Xception for Tomato Leaf Disease Detection: Hyperparameter Tuning and Fine-tuning Approaches

Yash Maini
University School of Automation and Robotics
Guru Gobind Singh Indraprastha University, East Delhi Campus, Surajmal Vihar, Delhi 110092, India.
mainiyash2@gmail.com

Sanjay Kumar Singh
University School of Automation and Robotics
Guru Gobind Singh Indraprastha University, East Delhi Campus, Surajmal Vihar, Delhi 110092, India.
sanjayksingh.012@gmail.com

Pranshu Saxena
School of Computer Science Engineering and Technology
Bennett University greater Noida, India
pranshusaxena@gmail.com

***Abstract*—Tomato cultivation is an important part of world agriculture, affecting both food security and economic stability. Tomato leaf diseases must be detected accurately to maintain crop health and maximize yield. This work improves disease classification by using deep learning models, notably CNN, Xception, DenseNet, and a hyper-tuned Xception model. We used transfer learning and data augmentation to improve model performance on the Kaggle dataset, which contains 10,000 training images and 1,000 validation images from the Plant Village Repository organized into ten groups. The hyper-tuned Xception model achieved the maximum performance by meticulously tweaking hyperparameters such as learning rate, batch size, and network architecture, with an accuracy of 97.23%, precision of 95.73%, recall of 94.40%, and F1-score of 95.06%. These results show a considerable improvement over the CNN, Xception, and DenseNet models. The results highlight the efficiency of sophisticated deep learning algorithms and hyperparameter tweaking in achieving high classification accuracy, making the proposed model a robust tool for practical tomato leaf disease diagnostics.**

Keywords—Leaf Image Analysis, hyperparameter, deep learning, Transfer Learning, Plant Disease Diagnosis.

I. INTRODUCTION

Tomatoes are among the most popular food crops in every kitchen throughout the world. Tomatoes (scientific name *Solanum lycopersicum*) can grow in practically any well-drained soil. Many gardeners grow tomatoes in their gardens to enjoy fresh, homegrown vegetables in their kitchens, which improves the flavor of their meals. However, farmers and gardeners sometimes struggle with the proper growth of tomato plants [1]. The plants may fail to bear fruit, or the tomatoes might develop unsightly, disease-ridden black spots on the bottom.

The detection of tomato plant disease may start by, diagnosing the infective portion in plants, noting the differences such as brown or black patches and holes on the plant and then looking for the insects. Tomatoes and similar crops like cabbage and potato must not be planted on the same farm for more than one time in three years as they compete for the same nutrients [2]. To preserve soil fertility, it's best to plant a member of the grass family, such as barley, wheat, millet, or sugarcane, before growing tomatoes.

Tomato-related problems can be divided into two categories: those produced by bacteria, fungi, or poor cultivation practices, which result in 16 diseases; and those caused by insects, which result in 5 unique diseases. The bacterium *Ralstonia solanacearum* causes severe bacterial wilt. This disease can survive in the soil for a long time and enters plant roots through natural openings created by secondary root growth or injuries produced by cultivation, transplanting, or insect activity [1]. Elevated moisture levels and hot temperatures hasten illness progression. The bacteria rapidly increase within the plant's water-conducting tissues, producing slime obstructing the vascular system. Despite this, the leaves may remain green. In a cross-sectional view of an infected stem, a brown discoloration is often accompanied by yellowish material exudation [2]. Some more diseases related to tomatoes are mentioned below in Table 1 and their pictorial representation is shown in Fig 1.

TABLE 1: TOMATO DISEASES RANKED BY SEVERITY FROM 0 (LEAST) TO 9(MOST).

Scale	Description
0	healthy: No disease.
1	Tomato_mosaic_virus: Mild viral leaf mottling.
2	Spider_mites Two-spotted_spider_mite: Tiny pests, leaf speckling.
3	Leaf_Mold: Fungal, yellow spots, mold on leaves.
4	Target_Spot: Fungal, brown leaf/fruit spots.
5	Septoria_leaf_spot: Fungal, small leaf spots.
6	Bacterial_spot: Bacterial, dark spots on leaves/fruit.
7	Early_blight: Fungal, leaf spots, defoliation.
8	Tomato_Yellow_Leaf_Curl_Virus: Viral, stunted growth, yellow leaves.
9	Late_blight: Severe fungal disease, rapid spread.

