Jurnal Informatika dan Sistem Informasi (JIFoSI)

Volume 7 Nomor 2 Agustus 2026
ISSN (Online): 2722-130X
Halaman: 47-57

OPTIMASI STRATEGIS INFERENSI U-NET EFISIEN UNTUK SEGMENTASI PEMBULUH DARAH RETINA

*** [1]Wisanggeni Atthoriq Kuswirasatya & [1]Faisal Muttaqin**
[1]Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

DOI: https://doi.org/10.33005/jifosi.v7i2.608

*Corresponding Author: 22081010127@student.upnjatim.ac.id

Submit: Juni 2026 | Accepted: Agustus 2026 | Published: Agustus 2026

ABSTRAK

Segmentasi pembuluh darah retina merupakan tahapan krusial dalam diagnosis penyakit oftalmologi seperti diabetic retinopathy, glaukoma, dan degenerasi makula. Arsitektur U-Net telah terbukti efektif untuk tugas segmentasi pada citra medis, namun implementasi konvensionalnya seringkali memerlukan sumber daya komputasi yang besar dan waktu inferensi yang lama. Penelitian ini mengusulkan serangkaian optimasi pada arsitektur U-Net baseline untuk segmentasi pembuluh darah retina dengan fokus pada efisiensi komputasi tanpa mengorbankan akurasi secara signifikan. Optimasi yang dilakukan meliputi: (1) penerapan Contrast Limited Adaptive Histogram Equalization (CLAHE) pada tahap preprocessing, (2) penggunaan fungsi aktivasi LeakyReLU sebagai pengganti ReLU, (3) reduksi kanal pada bottleneck dari 1024 menjadi 512, (4) perbaikan fungsi loss dengan Binary Cross Entropy with Logits, (5) penerapan Automatic Mixed Precision (AMP) selama pelatihan, dan (6) konvolusi-BatchNorm fusion pada tahap inferensi. Eksperimen dilakukan pada dataset DRIVE (Digital Retinal Images for Vessel Extraction) menggunakan lingkungan Google Colab dengan GPU NVIDIA T4. Hasil menunjukkan bahwa model teroptimasi mencapai peningkatan kecepatan inferensi hingga 3,51 kali (dari 6,90 FPS menjadi 24,26 FPS) dan pengurangan waktu pelatihan sebesar 57,1% (dari 17 menit 11 detik menjadi 7 menit 22 detik) dengan penurunan Jaccard Index yang minimal dari 0,6571 menjadi 0,6404. Temuan ini menunjukkan bahwa optimasi yang diusulkan berhasil menciptakan model yang lebih efisien dan cocok untuk deployment pada perangkat dengan sumber daya terbatas.

Kata kunci: U-Net, segmentasi retina, CLAHE, mixed precision, Conv-BN fusion, deep learning

STRATEGIC OPTIMIZATION OF AN EFFICIENT U-NET INFERENCE FOR RETINAL BLOOD VESSEL SEGMENTATION

ABSTRACT

Retinal blood vessel segmentation is a crucial step in the diagnosis of ophthalmic diseases such as diabetic retinopathy, glaucoma, and macular degeneration. The U-Net architecture has proven effective for segmentation tasks in medical imaging; however, its conventional implementation often requires significant computational resources and results in long inference times. This study proposes a series of optimizations to the baseline U-Net architecture for retinal blood vessel segmentation, focusing on computational efficiency without significantly compromising accuracy. The optimizations include: (1) the application of Contrast Limited Adaptive Histogram Equalization (CLAHE) during the preprocessing stage, (2) the use of the LeakyReLU activation function as a replacement for ReLU, (3) channel reduction at the bottleneck from 1024

to 512, (4) improvement of the loss function using Binary Cross Entropy with Logits, (5) the implementation of Automatic Mixed Precision (AMP) during training, and (6) convolution-BatchNorm fusion during the inference stage. Experiments were conducted on the DRIVE (Digital Retinal Images for Vessel Extraction) dataset using the Google Colab environment with an NVIDIA T4 GPU. The results show that the optimized model achieved a 3.51-fold increase in inference speed (from 6.90 FPS to 24.26 FPS) and a 57.1% reduction in training time (from 17 minutes 11 seconds to 7 minutes 22 seconds) with a minimal decrease in the Jaccard Index from 0.6571 to 0.6404. These findings demonstrate that the proposed optimization successfully produced a more efficient model suitable for deployment on resource-constrained devices.

Keywords: U-Net, Retina Segmentation, CLAHE, mixed precision, Conv-BN fusion, deep learning

PENDAHULUAN

Segmentasi pembuluh darah retina merupakan salah satu tahapan fundamental dalam diagnosis penyakit mata menggunakan citra fundus. Analisis terhadap struktur vascular retina memungkinkan deteksi dini berbagai kondisi patologis termasuk diabetic retinopathy, hipertensi retinal, glaukoma, dan degenerasi makula terkait usia [6]. Akurasi segmentasi yang tinggi sangat penting karena kesalahan kecil dalam identifikasi pembuluh darah dapat mengakibatkan diagnosis yang tidak tepat dan penanganan yang salah [8].

Dalam dekade terakhir, pendekatan berbasis deep learning telah mendominasi penelitian segmentasi citra medis. Arsitektur U-Net yang diperkenalkan oleh Ronneberger et al. pada tahun 2015 [10] telah menjadi standar de facto untuk berbagai tugas segmentasi citra biomedis karena struktur encoder-decoder simetrisnya dengan skip connections yang mampu menggabungkan informasi fitur tingkat rendah dan tinggi. Berbagai varian U-Net telah dikembangkan untuk meningkatkan performa, termasuk R2U-Net/ResUNet yang menambahkan residual connections [17], U-Net++ dengan nested skip pathways [16], serta Attention U-Net yang mengintegrasikan mekanisme attention [9].

