Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

RWKV-Med: A Lightweight Hybrid RWKV-Hyena Network with Axial Decoder Refinement for Skin Lesion Segmentation

Zeeshan Haider
Southwest Jiaotong University
Haiyu Huang
hyhuang@home.swjtu.edu.cn
Southwest Jiaotong University
Muqadar Ali
Southwest Jiaotong University
Muhammad Adnan Farooq Butt
Southwest Jiaotong University

Research Article

Keywords:

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11647761/v1

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

Additional Declarations: No competing interests reported.

RWKV-Med: A Lightweight Hybrid RWKV–Hyena Network with Axial Decoder Refinement for Skin Lesion Segmentation

Zeeshan Haider[1], Haiyu Huang[1*], Muqadar Ali[1], Muhammad Adnan Farooq Butt[1]

[1]School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China.

*Corresponding author(s). E-mail(s): hyhuang@home.swjtu.edu.cn;
Contributing authors: zeeshan@my.swjtu.edu.cn;
ali.muqadar@my.swjtu.edu.cn; adnanbutt@my.swjtu.edu.cn;

Abstract

Accurate segmentation of skin lesions in dermoscopic images supports computer-aided melanoma diagnosis, but most accurate networks are too large for mobile or point-of-care use. We present RWKV-Med, a lightweight hierarchical encoder–decoder that places two linear-complexity token mixers where each is most effective. Receptance weighted key value (RWKV) blocks, which combine a quad-directional token shift with bidirectional weighted key-value aggregation, process the high-resolution early stages, whereas HyenaPixel blocks, which apply an implicitly parameterised large-kernel convolution through the fast Fourier transform, process the low-resolution late stages. An all-MLP decoder fuses the four scales and is refined by a pointwise bottleneck module and an axial depthwise module with dual global pooling. With 0.69 million parameters and 0.55 giga floating-point operations (GFLOPs), RWKV-Med reaches a mean intersection over union (mIoU) and mean Dice coefficient (mDice) of 90.35/94.83, 84.60/91.52 and 84.95/91.76 on the ISIC2016, ISIC2017 and ISIC2018 benchmarks. It ranks first on all five metrics on ISIC2018 and has the highest mIoU averaged over the three datasets among twelve compared methods. Trained on ISIC2018 and tested without fine-tuning on PH2, it obtains the best mIoU of six models (89.60%). An ablation on ISIC2017 shows that each added component improves mIoU, by 1.10 (pointwise bottleneck), 1.61 (axial module) and 0.97 points (RWKV stages).

Keywords: skin lesion segmentation, lightweight network, RWKV, HyenaPixel, global convolution, dermoscopy

1 Introduction

Skin diseases affect people of every age and region. The Global Burden of Disease Study 2021 indicates that skin and subcutaneous conditions affect roughly a quarter of the world's population [1], and skin cancer accounted for an estimated 1.5 million new cases in 2022, about 330,000 of them melanoma [2].

Early detection matters. Five-year relative survival for melanoma exceeds 99% when the disease is localised but is 36% once it has spread to distant sites. Dermoscopy exposes sub-surface structures that help clinicians separate visually similar lesions, and deep learning has made automated lesion analysis practical [3]. Lesion segmentation is a basic step, since the extracted boundary feeds measurements such as size, border irregularity and asymmetry, and downstream classification. However, many accurate models are computationally demanding and hard to deploy on mobile and other resource-constrained devices. Lightweight yet accurate segmentation models could make computer-assisted assessment available where specialists and computing resources are scarce.

Convolutional encoders are efficient but have a limited receptive field, whereas self-attention captures global context at a cost that is quadratic in the number of tokens. Linear-complexity token mixers offer a middle path: state-space models such as Mamba [4–6], receptance weighted key value (RWKV) models [7, 8], and implicit global convolutions such as Hyena and HyenaPixel [9, 10]. These mixers differ. RWKV aggregates tokens with a learnable distance decay that retains a locality bias, which suits the many tokens of high-resolution feature maps. An FFT-based implicit large-kernel convolution, whose kernel spans the feature map, gives each position image-scale context at a cost that becomes negligible when the feature map is small.

Relation to our previous work. Our earlier model HyenaMed [11] used HyenaPixel blocks in every encoder stage. The present paper is a substantially extended study with the following differences: (i) a hybrid encoder that uses RWKV blocks in the two high-resolution stages; (ii) a redesigned decoder with the PBRP and ADWP refinement modules; (iii) a formal description and complexity analysis; (iv) a broader evaluation, comprising cross-dataset testing on PH2, measured GPU and CPU latency and memory, and an ablation that adds the RWKV stages step by step. The contributions of this paper are as follows.

- We propose **RWKV-Med**, a four-stage hybrid encoder that uses inverted-residual RWKV blocks at strides 4 and 8 and HyenaPixel (HPx) blocks at strides 16 and 32, so that each token mixer works at the resolution that suits it.
- We give a unified formulation of the encoder together with a complexity analysis showing that both mixers are (near-)linear in the number of tokens.
- We pair the encoder with an all-MLP decoder refined by PBRP and ADWP, which re-weights channels from dual global pooling over axially mixed features at negligible cost.
- With 0.69M parameters and 0.55 GFLOPs, RWKV-Med is competitive with or better than eleven CNN, Transformer and Mamba baselines and our earlier HyenaMed on ISIC2016, ISIC2017 and ISIC2018, and we analyse measured latency, generalisation to PH2 and the contribution of each component.

2 Related work

2.1 Lightweight skin lesion segmentation

Deep learning dominates skin lesion segmentation [3]. The U-Net family [12] is the standard template, but its encoder–decoder with dense skip connections is expensive. Lightweight variants replace convolutions with MLPs (UNeXt [13]), add multi-attention modules to a very small UNet (MALUNet [14]), or use grouped multi-axis attention (EGE-UNet [15]), reaching the million-parameter range or below; DSNet [16] combines detail-enhanced separable convolutions with dynamic gate attention. Hybrid CNN–Transformer models such as TransUNet [17] and Swin-Unet [18] add global context at the price of many more parameters.

2.2 Vision Transformers and token mixing

The Vision Transformer (ViT) [19] treats an image as a sequence of patch tokens processed by self-attention. Hierarchical four-stage designs with convolutional down-sampling, such as Swin [20] with shifted-window attention, give consistent gains on dense prediction, and other designs add convolutions to the Transformer block [21]. MetaFormer [22] argues that the block structure matters more than the specific mixer, which motivates cheaper mixers. SegFormer [23] pairs a hierarchical encoder with an all-MLP decoder and is the template for our decoder.