Each year, researchers advance and refine automatic methods for detecting and classifying plant leaf diseases. Techniques such as digital image processing, machine learning, and deep learning have been developed for this purpose. These methods are applied to a variety of plant leaves and demonstrate different learning rates. Automatic disease detection methods reduce pesticide use while increasing crop productivity and quality [3].

Advances in computer vision provide a chance to broaden and improve precise plant protection procedures, as well as Expand these approaches in precision agriculture [3]. Diseased leaf images can be identified and classified based on form and color using direct image processing methods or machine learning techniques including K-means clustering, SVM, ANN, CNN, and SR. Kamilaris and Prenafeta-Boldú [4] identified 40 research on land usage, plant species, soil classification, animal growth, weed identification, fruit counting, weather, product yield index, and moisture content. Estimation is used to provide solutions to many agricultural and food production concerns by using deep Learning approaches. It was reported in those investigations that the deep learning strategy produced greater accuracy and demonstrated better performance than traditional image processing approaches.

Deep learning with neural networks improves classification and accuracy by automatically extracting many features [5]. Convolutional Neural Network (CNN), a popular machine learning tool, outperforms standard classification methods in DNN [6]. Deep CNN was trained using meta-architectures such as VGGNet, LeNet, and ResNet to forecast tomato leaf disease, resulting in the best classification performance yet [7].

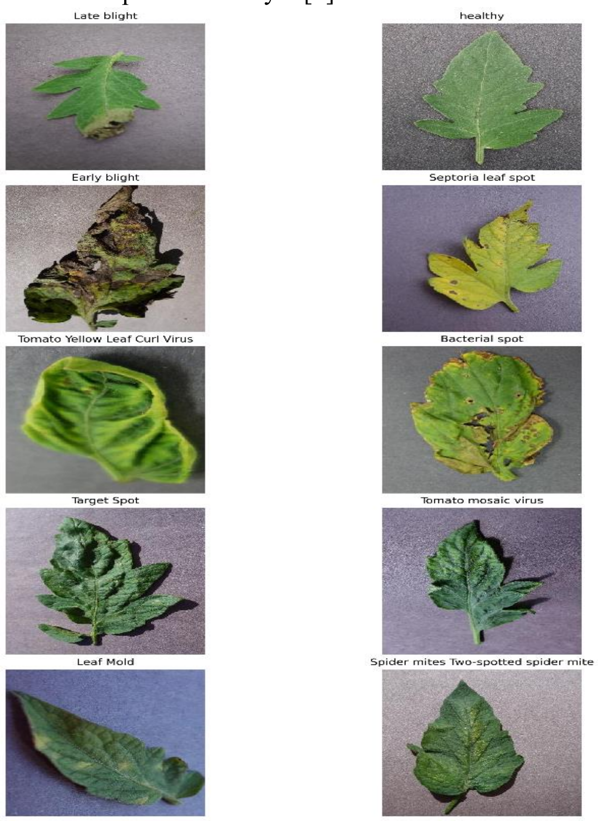

Fig 1. Sample images of Tomato leaves of 10 classes from the Dataset

This study attempts to improve tomato leaf spot disease detection by combining imaging-based expert systems with advanced deep-learning techniques. The research involves creating updated, hypertuned Convolutional Neural Network (CNN) models based on the Xception architecture. By methodically modifying and refining various parameters, the performance of these revised models is assessed and compared to multiple established architectures, such as the original Xception, DenseNet12, and classic CNN models. This comparative analysis is intended to evaluate advances in accuracy, precision, and overall diagnostic capability, shedding light on the efficacy of hyper-tuned Xception-based models in comparison to other cutting-edge deep learning methodologies. The goal is to determine the most effective approach for reliably diagnosing tomato leaf spot illnesses, which could lead to more dependable efficient tools for agricultural diagnostics.

