

AGU-NET: *Attention Ghost U-Net* Untuk Segmentasi Penyakit Polip Berbasis Citra Biomedis

Muhammad Anang Fathur Rohman¹, Hafidh Muhammad Akbar², Alfiki Diastama Afan Firdaus³, Heri Prasetyo⁴

¹Fakultas Teknologi Informasi dan Sains Data Universitas Sebelas Maret, Surakarta, 57126 Indonesia, anangmuhammad245@student.uns.ac.id

²Fakultas Teknologi Informasi dan Sains Data Universitas Sebelas Maret, Surakarta, 57126 Indonesia, email: hafidhmuhammadakbar1@student.uns.ac.id

³Fakultas Teknologi Informasi dan Sains Data Universitas Sebelas Maret, Surakarta, 57126 Indonesia, email: alfiki@student.uns.ac.id

⁴Fakultas Teknologi Informasi dan Sains Data Universitas Sebelas Maret, Surakarta, 57126 Indonesia, email: heri.prasetyo@staff.uns.ac.id

Corresponding Author: Muhammad Anang Fathur Rohman

INTISARI — Kolonoskopi adalah salah satu teknik untuk mendeteksi polip yang merupakan salah satu penyebab penyakit kanker kolorektal (CRC). Kolonoskopi memanfaatkan segmentasi citra polip untuk memperoleh informasi yang berharga dalam proses diagnosis dan pembedahan. Teknik segmentasi yang dilakukan dapat melibatkan metode pembelajaran mendalam (*deep learning*) di dalamnya. Perkembangan metode *deep learning* di bidang visi komputer (*computer vision*) mendorong banyak peneliti melakukan eksperimen yang tidak hanya berfokus pada peningkatan performa metode, namun juga pada peningkatan efisiensi memori pada model. Model dengan efisiensi memori yang tinggi dapat membuat biaya implementasi lebih terjangkau, karena model yang ringan tidak membutuhkan sumber daya (*resource*) yang besar. Metode-metode sebelumnya secara umum memiliki performa yang baik dalam melakukan segmentasi penyakit polip, namun memiliki jumlah parameter yang besar, sehingga efisiensi memorinya relatif rendah. Berdasarkan permasalahan tersebut, penulis mengajukan metode *Attention Ghost U-Net* untuk segmentasi citra medis polip. Metode ini menggunakan kerangka dasar U-Net dengan menggunakan *Residual Network*, *Bottleneck*, dan *Attention* untuk mendapatkan kinerja yang sebanding dengan metode-metode sebelumnya. Metode yang diajukan memiliki jumlah parameter yang lebih rendah daripada metode-metode sebelumnya tanpa mengabaikan tingkat akurasi. Hasil uji pada *dataset* CVC-ClinicDB dan Kvasir-SEG menunjukkan bahwa metode yang diusulkan memiliki parameter yang kecil, yaitu 1,17 juta. Metode yang diusulkan memberikan tingkat kinerja F1-Score sebesar 85,55%, dan nilai Intersection-over-Union (IoU) sebesar 78,22% untuk *dataset* Kvasir-SEG. Untuk *dataset* CVC-ClinicDB, tingkat kinerja F1-Score yang diberikan sebesar 92,13% dan IoU sebesar 86,52%. Dengan hasil tersebut, metode ini berhasil mengungguli beberapa metode sebelumnya, di antaranya ResU-Net, ResU-Net++, U-Net, U-Net++, SFA, ColoneSegNet, dan DoubleUNet dengan jumlah parameter yang lebih rendah.

KATA KUNCI — Segmentasi Citra Polip, *Deep Learning*, *Attention Ghost U-Net*, Efisiensi Memori

I. PENDAHULUAN

Kanker kolorektal (CRC) merupakan jenis kanker umum yang dapat dicegah melalui skrining kolonoskopi[12]. Skrining ini penting karena dapat mengidentifikasi polip yang bisa berkembang menjadi CRC[7][8]. Polip secara medis dikenali sebagai pertumbuhan jaringan pada usus besar atau rektum yang mungkin berubah menjadi kanker seiring waktu[18]. Deteksi dini polip penting, tetapi segmentasi citra polip dalam gambar medis sering sulit dan memakan waktu[2]. Segmentasi citra merupakan proses membagi citra menjadi bagian-bagian yang memiliki karakteristik serupa[9].

Penggunaan segmentasi citra medis, seperti pada polip, memiliki banyak manfaat dalam perawatan pasien, riset klinis, dan pengembangan teknologi[16][14]. Beberapa metode *deep learning*, seperti metode U-Net[15], ResU-Net[19], U-Net++[20], SFA[17], ResU-Net++^[5], ColoneSegNet^[6], dan DoubleUNet^[4], telah digunakan untuk segmentasi citra polip, tetapi seringkali memerlukan banyak parameter yang mempengaruhi kinerja karena sumber daya komputasi yang terbatas[11][3]. Oleh karena itu, kami mengusulkan AGU-NET (*Attention Ghost U-Net*) sebagai model *deep learning* baru untuk segmentasi polip dengan tujuan meningkatkan kinerja dan mengurangi kebutuhan parameter.

