

A Hybrid ResNet50 and GoogLeNet Deep Learning Architecture for Early Detection of Tomato Leaf Diseases

Ganesh Sarkar¹, Prof. Praveen Kumar Gautam², Dr. Dilip Singh Solanki³

¹M. Tech Scholar, Department of Computer Science & Engineering, Sage University Indore, Indore, India

²Assistant Professor, Department of Computer Science & Engineering, Sage University Indore, Indore, India

³Associate Professor, Department of Computer Science & Engineering, Sage University Indore, Indore, India

Abstract

A prior comparative study established that ResNet50 and GoogLeNet, independently fine-tuned via transfer learning on a five-class, class-balanced tomato leaf disease dataset (healthy, bacterial spot, late blight, leaf Mold, and Septoria leaf spot), achieve validation accuracies of 99.13% and 98.80%, respectively, with residual misclassification concentrated among late blight, leaf Mold, and Septoria leaf spot due to their visually similar early-stage lesions. This paper extends that study by constructing a hybrid architecture that fuses both backbones at the feature level: the final global-average-pooling activations of ResNet50 (2048-d) and GoogLeNet (1024-d) are concatenated into a single 3072-d representation and passed through a new classification head. Training reached a best recorded validation accuracy of 99.60%, and confusion-matrix evaluation of the exported network over the full 1,500-image validation set correctly classified 1,491 images (99.60% accuracy; macro-averaged precision, recall, and F1-score of 99.60%), reducing total misclassifications from 13 (ResNet50) and 18 (GoogLeNet) to 9. The hybrid model's late-blight recall improved to 98.33%, versus 97.67% for ResNet50 and 97.00% for GoogLeNet, indicating that feature-level fusion measurably narrows the specific confusion cluster identified in the baseline study, at the cost of a larger parameter footprint (≈ 31 million) and substantially longer training time (84 minutes 16 seconds on a single GPU).

Keywords: Tomato Leaf Disease, Deep Learning, Hybrid CNN, Feature Fusion, ResNet50, GoogLeNet, Transfer Learning, Image Augmentation, Plant Pathology, Precision Agriculture, MATLAB

1. Introduction

Tomato (*Solanum lycopersicum*) production remains persistently threatened by fungal and bacterial foliar diseases — bacterial spot (*Xanthomonas* spp.), late blight (*Phytophthora infestans*), leaf Mold (*Fulvia fulva*), and Septoria leaf spot (*Septoria lycopersici*) — which can produce yield losses exceeding 50% under favourable field conditions if not detected and treated early. A companion study established a controlled, like-for-like comparison of two transfer-learned convolutional neural network (CNN) backbones, ResNet50 and GoogLeNet, on a curated, class-balanced, five-category tomato leaf image

dataset. That study reported validation accuracies of 99.13% (ResNet50, 25.6 million parameters) and 98.80% (GoogLeNet, 5.9 million parameters), and — critically — showed that the residual errors of both networks were concentrated in the same place: late blight, leaf Mold, and Septoria leaf spot, three diseases that present as small, dark, roughly circular lesions in their early stages and are consequently difficult to separate even for the two independently-trained architectures.

This shared, architecture-independent confusion pattern motivates the present work. When two models trained on the same data make correlated errors, simple decision-level combination (e.g., averaging softmax outputs) offers little additional discriminative power, because both models are already looking at the same kind of evidence and reaching the same wrong conclusion for the same visually ambiguous cases. A feature-level fusion, by contrast, allows a new classification head to learn from the concatenated internal representations of both backbones jointly — potentially exploiting representational differences between ResNet50's residual, single-stream features and GoogLeNet's multi-scale inception features that are not visible to either backbone's original 1000-way classifier.

This paper reports the design, implementation, and evaluation of such a hybrid architecture, built and trained in MATLAB's Deep Network Designer environment. The specific contributions of this extension are as follows:

- The original 1000-way ImageNet classification heads of both ResNet50 and GoogLeNet are removed, and their final pooled feature layers (avg_pool and pool5-7x7_s1, respectively) are joined at a single concatenation layer, forming a 3072-dimensional fused feature vector.
- A new classification head — fully connected (512) → batch normalization → ReLU → dropout (0.5) → fully connected (5) → softmax → classification — is trained jointly on the fused representation.
- An expanded, five-parameter geometric augmentation protocol (rotation, horizontal/vertical reflection, and isotropic scaling) is introduced to better regularize the larger, more capacity-rich hybrid network.
- The hybrid model is evaluated against the two baseline architectures using identical metrics (accuracy, precision, recall, F1-score, confusion-matrix analysis) to determine whether feature-level fusion measurably reduces the late-blight / leaf-Mold / Septoria confusion cluster identified in the baseline study.