2.3 ConvNets and large kernels

Convolutional networks [24] recovered much of the ground lost to Transformers once they adopted similar macro-design and training recipes, as in ConvNeXt [25]. A major trend is to enlarge the receptive field with large kernels. SegNeXt [26] uses parallel strip convolutions of sizes 7 to 21, RepLKNet [27] scales dense kernels to 31×31, and CKConv [28] generates kernels from relative positions with a small network. Hyena [9] and HyenaPixel [10] generate global filters implicitly from positional encodings and apply them with the FFT, decoupling the parameter count from the kernel size and allowing kernels as large as the feature map at $\mathcal{O}(L \log L)$ cost. We use HyenaPixel as the low-resolution mixer.

2.4 Vision Mamba, RWKV and hybrid architectures

The selective-scan state-space model Mamba [4] models long sequences with linear complexity. Vision Mamba [5] scans patch tokens bidirectionally, VMamba [6] adds a 2D cross-scan, and MambaVision [29] mixes Mamba and self-attention layers. For dense prediction, SegMAN [30] combines state-space and local-attention modules, and EfficientViM [31] performs the state-space computation in a compact hidden state. In medical imaging, U-Mamba [32] and VM-UNet [33] place Mamba blocks inside U-shaped networks. RWKV [7] is a linear-complexity alternative combining the parallel training of Transformers with the recurrent form of RNNs. Vision-RWKV [8] adapts it to images with a quad-directional token shift (Q-Shift) and a bidirectional WKV operator (Bi-WKV), and RWKV-UNet [34] and U-RWKV [35] bring RWKV into

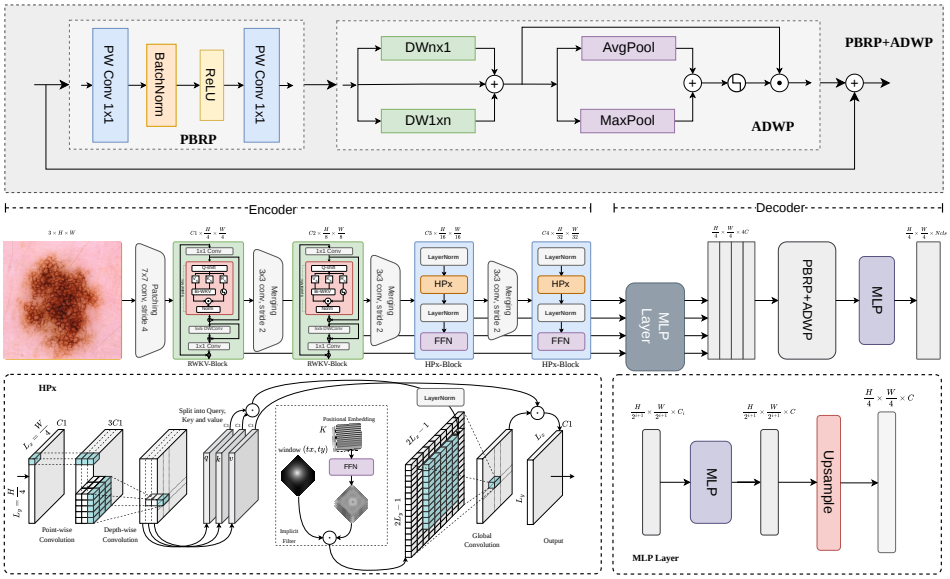

Fig. 1 Architecture of RWKV-Med. A 7×7 stride-4 patch embedding is followed by two RWKV stages and two HPx stages joined by 3×3 stride-2 merging convolutions. An all-MLP decoder unifies the four scales, and the PBRP+ADWP module (top) refines the fused feature before the classifier. The lower panels detail the HPx operator and the MLP layer

medical segmentation, the latter with a lightweight focus. These models use one kind of mixer throughout the encoder. RWKV-Med instead combines an RWKV mixer at high resolution with an implicit global convolution at low resolution, and adds a decoder refinement tailored to lesion masks.

3 Proposed method

3.1 Overview

Given a dermoscopic image $I \in \mathbb{R}^{3\times H\times W}$, RWKV-Med produces a mask $M \in \mathbb{R}^{N_{cls}\times H/4\times W/4}$ ($N_{cls} = 2$: lesion and background), which is bilinearly upsampled to the input size. The encoder has four stages with channel widths $C_1, \ldots, C_4$ and strides $4, 8, 16, 32$ (Fig. 1). A 7×7 stride-4 convolution embeds the image,

$$F_0 = \mathrm{Conv}_{7\times 7,\, s=4}(I), \tag{1}$$

and stage i, with feature map $F_i \in \mathbb{R}^{C_i\times H_i\times W_i}$, $H_i = H/2^{i+1}$, $W_i = W/2^{i+1}$, is

$$F_i = \begin{cases} \mathcal{R}_i\big(\mathrm{Merge}_i(F_{i-1})\big), & i \in \{1,2\}, \\ \mathcal{H}_i\big(\mathrm{Merge}_i(F_{i-1})\big), & i \in \{3,4\}, \end{cases} \tag{2}$$

where Merge_1 is the identity, $\mathrm{Merge}_{i\geq 2}$ is a 3×3 stride-2 convolution, $\mathcal{R}_i$ is a stack of d_i RWKV blocks and $\mathcal{H}_i$ a stack of d_i HPx blocks. For a 256×256 input the stages hold 4096, 1024, 256 and 64 tokens. The allocation follows the cost profile of the two mixers (Section 3.4): Bi-WKV costs $\mathcal{O}(TC)$ in the token count T and keeps a local bias through token shift and distance decay, which suits the large T of the first two stages, whereas the FFT convolution costs $\mathcal{O}(TC\log T)$ and gives a near-global receptive field that is cheap once T is small. The ablation in Table 6 adds the RWKV blocks to the first and then to the second stage.

3.2 RWKV block (stages 1–2)

Each RWKV block is an inverted-residual unit whose spatial mixer follows Vision-RWKV [8]. Let $X \in \mathbb{R}^{C\times h\times w}$ be the block input and $T = hw$ the number of tokens. *Q-Shift* moves a quarter of the channels from each of the four neighbouring positions,

$$X^{\dagger} = \mathrm{Concat}\big(X^{0:\frac{C}{4}}_{h-1,w},\ X^{\frac{C}{4}:\frac{C}{2}}_{h+1,w},\ X^{\frac{C}{2}:\frac{3C}{4}}_{h,w-1},\ X^{\frac{3C}{4}:C}_{h,w+1}\big) \tag{3}$$

$$\text{Q-Shift}_{*}(X) = X + (1-\mu_{*})\odot X^{\dagger}, \quad *\in\{k,v,r\}, \tag{4}$$

with learnable channel-wise mixing vectors μ_{*}. The key, value and receptance are

$$\begin{aligned} K &= \text{Q-Shift}_{k}(X)W_k, \qquad V = \text{Q-Shift}_{v}(X)W_v,\\ R &= \text{Q-Shift}_{r}(X)W_r. \end{aligned} \tag{5}$$