Metode AGU-NET diuji pada data citra medis polip dari *dataset* CVC-ClinicDB[10] dan Kvasir-SEG^[6] untuk mengatasi masalah pada metode sebelumnya dan meningkatkan performa.

II. METODE PENELITIAN

Penelitian ini mengikuti lima tahap metodologi penelitian, yaitu mengumpulkan dan mengatur dataset, merancang neural network, *hyperparameter tuning* dan pelatihan neural network, evaluasi neural network, dan membandingkan dengan metode yang sudah ada.

A. MENGUMPULKAN DAN MENGATUR DATASET

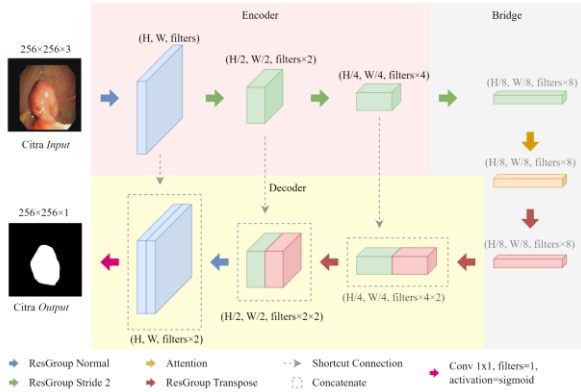
Penelitian ini menggunakan *dataset* CVC-ClinicDB^[10] dan Kvasir-SEG^[6]. *Dataset* CVC-ClinicDB terdiri dari 612 citra polip dengan resolusi 288×368 piksel, sedangkan *dataset* Kvasir-SEG terdiri dari 1000 citra polip dengan dimensi yang bervariasi, mulai dari 332×487 hingga 1920×1072 piksel. Kedua *dataset* ini telah dibagi menjadi *training set*, *validation set*, dan *test set* dengan rasio 80%:10%:10%.

Untuk menyamakan dimensi citra, dilakukan proses *resizing* citra menjadi ukuran 256×256 piksel. Selain itu, dilakukan augmentasi data pada *training set* untuk memperbanyak jumlah data dan mencegah *overfitting*. Augmentasi yang dilakukan meliputi *horizontal* dan *vertical flip*, serta rotasi. Setelah proses augmentasi, jumlah data pada *training set* menjadi 1960 untuk *dataset* CVC-ClinicDB dan 3200 untuk *dataset* Kvasir-SEG.

B. MERANCANG NEURAL NETWORK

Penelitian ini mengusulkan metode yang disebut AGU-NET, yang mengambil U-Net sebagai inspirasi dalam perancangan arsitektur. AGU-NET terdiri dari *encoder*, *decoder*, dan *bridge*. Pada arsitektur yang diajukan ini hanya

menggunakan tiga kedalaman saja. Hal ini ditujukan agar parameter yang dihasilkan sedikit. Selain itu, dilakukan modifikasi dengan mengganti *convolution* dengan *Residual Group* (ResGroup) yang merupakan implementasi *Improved ResNet* menggunakan *ghost bottleneck*. Arsitektur dari model *Attention Ghost U-Net* dapat dilihat pada Gambar 1.



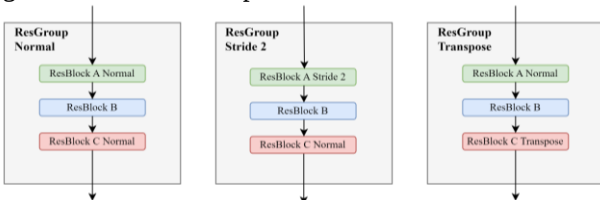
Gambar 1. Arsitektur *Attention Ghost U-Net*

Penelitian ini menggunakan *encoder* dengan 3 ResGroup (Normal, Stride 2, Stride 2) dan *bridge* (Stride 2, Attention, Transpose). Attention pada *bridge* digunakan untuk fokus pada aspek penting selama pelatihan. Decoder terdiri dari 2 ResGroup Transpose diikuti oleh ResGroup Normal.

Di akhir jaringan neural, terdapat *layer convolution* dengan kernel 1×1 , satu filter, dan aktivasi sigmoid. Hal ini dilakukan untuk mengurangi parameter, menghasilkan citra hitam putih, dan mengubah nilai piksel menjadi rentang 0-1. Jika nilai piksel lebih dari 0,5, dianggap sebagai bagian dari polip; jika kurang dari 0,5, maka sebaliknya.

1) RESIDUAL GROUP

Penelitian ini menggunakan tiga jenis Residual Group (ResGroup), seperti pada Gambar 2: ResGroup Normal, ResGroup Stride 2, dan ResGroup Transpose. Setiap ResGroup terdiri dari tiga jenis Residual Block (ResBlock) yaitu ResBlock A, ResBlock B, dan ResBlock C. ResGroup Stride 2 dibedakan dengan adanya penggunaan ResBlock A dengan stride 2, sementara pada ResGroup Transpose, ResBlock C diganti dengan ResBlock C Transpose.

