Iron welding spot segmentation using Nested UNet (UNet++) enhanced with diverse convolutional modules

Thomas Brian [a,1,*], Anggarjuna Puncak Pujiputra [b,2], Oskar Natan [b,3], Yohanes Yohanie Fridelin Panduman [c,4]

[a] Department of Marine Electrical Engineering, Politeknik Perkapalan Negeri Surabaya, Surabaya 60111, Indonesia
[b] Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia
[c] Graduated School of Information Science and Technology, University of Osaka, Osaka, Japan
[1] thomasbrian@ppns.ac.id; [2] anggarjunapuncak@ppns.ac.id; [3] oskarnatan@ugm.ac.id; [4] yohanes@ist.osaka-u.ac.jp
* corresponding author

ARTICLE INFO

Article history
Received October 9, 2025
Revised November 22, 2025
Accepted December 18, 2025
Available Online May 31, 2026

Keywords
CNN
Image Segmentation
Nested Unet
Welding Spot

ABSTRACT

Welding inspection plays an essential role in manufacturing industries to ensure the integrity and quality of weld joints. However, the prevalent manual inspection procedures are inherently subjective, prone to bias, and result in inconsistent quality assessments. Therefore, there is a strong need for an automated, intelligent system capable of objectively detecting welding spots. To address this, we propose an advanced segmentation model based on deep learning and computer vision techniques, specifically utilizing a Nested UNet (UNet++) architecture enhanced by extensive architectural modifications and comprehensive hyperparameter tuning. To further optimize segmentation performance, we systematically compare various convolutional blocks integrated into the bottleneck of the network architecture. Our experimental evaluation demonstrates that employing a VGG convolutional block at the bottleneck of Nested UNet achieves the highest performance, reaching an Intersection over Union (IoU) score of 76.18% and a validation loss of 0.1713 on our collected dataset. This research advances automated welding inspection by demonstrating that an enhanced Nested UNet with an optimized VGG bottleneck significantly improves weld spot segmentation accuracy. The proposed framework provides a robust benchmark for future studies on intelligent industrial visual inspection and deep learning-based quality assurance.

1. Introduction

Welding is a fundamental joining process widely used in manufacturing industries, especially in sectors such as automotive, where a single vehicle may contain thousands of weld points. Evaluating the quality of welded joints in metals or other materials is crucial to verify that they meet design criteria and functional standards, while also ensuring structural safety and operational reliability [1]. The manufacturing sector relies significantly on welding, which serves as a fundamental technique for effectively joining diverse metal components. The process applies heat and pressure to fuse materials, resulting in strong and durable joints [2]. Despite its importance, weld inspection is still commonly carried out through manual visual examination, which is highly dependent on human perception. Aside from requiring considerable time and manual effort, this approach is inherently subjective and prone to inconsistency, which limits its applicability in large scale industrial environments. To overcome these challenges, computer vision and deep learning techniques have emerged as promising alternatives. They are capable of analyzing images to reveal underlying structures and visual features [3]. In particular,

Convolutional Neural Networks (CNN) model built on the ResNet50 architecture is proposed to classify four weld defect types in radiographic images (pore, crack, lack of penetration, and no defect), with stratified cross-validation using regularization and data augmentation applied to minimize overfitting and improve generalization [4].

Deep learning constitutes a vital subset of machine learning that has gained significant attention and growing importance in recent years [5]. It allows computational models composed of multiple processing layers to learn data representations at different levels of abstraction [6]. Alternatively, deep learning has demonstrated substantial potential in addressing visual challenges, particularly in classification, detection, and forecasting tasks within real-world applications [7]. One of the most widely adopted architectures for image segmentation is the UNet model, a deep learning approach that has proven highly effective in addressing a wide range of challenges across artificial intelligence applications [8]. UNet is a widely used architecture for segmentation that requires a large amount of training data [9]. However, conventional UNet has limitations in handling complex features and multi-scale contextual information, especially in cases where the object boundaries are unclear or when the segmentation target varies in size and texture. The single skip connection between corresponding encoder and decoder layers may not be sufficient to capture the full range of semantic details required for precise segmentation. Conventional UNet and even the improved UNet++ architecture remain insufficient for weld-spot segmentation because weld spots are small, high-contrast targets embedded in highly textured metal surfaces, making them difficult to distinguish using standard encoder–decoder representations. In industrial welding imagery, additional challenges such as varying illumination, reflections, sparks, camera angle changes, and heterogeneous welding environments further degrade segmentation reliability. Despite the known benefits of deeper skip connections in UNet++, no prior work has examined how different convolutional bottleneck blocks affect the model's ability to capture the fine-grained local structure characteristic of weld spots. To address these shortcomings, our research explores the Nested UNet architecture, also referred to as UNet++. UNet++ is a neural network architecture in which the encoder and decoder sub-networks are linked by a series of nested, dense skip connections designed to minimize the semantic gap between their feature maps [10]. This architecture introduces densely nested skip pathways that bridge the semantic gap between encoder and decoder feature maps more effectively. These nested connections enable more refined feature aggregation and deeper supervision, resulting in improved segmentation performance, particularly in scenarios with intricate visual patterns such as weld spots.

In addition to adopting the UNet++ framework, we further investigate the impact of different convolutional blocks placed in the bottleneck of the architecture. By experimenting with various convolutional structures such as Dense, Inception, Residual, Squeeze and VGG blocks [11]. In more detail, the novelties presented in this study can be summarized as follows.

- We propose a Nested UNet (UNet++) model with diverse convolutional blocks. Our model outperforms the UNet model in previous studies [11].
- This study is novel in its systematic comparison of five bottleneck convolutional modules inside a redesigned UNet++ and its finding that the VGG bottleneck yields the best weld-spot segmentation performance.

The structure of this paper is as follows: Section 2 reviews existing literature on deep learning. Section 3 details the dataset, model architecture, and training configuration. Section 4 presents the experimental results and performance analysis. Finally, the conclusion and potential directions for future work are discussed in Sections 5 and 6, respectively.

2. Related Works

In recent years, object detection and tracking have become key research areas in computer vision. Deep learning models, especially those with bypass connections, have brought significant breakthroughs in these tasks, enabling real-time system performance. Neural networks enable algorithms to be trained on large datasets and learn from the information contained within them [12]. These algorithms work by identifying objects and predicting their positions using bounding boxes. Multi object detection and

tracking is an essential area in image processing and computer vision that has been widely studied. It focuses on predicting complete trajectories of multiple objects simultaneously within a video sequence. This study applies a frame cancellation method to minimize the computational time in deep learning and combines the DeepSORT algorithm with multiple YOLO detector versions [13]. Deep learning methods have been introduced as a solution to address the limitations faced by conventional object detection techniques [14]. Abdulabass et al. [15] proposed a real-time traffic sign detection system using the YOLOv3 algorithm, which processes entire images with a single neural network, predicts bounding boxes and class probabilities efficiently.

