BioMedInformatics

MDPI

Article

DFU-MambaKAN: A Lightweight Hybrid Mamba-KAN Architecture for Diabetic Foot Ulcer Screening and Severity Grading

Md Nafis Azad Nobel [1], Sazid Rahman Kazi [2], Tajul Islam Rafi [2], Mainuddin Adel Rafi [2], Umer Aqeel [2], Md Ranu Hossen [2], Md Al Ridwan [2] and Roise Uddin [2,*]

[1] Department of Computer Science, Pacific State University, Los Angeles, CA 90010, USA; p25576@psuca.edu
[2] Department of Information Technology, Pacific State University, Los Angeles, CA 90010, USA; p25481@psuca.edu (S.R.K.); p25840@psuca.edu (T.I.R.); p25768@psuca.edu (M.A.R.); p26244@psuca.edu (U.A.); p26050@psuca.edu (M.R.H.); p25845@psuca.edu (M.A.R.)
* Correspondence: ruddin@psuca.edu

Abstract

Diabetic foot ulcers (DFUs) are a major cause of lower-extremity amputation, and image-based decision support may assist screening and preliminary triage where specialist access is limited. We propose DFU-MambaKAN, a hybrid architecture combining a pure-PyTorch selective state-space (Mamba) block with a Gaussian radial-basis-function Kolmogorov–Arnold Network (RBF-KAN) feed-forward layer. The study evaluates two distinct tasks: (1) binary normal-versus-ulcer screening and (2) a dataset-specific four-class Wagner-Meggitt severity-grading task. DFU-MambaKAN contains 1.062 million parameters and was compared with ResNet50, EfficientNet-B0, MobileNetV3-Small, and ViT-Tiny under the manuscript's common downstream training protocol. The reported values are point estimates from one deterministic 70/15/15 split generated with seed 42; no confidence intervals, repeated-seed averages, or inferential significance tests were obtained. On binary screening, DFU-MambaKAN achieved 97.17% accuracy, macro-F1 0.970, and AUC 0.994. On four-class grading, it achieved 67.75% accuracy, macro-F1 0.677, and AUC 0.895, whereas the baselines achieved 98.14–99.00% accuracy. This large gap means that the present evidence does not establish competitiveness for multiclass severity grading. Possible contributors include training duration and convergence, dataset size and class definitions, label or image-quality uncertainty, hyperparameter selection, global token mixing, and architecture–task mismatch; extended learning-curve analysis and multi-seed evaluation are needed to test these explanations. An automated technical audit identified 342 exact duplicates among 1055 class-folder images in Dataset A (32.4%); no deduplicated-versus-nondeduplicated performance comparison was performed, so this finding is reported as a methodological caution rather than proof of accuracy inflation. The model's 1.062M parameter count supports parameter-efficient storage, but its measured single-image GPU latency was 90.54–93.14 ms, substantially higher than the baselines. Accordingly, the current evidence supports further investigation for batched or queued screening more strongly than immediate real-time mobile deployment. External clinical validation, multi-seed analysis, statistical testing, convergence studies, and component ablations remain necessary.

Keywords: diabetic foot ulcer; Mamba; state space models; Kolmogorov–Arnold Networks; lightweight deep learning; medical image classification; Wagner classification; explainable AI

Academic Editors: Kwang-Sig Lee and Hyuntae Park

Received: 2 July 2026
Revised: 29 July 2026
Accepted: 3 August 2026
Published: 6 August 2026

1. Introduction

1.1. Background and Real-World Relevance

Diabetes mellitus affected an estimated 463 million adults worldwide in 2019, with projections rising to 700 million by 2045 [1]. Among its most debilitating complications is the diabetic foot ulcer (DFU): a person with diabetes carries an estimated 19–34% lifetime risk of developing a foot ulcer [2], and DFUs are a primary precursor to non-traumatic lower-limb amputation. Once an ulcer occurs, recurrence rates exceed 60% within three to five years, and five-year mortality following ulceration has been reported in the range of 50–70% [2]. DFU is therefore not a localized cosmetic problem; it is a marker of severe systemic risk.

Clinically, the Wagner classification system comprises six grades (Grades 0–5) based primarily on wound depth and gangrene [3]. Binary image screening and severity grading serve different purposes. A binary result may support early referral or preliminary triage by flagging images that are consistent with an ulcer, whereas severity grading may assist prioritization, documentation consistency, longitudinal monitoring, and decision support after an ulcer has been identified. Neither task is intended to replace specialist assessment, and final diagnosis and treatment decisions require evaluation by qualified clinical personnel. The four-class experiment in this study uses only the Grade 1–4 folders supplied by Dataset B and therefore represents a dataset-specific subset rather than the complete six-grade Wagner system.

Most existing deep-learning approaches to this problem prioritize classification accuracy using comparatively large backbones, while deployment also depends on parameter count, storage, memory, throughput, and latency. This paper therefore examines accuracy–efficiency trade-offs across a coarse screening task and a finer-grained grading task. Because no smartphone or edge-device measurements were conducted, deployment interpretations are restricted to the measured NVIDIA T4 GPU, single-CPU-thread, and batched-GPU settings.

1.2. Contributions

This paper makes the following contributions:

1. We propose DFU-MambaKAN, a 1.062 M-parameter hybrid design that combines a pure-PyTorch selective state-space block, a depthwise-convolution branch, and Gaussian RBF-KAN feed-forward layers. The combined design is proposed rather than claimed to be an empirically optimal component combination.
2. We evaluate two independent tasks with different clinical decision-support roles: Stage 1 binary normal-versus-ulcer screening and Stage 2 dataset-specific four-class Grade 1–4 severity grading.
3. We report an automated technical data-hygiene audit, including class-folder filtering and exact-hash deduplication. The finding of 342 duplicates among 1055 Dataset A class-folder images (32.4%) is presented as a methodological caution because no direct deduplicated-versus-nondeduplicated accuracy comparison was performed.
4. We compare accuracy, macro-F1, AUC, parameter count, an implementation-dependent FLOP estimate, and measured GPU/CPU latency with ResNet50 [4], EfficientNet-B0 [5], MobileNetV3-Small [6], and ViT-Tiny [7] under the manuscript's common downstream protocol.
5. We explicitly report the weak four-class result (67.75% accuracy) relative to the 98.14–99.00% baseline range and discuss multiple possible explanations without treating under-convergence as established.

6. We provide approximate token-based Grad-CAM [8] visualizations and state their limitations: they are qualitative saliency maps, not validated lesion localizations or evidence of causal clinical reasoning.

2. Related Work

2.1. Classical and Lightweight CNN Approaches to DFU Analysis

Goyal et al. introduced DFUNet [9], one of the earliest CNN-based DFU classifiers, achieving an AUC of 0.961 on a curated set of 397 images, and later extended this line of work toward real-time, mobile-deployable detection [10]. The same research group subsequently released the DFUC2021 dataset and benchmark [11,12], the largest publicly available DFU dataset with explicit infection and ischaemia pathology labels (15,683 patches across four classes: control, infection, ischaemia, and both). Notably, even strong CNN backbones (VGG16, ResNet101, InceptionV3, DenseNet121, EfficientNet) evaluated on this four-class clinical sub-task achieved only macro-average F1 around 0.55 [12], a result we return to in Section 4 when discussing our own severity-grading findings. Yap et al. [13] further provide a comprehensive evaluation of deep learning across the DFU detection literature, reinforcing that fine-grained, multi-class clinical sub-tasks consistently underperform relative to coarser binary detection tasks across multiple independent studies.