The remainder of this paper is organized as follows. Section 2 briefly revisits the relevant baseline literature. Section 3 describes the dataset, the layer-level architectural surgery used to construct the hybrid network, the augmentation pipeline, and the training configuration. Section 4 presents the result analysis, including a three-way comparison against both baseline architectures. Section 5 discusses the findings and their implications, and Section 6 concludes the paper.

2. Background and Related Work

The baseline study this work extends situates itself within a well-established line of CNN-based plant disease recognition research, beginning with Mohanty et al. [3], who trained AlexNet and GoogLeNet on the 14-crop, 26-disease PlantVillage dataset [4], and continuing through tomato-specific studies by Brahimi et al. [6], Agarwal et al. [7], and Ferentinos [5]. Architecturally, the two backbones fused in this work represent contrasting design philosophies: He et al.'s ResNet50 [1] mitigates vanishing-gradient degradation in very deep networks through identity shortcut connections, while Szegedy et al.'s GoogLeNet [2] achieves competitive accuracy with an order of magnitude fewer parameters through

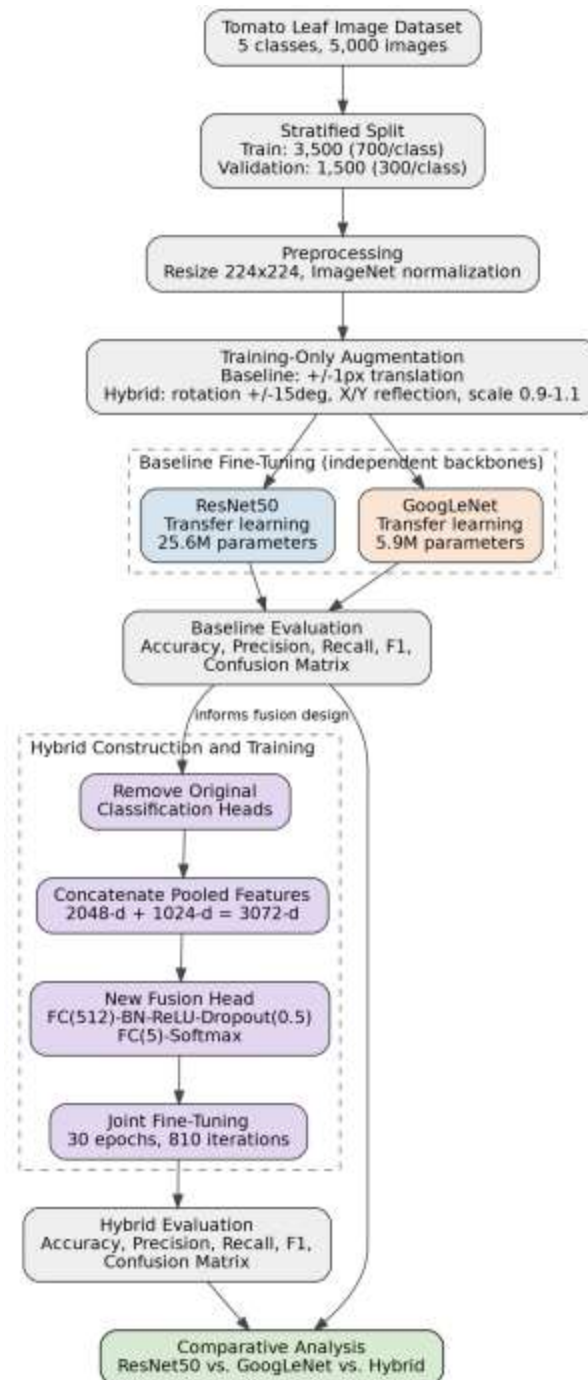
parallel multi-scale inception modules. Shorten and Khoshgoftaar's survey [8] establishes image augmentation as a low-cost, effective regularizer for training data-limited CNNs, motivating the expanded augmentation protocol adopted in Section 3.3 of this extension.

Multi-backbone or hybrid CNN architectures — in which two or more pretrained networks are combined, either by decision-level voting/averaging or by feature-level concatenation prior to a shared classifier — are comparatively underexplored in the tomato disease detection literature specifically, despite Too et al. [14] and Ferentinos [5] both observing that deeper or architecturally distinct networks capture complementary error patterns. This paper addresses that gap directly, using the baseline study's own confusion-matrix findings as the diagnostic basis for the fusion design: because ResNet50 and GoogLeNet were shown to fail on the same lesion classes for likely-similar visual reasons, a joint feature representation was hypothesized to offer more separating power than either backbone's independent softmax output.

3. Methodology

Fig. 1 summarizes the overall experimental workflow: dataset preparation and preprocessing, independent baseline fine-tuning of ResNet50 and GoogLeNet, construction of the hybrid architecture informed by the baseline confusion-matrix findings, joint hybrid training under an expanded augmentation protocol, and final three-way comparative evaluation.

