Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 7, No. 3, July 2026, pp: 1133-1147 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Efficient VGA-Net Modification Using ConvNeXt-Tiny and GATv2 for Retinal Vessel Segmentation

Billie Zandra Widiyanto, Wiharto, and Shaifudin Zuhdi

Department of informatics, Universitas Sebelas Maret, Surakarta, Indonesia.

Corresponding author: Wiharto. (e-mail: wiharto@staff.uns.ac.id), **Authors email**: Billie Zandra Widiyanto (e-mail: billie.zandra@student.uns.ac.id), Shaifudin Zuhdi (e-mail: szuhdi@staff.uns.ac.id)

Abstract **Retinal blood vessel segmentation plays a crucial role in the early detection of ocular diseases such as diabetic retinopathy, glaucoma, and macular degeneration. Existing hybrid architectures, such as VGA-Net, suffer from high computational complexity due to the VGG-16 backbone and limited attention expressiveness due to its static GAT module, yet no prior work has examined replacing both components within a patch-based graph architecture in which backbone feature quality directly conditions graph attention effectiveness. This study aims to improve the computational efficiency and topological modeling of VGA-Net by replacing VGG-16 with ConvNeXt-Tiny and substituting GAT with GATv2. The primary contribution is a 55% parameter reduction through the ConvNeXt-Tiny backbone substitution and improved vessel topology modeling through GATv2's dynamic attention mechanism, which produces fully dynamic attention coefficients per query node. Experiments were conducted on the DRIVE and STARE datasets using a consistent preprocessing pipeline, one-factor-at-a-time hyperparameter tuning, and a unified evaluation protocol across all compared methods. The proposed model achieves the lowest parameter count (5.3M) and GFLOPs (3.2443), with a competitive inference time of 61.00 ms per image, among all compared methods, while achieving competitive performance in sensitivity and topological continuity. On the DRIVE dataset, the model achieved the highest sensitivity of 0.8718 and the highest clDice of 0.8446. On the STARE dataset, the model achieved the highest sensitivity of 0.9383 and the highest clDice of 0.9055. These results demonstrate that the proposed model achieves a favorable efficiency-performance trade-off, leading to sensitivity and topological continuity at the lowest computational cost among all compared methods, at the expense of lower specificity, accuracy, Dice, and MCC relative to certain compared methods.**

Keywords **Retinal blood vessel segmentation; VGA-Net; ConvNeXt; GATv2; Graph Attention Network; Deep Learning.**

I. Introduction

Retinal blood vessel segmentation plays a critical role in the early detection of ocular diseases, including diabetic retinopathy, hypertension, macular degeneration, and glaucoma [1] [2] [3]. Automated analysis of retinal fundus images enables clinicians to detect structural changes in the vasculature before clinical symptoms manifest, substantially improving treatment outcomes [4] [5]. However, retinal vessels are extremely thin, highly branched, and densely packed, with boundaries that are difficult to delineate due to low contrast, uneven illumination, and pathological lesions such as hemorrhages and exudates [6] [7]. Thin capillaries are particularly prone to discontinuity and fragmentation, resulting in topologically incorrect outputs that compromise downstream clinical analysis [8].

Prior work on retinal vessel segmentation can be broadly grouped into five categories, namely CNN-based methods, U-Net variants, graph-based methods, transformer/attention-based methods, and lightweight segmentation methods, each addressing a different aspect of the accuracy-efficiency-topology trade-off.

Early CNN-based approaches, such as the diversified deep convolutional architecture proposed by Tani and Tešić [1]. established the feasibility of automated pixel-wise vessel classification directly from fundus images. U-Net variants have since become the dominant paradigm, demonstrating strong performance through encoder-decoder designs with skip connections [9] [10]. Subsequent enhancements, including residual connections, dense blocks, and multiscale feature aggregation, have further improved segmentation accuracy on standard benchmarks such as DRIVE and STARE [11] [12] [13]. However, these architectures share a fundamental limitation. Their local receptive fields restrict the ability to capture long-range spatial dependencies essential for preserving the topological connectivity of branching vessel structures, resulting in predictions that frequently exhibit discontinuities and missing segments [8] [14]. More recently, transformer/attention-based segmentation

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

networks have emerged to address these limitations directly within the convolutional paradigm. GeGLUNet [15], for instance, introduces attention gates into the skip connections of an encoder-decoder framework, allowing the network to selectively emphasize informative feature regions during reconstruction.

In parallel, graph-based methods have explored relational modeling as an alternative mechanism for capturing long-range topological dependencies. Graph Attention Networks (GATs) extend graph convolution with adaptive attention mechanisms that assign differential weights to neighboring nodes, making them well-suited for modeling complex vessel topology [16] [17]. Motivated by this, Jalali et al [8]. proposed VGA-Net, a hybrid architecture that constructs a spatial patch graph, applies GAT-based relational reasoning, and integrates graph and convolutional features via an AB-FFM fusion module, achieving state-of-the-art accuracy of 97.94% and 98.19% on DRIVE and STARE, respectively. VGA-Net carries two critical limitations. First, its VGG-16 backbone [18] introduces substantial computational overhead with approximately 138 million parameters [19], limiting deployment efficiency in clinical settings. Alotaibi et al. [20] demonstrated that ConvNeXt-Tiny requires substantially fewer parameters and lower GFLOPs than VGG-16, while Zhu et al. [21] achieved mDice improvement of 2.92% on DRIVE. In parallel, a growing body of lightweight segmentation research has targeted parameter efficiency directly, as demonstrated by Han et al. [22], who proposed ConvUNeXt and achieved a competitive Dice score of 0.8230 with only 3.5M parameters. Despite this evidence, no prior work has examined replacing the VGG-16 backbone with ConvNeXt-Tiny within a graph-based hybrid architecture such as VGA-Net.

VGA-Net employs a static GAT attention mechanism, where the attention scoring function applies a linear transformation followed by a non-linearity before joint node feature concatenation. This formulation is mathematically equivalent to a linear function on the input space, making all query nodes share identical attention rankings regardless of input. GATv2 addresses this limitation by fusing the joint node representation prior to applying the non-linearity, producing fully dynamic and query-dependent attention coefficients [23]. Empirically, Le et al. [24] demonstrated that GATv2 outperforms GAT in preserving retinal vessel topological continuity in an artery-vein segmentation task. Beyond retinal imaging, Zhang et al. [25] integrated a multi-scale GATv2 module into a diffusion-based framework for liver vessel segmentation. Their findings demonstrate that GATv2's dynamic attention mechanism is critical for preserving vessel geometry and continuity in complex branching structures. Nevertheless, the substitution of GAT with GATv2 within a patch-based graph architecture specifically designed for retinal vessel segmentation has not been previously investigated.

Motivated by these limitations, this study proposes a modified VGA-Net that replaces VGG-16 with ConvNeXt-Tiny and substitutes the static GAT module with GATv2. These modifications are interdependent, as the backbone determines the quality of the node features the graph attention module subsequently processes, so improving one alone cannot achieve both computational efficiency and topological accuracy. The core pipeline, including graph construction, AB-FFM-based feature fusion, and U-Net-based segmentation, is retained.

The following are the key contributions of this research.

1) As an architectural novelty, we propose a modification of the VGA-Net architecture by replacing the VGG-16 backbone with ConvNeXt-Tiny and substituting the static GAT module with GATv2, which produces fully dynamic attention coefficients per query node for more expressive modeling of topological relationships between retinal blood vessel segments.
2) As an efficiency contribution, the ConvNeXt-Tiny substitution reduces the number of parameters from approximately 11.7 million to 5.3 million (~54.6%) and GFLOPs from 3.3686 to 3.2443, achieving the lowest computational cost among all compared methods, while maintaining the highest sensitivity and clDice at the expense of lower specificity, Dice, and MCC relative to certain compared methods.
3) As an experimental novelty, we systematically tune hyperparameters using a one-factor-at-a-time (OFAT) approach for learning rate, dropout, number of attention heads, loss function, batch size, grid size, and patch size.
4) Complementing this experimental contribution, we conduct a comprehensive ablation study comparing four architecture variants and a comparison against six prior methods using SE, SP, ACC, Dice, clDice, and MCC metrics on the DRIVE and STARE datasets under a unified retraining protocol.

II. Method

Fig. 1 illustrates the overall research workflow, starting from dataset collection, preprocessing, model design using VGA-Net, hyperparameter tuning, model training, and final evaluation.

A. Dataset

The DRIVE dataset comprises 40 color fundus photographs with a resolution of 565×584 pixels, officially partitioned into 20 training and 20 test images, each accompanied by a field-of-view mask and manual

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

expert annotation [26]. In this study, the 20 training images were further subdivided into 16 for training and 4 for validation, while all 20 test images were reserved for final evaluation [27]. The STARE dataset consists of 20 retinal fundus images with a resolution of 700×605 pixels, encompassing diverse pathological conditions including diabetic retinopathy and hypertensive retinopathy [28]. Due to the absence of an official split, images were divided into 12 training, 3 validation, and 5 test images, consistent with comparable segmentation studies [29]. It is acknowledged that the small test set size of STARE may affect the stability of performance estimates. Representative samples of both datasets are shown in Fig. 2.