2.2. Efficient and Transformer-Based Vision Backbones

The efficiency-accuracy tradeoff central to this work builds on three well-established lightweight CNN/Transformer families. ResNet [4] introduced residual connections enabling stable training of much deeper networks. EfficientNet [5] systematically balances network depth, width, and input resolution via compound scaling. MobileNetV3 [6] combines hardware-aware neural architecture search with squeeze-and-excitation blocks, explicitly targeting mobile CPU latency. The Vision Transformer (ViT) [7], building on the original Transformer architecture [14], demonstrated that pure self-attention over image patches can match or exceed convolutional backbones, while the Swin Transformer [15] reintroduced hierarchical, windowed locality to reduce attention's quadratic cost. All four of ResNet50, EfficientNet-B0, MobileNetV3-Small, and ViT-Tiny serve as baselines in this work precisely because they represent the standard accuracy-efficiency spectrum against which any new lightweight proposal must be measured.

2.3. State Space Models and Mamba

Structured state space models address the quadratic attention cost of Transformers with linear-time sequence modeling. Mamba [16] introduced an input-dependent ("selective") discretization of a continuous-time state-space recurrence, achieving Transformer-competitive quality at linear computational cost. VMamba [17] adapted this mechanism to 2D vision via a cross-scan module, and U-Mamba [18] demonstrated Mamba's effectiveness as a long-range-dependency module for biomedical image segmentation. To the best of our knowledge, at the time of writing no published work applies a Mamba-family architecture specifically to diabetic foot ulcer image classification, a gap this paper directly addresses.

2.4. Kolmogorov–Arnold Networks

Kolmogorov–Arnold Networks (KAN) [19] replace the fixed linear-weight-plus-nonlinearity structure of a standard multilayer perceptron with learnable univariate functions on each network edge, motivated by the Kolmogorov–Arnold representation theorem. U-KAN [20] demonstrated that KAN layers can serve as a strong, parameter-efficient backbone component for medical image segmentation and generation, while Cheon [21] explored KAN's efficacy specifically in vision classification tasks. As with

Mamba, no prior published work has, to our knowledge, combined KAN layers with a DFU classification pipeline.

2.5. Explainability for Non-Convolutional Backbones

Grad-CAM [8] remains the most widely used post hoc explanation technique for CNNs, weighting the final convolutional feature map by the gradient of the target class score. Because the proposed backbone produces a globally mixed token sequence rather than a conventional final convolutional feature map, we apply the gradient-weighting principle to the final 14×14 token grid (Section 4.5). The resulting overlays are interpreted only as approximate qualitative saliency visualizations.

2.6. Summary of Gaps and Motivation

Table 1 summarizes the literature reviewed above. The study is motivated by three gaps: limited evaluation of Mamba–KAN combinations for DFU classification, incomplete joint reporting of accuracy and measured efficiency, and limited reporting of technical data-hygiene checks for public dataset mirrors. The duplicate-image analysis is treated as a caution about split construction rather than as the primary architectural contribution of the work.

Table 1. Summary of reviewed approaches and identified gaps.

Ref.	Method	Dataset	Task	Key Result	Limitation
[9]	DFUNet (custom CNN)	397 images (custom)	Binary detection	AUC 0.961	Very small dataset; no efficiency reporting
[12]	VGG16, ResNet101, InceptionV3, DenseNet121, EfficientNet	DFUC2021 (15,683 patches)	4-class infection/ischaemia	Macro-F1 $\approx$ 0.55 (best: EfficientNet-B0)	Fine-grained multiclass task remains difficult even for strong CNNs
[10]	CNN, mobile-optimized	DFUC2020-derived	Binary detection and localization	Real-time mobile feasibility shown	Accuracy–efficiency trade-off not benchmarked against modern lightweight backbones
[4]	ResNet50	ImageNet (general)	Image classification	Strong general baseline	High parameter count (23.5 M+)
[5]	EfficientNet-B0	ImageNet (general)	Image classification	Strong accuracy/FLOPs trade-off	CNN-only; no global sequence modeling
[6]	MobileNetV3-Small	ImageNet (general)	Image classification	Strong mobile-latency trade-off	Purely convolutional; limited long-range context
[7]	ViT	ImageNet (general)	Image classification	Competitive with CNNs at scale	Quadratic attention cost
[17]	VMamba	ImageNet (general)	Image classification	Linear-time vision backbone	Not evaluated on a medical or DFU task
[20]	U-KAN	Medical segmentation datasets	Segmentation and generation	Strong parameter efficiency	Not evaluated on DFU; segmentation rather than classification
This work	DFU-MambaKAN (proposed Mamba + RBF-KAN hybrid)	Two public DFU datasets (713 and 10,050 deduplicated images)	Binary normal-versus-ulcer screening and dataset-specific four-class Grade 1–4 severity grading	Architecture and two-task evaluation protocol; quantitative results reported in Section 4	Single-split evidence; no ablation or external clinical validation

3. Materials and Methods

3.1. Dataset Description

This work uses two independent, publicly available datasets hosted on Kaggle and retrieved programmatically via the kagglehub API. Dataset A supports Stage 1 binary

normal-versus-ulcer screening, whereas Dataset B supports Stage 2 dataset-specific four-class Grade 1–4 severity grading. Figure 1 summarizes the two-task evaluation pipeline from data acquisition and cleaning to task-specific reporting.

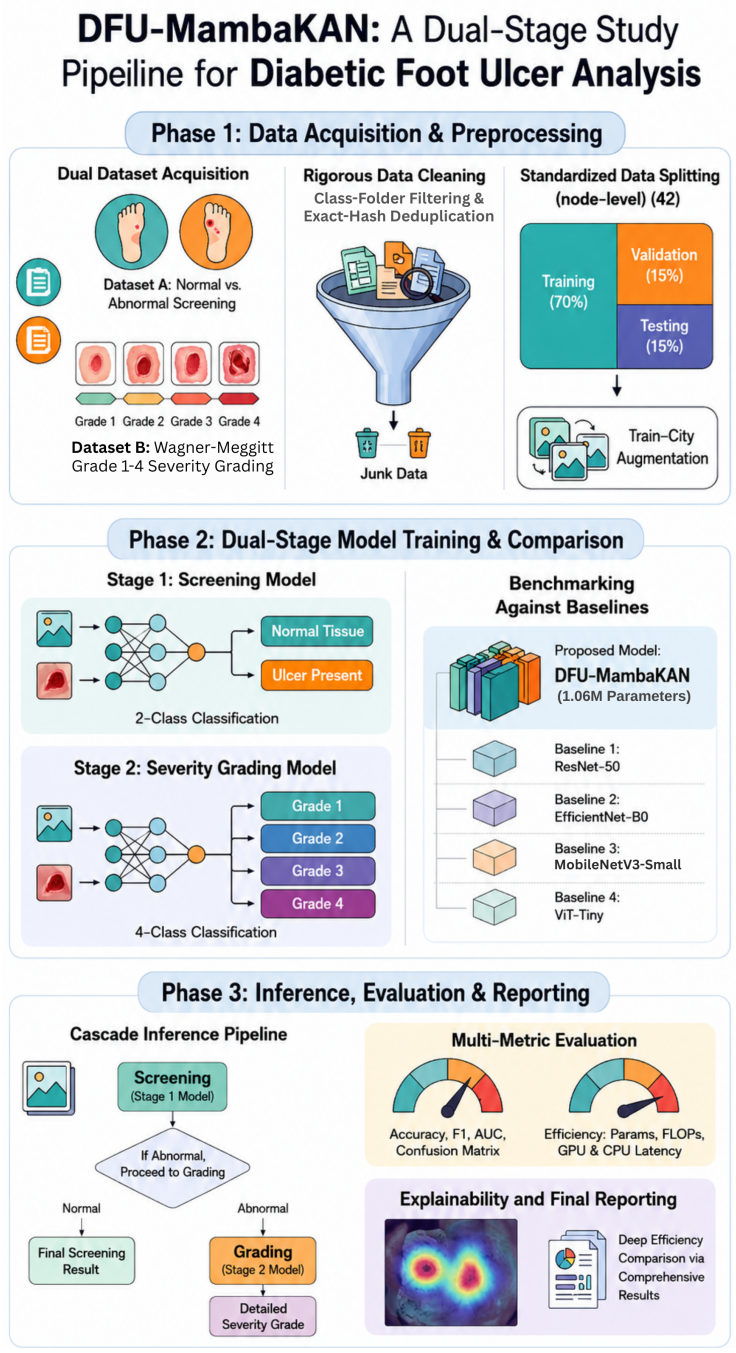

Figure 1. Overview of the study pipeline: Acquisition and cleaning of two independent datasets, separate model training and baseline benchmarking for Stage 1 binary normal-versus-ulcer screening and Stage 2 dataset-specific four-class Grade 1–4 severity grading, followed by task-specific evaluation and explainability reporting.

