



Segmentasi Citra Pengelasan Kapal Menggunakan *Convolutional Neural Network*

Thomas Brian^{1*}, Anggarjuna Puncak Pujiputra¹, Putri Nur Rahayu²,
Immanuel Freddy Augustino², Parman², Iskia Ipan Dua' Bone²

¹Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya

²Jurusan Teknik Bangunan Kapal, Politeknik Perkapalan Negeri Surabaya

*Corresponding Author's e-mail: thomasbrian@ppns.ac.id

Article History:

Received: October 10, 2025

Revised: October 25, 2025

Accepted: October 31, 2025

Keywords:

CNN, Image Segmentation,
Nested Unet, Welding Spot

Abstract: *Welding inspection plays a critical role in the shipbuilding industry to ensure the integrity and quality of welded joints. However, the prevailing manual inspection procedures are inherently subjective, prone to bias, and result in inconsistent quality assessments. Therefore, there is a strong need for an automated and intelligent system capable of objectively detecting welding points. To address this, we propose an advanced segmentation model based on deep learning and computer vision techniques, specifically utilizing the enhanced Nested UNet architecture with extensive architectural modifications and comprehensive hyperparameter tuning. To further optimize the segmentation performance, we systematically compare different convolutional blocks integrated into the network architecture. The dataset used consists of 548 welding images. Each image is manually annotated using the VGG Image Annotator (VIA) application by marking the weld point areas as polygons. This research focuses on the development of a Nested UNet model, a deep learning-based image segmentation model for detecting weld points from previous models using the UNet architecture. During the training process, performance on both the training and validation datasets is continuously monitored and recorded. This results in several logs recording the training loss, validation loss, training IoU, and validation IoU for each of the three types of convolutional blocks used in the dense bottleneck. Our experimental evaluation shows that the use of VGG - Dense - VGG convolutional blocks in Nested UNet yields the highest performance, achieving a Training Dice score of 0.92970 and a Validation Dice score of 0.89695 on our collected dataset.*

Copyright © 2025, The Author(s).

This is an open access article under the CC-BY-SA license



How to cite: Brian, T., Pujiputra, A. P., Putri Nur Rahayu, Augustino, I. F., Parman, P., & Bone, I. I. D. (2025). Segmentasi Citra Pengelasan Kapal Menggunakan Convolutional Neural Network. *SENTRI: Jurnal Riset Ilmiah*, 4(10), 2511–2525. <https://doi.org/10.55681/sentri.v4i10.4767>

INTRODUCTION

Pengelasan merupakan teknik penyambungan fundamental yang banyak digunakan dalam industri pembuatan kapal, di mana satu kapal dapat memiliki ribuan titik las yang tersebar pada berbagai bagian struktur. Evaluasi kualitas sambungan las menjadi aspek krusial untuk memastikan kesesuaiannya dengan spesifikasi desain dan standar keselamatan, sekaligus menjamin integritas struktur dan keandalan operasional kapal [1]. Meskipun demikian, pemeriksaan las hingga kini masih banyak dilakukan secara manual melalui inspeksi visual, yang sangat bergantung pada persepsi manusia. Pendekatan ini tidak hanya memerlukan waktu dan tenaga yang besar, tetapi juga bersifat subjektif dan

tidak konsisten, sehingga kurang efektif diterapkan pada skala industri yang luas. Kondisi ini menjadi semakin tidak efisien ketika diterapkan pada skala industri besar seperti galangan kapal yang memiliki ribuan sambungan. Untuk mengatasi keterbatasan tersebut, metode visi komputer dan pembelajaran mendalam (*deep learning*) muncul sebagai alternatif yang menjanjikan. Perkembangan teknologi visi komputer dan pembelajaran mendalam (*deep learning*) menawarkan alternatif yang menjanjikan untuk mengatasi permasalahan tersebut. Secara khusus, *Convolutional Neural Network (CNN)* telah menunjukkan performa unggul dalam mendeteksi, mengklasifikasikan, dan melakukan segmentasi objek pada citra industri maupun medis. Secara khusus, *Convolutional Neural Networks (CNN)* telah banyak digunakan karena kemampuannya mengekstraksi fitur penting secara otomatis dari citra, sehingga memungkinkan deteksi yang akurat dan andal dalam berbagai tugas inspeksi visual, termasuk pengelasan [2]. Salah satu arsitektur yang paling banyak diadopsi untuk segmentasi citra adalah *UNet*, sebuah pendekatan pembelajaran mendalam yang terbukti sangat efektif dalam mengatasi berbagai tantangan pada aplikasi kecerdasan buatan [3]. Namun, *UNet* konvensional memiliki keterbatasan dalam menangani fitur yang kompleks dan informasi kontekstual multi-skala, terutama ketika batas objek tidak jelas atau ketika target segmentasi bervariasi dalam ukuran maupun tekstur. Lapisan *encoder* dan *decoder* sering kali tidak cukup untuk menangkap detail semantik secara menyeluruh. Dalam beberapa tahun terakhir, deteksi dan pelacakan objek menjadi tema penelitian utama di bidang visi komputer. Kemajuan perangkat keras komputasi serta tersedianya dataset beranotasi berskala besar telah mendorong peningkatan signifikan pada kinerja algoritma visi komputer berbasis *deep learning* dalam tugas inti seperti klasifikasi, deteksi, dan segmentasi citra [4]. Salah satu arsitektur *CNN* yang populer untuk segmentasi citra adalah *UNet*, yang terkenal karena kemampuannya memanfaatkan *skip connection* antara lapisan *encoder* dan *decoder* untuk mempertahankan informasi spasial penting. Berbagai metode telah dikembangkan untuk deteksi multi-objek, namun akurasi masih terkendala oleh tantangan seperti kerumunan objek, latar belakang yang kompleks, dan kemiripan visual antar objek [5]. Metode *deep learning* kemudian diperkenalkan untuk mengatasi keterbatasan teknik deteksi konvensional. Sebagai contoh, Aliyyah et al. [6] memanfaatkan *Faster R-CNN* untuk mendeteksi berbagai kelas helm pengendara motor.