B. Preprocessing

To address challenges including low contrast, uneven illumination, and imaging noise, a four-stage preprocessing pipeline was applied, consisting of green-channel extraction, background intensity normalization, CLAHE, and unsharp masking. The green channel was selected due to its higher vessel-to-background contrast, and background-intensity normalization reduced boundary artifacts by replacing non-retinal regions with mean retinal intensity. CLAHE was applied with a clip limit of 2.0 and a tile grid size of 8×8 pixels, followed by unsharp masking with Ga Gaussian blur sigma equals 3 and a arpening strength of 0.2. Pixel values were normalized to $[0, 1]$ by dividing by 255, after which images were divided into non-overlapping patches of 64×64 pixels. Data

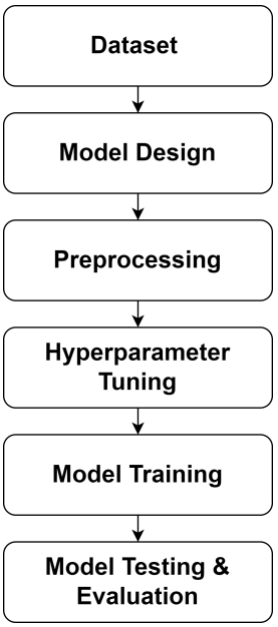


Fig. 1. Research workflow from dataset acquisition through preprocessing, model design, hyperparameter tuning, training, and evaluation.

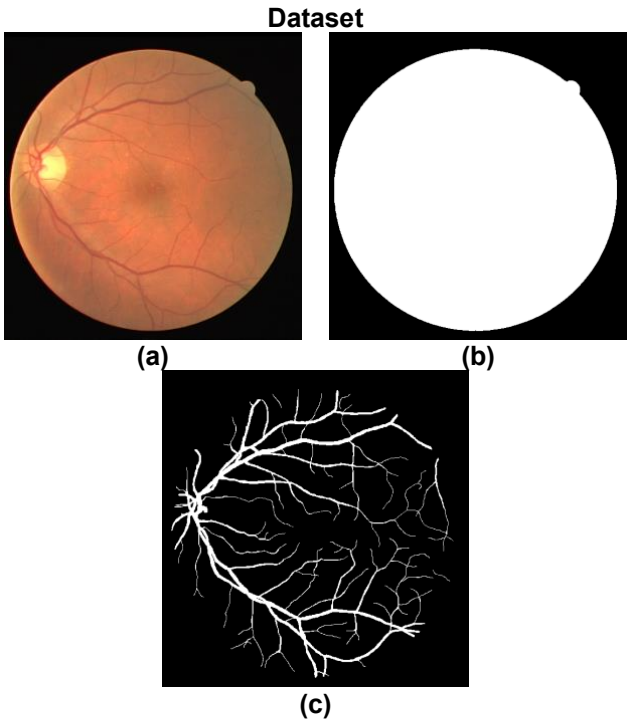


Fig. 2. Sample images from DRIVE and STARE datasets, showing the (a) original image, (b) mask and (c) ground truth.

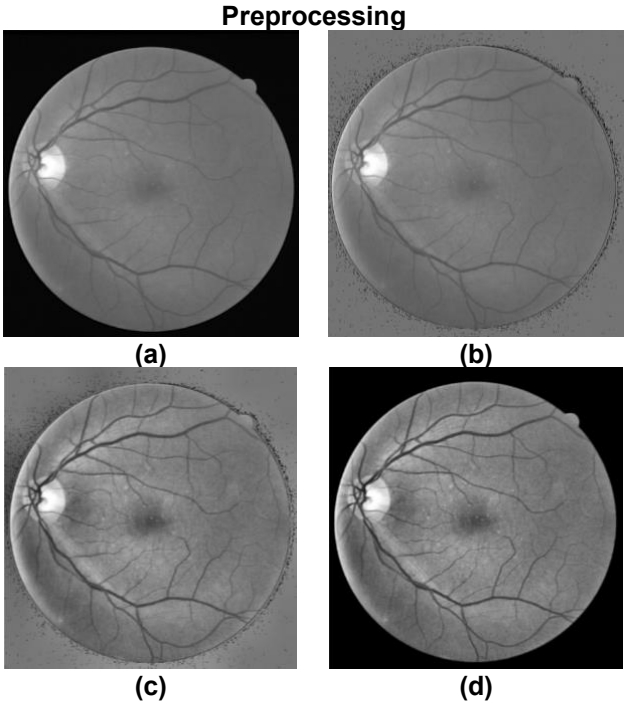


Fig. 3. Preprocessing pipeline, (a) green channel extraction, (b) background normalization, (c) CLAHE with clip limit=2.0 and tile grid=8×8, and (d) unsharp masking with Gaussian blur sigma=3 and strength=0.2, followed by normalization to [0,1].

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

augmentation during training included horizontal flip, vertical flip, rotation at multiples $of\ 90°$, brightness jitter, gamma correction, Gaussian noise, and contrast adjustment, as illustrated in Fig. 3.

C. Model Design

All experiments were conducted on a workstation equipped with an NVIDIA RTX A4000 GPU. This study proposes a modified VGA-Net architecture that incorporates two targeted substitutions: replacing the VGG-16 backbone with ConvNeXt-Tiny and replacing the standard GAT module with GATv2. The remaining pipeline components, including the AB-FFM feature fusion module, HDC encoder blocks, and U-Net decoder structure, are retained from the original VGA-Net [8]. The overall model formulation is expressed as shown in Eq. (1)

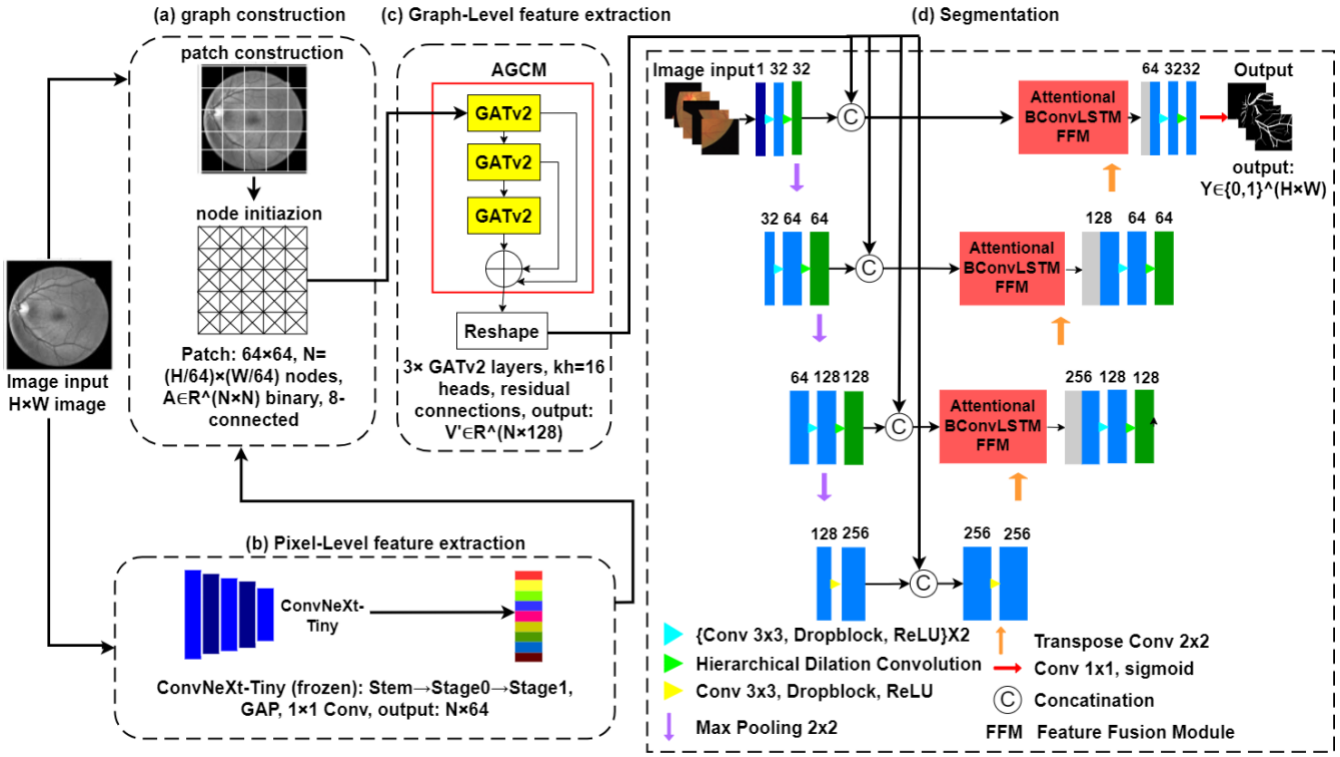


Fig. 4. Architecture of the proposed modified VGA-Net, comprising (a) graph construction, (b) pixel-level feature extraction using ConvNeXt-Tiny, (c) graph-level feature extraction using GATv2, and (d) segmentation via AB-FFM and U-Net decoder.