Semantic segmentation is the process of assigning a class label to each pixel in an image, grouping pixels with similar characteristics into the same category [16], [17]. Unlike image classification, which labels the entire image, semantic segmentation provides detailed pixel-level understanding, making it essential for tasks like medical imaging, autonomous driving, and land use mapping. Yang et al. [18] introduced a novel semantic segmentation model called CvT-UNet, which combines the strengths of CNNs and Transformers. The model integrates the Transformer's ability to capture global contextual information with the CNN's advantage in extracting spatial features. Vadduri et al. [19] introduced the Deep Attention U-Net for Cataract Diagnosis (DAUCD), a model that utilizes cutting-edge deep learning approaches to improve both cataract segmentation and classification in retinal images. UNet is a dedicated convolutional neural network architecture designed for segmenting biomedical images, particularly within the domain of medical imaging [20]. However, it often produces boundary artifacts, where the segmented edges do not perfectly align with the actual object boundaries in the image, leading to errors especially in regions with complex shapes or indistinct edges. This multi-model strategy incorporates several CNN architectures (Xception, DenseNet-201, and EfficientNet-B3). Convolutional Neural Networks (CNN) have demonstrated strong potential in a wide range of detection and classification tasks, including weld inspection. CNN are effective methods for classifying, identifying, and recognizing patterns in images, with the capability to capture visual details more accurately through an architecture that mimics the way the human brain processes visual information [21], [22], [23]. Several studies have proposed diverse CNN-based strategies for detecting and classifying welding defects with varying types and conditions. For instance, Kumaresan et al. [24] have explored automated defect detection systems using deep learning. One such approach utilizes transfer learning with CNN architectures like VGG16 and ResNet50, trained on a limited X-ray image dataset enhanced through real-time data augmentation. Despite the dataset's small and imbalanced nature across 15 classes, the VGG16-based model achieved strong performance with an average accuracy of 90%, indicating good generalization capability. Yadav et al. [25] presents a CNN-SVM hybrid model for welding defect classification, achieving outstanding results with precision, recall, and F1-scores consistently above 96% across defect types such as Ideal Weld, Cracks, Porosity, Undercut, and Slag Inclusion. The model reached an overall accuracy of 96.93%, demonstrating strong generalization, reliability, and significant potential for advancing automated welding fault detection to enhance industrial inspection protocols.

3. Method

This research focuses on developing a deep learning-based image segmentation model for detecting welding spots, using an enhanced version of the Nested UNet (UNet++) architecture. The methodology consists of four main components: dataset construction, network architecture design, evaluation metrics and loss functions, and training configuration.

3.1. Dataset Preparation

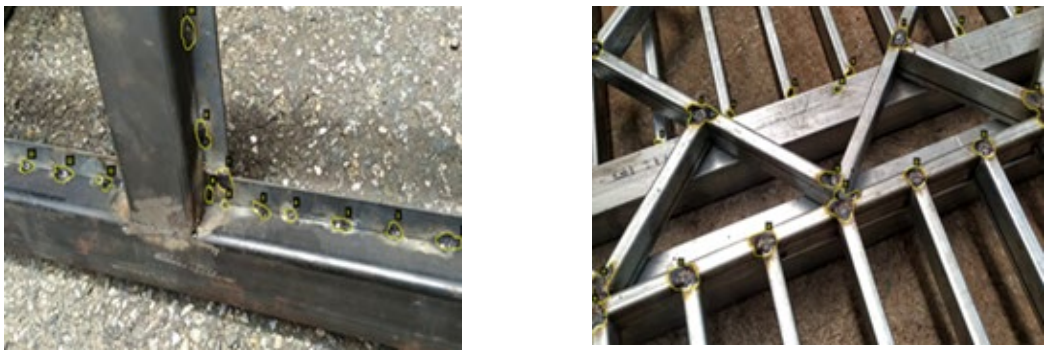
The dataset used in this research was adapted from a previous study conducted by Natan et al. [11], which focused on weld spot segmentation using a modified UNet architecture. The dataset was compiled from 34 high-definition welding videos collected from three different metal workshops located in Tulungagung, East Java, Indonesia. Each video was recorded in 1,920×1,080 resolution at 30 frames per second (FPS). From these videos, images were extracted every 15 frames, yielding approximately 2 images

per second. This process resulted in a total of 893 welding images. Several images from these welding factories can be seen in Fig. 1.

a. 1st Welding Factory b. 2nd Welding Factory

Fig. 1. Example Images of Iron Welding.

Each image was manually annotated using the VGG Image Annotator (VIA) tool, marking individual weld spot regions as polygons. Several annotated images with its corresponding welding spots region can be seen in Fig. 2. These annotations were then programmatically converted into binary segmentation masks using Python. The resulting dataset contains 12,712 labeled weld spot regions, where white pixels indicate weld spots (class 1) and black pixels denote the background (class 0). Several annotation masks can be seen in Fig. 3.



a. 1st Welding Factory b. 2nd Welding Factory

Fig. 2. Annotated Images using VIA Tools.

a. 1st Welding Factory b. 2nd Welding Factory

Fig. 3. Generated Mask.

To clarify the dataset construction process, we explain that frames were extracted at 15-frame intervals to avoid temporal redundancy, as consecutive frames in welding videos often contain minimal

variation and could otherwise introduce oversampling bias. The dataset naturally includes substantial variation in lighting conditions, camera angles, background clutter, and workshop environments across the three welding locations, contributing to stronger generalization. To prevent data leakage arising from near-duplicate frames originating from the same video sequence, we adopted a strict video-wise splitting strategy, ensuring that all frames from a single video are assigned exclusively to either the training or validation set.

3.2. Network Architecture

This research adopts the foundational architecture of Nested UNet with significant modifications to both its structure and hyperparameters, tailored to the specific needs of the research [26], [27]. The architecture consists of three main components: a contractive path, a symmetric expansive path, and nested blocks [28]. The contractive path is designed to extract discriminative features from the input image, while the symmetric expansive path facilitates accurate localization. The nested blocks serve to process tensors between skip connections, enhancing the conventional UNet design [29], [30]. In this research, the architecture is restructured into four main components: the contractive path, symmetric expanding path, nested block, and the bottleneck. Additionally, the convolutional blocks differ in shape, appearing more rectangular than square in design. The complete architecture of the model is illustrated in Fig. 4.

