

A Lightweight RepViT-M1 Framework for Tea Leaf Disease Classification: Benchmarking Against VGG16 for Efficient Edge Deployment

Sarada R¹, Juvvala Bala Ambedkar², Praveen Kumar Nalli³, Chaliki Archana⁴, Dhanalakota Sameera⁵, Soma Raju Kalidindi⁶

^{1,3,6}Department of Artificial Intelligence, Vishnu Womens University, Bhimavaram, India, rsarada.ai@svecw.edu.in, praveenkumar.nalli@svecw.edu.in, ksrajuai@svecw.edu.in

²Department of Computer Science & Engineering, Vishnu Institute of Technology, Bhimavaram, India, balaambedkar.j@vishnu.edu.in

⁴Department of Information Technology, Aditya University, Surampalem, Kakinada, India, archanach@adityauniversity.in

⁵Department of Computer Science & Engineering (AI & ML), Vidya Jyothi Institute of Technology, Aziz Nagar, Hyderabad, sameeradhanalakota@gmail.com

Corresponding author: praveenkumar.nalli@svecw.edu.in

Abstract: Exact and timely identification of tea (*Camellia sinensis*) leaf diseases is necessary for controlling yield and quality. However, field deployment is limited due to the large computational requirement of the standard deep convolutional networks. This paper presents a lightweight, deployment-oriented framework for classifying eight tea-leaf conditions from an imbalanced 885-image dataset and benchmarks it against a widely used VGG16 transfer-learning baseline. The baseline replaces the VGG16 classifier head and fine-tunes it on a class-balanced set, reaching 95.63% validation accuracy with a macro-averaged F1 of 0.957 but requiring roughly 134 million parameters. The proposed pipeline builds on a RepViT-M1 backbone (a mobile-oriented, parameterizable vision transformer) with a compact 128-dimensional bottleneck head, focal loss with label smoothing, differential learning rates, cosine annealing, and test-time augmentation. On a held-out test partition it attains 99.92% accuracy, a macro F1 of 0.999 and a one-vs-rest macro-ROC-AUC of 1.000 while using only about 4.8 million parameters, a reduction of roughly 28x relative to VGG16. An ablation over fine-tuning depth and a two-stage healthy-versus-diseased cascade are described as design options, and the trained model is exported to the ONNX interchange format (18 MB) to support cross-runtime inference on edge hardware. We report the results transparently, including a candid discussion of the class-balancing and evaluation protocol so that the efficiency gains are interpreted responsibly. The findings indicate that compact reparameterizable backbones can match far larger networks on tea-disease recognition, offering a practical route to on-device agricultural diagnostics.

Keywords: Tea leaf disease; RepViT; transfer learning; VGG16; focal loss; lightweight deep learning; ONNX; precision agriculture; image classification; edge deployment

1. Introduction

Tea is among the most widely consumed beverages in the world, and the health of the tea plant (*Camellia sinensis*) has direct economic consequences for millions of smallholder farmers. Foliar diseases such as algal leaf spot, anthracnose, bird-eye spot, brown blight, gray blight, red leaf spot and white spot degrade both yield and the sensory quality of the final product. In most producing regions, diagnosis still relies on visual inspection by trained agronomists, which is slow, subjective, and difficult to scale across large plantations. Automated image-based recognition therefore offers an attractive path to earlier intervention and more consistent decisions.

Deep convolutional neural networks have transformed plant-disease recognition over the past decade, with landmark studies demonstrating that transfer learning from large natural-image corpora such as ImageNet can be



adapted to leaf imagery with high accuracy [15][16][17]. However, many of the architectures that anchor these studies, including very deep networks such as VGG16 [1], carry more than one hundred million parameters. Such models are accurate but expensive to store and run, which is a poor match for the edge devices and mobile handsets that are realistically available at the point of care in a plantation. There is thus a persistent tension between accuracy and deployability.