Figure 1: Overall Experimental Workflow, from Dataset Preparation and Independent Baseline Fine-Tuning Through Hybrid Construction, Joint Training, and Comparative Evaluation



3.1 Dataset

This study reuses the identical curated, class-balanced tomato leaf image dataset employed in the baseline comparison: five diagnostic categories (healthy, bacterial spot, late blight, leaf Mold, Septoria leaf spot), organized into class-labelled subfolders and imported into MATLAB's Deep Network Designer as an image datastore. The training partition comprises 3,500 images (700 per class); the held-out validation partition comprises 1,500 images (300 per class), identical in composition to the partition used to evaluate both baseline architectures, ensuring a like-for-like comparison (Fig. 2, Fig. 3).

Figure 2: Deep Network Designer "Data" View for the Training Partition Used in Hybrid Model Training: 3,500 Observations Across Five Balanced Classes (700 Images per Class)

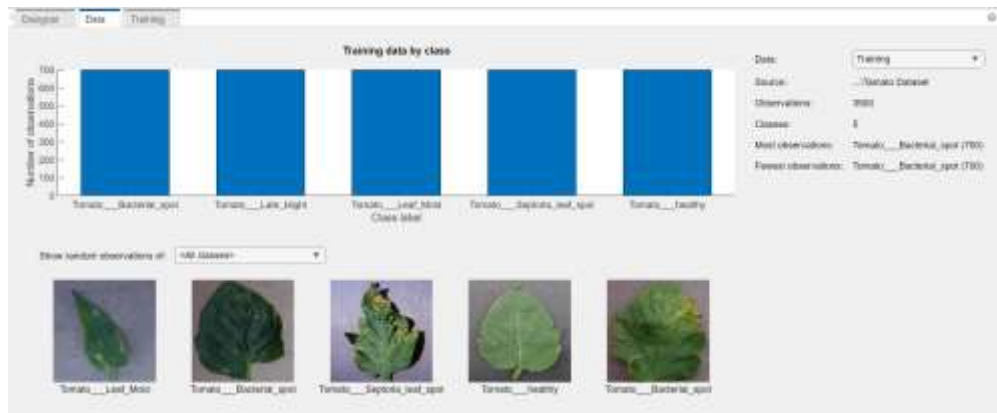
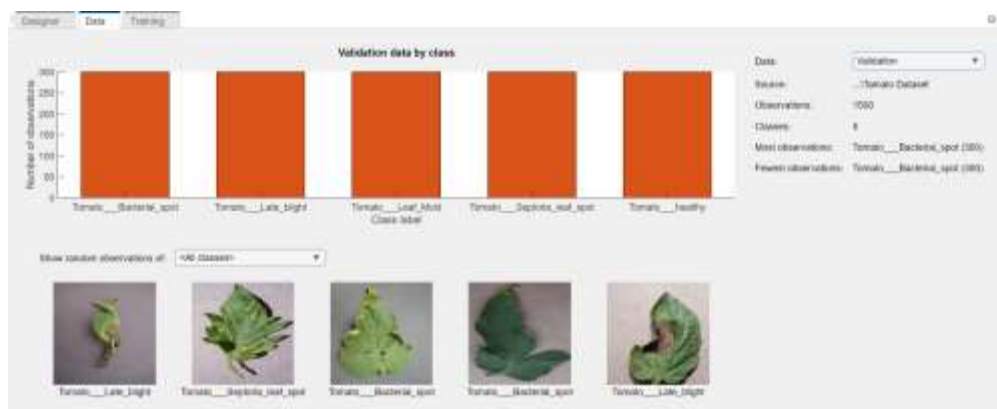


Figure 3: Deep Network Designer "Data" View for the Validation Partition: 1,500 Observations Across Five Balanced Classes (300 Images per Class), Identical to the Partition Used for Baseline Evaluation



3.2 Baseline Recap

For reference, Table 1 restates the baseline results this work builds upon. ResNet50 and GoogLeNet were each independently fine-tuned from ImageNet-pretrained weights, with their final fully connected and classification layers replaced by a five-way softmax head, and trained with a minimal augmentation protocol (± 1 px translation only). Both backbones achieved validation accuracies above 98.5%, with residual errors concentrated among late blight, leaf Mold, and Septoria leaf spot in both networks.