Meskipun U-Net dan variannya telah menunjukkan akurasi segmentasi yang sangat baik, implementasi konvensionalnya seringkali mengabaikan aspek efisiensi komputasi.

Dalam konteks klinis nyata, model dengan parameter yang sangat besar dan waktu inferensi yang lama menjadi tidak praktis untuk deployment pada perangkat dengan sumber daya terbatas seperti klinik di daerah terpencil atau perangkat mobile [14]. Survei komprehensif oleh Liu et al. [15] menyoroti bahwa meskipun model deep learning terkini mencapai akurasi tinggi pada segmentasi retina, beban komputasinya yang besar seringkali menjadi hambatan utama untuk aplikasi klinis. Demikian pula, penelitian oleh Zhao et al. [14] mengenai lightweight U-Net menunjukkan bahwa reduksi kompleksitas model dapat mencapai trade-off yang menguntungkan antara akurasi dan efisiensi.

Berdasarkan kode baseline yang dipublikasikan di platform Kaggle [13], teridentifikasi beberapa area yang dapat dioptimasi untuk meningkatkan efisiensi tanpa mengorbankan akurasi secara drastis. Kode baseline menggunakan arsitektur U-Net standar dengan beberapa karakteristik yang dapat menyebabkan inefisiensi: (1) bottleneck dengan 1024 kanal yang memerlukan komputasi konvolusi yang intensif, (2) fungsi loss yang menerapkan sigmoid sebelum Binary Cross Entropy yang dapat menyebabkan instabilitas numerik, (3) tidak adanya teknik mixed precision training yang dapat mempercepat pelatihan pada GPU modern, (4) preprocessing tanpa enhancement kontras lokal, dan (5) tidak adanya optimasi khusus untuk tahap inferensi. Penelitian ini bertujuan untuk mengoptimasi arsitektur U-Net baseline melalui serangkaian modifikasi yang terarah pada preprocessing, arsitektur model, fungsi loss, strategi pelatihan, dan optimasi inferensi. Kontribusi utama penelitian ini meliputi: (1) integrasi CLAHE pada pipeline preprocessing untuk meningkatkan visibilitas pembuluh darah, (2) modifikasi arsitektur dengan reduksi bottleneck dan penggunaan LeakyReLU, (3) perbaikan numerik pada fungsi loss menggunakan BCEWithLogits, (4) implementasi Automatic Mixed Precision (AMP) untuk akselerasi pelatihan, dan (5) penerapan Conv-BN fusion untuk akselerasi inferensi. Hipotesis penelitian ini adalah bahwa kombinasi optimasi tersebut dapat menghasilkan model yang secara signifikan lebih efisien dalam hal waktu pelatihan dan inferensi dengan penurunan akurasi yang dapat diterima.

Penelitian ini bertujuan untuk mengoptimasi arsitektur U-Net baseline melalui serangkaian modifikasi yang terarah pada preprocessing, arsitektur model, fungsi loss, strategi pelatihan, dan optimasi inferensi. Kontribusi utama penelitian ini meliputi: (1) integrasi CLAHE pada pipeline preprocessing untuk meningkatkan visibilitas pembuluh darah, (2) modifikasi arsitektur dengan reduksi bottleneck dan penggunaan LeakyReLU, (3) perbaikan numerik pada fungsi loss menggunakan BCEWithLogits, (4) implementasi Automatic Mixed Precision (AMP) untuk akselerasi pelatihan, dan (5) penerapan Conv-BN fusion untuk akselerasi inferensi. Hipotesis penelitian ini adalah bahwa kombinasi optimasi tersebut dapat menghasilkan model yang secara signifikan lebih efisien dalam hal waktu pelatihan dan inferensi dengan penurunan akurasi yang dapat diterima.

METODOLOGI

Dataset

Penelitian ini menggunakan dataset DRIVE (Digital Retinal Images for Vessel Extraction) yang merupakan dataset benchmark standar untuk evaluasi metode segmentasi pembuluh darah retina [4]. Dataset ini terdiri dari 40 citra fundus retina berwarna dengan resolusi 565 x 584 piksel, yang diperoleh dari program skrining diabetic retinopathy di Belanda. Data terbagi menjadi 20 citra untuk training dan 20 citra untuk testing. Setiap citra dalam set training memiliki dua anotasi manual yang dibuat oleh ahli oftalmologi, sedangkan set testing memiliki satu anotasi manual untuk evaluasi. Citra fundus pada dataset DRIVE menunjukkan variasi dalam hal pencahayaan, kontras, dan karakteristik vascular yang merepresentasikan kondisi klinis nyata.

Preprocessing dan Augmentasi Data

Tahap preprocessing merupakan langkah krusial untuk meningkatkan kualitas citra input sebelum masuk ke jaringan neural. Penelitian ini menerapkan Contrast Limited Adaptive Histogram Equalization (CLAHE) pada setiap citra training dan testing. CLAHE bekerja dengan membagi citra menjadi region lokal (tiles) dan menerapkan histogram equalization secara adaptif pada setiap region dengan pembatasan contrast enhancement untuk mencegah amplifikasi noise [5]. Implementasi CLAHE dilakukan pada channel luminance (L) dari ruang warna LAB dengan parameter clipLimit = 2.0 dan tileGridSize = (8, 8).

Berbeda dengan kode baseline yang tidak menerapkan teknik enhancement kontras, penerapan CLAHE bertujuan untuk meningkatkan visibilitas pembuluh darah yang tipis dan meningkatkan perbedaan kontras antara foreground (pembuluh darah) dengan background (retina). Liskowski et al. [11] menunjukkan bahwa preprocessing dengan metode kontras enhancement pada citra retina dapat meningkatkan akurasi segmentasi secara signifikan. Selain itu, mask dalam proses resize menggunakan interpolasi INTER_NEAREST untuk mempertahankan label biner tanpa memperkenalkan nilai interpolasi antara 0 dan 1.