II. LITERATURE REVIEW

Accurate classification of tomato leaf diseases is crucial for timely intervention, improving crop yield, and reducing agricultural losses. Tomato leaf disease classification has been solved using various machine learning and deep learning algorithms across multiple datasets, most notably the Plant Village dataset. A CNN comprising three convolutional and max-pooling layers, followed by two fully connected layers, scored 91.2% accuracy, 93.5% precision, and 92.4% F1. Hybrid techniques, such as rule-based feature selection and ECOC, increased accuracy to 98.8%. Transfer learning approaches such as AlexNet and SqueezeNet demonstrated potential, with SqueezeNet obtaining 94.3% accuracy and being chosen for mobile applications. Other strategies, such as frameworks developed by the Berkeley Vision and Learning Center, achieved 95.8% accuracy with fine-tuning. Furthermore, models such as the updated Faster R-CNN and GLCM feature extraction applied to sugar beet and other datasets demonstrated 95.5% and 98.1% accuracy gains, respectively. Multi-class SVM worked admirably in tomato leaf disease detection, obtaining 93.7% accuracy. For more generic detection, YOLO object detection improved feature representation but had a lower accuracy of 89%. LeNet was applied to the Plant Village repository and obtained 95% accuracy, demonstrating the capability of neural networks in difficult conditions. Finally, transfer learning using AlexNet on a Kaggle dataset outperformed ResNet and GoogleNet with 98% accuracy, 99% recall, and an F1-score of 98%. As seen in table 2.

TABLE 2. SUMMARY OF LITERATURE SURVEYS

References	**Dataset**	**Method Used**	**Key Contributions**	**Performance Evaluation**
[1]	Plant Village dataset	Convolutional Neural Networks (CNN)	Multi-class tomato leaf classification using CNN having 3 conv and 3 max-pooling layers with 2 FC layers	Accuracy-91.2%, Precision-93.5%, F1-Score-92.40%
[3]	Plant Village dataset	Hybrid Rule-Based Feature Selection and ECOC	Rule-based feature selection combined with ECOC for enhanced multi-class leaf classification	Accuracy-98.8%,
[4]	Gathered from Internet	DL framework developed by Berkley Vision and Learning Center	Utilized framework for extracting features from images, improving classification accuracy by increasing iterations and fine-tuning.	Accuracy-95.8%, Precision-96.3%,
[7]	Sugar beet leaf images dataset	Modified Faster R-CNN	The model outperformed other modern methods proposed in previous studies.	Accuracy-95.5%

References	Dataset	Method Used	Key Contributions	Performance Evaluation
[8]	Taken from plants directly	GLCM feature extraction and segmentation	Proposed OpenCv performed better than AlexNet and ANN..	Accuracy-98.1%
[9]	Taken from plants directly	Image Segmentation and Multi-class SVM	Multi-class SVM classifier successfully detected the tomato leaf disease	Accuracy-93.7%
[10]	Taken from plants and the Internet	Customized CNN and YOLO	YOLO object detection improved feature representation and generalization for tomato leaf classification.	Accuracy-89%
[11]	PlantVillage dataset	Transfer Learning with AlexNet and SqueezeNet	SqueezeNet is better for mobile DL classification than AlexNet	Accuracy-94.3%
[12]	Plant Village repository	LeNet	Feasibility of the NN approach under unfavorable conditions.	Accuracy-95%
[13]	Taken directly from Plants	DenseNet121, DenseNet161, Vgg16, ResNet34	Resnet34 gives better performance than other models	Accuracy-99.7%

The review of Table 2 on breast cancer classification using deep learning models highlights several key insights. Applying machine learning (ML) and deep learning (DL) approaches to tomato leaf disease classification shows great potential, but some crucial features require more investigation. While CNNs and hybrid models have amazing accuracy, can this translate into real-world utility? Models such as SqueezeNet, while suited for mobile use, may face issues when scaled to big farming operations requiring rapid, real-time processing. Another issue arises with models such as YOLO and LeNet, which, despite their strengths, fall short compared to other approaches. The constant excellent performance of transfer learning models, particularly AlexNet, which outperforms ResNet and GoogleNet, strongly justifies their employment. However, it has to be seen whether such accuracy advances can be sustained across a wide range of circumstances and environments. As these models mature, the crucial question is whether they can provide both scalability and constant reliability in the complex reality of modern agriculture.

