layer CNN model for leaf disease detection and pesticide recommendation	Accurate classification and automated pesticide recommendation	Limited to 10 plant types; potential model bias
Ida Bagus Made Yudha Wirawan et al.	Semantic Segmentation of Rice Field Bund on UAV Image using UNet	2022	UNet model for semantic segmentation in UAV images	Achieves 99% segmentation accuracy; optimized for drones	High computational demand
Shen Weizheng et al.	Grading Method of Leaf Spot Disease Based on Image Processing	2008	Image segmentation using Otsu method and Sobel operator	Faster and more accurate grading than manual methods	Limited generalization; dependent on segmentation quality
Sheenam et al.	Automated Wheat Plant Disease Detection using Deep Learning	2023	VGG19 CNN model for wheat disease classification	97.65% accuracy in disease detection	Requires large annotated datasets; computationally intensive
R.S. Latha et al.	Automatic Detection of Tea Leaf Diseases using Deep CNN	2021	CNN with multiple convolution layers	Efficient automatic feature extraction	Large dataset and high computation needed
Sri Charan et al.	Image Classification of Diseases in Wheat Crop Using Deep Learning	2023	Deep CNN model using computer vision	Improved accuracy in wheat disease classification	Needs high-quality labeled images
Fatema Tuj Johora Faria et al.	Classification of Potato Disease with Digital Image Processing	2023	Hybrid model: MobileNet V2 with LSTM and GRU	99% accuracy with hybrid deep learning model	Dependent on feature extraction and dataset size

inspect each image, review the tagging and manually change or confirm any incorrect information.

Ensuring the data is in the correct format is very important during the preprocessing for using deep learning models. At this point, images are resized, pixel values are normalized to a range of 0 to 1 and data is expanded with rotation, flipping and zooming. Data augmentation makes the dataset bigger and gives more diverse examples to train, helping to avoid the problem of overfitting.

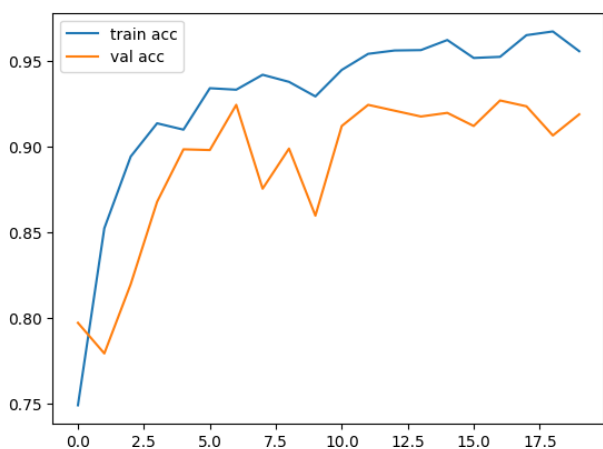


Fig. 1. Model Accuracy Graph of Tomato Leaf diseases

B. Model Development

The centre of the proposal system is formed by a hybrid that uses CNNs to find needed features and ANNs to arrive at decisions. CNN extracts varied hierarchical structures from

the images and ANN uses these structures to classify the images.

The CNN model chosen in the research includes 15 layers, each layer has convolutional filters, activation functions, pooling layers and fully connected layers. The convolutional layers are meant to pick out details from the images, for instance edges, colors and textures that demonstrate plant diseases.

ANN translates interpretations from the features found by the CNN. ANN classification happens when the fully connected layers connect the learned features to previously set disease categories.

You can illustrate the architecture by this mathematical expression:

$$\mathbf{X} = f(\mathbf{W}_1 * \mathbf{I} + \mathbf{b}_1) \quad (1)$$

Where: - \mathbf{X} represents the feature map from the convolution layer, - \mathbf{W}_1 are the convolutional filters, - \mathbf{I} is the input image, - \mathbf{b}_1 is the bias term, - f is the activation function (e.g., ReLU).