Augmentasi data dilakukan dengan teknik geometric transformations meliputi horizontal flip, vertical flip, dan rotasi 45 derajat. Dari 16 citra training asli (setelah split 80:20), proses transformasi tersebut menghasilkan 48 citra augmented baru. Dengan demikian, diperoleh total 64 citra pada dataset training (gabungan dari 16 citra asli dan 48 citra hasil augmentasi) yang siap digunakan untuk pemrosesan lebih lanjut. Citra dan mask tersebut diubah ukurannya menjadi 512 x 512 piksel sebelum disimpan ke dalam direktori penyimpanan lokal. Proses normalisasi citra dengan membagi nilai piksel dengan 255.0 tidak dilakukan pada tahap pembuatan file dataset augmentasi fisik tersebut. Pada fase pelatihan dan validasi, normalisasi dieksekusi secara dinamis (on-the-fly) di dalam metode __getitem__ pada kelas DriveDataset ketika data dipanggil oleh DataLoader. Sementara itu, pada fase pengujian (testing), normalisasi dilakukan secara prosedural langsung pada skrip pengujian utama (test block) sesaat setelah citra pengujian dimuat ke memori menggunakan OpenCV.

Arsitektur Model

Arsitektur model dalam penelitian ini merupakan modifikasi dari U-Net standar dengan beberapa perubahan signifikan. Secara umum, arsitektur terdiri dari empat encoder blocks, satu bottleneck, empat decoder blocks, dan layer output konvolusi 1x1. Setiap encoder block terdiri dari dua lapisan konvolusi 3x3

dengan batch normalization dan fungsi aktivasi, diikuti dengan max pooling 2x2. Setiap decoder block menggunakan transposed convolution 2x2 untuk upsampling, concatenation dengan skip connection dari encoder yang bersesuaian, dan dua lapisan konvolusi 3x3.

Modifikasi utama pada arsitektur meliputi: (1) Reduksi kanal pada blok bottleneck dari yang semula berdimensi 512 ke 1024 kanal (pada model baseline) menjadi 512 ke 512 kanal, yang secara signifikan memangkas jumlah parameter; (2) Penyesuaian dimensi kanal input pada lapisan konvolusi transposisi (ConvTranspose2d) di blok Decoder 1 dari 1024 menjadi 512 kanal untuk menyelaraskan dengan penciutan output dari bottleneck, di mana total kanal pasca-proses penggabungan (concatenation) dengan skip connection tetap berjumlah 1024 kanal (512 + 512) sebelum akhirnya direduksi kembali oleh blok konvolusi; dan (3) Penggantian fungsi aktivasi ReLU menjadi LeakyReLU dengan nilai negative slope sebesar 0.01 pada setiap blok konvolusi. Penggunaan LeakyReLU bertujuan untuk mengatasi masalah dying ReLU yang dapat menghambat pembelajaran fitur pada neuron dengan aktivasi negatif [1]. Evaluasi empiris oleh Xu et al. [3] menunjukkan bahwa penggunaan LeakyReLU memberikan konvergensi dan performa yang lebih baik pada jaringan konvolusi mendalam dibandingkan ReLU standar, yang mendukung penerapannya pada arsitektur U-Net di penelitian ini. Struktur lengkap arsitektur model disajikan pada Tabel 1.

Fungsi Loss

Fungsi loss yang digunakan adalah kombinasi antara Dice Loss dan Binary Cross Entropy (BCE), yang dikenal sebagai DiceBCELoss. Perbedaan mendasar dari implementasi baseline adalah penggunaan F.binary_cross_entropy_with_logits() sebagai pengganti tata cara manual berupa fungsi torch.sigmoid() yang langsung diikuti oleh F.binary_cross_entropy().

Tabel 1. Arsitektur U-Net Teroptimasi

Blok	Input Kanal	Output Kanal	Operasi Utama
Encoder 1	3	64	2 x (Conv 3x3 + BN + LeakyReLU), MaxPool
Encoder 2	64	128	2 x (Conv 3x3 + BN + LeakyReLU), MaxPool
Encoder 3	128	256	2 x (Conv 3x3 + BN + LeakyReLU), MaxPool
Encoder 4	256	512	2 x (Conv 3x3 + BN + LeakyReLU), MaxPool
Bottleneck	512	512	2 x (Conv 3x3 + BN + LeakyReLU)
Decoder 1	512	512	UpConv 2x2 + Concat + 2 x (Conv 3x3 + BN + LeakyReLU)
Decoder 2	512	256	UpConv 2x2 + Concat + 2 x (Conv 3x3 + BN + LeakyReLU)
Decoder 3	256	128	UpConv 2x2 + Concat + 2 x (Conv 3x3 + BN + LeakyReLU)
Decoder 4	128	64	UpConv 2x2 + Concat + 2 x (Conv 3x3 + BN + LeakyReLU)
Output	64	1	Conv 1x1

Sumber: Pribadi

Perubahan ini memberikan keuntungan numerik yang signifikan karena fungsi dengan arsitektur with logits menggabungkan operasi sigmoid dan BCE ke dalam satu kernel terintegrasi yang jauh lebih stabil secara numerik, terutama saat berhadapan dengan nilai probabilitas ekstrem.