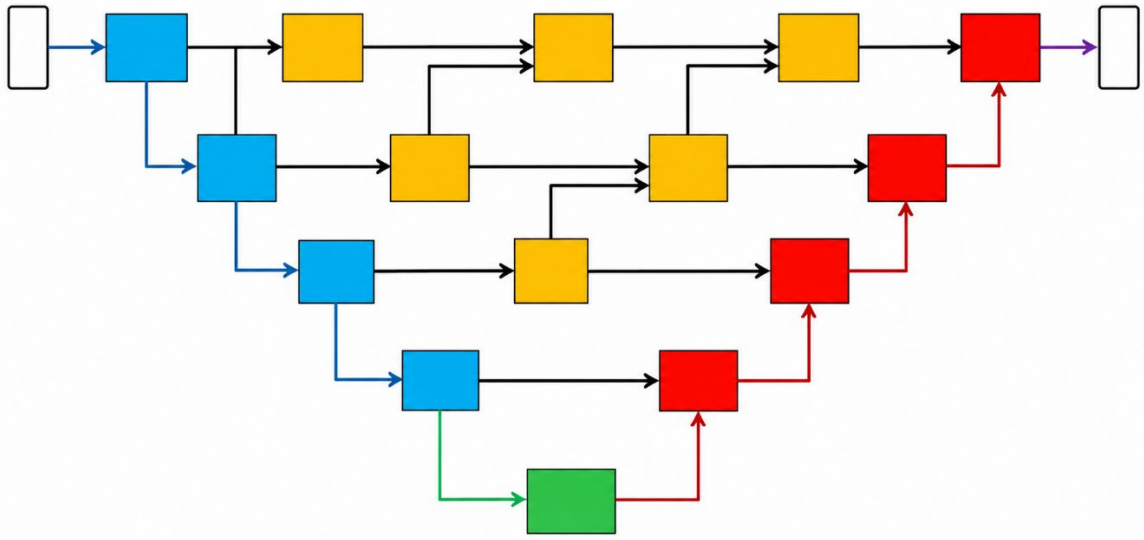
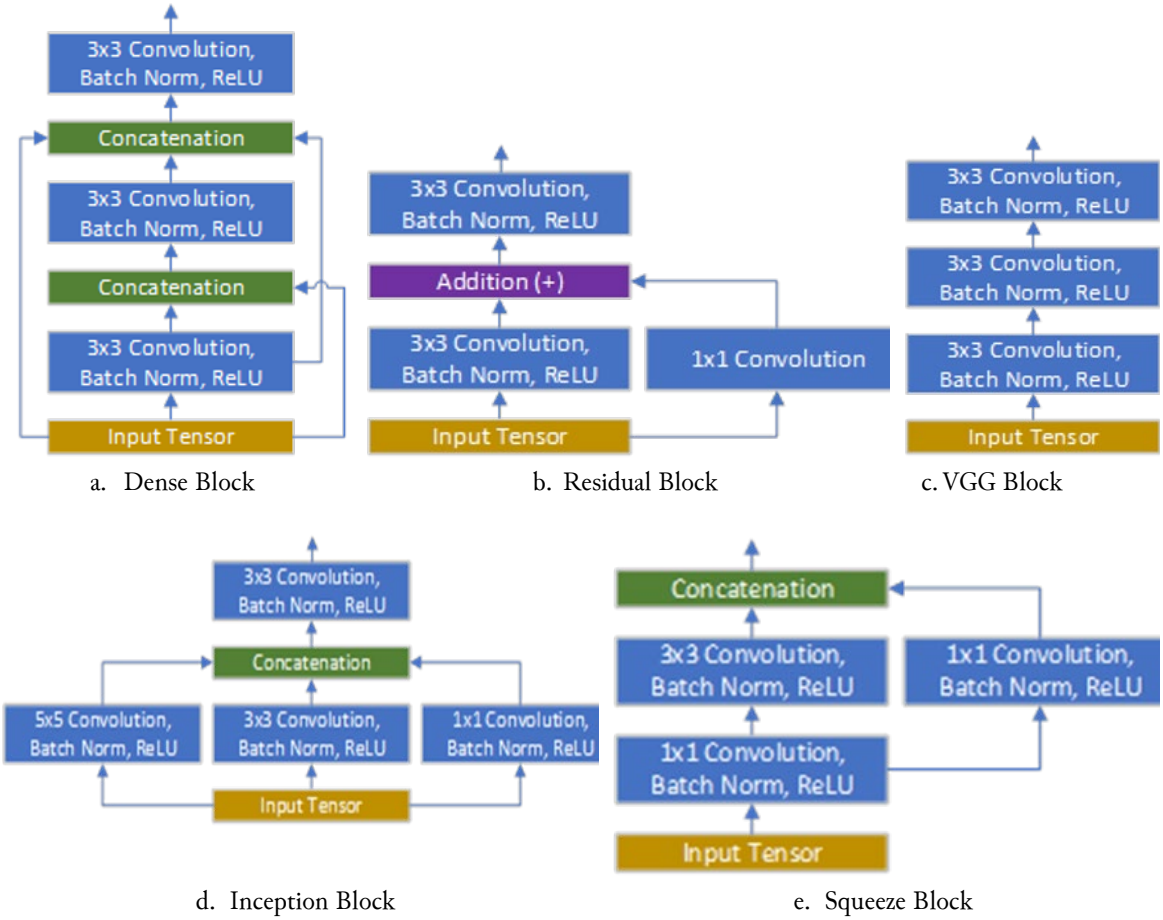


Fig. 4. Modified Nested UNet Architecture.

The blue line on the left side, representing the contractive path, consists of convolutional blocks with two 3x3 convolutions, batch normalization, and ReLU activation. Batch normalization is applied to each block to accelerate and stabilize the training process through normalization [31], [32]. This is followed by 2x2 max pooling with a stride of (2,2) to progressively reduce the spatial dimensions of the feature maps, while the number of channels is doubled in each block to increase the depth of the feature maps. On the right side, the red line represents the symmetric expanding path, which includes 2x2 upsampling to align with the corresponding spatial dimensions in the contractive path. It is followed by feature map concatenation illustrated with black line from the contractive path and two 3x3 convolutions with batch normalization and ReLU activation. In this path, the number of channels is halved at each step, reducing the depth of the feature maps. The black line also includes convolutional blocks responsible for generating the orange-colored feature maps. Finally, the purple line denotes a 1x1 convolutional layer with sigmoid activation used for mask prediction, producing a single output channel corresponding to the number of classes in the dataset, which in this case is one.