Segmentasi semantik merupakan proses memberikan label kelas pada setiap piksel dalam citra, dengan mengelompokkan piksel yang memiliki karakteristik serupa ke dalam kategori yang sama [7]. Tidak seperti klasifikasi citra yang hanya memberikan satu label pada keseluruhan gambar, segmentasi semantik memungkinkan pemahaman detail pada tingkat piksel, sehingga penting untuk aplikasi seperti pencitraan medis, kendaraan otonom, dan pemetaan penggunaan lahan. Yang et al. [8] memperkenalkan *CvT-UNet*, sebuah model segmentasi semantik baru yang menggabungkan keunggulan *CNN* dan *Transformer*. Model ini memanfaatkan kemampuan *Transformer* dalam menangkap informasi kontekstual global dengan keunggulan *CNN* dalam mengekstraksi fitur spasial. Vadduri et al. [9] juga mengembangkan *Deep Attention UNet* untuk diagnosis katarak, yang menggunakan pendekatan *deep learning* mutakhir untuk meningkatkan segmentasi dan klasifikasi katarak pada citra retina. Arsitektur *UNet* sendiri telah banyak digunakan pada segmentasi citra biomedis, khususnya di bidang pencitraan medis [10]. Namun, *UNet* sering menghasilkan artefak pada tepi (*boundary artifacts*), yaitu ketika batas segmentasi tidak sejajar sempurna dengan kontur objek sebenarnya, terutama pada area dengan bentuk kompleks atau tepi yang kabur. Putri et al. [11] mengusulkan pendekatan *ensemble*

CNN untuk mengidentifikasi aksara kerinci ke aksara latin, yang menunjukkan keunggulan strategi multi-model dengan konfigurasi hiperparameter yang beragam. Pendekatan berbasis *CNN* juga telah menunjukkan potensi besar dalam inspeksi las. Beberapa penelitian mengusulkan strategi *CNN* yang beragam untuk mendeteksi dan mengklasifikasikan cacat pengelasan dalam berbagai kondisi. Misalnya, Kumaresan et al. [12] meneliti sistem deteksi cacat otomatis berbasis *deep learning* dengan memanfaatkan *transfer learning* pada arsitektur *CNN* seperti *VGG16* dan *ResNet50*. Meskipun dataset kecil dan tidak seimbang pada 15 kelas, model berbasis *VGG16* mampu mencapai akurasi rata-rata 90%, menunjukkan kemampuan generalisasi yang baik. Oh et al. [13] mengembangkan algoritma deteksi cacat las otomatis pada citra radiografi menggunakan *Faster R-CNN* dengan akurasi tinggi, serta membandingkan hasilnya menggunakan *ResNet* dan *Inception-ResNet V2* sebagai *backbone*. Untuk mengatasi keterbatasan tersebut, penelitian ini mengeksplorasi arsitektur *Nested UNet* karena belum adanya pendekatan segmentasi berbasis *CNN* yang dirancang khusus untuk mendeteksi titik las secara presisi pada citra real industri. Selain itu, belum banyak penelitian yang mengevaluasi pengaruh variasi blok konvolusional pada performa model *Nested UNet* untuk domain pengelasan kapal. Arsitektur ini memperkenalkan jalur *densely nested skip pathways* yang lebih efektif dalam menjembatani kesenjangan semantik antara *feature map encoder* dan *decoder*. Sehingga memungkinkan agregasi fitur yang lebih halus dan supervisi yang lebih dalam, sehingga meningkatkan kinerja segmentasi, khususnya pada kasus dengan pola visual yang rumit seperti titik las. Selain mengadopsi kerangka *UNet*, penelitian ini juga meneliti pengaruh berbagai blok konvolusional yang ditempatkan pada bagian arsitektur *bottleneck*, diantaranya blok *Dense*, *Inception*, dan *VGG*. Adapun kontribusi utama dari penelitian adalah mengusulkan model *Nested UNet* dengan integrasi berbagai blok konvolusional. Model ini terbukti melampaui performa *UNet* pada penelitian sebelumnya [14]. Melalui studi eksperimental, kami mengidentifikasi kombinasi blok konvolusional yang paling efektif untuk meningkatkan kemampuan *Nested UNet* dalam mendeteksi titik las secara akurat. Secara praktis hasil penelitian ini memiliki implikasi langsung bagi industri perkapalan, khususnya dalam mengotomatisasi proses inspeksi titik las. Mengurangi ketergantungan pada pemeriksa manusia dan meningkatkan objektivitas penilaian kualitas las dan membuka peluang integrasi sistem inspeksi berbasis visi komputer pada jalur *quality control* di galangan kapal.

LANDASAN TEORI

Convolutional Neural Network (CNN) merupakan arsitektur jaringan saraf tiruan yang secara luas digunakan dalam pengolahan citra karena kemampuannya mengekstraksi fitur spasial secara otomatis melalui operasi konvolusi. Setiap lapisan konvolusional dalam *CNN* bekerja dengan menggunakan filter berukuran kecil yang digeser pada seluruh area citra untuk mengenali pola visual tertentu seperti tepi, sudut, atau tekstur. Informasi yang diekstraksi dari setiap lapisan akan semakin kompleks seiring dengan bertambahnya kedalaman jaringan. Struktur dasar *CNN* terdiri atas lapisan konvolusi, aktivasi non-linear seperti *ReLU*, normalisasi, dan *pooling layer* yang berfungsi untuk mereduksi dimensi spasial. Pendekatan ini terbukti efektif dalam berbagai aplikasi seperti klasifikasi, deteksi, dan segmentasi citra di bidang medis, otomotif, dan industri manufaktur [2][3]. Salah satu arsitektur *CNN* yang banyak diadopsi untuk tugas segmentasi citra adalah *UNet*. Arsitektur ini pertama kali diperkenalkan oleh Ronneberger et al. (2015) dan dikembangkan untuk segmentasi citra biomedis. *UNet* memiliki struktur simetris yang terdiri atas dua jalur

utama, yaitu *encoder* yang berfungsi mengekstraksi fitur penting dari citra masukan, dan *decoder* yang berfungsi mengembalikan resolusi spasial agar model mampu menghasilkan segmentasi dengan batas yang akurat. Kelebihan utama *UNet* terletak pada adanya *skip connection* antara lapisan *encoder* dan *decoder*, yang memungkinkan transfer informasi spasial tingkat rendah ke lapisan yang lebih tinggi. Meskipun demikian, *UNet* konvensional memiliki keterbatasan dalam menangkap informasi kontekstual multi-skala karena perbedaan semantik antara fitur yang dihasilkan oleh *encoder* dan *decoder* [5][15]. Untuk mengatasi hal ini, muncul varian *Nested UNet* yang memperkenalkan jalur koneksi bersarang (*dense skip connections*) sehingga mampu menghasilkan fitur yang lebih halus dan kaya konteks.