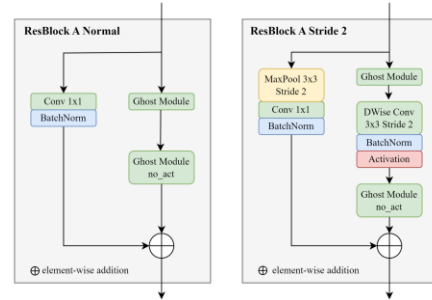


Gambar 2. Susunan ResGroup: ResGroup Normal, ResGroup Stride 2, dan ResGroup Transpose

2) RESIDUAL BLOCK

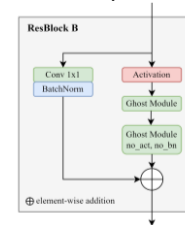
Penelitian ini menggunakan dua jenis Residual Block (ResBlock) A: ResBlock A dan ResBlock A Stride 2, terinspirasi dari Start ResBlock dalam *improved ResNet*. Kedua jenis ResBlock A ini menggunakan konsep *ghost bottleneck* dengan *ghost module*, *ghost module* tanpa *activation function* (*no_act*), dan *identity mapping*. Keduanya melibatkan konvolusi dan *batch normalization* untuk memproses *identity mapping*. Namun, pada ResBlock A Stride 2, terdapat tambahan operasi MaxPool 3×3 dengan stride 2 sebelum konvolusi, serta

konvolusi *depthwise* 3×3 dengan stride 2. ResBlock A Stride 2 digunakan sebagai proses downsampling dalam arsitektur U-Net, seperti Gambar 3.



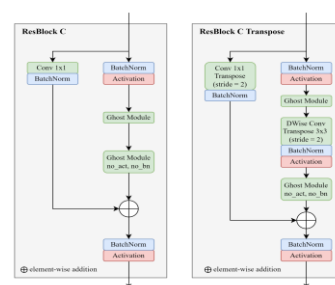
Gambar 3. Susunan ResBlock: ResBlock A Normal dan ResBlock A Stride 2

ResBlock B yang digunakan pada metode ini disusun berdasarkan *Middle ResBlock* pada *improved ResNet*. Sama seperti ResBlock A, ResBlock B juga dirancang dengan *ghost bottleneck*. ResBlock B disusun dengan *ghost module* dan *ghost module* tanpa *batch normalization* (*no_bn*) dan tanpa *activation function* (*no_act*), serta penambahan *identity mapping* ke output dari *ghost module*. Pada ResBlock B, seperti pada Gambar 4, terdapat penambahan *activation function* sebelum *ghost module*.



Gambar 4. Susunan ResBlock B

Pada penelitian ini, ResBlock C memiliki dua tipe: ResBlock C dan ResBlock C Transpose. Kedua tipe ResBlock C ini didasarkan pada *End ResBlock* dalam *improved ResNet*. Kedua tipe ResBlock C ini dibangun menggunakan *ghost bottleneck*, yang terdiri dari *ghost module* dan *ghost module* tanpa *batch normalization* (*no_bn*) dan *activation function* (*no_act*), dengan penambahan *identity mapping* ke output dari *ghost module*. Selain itu, terdapat *batch normalization* dan *activation function* sebelum *ghost module*, serta setelah penambahan *identity mapping*.

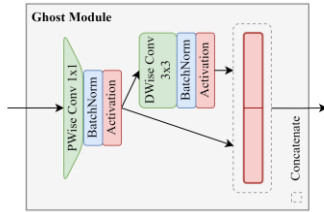


Gambar 5. Susunan ResGroup C: ResBlock C dan ResBlock C Transpose

Pada ResBlock C, seperti pada Gambar 5, terdapat *convolution* dan *batch normalization* yang digunakan untuk memproses *identity mapping*. Namun, pada ResBlock C Transpose, digunakan *convolution transpose* 3×3 dengan stride 2. Selain itu, pada ResBlock C Transpose terdapat *depthwise convolution transpose* 3×3 dengan stride 2. ResBlock C Transpose ini digunakan sebagai proses *upsampling* dalam arsitektur U-Net.

3) GHOST MODULE

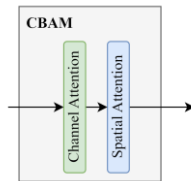
Dalam penelitian ini, digunakan ghost module yang dibuat oleh K. Han et al[13]. Ghost module menggabungkan *point-wise convolution* dan *depth-wise convolution* dengan *batch normalization* dan *activation function*. *Point-wise convolution* menghasilkan setengah feature map dari input, lalu hasilnya diproses oleh *depth-wise convolution*, dan hasil kedua konvolusi digabungkan. Ghost module berguna untuk mengurangi parameter untuk meningkatkan efisiensi komputasi. Ilustrasinya dapat dilihat pada Gambar 6.



