

Adaptive Gated DenseNet121 for Paddy Disease Classification: Performance Gains and Limits of Sparsity-Based Pruning

Mujahid Shahid¹

Kwame Nkrumah University of Science and Technology

Richard Owusu-Ansah¹

rowusuansah39@st.knust.edu.gh

Kwame Nkrumah University of Science and Technology

Samuel Appiah

Kwame Nkrumah University of Science and Technology

Article

Keywords: Paddy disease classification, DenseNet121, adaptive gating, sparse pruning, transfer learning, computer vision

Posted Date: September 8th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10755499/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

DenseNet121 is effective for rice disease recognition, but its dense feature reuse keeps many connections active during inference. We evaluated whether learnable layer gates could improve paddy disease classification and expose removable DenseNet connections. Using the Kaggle Paddy Disease Classification dataset, we trained an ImageNet-pretrained DenseNet121 baseline and an Adaptive Gated DenseNet121 on 8,325 training images and validated on 2,082 images across ten classes. The baseline achieved 96.78% accuracy and 96.64% macro F1-score. The adaptive model achieved 97.21% accuracy and 96.89% macro F1-score, with 97.16% macro precision and 96.65% macro recall. Pruning at threshold 0.825 reduced parameters from 6.96 million to 6.84 million while retaining 96.39% F1-score, but slightly higher thresholds caused severe collapse. Adaptive gating therefore improved classification and enabled limited pruning, but did not yet provide robust mobile compression.

Introduction

Rice is a staple crop for much of the world, and disease or pest damage can reduce yield, income and food security. Visual diagnosis by experts is useful but difficult to scale, especially where disease monitoring must occur repeatedly across many farms. Deep convolutional neural networks (CNNs) provide an automated route for leaf-image diagnosis because they learn discriminative symptom features directly from annotated images [1, 2].

DenseNet121 is particularly relevant to rice disease classification. Densely connected convolutional networks concatenate feature maps from earlier layers into later layers, improving feature reuse and gradient flow [3]. Ismail et al. used DenseNet121 for rice disease classification and reported 97.9% accuracy on a seven-disease rice dataset containing 11,467 images after augmentation [1]. That work establishes DenseNet121 as a strong reference architecture for rice disease recognition.

However, DenseNet's strength also raises a deployment question. Dense connectivity keeps many feature-producing layers active during inference. For mobile and field applications, it is valuable to know whether all dense-layer outputs are equally necessary, or whether the model can learn to suppress weak feature contributions. Sparse training and pruning methods, including Network Slimming, have shown that learned scaling factors can sometimes identify removable structures [4]. We therefore modified DenseNet121 by inserting learnable scalar gates into its dense-layer outputs.

This study asks two questions. First, can adaptive gating improve paddy disease classification relative to a standard DenseNet121 baseline? Second, can the learned gates support post-training pruning without destroying predictive performance? We answer these questions using saved Kaggle GPU experiments on a ten-class paddy disease dataset.

Related Works

Deep learning-based methods for identification of plant diseases were made possible by Mohanty et al. (2016) as they demonstrated that a convolutional neural network can be trained to identify both the type of crop and state of disease present in an image of a single leaf with very high accuracy. After this study, researchers transitioned from using a controlled dataset with highly uniform conditions and instead began working on developing field-image datasets which would allow them to work towards creating benchmark models specific to different crops. The Paddy Doctor project identified the lack of annotated images of real-world paddy disease conditions as a challenge and released an annotated collection of paddy leaf images acquired in the field. [2].

DenseNet (Huang et al.) is a deep network that connects every previous layer with all of its future layers using a process called "feature concatenation." Because this enables the use of previously used features from earlier in the network as well as reducing "vanishing gradient" problems, it explains why the pre-trained DenseNet-121 model continues to be a powerful baseline model in most studies regarding classification of rice diseases. The two other models mentioned here are examples of similar approaches based upon performance improvement; EfficientNet is an example of how the depth, width, and resolution of a deep neural network can be improved simultaneously via "compound scaling," and MobileNet-V2 is an example of how a network can be optimised for mobile deployment by utilising "inverted residual" and "depth-wise separable convolution. [3].