III. Proposed Methodology

A. Database

The Kaggle Dataset named Tomato leaf disease detection used in this study consists of tomato leaf images inspired by the Plant Village Repository. This dataset has a training set of 10000 and a validation set of 1000 (table 3). The sets are categorized into ten classes: Healthy, Bacterial Spot, Early Blight, Late Blight, Septoria Leaf Spot, Tomato Mosaic Virus, Tomato Yellow Leaf Curl Virus, Target Spot, Leaf Mold, and Spider Mites Two-spotted Spider Mite (sample images of each class is represented in Fig 1), essential for effective classification, prediction of tomato leaf disease using ML and DL techniques. Thus, tomato leaf disease detection from Kaggle is a valuable resource for developing and testing machine learning models, ultimately contributing to improved classification and detection of diseases related to tomato leaf.

TABLE 3. IMAGE COUNT OF ALL CLASSES/LABELS PRESENT IN DATASET

Labels	Images in the training set	Images in the validation set
Healthy	1000	100
Bacterial Spot	1000	100
Early Blight	1000	100
Late Blight	1000	100
Septoria Leaf Spot	1000	100
Tomato Mosaic Virus	1000	100
Tomato Yellow Leaf Curl Virus	1000	100
Target Spot	1000	100
Leaf Mold	1000	100
Spider Mites Two-spotted Spider Mite	1000	100

B. Deep Learning Architectures

This study examines two advanced convolutional neural network architectures: DenseNet and Xception. DenseNet introduces dense connections, which connect each layer to the layer before it, increasing feature reuse and enhancing gradient flow. Xception builds on the Inception architecture, utilizing depthwise separable convolutions to reduce computational cost while maintaining high accuracy. Both models, pre-trained on ImageNet, are commonly used in picture categorization and object detection tasks.

C. Proposed Model

In this study, we tuned the hyperparameters of the Xception model and applied fine tuning to improve its performance for picture categorization. We wanted to improve model accuracy and generalization by altering critical parameters like the number of units in thick layers, dropout rates, and learning rates. The pre-trained Xception base was employed, followed by further layers for categorization. three models, pre-trained on ImageNet, are commonly used in picture categorization and object detection tasks.

i. Depthwise Separable Convolutions (Xception)

Xception substitutes standard convolutions with depthwise separable convolutions, which drastically reduce parameters and processing costs. This method divides a normal convolution into two steps: depthwise convolution, in which each input channel applies its filter individually, and pointwise convolution, which uses a 1x1 kernel to process individual pixels as shown in Fig 2. This combination effectively reproduces the effect of regular convolutions. The center flow block of Xception is made up of eight similar blocks, each of which uses depthwise separable convolutions with residual connections. These residual linkages allow the output of each convolutional layer to be combined with the original input, resulting in improved gradient flow and training [14]. Each block contains depthwise and pointwise convolutions, batch normalization, and ReLU activations, which steadily increase the number of channels while maintaining spatial dimensions.