Namun, operasi sigmoid tidak sepenuhnya dihilangkan dari arsitektur fungsi loss teroptimasi ini. Operasi sigmoid hanya didepak dari jalur perhitungan BCE untuk mencegah instabilitas numerik (seperti gradient overflow/underflow), namun tetap dipertahankan secara spesifik pada awal jalur komputasi Dice Loss. Hal ini mutlak diperlukan guna memastikan nilai output (logits) dari model tetap ditransformasikan ke dalam rentang probabilitas biner 0 hingga 1 sebelum dilakukan kalkulasi perkalian matriks intersection dan union.

Strategi Pelatihan

Pelatihan model dilakukan dengan optimizer Adam menggunakan learning rate 1e-4 selama 50 epoch. Learning rate scheduler ReduceLROnPlateau dengan patience=5 diterapkan untuk mengurangi learning rate secara otomatis ketika validasi loss tidak membaik. Strategi pelatihan utama yang membedakan penelitian ini dari baseline adalah penerapan Automatic Mixed Precision (AMP) menggunakan fungsi konteks torch.amp.autocast() dan penyesuai gradien torch.amp.GradScaler().

AMP memungkinkan pelatihan dengan campuran precision float32 dan float16, di mana sebagian besar operasi konvolusi dan matmul dijalankan dalam float16 untuk memanfaatkan Tensor Cores pada GPU modern, sementara operasi yang sensitif terhadap precision dipertahankan dalam float32 [12]. Implementasi AMP menggunakan GradScaler untuk automatic gradient scaling yang mencegah gradient underflow pada float16. Batch size ditingkatkan dari 2 (baseline) menjadi 8 berkat penghematan memori yang dihasilkan oleh AMP. Selain itu, parameter non_blocking=True digunakan pada operasi transfer data ke GPU untuk mengaktifkan asynchronous data transfer yang mengurangi waktu idle GPU.

Optimasi Interferensi

Pada tahap inferensi (testing), dilakukan beberapa optimasi utama: (1) Conv-BN Fusion, yaitu penggabungan parameter BatchNorm ke dalam lapisan konvolusi mendahuluinya secara spesifik pada lapisan conv1 dan conv2 di setiap conv_block. Teknik ini menghilangkan overhead komputasi BatchNorm selama inferensi pada arsitektur U-Net plain dengan mengubah parameter gamma, beta, running mean, dan running variance menjadi modifikasi weight dan bias konvolusi secara langsung [2], sementara lapisan BatchNorm yang telah dilebur diubah menjadi lapisan identitas (Identity) untuk memangkas jalur komputasi. (2) Penggunaan torch.inference_mode() sebagai pengganti torch.no_grad() yang memberikan optimasi tambahan untuk inference-only operations; (3) Penerapan torch.amp.autocast() pada tahap inferensi untuk mengaktifkan mixed precision inference yang lebih cepat pada GPU dengan dukungan Tensor Cores.

Metrik Evaluasi

Performa model dievaluasi menggunakan enam metrik utama: (1) Jaccard Index (IoU) yang mengukur overlap antara prediksi dan ground truth; (2) F1-Score sebagai harmonic mean dari precision dan recall; (3) Sensitivity (Recall) yang mengukur kemampuan model mendeteksi pembuluh darah yang sebenarnya; (4) Precision yang mengukur ketepatan prediksi positif; (5) Accuracy yang mengukur keseluruhan ketepatan klasifikasi piksel; dan (6) Specificity yang mengukur kemampuan model mengidentifikasi background dengan benar. Selain metrik akurasi, efisiensi komputasi diukur melalui pencatatan waktu pelatihan dan kecepatan inferensi dalam satuan Frame Per Second (FPS). Untuk menjamin akurasi dan integritas pengujian, pengukuran FPS ini dibatasi secara ketat pada cakupan Pure Model Inference Speed (kecepatan murni forward pass model dan fungsi aktivasi akhir di dalam GPU). Pengukuran dieksekusi menggunakan fungsi penanda waktu yang diapit oleh torch.cuda.synchronize() guna memastikan seluruh operasi sirkuit GPU telah selesai sebelum waktu dicatat. Dengan demikian, overhead eksternal dari pipeline seperti pembacaan citra dari disk (I/O latency), transfer data dari host ke device (x.to(device)), operasi thresholding biner, kalkulasi metrik Scikit-Learn, serta proses penyimpanan gambar hasil ke media penyimpanan fisik sengaja dieksklusi dari metrik FPS ini agar evaluasi performa arsitektur berjalan objektif.

Lingkungan Eksperimen

Seluruh eksperimen dilakukan pada lingkungan Google Colab dengan runtime GPU NVIDIA T4. Spesifikasi GPU NVIDIA T4 meliputi 16 GB memori VRAM dengan arsitektur Turing yang mendukung mixed precision training melalui Tensor Cores. Framework yang digunakan adalah PyTorch dengan CUDA support. Ukuran input citra adalah 512 x 512 piksel dengan 3 channel warna (BGR) untuk input dan 1 channel grayscale untuk output segmentasi.