Recent developments in effective design of architecture fill up this gap. The concept of structural reparameterization was introduced in the paper by RepVGG [21]. It allows a multi-branch network to train and then those branches are folded to a single convolution at inference time. So, the accuracy of a complex network can be achieved with a simple one. The mobile vision-transformers family finds a new member through RepViT [2]. By inspecting lightweight convolutional design from the vision transformer perspective, it delivers compact yet accurate backbones. The attributes of RepViT a very useful technology for farm diagnostics of small hardware.

This paper contributes in three ways. To begin with, we develop a robust VGG16 transfer-learning baseline on an eight-class tea-leaf dataset and we report its accuracy together with its full parameter cost. Next, we introduce and assess a RepViT-M1 pipeline that utilizes a compact bottleneck head, focal loss integrated with label smoothing, varying learning rates, and test-time augmentation. Interestingly, it either matches or surpasses the baseline while utilizing a mere one twenty-eighth of the parameters. For a third contribution, we show a practical way to deploy our model, including exporting it to the ONNX interchange format. We also candidly discuss the class-balancing and evaluation protocol to ensure responsible interpretation of our efficiency gains.

2. Related Work

2.1 Deep learning for plant and leaf disease recognition

Image-based recognition of plant-disease grew up rapidly after the release of large annotated corpora and after a suitable convolutional network [3] became available. According to Mohanty et al., transfer learning is now used as the default strategy as deep networks trained on the PlantVillage collection yielded very high accuracy over dozens of crop-disease pairs. Ferentinos [16] expanded the focus to a broader range of species and field conditions, while Too et al. [17] systematically compared fine-tuning strategies over several convolutional backbones, highlighting the relationship between depth, accuracy, and training cost. The papers collectively show that pre-trained features generalize well to leaf images, yet at the same time inherit the parameter load of the backbones. Class imbalance is a common challenge that leads to the resampling and augmentation strategies investigated here [24][25].

2.2 Tea leaf disease classification

The work done with tea gained traction alongside. Hu and colleagues proposed an enhanced deep convolutional network for the identification of tea-leaf diseases in small samples. To remedy insufficient data, Hu et al. [18] employed support-vector refinement and augmentation. Chen and other researchers [19] proposed a convolutional model for the images of tea-leaf disease and reported a good performance in the field. In tea specific systems, as in the general plant-disease literature, common healthy leaves and diseases dominate the collection while rarer conditions are under-represented, so balancing the training distribution is a key area of concern. Through the years, the push for efficiency had produced a rich array of compact networks.

2.3 Efficient architectures and reparameterization

MobileNet and MobileNetV2 made depthwise-separable convolutions popular for mobile inference, while EfficientNet introduced principled compound scaling. The residual networks [5] and their descendants remain the strong backbone, and on the transformer side the Vision Transformer [4] demonstrated that attention-based models can rival convolutional networks at scale, and the Swin Transformer [8] and MobileViT [9] adapted this toward hierarchical and mobile-friendly designs. According to RepVGG, structural reparameterization decouples training-time and inference-time topology. Using such ideas, RepViT builds smaller, accurate mobile vision transformers. We chose RepViT-M1 as our backbone which comes from this lineage.

3. Materials and Methods

3.1 Dataset

The Research Work is based on a tea-leaf sickness image database organized into eight class folders consisting of 885 RGB photographs taken in natural conditions. The dataset shows significant class imbalance, with the most common condition featuring 143 images, and the healthy class having only 74 images. Upon inspection, duplicate rows and missing paths were not identified. Table 1 depicts breakdown of the raw dataset, per class.

S.No	Class	Number of images
1	red leaf spot	143
2	white spot	142
3	algal leaf	113
4	brown blight	113
5	bird eye spot	100
6	Anthracnose	100
7	gray light	100
8	healthy	74
Total		885

Table 1. Class distribution of the raw tea-leaf dataset (8 classes, 885 images).