Dataset A (laithjj/diabetic-foot-ulcer-dfu) supports Stage 1 binary normal-versus-ulcer screening with the classes Abnormal(Ulcer) and Normal(Healthy skin). The raw download contained 2662 files across eight folders. A compact allow-list retained only the two class folders and excluded six auxiliary organizational folders, yielding 1055 class-folder images. Exact MD5 deduplication removed 342 byte-identical files (32.4%), leaving 713 unique

images. The held-out test split contained 65 abnormal/ulcer images and 41 normal/healthy-skin images ($n = 106$). Class-specific training and validation counts were unavailable.

Dataset B (khalidsiddiqui2003/dfu-dataset-annotated-into-4-classes) supports Stage 2 dataset-specific four-class Grade 1–4 severity grading. The distributed dataset contained 10,062 images in four folders corresponding to Wagner-Meggitt Grades 1, 2, 3, and 4. Exact-hash deduplication removed 12 images (0.12%), leaving 10,050 unique images. The held-out test split contained 336 Grade 1, 376 Grade 2, 419 Grade 3, and 376 Grade 4 images ($n = 1507$), corresponding to approximately 22.3%, 25.0%, 27.8%, and 25.0% of that test split. These frequencies are relatively close. Class-specific training and validation counts were unavailable. Dataset B contains four folders corresponding to Wagner-Meggitt Grades 1–4. The experiment therefore evaluates a dataset-specific four-class subset of the six-grade Wagner system rather than the complete grading scale.

Dataset quality and audit scope. The automated technical audit covered folder structure, exact duplicate detection, and prevention of exact-hash split leakage. A systematic corrupted-file audit and perceptual near-duplicate analysis were not conducted. The audit was technical rather than clinical: wound-care specialists did not revalidate the image labels, ulcer severity grades, diagnostic validity, or clinical image quality. Label uncertainty and uncontrolled acquisition conditions may therefore contribute to error, particularly in Stage 2. The 32.4% duplicate rate in Dataset A is a methodological caution because exact or near-identical images distributed across splits can produce optimistic estimates. The duplicate analysis did not compare deduplicated and nondeduplicated model performance and therefore does not quantify the effect of deduplication on accuracy.

Figure 2 shows representative test-split samples from Dataset A. A limited visual inspection checked for obvious technical shortcuts such as rulers, color-calibration cards, repeated backgrounds, or watermarks. No obvious artifact was identified in the inspected samples, but this non-expert inspection cannot validate diagnoses, severity labels, lesion boundaries, or the absence of subtler confounding features.

Figure 2. Representative test-split samples, Dataset A: abnormal/ulcer class (**top row**) and normal/healthy-skin class (**bottom row**).

3.2. Data Preprocessing

Technical folder filtering and exact deduplication. Dataset A was restricted through an explicit allow-list to `Abnormal(Ulcer)` and `Normal(Healthy skin)`, which were remapped to `abnormal` and `normal`. This prevented the six auxiliary folders from being interpreted as classes. Each retained file was then hashed before splitting. For byte content b, $h = \mathrm{MD5}(b)$ was retained only when $h \notin H$, where H denotes the set of previously observed hashes; the new hash was then added to H. This procedure prevents byte-identical files from crossing splits, but it does not identify visually similar nonidentical images.

Train/validation/test split and reporting scope. Both datasets used a single deterministic 70/15/15 split. For N unique images, $n_{\text{test}} = \lfloor 0.15N \rfloor$, $n_{\text{val}} = \lfloor 0.15N \rfloor$, and $n_{\text{train}} = N - n_{\text{val}} - n_{\text{test}}$. A dedicated pseudo-random generator with seed 42 assigned indices, and the same split was used for all five architectures. The reported metrics are single-run point estimates for this split and seed. Cross-validation, confidence intervals, standard deviations, and repeated-seed averages were not obtained. Deterministic split indices do not guarantee identical training outcomes because weight initialization and data-loading behavior can remain stochastic; run-to-run variability was not quantified. The study-generated split indices, exact duplicate hashes, source code, package environment, and model configurations are available from the corresponding author on reasonable request, subject to the redistribution conditions of the source datasets.

Resizing and normalization. Each RGB image tensor $x \in \mathbb{R}^{3\times224\times224}$ was resized to 224×224 pixels. Channel-wise normalization produced x' according to

$$x' = \frac{x - \mu}{\sigma}, \quad \mu = [0.485, 0.456, 0.406],\ \sigma = [0.229, 0.224, 0.225], \tag{1}$$

where subtraction and division are applied independently to the red, green, and blue channels; μ and σ are the standard normalization values used with ImageNet-pretrained models. The four baseline networks used ImageNet-pretrained initialization.

Data augmentation. Augmentation was applied only to the training split and comprised random horizontal flipping and random rotation within $\pm15^\circ$. Brightness, contrast, and saturation were perturbed using a dimensionless relative augmentation factor of 0.2. The documented augmentation did not include hue perturbation. Validation and test images were not augmented.

Class weighting. The focal loss used inverse-frequency weights computed from the training subset only: $w_c = N_{\text{train}} / (C\, n_{c,\text{train}})$, where $n_{c,\text{train}}$ is the number of training images in class c, C is the number of classes, and $N_{\text{train}} = \sum_{c=1}^{C} n_{c,\text{train}}$. The realized class-specific training counts and weights were unavailable, and the test distribution was not used to infer them. The relatively close Stage 2 test proportions suggest that the benefit of weighting may be limited. An unweighted-loss comparison was not conducted, so the contribution of class weighting remains uncertain.

3.3. Model Architecture: DFU-MambaKAN

DFU-MambaKAN combines a convolutional patch-embedding stem, four hybrid blocks, and a linear classification head. Figure 3 illustrates the model data flow. Component ablation was not performed; consequently, the individual contributions and optimality of the stem, tokenization strategy, Mamba branch, RBF-KAN layers, and classification head remain undetermined.

Patch embedding. An input image $I \in \mathbb{R}^{3\times224\times224}$ passes through two convolutional layers with strides 2 and 8, giving an overall stride of 16. The output is a 14×14 spatial grid with $d = d_{\text{model}} = 96$ channels. Raster-order flattening forms the token matrix $X = [x_1, \ldots, x_L]^\top \in \mathbb{R}^{L\times d}$ with $L = 14 \cdot 14 = 196$ and $x_t \in \mathbb{R}^d$.

Hybrid block. Each of the four blocks first applies LayerNorm and sums a Minimal Mamba global-mixing branch with a depthwise-convolution local-mixing branch through a residual connection. A second LayerNorm precedes an FFN residual branch. Blocks 1 and 2 use a two-layer MLP FFN, whereas Blocks 3 and 4 use the RBF-KAN FFN. This allocation is an architectural design choice; without ablation, the study cannot establish that placing KAN only in the final two blocks is optimal.