Sparse and pruned networks are typically motivated by deployment constraints. For instance, previous work demonstrated that network slimming could be accomplished through sparsity regularisation of scaling factors and pruning of channels based on their learned scale. This paper continues in a similar spirit with an adaptive DenseNet approach; however, instead of applying sparse or gate regularisation to each layer's output, learnable gates are applied directly to the output of each dense layer within a DenseNet architecture. This is significant because, as opposed to training a reduced version of the network, the proposed method requests that the topological structure of the original DenseNet provide information about which feature contributions can be eliminated.

Numerous studies have emphasized on applying DenseNet models for plant disease classification. For instance, Saputra et al employed a deep learning algorithm using convolutional Neural network algorithm with DenseNet121, DenseNet169 and DenseNet 201 architectures for detecting diseases in rice leaves. From their experiment, they achieved an accuracy of 94% using DenseNet 121 and an accuracy of 89% using DenseNet169 and an accuracy of 92% using DenseNet201. From their experiment, DenseNet121 had the best accuracy of 94% [10]. Another study explored a hybrid DenseNet architectures for rice leaf disease detection by developing a series of an enhanced DenseNet models which integrate architectural enhancements and make use of learning parameters to get a high result for rice leaf classification. They implemented baseline and improved versions of DenseNet121, DenseNet169 and DenseNet201. They integrated Squeeze and Excitation (SE) blocks to improve channel-wise feature calibration. Their proposed hybrid framework combines Squeeze and Excitation improved DenseNet architectures with keras Tuner- based hyperparameter optimization which makes their model different from other DenseNet-based rice leaf detection [11]. Additionally, a DenseNet121 model was proposed by the

authors for paddy leaf disease classification. From their findings, the use of DenseNet121 optimizes the use of resource and also reduces crop losses and from their experiment analysis it shows that the DenseNet121 model achieved a higher precision, recall, f-measure and accuracy values than other models like VGG-16 and ResNet-50 [12].

Yang et al. developed a new detection transformer to detect disease and classify rice leaf. In developing this model, a dense higher-level composition feature pyramid network (DHLC-FPN) was employed and incorporated with the DETR (detection transformer) algorithm that could accurately detect three diseases on rice leaves including sheath blight, rice blast, and flax spot. The experimental results indicated that the proposed model significantly improved the ability to extract features from the feature pyramid network and resolved the problem resulting in lost high-level image and low-level semantic information in convolutional networks [13].

Archama et al. developed a methodology in order to improve the computational efficiency and classification performance for identification of rice plant diseases. The methodology utilised a modified version of the k-means segmentation algorithm to segment out areas where lesions may occur from the remainder of the rice plant image. In addition, the methodology utilised a new intensity-based colour feature extraction (NIBCFE) to extract colour features from the images. Furthermore, the methodology utilised both grey-level co-occurrence matrices (GLCM) and bit-patterns as additional features. Based upon the experimental results, it was demonstrated that the NSVM-BPN model produced a higher classification accuracy than all three other methodologies tested, which were compared against each other based upon their performance [14].

Chawal and Panday used Twin Support Vector Machine (TSVM) technique for rice plant disease classification. From their paper, they used a prototype system to classify rice leaf Blast disease and Bacterial blight using K-means clustering for segmentation. They tested on rice leaves to detect if they were affected by the blast disease or bacterial blight. From their experiment analysis, with a total of 146 number of rice leaves images, 100 was affected by blast, 36 was affected by blight and 10 was healthy. From their experiment analysis, their prototype system had classification accuracy more than 95% by the trained TSVM [15].

Aggarwal presented a pre-trained Deep Neural-Based feature selection supported by machine learning for rice leaf disease classification. From their proposed method, features were previously extracted by the use of 32 pre-trained models and they classified their images of rice leaf diseases such as bacterial blight, blast and brown spot with machine learning and ensemble learning classification. From their experiment, their proposed model identified 90.91% accuracy [16].