Bi-WKV aggregates all tokens of the flattened map with a distance-dependent decay. For token t and each channel,

$$\begin{aligned} \omega_{t,t'} &= \exp\Big(-\tfrac{|t-t'|-1}{T}\,w + k_{t'}\Big),\\ a_t &= \sum_{t'\neq t}\omega_{t,t'}\,v_{t'} + e^{u+k_t}v_t,\\ b_t &= \sum_{t'\neq t}\omega_{t,t'} + e^{u+k_t}, \qquad \mathrm{wkv}_t = a_t/b_t, \end{aligned} \tag{6}$$

where w (spatial decay) and u (bonus for the current token) are learnable channel-wise parameters and the sums run over all T tokens. The receptance gates the aggregated value and the result is normalised,

$$O = \mathrm{LN}\left(\sigma(R)\odot \mathrm{wkv}\right). \tag{7}$$

Bi-WKV can be evaluated in $\mathcal{O}(TC)$ time by a linear-time recurrence, in contrast with the $\mathcal{O}(T^2C)$ of self-attention. Writing the spatial mixer as $\mathrm{SM}(\cdot)$, the block wraps it between pointwise convolutions and adds a depthwise convolution for local detail (Fig. 1):

$$\begin{aligned} Z_0 &= \mathrm{PW}_{\uparrow}(X), \qquad Z_1 = Z_0 + \mathrm{SM}(Z_0),\\ Z_2 &= Z_1 + \mathrm{DW}_{5\times5}(Z_1), \qquad Y = X + \mathrm{PW}_{\downarrow}(Z_2), \end{aligned} \tag{8}$$

where $\mathrm{PW}_{\uparrow}$ and $\mathrm{PW}_{\downarrow}$ are 1×1 convolutions that expand and project the channels by a factor $e = 2$.

3.3 HyenaPixel block (stages 3–4)

An HPx block follows the MetaFormer layout [22] with the HyenaPixel operator [10] as token mixer:

$$\begin{aligned} X' &= X + \mathrm{HPx}\big(\mathrm{LN}(X)\big), \\ Y &= X' + \mathrm{FFN}\big(\mathrm{LN}(X')\big). \end{aligned} \tag{9}$$

Inside HPx, a pointwise followed by a depthwise convolution expands the input to $3C$ channels, split into query, key and value maps $q, k, v \in \mathbb{R}^{C \times L_y \times L_x}$ with $L_y = H_i$, $L_x = W_i$:

$$\begin{aligned} [q; k; v] &= \mathrm{DW}\big(\mathrm{PW}(X)\big), \\ y &= g\big(\mathrm{LN}(q \odot k)\big) \odot v. \end{aligned} \tag{10}$$

Here $g(\cdot)$ is a non-causal channel-wise 2D convolution with a large centred kernel $h \in \mathbb{R}^{C \times k \times k}$:

$$g(z)[m, n] = \sum_{m', n'} h[m - m',\, n - n']\; z[m', n']. \tag{11}$$

The kernel size is set slightly above the side of the feature map ($k = 19$ and $k = 9$ for the 16×16 and 8×8 maps of a 256×256 input), so that positions near the centre aggregate the whole map and border positions at least 62.5% of it along each axis. Instead of storing h we generate it with a small network from a positional embedding and shape it with a decaying window,

$$h = \mathrm{FFN}_{\theta}(\rho) \odot \mathrm{Win}, \tag{12}$$

where FFN_{θ} uses sinusoidal activations. The embedding ρ is a truncated complex-exponential basis over the offsets (m, n),

$$\begin{aligned} \rho_{k_x, k_y}(m, n) = \; & \exp\Big(\tfrac{i 2\pi k_x m}{L_x}\Big) \\ & \times \exp\Big(\tfrac{i 2\pi k_y n}{L_y}\Big), \end{aligned} \tag{13}$$

with $k_x < K_x$, $k_y < K_y$ (real and imaginary parts concatenated; the embedding dimension is 48 in both HPx stages), and the window imposes a learnable spatial decay and bias,

$$\begin{aligned} \mathrm{Win}(t_x, t_y) = \; & \exp\Big(-\alpha \sqrt{(t_x - c_x)^2 + (t_y - c_y)^2}\Big) \\ & + b, \end{aligned} \tag{14}$$

where (c_x, c_y) is the filter centre, α a learnable decay and b a bias. The convolution is evaluated in the frequency domain, $g(z) = \mathcal{F}^{-1}\big(\mathcal{F}(h) \odot \mathcal{F}(z)\big)$, with zero-padding to

$(2L_y - 1) \times (2L_x - 1)$. The feed-forward network uses StarReLU [22],

$$\begin{aligned} \mathrm{FFN}(x) &= W_2\ \mathrm{StarReLU}(W_1 x), \\ \mathrm{StarReLU}(x) &= s \cdot \big(\mathrm{ReLU}(x)\big)^2 + b, \end{aligned} \tag{15}$$

with scalar learnable scale s and bias b.

3.4 Complexity

Let $T_i = H_i W_i$ be the number of tokens at stage i. A self-attention mixer costs $\mathcal{O}(T_i^2 C_i)$, Bi-WKV costs $\mathcal{O}(T_i C_i)$ and an HPx mixer costs $\mathcal{O}(T_i C_i \log T_i)$ because of the FFT, so the whole encoder is (near-)linear in the number of pixels. The implicit filter FFN_θ has a fixed size, so the parameter count of HPx does not grow with the kernel size, which is why kernels larger than the side of the feature map are affordable in a sub-million-parameter model.

3.5 Decoder with PBRP and ADWP

The decoder follows SegFormer [23]. Each stage output is projected to a common width C by a 1×1 convolution with batch normalisation and ReLU and upsampled to $H/4 \times W/4$ (Fig. 1, lower right); the four maps are concatenated and fused,

$$\hat{F}_i = \mathrm{Up}\big(\mathrm{ReLU}(\mathrm{BN}(\mathrm{PW}(F_i)))\big), \tag{16}$$

$$F = \mathrm{BN}\Big(\mathrm{PW}_{4C\to C}\big(\mathrm{Concat}(\hat{F}_1, \ldots, \hat{F}_4)\big)\Big). \tag{17}$$

Two light-weight modules refine F before classification. *PBRP* (pointwise bottleneck with ReLU projection) applies two 1×1 projections that keep the width C, with batch normalisation and ReLU between them,

$$P = \mathrm{PW}_2\big(\mathrm{ReLU}(\mathrm{BN}(\mathrm{PW}_1(F)))\big). \tag{18}$$