The green line represents a convolutional block used to generate feature maps in the bottleneck section. This study compares several types of convolutional blocks for the bottleneck, including the VGG

block [33], inception block [34], squeeze and excitation block [35], residual block [36], and dense block [37], all illustrated in Fig. 5. The model input is resized to 128x256 pixels with three channels corresponding to the RGB color space. As outlined in the architecture shown in Fig. 4, the detailed layer structure and the output sizes of the feature maps for each convolutional block are listed in Table 1. All convolutional layers in both the contractive and symmetric expanding paths use 3x3 kernels with a stride of (1,1), and zero padding is applied to maintain consistent feature map dimensions. To ensure a fair and meaningful comparison across all bottleneck configurations, we explicitly designed each convolutional block to have approximately similar parameter counts, preventing performance differences from simply reflecting disparities in model capacity. In cases where minor parameter variations were unavoidable due to inherent structural differences between block types (e.g., Dense or Inception modules), these differences were kept as small as possible and are acknowledged in the manuscript. This controlled setup ensures that performance improvements can be attributed primarily to architectural characteristics rather than parameter size.



a. Dense Block b. Residual Block c. VGG Block

d. Inception Block e. Squeeze Block

Fig. 5. Convolutional Blocks for Bottleneck.

Weld spots require strong preservation of fine local features, meaning that any loss of detail at the bottleneck will propagate into the decoder and negatively affect the quality of the upsampled reconstruction. We therefore added an explanation that multi-scale or residual-based blocks, while powerful for general segmentation tasks, may unintentionally over-smooth or dilute these fine-grained boundaries, making the bottleneck design especially critical for accurate weld-spot segmentation. We added a statement confirming that all experimental components including data augmentation, learning rate schedule, optimizer settings, batch size, random seeds, and epoch limits were kept identical across all trials so that the only varying factor was the bottleneck module.

Table 1. Layers and Size of Feature Maps.

Path	Block	Layers	Feature Maps Size (HxWxCh.)
Input			128 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 128 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 128
	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 Fig. 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 128
	2	2x2 Upsampling Feature maps concatenation 3x3 Convolution → Batch normalization → ReLU 3x3 Convolution → Batch normalization → ReLU	64 x 128 x 64
	1	2x2 Upsampling Feature maps concatenation 3x3 Convolution → Batch normalization → ReLU 3x3 Convolution → Batch normalization → ReLU	128 x 256 x 32
Output		1x1 Convolution → Sigmoid	128 256 x 1

3.3. Evaluation Metrics and Loss Functions

For the segmentation task, the loss function combines binary cross-entropy (BCE) loss [38] with dice loss [39], [40], while the evaluation metric used is the Jaccard Index or Intersection over Union (IoU). Binary cross-entropy is used as the training loss function, simplifying the output layer by reducing the required number of nodes [41]. The formulations for both BCE and dice loss are presented in equation (1) and (2), respectively.

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

Here, N represents the total number of neurons in the output layer, calculated as 128 × 256 × 1 = 32,768 (width × height × number of classes), where y_i denotes the ground truth value and p_i refers to the predicted value.

$$Dice\ Loss = 1 - \frac{2\,|Yt \square\square\ Yp|}{|Yt| + |Yp|} \quad (2)$$

In this context, Yt represents the ground-truth value, while Yp refers to the predicted value. The overall loss is then computed as shown in equation (3).

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

The BCEDice loss is the combination of BCE loss and dice loss, where both coefficients "α" and "β" are assigned a value of 1, giving equal importance to each component in influencing the model. Additionally, this study incorporates L2 regularization, as proposed by Tessier et al. [42], with a weight decay of 0.0001 to reduce model complexity and enhance generalization. As a result, the overall loss is computed using equation (4). BCEDice was selected because it provided the most stable performance during experimentation, and explained that although focal loss was considered, its instability in early training led us to defer its use to future optimization efforts.

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

In this equation, w_d denotes the weight decay, w_i represents the individual model weights, and N is the total number of weights in the model. The metric function applied in this study is presented in equation (5).

$$J(Yt, Yp) = \frac{|Yt \cap Yp|}{|Yt \cup Yp|} = \frac{|Yt \cap Yp|}{|Yt|+|Yp|-|Yt \cap Yp|} \tag{5}$$

Here, J stands for the Jaccard Index, also referred to as Intersection over Union (IoU), with Yt representing the ground-truth value and Yp indicating the predicted value.

3.4. Training Configuration

The training was carried out on a GPU based virtual machine using cloud computing, equipped with a Tesla V100 provided by a cloud service. To enhance the model's generalization ability, data augmentation techniques such as rotation, resizing, normalization, saturation, contrast, and brightness adjustments were applied during image pre-processing [43]. The training used a batch size of 8, with each image resized to 128x256 pixels and 3 color channels (RGB). The dataset consisting of 893 images was divided into 80 percent for training and 20 percent for validation, resulting in 714 training images and 179 validation images.

As applied in previous studies, the model is trained using the stochastic gradient descent (SGD) algorithm [44], with an initial learning rate of 0.25 and a momentum of 0.9. To justify the use of a relatively high initial learning rate, we clarified that such a setting can be effective for accelerating early convergence, particularly when training on smaller-resolution industrial datasets where gradients tend to stabilize rapidly. We also added citations to prior studies employing similarly elevated learning rates for UNet and UNet++ variants under comparable training conditions, supporting the appropriateness of this choice. Furthermore, we explicitly stated that the learning rate is not fixed throughout training; instead, it is gradually reduced using a Reduce-on-Plateau scheduler, ensuring stable optimization as the model approaches convergence. The stochastic gradient descent algorithm, or in its application, is better known as the SGD algorithm, which is part of the machine learning algorithm for classification, regression and artificial neural networks and this algorithm is very efficient on large-scale datasets [45]. During training, the learning rate is halved if the validation loss does not improve over 15 consecutive epochs, with a minimum threshold set at 0.000025. Although the training is allowed to run up to 1,500 epochs, it will automatically stop if the validation IoU shows no improvement for 150 consecutive epochs. The training mechanism, which employs a batch processing approach, is illustrated in Fig. 6. During the training phase, a batch of 8 images is passed through the model to generate predicted masks for each image. These predictions are then compared to the corresponding ground truth masks to compute the loss and IoU. The average loss and IoU for each batch are also calculated, with the average batch loss used to update the model's weights during backpropagation. This process continues until all batches are processed. Once all batches in an epoch are completed, the average loss and IoU for the entire epoch are calculated and recorded to track model performance. The training loop runs for up to 1,500 epochs but will terminate early if no improvement in validation IoU is observed over 150 consecutive epochs. Equation (6) and (7) are showing how to calculate and differentiate between the average batch loss with the average loss. This is also applied for average batch IoU and average IoU.