$$\hat{Y} = \mathcal{D}(\mathcal{F}_{AB-FFM}(\mathcal{G}_{GATv2}(\mathcal{E}_{ConvNeXt}(I)))) \quad (1)$$

where I is the preprocessed retinal image, $I \in \mathbb{R}^{H\times W\times C}$, and $\hat{Y} \in \{0,1\}^{H\times W}$ is the predicted binary segmentation map, $\mathcal{E}_{ConvNeXt}$ denotes the ConvNeXt-Tiny encoder function, $\mathcal{G}_{GATv2}$ the graph attention module, $\mathcal{F}_{AB-FFM}$, the attentional bidirectional feature fusion module, and $\mathcal{D}$ the U-Net decoder. The architecture operates through four sequential stages, as illustrated in Fig. 4.

The four sequential stages of the architecture are described in detail as follows:

1. Graph Construction

Each preprocessed image $X \in \mathbb{R}^{B\times C\times H\times W}$ is partitioned into non-overlapping patches of 64×64 pixels, yielding N nodes arranged in a two-dimensional spatial grid. Each node is connected to its eight spatial neighbors under an 8-connected neighborhood scheme, including self-loops. Edge weights are binary with no distance-based weighting applied. The resulting adjacency matrix A is precomputed once before training and shared identically across all samples, where $Aij = 1$ if node vj is a direct neighbor of node vi. Adjacency normalization is handled implicitly within GATv2 via softmax over neighboring attention scores. The symmetric structure of A defines an undirected graph. The Grid Size and Patch Size jointly determine the cropped region size used during training, computed as Grid Size × Patch Size

2. Pixel-Level Feature Extraction

Each image patch is encoded through a frozen ConvNeXt-Tiny backbone pretrained on ImageNet [30]. Feature representations are extracted from three stages of the backbone, namely the Stem, Stage 0, and Stage 1 outputs, implementing a multi-scale side-output strategy. Each representation is reduced to 16 channels via 1×1 convolution, flattened through

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

Global Average Pooling, concatenated, and projected to a final node feature dimension of 64. Freezing the ConvNeXt-Tiny weights during training confines optimization to the projection layers only, reducing training complexity while preserving the representational quality of ImageNet-pretrained features [30]. This decision is justified by the limited dataset size, which makes full fine-tuning susceptible to overfitting given the large number of backbone parameters, and is consistent with prior work demonstrating that freezing the majority of pretrained parameters can yield performance comparable to full fine-tuning at substantially lower computational cost [31]. One ConvNeXt block computes a residual transformation as shown in Eq. (2) [30]

$$Y = X + F_{res}(X) \tag{2}$$

where X is the block input, Y is the block output, and $F_{res}(X)$ is the residual transformation function defined as shown in Eq. (3) and Eq. (4) [30]

$$\mathrm{C} = LN(DWConv_{7x7}(X)) \tag{3}$$

$$F_{res}(X) = PWConv_{1x1}(\emptyset(PWConv_{1x1}(\mathrm{C}))) \tag{4}$$

where $WConv_{7x7}$ denotes depthwise convolution with 7×7 kernel, LN denotes Layer Normalization, $PWConv_{1x1}$ denotes pointwise convolution, and $\emptyset$ denotes the Gaussian Error Linear Unit (GELU) activation function. The feature extraction mapping is expressed as shown in Eq. (5) [32]

$$F_{enc} = f_{enc}(X), \qquad F_{enc} \in \mathbb{R}^{B \times C_f \times H_f \times W_f} \tag{5}$$

where f_{enc} denotes the ConvNeXt-Tiny encoder function, and C_f, H_f, and W_f denote the number of channels and spatial dimensions of the resulting feature map.

3. Graph-Level Feature Extraction

An Attentive Graph Convolution Module (AGCM) based on GATv2 processes the node feature set $V = \{v_1, \ldots, v_N\}$. together with the precomputed adjacency matrix to produce topology-aware node representations. The distinction between GAT and GATv2 lies in the order of operations in the attention scoring function, as shown in Eq. (6) for GAT and Eq. (7) for GATv2 [16] [23]

$$e_{ij} = LeakyReLU(\alpha^{\top}[W\overrightarrow{h'_i} || W\overrightarrow{h'_j}]) \tag{6}$$

$$e_{ij} = \alpha^{\top}\, LeakyReLU([W\overrightarrow{h'_i} || W\overrightarrow{h'_j}]) \tag{7}$$

where $\alpha^{\top}$ is the learnable attention parameter vector, $||$ denotes vector concatenation, Wh'_i and Wh'_j are the linearly transformed features of nodes i and j, $LeakyReLU$ is the non-linear activation function, and e_{ij} Is the resulting attention score between node i and node j. The raw attention scores e_{ij} computed in Eq. (7) are then normalized across neighbors via softmax to obtain the final attention, as shown in Eq. (8) where αij denotes the normalized attention coefficient between the node i and node j.

$$\alpha_{ij} = Softmax(e_{ij}) \tag{8}$$

To enhance representational capacity, these attention coefficients are extended to multi-head attention with $k_h = 16$ heads is applied in parallel. The attention function ***Att*** is defined as shown in Eq. (9), where $Q, K, V \in \mathbb{R}^{N \times d}$ are the query, key, and value matrices, N is the number of nodes, d is the per-head feature dimension, and $D_{feature} = \sqrt{d}$ is the scaling factor.

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D_{feature}}}\right)V \tag{9}$$

Each attention head b then applies this function using learned projection matrices W_q^b, W_k^b, $and\ W_v^b$ as shown in Eq. (10), where $head_b$ denotes the output of the b attention head, vi and vj denote the feature vectors of the node i and node j respectively, and W_q^b, W_k^b, W_v^b are the learned query, key, and value projection matrices for $head_b$.

$$head_b(v_i, v_j) = Att(W_q^b v_i, W_k^b v_j, W_v^b v_j) \tag{10}$$

The outputs of all heads are concatenated and linearly projected as shown in Eq. (11), where $W_o \in \mathbb{R}^{(k_h \cdot d) \times d_{out}}$ is the learnable output projection matrix, $k_h = 16$, $d = 8$, and $d_{out} = 128$ is the final node feature dimension.

$$MH(v_i, v_j) = Concat(head_1, \ldots, head_b)W_o \tag{11}$$

Node representations are subsequently aggregated over the local neighborhood N_i, where N_i denotes the set of spatial neighbors of node i including self-loops, and stabilized through BatchNorm and ReLU as shown in Eq. (12)

$$v'_i = BN\left(ReLU\left(\sum_{j \in N_i} MH(v_i, v_j)\right)\right) \tag{12}$$

Three GATv2 layers with residual connections between them produce the final graph feature set $V' = \{v'_1, \ldots, v'_N\}$.

4. Segmentation

The segmentation stage reconstructs the binary vessel map by integrating graph features and convolutional features through a U-Net encoder-decoder with AB-FFM. Each encoder level I is formulated as shown in Eq. (13), where X_l denotes the encoder feature map at level l and X_{l-1} is the input from the previous level, and HDC denotes the Hierarchical Dilated Convolution module that expands the receptive field at each encoder stage.

$$X_l = maxpool(HDC(\{Conv_{3\times3} \circ DropBlock \circ ReLU\}^{\times 2}(X_{l-1}))) \tag{13}$$

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

Algorithm 1: Training & Inference Pipeline of Modified VGA-Net
Input: I (retinal images), G (ground truth), Split (Train/Val/Test)
Output: Ŷ (binary mask), Metrics {SE, SP, ACC, Dice, clDice, MCC}
// --- Phase 1: Preprocessing & Graph Construction ---
1. Extract green channel; apply CLAHE (clip=2.0, tile=8×8) and unsharp masking (sigma=3, strength=0.2); normalize to [0,1]
2. Partition image into non-overlapping patches of size 64×64 (N nodes)
3. Construct adjacency matrix A (8-connected neighbors + self-loops); precompute once

// --- Phase 2: Hyperparameter Tuning (OFAT) ---
4. Tune one-factor-at-a-time for {lr, dropout, attention heads, loss, batch_size}
5. Select optimal configuration: heads=16, GAT_layers=3, DropBlock=0.05, lr=1e-4, batch_size=1, loss=BCE+Tversky