Selain pengembangan arsitektur dasar, peningkatan performa *CNN* juga bergantung pada desain blok konvolusional yang digunakan. Beberapa jenis blok yang umum diimplementasikan antara lain *VGG Block*, *Dense Block*, dan *Inception Block*. Blok *VGG* menggunakan beberapa konvolusi berukuran kecil (3×3) secara berurutan yang diikuti dengan aktivasi *ReLU* dan normalisasi, sehingga mampu menangkap pola tekstur sederhana dengan stabilitas pelatihan yang baik [19]. Sementara itu, *Dense Block* menghubungkan setiap lapisan dengan semua lapisan berikutnya melalui mekanisme *feature reuse*, yang meningkatkan efisiensi parameter dan mengurangi risiko hilangnya gradien [21]. Di sisi lain, *Inception Block* mengombinasikan beberapa konvolusi dengan ukuran *kernel* berbeda secara paralel untuk mengekstraksi fitur multi-skala dalam satu tahap, sehingga jaringan dapat mengenali objek dengan ukuran dan bentuk yang bervariasi [20]. Dalam konteks segmentasi citra, kinerja model biasanya dievaluasi menggunakan metrik seperti *Dice Coefficient* dan *Intersection over Union (IoU)*, yang mengukur tingkat kesesuaian antara area hasil prediksi dengan *ground truth*. Nilai *dice* yang mendekati 1 menunjukkan segmentasi yang presisi. Selain itu, fungsi loss yang digunakan umumnya merupakan kombinasi antara *Binary Cross-Entropy (BCE)* dan *Dice Loss*, untuk menjaga keseimbangan antara piksel positif dan negatif pada citra biner [22][23]. Aplikasi *CNN* dalam bidang pengelasan telah dikaji oleh beberapa penelitian sebelumnya. Kumaresan et al. [12] menerapkan *transfer learning* menggunakan arsitektur *VGG16* dan *ResNet50* untuk klasifikasi cacat las, sementara Oh et al. [13] memanfaatkan *Faster R-CNN* untuk mendeteksi cacat pada citra radiografi pengelasan dengan akurasi tinggi. Meskipun hasilnya menjanjikan, sebagian besar studi tersebut berfokus pada klasifikasi atau deteksi objek tunggal, bukan segmentasi detail dari area titik las. Oleh karena itu, pendekatan berbasis *Nested UNet* dengan variasi blok konvolusional sebagaimana diusulkan dalam penelitian ini menjadi relevan untuk mengisi kesenjangan penelitian yang ada, sekaligus memberikan kontribusi nyata bagi pengembangan sistem inspeksi las otomatis dalam industri perkapalan.

METODE PENELITIAN

Penelitian ini termasuk dalam penelitian eksperimental dengan pendekatan kuantitatif. Jenis penelitian ini ditunjukkan oleh adanya proses pengujian terkontrol terhadap beberapa konfigurasi model *deep learning* berbasis *Nested UNet* yang dimodifikasi, dengan variabel bebas berupa variasi blok konvolusional (*VGG*, *Dense*, dan *Inception*) dan variabel terikat berupa nilai kinerja model seperti *Dice Coefficient* dan *Intersection over Union (IoU)*. Eksperimen dilakukan secara sistematis melalui tahapan persiapan dataset, perancangan arsitektur, pelatihan, serta evaluasi performa menggunakan metrik numerik yang terukur. Dengan demikian, penelitian ini merupakan eksperimen laboratorium

berbasis komputasi yang bertujuan menguji dan membandingkan efektivitas berbagai desain arsitektur CNN untuk segmentasi citra pengelasan secara objektif dan kuantitatif. Penelitian ini berfokus pada pengembangan model *Nested UNet*, segmentasi citra berbasis *deep learning* untuk mendeteksi titik las dari model sebelumnya menggunakan arsitektur *UNet*. Metodologi penelitian terdiri dari empat komponen utama, yaitu: persiapan *dataset*, perancangan arsitektur jaringan, metrik evaluasi, *loss functions*, serta konfigurasi pelatihan.

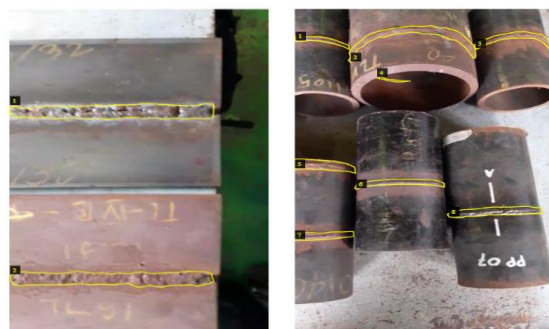
Persiapan *Dataset*

Dataset dikompilasi dari 22 video pengelasan beresolusi tinggi yang dikumpulkan dari bengkel las Politeknik Perkapalan Negeri Surabaya. Setiap video direkam dengan resolusi 1.280×720 pada kecepatan 30 *frame per second* (FPS). Dari video tersebut, citra diekstraksi setiap 15 *frame*, menghasilkan sekitar 2 citra per detik. Proses ini menghasilkan total 548 citra pengelasan. Beberapa contoh citra dari bengkel las tersebut ditampilkan pada Gambar 1.

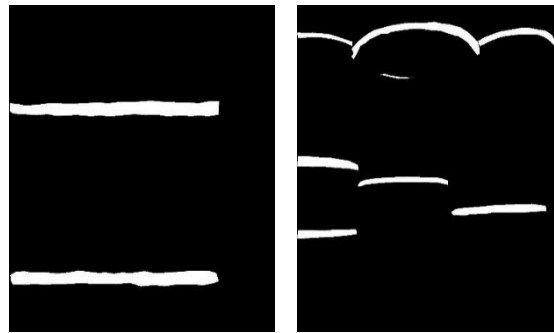


Gambar 1. Citra Pengelasan

Setiap citra dianotasi secara manual menggunakan aplikasi *VGG Image Annotator* (VIA) dengan menandai area titik las sebagai poligon. Beberapa contoh citra beranotasi beserta area titik las yang sesuai ditampilkan pada Gambar 2. Anotasi tersebut kemudian secara programatis dikonversi menjadi *binary segmentation mask* menggunakan *Python*. *Dataset* yang dihasilkan memuat 2.602 area titik las berlabel, di mana piksel berwarna putih menunjukkan titik las (kelas 1) dan piksel hitam merepresentasikan latar belakang (kelas 0). Beberapa contoh *annotation mask* dapat dilihat pada Gambar 3.