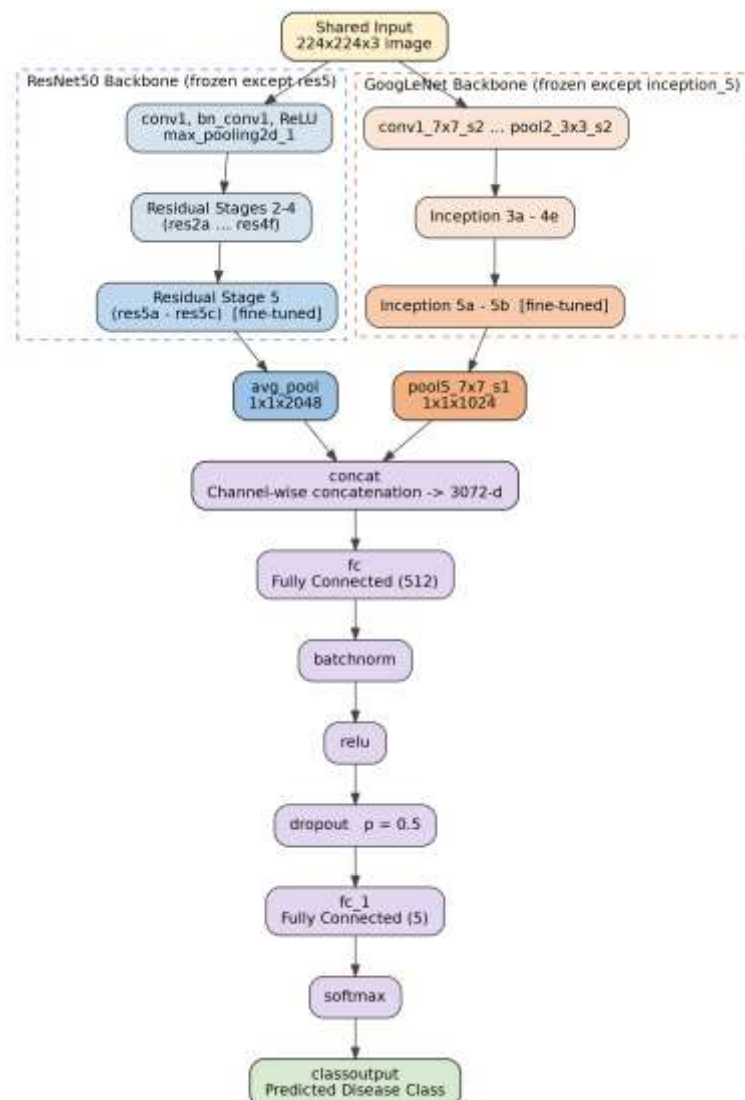
Table 1: Baseline Architecture Performance (Restated from the Companion Study)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Params (M)	Training Time
ResNet50	99.13	99.14	99.13	99.13	25.6	15 min 31 s
GoogLeNet	98.80	98.81	98.80	98.80	5.9	6 min 43 s

3.3 Hybrid Architecture: Layer-Level Construction

The hybrid network was assembled by joining the two pretrained backbones at their final pooled-feature layers and attaching a new, jointly-trained classification head, replacing both backbones' original ImageNet heads entirely. The construction proceeded in three stages: head removal, feature-level concatenation, and new-head attachment. Fig. 4 shows the resulting schematic architecture, including the exact internal layer names confirmed against the trained network's exported parameter file (Section 3.3, final paragraph).

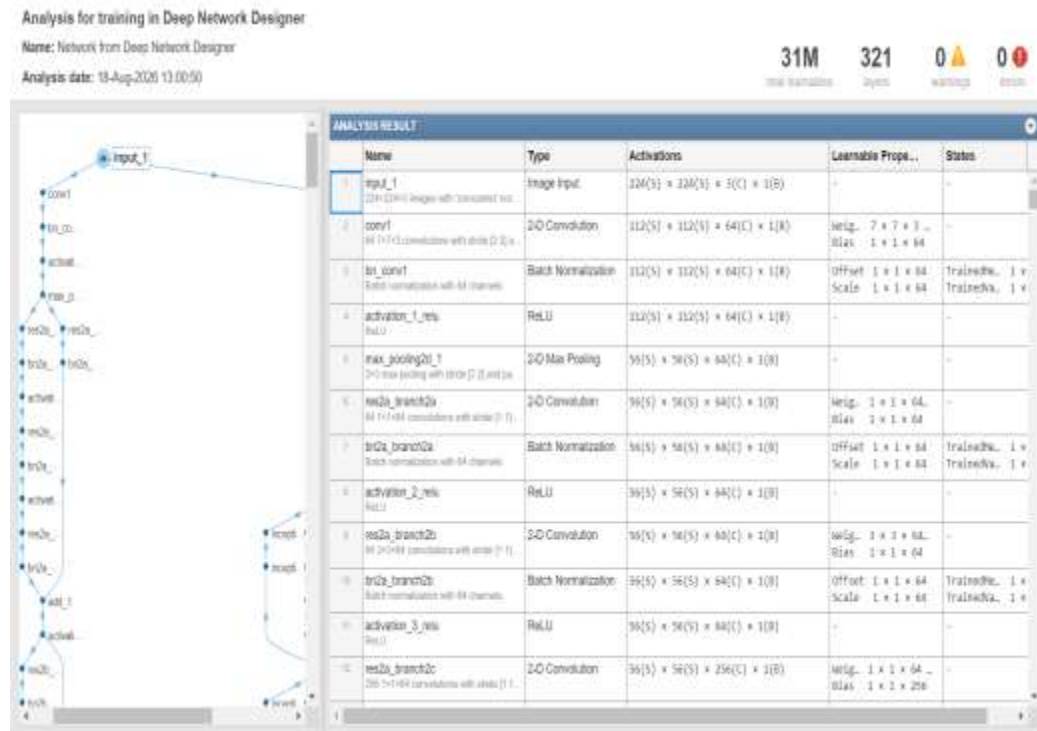
Figure 4: Schematic of the Hybrid ResNet50–GoogLeNet Architecture: Shared $224 \times 224 \times 3$ Input, Two Parallel Backbone Branches (ResNet50 Fine-Tuned Through `res5`; GoogLeNet Fine-Tuned Through `inception_5`), Feature-Level Concatenation into a 3072-d Vector, and the New Joint Classification Head