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

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

In this context, Avg refers to the average, N represents the number of images per batch which is 8, and M denotes the total number of batches. Given a dataset of 893 images (714 for training and 179 for validation), there are 90 training batches and 23 validation batches. The validation phase follows a process similar to the training phase, with the key difference being the absence of backpropagation, meaning the model weights are not updated. As previously explained, the average loss and IoU per epoch are used as criteria for reducing the learning rate or terminating the training process.

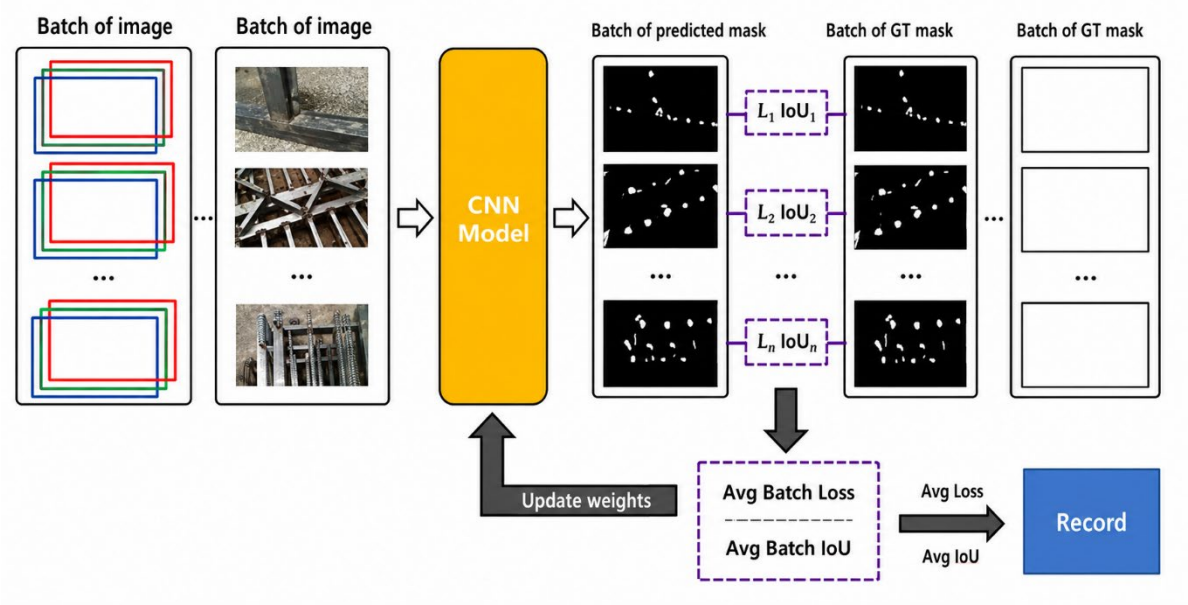


Fig. 6. Training Mechanism using Batch Processing Technique.

4. Results and Discussion

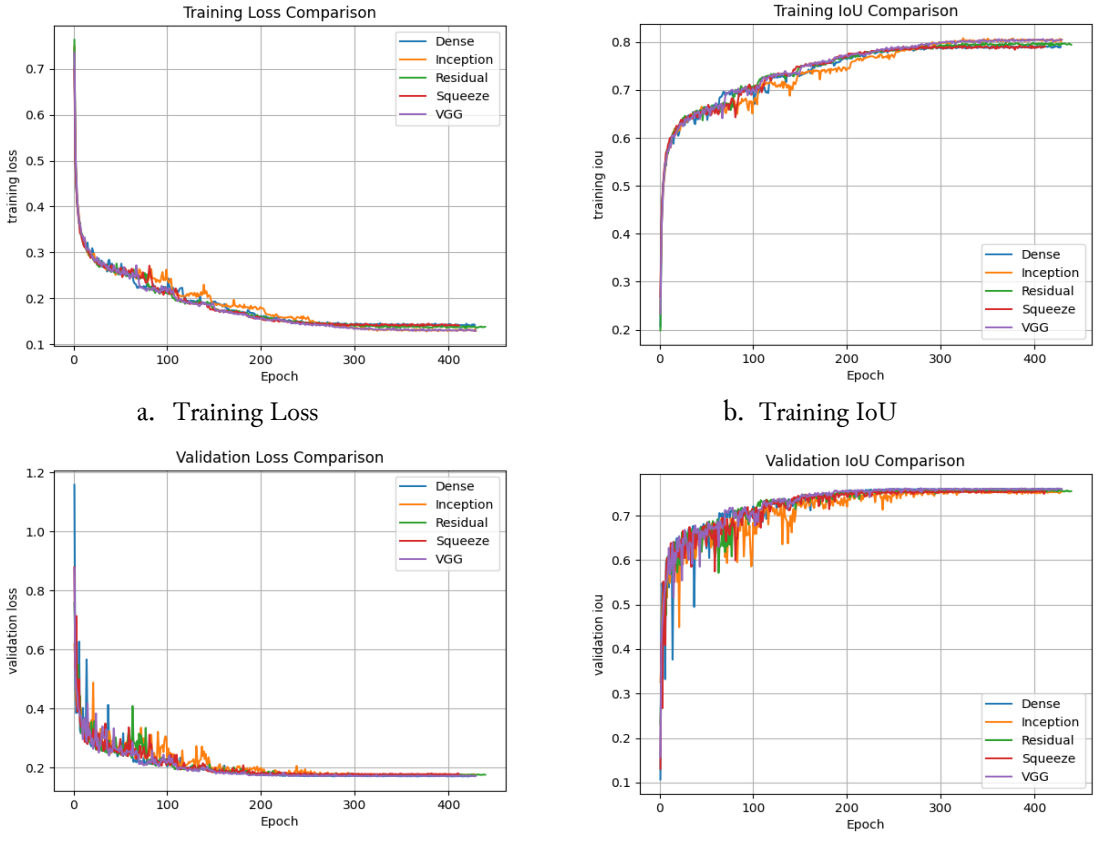
As stated in the previous chapter, the model's performance is evaluated using IoU and loss as key metrics. Throughout the training process, the performance on both the training and validation datasets is continuously monitored and recorded. As a result, multiple logs will be generated for training loss, validation loss, training IoU, and validation IoU for each of the five types of convolutional blocks used in the bottleneck. The comparison of these five bottleneck variations across each parameter is illustrated in Fig. 7.