Alsakar et al discussed a method for detecting and classifying rice plant leaf diseases by fusing different features which includes color texture with Local Binary Pattern (LBP) and color features with color correlogram model (CC). Their proposed model go through five stages. First, the acquisition of input images for RGB. Second, there is an application of image preprocessing of data augmentation to handle imbalance issues. Third, there is an extraction of feature stage for the extraction of color features using

(CC) which uses multi-level multi-channel local information. Fourth, there is a feature fusion stage which provides complementary information by concatenating two feature types. Finally, there is an application of rice image classification using a one-against- all support vector machine (SVM) [17].

Hasan et al presented an Artificial Intelligence (AI) and an effective computer vision techniques to detect rice diseases. They employed SVM classifier to combine with a deep CNN. Their proposed model was able to identify and classify rice diseases of nine different types and an accuracy of 97% was achieved [18].

Raigonda and Anjali explored a hybrid deep learning algorithm CNN and VGG-16 for identification and early diagnosis of rice leaf disease. In their experiment, they focused on three rice leaf diseases such as brown spot, leaf blast and leaf blight with a total dataset of 1800 images and four classes were used. Their experiment results using CNN and Transfer learning had an accuracy of 95.6% [19]”

Results

Adaptive gating modifies DenseNet121 while preserving transfer learning

The baseline used DenseNet121 with ImageNet pre-training and a ten-class classifier head. The proposed model preserved the DenseNet121 backbone but replaced each dense layer by a gated dense layer. For dense layer i , the feature map $f_i(x)$ was multiplied by $\text{sigmoid}(g_i)$, where g_i is a trainable scalar logit. The total objective combined cross-entropy with a gate sparsity term, $L = L_{CE} + \lambda \sum_i |\text{sigmoid}(g_i)|$. DenseNet121 contains 58 dense-layer outputs, so this modification added only 58 trainable gate parameters.

Dataset scope exceeded the seven-class base-paper setting

The base paper used seven rice disease classes and 11,467 total images after augmentation [1]. Our experiments used ten paddy disease or condition classes from the Kaggle Paddy Disease Classification data [5], yielding 10,407 labelled images. A stratified split produced 8,325 training images and 2,082 validation images. The wider class set includes disease, pest damage and normal leaf categories, making the task broader than a disease-only setting.

Table 1. Dataset scope compared with the DenseNet121 base paper.

Study	Classes	Total labelled images	Split
Ismail et al. [1]	7 rice disease classes	11,467 after augmentation	8,026 train / 3,441 test
This study	10 paddy disease/condition classes	10,407 labelled images	8,325 train / 2,082 validation

Adaptive gated DenseNet121 outperformed the saved baseline

The DenseNet121 baseline reached its best macro F1-score at epoch 16, with 96.78% accuracy, 97.35% macro precision, 96.03% macro recall and 96.64% macro F1-score. The Adaptive Gated DenseNet121 reached its best checkpoint at epoch 16, with 97.21% accuracy, 97.16% macro precision, 96.65% macro recall and 96.89% macro F1-score. The adaptive model therefore improved accuracy by 0.43 percentage points and macro F1-score by 0.25 percentage points.

Table 2. Main validation performance.

Model	Best epoch	Accuracy	Precision	Recall	F1-score	Validation loss
DenseNet121 baseline	16	96.78%	97.35%	96.03%	96.64%	0.1364
Adaptive Gated DenseNet121	16	97.21%	97.16%	96.65%	96.89%	0.1441

Confusion matrices showed strong but imperfect separation

Both models produced strongly diagonal confusion matrices. The adaptive model correctly classified 346 of 348 blast validation images, 287 of 288 dead-heart images and 216 of 218 tungro images. Remaining errors concentrated in visually similar or symptom-overlapping categories, including brown spot, downy mildew, hispa and normal leaves.

Learned gates decreased but remained tightly clustered

Gate values decreased during training, confirming that the sparsity penalty updated the adaptive parameters. At epoch 1, the mean gate value was 0.8784. At the best checkpoint epoch, the gate distribution was narrow, with values near 0.827. By epoch 22, the mean value had decreased to 0.8021. This indicates soft feature reweighting rather than clearly binary connection selection.