ADWP (axial depthwise convolution with pooling) first mixes spatial context along the two axes with depthwise strip convolutions ($n = 7$) and an identity path,

$$A = P + \mathrm{DW}_{n\times 1}(P) + \mathrm{DW}_{1\times n}(P), \tag{19}$$

then derives a channel gate from global average and global max pooling, in the spirit of channel attention, using a shared two-layer 1×1 bottleneck MLP ($C \to C/r \to C$ with ReLU and no bias, $r = 16$), and re-weights A channel-wise ($s \in \mathbb{R}^{C\times 1\times 1}$) with a residual connection to P:

$$s = \sigma\big(\mathrm{MLP}(\mathrm{GAP}(A)) + \mathrm{MLP}(\mathrm{GMP}(A))\big), \tag{20}$$

$$\bar{A} = P + s \odot A. \tag{21}$$

A residual connection preserves the fused feature and a linear classifier produces the mask,

$$F' = F + \bar{A}, \qquad M = \mathrm{Linear}(C, N_{cls})(F'). \tag{22}$$

Strip convolutions approximate an $n \times n$ window with $2nC$ instead of n^2C weights and the pooled descriptor adds no spatial cost; with $C = 64$, $n = 7$ and $r = 16$, ADWP has about 1.5 thousand parameters (1,024 in the strip convolutions including biases and 512 in the gate), so it adds negligible parameters and FLOPs.

3.6 Training objective

Training uses the sum of a cross-entropy and a Dice loss, $\mathcal{L}_{seg} = \mathcal{L}_{CE} + \mathcal{L}_{Dice}$, on the main prediction, and the same two losses on an auxiliary prediction,

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \lambda \, \mathcal{L}_{aux}, \tag{23}$$

with $\lambda = 0.4$. The auxiliary head is an FCN head (one 3×3 convolution, 64 channels, dropout 0.1) on the stage-3 feature map, and it is used only during training, so it adds no cost at inference.

4 Experiments

4.1 Datasets and evaluation protocol

We evaluate on the binary lesion-segmentation tasks of ISIC2016, ISIC2017 and ISIC2018 with 800/100/379 (ISIC2016), 2,000/150/600 (ISIC2017) and 2,594/100/1,000 (ISIC2018) training/validation/test images. To test generalisation, models trained on ISIC2018 are evaluated without fine-tuning on the PH2 database (200 images, used as provided without any pre-processing).

We report the class-averaged metrics of the MMSegmentation evaluator: mIoU, mDice, overall pixel accuracy (aAcc), mean precision (mPre) and mean recall (mRec), averaged over the lesion and background classes. Because the background is the larger and easier class, these scores are higher than foreground-only scores and are not directly comparable with papers that report only lesion Dice or IoU. Efficiency is measured by parameters, FLOPs at 256×256 input, model size, and measured latency and peak memory (Section 4.4).

4.2 Implementation details

All models use MMSegmentation on PyTorch 2.1.0 (CUDA 11.8 runtime) and are trained on two NVIDIA A800 80 GB PCIe GPUs with synchronised batch normalisation. Training runs for 90,000 iterations with batch size 4 and inputs resized to 256×256, using AdamW with initial learning rate 1×10^{-4}, $\beta = (0.9, 0.999)$, weight decay 0.1, layer-wise learning-rate decay and linear (polynomial of power 1) decay to zero; the validation set is evaluated every 1,000 iterations. The stochastic-depth rate is 0.05 and the dropout rate of the decoder is 0.01. Augmentation consists of random resizing, cropping, horizontal flipping and photometric distortion. All baselines are trained with identical settings. RWKV-Med has one block per stage (depths

$(d_1, d_2, d_3, d_4) = (1, 1, 1, 1)$) and channel widths $(C_1, C_2, C_3, C_4) = (64, 64, 64, 128)$. Stages 1–2 use RWKV blocks, and stages 3–4 use HPx blocks with a filter embedding dimension of 48 and kernel sizes of 19 and 9, respectively. HyenaMed uses HPx blocks in all four stages, with widths $(32, 32, 64, 128)$, embedding dimensions $(24, 24, 32, 48)$ and kernel sizes $(75, 37, 19, 9)$; RWKV-Med therefore differs in the two replaced stages, in doubled widths in stages 1–2 and in a stage-3 embedding dimension of 48 instead of 32, while the kernel sizes of stages 3–4 are unchanged. The decoder width is $C = 64$. The FFN expansion ratio is 4, and the ADWP strip length is $n = 7$ with gate reduction ratio $r = 16$. Each model is trained once and all results are from a single run.

4.3 Comparison with state-of-the-art methods

We compare with eleven baselines spanning CNNs (U-Net [12], DeepLabV3+ [36], MobileNetV3 [37], Fast-SCNN [38], SegNeXt [26], PIDNet [39]), a Transformer (SegFormer [23]), Mamba-based models (Vision Mamba [5], SegMAN [30], EfficientViM [31]) and S2Net [40], and additionally with our previous model HyenaMed [11]. Tables 1 and 2 give the results.

Accuracy. On ISIC2018, RWKV-Med is first on all five metrics (84.95% mIoU, 91.76% mDice), 1.51 mIoU and 0.90 mDice points above the best competitor, SegNeXt. On ISIC2016 it is second, 0.15 mIoU and 0.09 mDice points below SegNeXt. On ISIC2017 it is third, trailing EfficientViM by 0.67 mIoU and 0.35 mDice points although EfficientViM has 31.5× more parameters, and is on par with Vision Mamba (84.60 vs. 84.56 mIoU). Recall is its weakest metric on ISIC2016 and ISIC2017 (Table 2). Averaged over the three datasets it has the highest mIoU (86.63%) and mDice (92.70%), ahead of SegNeXt (86.30%, 92.51%) and Vision Mamba (85.79%, 92.20%).

Efficiency. RWKV-Med needs 0.69M parameters and 0.55 GFLOPs; only HyenaMed has fewer parameters (0.68M), and only HyenaMed (0.30 G) and Fast-SCNN (0.23 G) need fewer FLOPs. Compared with U-Net it needs 42.0× fewer parameters and 92.2× fewer FLOPs, and compared with Vision Mamba 17.5× and 16.8× fewer. It improves on SegFormer (3.71M, 1.77 G) by 5.00, 3.67 and 6.07 mIoU points and on MobileNetV3 by 3.09, 0.84 and 2.32 points on the three datasets. Against HyenaMed, the hybrid encoder gains 0.84, 0.97 and 2.39 mIoU points for 0.01M extra parameters and 0.25 extra GFLOPs (1.83×).

4.4 Accuracy–efficiency analysis

Fig. 2 plots mean mIoU against parameters and FLOPs. RWKV-Med lies on the accuracy–efficiency frontier in both views: no compared model with fewer parameters is more accurate, and the more accurate SegNeXt, Vision Mamba and DeepLabV3+ need 6.1×, 17.5× and 17.8× more parameters. FLOPs and parameters are only proxies for speed on real hardware, so Table 3 reports measured latency.