Gambar 2. Anotasi Citra dengan VIA Tools

Gambar 3. Hasil *Masking*

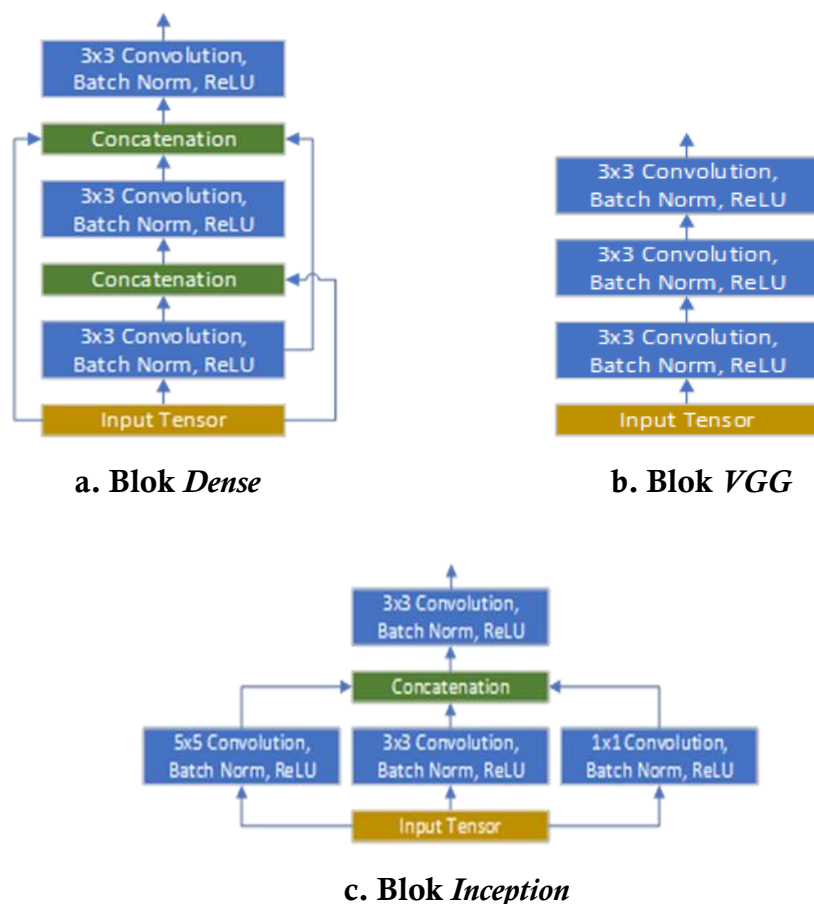
Arsitektur Jaringan

Penelitian ini mengadopsi arsitektur dasar *Nested UNet* dengan modifikasi signifikan pada struktur maupun hiperparameternya, yang disesuaikan dengan kebutuhan penelitian [15]. Arsitektur tersebut terdiri atas tiga komponen utama: jalur kontraktif, jalur ekspansif simetris, dan blok bersarang [16]. Jalur kontraktif berfungsi untuk mengekstraksi fitur diskriminatif dari citra masukan, sedangkan jalur ekspansif simetris berperan dalam menghasilkan lokalisasi yang akurat. Blok bersarang digunakan untuk memproses tensor pada jalur *skip connection*, sehingga menyempurnakan desain *UNet* konvensional [17]. Dalam penelitian ini, arsitektur direstrukturisasi menjadi empat komponen utama: jalur kontraktif, jalur ekspansif simetris, blok bersarang, dan bottleneck. Selain itu, blok konvolusional yang digunakan memiliki bentuk berbeda, yaitu lebih menyerupai persegi panjang dibandingkan persegi. Arsitektur lengkap dari model ini ditunjukkan pada Gambar 4.

Gambar 4. Arsitektur *Nested UNet* yang Dimodifikasi

Garis biru di sisi kiri, merepresentasikan jalur kontraktif yang terdiri atas blok konvolusional dengan dua konvolusi 3x3, *batch normalization*, dan aktivasi *ReLU*. *Batch normalization* diterapkan pada setiap blok untuk mempercepat sekaligus menstabilkan proses pelatihan melalui normalisasi [18]. Setelah itu dilakukan *max pooling* 2x2 dengan *stride* (2,2) untuk secara bertahap mengurangi dimensi spasial dari *feature map*, sementara jumlah kanal digandakan pada setiap blok untuk meningkatkan kedalaman *feature map*. Pada sisi kanan, garis merah menggambarkan jalur ekspansif simetris, yang mencakup *upsampling* 2x2 untuk menyelaraskan dimensi spasial dengan jalur kontraktif yang sesuai.

Proses ini dilanjutkan dengan konkatenasi *feature map* (ditunjukkan dengan garis hitam dari jalur kontraktif) dan dua konvolusi 3x3 dengan *batch normalization* serta aktivasi *ReLU*. Pada jalur ini, jumlah kanal dibagi dua pada setiap langkah, sehingga kedalaman *feature map* berkurang. Garis hitam juga mencakup blok konvolusional yang bertanggung jawab menghasilkan *feature map* berwarna oranye. Terakhir, garis ungu menunjukkan lapisan konvolusional 1x1 dengan aktivasi sigmoid yang digunakan untuk prediksi *mask*, menghasilkan satu kanal keluaran sesuai dengan jumlah kelas dalam dataset, yang dalam hal ini adalah satu. Garis hijau merepresentasikan blok konvolusional yang digunakan untuk menghasilkan *feature map* pada bagian *bottleneck*. Penelitian ini membandingkan beberapa jenis blok konvolusional pada *bottleneck*, diantaranya blok *VGG* [19], blok *Inception* [20], dan blok *Dense* [21], yang semuanya ditunjukkan pada Gambar 5.



Gambar 5. Blok Konvolusional pada *Bottleneck*

Data masukan model diubah ukurannya menjadi 256×256 piksel dengan tiga kanal yang sesuai dengan ruang warna *RGB*. Seperti arsitektur jaringan yang ditunjukkan pada Gambar 4. Struktur lapisan secara rinci beserta ukuran keluaran *feature maps* untuk setiap blok konvolusional disajikan pada Tabel 1.

Tabel 1. *Layers dan Size* pada Feature Maps

Path	Block	Layers	Feature Maps Size (HxWxCh.)
Input			256 x 256 x 3
Contractive Path	1	- 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU - 2x2 Maxpooling with stride (2,2)	64 x 256 x 32
	2	- 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU - 2x2 Maxpooling with stride (2,2)	32 x 64 x 64
	3	- 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU - 2x2 Maxpooling with stride (2,2)	16 x 32 x 256
	4	- 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU - 2x2 Maxpooling with stride (2,2)	8 x 16 x 256
Bottleneck		Refers to One Convolutional Blocks in Figure 5	8 x 16 x 512
Symmetric Expanding Path & Nested block	4	- 2x2 Upsampling - Feature maps concatenation - 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU	16 x 32 x 256
	3	- 2x2 Upsampling - Feature maps concatenation - 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU	32 x 64 x 256
	2	- 2x2 Upsampling - Feature maps concatenation - 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU	64 x 256 x 64

	1	- 2x2 Upsampling - Feature maps concatenation - 3x3 Convolution → Batch normalization → ReLU - 3x3 Convolution → Batch normalization → ReLU	256 x 256 x 32
Output		- 1x1 Convolution → Sigmoid	256 x 256 x 1

Semua lapisan konvolusional pada jalur kontraktif maupun jalur ekspansif simetris menggunakan *kernel* 3×3 dengan *stride* (1,1), serta menerapkan *zero padding* untuk mempertahankan konsistensi dimensi *feature maps*.