Pruning was active but fragile

Thresholds up to 0.823 did not reduce parameters. At threshold 0.825, pruning reduced parameters from 6,964,164 to 6,839,683 and model size from 26.89 MB to 26.41 MB while retaining 96.39% macro F1-score. However, threshold 0.827 reduced the model to 5,111,024 parameters but collapsed macro F1-score to 5.05%. Thus, the useful pruning window was extremely narrow.

Table 3. Adaptive model pruning threshold response.

Threshold	Parameters after pruning	Size (MB)	Accuracy	F1-score
0.750	6,964,164	26.89	97.21%	96.89%
0.800	6,964,164	26.89	97.21%	96.89%
0.820	6,964,164	26.89	97.21%	96.89%
0.823	6,964,164	26.89	97.21%	96.89%
0.825	6,839,683	26.41	96.88%	96.39%
0.827	5,111,024	19.73	17.20%	5.05%
0.829	1,149,389	4.43	16.95%	2.90%
0.831	858,827	3.31	16.95%	2.90%
0.833	713,546	2.74	16.95%	2.90%

Discussion

The results support the first hypothesis: adaptive gating can improve DenseNet121-based paddy disease classification. The improvement was modest but meaningful because it was achieved by adding only 58 scalar gate parameters to the baseline architecture. This suggests that DenseNet121 feature reuse can benefit from learnable modulation rather than fixed use of every dense-layer output.

The results only partially support the pruning hypothesis. The adaptive model showed active pruning at threshold 0.825, but the compression was small and the next higher thresholds caused severe loss of classification performance. The learned gates therefore formed a fragile pruning boundary rather than a robust sparse structure. This behaviour is consistent with a model that uses gates as soft attention-like weights, not as hard architectural selectors.

Compared to Ismail et al. [1], DenseNet121 is used in the more general case of ten-class paddy disease and conditions. The test results are very similar to those of Ismail et al. (97.9%) for this multi-ten class problem but were obtained from a larger, separate public dataset, with the test/validation sets partitioned differently. Because of these differences, caution must be exercised when directly comparing the numeric results. This study has been developed as a study of architecture using fixed training

parameters and saved Kaggle GPU simulations. These results suggest that adaptive gating can enhance the ability of DenseNet121 to classify diseases/conditions and can lead to some degree of pruning at optimally selected thresholds. Due to the sensitivity of the process of training large-scale deep neural networks to initialisation, orderings of the data used in training, and augmentations applied during training, the results presented here should be taken as indicative of single-run performance rather than representative of the distribution of such performance over repeated runs. Stronger statistical validation would involve repeating the same experiments (i.e., same models trained on the same data) but across multiple random seed states and reporting the mean $\pm$ std. of each of accuracy, precision, recall, and F-score. Future experiments could also examine if the results translate to actual mobile efficiencies by examining post-pruning fine-tune latencies, latency-based pruning criteria, and actual device runtime inference times.

Methods

Dataset

Experiments used the Kaggle Paddy Disease Classification dataset [5], derived from the Paddy Doctor dataset [6]. The ten classes were bacterial leaf blight, bacterial leaf streak, bacterial panicle blight, blast, brown spot, dead heart, downy mildew, hispa, normal and tungro. The labelled data were split into 8,325 training images and 2,082 validation images using a stratified 80/20 split. The hidden-label Kaggle test images were not used for reported metrics.

Image preprocessing

Images were resized to 224 x 224 pixels. Training images were augmented using random horizontal flipping and random rotation. All images were converted to tensors and normalized using ImageNet channel means and standard deviations.

DenseNet121 baseline

The baseline used torchvision DenseNet121 initialized with ImageNet weights. The original 1,000-class classifier was replaced with a ten-class linear classifier. The model was trained using cross-entropy loss and AdamW optimization.

Adaptive gated DenseNet121

Each DenseNet dense-layer output was multiplied by a learnable scalar gate. The gate was defined as $\text{sigmoid}(g_i)$, where g_i is a trainable logit initialized with $\text{gate_init} = 2.0$. For feature output $f_i(x)$, the

gated output was $y_i = \text{sigmoid}(g_i) f_i(x)$. The loss was $L = L_{\text{CE}} + \lambda \sum_i |\text{sigmoid}(g_i)|$, with $\lambda = 5 \times 10^{-4}$ in the Adaptive 2 experiment.