Latency was measured with batch size 1 and 256×256 inputs on an NVIDIA A800 80 GB PCIe GPU and an AMD EPYC 7453 28-core CPU (50 warm-up iterations, mean of three timed runs). RWKV-Med needs 4.31 ms (GPU) and 10.82 ms (CPU) per image, 8.8% and 10.7% more than HyenaMed (3.96 and 9.77 ms), with a peak memory

Table 1 Comparison with state-of-the-art methods on ISIC2016, ISIC2017 and ISIC2018 (%). Params in M; FLOPs in G; and Size in MB

		Efficiency			Performance		
Dataset	**Method**	**Params**	**FLOPs**	**Size**	**mIoU**	**mDice**	**aAcc**
ISIC2016	U-Net [12]	28.98	50.73	231.5	87.43	94.63	95.82
	DeepLabV3+ [36]	12.31	13.56	104.9	89.93	94.58	96.75
	MobileNetV3 [37]	1.13	1.05	16.5	87.26	93.02	95.80
	Fast-SCNN [38]	1.39	0.23	24.0	88.87	93.97	96.41
	SegFormer [23]	3.71	1.77	51.3	85.35	91.83	93.39
	SegNeXt [26]	4.22	1.57	60.4	90.50	94.92	96.94
	PIDNet [39]	7.71	1.48	73.7	89.72	94.47	96.65
	Vision Mamba [5]	12.10	9.26	139.0	89.59	94.40	96.66
	SegMAN [30]	6.38	1.47	73.5	87.48	93.15	95.84
	EfficientViM [31]	21.70	1.42	248.9	84.99	91.65	94.80
	S2Net [40]	13.57	1.46	171.6	89.46	94.31	96.68
	HyenaMed [11]	0.68	0.30	19.83	89.51	94.34	96.67
	RWKV-Med (Ours)	0.69	0.55	24.1	90.35	94.83	96.92
ISIC2017	U-Net [12]	28.98	50.73	231.5	79.89	88.63	90.50
	DeepLabV3+ [36]	12.31	13.56	104.9	84.12	91.23	93.03
	MobileNetV3 [37]	1.13	1.05	16.5	83.76	91.01	92.83
	Fast-SCNN [38]	1.39	0.23	24.0	82.23	90.06	92.11
	SegFormer [23]	3.71	1.77	51.3	80.93	89.26	91.31
	SegNeXt [26]	4.22	1.57	60.4	84.96	91.74	93.38
	PIDNet [39]	7.71	1.48	73.7	82.01	89.95	91.71
	Vision Mamba [5]	12.10	9.26	139.0	84.56	91.51	93.21
	SegMAN [30]	6.38	1.47	73.5	83.41	90.81	92.58
	EfficientViM [31]	21.70	1.42	248.9	85.27	91.87	93.34
	S2Net [40]	13.57	1.46	171.6	81.76	89.76	91.96
	HyenaMed [11]	0.68	0.30	19.83	83.63	90.95	92.65
	RWKV-Med (Ours)	0.69	0.55	24.1	84.60	91.52	93.23
ISIC2018	U-Net [12]	28.98	50.73	231.5	77.53	87.12	89.03
	DeepLabV3+ [36]	12.31	13.56	104.9	82.49	90.27	91.82
	MobileNetV3 [37]	1.13	1.05	16.5	82.63	90.36	91.83
	Fast-SCNN [38]	1.39	0.23	24.0	83.29	90.76	92.30
	SegFormer [23]	3.71	1.77	51.3	78.88	87.97	90.03
	SegNeXt [26]	4.22	1.57	60.4	83.44	90.86	92.22
	PIDNet [39]	7.71	1.48	73.7	76.60	86.58	88.06
	Vision Mamba [5]	12.10	9.26	139.0	83.21	90.71	92.25
	SegMAN [30]	6.38	1.47	73.5	79.76	88.58	90.12
	EfficientViM [31]	21.70	1.42	248.9	82.99	90.58	91.89
	S2Net [40]	13.57	1.46	171.6	82.36	90.17	91.85
	HyenaMed [11]	0.68	0.30	19.83	82.56	90.31	91.85
	RWKV-Med (Ours)	0.69	0.55	24.1	84.95	91.76	93.08

of 150.77 MB, within 0.5 MB of HyenaMed and below MobileNetV3, DeepLabV3+ and EfficientViM. It is faster than SegNeXt and EfficientViM, the most accurate baselines on ISIC2016 and ISIC2017, but it is not the fastest model: Fast-SCNN, DeepLabV3+ and HyenaMed are faster on the GPU, and Fast-SCNN, DeepLabV3+, MobileNetV3 and HyenaMed on the CPU. DeepLabV3+ is faster on both devices although it needs 24.7 times more FLOPs, so FLOPs overstate the speed advantage over convolutional

Table 2 Comparison of mean precision (mPre) and mean recall (mRec) metrics across ISIC datasets

	ISIC2016		ISIC2017		ISIC2018	
Method	mPre	mRec	mPre	mRec	mPre	mRec
U-Net [12]	92.01	94.34	88.39	88.89	86.42	87.97
DeepLabV3+ [36]	93.95	95.25	93.46	89.57	89.98	90.59
MobileNetV3 [37]	92.30	93.78	92.97	89.52	89.76	91.05
Fast-SCNN [38]	93.70	94.25	92.24	88.45	90.81	90.70
SegFormer [23]	94.12	89.90	90.37	88.34	88.22	87.73
SegNeXt [26]	94.23	95.65	93.42	90.41	90.09	91.77
PIDNet [39]	93.47	95.55	90.22	89.69	85.14	89.69
Vision Mamba [5]	94.01	94.80	93.31	90.09	90.65	90.78
SegMAN [30]	92.08	94.33	92.09	89.74	87.44	90.18
EfficientViM [31]	89.78	93.90	93.90	90.27	90.30	90.90
S2Net [40]	94.89	93.75	92.65	87.77	90.38	89.98
HyenaMed [11]	94.54	94.15	91.93	90.10	89.95	90.70
RWKV-Med (Ours)	94.56	95.11	93.46	90.04	91.44	92.11

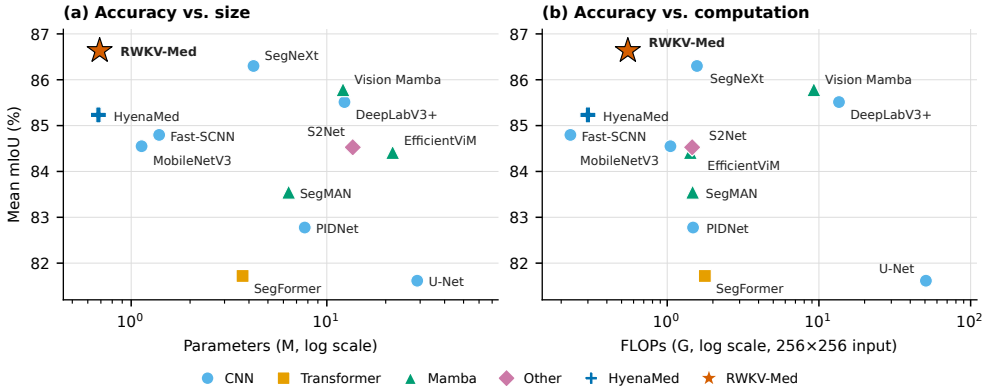

Fig. 2 Mean mIoU over ISIC2016–2018 versus (a) parameters and (b) FLOPs.

networks, probably because the FFT and the bidirectional WKV aggregation are less optimised on current hardware than dense convolutions (we did not profile individual operators).