// --- Phase 3: Training (100 epochs) ---
6. *for* epoch = 1 to 100 *do*
7. Apply data augmentation (flip, rotation 90°, brightness, gamma, noise, contrast)
// Pixel-Level Feature Extraction (ConvNeXt-Tiny)
8. Extract multi-scale features from Stem, Stage 0, Stage 1 of ConvNeXt-Tiny
9. Reduce to 16 channels via 1×1 Conv, GAP, concatenation, project to d=64 (node features)
// Graph-Level Feature Extraction (GATv2)
10. Compute dynamic attention scores: $e_{ij} = \alpha^{\top} LeakyReLU([\overrightarrow{Wh'_i}||\overrightarrow{Wh'_j}])$[Eq. 7]
11. Apply softmax normalization and aggregate over 8 neighbors [Eq. 8–12]
12. Pass through 3 GATv2 layers (16 heads, residual connections) → V'
// Segmentation (U-Net + AB-FFM)
13. Project V' to spatial domain; fuse with encoder skip connections via AB-FFM [Eq. 15]
14. Decode via U-Net (HDC blocks + DropBlock) [Eq. 13–16]
15. Generate output logits via 1×1 Conv + Sigmoid [Eq. 17]
// Optimization
16. Compute BCE+Tversky loss; update trainable parameters via Adam [Eq. 18–20]
17. Evaluate validation loss each epoch; save best checkpoint
18. *end for*

// --- Phase 4: Inference & Evaluation ---
19. Load best checkpoint; forward pass on test set (stride=64, without TTA)
20. Binarize output: Ŷ = **1**[sigmoid(output) ≥ 0.5]
21. Compute TP, TN, FP, FN; return SE, SP, ACC, Dice, clDice, MCC [Eq. 21–30]

A bottleneck block at 256 channels processes the deepest encoder features without spatial downsampling, as shown in Eq. (14), where X_3 is the output of the third encoder level and H_b and W_b denote the spatial dimensions at the bottleneck.

$$Z_b = \{Conv_{3\times3} \circ DropBlock \circ ReLU\}^{\times 2}(X_3); \quad Z_b \in \mathbb{R}^{256\times H_b \times W_b} \tag{14}$$

At each decoder level l, the graph feature set V' is projected into the spatial domain and fused with decoder and encoder features through AB-FFM as shown in Eq. (15), where Dl denotes the upsampled decoder feature at level l, El denotes the encoder skip connection feature at level l, and Gl denotes the spatially projected GATv2 graph feature at level l.

$$F_l = F_{AB-FFM}(D_l||E_l||G_l) \tag{15}$$

The decoder progressively upsamples through channel dimensions 256→128→64→32 as shown in Eq. (16), where D_{l-1} is the decoded feature map at level $l-1$ and $UpConv$ denotes the 2×2 transposed convolution operation for spatial upsampling.

$$D_{l-1} = \{Conv_{3\times3} \circ DropBlock \circ ReLU\}^{\times 2}(UpConv(F_l)) \tag{16}$$

The final segmentation map is produced by a 1×1 convolution followed by sigmoid activation as shown in Eq. (17), where D_o is the final decoder output feature map.

$$\hat{Y} = sigmoid(Conv_{1\times1}(D_o)); \quad \hat{Y} \in \{0,1\}^{H\times W} \tag{17}$$

HDC modules expand the receptive field without reducing spatial resolution, while DropBlock regularization mitigates overfitting [8]. Algorithm 1 summarizes the complete training and inference pipeline of the proposed modified VGA-Net, encompassing all stages from preprocessing and graph construction through model optimization and final performance evaluation.

D. Hyperparameter Tuning

Hyperparameter tuning was conducted using a one-factor-at-a-time (OFAT) approach, in which each parameter is evaluated independently while all others

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

are held at their baseline values, as summarized in Table 1. The learning rate remained unchanged at 1.00×10^{-4} throughout the tuning process, as alternative values consistently resulted in unstable convergence. The Adam optimizer was selected based on its stable convergence behavior compared to AdamW and SGD, updating model parameters using bias-corrected first and second moment estimates as shown in Eq. (18), Eq. (19), and Eq. (20) [33].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (18)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (19)$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \hat{m}_t / \left(\sqrt{\hat{v}_t} + \varepsilon\right) \quad (20)$$

where m_t and v_t denote the first and second moment estimates at timestep t, β_1 and β_2 are the exponential decay rates for the first and second moments, g_t is the gradient of the loss with respect to the parameters at timestep t, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected moment estimates, α is the learning rate, θ_t is the parameter vector at timestep t, and ε is a small constant for numerical stability. The loss function was evaluated across Dice loss, BCE loss, BCE+Dice, and BCE+Tversky, with the latter designed to penalize false negatives more aggressively and improve sensitivity toward thin vessel structures. While OFAT does not capture interaction effects among parameters, more systematic approaches, such as Bayesian optimization, were not applied due to the computational constraints of the graph-based pipeline and limited dataset size.

Table 1. Comparison of baseline and optimized hyperparameter configurations obtained through OFAT tuning based on combined validation performance across DRIVE and STARE.

Hyperparameter	Baseline	Optimized
Learning Rate	1.00×10^{-4}	1.00×10^{-4}
DropBlock Probability	0.15	0.05
DropBlock Block Size (pixels)	7	7
GAT Dropout	0.10	0.30
Attention Heads	8	16
GAT Layers	2	3
Optimizer	Adam	Adam
Loss Function	Dice	0.5 BCE + 0.5 Tversky
Batch Size	2	1
Grid Size (divisions)	6	8
Patch Size (pixels)	48	64

E. Model Evaluation

Binary segmentation masks were obtained by applying a fixed threshold of 0.5 to the sigmoid output of the model. Performance was evaluated using six metrics derived from the binary confusion matrix, categorizing each pixel prediction into True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [34]. Accuracy (ACC) measures the proportion of correctly classified pixels over all pixels [35] [36] as defined in Eq. (21)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (21)$$

Sensitivity (SE) quantifies the model's ability to detect actual vessel pixels, critical for capturing thin vessels prone to being missed [35] [36] as defined in Eq. (22)

$$\mathrm{SE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (22)$$

Specificity (SP) quantifies the model's ability to correctly identify background pixels [35] [36] as defined in Eq. (23)

$$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \quad (23)$$

Dice Score (Dice) measures spatial overlap between predicted and ground truth masks, penalizing both false positives and false negatives equally [36], as defined in Eq. (24)

$$\mathrm{Dice} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \quad (24)$$

Centerline Dice (clDice) evaluates topological continuity by measuring overlap between skeletonized centerlines of predicted and ground truth masks [34], as defined in Eq. (25).

$$\mathrm{clDice} = 2\frac{\left(\frac{|\mathrm{Skel(P)} \cap \mathrm{D}|}{|\mathrm{Skel(P)}|}\right) \times \left(\frac{|\mathrm{Skel(D)} \cap \mathrm{P}|}{|\mathrm{Skel(D)}|}\right)}{\left(\frac{|\mathrm{Skel(P)} \cap \mathrm{D}|}{|\mathrm{Skel(P)}|}\right) + \left(\frac{|\mathrm{Skel(D)} \cap \mathrm{P}|}{|\mathrm{Skel(D)}|}\right)} \quad (25)$$

where P denotes the predicted mask, D denotes the ground truth mask, and Skel(·) denotes the skeletonization operator. Matthews Correlation Coefficient (MCC) is a balanced metric robust to class imbalance, ranging from -1 to $+1$ where $+1$ indicates perfect prediction [37] [38]. as defined in Eq. (26)

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})\ (\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \quad (26)$$

III. Result

A. Hyperparameter Tuning

Hyperparameter tuning was performed using a one-factor-at-a-time (OFAT) strategy, where each parameter was evaluated independently while others were fixed at baseline values (Table 1). Parameter selection was guided by minimizing validation loss while maintaining balanced performance across Dice, clDice, sensitivity (SE), and MCC. The tuning process consistently improved validation performance across all stages, leading to the optimized configuration

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

Table 2. Ablation study results comparing four architecture variants on the DRIVE and STARE datasets, evaluated across inference time, and segmentation performance metrics.