The model can be considered neither overfitting nor underfitting, as the validation loss and validation IoU closely align with the training loss and training IoU. As previously noted, the training process is halted when there is no improvement in validation IoU, which explains why each model may complete training at a different number of epochs. Further details of the comparison can be found in Table 2. The inception block shows the best performance on the training data, achieving an IoU score of 0.8081 and a loss of 0.1277, followed closely by the VGG block with an IoU of 0.8063 and a loss of 0.1284. On the validation data, the VGG block performs best with an IoU of 0.7618, followed by the dense block with an IoU of 0.7610. In terms of validation loss, the dense block records the lowest value at 0.1706, with the VGG block coming next at 0.1713. Regarding convergence speed, the squeeze block completes training in the fewest epochs 411, followed by the dense block with 428 epochs. In deep learning, convergence refers to the model's ability to learn and respond accurately to training patterns within an acceptable error margin [46].

Table 2. Inference Result on Test Images.

Model	Conv. Block	Highest Training IoU	Lowest Training Loss	Highest Validation IoU	Lowest Validation Loss	Total Epochs
UNet-based models [4]	Dense	0.7960	0.1374	0.7545	0.1769	409
	Inception	0.7789	0.1500	0.7485	0.1812	362
	Residual	0.7936	0.1388	0.7540	0.1778	658
	Squeeze	0.7996	0.1341	0.7530	0.1782	443
	VGG	0.7800	0.1494	0.7502	0.1803	647
Nested UNet-based models (ours)	Dense	0.7923	0.1395	0.7610	0.1706	428
	Inception	0.8081	0.1277	0.7592	0.1718	429
	Residual	0.7986	0.1344	0.7582	0.1735	439
	Squeeze	0.7930	0.1394	0.7559	0.1754	411
	VGG	0.8063	0.1284	0.7618	0.1713	429

Among the evaluated parameters, the highest IoU on the validation data is used as the primary criterion for determining the best model performance. Since model weights are updated using training data, strong performance during training alone is not meaningful if the results on validation data are poor or average. Similarly, faster convergence is not a decisive factor if the model's performance is inadequate, as this metric is not relevant for deployment. On the other hand, achieving the highest validation IoU or the lowest validation loss indicates better generalization capability during deployment. Based on these considerations, the Nested UNet model with a VGG convolutional block in the bottleneck demonstrates the best overall performance, as it achieves the highest validation IoU and ranks among the top two for both training IoU and validation loss. An example of the model's inference output is shown in Fig. 8.



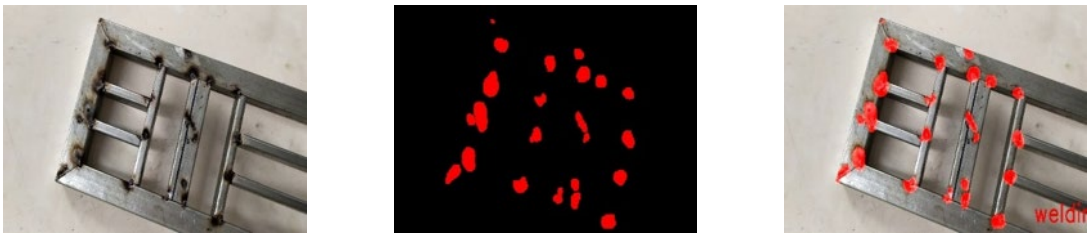
a. Training Loss b. Training IoU

c. Validation Loss d. Validation IoU

Fig. 7. Model's Performance Comparison.

In addition to the numerical comparison, we incorporated a qualitative analysis of the segmentation outputs to better understand model behavior across challenging scenarios. The VGG-based bottleneck demonstrates consistently sharper and more coherent weld-spot boundaries, showing strong ability to preserve fine-grained local structures. In contrast, Dense and Inception blocks occasionally over-segment surrounding metal textures, likely due to their stronger emphasis on multi-scale aggregation and high-connectivity patterns that are not dominant in weld-spot shapes. We also discuss observable failure modes, particularly under low-lighting conditions, high surface reflectivity, or intense welding sparks, where all models, including the VGG variant, may partially miss weld spots or produce slight boundary inaccuracies. These qualitative evaluations complement the quantitative results and offer deeper insight into the strengths and limitations of each architectural choice. Even small IoU improvements meaningfully reduce missed weld spots, increase robustness to sparks and welding noise, and yield sharper boundaries that lower downstream errors in automated tasks such as weld-spot counting and quality estimation.

We also added a hypothesis to explain why the VGG bottleneck outperforms the more complex alternatives. VGG's stacked 3×3 convolutional layers are highly effective at preserving fine-grained local details, enabling the model to capture the small, bright, and highly localized appearance of weld spots with greater precision. In contrast, multi-scale modules such as Inception, or connectivity-heavy structures like Dense and Residual blocks, tend to emphasize broader contextual patterns that may oversmooth or dilute the subtle boundary cues of weld spots. Given that weld spots occupy only small regions within high-texture metal backgrounds, the ability to retain strong local spatial representation becomes more valuable than global multi-scale feature extraction, which supports the superior performance observed in the VGG-based configuration.



a. 1st Input Image a. 1st Predicted Mask. a. 1st Apply Mask

b. 2nd Input Image b. 2nd Predicted Mask c. 2nd Apply Mask

Fig. 8. Inference Result on Test Images.

5. Conclusion

This research concludes that the Nested UNet model with a VGG convolutional block in the bottleneck section has the best performance compared to the others. The VGG block is identified as the best based on its highest validation IoU score of 0.7618, which is the primary metric used due to its indication of the model's generalization on unseen data. Additionally, the VGG block ranks second in both training IoU score of 0.8063 and training loss score of 0.1284. While training metrics and convergence time provide supporting insights, they are considered secondary as they do not directly reflect the model's performance during real-world deployment. The key novelty of this study is the systematic evaluation of five convolutional bottleneck modules within a redesigned Nested UNet (UNet++) architecture, showing that VGG's stacked 3×3 convolutions provide the most effective

representation for weld-spot segmentation. Future work may focus on extending the model to assess welding quality and defects, as well as developing a post-processing algorithm to count welding spots. Further exploration could involve modifying or deepening convolutional blocks, optimizing hyperparameters automatically, and reducing model size and complexity for faster deployment on specific devices.

Acknowledgment

The authors thank Politeknik Perkapalan Negeri Surabaya, Universitas Gadjah Mada and The University of Osaka, for their support of this work.