Gambar 6. Susunan Ghost Module

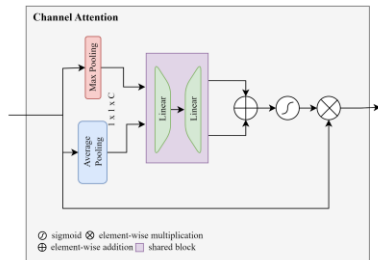
4) ATTENTION MODULES

Pada penelitian ini ditambahkan *attention* pada *bridge* untuk mengoptimalkan kerja dari AGU-NET. *Attention* digunakan agar model fokus pada bagian-bagian tertentu dari data input yang dianggap suatu informasi yang penting. *Attention modules* memiliki banyak jenis, antara lain: CBAM (Convolutional Block Attention Module), CAM (Channel Attention Module), dan SAM (Spatial Attention Module)



Gambar 7. Susunan Module CBAM

CBAM (Convolutional Block Attention Module) merupakan *multi-layered attention* yang tersusun dari perpaduan Channel Attention Module (CAM) dan Spatial Attention Module (SAM). Kedua modul ini bekerja bersama memperbaiki representasi fitur dalam jaringan saraf konvolusi dengan menyesuaikan *channel* dan spasial secara adaptif. Ilustrasi dari kerangka model dengan CBAM dapat dilihat pada Gambar 7.



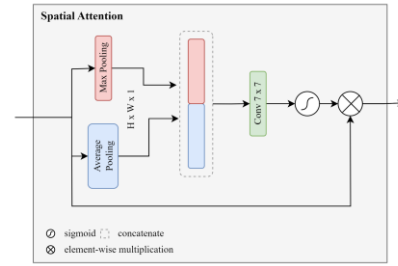
Gambar 8. Susunan Module Channel Attention

Seperti terlihat pada Gambar 8, Channel attention module (CAM) mengolah *feature map* dengan *average pooling* dan *global max pooling* yang untuk menghasilkan dua deskriptor konteks spasial, yaitu F^c_{avg} dan F^c_{max} . Selanjutnya, *feature map* akan dimasukkan ke dalam *bottleneck Multi-Layer Perceptron* (MLP) yang terdiri dari dua lapisan *fully connected layer*. Lapisan pertama berfungsi untuk mengurangi jumlah input *feature map* dengan rasio tertentu, kemudian jumlahnya dikembalikan pada lapisan kedua. Hasilnya kemudian melewati

fungsi aktivasi sigmoid (σ) untuk mendapatkan nilai antara 0 hingga 1 yang menunjukkan tingkat kepentingan informasi yang dibawa setiap *channel*. Nilai tersebut digunakan dalam operasi perkalian dengan setiap elemen pada input *feature map*, sehingga menghasilkan *feature map* dengan bobot yang sesuai misal W_0 dan W_1 . Setelah transformasi dilakukan, hasil MLP dari masing-masing operasi *pooling* dijumlahkan untuk menghasilkan $M_c(F)$ (output akhir). Perhitungan *channel attention* dilihat pada persamaan (3.1).

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (3.1)$$

$$= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max})))$$



Gambar 9. Susunan Module Spatial Attention

Berbeda dengan *channel attention*, *spatial attention* memfokuskan pada bagian atau area mana yang dianggap informatif. *Spatial attention* terdiri dari 3 langkah berurutan. Ilustrasi dari kerangka Spatial Attention Module (SAM) dapat dilihat pada Gambar 9. Langkah yang pertama *tensor* input dengan dimensi $c \times h \times w$ dibagi menjadi 2 *channel* F^s_{max} dan F^s_{avg} dengan dimensi $1 \times h \times w$. Setiap *channel* merepresentasikan *max-pooling* dan *average-pooling* melewati tiap *channel*. Hal ini digunakan sebagai input untuk lapisan konvolusi dengan *kernel size* 7×7 ($f^{7 \times 7}$) yang menghasilkan *feature map* dengan 1 *channel* dengan dimensinya menjadi $1 \times h \times w$. Jadi, lapisan konvolusi ini digunakan untuk mempertahankan dimensi spasial dengan menggunakan *padding*. Output dari dari konvolusi kemudian diteruskan ke lapisan aktivasi sigmoid (σ). Sigmoid akan memetakan semua nilai ke rentang antara 0 dan 1. Kemudian setelah itu dilakukan perkalian antara *attention map* dan input *feature map*. Secara singkat, perhitungan *spatial attention* dapat dilihat pada persamaan (3.2).

$$M_s(F) = \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \quad (3.2)$$

$$= \sigma(f^{7 \times 7}([F^s_{avg}; F^s_{max}]))$$

C. HYPERPARAMETER TUNING DAN PELATIHAN NEURAL NETWORK

Langkah selanjutnya adalah melakukan *hyperparameter tuning* agar mendapatkan *hyperparameter* yang terbaik dan efisien untuk mencapai kinerja yang optimal[1]. *Hyperparameter* merupakan nilai yang diatur untuk melakukan proses pelatihan pada model. Dalam penelitian ini, *hyperparameter tuning* akan dilakukan untuk pengujian *hyperparameter* jumlah *kernel*, jenis *activation*, jenis *attention*, jumlah *attention*, dan nilai bias. Setelah mendapatkan *hyperparameter* yang paling optimal, dilakukan pelatihan terhadap *neural network* yang telah disusun. Citra input yang digunakan berupa citra polip berwarna dari set data CVC-ClinicDB^[10] dan Kvasir-SEG^[6] yang diambil melalui endoskopi. Model akan menghasilkan citra tersegmentasi dengan satu *channel*. Citra prediksi yang dihasilkan kemudian

dicocokkan dengan citra target untuk dihitung dengan *loss function* dan mengetahui seberapa baik kinerja model.