4.5 Statistical significance

To check that the ISIC2018 differences are not due to chance, we applied a two-sided paired Wilcoxon signed-rank test to the per-image lesion Dice of RWKV-Med and each of five baselines on the 1,000 test images, with Holm correction for five comparisons (Table 4). All adjusted p-values, from 2.96×10^{-16} (DeepLabV3+) to 1.11×10^{-38} (SegNeXt), are below 0.05, including that for HyenaMed (6.64×10^{-22}). The test covers one dataset and a single trained model per method and says nothing about effect size,

Table 3 Inference time and memory usage comparison

Model	GPU (ms)	CPU (ms)	Mem. (MB)
DeepLabV3+	3.38	6.72	193.02
MobileNetV3	5.03	8.14	189.69
Fast-SCNN	1.57	6.64	136.97
SegNeXt	8.00	16.20	159.99
EfficientViM	6.61	15.52	223.84
HyenaMed	3.96	9.77	150.29
RWKV-Med (Ours)	4.31	10.82	150.77

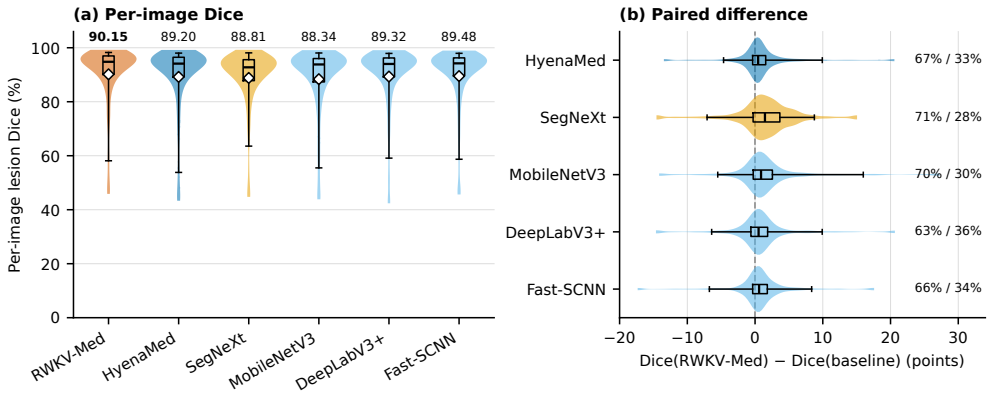

Fig. 3 Per-image lesion Dice on the 1,000 ISIC2018 test images. (a) Distribution for each model (violin; box: quartiles, whiskers: 5th–95th percentiles, line: median, diamond and number: mean). (b) Paired difference between RWKV-Med and each baseline; the percentages give the share of images on which RWKV-Med is better / worse. Violin tails are trimmed at the 2nd and 98th percentiles for display

which is 0.9 to 1.5 points of mDice in Table 1. Fig. 3 shows the distributions: RWKV-Med has the highest mean per-image lesion Dice (90.15%, against 88.34–89.48%) and is better than each baseline on 63% (DeepLabV3+) to 71% (SegNeXt) of the images. The advantage lies in the bulk of the distribution rather than in the tail: its 5th percentile (58.1%) is below that of DeepLabV3+ (59.1%), Fast-SCNN (58.7%) and SegNeXt (63.5%), and 31 of the 1,000 images have a Dice below 0.5 against 32–42 for the baselines, so hard cases remain a limitation shared by all six models. The tests validate that the ISIC2018 advantage over these five baselines is statistically reliable on this test set.

4.6 Cross-dataset generalisation

Table 5 evaluates models trained on ISIC2018 on PH2 without fine-tuning. RWKV-Med obtains the best aAcc, mIoU, mDice and mPre (95.12%, 89.60%, 94.48% and 94.03%), improving on HyenaMed by 1.11 mIoU and 0.63 mDice points and on the strongest baseline, Fast-SCNN, by 0.26 and 0.14 points. SegNeXt generalises worst

Table 4 Paired Wilcoxon signed-rank tests on per-image Dice on ISIC2018, RWKV-Med versus each baseline (Holm p: Holm-adjusted p-value)

RWKV-Med vs.	Raw p	Holm p	Significant
HyenaMed	3.3×10^{-22}	6.6×10^{-22}	Yes
SegNeXt	2.2×10^{-39}	1.1×10^{-38}	Yes
MobileNetV3	3.5×10^{-38}	1.4×10^{-37}	Yes
DeepLabV3+	3.0×10^{-16}	3.0×10^{-16}	Yes
Fast-SCNN	8.3×10^{-24}	2.5×10^{-23}	Yes

Table 5 Cross-dataset test comparison: trained on ISIC2018, tested on PH2 dataset.

Method	aAcc	mIoU	mDice	mPre	mRec
DeepLabV3+	94.86	89.14	94.22	93.53	95.05
Fast-SCNN	94.97	89.34	94.34	93.71	95.07
SegNeXt	93.48	86.48	92.70	91.95	93.63
EfficientViM	94.84	89.15	94.23	93.40	95.28
HyenaMed	94.51	88.49	93.85	93.06	94.84
RWKV-Med (Ours)	95.12	89.60	94.48	94.03	94.98

(86.48% mIoU, 3.12 points below RWKV-Med). The mRec of RWKV-Med (94.98%) is fourth, behind EfficientViM, Fast-SCNN and DeepLabV3+, echoing its lower recall on ISIC2016 and ISIC2017. Because its margins over the three strongest baselines are below 0.5 mIoU points and each model was trained once, these results show that RWKV-Med transfers at least as well as the other lightweight models rather than a clear advantage.

4.7 Ablation study

Table 6 builds RWKV-Med up on ISIC2017 from a plain HyenaPixel baseline (all-HPx encoder, all-MLP decoder) under the protocol of Section 4.2. The SegFormer, baseline, PBRP and ADWP rows follow the ablation of HyenaMed [11]; the two RWKV rows are new and use the RWKV-Med configuration, which differs from that of HyenaMed in the widths of stages 1–2 and the stage-3 embedding dimension, so the RWKV gains include these changes.