Declarations

Author contribution. Thomas Brian: conceptualization, methodology, writing, data curation, data analysis, software, and data evaluation. Anggarjuna Puncak Pujiputra, Oskar Natan, and Yohanes Yohanie Fridelin Panduman: supervision, conceptualization, and writing review.
Funding statement. This research is self-funded.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] T. W. Purnomo, F. Danitasari, and D. Handoko, "Weld Defect Detection and Classification based on Deep Learning Method: A Review," *J. Ilmu Komput. dan Inf.*, vol. 16, no. 1, pp. 77–87, Mar. 2023, doi: 10.21609/jiki.v16i1.1147.

[2] R. Ghimire and R. Selvam, "Machine Learning-Based Weld Classification for Quality Monitoring," in *RAiSE-2023, Basel Switzerland: MDPI*, Mar. 2024, p. 241. doi: 10.3390/engproc2023059241.

[3] S. Y. Prasetyo, G. Z. Nabiilah, Z. N. Izdihar, and S. M. Isa, "Pneumonia Detection on X-Ray Imaging using Softmax Output in Multilevel Meta Ensemble Algorithm of Deep Convolutional Neural Network Transfer Learning Models," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 2, p. 319, Jul. 2023, doi: 10.26555/ijain.v9i2.884.

[4] D. Palma-Ramírez et al., "Deep convolutional neural network for weld defect classification in radiographic images," *Heliyon*, vol. 10, no. 9, p. e30590, May 2024, doi: 10.1016/j.heliyon.2024.e30590.

[5] M. Murinto and S. Winiarti, "Modified particle swarm optimization (MPSO) optimized CNN's hyperparameters for classification," *Int. J. Adv. Intell. Informatics*, vol. 11, no. 1, p. 133, Feb. 2025, doi: 10.26555/ijain.v11i1.1303.

[6] N. A. M. Roslan, N. M. Diah, Z. Ibrahim, Y. Munarko, and A. E. Minarno, "Automatic plant recognition using convolutional neural network on malaysian medicinal herbs: the value of data augmentation," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, p. 136, Mar. 2023, doi: 10.26555/ijain.v9i1.1076.

[7] R. Cahyaningtyas, S. Madenda, B. Bertalya, and D. Indarti, "Solar module defects classification using deep convolutional neural network," *Int. J. Adv. Intell. Informatics*, vol. 11, no. 3, p. 499, Aug. 2025, doi: 10.26555/ijain.v11i3.1818.

[8] 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," *Emerg. Sci. J.*, vol. 7, no. 2, pp. 498–506, Feb. 2023, doi: 10.28991/ESJ-2023-07-02-014.

[9] A. Desiani et al., "Simple Data Augmentation and U-Net CNN for Neclui Binary Segmentation on Pap Smear Images," *J. Electron. Electromed. Eng. Med. Informatics,* vol. 6, no. 3, pp. 264–275, Jul. 2024, doi: 10.35882/jeeemi.v6i3.442.

[10] W. Baccouch, S. Oueslati, B. Solaiman, and S. Labidi, "A comparative study of CNN and U-Net performance for automatic segmentation of medical images: application to cardiac MRI," *Procedia Comput. Sci.*, vol. 219, pp. 1089–1096, Jan. 2023, doi: 10.1016/j.procs.2023.01.388.

[11] O. Natan, D. U. K. Putri, and A. Dharmawan, "Deep learning-based weld spot segmentation using modified unet with various convolutional blocks," *ICIC Express Lett. Part B Appl.*, vol. 12, no. 12, pp. 1169–1176, Dec. 2021, Accessed: Jun. 06, 2026. [Online]. Available: https://www.researchgate.net/publication/355732437_Deep_Learning-Based_Weld_Spot_Segmentation_Using_Modified_UNet_with_Various_Convolutional_Blocks.

[12] J. Videira, P. D. Gaspar, V. N. da G. de J. Soares, and J. M. L. P. Caldeira, "Detecting and monitoring the development stages of wild flowers and plants using computer vision: approaches, challenges and opportunities," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 3, p. 347, Oct. 2023, doi: 10.26555/ijain.v9i3.1012.

[13] R. N. Razak and H. N. Abdullah, "Improving multi-object detection and tracking with deep learning, DeepSORT, and frame cancellation techniques," *Open Eng.*, vol. 14, no. 1, pp. 1–18, Sep. 2024, doi: 10.1515/eng-2024-0056.

[14] N. Shamna and B. Aziz Musthafa, "Feature Extraction Method using HoG with LTP for Content-Based Medical Image Retrieval," *Int. J. Electr. Comput. Eng. Syst.,* vol. 14, no. 3, pp. 267–275, Mar. 2023, doi: 10.32985/ijeces.14.3.4.

[15] D. F. Abdulabass and M. E. Abdulmunim, "Traffic Sign Detection Using You Only Look Once (YOLOv3) Technique," *Iraqi J. Sci.*, vol. 65, no. 10, pp. 5741–5753, Oct. 2024, doi: 10.24996/IJS.2024.65.10.34.

[16] D. Liu, D. Zhang, L. Wang, and J. Wang, "Semantic segmentation of autonomous driving scenes based on multi-scale adaptive attention mechanism," *Front. Neurosci.*, vol. 17, p. 1291674, Oct. 2023, doi: 10.3389/fnins.2023.1291674.

[17] S. Mhatre, A. Singh, and S. Shinde, "Enhancing Nucleus and Cytoplasm Segmentation in Cervical Cytology Using Deep Learning," in *2024 IEEE Pune Section International Conference (PuneCon)*, IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/PuneCon63413.2024.10895296.

[18] 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, no. 15, p. e34738, Aug. 2024, doi: 10.1016/j.heliyon.2024.e34738.

[19] M. Vadduri, "DAUCD: Deep Attention U-Net for Cataract Detection Leveraging CNN Frameworks," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 1, p. 2025, doi: 10.22266/ijies2025.0229.15.

[20] 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," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 4, 2024, doi: 10.22266/ijies2024.0831.34.

[21] I. B. Santoso, Supriyono, and S. N. Utama, "Multi-Model of Convolutional Neural Networks for Brain Tumor Classification in Magnetic Resonance Imaging Images," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 5, pp. 741–758, Oct. 2024, doi: 10.22266/ijies2024.1031.56.

[22] A. ANHAR and R. A. PUTRA, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 2, p. 466, Apr. 2023, doi: 10.26760/elkomika.v11i2.466.

[23] S. U. Masruroh, A. Fiade, M. I. Tanggok, R. A. Putri, and L. A. Pratiwi, "Convolutional Neural Network for Colorization of Black and White Photos," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 2, pp. 393–404, Mar. 2023, doi: 10.30812/matrik.v22i2.2652.