D. EVALUASI NEURAL NETWORK

Setelah dilakukan pelatihan, AGU-NET akan dievaluasi untuk mengetahui kelayakan model. Pengujian dapat menggunakan beberapa metrik evaluasi, seperti F1-Score dan IoU (*Intersection over Union*). F1-Score dan IoU dihitung untuk mengukur sejauh mana citra prediksi AGU-NET cocok dengan citra target yang sebenarnya. Skala nilai F1-Score dan IoU berkisar antara 0% hingga 100%. Nilai yang lebih tinggi menunjukkan performa AGU-NET yang lebih baik. F1-Score dan IoU dapat dihitung dengan menggunakan persamaan (3.3) dan (3.4).

$$F1 - Score = \frac{2TP}{2TP + FP + FN} \quad (3.3)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (3.4)$$

TP (*True Positive*) adalah jumlah piksel yang benar-benar cocok antara prediksi dan target, FP (*False Positive*) adalah jumlah piksel yang salah diprediksi sebagai objek positif, dan FN (*False Negative*) adalah jumlah piksel yang salah diprediksi sebagai objek negatif.

Selain itu, evaluasi juga dapat mempertimbangkan jumlah parameter yang digunakan oleh AGU-NET. Jumlah parameter mencerminkan kompleksitas dan ukuran model, serta dapat menjadi indikator efisiensi memori. Semakin sedikit jumlah parameter, semakin efisien model AGU-NET dalam hal penggunaan sumber daya memori.

Dengan melakukan evaluasi menggunakan metrik-metrik ini, kita dapat mendapatkan pemahaman yang lebih baik tentang performa AGU-NET setelah pelatihan dan sejauh mana model tersebut cocok dengan tugas segmentasi pada *dataset* CVC-Clinic DB dan Kvasir-SEG.

E. MEMBANDINGKAN DENGAN METODE YANG SUDAH ADA

Perbandingan dilakukan berdasarkan hasil nilai F1-Score pada set pengujian, IoU pada set pengujian, dan jumlah parameter. AGU-NET akan dibandingkan dengan metode-metode lain seperti U-Net^[15], ResU-Net^[19], U-Net++^[20], SFA^[17], ResU-Net++^[5], ColoneSegNet^[6], dan DoubleUNet^[4]. Dalam perbandingan ini, kita dapat melihat apakah AGU-NET mampu mengungguli metode-metode tersebut. Perbandingan ini memberikan wawasan posisi AGU-NET dalam ranah segmentasi citra biomedis.

III. HASIL DAN PEMBAHASAN

Pada bagian ini akan ditunjukkan hasil pelatihan model yang diajukan dengan menggunakan model AGU-NET (*Attention Ghost U-Net*) pada *dataset* Kvasir-SEG dan CVC-ClinicDB.

A. SITUASI PERCOBAAN

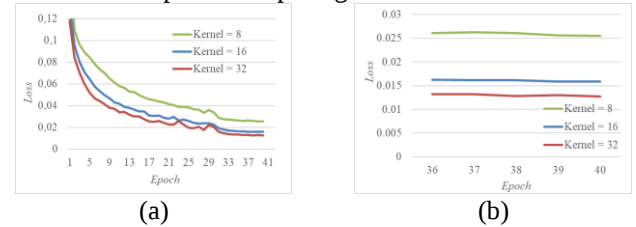
Percobaan dilakukan pada *website* Kaggle menggunakan akselerator TPU v3-8 dan bahasa pemrograman Python dengan *framework* Tensorflow. *Dataset* yang digunakan untuk melakukan training model adalah *dataset* dari Kvasir-SEG dan CVC-ClinicDB dengan *hyperparameter* seperti pada Tabel I.

TABEL I
HYPERPARAMETER

Hyperparameter	Nilai
Ukuran Batch	64
Jumlah Kernel	8
Jumlah Epoch	40
Optimizer	Adam
Learning Rate	0.001 (awal) (dibagi 10 setelah epoch 30 setiap 5 epoch)
Activation Function	ReLU
Jenis Attention	Channel
Rasio Attention	4
Penggunaan Nilai Bias	True
Jumlah Residual Block	1
Loss Function	Dice Loss

B. PERCOBAAN JUMLAH KERNEL

Percobaan pertama adalah menguji pengaruh jumlah kernel (8, 16, 32) terhadap performa *neural network*. Hasil percobaan jumlah kernel dapat dilihat pada gambar 10.