3.2 Preprocessing and class balancing

All the images were decoded to three-channel RGB and normalized by the standard ImageNet channel statistics (mean 0.485/0.456/0.406, std 0.229/0.224/0.225). Two balancing strategies were deployed in the two pipelines. To ensure a balanced data set, the VGG16 baseline applied random resampling with replacement, such that every class matched the count of the majority class. This was then split into training and validation partitions. The RepViT-M1 pipeline made use of an augmentation-based scheme. Every class was raised to a fixed per-class cap here by writing distinct augmented copies to disk. The geometric and photometric transformations (small-angle rotation, where no augmented row's pixels were same, of up to seven degrees, horizontal and vertical flips, bounded translation and scaling, random resized cropping, and brightness/contrast jitter) were used to achieve this. The rotation was purposely kept small because several classes are defined by orientation. In Section 4.5, the interaction between balancing and the train/validation/test split will be addressed, along with the interpretation of the reported accuracies.

3.3 Baseline architecture: VGG16 transfer learning

The baseline used a VGG16 network pre-trained on ImageNet [1][13]. The convolutional feature extractor was frozen and the final fully connected layer was replaced by a linear layer matching the number of classes. Inputs were resized to 224x224 and augmented at training time with horizontal flips and small rotations. The model was optimized with Adam (learning rate 1×10^{-4} , weight decay 1×10^{-4}) using cross-entropy loss, a plateau-based learning-rate scheduler, and gradient-norm clipping, for up to forty epochs with best-checkpoint selection on validation loss. This configuration follows common transfer-learning practice for leaf-disease recognition.

3.4 Proposed architecture: RepViT-M1 with a bottleneck head

The proposed model wraps a RepViT-M1 backbone obtained from the timm library [2][26] in a lightweight classification head. The backbone returns a 384-dimensional pooled feature vector, which passes through batch normalization [6], a 128-dimensional bottleneck with ReLU activation and dropout, and a final linear layer. All four backbone stages were unfrozen so that the ImageNet features could adapt to leaf imagery, while the head and the

backbone were assigned separate learning rates. Inputs were processed at a higher resolution of 484x484, chosen so that fine, hairline defects survive downsampling. The full model contains approximately 4.77 million parameters, of which about 4.75 million are trainable.

3.5 Loss function and training strategy

To address residual class difficulty the RepViT-M1 pipeline used a focal loss [10] with focusing parameter $\gamma = 2.0$ and label smoothing [27] of 0.05, which down-weights easy examples and discourages over-confident predictions. Optimization used AdamW [11] with decoupled weight decay, differential learning rates (1×10^{-3} for the head and 5×10^{-5} for the backbone), and a cosine-annealing schedule [12]. Mixed-precision training, gradient-norm clipping, and early stopping on the macro-F1 metric were applied. At inference, test-time augmentation averaged the softmax outputs over the original image and its horizontal and vertical flips. The training data pipeline additionally applied on-the-fly affine jitter, color jitter and random erasing.

3.6 Ablation and two-stage cascade

Two further design elements were formulated on top of the main classifier. An ablation study progressively unfreezes backbone stages (frozen backbone, last stage only, last two stages) to attribute the accuracy contribution of deeper fine-tuning, using a shorter training budget so the effect of transfer depth can be isolated. A two-stage cascade was also formulated: a binary gate first separates healthy from diseased leaves, after which a dedicated classifier resolves the specific disease only on leaves the gate flags as diseased. Because the healthy class is by far the most separable, decomposing the problem in this way is a natural route to reducing confusion among the visually similar disease classes, and both sub-models train on class-balanced data.

3.7 Deployment via ONNX export

To decouple deployment from the training framework, the trained RepViT-M1 model was exported to the Open Neural Network Exchange (ONNX) format [28] at opset 17 with a dynamic batch axis, producing an 18 MB portable artifact that can be executed by ONNX Runtime, TensorRT or OpenVINO. The required preprocessing (resize, scale to the unit interval, channel normalization and NCHW rearrangement) is documented alongside the export so that inference can be reproduced outside Python.

3.8 Evaluation metrics

Models were evaluated with overall accuracy, balanced accuracy, and precision, recall and F1-score reported under both macro and weighted averaging. Class separability was additionally summarized by the one-vs-rest ROC-AUC, aggregated as a macro and a support-weighted mean. Macro-averaged metrics were used for model selection because they weight every class equally and are therefore sensitive to errors on the rarer conditions.