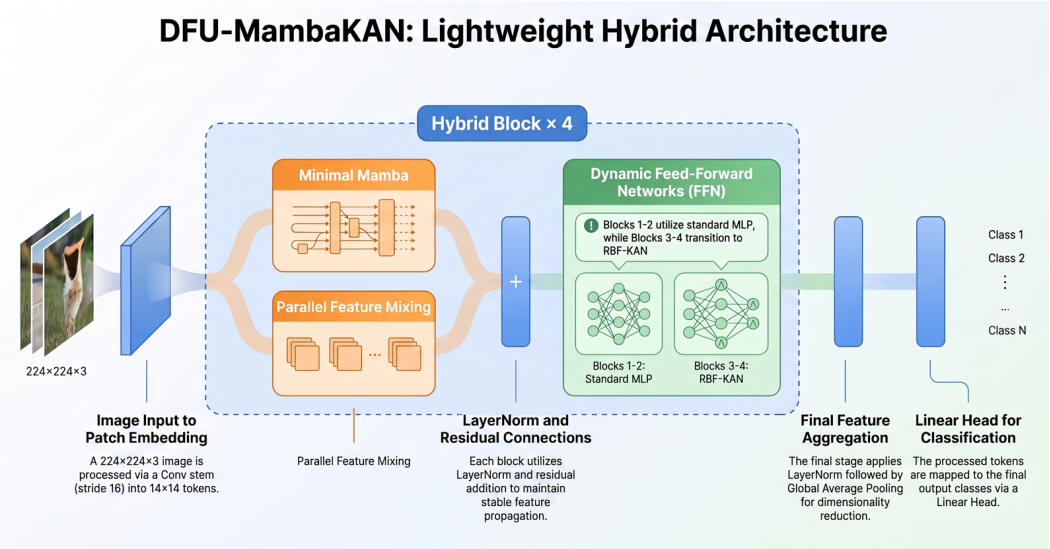

Figure 3. DFU-MambaKAN architecture: A convolutional patch-embedding stem feeds four stacked hybrid blocks (parallel Mamba + depthwise-conv mixing, followed by an MLP or RBF-KAN feed-forward network), ending in a pooled linear classification head.

Minimal Mamba block. Let $X \in \mathbb{R}^{L \times d}$ be the input token sequence. After causal convolution and learned projections, the block forms a projected input $x_t \in \mathbb{R}^{d_{\text{inner}}}$ and input-dependent quantities $\Delta_t \in \mathbb{R}^{d_{\text{inner}}}$ and $B_t, C_t \in \mathbb{R}^{d_{\text{inner}} \times n}$ at token position $t \in \{1, \ldots, L\}$. Here $n = 16$ is the state dimension and softplus is applied to Δ_t to keep its entries positive. The hidden state $h_t \in \mathbb{R}^{d_{\text{inner}} \times n}$ evolves according to

$$\bar{A}_t = \exp(\Delta_t \cdot A), \tag{2}$$

$$h_t = \bar{A}_t \odot h_{t-1} + \Delta_t \odot B_t \odot x_t, \tag{3}$$

$$y_t = C_t^{\top} h_t + D \odot x_t, \tag{4}$$

where $A \in \mathbb{R}^{d_{\text{inner}} \times n}$ is a learned state-transition parameter constrained through its parameterization to represent negative continuous-time dynamics, $\bar{A}_t \in \mathbb{R}^{d_{\text{inner}} \times n}$ is its input-dependent discretization, $D \in \mathbb{R}^{d_{\text{inner}}}$ is a learned skip coefficient, $y_t \in \mathbb{R}^{d_{\text{inner}}}$ is the output token, $\exp(\cdot)$ and softplus act elementwise, and $\odot$ denotes elementwise multiplication with broadcasting across the state dimension. The notation $C_t^{\top} h_t$ denotes contraction over the n state entries for each inner channel. The recurrence is evaluated for $L = 196$ positions in an explicit pure-PyTorch loop without a fused custom CUDA kernel.

RBF-KAN feed-forward layer. For an input vector $x \in \mathbb{R}^{d_{\text{in}}}$ and output index $o \in \{1, \ldots, d_{\text{out}}\}$, the implemented layer is

$$\text{KAN}(x)_o = \underbrace{\sum_{i=1}^{d_{\text{in}}} w_{io}\, x_i}_{\text{base linear term}} + \sum_{i=1}^{d_{\text{in}}} \sum_{k=1}^{K} c_{i,o,k} \exp\left(-\frac{(x_i - g_k)^2}{2\rho^2} \right), \tag{5}$$

where i indexes input features, o indexes outputs, w_{io} is the learned base-linear weight, K is the number of Gaussian basis functions per edge, $c_{i,o,k}$ is a learned basis coefficient, g_k is a fixed RBF centre on the bounded input grid, and $\rho > 0$ is the shared basis width. The exponential is applied to each scalar input–centre pair. This Gaussian-basis formulation follows the parameter-efficient per-edge nonlinearity principle of KANs [19]; its independent value in this architecture is not established without an ablation.

Classification head. Let $Z \in \mathbb{R}^{196 \times 96}$ denote the final token representation after the fourth hybrid block. LayerNorm is applied to Z, and mean pooling produces $\bar{z} = 196^{-1} \sum_{t=1}^{196} Z_t \in \mathbb{R}^{96}$. A learned linear map produces logits $s \in \mathbb{R}^{C}$, where $C = 2$ for

Stage 1 and $C = 4$ for Stage 2; class probabilities are obtained by softmax during evaluation and loss computation.

Architectural rationale and evidential scope. The Mamba branch is intended to provide global sequence mixing, the depthwise-convolution branch to preserve local spatial bias, and the RBF-KAN layers to provide learnable nonlinear edge functions. These are design motivations, not empirically isolated contributions. The complete model contains 1.062 M trainable parameters. Its reported 0.2246 GFLOPs is an implementation-dependent lower-bound estimate because several recurrence and elementwise operations are excluded (Section 4.3).

3.4. Hyperparameter Configuration

Table 2 states the downstream protocol applied to DFU-MambaKAN and to ResNet50, EfficientNet-B0, MobileNetV3-Small, and ViT-Tiny. The four baselines used ImageNet-pretrained initialization, whereas no external pretraining is reported for DFU-MambaKAN. The original large-scale ImageNet training recipes of MobileNetV3, EfficientNet, ResNet, and ViT were not reproduced. The comparison therefore evaluates these architectures under this manuscript's common DFU protocol rather than reproducing the optimization procedures of the original architecture papers. Differences in optimizer settings, scheduler design, augmentation, regularization, and training duration relative to those original recipes may affect the observed baseline and proposed-model performance.

Table 2. Unified downstream training protocol and model-specific initialization.

Setting	DFU-MambaKAN	All Four Baselines
Optimizer	AdamW [22]	AdamW [22]
Learning rate	3×10^{-4}	3×10^{-4}
Weight decay	0.05	0.05
LR schedule	Cosine annealing	Cosine annealing
Loss	Class-weighted focal loss, $\gamma = 2.0$	Class-weighted focal loss, $\gamma = 2.0$
Batch size, Stage 1/Stage 2	32/128	32/128
Training duration	25 epochs	25 epochs
Image resolution	224×224	224×224
Training augmentation	Horizontal flip; $\pm 15^{\circ}$ rotation; brightness/contrast/saturation factor 0.2	Same
Initialization	No external pretraining reported	ImageNet pretrained
Early stopping	Fixed 25-epoch schedule; no early-stopping criterion reported	Fixed 25-epoch schedule; no early-stopping criterion reported
Architecture-specific settings	Patch size 16; $d_{\text{model}} = 96$; state dimension 16; 4 blocks	Standard architecture definitions; classifier heads adapted to 2 or 4 classes
Data split/seed	70%/15%/15%; seed 42	Same split and seed

The fixed 25-epoch budget and cosine scheduler were used for downstream DFU training and fine-tuning. Original ImageNet training schedules are not directly equivalent to fine-tuning on these DFU datasets; nevertheless, training duration and scheduler selection can affect convergence. Extended Stage 2 training, learning-rate warm-up, and systematic hyperparameter search were not evaluated. The contribution of training duration to the 67.75% Stage 2 result therefore remains uncertain.