Metrik Evaluasi dan Loss Functions

Pada proses segmentasi, *loss functions* yang digunakan merupakan kombinasi antara *binary cross-entropy (BCE) loss* [22] dan *dice loss* [23], sedangkan metrik evaluasi yang digunakan adalah *Jaccard Index* atau *Intersection over Union (IoU)*. Perhitungan dari *BCE loss* dan *dice loss* masing-masing ditunjukkan pada Persamaan (1) dan (2).

$$(1) \quad BCE \text{ Loss} = \frac{1}{N} \sum_{i=0}^N -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

Di sini, N merepresentasikan jumlah total neuron pada lapisan keluaran, dengan perhitungan $128 \times 256 \times 1 = 32.768$ (lebar × tinggi × jumlah kelas), dengan y_i menunjukkan nilai *ground truth* dan y_i merujuk pada nilai prediksi.

$$Dice \text{ Loss} = 1 - \frac{2 |Y_t \cap Y_p|}{|Y_t| + |Y_p|} \quad (2)$$

Dalam konteks ini, Y_t merepresentasikan nilai *ground truth*, sedangkan Y_p merujuk pada nilai prediksi. Nilai *loss* keseluruhan kemudian dihitung sebagaimana ditunjukkan pada Persamaan (3).

$$BCEDice \text{ loss} = \alpha BCE \text{ Loss} + \beta Dice \text{ Loss} \quad (3)$$

BCEDice loss merupakan kombinasi antara *BCE loss* dan *dice loss*, di mana kedua koefisien α dan β diberi nilai 1 sehingga masing-masing komponen memiliki pengaruh yang sama terhadap model. Selain itu, penelitian ini juga mengintegrasikan regularisasi L2 sebagaimana diusulkan oleh Tessier et al. [24], dengan *weight decay* sebesar 0,0001 untuk mengurangi kompleksitas model dan meningkatkan kemampuan generalisasi. Dengan demikian, *loss* keseluruhan dihitung menggunakan Persamaan (4).

$$Total \text{ loss} = BCEDice \text{ loss} + w_d \sum_{i=0}^N (w_i)^2 \quad (4)$$

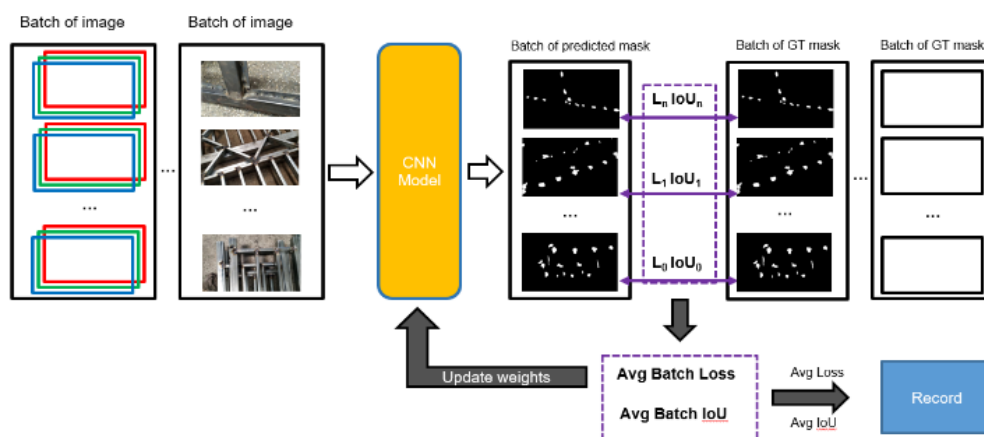
Pada persamaan ini, w_d merepresentasikan *weight decay*, w_i menunjukkan bobot individual model, dan N adalah jumlah total bobot dalam model. Fungsi metrik yang digunakan dalam penelitian ini ditunjukkan pada Persamaan (5).

$$J(Y_t, Y_p) = \frac{|Y_t \cap Y_p|}{|Y_t \cup Y_p|} = \frac{|Y_t \cap Y_p|}{|Y_t| + |Y_p| - |Y_t \cap Y_p|} \quad (5)$$

Di sini, J merepresentasikan *Jaccard Index* atau yang juga dikenal sebagai *Intersection over Union* (IoU), dengan Y_t menunjukkan nilai *ground truth* dan Y_p menunjukkan nilai prediksi.

Konfigurasi Training

Pelatihan dilakukan pada komputer dengan spesifikasi *GPU RTX 3060Ti*, *RAM 16 GB*, dan prosesor *Intel i7*. Untuk meningkatkan kemampuan generalisasi model, teknik data augmentation seperti rotasi, pengubahan ukuran, normalisasi, penyesuaian saturasi, kontras, dan kecerahan diterapkan selama tahap *preprocessing* citra [25]. Pelatihan menggunakan *batch size* sebesar 2, dengan setiap citra diubah ukurannya menjadi 256×256 piksel dan 3 kanal warna (*RGB*). *Dataset* yang terdiri atas 548 citra dibagi menjadi 80 persen untuk *training* dan 20 persen untuk *validation*, sehingga diperoleh 438 citra *training* dan 110 citra *validation*. Sebagaimana diterapkan pada penelitian sebelumnya, model dilatih menggunakan algoritma *stochastic gradient descent* (*SGD*) [26] dengan *learning rate* awal sebesar 0,1 dan momentum 0,9. Selama *training*, *learning rate* akan dibagi dua apabila *validation loss* tidak menunjukkan perbaikan selama 15 epoch berturut-turut, dengan ambang minimum ditetapkan pada 0,000025. Meskipun *training* diizinkan berlangsung hingga 1.000 *epoch*, proses akan berhenti secara otomatis jika nilai *validation IoU* tidak meningkat selama 100 *epoch* berturut-turut. Mekanisme *training* dengan pendekatan *batch processing* ditunjukkan pada Gambar 6.