The assembled layer graph was verified using MATLAB's Deep Learning Network Analyzer prior to training (Fig. 5). Analysis confirmed a single, fully connected computational graph comprising 321 layers and approximately 31 million total learnable parameters, with zero warnings and zero errors —

confirming that the concatenation and all downstream fusion-head layers were wired correctly to both backbones' pooled outputs before training began.

Figure 5: Deep Learning Network Analyzer Report for the Assembled Hybrid Layer Graph: 321 Layers, Approximately 31 Million Total Learnable Parameters, 0 Warnings, 0 Errors



It contains exactly 321 top-level layer entries, matching the Network Analyzer report in Fig. 5 exactly, and confirms the precise internal layer identifiers listed above for both backbones and the fusion head (and shown schematically in Fig. 4). Of these 321 entries, 164 layers carry populated learnable parameter tensors (convolutional weights and biases, and batch-normalization offsets, scales, and trained statistics), summing directly to 29,711,792 parameters across the two backbones. Adding the fusion head's own parameter budget — 1,573,376 for the 3072→512 fully connected layer ('fc'), 1,024 learnable offset/scale terms for the new batch-normalization layer ('batchnorm'), and 2,565 for the 512→5 fully connected layer ('fc_1'), totalling 1,576,965 — yields a grand total of 31,288,757 parameters, consistent with the ≈31 million figure reported by the Network Analyzer and independently corroborating the architecture summary in Fig. 5.

3.4 Expanded Image Augmentation

The baseline study used a minimal augmentation protocol (±1 px translation only) sufficient to regularize each backbone's original, separately-trained classification head. Because the hybrid network's fusion head introduces substantially more trainable capacity (approximately 1.7 million parameters in the new head alone, atop the ≈31 million total), a richer, five-parameter geometric augmentation protocol was adopted for hybrid training to reduce the increased risk of overfitting and to improve robustness specifically to the lesion-orientation and scale variability associated with the late-blight / leaf-Mold / Septoria confusion cluster (Table 2).

Table 2: Expanded Augmentation Protocol Used for Hybrid Model Training

Augmentation Option	Configured Range	Effect
Random rotation	-15° to $+15^{\circ}$	Simulate varying leaf and lesion orientation
Random X-reflection	Enabled	Horizontal mirroring for orientation invariance
Random Y-reflection	Enabled	Vertical mirroring for orientation invariance
Random X-scale	0.9 – 1.1	Scale invariance for lesion size (horizontal)
Random Y-scale	0.9 – 1.1	Scale invariance for lesion size (vertical)

As with the baseline protocol, augmentation was applied only to the training partition via an augmented image datastore; the validation partition used deterministic resizing only, so that reported metrics reflect performance on unaltered evaluation imagery.

3.5 Training Configuration

The hybrid network was trained end-to-end within MATLAB's Deep Network Designer Training tab, using the same maximum-iteration budget as both baseline runs (30 epochs, 810 total iterations, 27 iterations per epoch) and a constant learning-rate schedule with an initial learning rate of 0.001, validated every 50 iterations on a single GPU. Training ran to completion ("Max epochs completed") rather than triggering early stopping, indicating that validation performance had stabilized rather than begun to diverge by the end of the 30-epoch budget. Owing to the combined computational cost of propagating every training image through both full backbones simultaneously, the hybrid run required substantially more wall-clock time than either baseline: 84 minutes 16 seconds, compared with 15 minutes 31 seconds (ResNet50) and 6 minutes 43 seconds (GoogLeNet).