Method	Dataset	Inference time (ms/image)	SP	SE	ACC	Dice	clDice	MCC
VGG-16 + GAT	DRIVE	68.60	0.9784	0.8569	0.9677	0.8224	0.8421	0.8060
VGG-16 + GATv2	DRIVE	68.48	0.9781	0.8566	0.9674	0.8210	**0.8451**	0.8045
ConvNeXt-Tiny + GAT	DRIVE	61.07	**0.9822**	0.8479	**0.9681**	**0.8225**	0.8377	0.8060
ConvNeXt-Tiny + GATv2	DRIVE	**61.00**	0.9762	**0.8718**	0.9671	0.8220	0.8446	**0.8061**
VGG-16 + GAT	STARE	84.09	0.9739	0.9310	0.9704	0.8344	0.9038	0.8242
VGG-16 + GATv2	STARE	84.27	0.9739	**0.9434**	0.9715	0.8409	**0.9076**	0.8319
ConvNeXt-Tiny + GAT	STARE	**74.77**	0.9727	0.9407	0.9701	0.8344	0.9023	0.8251
ConvNeXt-Tiny + GATv2	STARE	74.93	**0.9764**	0.9383	**0.9733**	**0.8492**	0.9055	**0.8397**

summarized in Table 1. Key changes from the baseline include an increase in attention heads from 8 to 16, an increase in GAT layers from 2 to 3, a reduction in DropBlock probability from 0.15 to 0.05, an increase in GAT dropout from 0.1 to 0.30, a reduction in batch size from 2 to 1, an increase in grid size from 6 to 8, an increase in patch size from 48 to 64 pixels, and a change in loss function from Dice to BCE+Tversky. Overall, the selected configuration prioritizes sensitivity and topological continuity while maintaining stable convergence under the small dataset constraint.

B. Ablation Study

An ablation study was conducted to evaluate the individual and combined contributions of the two proposed modifications across four architecture variants on the DRIVE and STARE datasets under identical training settings, as summarized in Table 2. On the DRIVE dataset, replacing VGG-16 with ConvNeXt-Tiny reduced the parameter count from 11,734,807 to 5,326,567 (−54.6%) and GFLOPs from 3.3686 to 3.2443 (−3.7%), while decreasing inference time from 68.60 ms to 61.07 ms per image (−11.1%), without meaningful degradation in segmentation performance. Substituting GAT with GATv2 in the ConvNeXt-Tiny backbone configuration produced the proposed model (ConvNeXt-Tiny+GATv2), which achieved an SE of 0.8718 (+0.0149 over VGG-16+GAT) and a clDice of 0.8446 (+0.0025), while recording a marginal decrease in SP (−0.0022), ACC (−0.0006), and Dice (−0.0004) relative to the VGG-16+GAT baseline. On the STARE dataset, the proposed model achieved an SE of 0.9383 (+0.0073 over VGG-16+GAT) and a clDice of 0.9055 (+0.0017), alongside improvements in SP (+0.0025), ACC (+0.0029), Dice (+0.0148), and MCC (+0.0155), with inference time reduced from 84.09 ms to 74.93 ms (−10.9%). ConvNeXt-Tiny-based variants exhibited more stable convergence across both datasets. Small differences in Dice and clDice across variants should be interpreted as indicative trends rather than statistically confirmed conclusions

IV. Discussion

A. Ablation Analysis

The ablation study was designed to quantify the individual and combined contributions of the two proposed architectural modifications relative to the VGG-16+GAT baseline. The substitution of VGG-16 with ConvNeXt-Tiny produced a substantial reduction in parameter count from approximately 11.7 million to 5.3 million and GFLOPs from 3.3686 to 3.2443 across both GAT and GATv2 variants (Table 3) This reduction did not result in meaningful degradation of segmentation performance, indicating sufficient representational capacity despite the lower parameter count. Furthermore, ConvNeXt-Tiny-based variants exhibited more stable convergence across both datasets, as reflected in consistent reductions in inference time from 68.60 ms to 61.07 ms on DRIVE and from 84.09 ms to 74.93 ms on STARE. Substitution of GAT with GATv2 produced consistent improvements in clDice across both datasets and backbone configurations, indicating more accurate topological continuity modeling. ConvNeXt-Tiny+GATv2 was selected as the optimal configuration based on fulfillment of both primary study objectives. Although VGG-16+GATv2 yields marginally higher SE on STARE and the highest clDice on that dataset, these

Table 3. Comparison of total parameters and computational complexity (GFLOPs) among the four architecture variants, calculated using a fixed input size of 64×64×1 per patch.

Method	Parameters	GFLOPs
VGG-16 + GAT	11,734,807	3.3686
VGG-16 + GATv2	11,735,191	3.3686
ConvNeXt-Tiny + GAT	**5,326,567**	**3.2443**
ConvNeXt-Tiny + GATv2	5,326,951	**3.2443**

Note: ConvNeXt-Tiny-based variants achieve a 54.6% reduction in parameters and a 3.7% reduction in GFLOPs relative to VGG-16 variants.

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

gains are accompanied by more than twice the parameter count, higher GFLOPs, and longer inference time. However, lower specificity across GATv2-based variants warrants caution in clinical interpretation. Small differences in Dice and clDice across variants should be interpreted as indicative trends rather than statistically confirmed conclusions.

B. Comparison

To ensure comparability, all six baseline models were implemented using official public source codes and retrained under identical preprocessing, train-test split (DRIVE: 20/20, STARE: 15/5), and hardware conditions (NVIDIA RTX A4000). Training hyperparameters were unified following the proposed method's tuning configuration rather than each baseline's original publication. While this ensures a consistent evaluation protocol, unified hyperparameters may not represent optimal settings for each individual baseline, and performance differences should be interpreted accordingly.

As shown in Table 4 and Table 5 the proposed method achieves the highest sensitivity and clDice on both datasets among all compared methods. On the DRIVE dataset, the proposed method achieved an SE of 0.8718, surpassing VGA-Net by +0.0149 and IMFF-Net by +0.0827, and a clDice of 0.8446, surpassing VGA-Net by +0.0025 and IMFF-Net by +0.0295. On the STARE dataset, the proposed method achieved an SE of 0.9383, surpassing VGA-Net by +0.0073 and IMFF-Net by +0.0220, and a clDice of 0.9055, surpassing VGA-Net by +0.0017 and IMFF-Net by +0.0024. However, the proposed method records lower SP, ACC, Dice, and MCC relative to certain baselines on both datasets. On DRIVE, SP is 0.9762 (−0.0022 vs

Table 4. Comparison of the proposed method with prior works on the DRIVE dataset across six segmentation performance metrics.

Model	Year	Inference time (ms/image)	SP	SE	ACC	Dice	clDice	MCC
MFI-Net [39]	2022	164.56	0.9860	0.7865	0.9686	0.8132	0.8035	0.8035
BCU-Net [40]	2023	237.14	0.9829	0.8248	0.9691	**0.8228**	0.8297	**0.8066**
IMFF-Net [29]	2024	**54.54**	**0.9870**	0.7891	**0.9696**	0.8190	0.8151	0.8040
VGA-Net [8]	2024	68.60	0.9784	0.8569	0.9677	0.8224	0.8421	0.8060
DG-Net [41]	2025	115.74	0.9849	0.8081	0.9694	0.8214	0.8215	0.8057
GeGLUNet [15]	2026	69.54	0.9853	0.8023	0.9693	0.8196	0.8153	0.8040
Proposed Method	2026	61.00	0.9762	**0.8718**	0.9671	0.8220	**0.8446**	0.8061

Table 5. Comparison of the proposed method with prior works on the STARE dataset across six segmentation performance metrics.

Model	Year	Inference time (ms/image)	SP	SE	ACC	Dice	clDice	MCC
MFI-Net [39]	2022	204.02	0.9811	0.8955	0.9742	0.8474	0.8968	0.8353
BCU-Net [40]	2023	295.70	0.9780	0.9218	0.9735	0.8475	0.9054	0.8368
IMFF-Net [29]	2024	**66.93**	0.9799	0.9163	0.9748	**0.8533**	0.9031	**0.8424**
VGA-Net [8]	2024	84.09	0.9739	0.9310	0.9704	0.8344	0.9038	0.8242
DG-Net [41]	2025	143.45	0.9785	0.9231	0.9741	0.8507	0.9052	0.8401
GeGLUNet [15]	2026	86.00	**0.9830**	0.8861	**0.9753**	0.8514	0.8991	0.8392
Proposed Method	2026	74.93	0.9764	**0.9383**	0.9733	0.8492	**0.9055**	0.8397

Note: All baseline methods in Table 4 and Table 5 were reimplemented using their official public source codes and retrained under identical preprocessing, train-test split, and hardware conditions. SP: Specificity; SE: Sensitivity; ACC: Accuracy; Dice: Dice Score; clDice: Centerline Dice; MCC: Matthews Correlation Coefficient. Bold indicates the best value and underline indicates the second-best value for each metric.