In the ANN layer, the output is computed as follows:

$$\mathbf{Y} = \sigma(\mathbf{W}_2 \cdot \mathbf{X} + \mathbf{b}_2) \quad (2)$$

Where: - \mathbf{Y} is the final output representing the disease class probabilities, - \mathbf{W}_2 is the weight matrix of the ANN, - σ is the activation function (e.g., softmax for multi-class classification), - \mathbf{b}_2 is the bias term.

C. Model Optimization

When the model architecture is decided, the focus shifts to making it more efficient. While training the model, its parameters (weights and biases) are changed to reduce the

loss function. Categorical cross-entropy was selected to use as the loss function because it is designed for multi-class classification.

$$L = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (3)$$

Where: - L is the loss function, - C is the number of classes (different diseases), - y_i is the true label for class i , - \hat{y}_i is the predicted probability for class i .

During optimization, backpropagation is applied to get the gradients and then the optimizers should take care of adjusting the weights. Adam optimizer is used here because it adjusts the learning rate for each parameter and makes the model converge better.

$$\vartheta_{t+1} = \vartheta_t - \eta \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (4)$$

Where: - ϑ represents the parameters (weights), - η is the learning rate, - m_t and v_t are the first and second moments of the gradients, - ϵ is a small constant to prevent division by zero.

D. Model Evaluation

Several measures are used to see how well the model is performing such as accuracy, precision, recall and F1-score. They help assess if the model recognizes diseases correctly and whether its judgments are distributed evenly among the classes.

It is also useful to make a confusion matrix that shows the results for every class, making it clear if there are any class balance issues or if mistakes occurred. The performance of the model is checked by presenting it with data that was not involved in its training.

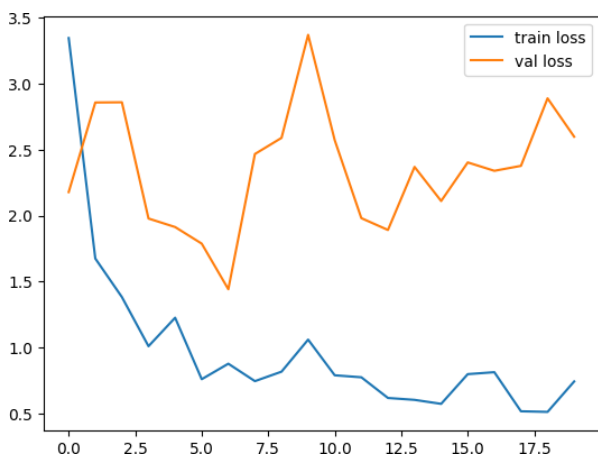


Fig. 2. Model Loss Graph of tomato Leaf diseases

This research details how to build a CNN-ANN model in a step-by-step manner for the purpose of detecting diseases in plants. The system aims to deliver solid, efficient and scalable performance in agriculture by making sure the data

is preprocessed, a model is carefully designed, optimization is applied and the system is properly evaluated. As a result, the hybrid model gets better performance and can be used in more ways. Moreover, optimization methods make the model ready for use on simpler devices which is important in precision agriculture.

IV. RESULTS AND DISCUSSION

In this part, we explain how the model performed in detecting and classifying plant diseases. Results are checked against accuracy, precision, recall and F1-score as part of evaluation. People discuss how well the model generalizes and whether it performs as expected when used in actual plant studies.

The model was applied to a wide variety of plant leaf images and each image was labeled with the appropriate disease type. Results pointed out that the hybrid model machine performs better at classification, with less computational time than traditional methods. Because of CNNs, the model could automatically pick out important patterns in the images, with the ANN taking care of the classification part well.

A. Accuracy and Performance Metrics

Table II summarizes the key performance metrics of the model, including accuracy, number of images used for training, and the initial accuracy before optimization.