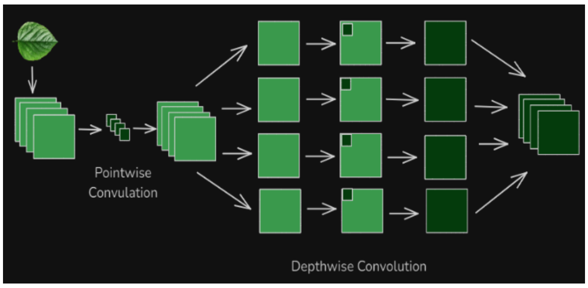

Fig 2. Depthwise Separable Convolutions architecture

ii. Tuning of Hyperparameters

Combining Bayesian Optimization and Population-Based Training (PBT), a hybrid technique that mixes global exploration with dynamic hyperparameter tuning can efficiently optimize hyperparameters for complicated models such as Xception. Initially, Bayesian optimization [15] is used to efficiently explore the hyperparameter space, picking promising values for important parameters like units_dense1, dropout, units_dense2, and learning_rate. For the Xception model, the optimum hyperparameters discovered using Bayesian optimization are mentioned in Table 4.

TABLE 4. HYPERPARAMETER VALUES OBTAINED USING BAYESIAN OPTIMIZATION

units_dense1	448
dropout	0.2
units_dense2	160
learning_rate	0.000791

The objective function in Bayesian Optimization aims to maximize the model's performance, represented as:

$$\theta^* = arg\ max\ f(\theta)$$

The hyperparameters (e.g., units_dense1, dropout, units_dense2, learning_rate) are represented by θ, whereas the validation accuracy or loss is measured by $f\ (\theta)$. These parameters aid in the initialization of a population of Xception models with various yet promising hyperparameter values.

Once initialized, PBT takes over and continuously evaluates the model population during training. It updates underperforming models with the hyperparameters of high-performing models while allowing for dynamic change via perturbations [16]. For example, the learning rate can be changed as follows:

$$\eta' = \eta \times (1 + \delta)$$

δ is a minor disturbance factor. This change guarantees that the model responds to the changing dynamics of training. Real-time tuning is critical for Xception, as the learning rate is often reduced as training goes to avoid overshooting the loss minima.

Combining Bayesian Optimization's global search and PBT's dynamic tuning produces significant benefits. This hybrid technique can result in a 2-3% boost in validation accuracy over typical static tuning methods while also reducing training time by 20-30% because the model converges faster due to the continuously developing hyperparameters. As a result, this strategy efficiently optimizes Xception, balancing exploration and exploitation while minimizing computing overhead.

iii. Finetuning of Xception

The proposed modified Xception model, seen in Fig. 3, enhances the existing Xception architecture with several significant enhancements aimed at increasing both performance and efficiency. The incorporation of dilated convolutions is a significant improvement. Dilated convolutions increase the receptive field while decreasing the number of parameters and computational complexity. This is mathematically represented as:

$$\boldsymbol{Y(i,j) = \sum m = 0K - 1X(i + m \cdot D, j + nD)\,.\,W(m,n) + b}$$

In addition, the model uses squeeze-and-excitation (SE) blocks to adaptively adjust feature maps, allowing the network to focus more effectively on relevant characteristics. The SE blocks recalibrate feature maps through the following equation:

$$\boldsymbol{\sigma c = sigmoid(W2 \cdot ReLU(W1 \cdot zc))}$$

The architecture also includes a more efficient sequence of custom layers, such as thick layers with batch normalization and dropout to improve regularization. By reducing the number of residual blocks and utilizing skip connections, the enhanced Xception tackles the fading gradient issue while remaining lightweight. These enhancements improve the model's performance.

D. Performance Metrics

Key metrics to consider when assessing the performance of deep learning models are accuracy, recall, precision, and F1-score, particularly for classification tasks. By computing the ratio of accurate forecasts to total predictions, accuracy evaluates the model's overall soundness. Although accuracy has its uses, unbalanced datasets cannot respond well to it as an indication. By calculating the percentage of actual positive outcomes among all expected positive outcomes, precision focuses on the caliber of positive forecasts. Recall demonstrates how well the model can separate real positives from all pertinent examples and shows how well it can catch all positive cases. Finally, the F1-score balances precision and recall by computing their harmonic mean, making it a more comprehensive metric when dealing with class imbalances, ensuring that false positives and false negatives are considered. These metrics together provide a well-rounded understanding of model performance