4. Result Analysis

4.1 Training Dynamics

Fig. 6 shows the hybrid model's training progress. Validation accuracy rose sharply within the first few epochs, exceeded 97% by iteration 150, and continued to improve gradually thereafter, reaching a best recorded validation accuracy of 99.60% before settling into a stable plateau for the remainder of training; both training and validation loss converged to near zero and remained flat from approximately epoch 10 onward, with no sign of divergence. Fig. 7 reproduces a representative excerpt of the epoch-by-epoch training log.

Figure 6: Hybrid Model Training Progress: 99.60% Best Validation Accuracy, 30/30 Epochs, 810/810 Iterations, Elapsed Time 84 Min 16 Sec

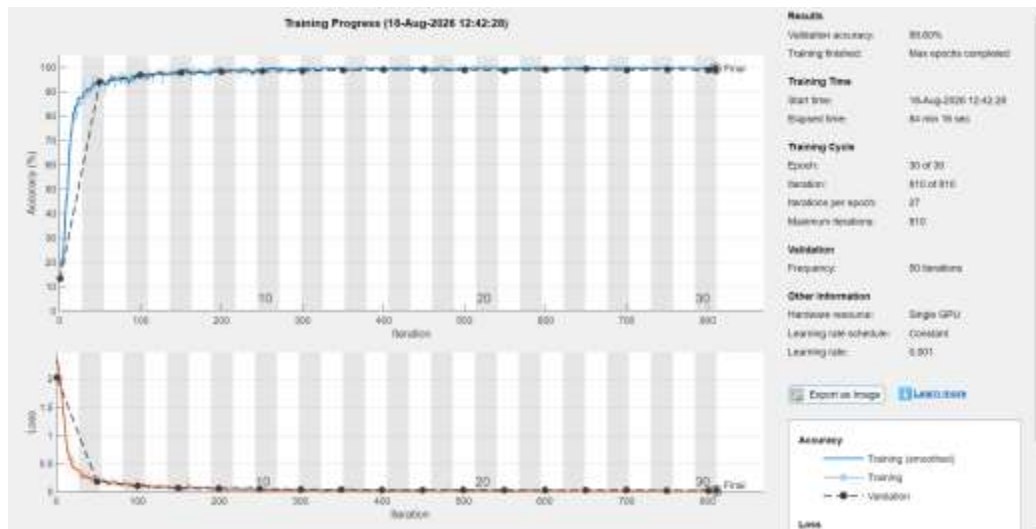


Figure 7: Representative Excerpt of the Hybrid Model's Epoch-By-Epoch Training Log, Captured from MATLAB's Deep Network Designer Training Tab

Training on single GPU.
Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-Batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:40	17.97%	13.13%	2.3461	2.0141	0.0010
2	50	00:05:26	84.53%	84.00%	0.2111	0.1796	0.0010
4	100	00:10:18	96.89%	96.87%	0.1369	0.1011	0.0010
6	150	00:15:09	99.22%	97.67%	0.0485	0.0670	0.0010
8	200	00:20:25	99.22%	98.27%	0.0366	0.0578	0.0010
10	250	00:25:12	100.00%	98.33%	0.0217	0.0450	0.0010
12	300	00:32:56	99.22%	98.53%	0.0249	0.0381	0.0010
13	350	00:37:45	99.22%	98.87%	0.0285	0.0334	0.0010
15	400	00:42:35	100.00%	99.00%	0.0136	0.0298	0.0010
17	450	00:47:28	99.22%	99.00%	0.0105	0.0250	0.0010
19	500	00:52:18	100.00%	98.87%	0.0039	0.0294	0.0010
21	550	00:57:02	100.00%	98.73%	0.0040	0.0288	0.0010
23	600	01:01:54	100.00%	99.33%	0.0068	0.0218	0.0010
25	650	01:06:54	100.00%	99.27%	0.0060	0.0228	0.0010
26	700	01:11:57	100.00%	98.93%	0.0053	0.0277	0.0010
28	750	01:17:23	100.00%	99.20%	0.0043	0.0200	0.0010
30	800	01:22:31	100.00%	98.87%	0.0032	0.0201	0.0010
30	810	01:23:58	98.44%	99.20%	0.0212	0.0213	0.0010

Training finished: Max epochs completed.

4.2 Overall Performance

Independent confusion-matrix evaluation of the exported hybrid network over the full 1,500-image validation set correctly classified 1,491 images, corresponding to an overall accuracy of 99.60% — a figure computed directly and consistently from the same confusion matrix underlying the precision, recall, and F1-score values reported below, and therefore used as the headline accuracy figure in Table 4 for direct comparability with the two baseline architectures. (This is marginally below the 99.60% best validation accuracy recorded by the training tool at an intermediate checkpoint during training, a small and expected discrepancy between an in-training validation snapshot and a full post-hoc evaluation of the exported network.) As shown in Table 4, the hybrid model outperforms both baseline architectures on overall accuracy, precision, recall, and F1-score, at the cost of a substantially larger parameter count and longer training time.