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

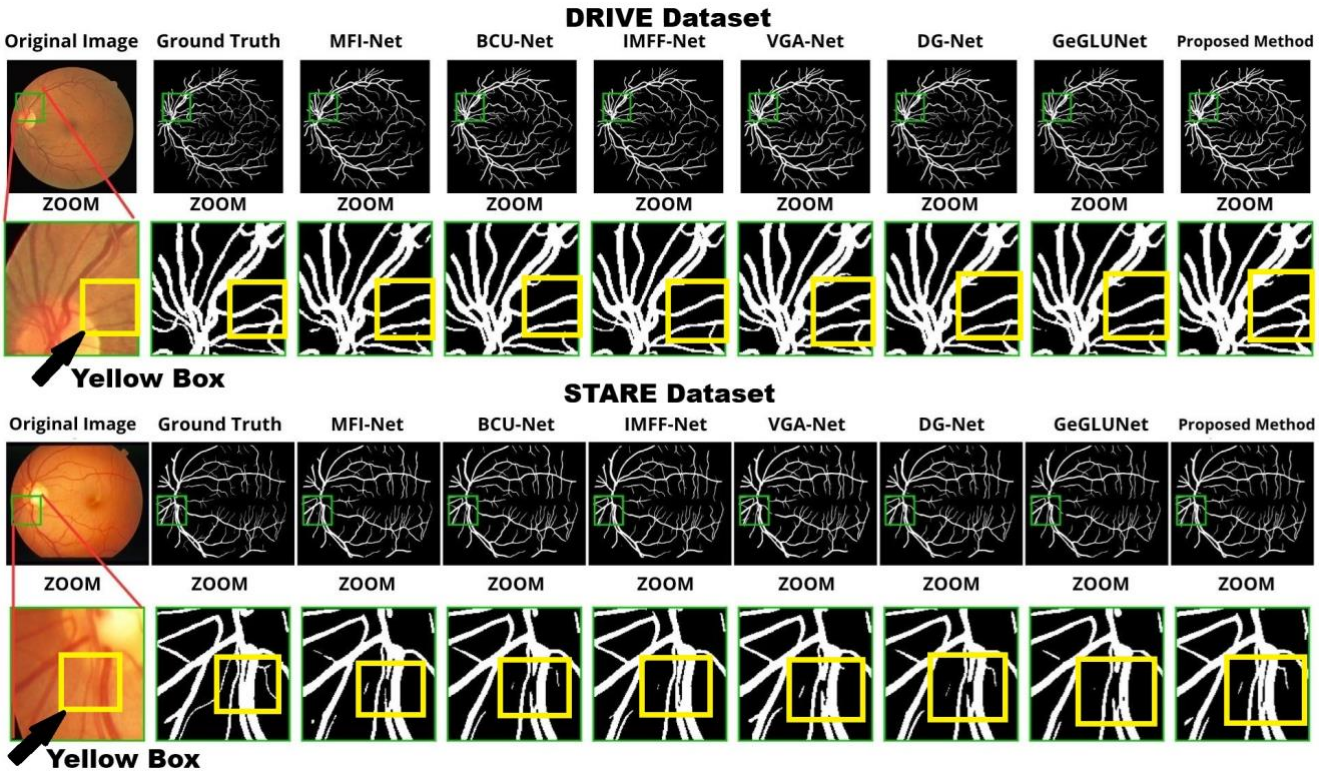


Fig. 5. Retinal vessel segmentation results of the proposed method and prior segmentation networks on the DRIVE and STARE datasets. Green boxes and red line indicate zoomed area. Yellow boxes highlight thin vessel regions where the proposed method preserves continuous vessel structures, whereas compared methods exhibit discontinuous or broken segments.

VGA-Net, −0.0108 vs IMFF-Net), and Dice is 0.8220 (−0.0004 vs VGA-Net, −0.0030 vs IMFF-Net). On STARE, SP is 0.9764 (+0.0025 vs VGA-Net, −0.0035 vs IMFF-Net), and Dice is 0.8492 (+0.0148 vs VGA-Net, −0.0041 vs IMFF-Net).

Table 4, Table 5, and Table 6 should be interpreted jointly, as Table 4 and Table 5 report segmentation performance while Table 6 reports the computational cost required to achieve it. As reported in Table 6, the proposed method achieves the lowest parameter count (5,326,951) and GFLOPs (3.2443) among all compared methods, representing reductions of 54.6% in parameters and 3.7% in GFLOPs relative to VGA-Net, and 84.6% in parameters relative to IMFF-Net (34,684,354). Inference time is 61.00 ms per image on DRIVE and 74.93 ms on STARE, which is faster than VGA-Net (68.60 ms and 84.09 ms, respectively) by 7.60 ms and 9.16 ms, though slightly slower than IMFF-Net (54.54 ms and 66.93 ms) by 6.46 ms and 8.00 ms. The relatively higher latency compared to IMFF-Net reflects the sequential overhead of the graph-based pipeline rather than parameter inefficiency. The consistent advantage in sensitivity and clDice across both datasets indicates that GATv2-based inter-patch reasoning generalizes effectively under diverse pathological conditions, including the more varied pathological presentation of STARE relative to DRIVE.

Lower specificity, Dice, and MCC relative to certain baselines reflect mechanisms that explicitly suppress background noise or correct class imbalance, which the proposed pipeline does not incorporate. IMFF-Net [29], the strongest rival in specificity and accuracy on both datasets, employs squeeze-and-excitation pooling (APF) and weighted encoder-decoder fusion (UDFF) to suppress noise while preserving boundaries. GeGLUNet [15], highest in specificity on STARE, gates encoder features at each skip connection before decoding, filtering background signal propagation, reinforced by edge-aware and contrastive loss. BCU-Net [39], marginally ahead in Dice and MCC on DRIVE, bridges ConvNeXt and U-Net branches through a Multilabel Recall Loss module that directly corrects class imbalance. MFI-Net [40] and DG-Net [41] similarly trade off connectivity recovery for background discrimination through multi-scale interactions and directional convolutions, respectively. In contrast, GATv2's inter-patch propagation connects high-confidence vessel regions to ambiguous neighbors via 8-connected adjacency, enabling the proposed method's sensitivity and clDice advantage while

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

simultaneously allowing background-adjacent signal to leak into neighboring patches and lower specificity.

This trade-off carries different clinical weight. False negatives risk missing early disease signs such as capillary dropout in diabetic retinopathy, while false positives risk misclassifying hemorrhages or exudates as vessels, distorting density or tortuosity measurements in hypertensive retinopathy grading. The proposed method's sensitivity-prioritizing behavior suits screening contexts where missing a vessel carries a greater risk than an occasional false positive, whereas higher specificity of IMFF-Net or GeGLUNet may be preferable for precise quantitative measurement workflows. Qualitative results on DRIVE and STARE (Fig. 5.) confirm this topological continuity advantage in thin and branching vessel regions.

Table 6. Comparison of parameters and GFLOPs between the proposed method and prior methods.

Model	Parameters	GFLOPs
MFI-Net [39]	7,163,952	6.3456
BCU-Net [40]	102,305,314	7.0916
IMFF-Net [29]	34,684,354	3.7874
VGA-Net [8]	11,734,807	3.3686
DG-Net [41]	13,056,307	4.4659
GeGLUNet [15]	59,636,261	5.9484
Proposed Method	**5,326,951**	**3.2443**

Note: Parameters and GFLOPs were calculated under identical input resolution (64×64×1), software framework, and hardware conditions (NVIDIA RTX A4000) for all compared methods.

V. Conclusion

This study proposes a modified VGA-Net that replaces VGG-16 with ConvNeXt-Tiny and substitutes the static GAT with GATv2. Experimental results on DRIVE and STARE demonstrate that these modifications improve both computational efficiency and topological modeling. The proposed model achieves the lowest parameter count (5.3M) and GFLOPs (3.2443) among all compared methods, representing a 54.6% parameter reduction relative to VGA-Net, with inference time of 61.00 ms on DRIVE and 74.93 ms on STARE, faster than VGA-Net by 7.60 ms and 9.16 ms, respectively. On DRIVE, the model achieves the highest sensitivity (0.8718) and clDice (0.8446). On STARE, the highest sensitivity (0.9383) and clDice (0.9055). However, specificity, accuracy, Dice, and MCC are lower than those of certain compared methods on both datasets, reflecting a graph-based design that prioritizes vessel connectivity over background suppression through BCE+Tversky loss and large-patch training. The proposed model is therefore more suitable for screening applications prioritizing sensitivity, while methods with higher specificity, such as IMFF-Net or GeGLUNet, may be preferable for quantitative clinical measurements.

Limitations remain in detecting elongated thin vessels within homogeneous background regions. All experiments used a fixed random seed without repeated runs, leaving performance variance unquantified, particularly for STARE, given its 5-image test set. Future work may include repeated runs with different seeds, leave-one-out cross-validation for STARE, topology-aware loss functions such as clDice loss or Betti number regularization, and validation on additional external fundus datasets.

Acknowledgment

The authors would like to express sincere gratitude to Universitas Sebelas Maret for providing research funding through the Hibah Riset Group scheme as stipulated in contract No. 462/UN27.22/PT.01.03/2026.

Funding

This research was supported by Universitas Sebelas Maret through the Hibah Riset Group scheme under contract No. 462/UN27.22/PT.01.03/2026.