Training and evaluation

Training used PyTorch and torchvision on Kaggle GPU. Models were trained with batch size 32, learning rate 1×10^{-4} and AdamW weight decay 1×10^{-4} . Validation was performed after every epoch, and the best checkpoint was selected using macro F1-score. Metrics included accuracy, macro precision, macro recall, macro F1-score, validation loss, confusion matrix, parameter count and model size.

Pruning

After training, the best adaptive checkpoint was evaluated under several gate thresholds. When a dense-layer gate was below the threshold, the corresponding layer was replaced by a zero-output layer with matching channel dimensions. This preserved tensor shapes while removing stored parameters associated with pruned dense-layer modules.

Statistical analysis

No inferential statistical test was applied because the saved experiments consist of one run per model. Results are reported as descriptive validation metrics over $n = 2,082$ validation images. Future repeated-seed experiments should report mean, standard deviation and appropriate paired or non-parametric comparisons.

Ethics statement

This study used publicly available plant image data and did not involve human participants, human tissue, live vertebrates or higher invertebrates.

Use of artificial intelligence assistance

A large language model was used to assist with code organization and formatting.

Data availability

The dataset presented and used in this study are the publicly available Paddy Disease Classification dataset on Kaggle: <https://www.kaggle.com/competitions/paddy-disease-classification/data>.

Code availability

The PyTorch implementation, Kaggle notebooks and report-generation scripts are available in the project workspace on Github [20].

Declarations

Acknowledgements

The authors acknowledge the Kaggle Paddy Disease Classification competition and the Paddy Doctor dataset contributors for providing public benchmark data.

Author contributions

M.S. developed the model, conducted the experiments, and evaluated the model performance. R.O. conducted the literature review and contributed to identifying the research gap. S.A. designed the study and guided the overall research framework. All authors reviewed and revised the manuscript and approved the final version.

Funding

No funding was received for this manuscript.

Additional Information

Competing interests

The authors declare no competing interests.

References

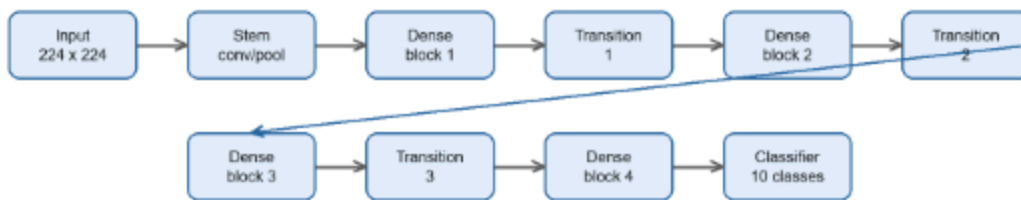
1. Ismail, A., Hamdy, W., Ibrahim, A. H. & Awad, W. A. Classification of rice plant diseases using efficient DenseNet121. *Sci. Rep.* 16, 7482 (2026).
2. Mohanty, S. P., Hughes, D. P. & Salathe, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1419 (2016).
3. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. Preprint at <https://arxiv.org/abs/1608.06993> (2016).

4. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S. & Zhang, C. Learning efficient convolutional networks through network slimming. Preprint at <https://arxiv.org/abs/1708.06519> (2017).
5. Kaggle. Paddy disease classification. <https://www.kaggle.com/competitions/paddy-disease-classification/data> (2022).
6. Petchiammal, A., Briskline Kiruba, S., Murugan, D. & Pandarasamy, A. Paddy Doctor: a visual image dataset for automated paddy disease classification and benchmarking. Preprint at <https://arxiv.org/abs/2205.11108> (2022).
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. MobileNetV2: inverted residuals and linear bottlenecks. Preprint at <https://arxiv.org/abs/1801.04381> (2018).
8. Tan, M. & Le, Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. Preprint at <https://arxiv.org/abs/1905.11946> (2019).
9. Pacal, I. et al. A systematic review of deep learning techniques for plant diseases. *Artif. Intell. Rev.* 57, 304 (2024).
10. Saputra, A. D., Hindarto, D., & Santoso, H. Disease classification on rice leaves using DenseNet121, DenseNet169, DenseNet201. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 7(1), 48–55 (2023). <https://doi.org/10.33395/sinkron.v8i1.11906> (2023).
11. Singh, J. P., Ghosh, D., Kumar, A., Bilgaiyan, S., Kumar, R., & Singh, J. Hybrid DenseNet architectures and KerasTuner-based optimization for rice leaf disease detection. *IEEE Access*, 14, 26845–26868. <https://doi.org/10.1109/ACCESS.2026.3664467> (2026).
12. Sahasranamam, V., Ramesh, T., Karthikkumar, A., & Muthumanickam, D. AI-driven paddy leaf disease classification and prediction using DenseNet-121. In P. Ahlawat et al. (Eds.), *Proceedings of the 6th International Conference on Deep Learning, Artificial Intelligence and Robotics (ICDLAIR 2024)* (Advances in Intelligent Systems Research, Vol. 193). Atlantis Press. https://doi.org/10.2991/978-94-6463-740-3_24 (2025).
13. Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., & Liu, N. Disease detection and identification of rice leaf based on improved detection transformer. *Agriculture*, 14(1), Article 93. <https://doi.org/10.3390/agriculture14010093> (2024).
14. Archana, K. S., Srinivasan, S., Prasanna Bharathi, S., Balamurugan, R., Prabakar, T. N., & Sagai Francis Britto, A. A novel method to improve computational and classification performance of rice plant disease identification. *The Journal of Supercomputing*, 78, 1085–1112. <https://doi.org/10.1007/s11227-021-04245-x> (2022).
15. Chawal, B., & Panday, S. P. Rice plant disease detection using Twin Support Vector Machine (TSVM). *Journal of Science and Engineering (JScE)*, 7, 61–68. (2019).
16. Aggarwal, M., Khullar, V., Goyal, N., Singh, A., Tolba, A., Thompson, E. B., & Kumar, S. Pre-trained deep neural network-based features selection supported machine learning for rice leaf disease classification. *Agriculture*, 13(5), 936. <https://doi.org/10.3390/agriculture13050936> (2023).
17. Alsakar, Y. M., Sakr, N. A., & Elmogy, M. An enhanced classification system of various rice plant diseases based on multi-level handcrafted feature extraction technique. *Scientific Reports*, 14,

18. Hasan, M. J., Mahbub, S., Alom, M. S., & Nasim, M. A. Rice disease identification and classification by integrating support vector machine with deep convolutional neural network. Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT 2019). IEEE. <https://doi.org/10.1109/ICASERT.2019.8934564> (2019).
19. Raigonda, M. R., & Yedve, A. Identification and classification of rice leaf disease using hybrid deep learning. Journal of Scientific Research and Technology, 3(6), 1–7. <https://www.jsrtjournal.com/>
20. Github repository. <https://github.com/lordbinary11/densenet-research.git> (2026).

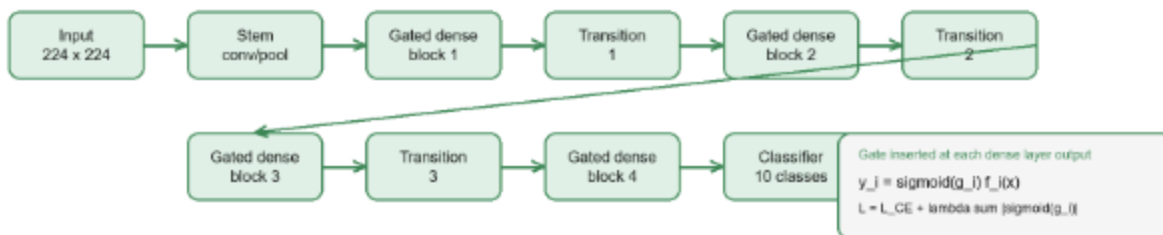
Figures

a DenseNet121 baseline



Dense-layer output: concatenate all previous features; all dense connections remain active.

b Proposed adaptive gated DenseNet121

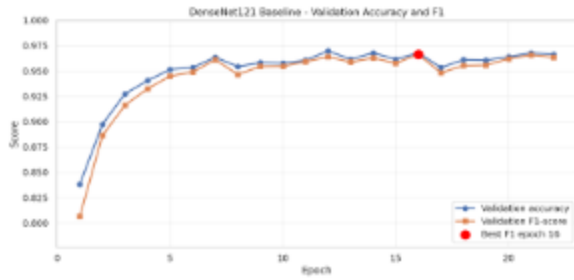


The adaptive model adds 56 scalar gate logits to DenseNet121 while preserving the classifier head.