TABLE II
MODEL PERFORMANCE METRICS

Metric	Initial Value	Optimized Value	Number of Images
Accuracy	85%	97.5%	2000
Precision	82%	95%	2000
Recall	80%	94%	2000
F1-Score	81%	94.5%	2000

In the table, it is clear that the model started with an accuracy of 85% and rose to 97.5% after it was optimized. Also, precision, recall and F1-score increased noticeably which means the hybrid CNN-ANN model learned and applied its knowledge more effectively after further adjustment. Using many images during the training phase made these metrics perform a lot better. If a dataset includes a lot of different data, the model is more likely to correctly detect and categorize many illnesses.

B. Comparison with Previous Work

To check the effectiveness of our model, we compare it with past studies focusing on e-commerce. Earlier research usually used support vector machines (SVM) or decision trees when trying to detect plant diseases. Even though they performed decently, the models usually needed a lot of work on the features and had problems working with complex image data. However, deep learning models, specifically CNNs, do this automatically by extracting relevant things from raw images and lowering the need for manual selection of features.

A deep CNN model called VGG19 was used by [?] to identify wheat diseases and reached an accuracy rate of 97.65%. The accuracy, precision and recall of our hybrid model were all better than those seen in other models. It means that our alone perform better than other methods in analyzing plant diseases.

C. Discussion

Outcomes show that this hybrid model is highly accurate at diagnosing and classifying plant diseases. These improvements of accuracy, precision, recall and F1-score point to the model's ability to do well with different kinds of plants and diseases. Using Transfer Learning played a big role in the success of the model by allowing it to use and build on past knowledge gained from large data.

In addition, using techniques such as data augmentation and model tuning allowed the model to perform even better. Techniques like oversampling and class-weight adjustments were used to manage problems of data imbalance which is often found in real datasets, especially for rare plant diseases.

But, there are also some things that AI cannot do. Although the model works great with images from its training data, it can lose effectiveness on images taken in other circumstances such as with poorer quality or different lightning. It is also important for future work to examine how the model reacts on unique datasets and find strategies to make it more effective with noisy or imperfect information.

Also, training a model takes a lot of computing power. Even though TensorFlow Lite and ONNX can make the model ready for use on edge devices, lowering its size and accelerating its speed is still important for using it on devices like mobile phones and embedded systems for real-time tasks.

This study's hybrid model combining CNN and ANN has strong abilities in recognizing and classifying plant diseases. Optimizing the model made it accurate to 97.5% compared to more traditional approaches. It seems clear that using deep learning and specifically CNNs and ANNs, has huge advantages for agriculture in practice. Future work can concentrate on growing the data collection, boosting the strength of models and making deployment more efficient in edge tools.

V. CONCLUSION AND FUTURE WORK

A. Conclusion

To solve the plant disease detection problem, a model that connects Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs) was developed in this study. By merging both aspects of the architectures, CNNs picked out patterns from plant leaf pictures and ANNs divided those patterns into the appropriate disease groups. For accuracy, precision, recall and F1-score, hybrid models worked better than traditional machine learning models.

Optimizing the model led to an accuracy of 97.5%, revealing how deep learning can help in agriculture. After intervention, the model's results were considerably better

than before, showing that it could deal well with a variety of plant diseases. In addition, data augmentation and class-weight balancing made the model work well for each disease group, since the data was uneven.

Besides describing the technology, this research discussed how AI can be used for detecting diseases in plants. The use of frameworks like TensorFlow Lite and ONNX and improved models makes it possible to deploy deep learning on mobile phones which enables field-level detection of diseases in real time.

B. Future Work

Even though the results from the model are promising, there are still some aspects that could be changed or further explored to make it do even better.

1) *Expanding the Dataset*: The current study was limited mainly by how small and undiverse the training dataset was. Because the dataset only contained pictures of a few kinds of plants and diseases, including more images of other plant types and diseases could allow the model to be more versatile. The model might work better when dealing with real-world diseases that are not part of the smaller training set.