3.5. Evaluation Metrics

Let TP, TN, FP, FN denote true positives, true negatives, false positives, and false negatives respectively (extended to the multi-class case via a one-vs-rest decomposition for Stage 2).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Macro-F}_1 = \frac{1}{C} \sum_{c=1}^{C} F_{1,c} \tag{9}$$

Metric interpretation. Macro-averaged precision, recall, and F_1 weight each class equally and are reported alongside accuracy. The Stage 2 test supports (336, 376, 419, and 376) are relatively close, so macro averaging is retained for class-symmetric reporting rather than justified by a strong measured imbalance. AUC-ROC is computed one-versus-rest and macro-averaged for Stage 2; it summarizes ranking across thresholds but does not compensate for low hard-decision accuracy or establish clinical utility. Confusion matrices describe the observed error counts but do not by themselves validate clinical severity ordering or treatment relevance.

Training loss. The model is trained with a class-weighted focal loss [23]:

$$\text{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{10}$$

where $0 < p_t \leq 1$ is the predicted probability assigned to the true class, $\gamma = 2.0$ is the focusing parameter, and $\alpha_t = w_c$ is the training-set inverse-frequency weight for the true class defined in Section 3.2. When $\gamma = 0$ and $\alpha_t = 1$, the expression reduces to cross-entropy, $-\log(p_t)$. Focal loss and class weighting were applied in the reported protocol, but no plain-cross-entropy or unweighted ablation was conducted; their benefit is therefore not isolated empirically.

Statistical scope. The tables report single-run point estimates. Inferential comparisons, confidence intervals, standard deviations, paired-bootstrap intervals, McNemar tests, and repeated-seed comparisons were not obtained; observed between-model differences should therefore be interpreted descriptively. Future validation should retain paired test predictions for McNemar's test, use paired bootstrap resampling for metric differences, and report repeated-seed or cross-validation distributions.

Efficiency metrics. Parameter count is the number of trainable weights and is not identical to storage footprint or peak memory, neither of which was measured. FLOPs were estimated with `fvcore`. The reported 0.2246 GFLOPs is an implementation-dependent lower bound because the tool omits exponentiation, softplus, recurrent elementwise multiplication/addition, and related operations repeated at each of 196 token positions in four blocks. Latency was measured as mean forward-pass wall-clock time over 50 runs following 5 warm-up runs at batch size 1 on an NVIDIA T4 GPU and one CPU thread. Stage 1 GPU throughput was additionally measured at batch size 32. No smartphone, embedded accelerator, mobile CPU, power-consumption, peak-memory, or serialized-model-size measurements were performed.

4. Results and Discussion

4.1. Stage 1: Binary Normal-vs.-Ulcer Screening

Table 3 reports test-set performance for all five models on the held-out, deduplicated, leakage-free Stage 1 test split ($n = 106$; 65 abnormal, 41 normal).

Table 3. Stage 1 test results for binary normal-versus-ulcer screening ($n = 106$).

Model	Acc.	F1	AUC	Params (M)	GPU (ms)
DFU-MambaKAN	0.9717	0.9700	0.9944	1.06	90.54
ResNet50	0.9906	0.9901	0.9996	23.51	7.46
EfficientNet-B0	1.0000	1.0000	1.0000	4.01	9.12
MobileNetV3-Small	0.9811	0.9803	0.9977	1.52	7.12
ViT-Tiny	0.9906	0.9901	1.0000	5.52	6.28

For this single split and seed, DFU-MambaKAN was 2.83 accuracy points below EfficientNet-B0 and 0.94 points below MobileNetV3-Small. Its 1.062 M parameters are 0.458 M fewer than MobileNetV3-Small's 1.52 M, a reduction of approximately 30.1%, and 22.45 M fewer than the 23.51 M-parameter ResNet50. Per-class precision/recall/F_1 values were 0.97/0.98/0.98 for abnormal and 0.97/0.95/0.96 for normal. The AUC was 0.9944. These single-run point estimates do not establish equivalence, superiority, robustness, or clinical generalization relative to the baselines. The Stage 1 screening result is promising within the evaluated split.

Figure 4 shows 64 of 65 abnormal cases and 39 of 41 normal cases correctly classified, with 1 abnormal image predicted as normal and 2 normal images predicted as abnormal. Binary screening could support preliminary referral or triage, but these counts do not establish patient-level safety, clinical sensitivity under deployment conditions, or suitability for treatment decisions.

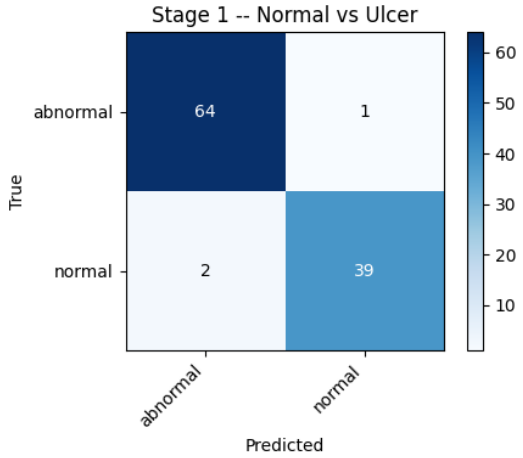

Figure 4. Stage 1 confusion matrix for DFU-MambaKAN on the binary normal-versus-ulcer screening test set ($n = 106$).

4.2. Stage 2: Dataset-Specific Four-Class Grade 1–4 Severity Grading

Table 4 reports test-set performance for Stage 2 dataset-specific four-class Grade 1–4 severity grading ($n = 1507$; class support 336/376/419/376 for Grades 1–4).

The four-class result shows a large performance gap. DFU-MambaKAN achieved 67.75% accuracy, which is 30.39 points below ResNet50 (98.14%), 30.59 points below EfficientNet-B0 (98.34%), 31.25 points below MobileNetV3-Small (99.00%), and 31.12 points below ViT-Tiny (98.87%). Its macro-F1 of 0.6772 is likewise far below the 0.9811–0.9899

baseline range. Accordingly, the proposed model is not competitive for multiclass severity grading under the reported protocol.

Table 4. Stage 2 test results for dataset-specific four-class Grade 1–4 severity grading ($n = 1507$).

Model	Acc.	F1	AUC	Params (M)	GPU (ms)
DFU-MambaKAN	0.6775	0.6772	0.8955	1.06	93.14
ResNet50	0.9814	0.9811	0.9964	23.52	7.59
EfficientNet-B0	0.9834	0.9834	0.9980	4.01	10.86
MobileNetV3-Small	0.9900	0.9899	0.9972	1.52	7.05
ViT-Tiny	0.9887	0.9886	0.9980	5.53	6.42

During development, validation accuracy increased from 44.8% at epoch 1 to 72.3% at epoch 14 of the 25-epoch run. Under-training or under-convergence is therefore one possible explanation, but no extended-training experiment, complete comparative learning-curve analysis, or alternative scheduler experiment tested this hypothesis. Other plausible factors include the larger dataset, dataset-specific class definitions, uncontrolled image quality, uncertain or noisy labels without clinical re-audit, hyperparameter selection, the global token-mixing behavior of the sequential scan, limited preservation of strict spatial correspondence, and architecture–task mismatch for fine-grained grade boundaries. Additional epochs, full learning-curve analysis, and multi-seed evaluation are needed to determine the contribution of convergence.

The proposed model's Stage 2 AUC was 0.8955, compared with 0.9964–0.9980 for the baselines. Although AUC is numerically closer than accuracy, it remains lower and does not demonstrate equivalent class separability or clinical usefulness. Results from the different DFUC2021 infection/ischaemia task [12] illustrate that multiclass DFU problems can be challenging, but they do not explain the much larger within-split gap observed here. Statistical significance was not tested; all between-model differences are descriptive single-run point-estimate comparisons.