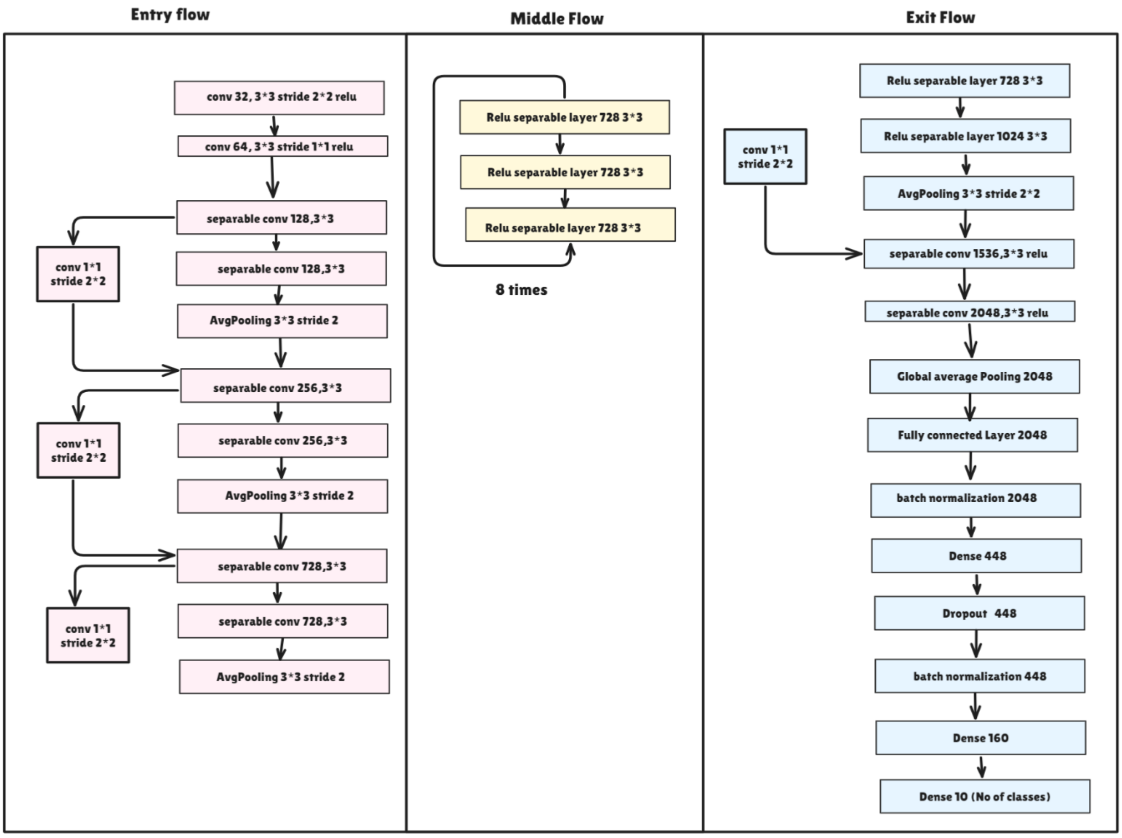

Fig. 3. The layering architecture of the proposed finetuned Xception

IV. Result and Discussion

Our transfer learning approach aimed to identify diseased tomato leaf images among normal images correctly. For this, we several models mentioned above and trained them separately. We trained and tested on a Google Colab machine equipped with an Intel(R) Core (TM) i5-12500H (Intel Corporation, Santa Clara, CA, USA) processor, 12 GB of system RAM, and a T4 inbuilt GPU. For training, we employed the Adam optimizer and the cross-entropy loss function. Hypertuned Xception was fine-tuned and was trained for 50 iterations, with a learning rate of 0.0000791 with a tolerance of 10 which stopped at 27 epochs it achieved an overall accuracy of 97.23%.