HASIL DAN PEMBAHASAN

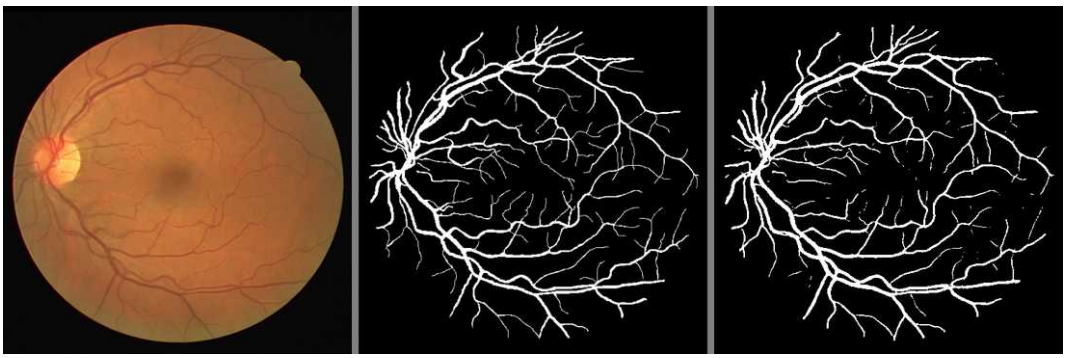
Performa Segmentasi

Hasil evaluasi performa segmentasi antara model baseline dan model teroptimasi pada dataset DRIVE disajikan pada Tabel 2. Model baseline mencapai Jaccard Index 0,6571, F1-Score 0,7927, Sensitivity 0,7662, Precision 0,8266, Accuracy 0,9652, dan Specificity 0,9845. Model teroptimasi menghasilkan Jaccard Index 0,6404, F1-Score 0,7806, Sensitivity 0,7879, Precision 0,7784, Accuracy 0,9615, dan Specificity 0,9784.

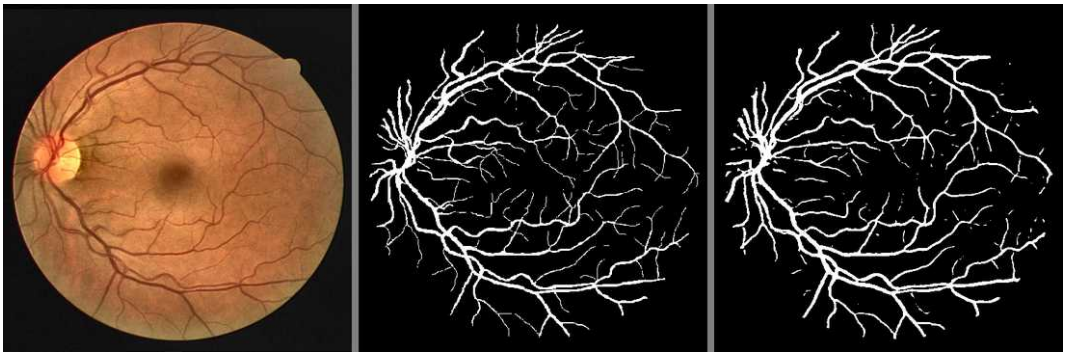
Tabel 2. Arsitektur U-Net Teroptimasi

Metrik	Baseline	Teroptimasi
Jaccard Index	0,6571	0,6404
F1-Score	0,7927	0,7806
Sensitivity	0,7662	0,7879
Precision	0,8266	0,7784
Accuracy	0,9652	0,9615
Specificity	0,9845	0,9784

Sumber: Pribadi



Gambar 1. ***Database Mirroring Architecture***



Gambar 2. ***Hasil dari Model Teroptimasi (Modifikasi)***

Gambar 1 dan 2 diatas menunjukkan perbandingan kualitatif hasil segmentasi pembuluh darah retina pada dataset DRIVE. Dari kiri ke kanan pada masing-masing panel: Citra fundus asli, ground truth (anotasi ahli), dan hasil prediksi biner model Dapat diamati bahwa model teroptimasi mengalami penurunan Jaccard Index sebesar 0,0167 (2,54%) dan F1-Score sebesar 0,0121 (1,53%) dibandingkan dengan baseline. Penurunan ini dianggap dapat diterima mengingat signifikan peningkatan efisiensi komputasi yang diperoleh. Menariknya, model teroptimasi menunjukkan Sensitivity yang lebih tinggi (0,7879 vs 0,7662), yang mengindikasikan kemampuan lebih baik dalam mendeteksi pembuluh darah yang ada. Hal ini kemungkinan disebabkan oleh penerapan CLAHE yang meningkatkan visibilitas pembuluh darah tipis dan LeakyReLU yang memungkinkan pembelajaran representasi yang lebih kaya.

Penurunan pada metrik Precision (0,8266 menjadi 0,7784) mengindikasikan bahwa model teroptimasi mendeteksi lebih banyak piksel positif di luar pembuluh darah utama jika dibandingkan dengan baseline.

Fenomena ini dapat dijelaskan oleh dua faktor komputasional mendasar. Pertama, penerapan CLAHE meningkatkan kontras secara lokal, yang pada beberapa region fundus juga memperkuat intensitas noise atau artefak latar belakang sehingga menyerupai struktur pembuluh darah halus.

Kedua, terdapat faktor bias metrik pada model baseline yang dipicu oleh kesalahan metode pengubahan ukuran (resize) mask target. Pada model baseline, mask ground truth di-resize menggunakan interpolasi bilinear (default) yang mendegradasi nilai biner pada tepian pembuluh darah menjadi nilai kontinu (efek blur abu-abu). Karakteristik target yang kabur ini melatih model baseline secara konservatif untuk hanya memprediksi area inti vaskular yang memiliki kepastian tinggi. Ketika ambang batas (threshold) biner > 0.5 diterapkan pada tahap pengujian, hasil prediksi pembuluh darah model baseline menjadi sangat kurus under-segmentation. Prediksi yang terlalu sempit ini meminimalkan peluang luberan piksel ke area latar belakang, sehingga secara semu menekan jumlah false positive (FP) secara ekstrem dan mendongkrak metrik Precision secara artifisial, meskipun sebenarnya model tersebut kehilangan banyak detail batas vaskular.