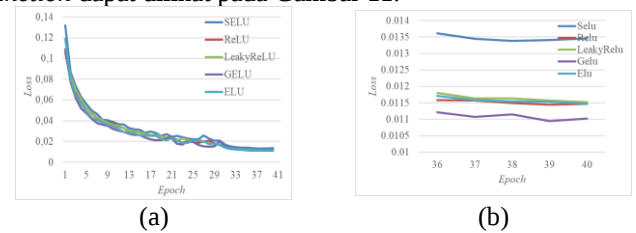


Gambar 10. Hasil percobaan jenis *activation function*: a) semua epoch, b) 5 epoch terakhir

Dari hasil percobaan pada Gambar 10, jumlah *kernel* mempengaruhi nilai *loss neural network*. Jumlah *kernel* 32 memiliki *loss* terendah, maka akan digunakan pada percobaan selanjutnya.

C. PERCOBAAN JENIS ACTIVATION FUNCTION

Percobaan kedua adalah menguji pengaruh jenis *activation function* (ReLU, LeakyReLU, ELU, SELU, GELU) terhadap performa *neural network*. Hasil percobaan jenis *activation function* dapat dilihat pada Gambar 11.

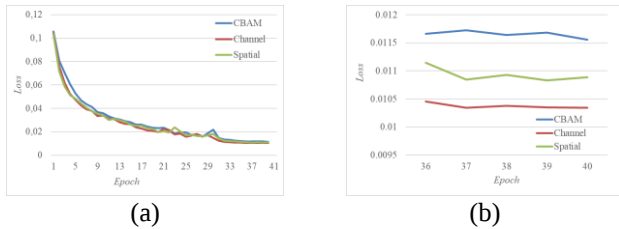


Gambar 11. Hasil percobaan jenis *activation function*: a) semua epoch, b) 5 epoch terakhir

Dari hasil percobaan pada Gambar 11, jenis *activation function* mempengaruhi nilai *loss neural network*. jenis *activation function* GELU memiliki *loss* terendah, maka akan digunakan pada percobaan selanjutnya.

D. PERCOBAAN JENIS ATTENTION

Percobaan ketiga adalah menguji pengaruh jenis *attention* (CBAM, *channel attention*, *spatial attention*) terhadap performa *neural network*. Hasil percobaan jenis *attention* dapat dilihat pada Gambar 12.

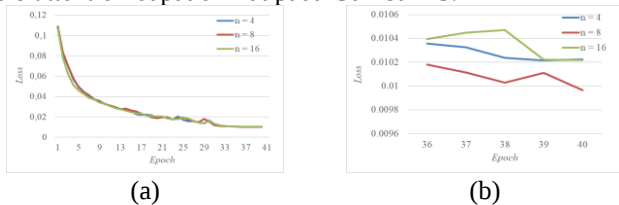


Gambar 12. Hasil percobaan jenis *attention*: a) semua *epoch*, b) 5 *epoch* terakhir

Dari hasil percobaan pada Gambar 12, jenis *attention* mempengaruhi nilai *loss neural network*. Jenis *attention* berupa *channel attention* memiliki *loss* terendah, maka akan digunakan pada percobaan selanjutnya.

E. PERCOBAAN RASIO ATTENTION

Percobaan keempat adalah menguji pengaruh rasio *attention* (4, 8, 16) terhadap performa *neural network*. Hasil percobaan rasio *attention* dapat dilihat pada Gambar 13.

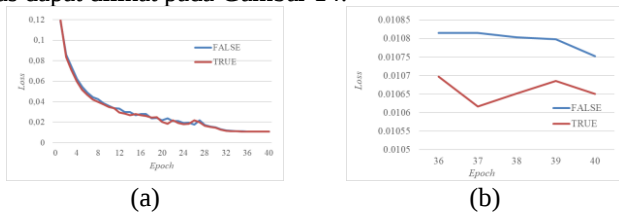


Gambar 13. Hasil percobaan rasio *attention*: a) semua *epoch*, b) 5 *epoch* terakhir

Dari hasil percobaan pada Gambar 13, rasio *attention* mempengaruhi nilai *loss neural network*. Rasio *attention* sebesar 8 memiliki *loss* terendah, maka akan digunakan pada percobaan selanjutnya.

F. PERCOBAAN PENGGUNAAN NILAI BIAS

Percobaan kelima adalah menguji pengaruh nilai bias (*True*, *False*) terhadap performa *neural network*. Hasil percobaan nilai bias dapat dilihat pada Gambar 14.



Gambar 14. Hasil percobaan nilai bias: a) semua *epoch*, b) 5 *epoch* terakhir

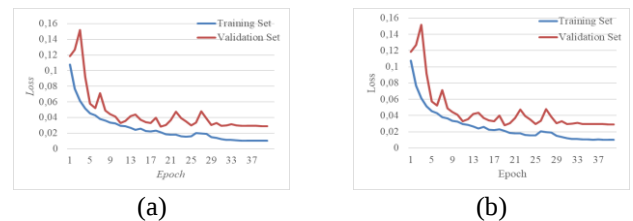
Dari hasil percobaan pada Gambar 14, nilai bias mempengaruhi nilai *loss neural network*. Nilai bias berupa *True* memiliki *loss* terendah, maka akan digunakan pada percobaan selanjutnya.