The baseline matches SegFormer (80.92% vs. 80.93% mIoU; 91.55% vs. 91.31% aAcc) with 5.7 times fewer parameters (0.653M vs. 3.710M) and 8.7 times fewer FLOPs (0.204G vs. 1.770G). PBRP raises mIoU by 1.10 points to 82.02% for 0.028M parameters and 0.101G FLOPs, and ADWP adds 1.61 points for 0.001M parameters and no extra FLOPs, which makes it the most cost-effective part of the decoder; together they lift mIoU from 80.92% to 83.63%. An RWKV block in stage 1 improves mIoU by 0.64 points to 84.27% for 0.045M parameters and 0.258G FLOPs, and RWKV blocks in stages 1–2 add a further 0.33 points while lowering the parameters to 0.692M and the FLOPs to 0.559G. The smaller gain at stage 2 suggests that the benefit of the locality-biased RWKV mixer is concentrated at the highest resolution; RWKV blocks

Table 6 Ablation study on ISIC2017 (%). ΔmIoU is the change relative to the previous configuration

Configuration	mIoU	ΔmIoU	mDice	aAcc	Par.(M)	GFLOPs
SegFormer	80.93	–	89.26	91.31	3.710	1.770
Baseline	80.92	–	89.24	91.55	0.653	0.204
+ PBRP	82.02	+1.10	89.98	91.96	0.681	0.305
+ PBRP + ADWP	83.63	+1.61	90.95	92.65	0.682	0.305
+ RWKV, stage 1	84.27	+0.64	91.32	93.10	0.727	0.563
+ RWKV, stages 1–2	84.60	+0.33	91.52	93.23	0.692	0.559

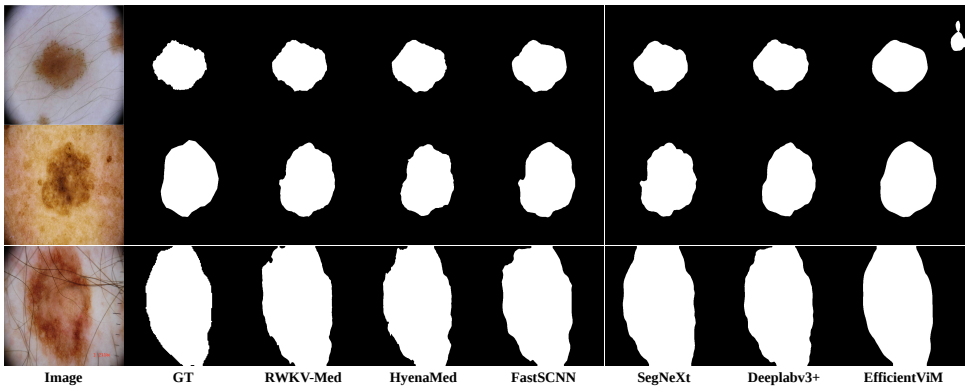

Fig. 4 Segmentation results on ISIC2016–2018 (top to bottom). From left to right: input image, ground truth (GT), RWKV-Med, HyenaMed, and the comparison methods Fast-SCNN, SegNeXt, DeepLabV3+ and EfficientViM

in stages 3 and 4 were not tested. Every step improves all three accuracy metrics, which validates that the decoder modules and the RWKV stages make distinct and additive contributions, and the full model reaches 84.60% mIoU with fewer than 0.7M parameters and 0.56G FLOPs.

4.8 Qualitative results

Fig. 4 shows one example per dataset. All six models localise the lesion and recover its overall shape, so the visible differences are confined to boundaries and isolated false positives. In the first example EfficientViM adds a spurious region in the upper right corner, whereas the other five models produce a single compact region; in the second and third examples the masks of RWKV-Med and HyenaMed follow the ground-truth outline with small boundary deviations and no model shows a clear advantage.

5 Discussion

Why a hybrid helps. The ablation supports the hybrid design: RWKV blocks in the two high-resolution stages add 0.97 mIoU points on ISIC2017, two thirds of it (0.64)

from the first stage alone, which suggests that a locality-biased, distance-decayed aggregation is most useful where the token count is largest. The comparison with HyenaMed points the same way (mIoU +0.84, +0.97 and +2.39 on ISIC2016–2018 and +1.11 on PH2 for 0.01M parameters). The ablation attributes the largest share of the gain to the first RWKV stage, and the changes in width and embedding dimension relative to HyenaMed add only 0.01M parameters.

Scope and future work. RWKV-Med is designed for the accuracy–cost trade-off rather than for the top score on every benchmark. It is the most accurate model in mIoU on ISIC2018 and PH2, within 0.15 points of the best on ISIC2016 and within 0.67 points on ISIC2017, where it uses 31.5 times fewer parameters than EfficientViM. Its recall is slightly below that of the best baselines on ISIC2016, ISIC2017 and PH2, so screening-oriented variants could add a recall-weighted loss. Relative to HyenaMed it adds 0.55G FLOPs in exchange for gains on all four test sets, and its latency (4.31 ms on the GPU and 10.82 ms on the CPU) is higher than that of Fast-SCNN and DeepLabV3+, which are less accurate on ISIC2018; measurement on embedded and mobile hardware is the next step. Each model was trained once, and the paired tests on ISIC2018 show that its gains there are systematic, so repeating them with multiple seeds on the other datasets would extend this evidence. Finally, all experiments use 256×256 dermoscopic images, class-averaged metrics and public benchmarks, and a failure-case study, higher-resolution inputs, more diverse skin tones and prospective clinical validation are directions for future work.

6 Conclusion

We presented RWKV-Med, a lightweight skin lesion segmentation network that uses RWKV blocks in its high-resolution stages and HyenaPixel large-kernel convolution blocks in its low-resolution stages, decoded by an all-MLP head refined with PBRP and ADWP. With 0.69M parameters and 0.55 GFLOPs it is first on all metrics on ISIC2018, second on ISIC2016 and competitive on ISIC2017, and it improves on HyenaMed on all three datasets and on PH2, where it obtains the best mIoU of six models without fine-tuning. The ablation shows that PBRP, ADWP and the RWKV stages raise ISIC2017 mIoU by 1.10, 1.61 and 0.97 points. Its measured latency is higher than that of the fastest baselines, so future work will address inference speed and embedded deployment.

Acknowledgements. The authors thank the School of Computing and Artificial Intelligence, Southwest Jiaotong University, for providing computational resources.

Declarations

- **Funding.** The authors received no specific funding for this work.
- **Competing interests.** The authors declare no conflict of interest.
- **Ethics approval and consent to participate.** Only publicly available, de-identified benchmark datasets were used and no new data were collected from human participants.