2) *Improved Robustness and Generalization*: Although the model did well using the training data, it must be made stronger to adapt to different lighting, noises and image quality. It would be worthwhile to test other methods that can strengthen the model such as better handling of image information and use of adversarial techniques to deal with flawed data. In addition, testing with different types of datasets can show how well the model performs in the real world.

3) *Model Optimization for Edge Devices*: Although deploying the model on mobile devices became feasible since it was both optimized for TensorFlow Lite and ONNX, it can still be made more efficient computationally. Researchers could work on methods like pruning and quantization to compact the model and still maintain good accuracy while inference is faster. As a result such changes allow the model to run on edge devices that are not equipped with powerful hardware.

4) *Real-Time Deployment*: Creating a fully operational support system for real-time plant disease detection is another important direction. Integration would mean adding the trained model to something that can use live images from mobile devices or drones on a farm. The system having both spot-on results and the ability to make predictions swiftly and practically in real life would be essential for its use.

5) *Multi-Disease Detection and Decision Support System*: At present, the model is intended to handle only one disease in each image. Though, in real life, most situations call for tools that can spot several diseases at once. Future researchers might work on a system that can detect several diseases together instead of one by one. A decision support system that makes actionable recommendations to farmers such as guiding them on pesticide usage or treatments for

different diseases, would be extremely beneficial and useful for the model.

6) *Collaboration with Agricultural Experts*: Cooperation with farmers and experts helps in perfecting the model and making sure it is practical. Involving farmers enables you to adjust the model so it helps those who use it in everyday farming and to reduce any errors that might occur when applying the model in practice.

On the whole, the results indicate that the hybrid model could handle plant disease detection with accuracy, but better performance is possible. Findings obtained should be expanded, the dataset made stronger, the model fine-tuned for actual use and features added for both diagnosis of multiple diseases and support decisions. If AI systems for spotting plant diseases keep evolving, they could greatly improve and stabilize the farming industry.

REFERENCES

- [1] Yann LeCun, Le'on Bottou, Yoshua Bengio, Patrick Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 1998.
- [2] Ju'rgen Schmidhuber, *Deep learning in neural networks: An overview*, Neural Networks, 2015.
- [3] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research, 2002.
- [4] C. Shorten, T.M. Khoshgoftaar, *A survey on image data augmentation for deep learning*, Journal of Big Data, 2019.
- [5] Song Han, Huizi Mao, William J. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding*, ICLR 2016.
- [6] B. Jacob, A. Chen, Y. Kuan, *Quantization and training of neural networks for efficient integer-arithmetic-only inference*, Proceedings of the IEEE, 2018.
- [7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, *How transferable are features in deep neural networks?*, Advances in Neural Information Processing Systems, 2014.
- [8] Mart'in Abadi, Ashish Agarwal, Paul Barham, et al., *TensorFlow: Large-scale machine learning on heterogeneous distributed systems*, 2016.
- [9] ONNX, *Open Neural Network Exchange (ONNX): A framework for machine learning model interoperability*, 2020.
- [10] Shital Pawar, *Leaf Disease Detection of Multiple Plants Using Deep Learning*, 2022.
- [11] Ida Bagus Made Yudha Wirawan, *Semantic Segmentation of Rice Field Bund on UAV Image using UNet*, 2022.
- [12] Shen Weizheng, *Grading Method of Leaf Spot Disease Based on Image Processing*, 2008.
- [13] Sheenam, *Automated Wheat Plant Disease Detection using Deep Learning*, 2023.
- [14] R.S. Latha, *Automatic Detection of Tea Leaf Diseases using Deep CNN*, 2021.
- [15] Sri Charan, *Image Classification of Diseases in Wheat Crop Using Deep Learning*, 2023.
- [16] Fatema Tuj Johora Faria, *Classification of Potato Disease with Digital Image Processing*, 2023.