Table 4. Comparing the performance of all transfer learning techniques by evaluating their Accuracy, Precision, Recall, and F1 score, and on epochs layer

	Epochs	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
CNN	27	86.44	84.82	83.97	84.39
Xception	27	87.39	85.90	84.30	85.09
DenseNet	27	96.90	92.07	94.10	93.07
Propoesed Method	**27**	**97.23**	**95.73**	**94.40**	**95.06**

Table 4 provides a comparison of 4 deep learning models CNN, Xception, DenseNet, and hypertuned Xception—based on their performance across four key metrics: accuracy, precision, recall, and F1-score, along with the number of epochs trained. present exhibits the best overall performance, achieving the highest accuracy (**97.23**%), precision (**95.73**%), recall (94.40%), and F1 score (**95.06**%) after 27 epochs. DenseNet follows closely with an accuracy of 96.90%, along with strong precision (92.07%) and F1 score (93.07%) after 27 epochs. Xceptions perform, achieving a balanced precision (85.90%) and recall (84.30%) with an F1 score of 85.09% after 27 epochs. Basic CNN trails the other model with the lowest 86.44.30% accuracy and an F1 score of 84.39%.

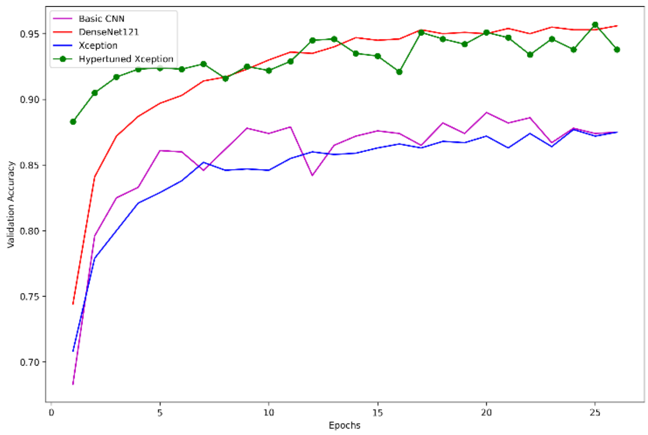

Fig. 4. Performance graph of proposed hyper-tuned Xception with conventional deep networks based on accuracy

Fig. 4 compares the training accuracy of four models-Basic CNN, Xception, DenseNet, and hypertuned Xception—over 27 epochs. Hypertuned Xception consistently shows the highest accuracy, starting at 0.88 and progressing to 0.95 by epoch 27 although there are many dips in between, indicating strong performance across the training process. DenseNet begins at a lower accuracy (0.74) and significantly improves, reaching 0.95 by the final epoch.

Xception and Proposed CNN follow a similar trend, with both maintaining competitive accuracy levels of around 0.70–0.85 giving an edge-to-edge competition to each other Overall, DenseNet and Hypertuned Xception outperform the other models, while CNN and Xception show slower progress compared to the others, highlighting varying levels of model performance throughout the training cycle. We can further extend this discussion to the training loss where we see the same results for different models in Fig 5.

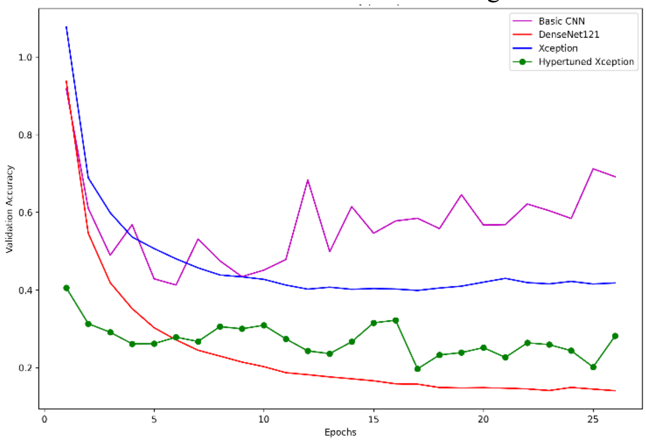

Fig. 5. Training loss of proposed hyper-tuned Xception with other conventional deep networks.