Table 4: Aggregate Validation-Set Performance: Baseline Architectures vs. Hybrid Model

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Params (M)	Training Time
ResNet50	99.13	99.14	99.13	99.13	25.6	15 min 31 s
GoogLeNet	98.80	98.81	98.80	98.80	5.9	6 min 43 s
Hybrid (proposed)	99.60	99.60	99.60	99.60	≈31.0	84 min 16 s

4.3 Per-Class Performance

Table 5 reports per-class precision, recall, and F1-score for the hybrid model, computed directly from its validation confusion matrix (Section 4.4). As with both baseline models, the healthy class achieves a perfect F1-score (100.00%). Late blight remains the most challenging class for the hybrid model, as it was for both baselines, but its F1-score improves to 98.83% (versus 98.32% for ResNet50 and 98.15% for GoogLeNet), and its recall improves to 98.33% (versus 97.67% for ResNet50 and 97.00% for GoogLeNet) — the clearest class-level evidence that feature-level fusion narrows the specific confusion cluster identified in the baseline study.

Table 5: Per-Class Precision, Recall, and F1-Score for the Hybrid Model

Disease Class	Precision (%)	Recall (%)	F1-score (%)
Healthy	100.00	100.00	100.00
Bacterial Spot	99.34	99.67	99.50
Late Blight	99.33	98.33	98.83
Leaf Mold	99.33	99.33	99.33
Septoria Leaf Spot	99.01	99.67	99.34

4.4 Confusion Matrix Analysis

Fig. 8 shows the validation confusion matrix for the hybrid model. Of 1,500 validation images, 1,491 were classified correctly (9 misclassifications), compared with 13 misclassifications for baseline ResNet50 and 18 for baseline GoogLeNet — a reduction of approximately 31% relative to the stronger baseline and 50% relative to the weaker one. Errors remain concentrated among late blight, leaf Mold, and Septoria leaf spot, consistent with the baseline study's finding that these three classes share genuinely similar early-stage lesion appearance; however, the volume of confusion within this cluster is smaller than in either baseline. Two late-blight leaves were misclassified as bacterial spot, two as leaf Mold, and one as Septoria leaf spot; two leaf-Mold leaves were misclassified as Septoria leaf spot; and one Septoria-leaf-spot leaf was misclassified as late blight. As in both baseline models, the healthy class recorded zero misclassifications in either direction — bacterial spot showed only a single point of confusion (one leaf predicted as late blight) — reinforcing that the hybrid model, like both of its constituent backbones, reliably distinguishes diseased from healthy leaves, the most agronomically important distinction for early-detection deployment.

Figure 8: Validation Confusion Matrix for the Hybrid Model (1,500 Images; True Class on Rows, Predicted Class on Columns). 1,491 of 1,500 Images (99.60%) Classified Correctly

True Class	Tomato___Bacterial_spot	299	1			
	Tomato___Late_blight	2	295	2	1	
	Tomato___Leaf_Mold			298	2	
	Tomato___Septoria_leaf_spot		1		299	
	Tomato___healthy					300
		Tomato___Bacterial_spot	Tomato___Late_blight	Tomato___Leaf_Mold	Tomato___Septoria_leaf_spot	Tomato___healthy

4.5 Comparative Discussion

Table 6 summarizes the three-way comparison directly. The hybrid model achieves the highest overall accuracy of the three architectures while also achieving the lowest total misclassification count, confirming that feature-level fusion — rather than merely averaging two already-correlated decision boundaries — allows the new classification head to exploit complementary information present in the concatenated ResNet50/GoogLeNet feature space. This comes at a clear and quantifiable cost: the hybrid network's parameter count (≈ 31 million) exceeds ResNet50's alone (25.6 million) and is roughly 5.3 times GoogLeNet's (5.9 million), and its training time (84 min 16 s) is more than 5 times ResNet50's and over 12 times GoogLeNet's, since every training image must be propagated through both full backbones on every iteration.

Table 6: Summary: Accuracy Gain vs. Computational Cost

Metric	ResNet50	GoogLeNet	Hybrid
Validation accuracy	99.13%	98.80%	99.60%
Misclassifications (/1500)	13	18	9
Late blight recall	97.67%	97.00%	98.33%
Learnable parameters	25.6 M	5.9 M	≈ 31.0 M
Training time	15 min 31 s	6 min 43 s	84 min 16 s

5. Discussion

The results support the central hypothesis motivating this extension: because the baseline ResNet50 and GoogLeNet models were shown to fail on the same visually ambiguous cases, a feature-level fusion of

their internal representations — rather than a simple decision-level combination — was expected to offer genuine, not merely averaged, discriminative improvement. The observed reduction in total misclassifications (13 and 18 for the two baselines, down to 9 for the hybrid) and the specific improvement in late-blight recall (the weakest class in both baselines) are consistent with this hypothesis, though the margin is modest in absolute terms — a difference of four to nine correctly classified images out of 1,500 — and should be interpreted with the same caution the baseline study applied to its own single-run, single-split evaluation.