Figure 1

DenseNet121 baseline and proposed adaptive gated DenseNet121 architectures. a, DenseNet121 uses a convolutional stem, four dense blocks separated by transition layers and a classifier head. b, the proposed model inserts scalar sigmoid gates at dense-layer outputs and adds a sparsity penalty to the training objective.

a



b

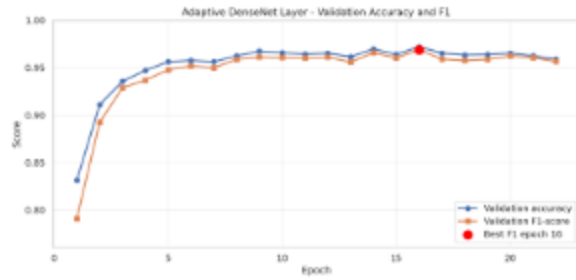
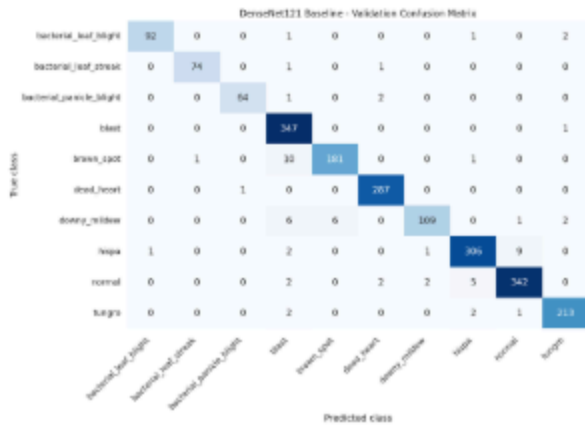


Figure 2

Validation accuracy and macro F1-score during training. a, DenseNet121 baseline training curve. b, Adaptive Gated DenseNet121 training curve. Red markers indicate the best macro F1-score checkpoint.

a



b

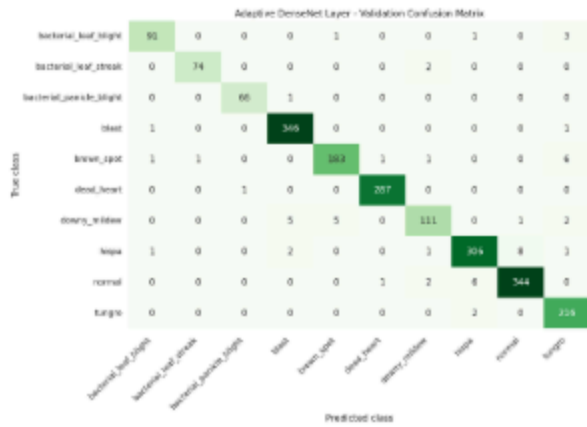


Figure 3

Validation confusion matrices. a, DenseNet121 baseline confusion matrix. b, Adaptive Gated DenseNet121 confusion matrix. Rows indicate true classes and columns indicate predicted classes.