Fig 4 and Fig 5 also give a visual comparison of how well each model performs by showing the results of its training accuracy and training loss on sample images across all models. It also shows, from the side-by-side visualization, that DenseNet and Hypertuned Xception have very consistent accuracy. More importantly, it gives an idea of how the two models interpret and process the image data. On the other hand, Xception performs not great as compared to both. This gives us the idea to select this model to tune its parameters and later apply fine-tuning to it.

For tomato leaf disease detection, a detailed analysis of leaf images is essential. While deep learning models like HyperTuned Xception, DenseNet, Xception, and CNN have been applied, explainability remains an issue because these models frequently provide little insight into how they make their final predictions. To improve the decision support process, these models should not only produce accurate predictions but also provide reasons for their judgments, maybe through semantic segmentation that highlights relevant parts of the leaf. In this investigation, the HyperTuned Xception model fared better than the rest, followed by DenseNet, Xception, and CNN. However, a fundamental barrier is the scarcity of image data, which can impair accuracy and prevent deeper models from achieving peak performance. In future study, incorporating larger, annotated datasets and data augmentation approaches may improve model accuracy while lowering the danger of overfitting.

V. CONCLUSION

To summarize, this study demonstrates the significant potential of deep learning models, notably the hyper-tuned Xception, for improving tomato leaf disease categorization. The proposed model improves accuracy and efficiency significantly by optimizing hyperparameters such as learning rate, dropout, and dense units, as well as using advanced approaches like as dilated convolutions and squeeze-and-excitation (SE) blocks. The increases in performance parameters such as accuracy, precision, recall, and F1 score highlight the importance of model fine-tuning in achieving improved generalization. This makes the optimized model's excellent candidates for agricultural diagnostics applications. Future work will incorporate explainability elements to test these models and provide transparency in real-world applications.

REFERENCES

[1] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato leaf disease detection using convolution neural network," in *Procedia Computer Science*, vol. 167, pp. 293-301, 2020.

[2] E. Moriones and J. Navas-Castillo, "Tomato yellow leaf curl virus, an emerging virus complex causing epidemics worldwide," in *Virus Research*, vol. 71, no. 1-2, pp. 123-134, 2000.

[3] M. Kaur and R. Bhatia, "Development of an improved tomato leaf disease detection and classification method," in *2019 IEEE Conference on Information and Communication Technology*, 2019, pp. 1-5.

[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," in *Computational Intelligence and Neuroscience*, vol. 2016, no. 1, pp. 3289801, 2016.

[5] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," in *Computers and Electronics in Agriculture*, vol. 147, pp. 70-90, 2018.

[6] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," in *Sensors*, vol. 17, no. 9, pp. 2022, 2017.

[7] M. M. Ozguven and K. Adem, "Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms," in *Physica A: Statistical Mechanics and its Applications*, vol. 535, pp. 122537, 2019.

[8] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. Sophia, and B. Pavithra, "Tomato leaf disease detection using deep learning techniques," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 979-983.

[9] S. Vetal and R. S. Khule, "Tomato plant disease detection using image processing," in *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 6, pp. 293-297, 2017.

[10] S. Adhikari, D. Unit, B. Shrestha, and B. Baiju, "Tomato plant diseases detection system," in *1st KEC Conference Proceedings*, vol. 1, pp. 81-86, 2018.

[11] H. Durmuş, E. O. Güneş, and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," in *2017 6th International Conference on Agro-Geoinformatics*, 2017, pp. 1-5.

[12] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1-5.

[13] M. Ouhami, Y. Es-Saady, M. E. Hajji, A. Hafiane, R. Canals, and M. E. Yassa, "Deep transfer learning models for tomato disease detection," in *Image and Signal Processing: 9th International Conference, ICISP 2020*, Marrakesh, Morocco, 2020, pp. 65-73.

[14] N. Anshar Ali and T. Regula, "Enhancing Facial Gender Classification: A Deep Dive into Xception Neural Networks and Ensemble Frameworks," 2024.

[15] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[16] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, and K. Kavukcuoglu, "Population based training of neural networks," in *arXiv preprint arXiv:1711.09846*, 2017.