Per-class precision/recall/F_1 values were 0.64/0.72/0.68, 0.67/0.61/0.64, 0.65/0.59/0.62, and 0.75/0.80/0.77 for Grades 1–4, respectively. Figure 5 records 12 Grade 1 images predicted as Grade 4 and 13 Grade 4 images predicted as Grade 1; it also records 67 Grade 2 images predicted as Grade 1 and 58 Grade 3 images predicted as Grade 2. These counts describe the observed confusion pattern but do not prove that the network learned a clinically valid ordinal severity representation. Severity grading in this form may eventually support preliminary prioritization, documentation, or longitudinal decision support, but it is not suitable for replacing specialist assessment or directing treatment on the basis of the present evidence.

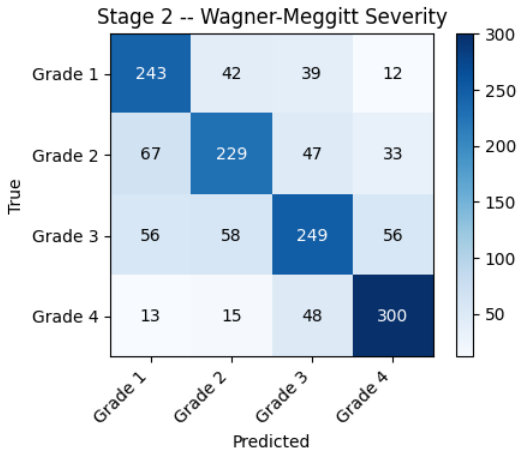

Figure 5. Stage 2 confusion matrix for DFU-MambaKAN on the dataset-specific four-class Grade 1–4 test set ($n = 1507$).

4.3. Efficiency Analysis

Table 5 separates trainable parameter count, the lower-bound FLOP estimate, single-image latency, and batched throughput. These metrics describe different deployment constraints and should not be treated as interchangeable.

Table 5. DFU-MambaKAN efficiency profile. Stage 2 batched throughput and latency-per-image were not measured.

Metric	Stage 1	Stage 2
Parameters (M)	1.062	1.062
FLOPs (G, fvcorelower bound)	0.2246	0.2246
GPU latency, batch = 1 (ms)	90.54	93.14
CPU (1 thread) latency, batch = 1 (ms)	93.78	90.86
GPU throughput, batch = 32 (images/s)	332.9	Not measured
GPU latency-per-image, batch = 32 (ms)	3.00	Not measured

Deployment interpretation. DFU-MambaKAN contains 1.062 M trainable parameters, compared with 1.52 M for MobileNetV3-Small, 4.01 M for EfficientNet-B0, 5.52–5.53 M for ViT-Tiny, and 23.51–23.52 M for ResNet50. Parameter count can influence serialized storage and weight memory, but storage footprint, activation memory, and peak device memory were not measured. At batch size 1, DFU-MambaKAN required 90.54 ms (Stage 1) and 93.14 ms (Stage 2) on the NVIDIA T4 GPU, whereas the baselines required 6.28–10.86 ms. Its single-thread CPU measurements were 93.78 ms and 90.86 ms and do not represent mobile-device performance. At batch size 32, Stage 1 throughput reached 332.9 images/s, or 3.00 ms/image; the corresponding Stage 2 batched measurements were not obtained. The results support parameter-efficient storage and batched, offline, retrospective, or queued screening more strongly than immediate single-image real-time inference. Smartphone or edge suitability requires device-specific latency, memory, and power evaluation. Accuracy vs Model Size comparision across both classification stages are highlighted in Figure 6. Performance comparison of different models are outlined in Figure 7.

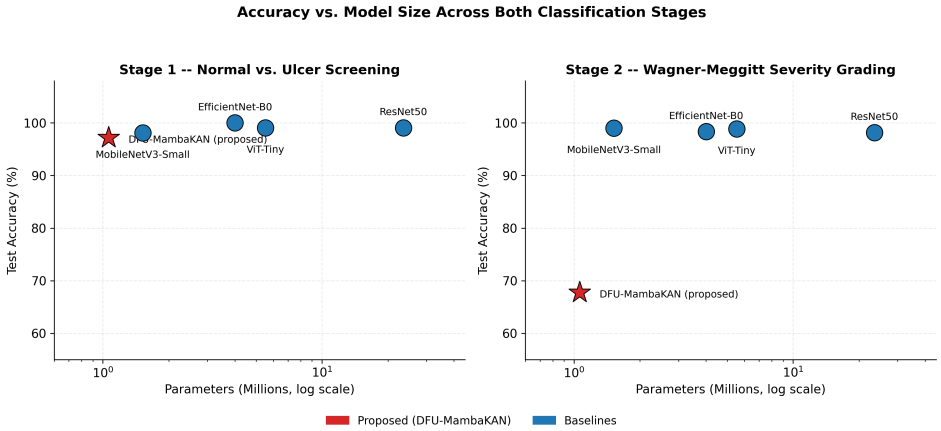

Figure 6. Test accuracy versus parameter count (log scale) for the two independent tasks. The proposed model (red star) has the lowest parameter count in both panels; the accuracy gap to the baselines is small for Stage 1 binary normal-versus-ulcer screening and substantial for Stage 2 dataset-specific four-class Grade 1–4 severity grading.

The latency pattern is consistent with the explicit pure-PyTorch sequential scan in Equation (4): 196 token positions across four blocks invoke many small operations, so launch and recurrence overhead can dominate at batch size 1. A fused implementation could reduce this overhead, but this possibility was not evaluated.

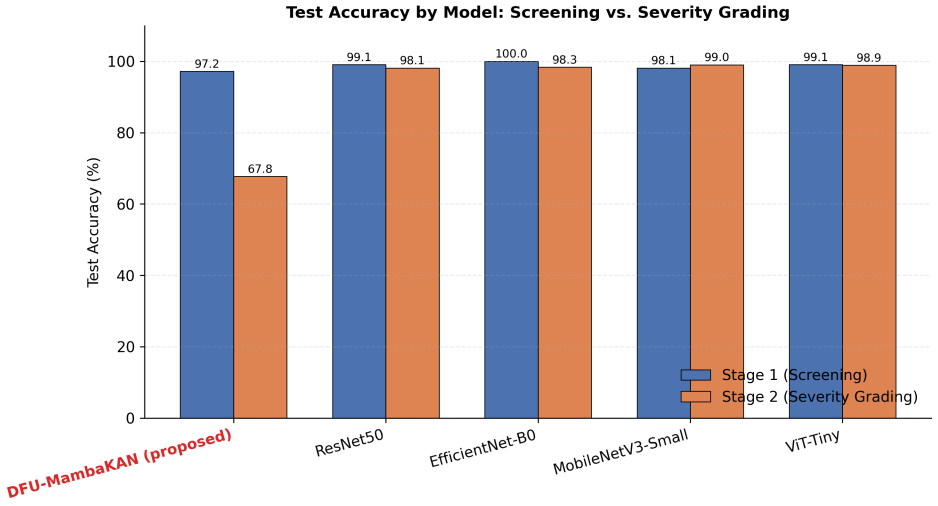

Figure 7. Per-model test accuracy for Stage 1 binary normal-versus-ulcer screening and Stage 2 dataset-specific four-class Grade 1–4 severity grading. All four baselines exceed 98% accuracy on both tasks, whereas the proposed model shows a substantial reduction from Stage 1 to Stage 2.

FLOP-count limitation. The 0.2246 GFLOPs value is an implementation-dependent lower-bound estimate from `fvcore`. It excludes exponentiation, softplus, recurrent elementwise multiplication and addition, and related per-position operations in Equation (4), repeated over 196 positions in each of four blocks. It is therefore not a complete measure of computational cost and should not be compared with fully counted operations as though definitions were identical. The measured latency results demonstrate directly that a lower reported FLOP estimate does not imply lower single-image inference time.