Author Contribution

Billie Zandra Widiyanto contributed to the Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, and Writing – Original Draft. Wiharto provided Supervision, contributed to the Methodology and Project Administration, and participated in Writing – Review & Editing. Shaifudin Zuhdi contributed to the Methodology, Validation, and Writing – Review & Editing. All authors have read and approved the final manuscript and agreed to be accountable for all aspects of the work.

Data Availability

The DRIVE dataset used in this study is publicly available at https://drive.grand-challenge.org/ (accessed on May 7, 2026). The STARE dataset is publicly available at https://cecas.clemson.edu/~ahoover/stare/ (accessed on May 7, 2026).

Code Availability

The code and experimental scripts used in this study are available from the corresponding author upon reasonable request.

Declarations

Ethical Approval

This study utilized publicly available benchmark datasets, namely DRIVE and STARE, which do not contain personally identifiable information and do not involve human subjects, clinical interventions, or patient

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

data collection. Ethical approval was therefore not required for this study.

Consent to Participate
Not applicable, as this study did not involve human participants.

Consent for Publication
Not applicable, as this study used publicly available anonymized datasets and did not involve identifiable human participant data.

Competing Interests
The authors declare no competing interests.

Declaration of Generative AI Use
During the preparation of this manuscript, the authors used an AI-based language tool solely for grammar refinement.

References

[1] T. A. Tani and J. Tešić, "Advancing Retinal Vessel Segmentation With Diversified Deep Convolutional Neural Networks," *IEEE Access*, vol. 12, pp. 141280–141290, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3467117

[2] K.-W. Huang, Y.-R. Yang, Z.-H. Huang, Y.-Y. Liu, and S.-H. Lee, "Retinal Vascular Image Segmentation Using Improved UNet Based on Residual Module," *Bioengineering*, vol. 10, no. 6, p. 722, Jun. 2023, doi: https://doi.org/10.3390/bioengineering10060722

[3] M. Matloob Abbasi, S. Iqbal, K. Aurangzeb, M. Alhussein, and T. M. Khan, "LMBiS-Net: A lightweight bidirectional skip connection based multipath CNN for retinal blood vessel segmentation," *Sci. Rep.*, vol. 14, no. 1, p. 15219, Jul. 2024, doi: https://doi.org/10.1038/s41598-024-63496-9

[4] N. Chen, Z. Zhu, W. Yang, and Q. Wang, "Progress in clinical research and applications of retinal vessel quantification technology based on fundus imaging," *Front. Bioeng. Biotechnol.*, vol. 12, Feb. 2024, doi: https://doi.org/10.3389/fbioe.2024.1329263

[5] J. Liang, Y. Jiang, and H. Yan, "Skip connection information enhancement network for retinal vessel segmentation," *Med. Biol. Eng. Comput.*, vol. 62, no. 10, pp. 3163–3178, Oct. 2024, doi: https://doi.org/10.1007/s11517-024-03108-w

[6] T. A. Soomro *et al.*, "Impact of Novel Image Preprocessing Techniques on Retinal Vessel Segmentation," *Electronics (Basel).*, vol. 10, no. 18, p. 2297, Sep. 2021, doi: https://doi.org/10.3390/electronics10182297

[7] A. A. Abdulsahib, M. A. Mahmoud, H. Aris, S. S. Gunasekaran, and M. A. Mohammed, "An Automated Image Segmentation and Useful Feature Extraction Algorithm for Retinal Blood Vessels in Fundus Images," *Electronics (Basel).*, vol. 11, no. 9, p. 1295, Apr. 2022, doi: https://doi.org/10.3390/electronics11091295

[8] Y. Jalali, M. Fateh, and M. Rezvani, "VGA-Net: Vessel graph based attentional U-Net for retinal vessel segmentation," *IET Image Process.*, vol. 18, no. 8, pp. 2191–2213, Jun. 2024, doi: https://doi.org/10.1049/ipr2.13102

[9] A. E. Ilesanmi, T. Ilesanmi, and G. A. Gbotoso, "A systematic review of retinal fundus image segmentation and classification methods using convolutional neural networks," *Healthcare Analytics*, vol. 4, p. 100261, Dec. 2023, doi: https://doi.org/10.1016/j.health.2023.100261

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Lecture Notes in Computer Science, vol. 9351, Springer, Cham, 2015, pp. 234–241. doi: https://doi.org/10.1007/978-3-319-24574-4_28

[11] S. Xu, Z. Chen, W. Cao, F. Zhang, and B. Tao, "Retinal Vessel Segmentation Algorithm Based on Residual Convolution Neural Network," *Front. Bioeng. Biotechnol.*, vol. 9, p. 786425, Dec. 2021, doi: https://doi.org/10.3389/fbioe.2021.786425

[12] Z. Li, M. Jia, X. Yang, and M. Xu, "Blood Vessel Segmentation of Retinal Image Based on Dense-U-Net Network," *Micromachines (Basel).*, vol. 12, no. 12, p. 1478, Nov. 2021, doi: https://doi.org/10.3390/mi12121478

[13] T. M. Khan, A. Robles-Kelly, S. S. Naqvi, and M. Arsalan, "Residual Multiscale Full Convolutional Network (RM-FCN) for High Resolution Semantic Segmentation of Retinal Vasculature," in Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR 2021), Lecture Notes in Computer Science, Springer, Cham, 2021, pp. 324–333. doi: https://doi.org/10.1007/978-3-030-73973-7_31

[14] W. Jiangtao, N. I. R. Ruhaiyem, and F. Panpan, "A Comprehensive Review of U-Net and Its Variants: Advances and Applications in Medical Image Segmentation," *IET Image Process.*, vol. 19, no. 1, p. e70019, Jan. 2025, doi: https://doi.org/10.1049/ipr2.70019

[15] A. F. M. Abdun Noor, M. I. Ahasan, M. A. Khan, and G. Yang, "GeGLUNet: Structural Retinal Vessel Segmentation via Attention-Gated GeGLU and Contrastive Supervision," n Pattern Recognition and Computer Vision, 8th Chinese Conference, PRCV 2025, Shanghai, China, October 15–18, 2025, Proceedings, Part XIV, Lecture Notes in Computer Science, Springer, Singapore, 2026, pp. 494–507. doi: https://doi.org/10.1007/978-981-95-5631-1_35

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

[16] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, "Graph Attention Networks: A Comprehensive Review of Methods and Applications," *Future Internet*, vol. 16, no. 9, p. 318, Sep. 2024, doi: https://doi.org/10.3390/fi16090318

[17] P. Cibier and J.-G. Mailly, "Graph Convolutional Networks and Graph Attention Networks for Approximating Arguments Acceptability," in Computational Models of Argument, Frontiers in Artificial Intelligence and Applications, vol. 388, IOS Press, 2024, pp. 25–36. Aug. 2024, doi: https://doi.org/10.3233/FAIA240307

[18] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: https://doi.org/10.1186/s40537-021-00444-8

[19] A. A. Ramadhan and M. Baykara, "A Novel Approach to Detect COVID-19: Enhanced Deep Learning Models with Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 18, p. 9325, Sep. 2022, doi: https://doi.org/10.3390/app12189325

[20] F. A. Alotaibi *et al.*, "GPTNeXt: Biomedical Image Classification Investigations," *Diagnostics*, vol. 16, no. 4, p. 581, Feb. 2026, doi: https://doi.org/10.3390/diagnostics16040581

[21] S. Zhu, P. Wang, and K. Shen, "ProNet Adaptive Retinal Vessel Segmentation Algorithm Based on Improved UperNet Network," *Computers, Materials & Continua*, vol. 78, no. 1, pp. 283–302, 2024, doi: https://doi.org/10.32604/cmc.2023.045506

[22] Z. Han, M. Jian, and G.-G. Wang, "ConvUNeXt: An efficient convolution neural network for medical image segmentation," *Knowl. Based. Syst.*, vol. 253, p. 109512, Oct. 2022, doi: https://doi.org/10.1016/j.knosys.2022.109512

[23] S. N. Yousafzai *et al.*, "A multi-scale simplicial transformer with graph attention for facial emotion recognition," *Ain Shams Engineering Journal*, vol. 16, no. 10, p. 103584, Oct. 2025, doi: https://doi.org/10.1016/j.asej.2025.103584

[24] D. Le *et al.*, "Deep learning for artery–vein classification in optical coherence tomography angiography," *Exp. Biol. Med.*, vol. 248, no. 9, pp. 747–761, May 2023, doi: https://doi.org/10.1177/15353702231181182

[25] X. Zhang, A. Broersen, G. Van Erp, S. Pintea, and J. Dijkstra, "Continuous and complete liver vessel segmentation with graph-attention guided diffusion," *Knowl. Based. Syst.*, vol. 331, p. 114686, Jan. 2026, doi: https://doi.org/10.1016/j.knosys.2025.114686

[26] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-Based Vessel Segmentation in Color Images of the Retina," *IEEE Trans. Med. Imaging*, vol. 23, no. 4, pp. 501–509, Apr. 2004, doi: https://doi.org/10.1109/TMI.2004.825627