Sebaliknya, model teroptimasi secara tepat menerapkan interpolasi cv2.INTER_NEAREST pada mask, yang mempertahankan nilai biner murni (0 atau 1) sesuai dengan anotasi asli dari ahli medis. Integrasi target biner yang tegas ini, dikombinasikan dengan peningkatan kontras lokal dari CLAHE, mendorong model teroptimasi untuk memetakan batas vaskular secara lebih utuh dan agresif dalam melacak pembuluh darah tipis (terbukti dari peningkatan Sensitivity menjadi 0,7879). Keagresifan model dalam menangkap detail ini membawa konsekuensi berupa sedikit peningkatan false positive di area transisi batas luar vaskular. Oleh karena itu, perbaikan ini memberikan penilaian metrik Precision yang jauh lebih jujur, objektif, dan merepresentasikan kondisi klinis yang sebenarnya, meskipun secara teoretis mencatatkan nilai absolut yang lebih rendah dibandingkan baseline.

Efisien Komputasi

Aspek efisiensi komputasi merupakan fokus utama penelitian ini. Perbandingan efisiensi antara model baseline dan teroptimasi disajikan pada Tabel 3. Berdasarkan hasil pengujian murni pada sirkuit komputasi GPU (Pure Model Inference), model teroptimasi berhasil mencatatkan lompatan performa dengan kecepatan inferensi mencapai 24,26 FPS, berbanding terbalik dengan model baseline yang tertahan pada angka 6,90 FPS. Capaian ini merepresentasikan akselerasi throughput model sebesar 3,51 kali lipat lebih cepat. Peningkatan dramatis pada kemampuan computational throughput ini secara langsung dipicu oleh kombinasi empat faktor strategis: (1) reduksi bottleneck kanal yang secara masif memangkas beban operasi konvolusi matematika; (2) penerapan Conv-BN fusion yang sukses melenyapkan beban overhead aritmatika lapisan BatchNorm pada fase pengujian; (3) utilisasi mixed precision inference melalui torch.amp.autocast() yang memaksimalkan efisiensi pemrosesan data pada matriks Tensor Cores GPU NVIDIA T4 ; serta (4) penguncian memori lewat torch.inference_mode() yang membebaskan sistem dari beban kerja gradient tracking. Melalui isolasi pengukuran ini, terbukti bahwa optimasi arsitektural dan optimasi tingkat kernel yang diusulkan mampu menghasilkan model berkemampuan real-time processing di sisi perangkat keras (GPU bound) tanpa terdistorsi oleh fluktuasi kecepatan operasi Input/Output (I/O) media penyimpanan lokal.

Tabel 3. Perbandingan Efisiensi Komputasi

Parameter	Baseline	Teroptimasi
Waktu Pelatihan	17 menit 11 detik	7 menit 22 detik
Epoch Time (rata-rata)	~20 detik	~8 detik
Kecepatan Inferensi (FPS)	6,90	24,26
Speedup Inferensi	1x	3,51x
Batch Size	2	8
Reduksi Waktu Pelatihan	-	57,1%

Sumber: Pribadi

Waktu pelatihan juga mengalami pengurangan signifikan dari 17 menit 11 detik menjadi 7 menit 22 detik (reduksi 57,1%). Pengurangan ini disebabkan oleh: (1) penerapan AMP yang mempercepat operasi

konvolusi melalui Tensor Cores; (2) peningkatan batch size dari 2 menjadi 8 yang mengurangi jumlah iterasi per epoch; (3) penerapan parameter pin_memory=True secara khusus pada DataLoader pelatihan serta argumen non_blocking=True pada proses transfer data ke GPU yang mengurangi waktu idle GPU. Perlu dicatat bahwa waktu augmentasi data pada model teroptimasi sedikit lebih lama (50 detik vs 44 detik untuk 16 citra) karena adanya operasi CLAHE, namun overhead ini dapat diterima mengingat manfaatnya pada kualitas segmentasi.

Analisis Konvergensi Pelatihan

Analisis konvergensi pelatihan menunjukkan perbedaan pola yang signifikan antara model baseline dan teroptimasi. Model baseline mencapai validasi loss terbaik sebesar 0,3884 pada epoch 47–50, sedangkan model teroptimasi mencapai validasi loss terbaik sebesar 0,7201 pada epoch 50. Meskipun secara matematis komponen fungsi loss berbasis logits (F.binary_cross_entropy_with_logits) memiliki skala nilai yang identik dengan kombinasi sigmoid dan BCE manual, nilai akhir validasi loss model teroptimasi secara konsisten terpantau lebih tinggi. Fenomena ini tidak disebabkan oleh perbedaan skala fungsi loss, melainkan oleh kombinasi empat faktor komputasional dan struktural data.

Pertama (Frekuensi Pembaruan Bobot): Peningkatan batch size dari 2 menjadi 8 secara drastis mengurangi frekuensi pembaruan bobot (weight updates) jaringan di setiap epoch-nya. Berdasarkan jumlah data training yang digunakan, model baseline (batch size = 2) melakukan 32 kali pembaruan bobot per epoch, atau akumulasi sebanyak 1.600 kali pembaruan sepanjang 50 epoch. Sementara itu, model teroptimasi (batch size = 8) hanya melakukan 8 kali pembaruan bobot per epoch, yang berarti hanya terjadi 400 kali pembaruan bobot sepanjang 50 epoch. Karena frekuensi iterasinya berkurang hingga empat kali lipat, model teroptimasi belum mencapai titik konvergensi penuh pada epoch ke-50 dan memerlukan jumlah epoch yang lebih panjang untuk mencapai nilai loss absolut yang setara dengan baseline.

Kedua, Modifikasi arsitektur berupa reduksi kanal pada bottleneck dari 1024 menjadi 512 kanal secara signifikan memangkas kapasitas model untuk menghafal (overfitting) variasi detail pada dataset training yang relatif kecil. Pemangkasan ini menyebabkan kurva penurunan loss berjalan lebih landai namun menghasilkan generalisasi fitur pembuluh darah yang lebih sehat.