3.9 Experimental setup

Experiments were implemented in PyTorch [14] with the timm [26] library and executed on a single CUDA-capable GPU with automatic mixed precision, torch version 2.5.1. A fixed random seed of 42 was used throughout for reproducibility. The RepViT-M1 pipeline used a batch size of 32; training halted early at epoch 12 under the macro-F1 early-stopping criterion.

4. Results and Discussion

4.1 VGG16 baseline

After balancing, the VGG16 baseline was trained on 915 images and validated on 229. The best checkpoint reached a validation accuracy of 95.63% with a validation loss of 0.080, a macro-averaged F1 of 0.957 and a weighted F1 of 0.957. Per-class scores are reported in Table 2. The healthy class was classified perfectly, and gray blight, bird-eye spot and red leaf spot all exceeded an F1 of 0.96; the weakest class was brown blight at an F1 of 0.915. The results

show that VGG16 transfer learning can be a good baseline model; however, the model has a total of roughly 134.29 million parameters, with 119.58 million parameters trainable in it.

Class No	Precision	Recall	F1-score
1	1	0.931	0.964
2	0.964	0.931	0.947
3	0.9	0.964	0.931
4	0.9	0.931	0.915
5	0.966	0.966	0.966
6	0.931	0.964	0.947
7	1	0.966	0.982
8	1	1	1
Macro avg	0.958	0.957	0.957
Weighted avg	0.958	0.956	0.957

Table 2. VGG16 per-class validation results (accuracy = 95.63%, $n = 229$).

4.2 RepViT-M1 pipeline

The RepViT-M1 classifier converged quickly, reaching a validation macro-F1 above 0.99 within the first two epochs. After 12 epochs, the classifier achieved a best validation macro-F1 of 0.9996, and early stopping occurred at this point. On the test data that the model did not train on, it achieved a 99.92% accuracy and a 99.92% balanced accuracy. Moreover, it achieved a macro precision, recall and F1 of 0.999 while having a one-vs-rest macro ROC-AUC of 1.000. All this was achieved while only utilising approximately 4.75 million parameters. Table three compares the two backbones with each other. The key outcome is that the RepViT-M1 model matches or beats the VGG16 baseline and does so with a model counting that is almost 28 times less than the baseline. This is the efficiency argument of the paper.

Model	Total params	Trainable params	Accuracy
VGG16 (baseline)	134.29 M	119.58 M	95.63%
RepViT-M1 (proposed)	4.77 M	4.75 M	99.92%

Table 3. Backbone comparison. VGG16 accuracy is on the validation split; RepViT-M1 accuracy is on a held-out test split (see Section 4.5).

4.3 Overall discussion

When summarised, the activities available public acknowledgements are patent specifications, which assist small companies in assessing the needs of patent protection, granting existing patent holders time to change or maintain their inventions, and providing claim language to recognised patent owners. The design choices of the RepViT-M1 pipeline are also known to help on imbalanced imagery: focal loss with label smoothing, differential learning rates, higher input resolution for fine defects, and test-time augmentation. The designs of ablation and cascade provide more levers for practitioners who need to give up a little accuracy for less latency or a clearer separation of the dominant healthy class.

4.4 Methodological considerations and threats to validity

The obligation of scientific honesty requires a clear statement about how balancing interacts with data splitting, as this bears directly on how the nearly-perfect figures of RepViT-M1 should be read. The VGG16 baseline balanced

the dataset by resampling with replacement before the train/validation split, which means exact duplicate copies of an image can appear in both partitions; this tends to inflate validation accuracy and is a known pitfall in imbalanced-image studies [25]. The RepViT-M1 pipeline avoids exact-duplicate leakage by writing pixel-distinct augmented copies, but because those copies are generated before the split, augmented derivatives of the same source photograph can still be distributed across the training, validation and test partitions. Given the small number of unique source images (885), this shared-source effect is the most plausible explanation for the extremely high test scores, and the reported 99.92% should therefore be read as an upper bound obtained under this protocol rather than as a guarantee of field performance. The fair and stable conclusion of the study is the relative one: RepViT-M1 is at least as accurate as VGG16 at roughly one twenty-eighth of the parameters, which holds regardless of the absolute inflation, since both models are evaluated on comparably balanced data.