G. HASIL PERCOBAAN

Setelah percobaan, terdapat perubahan *hyperparameter* (jumlah *kernel*, *activation function*, dan rasio *attention*) untuk mendapatkan performa terbaik pada *neural network*. Tabel II menjelaskan *hyperparameter* performa terbaik dengan total parameter 1.178.161, seperti pada Gambar 15. *Neural network* ini selanjutnya akan dilatih menggunakan dataset CVC-ClinicDB dan Kvasir-SEG.

TABEL II
HYPERPARAMETER DENGAN PERFORMA TERBAIK

Hyperparameter	Nilai
Ukuran Batch	64
Jumlah Kernel	32
Jumlah Epoch	40
Optimizer	Adam
Learning Rate	0.001 (awal) (dibagi 10 setelah epoch 30 setiap 5 epoch)
Activation Function	GELU
Jenis Attention	Channel Attention
Rasio Attention	8
Penggunaan Nilai Bias	True
Loss Function	Dice Loss



Gambar 15. Hasil percobaan dengan dataset: a) Kvasir-SEGh, b) CVC-ClinicDB

Model *neural network* dilatih dan diuji pada dataset Kvasir-SEG dan CVC-ClinicDB untuk mendapatkan F1-Score dan IoU yang terdapat pada tabel V. Model juga dapat digunakan untuk membuat prediksi citra pada dataset, contohnya terlihat di tabel III dan IV. Secara umum perbandingan hasil dapat dilihat pada Tabel V.

TABEL III
CITRA HASIL PENGUJIAN DATASET KVASIR-SEG

Citra Input	Citra Label	Citra Prediksi	F1-Score/IoU
			98,33%/96,72%
			98,30%/96,65%
			97,73%/95,56%

TABEL IV
CITRA HASIL PENGUJIAN DATASET CVC-CLINICDB

Citra Input	Citra Label	Citra Prediksi	F1-Score/IoU
			98,80%/97,62%
			97,66%/95,43%
			97,45%/95,03%

TABEL V
HASIL PENGUJIAN

Dataset	F1-Score	IoU
Kvasir-SEG	85,55%	78,22%
CVC-ClinicDB	92,13%	86,52%

H. PERBANDINGAN DENGAN METODE SEBELUMNYA

Nilai F1-Score dan IoU yang dihasilkan pada proses pengujian model *neural network* dengan *test set* dari dataset Kvasir-SEG dan CVC-ClinicDB akan dibandingkan dengan hasil pengujian pada metode-metode sebelumnya. Performa algoritma *Attention Ghost U-Net* akan dibandingkan dengan U-Net[15], ResU-Net[19], U-Net++[20], SFA[17], ResU-Net++[5], ColoneSegNet[6], dan DoubleUNet[4]. Hasil perbandingan dapat dilihat pada tabel VI.

TABEL VI
PERBANDINGAN DENGAN METODE SEBELUMNYA

Metode	CVC-ClinicDB		Kvasir-SEG		Jumlah Parameter
	F1-Score	IoU	F1-Score	IoU	
SFA[17]	70,0%	60,7%	72,3%	61,1%	n/a
ResU-Net[19]	77,9%	n/a	79,1%	n/a	80,6 juta
U-Net++[20]	79,4%	72,9%	82,1%	74,3%	90,4 juta
U-Net[15]	82,3%	75,5%	81,8%	74,6%	43,93 juta
ResU-Net++[5]	79,6%	79,6%	81,3%	79,30%	4,06 juta
ColoneSegNet[6]	88,62%	82,48%	79,20%	69,80%	5,01 juta
DoubleUNet[4]	92,39%	86,11%	81,29%	73,32%	29,3 juta
Metode usulan (AGU-NET)	92,13%	86,52%	85,55%	78,22%	1,17 juta

Pada tabel VI dapat disimpulkan bahwa tingkat akurasi dari metode yang diajukan relatif lebih tinggi dari metode yang diajukan sebelumnya dan jumlah parameter yang dihasilkan dari metode ini berkurang drastis menjadi 1,17 juta.

IV. KESIMPULAN

Pada penelitian ini, telah dihasilkan sebuah arsitektur *convolutional neural network* dengan hasil akurasi dan kinerja yang lebih baik, yaitu *Attention Ghost U-Net* dengan jumlah parameter yang relatif kecil karena menggunakan dasar U-Net yang sering digunakan untuk permasalahan *computer vision*. Penggantian *convolution* dengan susunan *improved ResNet* pada *ghost bottleneck* juga meningkatkan performa dari U-Net. Penggunaan *ghost bottleneck* juga menghasilkan jumlah parameter yang lebih sedikit sehingga AGU-NET memiliki tingkat efisiensi memori yang tinggi daripada metode-metode yang ada sebelumnya. Model dilatih selama 40 *epoch* menggunakan dataset CVC-ClinicDB dan Kvasir-SEG. F1-Score dan IoU yang dihasilkan oleh AGU-NET pada dataset CVC-ClinicDB secara berurutan sebesar 92,13% dan 86,52%, sedangkan dataset Kvasir-SEG menghasilkan F1-Score sebesar 85,55% dan IoU sebesar 78,22%. Jumlah parameter yang sedikit yaitu 1,17 juta membuat metode yang diajukan memiliki tingkat efisiensi memori lebih tinggi jika dibanding metode-metode sebelumnya, yaitu U-Net[15], ResU-Net[19], U-Net++[20], SFA[17], ResU-Net++[5], ColoneSegNet[6], dan DoubleUNet[4].