Gambar 6. Mekanisme Pelatihan dengan Teknik Batch Processing

Pada tahap *training*, satu *batch* yang terdiri dari 2 citra dilewatkan melalui model untuk menghasilkan *predicted mask* pada setiap citra. Hasil prediksi ini kemudian dibandingkan dengan *ground truth mask* yang sesuai untuk menghitung nilai *loss* dan *IoU*. Rata-rata *loss* dan *IoU* untuk setiap *batch* juga dihitung, dengan rata-rata *batch loss*

digunakan untuk memperbarui bobot model selama *backpropagation*. Proses ini berlangsung hingga seluruh *batch* selesai diproses. Setelah semua *batch* dalam satu *epoch* selesai, rata-rata *loss* dan *IoU* untuk keseluruhan *epoch* dihitung dan dicatat untuk memantau kinerja model. Siklus pelatihan berjalan hingga 1.000 *epoch*, tetapi akan dihentikan lebih awal apabila tidak ada peningkatan nilai *IoU* validasi selama 100 *epoch* berturut-turut. Persamaan (6) dan (7) menunjukkan cara menghitung serta membedakan antara rata-rata *batch loss* dengan rata-rata *loss*. Hal yang sama juga diterapkan untuk rata-rata *batch IoU* dan rata-rata *IoU*.

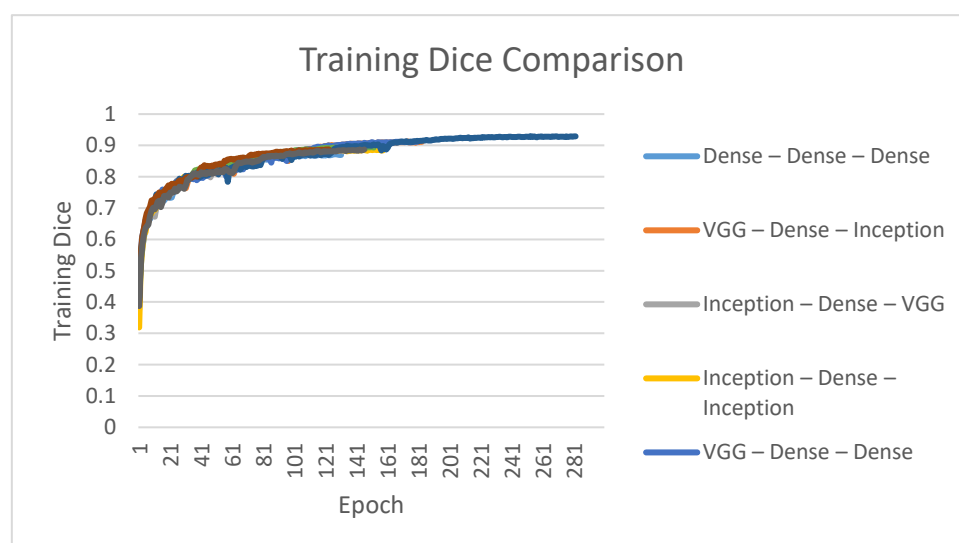
$$Avg\ Batch\ Loss = \frac{1}{N} \sum_{i=0}^N BCEDice\ Loss_i \quad (6)$$

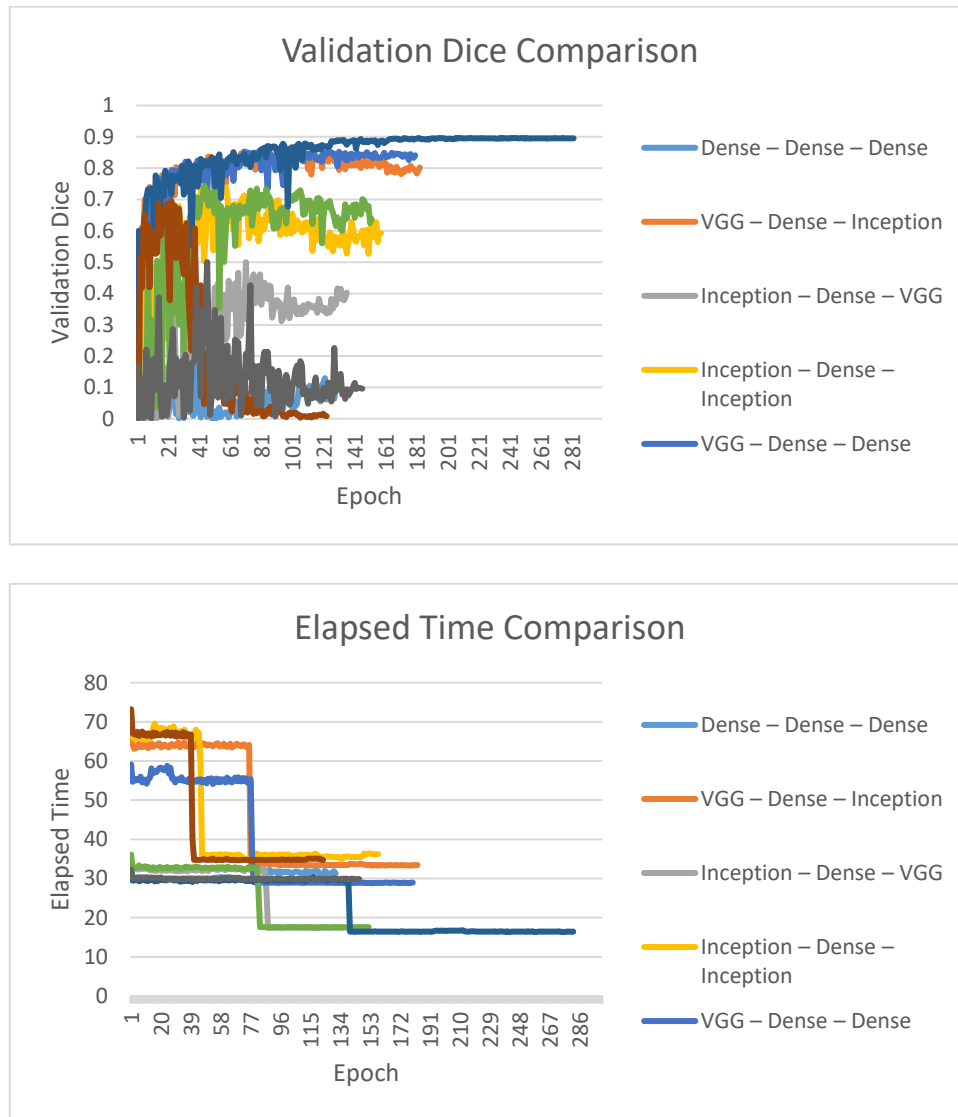
$$Avg\ Loss = \frac{1}{M} \sum_{i=0}^M Avg\ Batch\ Loss_i \quad (7)$$

Dalam konteks ini, *Avg* mengacu pada nilai rata-rata, *N* merepresentasikan jumlah citra per *batch* yaitu 2, dan *M* menunjukkan jumlah total *batch*. Dengan *dataset* yang terdiri atas 548 citra (438 untuk *training* dan 110 untuk *validation*), terdapat 219 *batch training* dan 55 *batch validation*. Proses validasi dilakukan dengan alur yang serupa dengan pelatihan, dengan perbedaan utama tidak adanya *backpropagation*, sehingga bobot model tidak diperbarui. Sebagaimana dijelaskan sebelumnya, rata-rata *loss* dan *IoU* per *epoch* digunakan sebagai kriteria untuk mengurangi *learning rate* atau menghentikan proses *training*.