4.4. Ablation and Ranking Limitations

Systematic component ablation was not performed. The combined architecture is therefore presented as a proposed design rather than a proven optimal combination, and the contributions of the Mamba branch, RBF-KAN layer, convolutional stem, depthwise-convolution branch, raster tokenization strategy, and mean-pooled classification head remain unisolated. A future parameter-matched ablation should include: (i) Mamba with a conventional MLP and without KAN; (ii) KAN with local convolutional mixing and without Mamba; (iii) replacement of RBF-KAN by a conventional MLP of comparable parameter count; (iv) alternative state-space or token-mixing blocks; (v) removal of the depthwise-convolution branch; (vi) alternative stem and tokenization choices; and (vii) parameter-matched controls for the classification head and total model capacity.

Each table entry is a single-run point estimate from one split. Run-to-run variability was not quantified using a predefined repeated-seed protocol. Multi-seed averages, standard deviations, confidence intervals, and k-fold estimates are required before robustness, model ranking, or reproducibility can be assessed.

4.5. Explainability

DFU-MambaKAN does not expose a conventional final convolutional feature map. The token-based Grad-CAM visualization uses the final pre-pooling token tensor $Z \in \mathbb{R}^{L \times d}$ from the fourth hybrid block, where $L = 196$ and $d = 96$. For target-class logit $s^{(c)}$, the separately channel-averaged gradient and activation at token t are defined as

$$\bar{g}_t^{(c)} = \frac{1}{d} \sum_{k=1}^{d} \frac{\partial s^{(c)}}{\partial Z_{t,k}}, \qquad \bar{z}_t = \frac{1}{d} \sum_{k=1}^{d} Z_{t,k}, \tag{11}$$

and the token saliency score is

$$M_t^{(c)} = \mathrm{ReLU}\left(\bar{g}_t^{(c)} \bar{z}_t\right), \qquad t = 1, \ldots, L. \tag{12}$$

This implementation uses the product of the separately channel-averaged gradients and activations, rather than the channel-wise mean of gradient–activation products. The $L = 196$ scores are reshaped in raster order into a 14×14 map and bilinearly upsampled to 224×224 for overlay. Figure 8 illustrates the procedure; its numerical gradient and activation values are illustrative. Because the Mamba recurrence globally mixes the sequence before this tensor is extracted, strict one-token-to-one-local-feature correspondence is weakened even though the raster grid can be reconstructed geometrically.

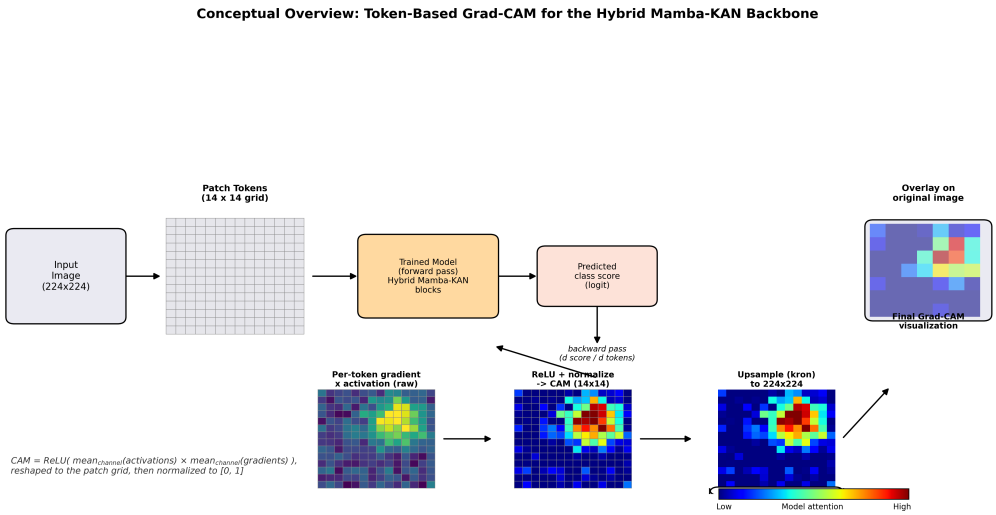

Figure 8. Conceptual illustration of the token-based Grad-CAM procedure. The numerical gradient, activation, and saliency values are illustrative rather than actual model outputs.

Figure 9 shows token-based Grad-CAM overlays for three correctly classified Stage 1 abnormal cases. The 14×14 resolution means that each grid position originates from a 16×16 input patch before global mixing, and the resulting maps are coarse and diffuse. The overlays are approximate qualitative saliency visualizations rather than clinically validated lesion localizations. They do not establish causal reasoning, diagnostic validity, or agreement with expert-defined ulcer boundaries. Alternative explanation methods and expert annotations were not evaluated.

4.6. Limitations

The principal limitations are: (i) each result is a single-run point estimate from one deterministic 70/15/15 split with seed 42; *k*-fold evaluation, repeated-seed analysis, confidence intervals, standard deviations, and inferential statistical testing were not performed; (ii) class-specific training and validation distributions were unavailable; (iii) Dataset A contains only 713 unique images after exact-hash deduplication; (iv) labels and image quality were not revalidated by clinical experts, and external cross-institutional and prospective clinical validation were not performed; (v) the 67.75% Stage 2 accuracy is substantially below all baselines, and extended training and comparative convergence studies were not performed; (vi) hyperparameters were not systematically optimized, and the original large-scale baseline recipes were not reproduced; (vii) component ablation and parameter-matched controls were not performed, leaving the contributions of Mamba, KAN, the stem, tokenization, local branch, and head unresolved; (viii) the 0.2246 GFLOPs value is a lower-bound estimate that omits recurrence-related operations; (ix) storage footprint, peak

memory, energy use, and smartphone or embedded-device latency were not measured; (x) token-based Grad-CAM overlays were not compared with alternative explanation methods or expert annotations and remain approximate qualitative saliency visualizations; and (xi) the exact-hash duplicate audit did not compare model performance before and after deduplication and therefore does not quantify the effect of deduplication on accuracy. These limitations preclude conclusions of universal superiority, strong reproducibility, clinical deployment readiness, real-time mobile suitability, or treatment-decision validity.

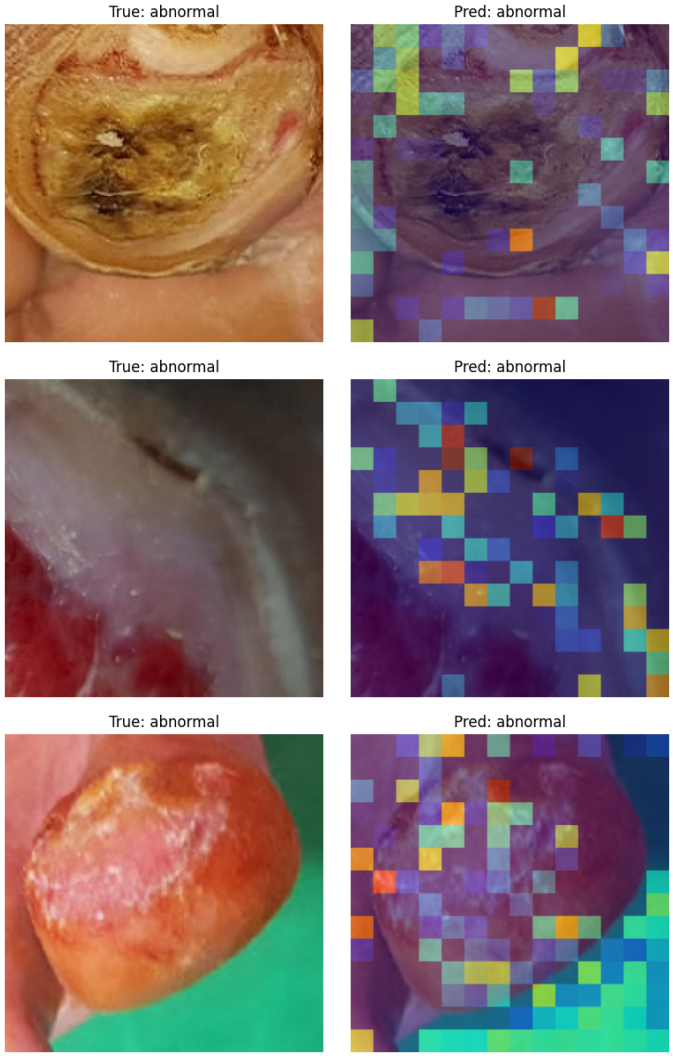

Figure 9. Token-based Grad-CAM overlays for three correctly classified Stage 1 abnormal test images. **Left**: original image; **right**: approximate qualitative saliency visualization.