The accuracy gain is not free. The hybrid model's parameter count and training time place it firmly outside the resource-constrained, on-device deployment scenario for which GoogLeNet was favoured in the baseline study, and even outside the profile of ResNet50 alone. Practically, the hybrid architecture is best suited to centralized or cloud-based diagnostic services where peak accuracy is paramount and training/inference compute is not the binding constraint — a more extreme version of the same accuracy-versus-efficiency trade-off the baseline study identified between ResNet50 and GoogLeNet individually.

Several limitations apply equally to this extension as to the baseline study. The evaluation reported here reflects a single training run on a single stratified data split; reporting variance across multiple random seeds and cross-validation folds — as recommended as future work in the baseline study — would strengthen confidence in the precise accuracy figures reported, particularly given the modest absolute margin between the hybrid model and ResNet50 alone. The expanded augmentation protocol (Table 2) is a deliberate methodological deviation from the baseline's minimal augmentation and was introduced specifically to regularize the hybrid head's additional capacity; it was not applied to the baseline models, so part of the hybrid model's accuracy improvement cannot be attributed to fusion alone and may be partly attributable to the richer augmentation. Isolating these two effects — fusion architecture versus augmentation protocol — by retraining a fusion-free baseline under the same expanded augmentation would be a natural next step. Finally, as in the baseline study, the underlying dataset is derived from a curated source with relatively uniform, plain backgrounds; performance under fully unconstrained field conditions remains untested and is likely to be lower than reported here for all three architectures.

6. Conclusion

This paper presented a hybrid ResNet50–GoogLeNet architecture for early detection of tomato leaf fungal and bacterial diseases, constructed by removing both backbones' original ImageNet classification heads, concatenating their pooled feature outputs (2048-d and 1024-d, respectively) into a single 3072-d representation, and training a new fully connected → batch normalization → ReLU → dropout → softmax classification head jointly on the fused features. The assembled network — 321 layers, approximately 31 million learnable parameters, verified with zero warnings and zero errors — was trained for 30 epochs under an expanded five-parameter geometric augmentation protocol, reaching a best recorded validation accuracy of 99.60% and a confusion-matrix-verified accuracy of 99.6% (1,491 of 1,500 validation images correctly classified). This exceeds both constituent baseline architectures (99.13% for ResNet50, 98.80% for GoogLeNet) and reduces total misclassifications from 13 and 18, respectively, to 9, with the clearest class-level improvement occurring in late blight — the class both baseline models found most difficult, and the specific confusion cluster this fusion architecture was designed to address. This improvement comes at the cost of a larger combined parameter footprint and substantially longer training time than either baseline alone, positioning the hybrid architecture as best

suited to accuracy-critical, compute-permissive deployment scenarios rather than the resource-constrained, on-device screening use case for which GoogLeNet was favoured individually.

Future work should isolate the relative contributions of the fusion architecture and the expanded augmentation protocol to the observed accuracy gain, report variance across multiple random seeds and cross-validation folds, and evaluate the hybrid model's generalization to field-collected imagery with natural, uncontrolled backgrounds, consistent with the future-work directions identified in the baseline study.

7. Acknowledgment

The authors thank the Department of Computer Science & Engineering, Sage University Indore, for providing the computational resources and academic guidance that supported this work.

8. References

1. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
2. C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1–9.
3. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, 2016.
4. D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv:1511.08060*, 2015.
5. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.
6. M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.
7. M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "Tomato leaf disease detection using convolutional neural network," *Procedia Comput. Sci.*, vol. 167, pp. 293–301, 2020.
8. C. Shorten and T. M. Khoshgofaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, 2019.
9. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent. (ICLR)*, 2015.
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 25, 2012, pp. 1097–1105.
11. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2009, pp. 248–255.
12. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, art. 3289801, 2016.
13. 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," *Sensors*, vol. 17, no. 9, p. 2022, 2017.
14. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019.

15. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 1251–1258.
16. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 4700–4708.
17. J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” Biosyst. Eng., vol. 172, pp. 84–91, 2018.
18. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.
19. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Int. Conf. Learn. Represent. (ICLR), 2015.
20. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. 32nd Int. Conf. Mach. Learn. (ICML), 2015, pp. 448–456.
21. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Int. Conf. Learn. Represent. (ICLR), 2018.
22. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 6023–6032.
23. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 618–626.