Ketiga, Terdapat pengaruh fundamental dari perbaikan metode pengubahan ukuran (resize) mask. Pada model baseline, mask ground truth di-resize menggunakan interpolasi default (bilinear) yang menghasilkan degradasi nilai kontinu (efek blur abu-abu) di sepanjang tepian pembuluh darah. Kondisi ini membuat perhitungan BCE pada baseline secara artifisial bernilai lebih rendah karena distribusi target yang melandai. Sebaliknya, model teroptimasi secara tepat menggunakan interpolasi cv2.INTER_NEAREST yang mempertahankan nilai biner murni (0 atau 1) sesuai anotasi asli ahli medis. Ketegasan nilai target biner ini menyebabkan ketidakcocokan piksel batas sekecil apa pun langsung diganjar penalti loss yang lebih tinggi secara numerik.

Keempat, Penerapan CLAHE meningkatkan kontras secara lokal yang pada beberapa region dapat memperkuat noise atau artefak latar belakang sehingga menyerupai struktur pembuluh darah halus. Akibatnya, model teroptimasi dipaksa menghitung loss pada citra input yang memiliki tingkat gangguan latar belakang yang lebih menonjol, sehingga nilai error (loss) pada awal-awal epoch hingga akhir pelatihan akan bertahan lebih tinggi dibandingkan citra baseline yang mulus namun miskin detail vascular.

Meskipun nilai loss absolutnya berbeda, kedua model tetap menunjukkan pola konvergensi yang serupa dengan kurva validasi loss yang melandai dan stabil di epoch-akhir pelatihan. Di sisi lain, model teroptimasi menunjukkan variasi waktu per epoch yang sedikit lebih dinamis (7–11 detik) dibandingkan baseline yang cenderung konstan pada kisaran 19–21 detik. Variabilitas ini merupakan dampak dari mekanisme penjadwalan dinamis (dynamic scheduling) pada shared environment GPU Google Colab. Kendati demikian, rata-rata waktu per epoch model teroptimasi (sekitar 8 detik) terbukti secara signifikan lebih cepat dibandingkan model baseline (sekitar 20 detik).

Pembahasan

Hasil penelitian ini menunjukkan bahwa pendekatan multi-faceted optimization dapat menghasilkan model yang secara signifikan lebih efisien dengan penurunan akurasi yang minimal. Kontribusi setiap komponen optimasi dapat dianalisis sebagai berikut:

Penerapan CLAHE pada preprocessing memberikan dampak positif pada sensitivity dengan meningkatkan visibilitas pembuluh darah tipis yang sulit terdeteksi. Hal ini sejalan dengan temuan Liu et al. [15] yang menunjukkan dalam survei komprehensifnya bahwa penerapan contrast enhancement seperti CLAHE pada tahap preprocessing secara konsisten meningkatkan performa segmentasi pembuluh darah pada dataset DRIVE. Penggunaan ruang warna LAB dengan CLAHE pada channel L memungkinkan enhancement kontras yang tidak mempengaruhi informasi chromaticity.

Reduksi bottleneck dari 1024 menjadi 512 kanal merupakan kontribusi terbesar terhadap peningkatan efisiensi. Bottleneck dengan 1024 kanal pada U-Net standar menghasilkan jumlah parameter yang sangat besar pada lapisan konvolusi 3x3 (512x1024x3x3 + 1024x1024x3x3 = 14,16 juta parameter untuk bottleneck saja). Dengan reduksi menjadi 512 kanal, jumlah parameter pada blok bottleneck berkurang secara signifikan dari 14,16 juta menjadi 4,72 juta parameter (reduksi sebesar 66,7%). Strategi ini terinspirasi dari konsep bottleneck design pada MobileNetV2 [7] yang menunjukkan bahwa representasi terkompresi pada bottleneck dapat mempertahankan informasi yang cukup untuk tugas segmentasi.

Penggunaan LeakyReLU dengan slope 0,01 memberikan karakteristik aktivasi yang memungkinkan aliran gradient kecil pada nilai negatif, mengatasi masalah dying ReLU yang dapat menghambat pembelajaran fitur. Dalam konteks segmentasi pembuluh darah retina di mana sebagian besar piksel adalah background (nilai 0), kemampuan LeakyReLU untuk mempertahankan aktivasi kecil pada neuron yang menerima input negatif dapat membantu model mempelajari representasi yang lebih nuansa untuk pembuluh darah tipis.

Penerapan AMP menunjukkan efektivitasnya dalam mempercepat pelatihan dan inferensi pada GPU NVIDIA T4 yang dilengkapi Tensor Cores. Micikevicius et al. [12] melaporkan speedup lebih dari 2x pada berbagai tugas segmentasi citra medis 3D dengan AMP. Hasil penelitian ini memperkuat temuan tersebut pada konteks segmentasi retina 2D dengan speedup training mendekati 2,3x dan speedup inference mencapai 3,5x.

Conv-BN fusion pada tahap inferensi merupakan teknik well-established yang menghilangkan overhead komputasi BatchNorm dengan menggabungkan parameter statistiknya langsung ke dalam lapisan konvolusi standar pada bagian encoder, bottleneck, dan decoder [2]. Optimasi ini murni menyasar operasi konvolusi dasar tanpa melibatkan koneksi residual (shortcut), sehingga tidak memengaruhi nilai luaran (output) dari model U-Net teroptimasi, melainkan murni mengurangi jumlah operasi aritmatika dan meminimalkan latensi pemrosesan saat inferensi dijalankan. Ketika dikombinasikan dengan mixed precision inference, model dapat mencapai throughput yang jauh lebih tinggi pada hardware yang sama.

KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan dan mengevaluasi serangkaian optimasi pada arsitektur U-Net untuk segmentasi pembuluh darah retina. Kombinasi enam strategi optimasi yang terintegrasi meliputi CLAHE preprocessing, modifikasi arsitektur dengan bottleneck tereduksi, penggunaan LeakyReLU, perbaikan fungsi loss dengan BCEWithLogits, Automatic Mixed Precision training, dan Conv-BN fusion untuk inferensi, telah terbukti efektif dalam menciptakan model yang secara signifikan lebih efisien secara komputasi.

Model teroptimasi mencapai peningkatan kecepatan inferensi hingga 3,51 kali lipat (24,26 FPS vs 6,90 FPS) dan pengurangan waktu pelatihan sebesar 57,1% (7 menit 22 detik vs 17 menit 11 detik) dengan penurunan Jaccard Index yang minimal (0,6404 vs 0,6571). Peningkatan pada metrik Sensitivity (0,7879 vs 0,7662) menunjukkan bahwa model teroptimasi memiliki kemampuan lebih baik dalam mendeteksi pembuluh darah retina yang ada, yang merupakan karakteristik penting dalam konteks diagnosis klinis.

Kontribusi utama penelitian ini terletak pada demonstrasi bahwa trade-off antara akurasi dan efisiensi dapat diatur secara menguntungkan melalui kombinasi teknik optimasi yang komplementer. Model yang dihasilkan memiliki potensi aplikasi yang luas pada sistem diagnosis bantu (Computer-Aided Diagnosis) yang memerlukan inferensi real-time pada perangkat dengan sumber daya terbatas. Peningkatan FPS yang

signifikan membuka kemungkinan deployment pada klinik di daerah terpencil, perangkat mobile untuk skrining mata, dan sistem telemedicine yang memerlukan pemrosesan citra retina dengan latensi rendah.

Untuk penelitian masa depan, direkomendasikan beberapa arah pengembangan: (1) Evaluasi cross-dataset untuk mengukur generalisasi model; (2) Eksperimen dengan arsitektur attention mechanism seperti CBAM atau squeeze-and-excitation untuk meningkatkan precision tanpa mengorbankan sensitivity; (3) Implementasi teknik post-processing seperti conditional random fields (CRF) untuk memperhalus output segmentasi; (4) Evaluasi pada hardware edge devices seperti NVIDIA Jetson atau perangkat mobile untuk mengukur kelayakan deployment pada perangkat dengan sumber daya sangat terbatas; dan (5) Eksplorasi teknik knowledge distillation untuk transfer knowledge dari model besar ke model yang lebih kompak.

Batasan

Penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pertama, eksperimen dilakukan hanya pada dataset DRIVE; generalisasi ke dataset retina lainnya seperti CHASE_DB1, STARE, atau HRF belum dievaluasi. Kedua, model teroptimasi menunjukkan penurunan precision yang mengindikasikan peningkatan false positives, yang dapat menjadi masalah pada aplikasi klinis tertentu. Ketiga, penggunaan GPU NVIDIA T4 pada Google Colab merupakan environment shared yang dapat menyebabkan variabilitas pada pengukuran waktu.

DAFTAR RUJUKAN

[1] A. Maas, A. Hannun, and A. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," 2013. Available: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

[2] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, doi: https://doi.org/10.1109/cvpr.2018.00286

[3] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," *arXiv.org*, Nov. 27, 2015, doi: https://doi.org/10.48550/arXiv.1505.00853

[4] 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 Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, Apr. 2004, doi: https://doi.org/10.1109/tmi.2004.825627

[5] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," *Graphics Gems*, pp. 474–485, 1994, doi: https://doi.org/10.1016/b978-0-12-336156-1.50061-6

[6] M. D. Abramoff, M. K. Garvin, and M. Sonka, "Retinal Imaging and Image Analysis," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 169–208, 2010, doi: https://doi.org/10.1109/rbme.2010.2084567

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, doi: https://doi.org/10.1109/cvpr.2018.00474

[8] N. Patton et al., "Retinal image analysis: Concepts, applications and potential," *Progress in Retinal and Eye Research*, vol. 25, no. 1, pp. 99–127, Jan. 2006, doi: https://doi.org/10.1016/j.preteyeres.2005.07.001

[9] O. Oktay et al., "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv.org*, May 20, 2018, doi: https://doi.org/10.48550/arXiv.1804.03999

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *arXiv (Cornell University)*, May 2015, doi: https://doi.org/10.48550/arxiv.1505.04597

[11] P. Liskowski and K. Krawiec, "Segmenting Retinal Blood Vessels With Deep Neural Networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 11, pp. 2369–2380, Nov. 2016, doi: https://doi.org/10.1109/tmi.2016.2546227

[12] Paulius Micikevicius et al., "Mixed Precision Training," Oct. 2017, doi: https://doi.org/10.48550/arxiv.1710.03740

[13] surajkumarpal, "Retina-Blood-Vessel-Segmentation-U-Net," *Kaggle.com*, Jan. 14, 2024. https://www.kaggle.com/code/surajkumarpal/retina-blood-vessel-segmentation-u-net (accessed Jun. 09, 2026).

[14] Y. Zhao and L. Lin, "A Lightweight U-Net for Medical Image Segmentation," *2022 Photonics & Electromagnetics Research Symposium (PIERS)*, pp. 1–5, Apr. 2024, doi: https://doi.org/10.1109/piers62282.2024.10618503

[15] Z. Liu, Mohd Shahrizal Sunar, T. S. Tan, and W. Hazabbah, "Deep learning for retinal vessel segmentation: a systematic review of techniques and applications," *Medical & Biological Engineering & Computing*, Feb. 2025, doi: https://doi.org/10.1007/s11517-025-03324-y

[16] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1–1, 2019, doi: https://doi.org/10.1109/tmi.2019.2959609

[17] Zahangir Alom, Md. Mahmudul Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation," Feb. 2018, doi: https://doi.org/10.48550/arxiv.1802.06955