Another consideration is external validity. Due to the limited capture conditions and small size of the dataset, the absolute accuracies may not transfer to new plantations, cameras, seasons or lighting. Another reason is that the two pipelines differ in more than just their backbone, with differences in input resolution, loss, augmentation and the presence of a test split, so we view Table 3 as a system-level comparison, rather than a controlled single-variable experiment.

4.5 Recommendations for a leakage-free protocol

To get the unbiased absolute estimates, we must split the 885 unique source images first in any future runs, then the upsample or augment the training partition only; the validation and test partitions must be wholly composed of untouched originals. Having group-aware splitting that keeps all augmented derivatives of a source image in a single fold, k-fold cross-validation along with an external test set would provide further support. RepViT-M1's performance gain is likely to persist under this protocol even if accuracy levels degrade substantially.

5. Conclusion and Future Work

A lightweight and deployment-oriented tea-leaf disease classification framework is presented in this paper which is benchmarked against a VGG16 transfer-learning baseline. The eight-class dataset used in this paper is imbalanced. The proposed RepViT-M1 pipeline that features a compact bottleneck head, focal loss, different learning rates, cosine annealing, and test-time augmentation matched or more often exceeded the baseline while using around 28x fewer parameters. It also exported cleanly to an 18 MB ONNX artifact for cross-runtime inference. We reported these results transparently, including an exposition of how class balancing and data splitting inflate absolute accuracy, so that the durable contribution, the efficiency of compact reparameterizable backbones, is responsively interpreted.

Future work will include a re-evaluation of the pipeline under a strict group aware, leakage free protocol with k-fold cross-validation and external test set; extending and diversifying the dataset to improve external validity; adding interpretability through gradient-based saliency e.g. Grad-CAM [20] to build agronomist trust; benchmarking on-device latency and energy on representative edge hardware to quantify the practical payoff of the parameter reduction.

6. Acknowledgment

The authors express their heartfelt gratitude to the Department of Artificial Intelligence, Shri Vishnu Engineering College for Women for their support. The department also offered the GPU server available in the HPC Lab of the department.

REFERENCES

1. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
2. A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding, "RepViT: Revisiting mobile CNN from ViT perspective," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2024, pp. 15909-15920.
3. 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.

4. A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.
5. 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.
6. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. Int. Conf. Mach. Learn. (ICML), 2015, pp. 448-456.
7. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105-6114.
8. Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 10012-10022.
9. S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," in Proc. Int. Conf. Learn. Represent. (ICLR), 2022.
10. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980-2988.
11. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2019.
12. I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.
13. 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.
14. A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 32, 2019, pp. 8026-8037.
15. S. P. Mohanty, D. P. Hughes, and M. Salathe, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, art. 1419, 2016.
16. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311-318, 2018.
17. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272-279, 2019.
18. G. Hu, X. Yang, Y. Zhang, and M. Wan, "Identification of tea leaf diseases by using an improved deep convolutional neural network," *Sustainable Computing: Informatics and Systems*, vol. 24, art. 100353, 2019.
19. J. Chen, Q. Liu, and L. Gao, "Visual tea leaf disease recognition using a convolutional neural network model," *Symmetry*, vol. 11, no. 3, art. 343, 2019.
20. 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.
21. X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, pp. 13733-13742.
22. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
23. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510-4520.
24. C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, art. 60, 2019.
25. M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249-259, 2018.
26. R. Wightman, "PyTorch Image Models (timm)," GitHub repository, 2019. [Online]. Available: <https://github.com/huggingface/pytorch-image-models>
27. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2818-2826.
28. J. Bai, F. Lu, K. Zhang et al., "ONNX: Open Neural Network Exchange," GitHub repository, 2019. [Online]. Available: <https://github.com/onnx/onnx>