- **Data availability.** The ISIC2016, ISIC2017 and ISIC2018 datasets are available from the International Skin Imaging Collaboration archive and the PH2 database from its authors.
- **Code availability.** The code, configurations and trained weights will be made publicly available at https://github.com/zeehydr upon acceptance of the paper.
- **Author contributions.** Zeeshan Haider implemented the RWKV-Med framework and the main experiments, built the training and evaluation pipeline and the ablations, prepared the figures and tables and wrote the initial draft. Haiyu Huang conceived the study and the overall methodology, supervised the project, secured resources, co-wrote the manuscript and led the final revisions. Muqadar Ali contributed ideas, experimental design and result analysis, helped with writing and guided the submission process. Muhammad Adnan Farooq Butt performed the statistical analyses and significance testing, cross-checked the results and baselines, verified the ablation study and edited the manuscript.

References

[1] Naldi, L., Cazzaniga, S., Flohr, C., et al.: The worldwide burden of skin diseases: Lessons from the Global Burden of Disease data. J. Eur. Acad. Dermatol. Venereol. (2026)

[2] Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R.L., Soerjomataram, I., Jemal, A.: Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: Cancer J. Clin. **74**(3), 229–263 (2024)

[3] Mirikharaji, Z., Abhishek, K., Bissoto, A., et al.: A survey on deep learning for skin lesion segmentation. Med. Image Anal. **88** (2023)

[4] Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)

[5] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision Mamba: Efficient visual representation learning with bidirectional state space model. In: Proc. Int. Conf. Mach. Learn. (ICML) (2024)

[6] Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: VMamba: Visual state space model. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 37 (2024)

[7] Peng, B., Alcaide, E., Anthony, Q., *et al.*: RWKV: Reinventing RNNs for the transformer era. In: Findings Assoc. Comput. Linguistics: EMNLP, pp. 14048–14077 (2023)

[8] Duan, Y., Wang, W., Chen, Z., Zhu, X., Lu, L., Lu, T., Qiao, Y., Li, H., Dai, J., Wang, W.: Vision-RWKV: Efficient and scalable visual perception with RWKV-like architectures. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2025)

[9] Poli, M., Massaroli, S., Nguyen, E., Fu, D.Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., Ré, C.: Hyena hierarchy: Towards larger convolutional language models. In: Proc. Int. Conf. Mach. Learn. (ICML), pp. 28043–28078 (2023)

[10] Spravil, J., Houben, S., Behnke, S.: HyenaPixel: Global image context with convolutions. arXiv preprint arXiv:2402.19305 (2024)

[11] Haider, Z., Haiyu, H., Butt, M.A.F., Ali, M.: Hyenamed: Lightweight skin lesion segmentation via convolutional global context modeling. In: Int. Conf. on Frontiers of Eng. and Emerging Tech. (FET), pp. 1–7 (2026)

[12] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), pp. 234–241 (2015)

[13] Valanarasu, J.M.J., Patel, V.M.: UNeXt: MLP-based rapid medical image segmentation network. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), pp. 23–33 (2022)

[14] Ruan, J., Xiang, S., Xie, M., Liu, T., Fu, Y.: MALUNet: A multi-attention and light-weight UNet for skin lesion segmentation. In: Proc. IEEE Int. Conf. Bioinformatics Biomed. (BIBM), pp. 1150–1156 (2022)

[15] Ruan, J., Xie, M., Gao, J., Liu, T., Fu, Y.: EGE-UNet: An efficient group enhanced UNet for skin lesion segmentation. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), pp. 481–490 (2023)

[16] Chen, Y., Yang, G., Dong, X., Zeng, J., Qin, C.: DSNet: A lightweight segmentation model for segmentation of skin cancer lesion regions. IEEE Access (2025)

[17] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: TransUNet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)

[18] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-Unet: Unet-like pure transformer for medical image segmentation. In: Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW), pp. 205–218 (2022)

[19] Dosovitskiy, A., Beyer, L., Kolesnikov, A., *et al.*: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2021)

[20] 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: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp. 10012–10022 (2021)

[21] Mehta, S., Rastegari, M.: MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2022)

[22] Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., Wang, X.: MetaFormer baselines for vision. IEEE Trans. Pattern Anal. Mach. Intell. **46**(2), 896–912 (2024)

[23] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, pp. 12077–12090 (2021)

[24] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770–778 (2016)

[25] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 11976–11986 (2022)

[26] Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., Hu, S.-M.: SegNeXt: Rethinking convolutional attention design for semantic segmentation. In: Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 35, pp. 1140–1156 (2022)

[27] Ding, X., Zhang, X., Han, J., Ding, G.: Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 11963–11975 (2022)

[28] Romero, D.W., Kuzina, A., Bekkers, E.J., Tomczak, J.M., Hoogendoorn, M.: CKConv: Continuous kernel convolution for sequential data. In: Proc. Int. Conf. Learn. Represent. (ICLR) (2022)

[29] Hatamizadeh, A., Kautz, J.: MambaVision: A hybrid Mamba-Transformer vision backbone. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2025)

[30] Fu, Y., Lou, M., Yu, Y.: SegMAN: Omni-scale context modeling with state space models and local attention for semantic segmentation. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2025)

[31] Lee, S., Choi, J., Kim, H.J.: EfficientViM: Efficient vision Mamba with hidden state mixer based state space duality. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2025)

[32] Ma, J., Li, F., Wang, B.: U-Mamba: Enhancing long-range dependency for biomedical image segmentation. arXiv preprint arXiv:2401.04722 (2024)

[33] Ruan, J., Li, J., Xiang, S.: VM-UNet: Vision Mamba UNet for medical image

segmentation. arXiv preprint arXiv:2402.02491 (2024)

[34] Jiang, J., Zhang, J., Liu, W., Gao, M., Hu, X., Xue, Z., Liu, Y., Yan, S.: RWKV-UNet: Improving UNet with long-range cooperation for effective medical image segmentation. arXiv preprint arXiv:2501.08458 (2025)

[35] Ye, H., Tang, F., Zhao, P., Huang, Z., Zhao, D., Bian, M., Zhou, S.K.: U-RWKV: Lightweight medical image segmentation with direction-adaptive RWKV. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI) (2025)

[36] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proc. Eur. Conf. Comput. Vis. (ECCV), pp. 801–818 (2018)

[37] Howard, A., Sandler, M., Chu, G., *et al.*: Searching for MobileNetV3. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), pp. 1314–1324 (2019)

[38] Poudel, R.P.K., Liwicki, S., Cipolla, R.: Fast-SCNN: Fast semantic segmentation network. arXiv preprint arXiv:1902.04502 (2019)

[39] Xu, J., Xiong, Z., Bhattacharyya, S.P.: PIDNet: A real-time semantic segmentation network inspired by PID controllers. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 19529–19539 (2023)

[40] Yang, G., Shi, D., Wang, Y., Zhao, Z.: Single-branch network with self-coaching for real-time semantic segmentation. Pattern Recognition **179**, 113719 (2026)