REFERENSI

- [1] A. I. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. R. Griffiths, A. M. Maraval, H. Jianye, J. Wang, J. Peters, & H. Bou-Ammar, "HEBO: Pushing The Limits of Sample-Efficient Hyper-parameter Optimisation," *Journal of Artificial Intelligence Research*, vol. 74, pp. 1269–1349, 2022. doi:10.1613/jair.1.13643.
- [2] C. Cao, R. Wang, Y. Yu, H. Zhang, Y. Yu, & C. Sun, "Gastric polyp detection in gastroscopic images using Deep Neural Network," *PLOS ONE*, vol. 16, no. 4, 2021. doi:10.1371/journal.pone.0250632.
- [3] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yui, H. Zhang, & Y. Zhang, "," *Mobile Information Systems*, vol. 2020, pp. 1–19, 2020. doi:10.1155/2020/8454327.
- [4] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, & H. D. Johansen, "DoubleU-net: A deep convolutional neural network for medical image segmentation," *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, 2020. doi:10.1109/cbms49503.2020.00111.
- [5] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. D. Lange, P. Halvorsen, & H. D. Johansen, "ResUNet++: An Advanced Architecture for Medical Image Segmentation," *2019 IEEE International Symposium on Multimedia (ISM)*, 2019. doi:10.1109/ism46123.2019.00049.
- [6] D. Jha, S. Ali, N. K. Tomar, H. D. Johansen, D. Johansen, J. Rittscher, M. A. Riegler, & P. Halvorsen, "Real-time polyp detection, localization and segmentation in colonoscopy using Deep Learning," *IEEE Access*, vol. 9, pp. 40496–40510, 2021. doi:10.1109/access.2021.3063716.
- [7] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, & L. Shao, "PraNet: Parallel reverse attention network for polyp segmentation," *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pp. 263–273, 2020. doi:10.1007/978-3-030-59725-2_26.
- [8] F. Haggard & R. Boushey, "Colorectal cancer epidemiology: Incidence, mortality, survival, and risk factors," *Clinics in Colon and Rectal Surgery*, vol. 22, no. 04, pp. 191–197, 2009. doi:10.1055/s-0029-1242458.
- [9] H. Liao, S. K. Zhou, & J. Luo, "Introduction," *Deep Network Design for Medical Image Computing*, pp. 1–9, 2023. doi:10.1016/b978-0-12-824383-1.00008-3.
- [10] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodriguez, & F. Vilarinho, "WM-Dova Maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015. doi:10.1016/j.compmedimag.2015.02.007.
- [11] J. Cao, B. Chen, C. Sun, L. Hu, S. Wu, & X. Peng, "Understanding performance problems in Deep Learning Systems," *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022. doi:10.1145/3540250.3549123.
- [12] J. Silva, A. Hiastace, O. Romain, X. Dray, & B. Granado, "Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 2, pp. 283–293, 2013.
- [13] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, & C. Xu, "Ghostnet: More features from Cheap Operations," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. doi:10.1109/cvpr42600.2020.00165.
- [14] L. Liu, J. Cheng, Q. Quantum F.-X. Wu, Y.-P. Wang, & J. Wang, "A survey on U-shaped networks in medical image segmentations," *Neurocomputing*, vol. 409, pp. 244–258, 2020. doi:10.1016/j.neucom.2020.05.070.
- [15] O. Ronneberger, "Invited talk: U-net convolutional networks for biomedical image segmentation," *Informatik aktuell*, pp. 3–3, 2017.
- [16] R. Merjula and J. Chandra, "Classification of myocardial ischemia in delayed contrast enhancement using machine learning," *Intelligent Data Analysis for Biomedical Applications*, pp. 209–235, 2019. doi:10.1016/b978-0-12-815553-0.00011-2.
- [17] Y. Fang, C. Chen, Y. Yuan, & K. Tong, "Selective feature aggregation network with area-boundary constraints for polyp segmentation," *Lecture Notes in Computer Science*, pp. 302–310, 2019.
- [18] Y.-M. Lee, S.-A. Kim, G.-S. Chou, S.-Y. Park, S. W. Jeno, H. S. Lee, S.-J. Lee, S. Heo, & D.-H. Lee, "Association of colorectal polyps and cancer with low-dose persistent organic pollutants: A case-control study," *PLOS ONE*, vol. 13, no. 12, 2018. doi:10.1371/journal.pone.0208546.
- [19] Z. Zhang, Q. Liu, & Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018. doi:10.1109/lgrs.2018.2802944.
- [20] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, & J. Liang, "UNet++: A nested U-Net Architecture for Medical Image segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, 2018. doi:10.1007/978-3-030-00889-5_1.