HASIL DAN PEMBAHASAN

Sebagaimana dijelaskan pada bagian sebelumnya, kinerja model dievaluasi menggunakan *IoU* dan *loss* sebagai metrik utama. Selama proses *training*, kinerja pada *dataset training* maupun *validation* dipantau dan dicatat secara berkelanjutan. Dengan demikian, akan dihasilkan beberapa *log* yang mencatat *training loss*, *validation loss*, *training IoU*, dan *validation IoU* untuk masing-masing dari tiga jenis blok konvolusional yang digunakan pada *dense bottleneck*. Perbandingan variasi *encoder* dan *decoder* terhadap setiap parameter ditunjukkan pada Gambar 7.





Gambar 7. Perbandingan Kinerja Model

Model dapat dikategorikan tidak mengalami *overfitting* maupun *underfitting*, karena nilai *validation loss* dan *validation IoU* menunjukkan kesesuaian yang erat dengan *training loss* dan *training IoU*. Seperti telah dijelaskan sebelumnya, proses pelatihan dihentikan apabila tidak terdapat peningkatan pada *validation IoU*, yang menjelaskan mengapa setiap model dapat menyelesaikan pelatihan pada jumlah *epoch* yang berbeda. Rincian lebih lanjut mengenai perbandingan tersebut disajikan pada Tabel 2.

Tabel 2. Nested UNet Performance Comparison

N o	Convolutional Block	Highest Training Dice	Highest Validation Dice	Highest Elapsed Time	Total Epoch
1	Dense - Dense - Dense	0,87248	0,29698	32,50059	131
2	VGG - Dense - Inception	0,91269	0,86022	69,87927	183
3	Inception - Dense - VGG	0,88615	0,49991	34,54659	136

4	Inception - Dense - Inception	0,88769	0,73975	72,14195	158
5	VGG - Dense - Dense	0,91199	0,85652	59,12399	180
6	Dense - Dense - VGG	0,89545	0,74869	36,09930	152
7	VGG - Dense - VGG	0,92970	0,89695	31,29886	282
8	Dense - Dense - Inception	0,88743	0,69525	73,22974	123
9	Inception - Dense - Dense	0,88696	0,50042	32,27743	146

Blok *VGG - Dense - VGG* menunjukkan kinerja terbaik pada data *training*, dengan capaian *training dice score* sebesar 0,92970 dan *validation dice score* sebesar 0,89695. Dari sisi kecepatan konvergensi, blok *VGG - Dense - VGG* menyelesaikan pelatihan pada *epoch* ke-282. Dalam *deep learning*, konvergensi merujuk pada kemampuan model untuk belajar dan merespons pola pelatihan secara akurat dalam batas kesalahan yang dapat diterima. Dari parameter yang dievaluasi, nilai *dice* tertinggi pada data validasi digunakan sebagai kriteria utama dalam menentukan kinerja terbaik model. Karena bobot model diperbarui menggunakan data pelatihan, performa tinggi pada pelatihan saja tidak bermakna apabila hasil pada data validasi rendah atau rata-rata. Demikian pula, kecepatan konvergensi bukan merupakan faktor penentu apabila kinerja model tidak memadai, karena metrik ini tidak relevan untuk penerapan. Sebaliknya, pencapaian *validation dice* tertinggi atau *validation loss* terendah menunjukkan kemampuan generalisasi yang lebih baik saat implementasi. Berdasarkan pertimbangan tersebut, model *Nested UNet* dengan blok konvolusional *VGG - Dense - VGG* merupakan model dengan kinerja keseluruhan terbaik, karena berhasil mencapai nilai *dice* tertinggi pada pelatihan maupun validasi. Hasil tersebut efektif dalam mengekstraksi pola spasial pada berbagai tingkat abstraksi, teori *transfer learning* dan *convolutional representation* dari Simonyan & Zisserman (2014) dapat mendasari pemilihan blok VGG dalam arsitektur. Sementara konsep *dense connectivity* dari Huang et al. (2017) menjelaskan mekanisme *feature reuse* pada *DenseNet* yang meningkatkan efisiensi pembelajaran.

KESIMPULAN

Dari penelitian ini dapat disimpulkan bahwa model *Nested UNet* dengan blok konvolusional *VGG - Dense - VGG* menunjukkan kinerja terbaik dibandingkan dengan blok lainnya. Blok ini diidentifikasi sebagai yang terbaik berdasarkan capaian *training dice score* tertinggi sebesar 0,92970 dan *validation dice score* sebesar 0,89695, yang digunakan sebagai metrik utama karena mencerminkan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Sementara itu, metrik *training* dan waktu konvergensi hanya memberikan wawasan pendukung dan dianggap sekunder karena tidak secara langsung merefleksikan kinerja model dalam penerapan nyata. Penelitian selanjutnya dapat difokuskan pada pengembangan model untuk menilai kualitas dan cacat pengelasan, serta merancang algoritme *post-processing* untuk menghitung jumlah titik las. Eksplorasi lebih lanjut juga dapat mencakup modifikasi atau pendalaman blok konvolusional, optimisasi hiperparameter secara otomatis, serta reduksi ukuran dan kompleksitas model untuk mempercepat implementasi pada perangkat tertentu.

PENGAKUAN/ACKNOWLEDGEMENTS

Penelitian ini dilaksanakan dengan dana DIPA Politeknik Perkapalan Negeri Surabaya melalui nomor 295.189/PL19.11/AL.04/2024. Penulis juga menyampaikan

terima kasih kepada staf laboratorium di bengkel las Politeknik Perkapalan Negeri Surabaya atas bantuan mereka dalam proses pengumpulan dataset.