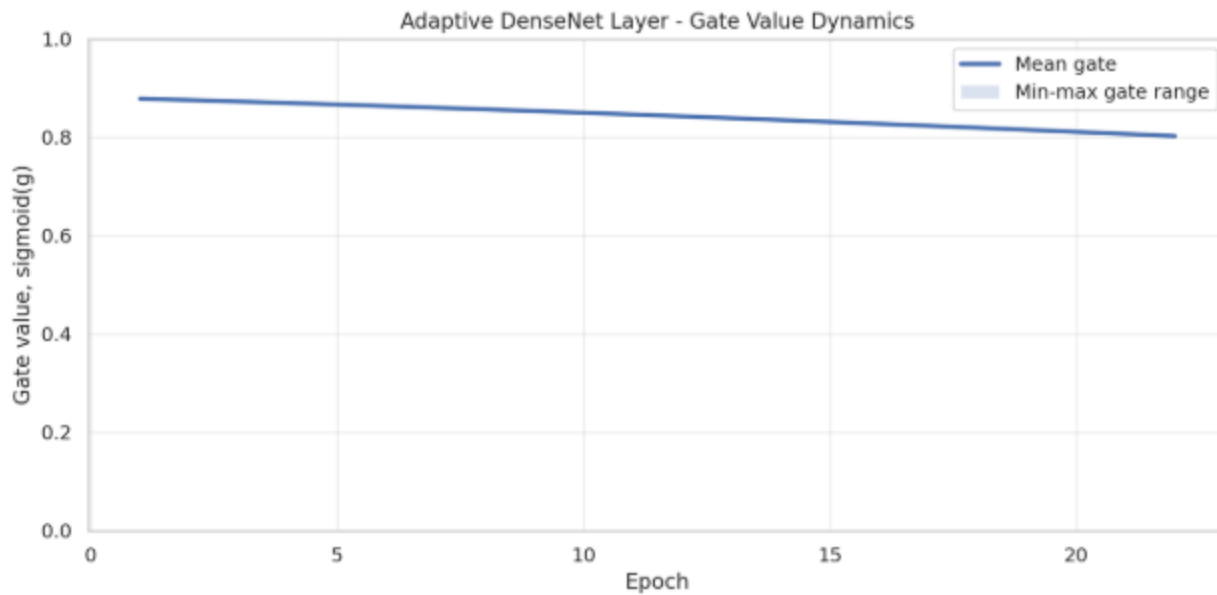


Figure 4

Gate-value dynamics in the adaptive model. The line shows mean sigmoid gate value and the shaded band shows the min-max gate range across 58 dense-layer gates.

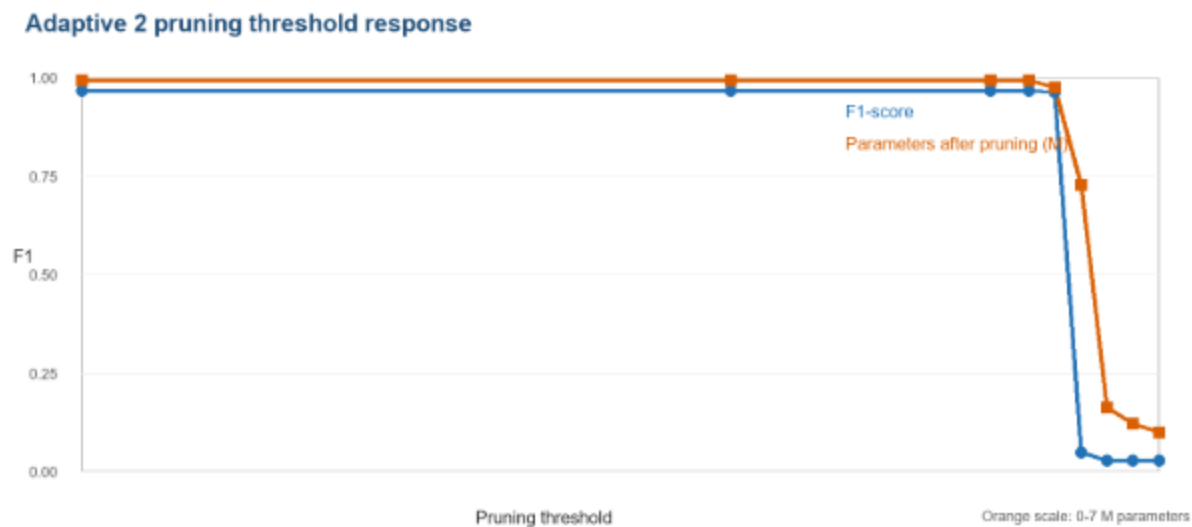


Figure 5

Pruning threshold response. Macro F1-score and remaining parameter count are shown across gate-pruning thresholds. Threshold 0.825 gave modest compression with limited F1 loss, whereas stronger thresholds collapsed performance.