[27] Y. Xu, R. Quan, W. Xu, Y. Huang, X. Chen, and F. Liu, "Advances in Medical Image Segmentation: A Comprehensive Review of Traditional, Deep Learning and Hybrid Approaches," *Bioengineering*, vol. 11, no. 10, p. 1034, Oct. 2024, doi: https://doi.org/10.3390/bioengineering11101034

[28] A. D. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans. Med. Imaging*, vol. 19, no. 3, pp. 203–210, Mar. 2000, doi: https://doi.org/10.1109/42.845178

[29] M. Liu, Y. Wang, L. Wang, S. Hu, X. Wang, and Q. Ge, "IMFF-Net: An integrated multi-scale feature fusion network for accurate retinal vessel segmentation from fundus images," *Biomed. Signal Process. Control*, vol. 91, p. 105980, May 2024, doi: https://doi.org/10.1016/j.bspc.2024.105980

[30] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 11966–11976. doi: https://doi.org/10.1109/CVPR52688.2022.01167

[31] Z. Zhang *et al.*, "Gradient-based Parameter Selection for Efficient Fine-Tuning," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2024, pp. 28566–28577. doi: https://doi.org/10.1109/CVPR52733.2024.02699

[32] S. Woo *et al.*, "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2023, pp. 16133–16142. doi: https://doi.org/10.1109/CVPR52729.2023.01548

[33] C. Ji, "A Survey of Neural Network Optimization Algorithms," in *2024 IEEE 4th International Conference on Data Science and Computer Application (ICDSCA)*, IEEE, Nov. 2024, pp. 1–7. doi: https://doi.org/10.1109/ICDSCA63855.2024.10859435

[34] S. Shit *et al.*, "clDice - a Novel Topology-Preserving Loss Function for Tubular Structure Segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2021, pp. 16555–16564.

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747

doi: https://doi.org/10.1109/CVPR46437.2021.01629
[35] Z. Liu, M. S. Sunar, T. S. Tan, and W. H. W. Hitam, "Deep learning for retinal vessel segmentation: a systematic review of techniques and applications," *Med. Biol. Eng. Comput.*, vol. 63, no. 8, pp. 2191–2208, Aug. 2025, doi: https://doi.org/10.1007/s11517-025-03324-y
[36] Q. Qin and Y. Chen, "A review of retinal vessel segmentation for fundus image analysis," *Eng. Appl. Artif. Intell.*, vol. 128, p. 107454, Feb. 2024, doi: https://doi.org/10.1016/j.engappai.2023.107454
[37] D. Chicco and G. Jurman, "A statistical comparison between Matthews correlation coefficient (MCC), prevalence threshold, and Fowlkes–Mallows index," *J. Biomed. Inform.*, vol. 144, p. 104426, Aug. 2023, doi: https://doi.org/10.1016/j.jbi.2023.104426
[38] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Min.*, vol. 14, no. 1, p. 13, Feb. 2021, doi: https://doi.org/10.1186/s13040-021-00244-z
[39] Y. Ye, C. Pan, Y. Wu, S. Wang, and Y. Xia, "MFI-Net: Multiscale Feature Interaction Network for Retinal Vessel Segmentation," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 9, pp. 4551–4562, Sep. 2022, doi: https://doi.org/10.1109/JBHI.2022.3182471
[40] H. Zhang *et al.*, "BCU-Net: Bridging ConvNeXt and U-Net for medical image segmentation," *Comput. Biol. Med.*, vol. 159, p. 106960, Jun. 2023, doi: https://doi.org/10.1016/j.compbiomed.2023.106960
[41] Z. Li, X. Zhang, M. Zhao, F. Shi, and W. Zhou, "Direction-guided network for retinal vessel segmentation in OCTA images," *Biomed. Signal Process. Control*, vol. 103, p. 107455, May 2025, doi: https://doi.org/10.1016/j.bspc.2024.107455

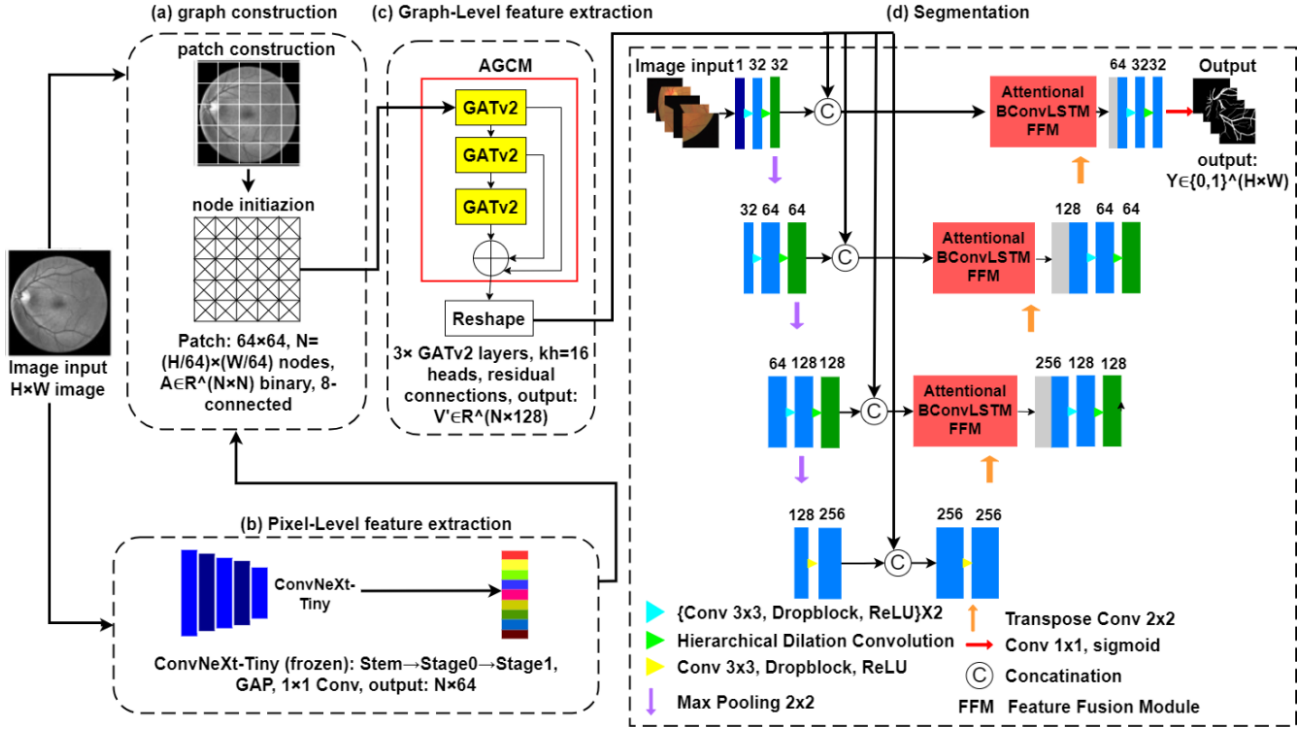
Author Biography

Billie Zandra Widiyanto received the B.S. degree in Informatics from the Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Surakarta, Indonesia, in 2026. During his undergraduate studies, he participated in the Bangkit Academy program, a Google-led initiative, specializing in the machine learning learning path, where he developed a waste detection application as part of the program's capstone project. His undergraduate interests in deep learning, computer vision, medical image analysis, and graph neural networks. He is currently pursuing opportunities to further contribute to development of efficient learning architectures for medical image.

Wiharto is a lecturer and researcher in the Department of Informatics, Faculty of Information Technology and Data Science, Universitas Sebelas Maret, Surakarta, Indonesia. He received his Ph.D. in Biomedical Informatics from Universitas Gadjah Mada, following prior training in information technology and computer science. He is affiliated with the Computational Science and Engineering Research Group, where his work centers on the application of artificial intelligence and computational intelligence to biomedical problems. His research focuses on the development of intelligent decision-support systems for medical diagnosis and health informatics. He has contributed to the field through supervising undergraduate and postgraduate students, publishing in peer-reviewed national and international journals, and participating in collaborative, interdisciplinary research initiatives.

Shaifudin Zuhdi received the B.S. degree in Mathematics from Universitas Sebelas Maret, Surakarta, Indonesia, and the M.Cs. degree in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia. He is currently a lecturer in the Department of Informatics, Faculty of Information Technology and Data Science, Universitas Sebelas Maret. He is affiliated with the Computational Science and Engineering Research Group at Universitas Sebelas Maret. His research interests include machine learning, data science, and computational intelligence, with applications in classification and predictive modeling. He has been involved in several funded research projects and actively contributes to academic supervision and interdisciplinary research activities.

Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747



Manuscript received 12 May 2026; Revised 3 June 2026; Accepted 2 July 2026; Available online 10 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1747