DAFTAR REFERENSI

1. T. W. Purnomo, F. Danitasari, and D. Handoko, "Weld Defect Detection and Classification based on Deep Learning Method: A Review", *Journal of Computer Science and Information*, Vol. 16, No. 1, pp. 77-87, 2023.
2. A. I. Gavrilov and M. T. Do, "Classification of Weld Defects based on Convolution Neural Network", *Instrument Engineering*, No. 2, pp. 23-36, 2021.
3. S. Surono, M. Rivaldi, D. A. Dewi, and N. Irsalinda, "New Approach to Image Segmentation: U-Net Convolutional Network for Multiresolution CT Image Lung Segmentation", *Emerging Science Journal*, Vol. 7, No. 2, pp. 498-506, 2023.
4. P. Asri, Y. Widiarti, E. P. Purwanti, E. Wismawati, and M. F. T. Arifin, "Obstacles Detection in Underwater Environment Using ROV Based on Convolutional Neural Network", *Indonesian Journal of Computing and Cybernetics Systems*, Vol. 19, No. 1, pp. 105-116, 2024.
5. E. Akanksha and P. R. K. Rao, "A Feature Extraction Approach for Multi-Object Detection Using HoG and LTP", *Intelligent Networks and Systems Society*, Vol. 14, No. 5, pp. 259-268, 2021.
6. A. N. Azhari and Wahyono, "Automatic Detection of Helmets on Motorcyclists Using Faster -RCNN", *Indonesian Journal of Computing and Cybernetics Systems*, Vol. 16, No. 4, pp. 357-366, 2022.
7. M. Ansari, A. Choudhary, and S. Bhosale, "Semantic Segmentation using Convolutional Neural Networks", *International Journal of Engineering Research in Computer Science and Engineering*, Vol. 10, No. 6, pp. 31-34, 2023.
8. L. Yang, H. Wang, W. Meng, and H. Pan, "CvT-UNet: A weld pool segmentation method integrating a CNN and a transformer", *Heliyon*, Vol. 10, 2024.
9. M. Vadduri and P. Kuppusamy, "DAUCD: Deep Attention U-Net for Cataract Detection Leveraging CNN Frameworks", *Intelligent Networks and Systems Society*, Vol. 18, No. 1, pp. 184-197, 2025.
10. H. K. Abass, M. H. Al-hashimi, A. S. Al-Zubaidi, and M. Al-Mukhtar, "GastroFPN Advanced Deep Segmentation Model for Gastrointestinal Disease with Enhanced Feature Pyramid Network Decoder", *Intelligent Networks and Systems Society*, Vol. 17, No. 4, pp. 438-450, 2024.
11. T. A. Putri, T. Suratno, and U. Khaira, "Identification of Incung Characters (Kerinci) to Latin Characters Using Convolutional Neural Network", *Indonesian Journal of Computing and Cybernetics Systems*, Vol. 16, No. 2, pp. 205-216, 2022.
12. S. Kumaresan, K. S. Jai Aultrin, S. S. Kumar, and M. D. Anand, "Deep learning-based weld defect classification using VGG16 transfer learning adaptive fine-tuning", *International Journal on Interactive Design and Manufacturing*, Vol. 17, pp. 2999-3010, 2023.
13. S. Oh, M. Jung, C. Lim, and S. Shin, "Automatic Detection of Welding Defects Using Faster R-CNN", *Applied Sciences*, Vol. 10, No. 23, pp. 8629, 2020.
14. O. Natan, D. U. K. Putri, and A. Dharmawan, "Deep Learning-based Weld Spot Segmentation using Modified UNet with Various Convolutional Blocks", In: *ICIC Express Letter B: Applications*, Vol. 12, No. 12, pp. 1169-1176, 2021.

15. Y. Yang, S. Dasmahapatra, and S. Mahmoodi, "ADS_UNet: A Nested UNet for Histopathology Image Segmentation", *Expert Systems with Applications*, Vol. 226, 2023.
16. C. Mathews and A. Mohamed, "Nested U-Net with Enhanced Attention Gate and Compound Loss for Semantic Segmentation of Brain Tumor from Multimodal MRI", *Intelligent Networks and Systems Society*, Vol. 15, No. 4, pp. 248-256, 2024.
17. M. Anand and M. A. Sundaram, "Channel and Spatial Attention Aware UNet Architecture for Segmentation of Blood Vessels, Exudates and Microaneurysms in Diabetic Retinopathy", *Intelligent Networks and Systems Society*, Vol. 17, No. 2, pp. 1-16, 2024.
18. B. K. Triwijoyo, A. Adil, and A. Anggrawan, "Convolutional Neural Network with Batch Normalization for Classification of Emotional Expressions Based on Facial Images", *Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 21, No. 1, pp. 197-204, 2021.
19. Herman, A. P. Putri, M. N. M. M. Noor, H. Darwis, L. N. Hayati, Irawati, and I. As'ad, "A Comparative Study on Efficacy of CNN VGG-16, DenseNet121, ResNet50V2, And EfficientNetB0 in Toraja Carving Classification", *Indonesian Journal of Data and Science*, Vol. 6 No. 1, pp. 123-133, 2025.
20. T. Barakbayeva and M. Fatih Demirci, "Fully automatic CNN design with inception blocks", In: *Thirteenth International Conference on Digital Image Processing*, Singapore, 2021.
21. W. W. Kusuma, R. R. Isnanto, and A. Fauzi, "Comparison Analysis of CNN Model VGG16 and DenseNet121 using TensorFlow Framework to Detect Animal Images", *Jurnal Teknik Komputer*, Vol. 1, No. 4, pp. 141-147, 2023.
22. A. U. Ruby, P. Theerthagiri, I. J. Jacob, and Y. Vamsidhar, "Binary cross entropy with deep learning technique for Image classification", *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 4, pp. 5393-5397, 2020.
23. Q. Ming and X. Xiao, "Towards Accurate Medical Image Segmentation With Gradient-Optimized Dice Loss", *IEEE Signal Processing Letters*, Vol. 31, pp. 191-195, 2024.
24. H. Tessier, V. Gripon, M. Léonardon, M. Arzel, T. Hannagan, and D. Bertrand, "Rethinking Weight Decay for Efficient Neural Network Pruning", *Journal of Imaging*, Vol. 8, pp. 64, 2022.
25. A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations", *Information*, Vol. 11, No. 2, pp. 125, 2020.
26. S. Prongnuch and S. Sitjongsataporn, "Stability and Steady-State Performance of Hammerstein Spline Adaptive Filter Based on Stochastic Gradient Algorithm", *Intelligent Networks and Systems Society*, Vol. 13, No. 3, pp. 112-123, 2020.