5. Conclusions

This study evaluated DFU-MambaKAN as a proposed 1.062M-parameter hybrid architecture using a two-task evaluation with one deterministic 70/15/15 split and seed 42. Stage 1 binary normal-versus-ulcer screening achieved a single-run point estimate of 97.17% accuracy, macro-F1 0.9700, and AUC 0.9944. Stage 2 dataset-specific four-class Grade 1–4 severity grading achieved 67.75% accuracy, compared with 98.14–99.00% for the four baselines, and was therefore not competitive under the reported protocol. Underconvergence is one possible explanation rather than a confirmed cause; dataset characteristics, label quality, hyperparameters, global token mixing, and architecture–task mismatch also require investigation.

DFU-MambaKAN uses fewer parameters than all baselines, including 1.062 M versus 1.52 M for MobileNetV3-Small, but parameter efficiency did not translate into low single-image latency: the measured NVIDIA T4 latency was 90.54–93.14 ms, compared with 6.28–10.86 ms for the baselines. The Stage 1 batch-32 throughput of 332.9 images/s supports further evaluation for offline, retrospective, or queued screening workflows. It does not establish smartphone, edge-device, immediate real-time, or clinical deployment suitability. Binary screening may assist preliminary referral or triage, while automated grading may eventually support prioritization, documentation, or longitudinal monitoring [1,2]; neither replaces qualified clinical assessment or determines treatment.

The exact-hash audit removed 342 duplicates among 1055 Dataset A class-folder images (32.4%). The absence of a deduplicated-versus-nondeduplicated comparison means that the effect of deduplication on accuracy remains unquantified; the result is a caution about leakage risk rather than a measured demonstration of accuracy inflation. Priority future work includes complete learning-curve and extended-training analysis, multi-seed and *k*-fold evaluation, paired bootstrap or McNemar testing, systematic component and parameter-matched ablations, expert label and image-quality audit, external and prospective clinical validation, and device-specific measurements of storage, memory, latency, throughput, and power.

Author Contributions: M.N.A.N.: Conceptualization, Methodology, Writing original draft preparation. S.R.K.: Methodology, Software, Data curation. T.I.R.: Software, Validation, Visualization. M.A.R. (Mainuddin Adel Rafi): Validation, Formal analysis. U.A.: Validation, Investigation, Visualization, Writing review and editing. M.R.H.: Formal analysis, Investigation. M.A.R. (Md Al Ridwan): Investigation, Data curation. R.U.: Conceptualization, Resources, Supervision, Project administration, Funding acquisition, Writing review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source datasets analyzed in this study are publicly available on Kaggle: Dataset A at https://www.kaggle.com/datasets/laithjj/diabetic-foot-ulcer-dfu accessed on 15 June 2026 and Dataset B at https://www.kaggle.com/datasets/khalidsiddiqui2003/dfu-dataset-annotated-into-4-classes accessed on 15 June 2026. The study-generated deduplicated split indices, exact duplicate hashes, source code, package environment, and model configurations are available from the corresponding author on reasonable request, subject to the redistribution conditions of the source datasets. Dataset images should be obtained from their original public sources.

Acknowledgments: The author(s) acknowledge the creators and maintainers of the public Kaggle-hosted datasets used in this study, and the original authors of the open-source baseline architectures and tooling on which this work builds.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sun, H.; Saeedi, P.; Karuranga, S.; Pinkepank, M.; Ogurtsova, K.; Duncan, B.B.; Stein, C.; Basit, A.; Chan, J.C.; Mbanya, J.C.; et al. IDF Diabetes Atlas: Global, Regional and Country-Level Diabetes Prevalence Estimates for 2021 and Projections for 2045. *Diabetes Res. Clin. Pract.* **2022**, *183*, 109119. [CrossRef] [PubMed]
2. Armstrong, D.G.; Boulton, A.J.M.; Bus, S.A. Diabetic Foot Ulcers and Their Recurrence. *N. Engl. J. Med.* **2017**, *376*, 2367–2375. [CrossRef] [PubMed]
3. Wagner, F.W., Jr. The Dysvascular Foot: A System for Diagnosis and Treatment. *Foot Ankle* **1981**, *2*, 64–122. [CrossRef] [PubMed]

4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778. [CrossRef]
5. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), PMLR, Long Beach, CA, USA, 9–15 June 2019; Volume 97; pp. 6105–6114.
6. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Reoublic of Korea, 27 October–2 November 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1314–1324. [CrossRef]
7. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
8. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 618–626. [CrossRef]
9. Goyal, M.; Reeves, N.D.; Davison, A.K.; Rajbhandari, S.; Spragg, J.; Yap, M.H. DFUNet: Convolutional Neural Networks for Diabetic Foot Ulcer Classification. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *4*, 728–739. [CrossRef]
10. Goyal, M.; Reeves, N.D.; Rajbhandari, S.; Yap, M.H. Robust Methods for Real-Time Diabetic Foot Ulcer Detection and Localization on Mobile Devices. *IEEE J. Biomed. Health Inform.* **2019**, *23*, 1730–1741. [CrossRef] [PubMed]
11. Goyal, M.; Reeves, N.D.; Rajbhandari, S.; Ahmad, N.; Wang, C.; Yap, M.H. Recognition of Ischaemia and Infection in Diabetic Foot Ulcers: Dataset and Techniques. *Comput. Biol. Med.* **2020**, *117*, 103616. [CrossRef] [PubMed]
12. Yap, M.H.; Cassidy, B.; Pappachan, J.M.; O'Shea, C.; Gillespie, D.; Reeves, N.D. Analysis Towards Classification of Infection and Ischaemia of Diabetic Foot Ulcers. In *Proceedings of the IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), Virtually, 27–30 July 2021*; IEEE: Piscataway, NJ, USA, 2021. [CrossRef]
13. Yap, M.H.; Hachiuma, R.; Alavi, A.; Brüngel, R.; Cassidy, B.; Goyal, M.; Zhu, H.; Rückert, J.; Olshansky, M.; Huang, X.; et al. Deep Learning in Diabetic Foot Ulcers Detection: A Comprehensive Evaluation. *Comput. Biol. Med.* **2021**, *135*, 104596. [CrossRef] [PubMed]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 10012–10022.
16. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv* **2023**, arXiv:2312.00752.
17. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; Liu, Y. VMamba: Visual State Space Model. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 10–15 December 2024.
18. Ma, J.; Li, F.; Wang, B. U-Mamba: Enhancing Long-Range Dependency for Biomedical Image Segmentation. *arXiv* **2024**, arXiv:2401.04722.
19. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. KAN: Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2404.19756.
20. Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Yuan, Y. U-KAN Makes Strong Backbone for Medical Image Segmentation and Generation. *arXiv* **2024**, arXiv:2406.02918.
21. Cheon, M. Demonstrating the Efficacy of Kolmogorov-Arnold Networks in Vision Tasks. *arXiv* **2024**, arXiv:2406.14916.
22. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2019**, arXiv:1711.05101.
23. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 2980–2988.