[24] S. Kumaresan, K. S. J. Aultrin, S. S. Kumar, and M. D. Anand, "Deep learning-based weld defect classification using VGG16 transfer learning adaptive fine-tuning," *Int. J. Interact. Des. Manuf.*, vol. 17, no. 6, pp. 2999–3010, Dec. 2023, doi: 10.1007/s12008-023-01327-3.

[25] V. Yadav, D. Banerjee, N. Sharma, and V. Singh, "Automated Welding Defect Recognition through Deep Learning Fusion: CNN and SVM Integration," i*n 2023 4th International Conference on Intelligent Technologies (CONIT)*, IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/CONIT61985.2024.10627403.

[26] Y. Yang, S. Dasmahapatra, and S. Mahmoodi, "ADS_UNet: A nested UNet for histopathology image segmentation," *Expert Syst. Appl.*, vol. 226, no. September, p. 120128, Sep. 2023, doi: 10.1016/j.eswa.2023.120128.

[27] Z. Jian et al., "An Improved Nested U-Net Network for Fluorescence In Situ Hybridization Cell Image Segmentation," *Sensors*, vol. 24, no. 3, p. 928, Jan. 2024, doi: 10.3390/s24030928.

[28] C. Mathews and A. Mohamed, "Nested U-Net with Enhanced Attention Gate and Compound Loss for Semantic Segmentation of Brain Tumor from Multimodal MRI," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 4, p. 2022, Aug. 2022, doi: 10.22266/ijies2022.0831.23.

[29] M. A. Sundaram, "Channel and Spatial Attention Aware UNet Architecture for Segmentation of Blood Vessels, Exudates and Microaneurysms in Diabetic Retinopathy," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 2, pp. 1–16, Apr. 2024, doi: 10.22266/ijies2024.0430.01.

[30] W. Zhang, Y. Liu, Y. Feng, L. Quan, G. Zhang, and C. Zhang, "Deep-broad learning network model for precision identification and diagnosis of grape leaf diseases," *Front. Plant Sci.*, vol. 16, p. 1611301, Sep. 2025, doi: 10.3389/FPLS.2025.1611301/TEXT.

[31] S. Lika, Y. Pernando, and A. Kurniawan, "Lightweight Deep Learning for Object Detection on Mobile Device," Bull. Informatics Data Sci., vol. 2, no. 2, p. 106, Nov. 2023, doi: 10.61944/bids.v2i2.82.

[32] B. K. Triwijoyo, A. Adil, and A. Anggrawan, "Convolutional Neural Network With Batch Normalization for Classification of Emotional Expressions Based on Facial Images," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 197–204, Nov. 2021, doi: 10.30812/matrik.v21i1.1526.

[33] Herman et al., "A Comperative Study on Efficacy of CNN VGG-16, DenseNet121, ResNet50V2, And EfficientNetB0 in Toraja Carving Classification," *Indones. J. Data Sci.*, vol. 6, no. 1, pp. 123–133, Mar. 2025, doi: 10.56705/ijodas.v6i1.220.

[34] B. Tasci, M. R. Acharya, M. Baygin, S. Dogan, T. Tuncer, and S. B. Belhaouari, "InCR: Inception and concatenation residual block-based deep learning network for damaged building detection using remote sensing images," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 123, no. September, p. 103483, Sep. 2023, doi: 10.1016/j.jag.2023.103483.

[35] K. Benali, L. Bouzar-Benlabiod, and A. McIntyre, "Squeeze and Excitation Block Based Neural Architecture Search with Randomization for CNN Construction," in *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI),* IEEE, Aug. 2024, pp. 49–54. doi: 10.1109/IRI62200.2024.00022.

[36] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," Appl. Sci., vol. 12, no. 18, p. 8972, Sep. 2022, doi: 10.3390/app12188972.

[37] W. W. Kusuma, R. R. Isnanto, and A. Fauzi, "Comparison Analysis of CNN Model VGG16 and DenseNet121 using TensorFlow Framework to Detect Animal Images," J. Tek. Komput., vol. 1, no. 4, pp. 141–147, 2023, [Online]. Available: https://ejournal3.undip.ac.id/index.php/jtk/article/view/37009/28851.

[38] A. U. Ruby, Prasannavenkatesan Theerthagiri, Jeena Jacob, and Y.Vamsidhar, "Binary cross entropy with deep learning technique for Image classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020, doi: 10.30534/ijatcse/2020/175942020.

[39] Q. Ming and X. Xiao, "Towards Accurate Medical Image Segmentation With Gradient-Optimized Dice Loss," *IEEE Signal Process. Lett.,* vol. 31, pp. 191–195, 2024, doi: 10.1109/LSP.2023.3329437.

[40] M. Yeung, L. Rundo, Y. Nan, E. Sala, C.-B. Schönlieb, and G. Yang, "Calibrating the Dice Loss to Handle Neural Network Overconfidence for Biomedical Image Segmentation," *J. Digit. Imaging*, vol. 36, no. 2, pp. 739–752, Dec. 2022, doi: 10.1007/s10278-022-00735-3.

[41] J. Kim and H.-S. Lim, "Neural Network With Binary Cross Entropy for Antenna Selection in Massive MIMO Systems: Convolutional Neural Network Versus Fully Connected Network," *IEEE Access*, vol. 11, pp. 111410–111421, 2023, doi: 10.1109/ACCESS.2023.3322679.

[42] H. Tessier, V. Gripon, M. Léonardon, M. Arzel, T. Hannagan, and D. Bertrand, "Rethinking Weight Decay for Efficient Neural Network Pruning," *J. Imaging*, vol. 8, no. 3, p. 64, Mar. 2022, doi: 10.3390/jimaging8030064.

[43] M. S. Harisha, A. A. Bhosale, and M. Narender, "Deep learning-powered segmentation and classification of diabetic retinopathy for enhanced diagnostic precision," *Artif. Intell. Heal.*, vol. 1, no. 4, p. 30, Sep. 2024, doi: 10.36922/aih.2783.

[44] Y. Tian, Y. Zhang, and H. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning," *Mathematics*, vol. 11, no. 3, p. 682, Jan. 2023, doi: 10.3390/math11030682.

[45] F. T. Admojo and Y. I. Sulistya, "Analisis Performa Algoritma Stochastic Gradient Descent (SGD) Dalam Mengklasifikasi Tahu Berformalin," *Indones. J. Data Sci.*, vol. 3, no. 1, pp. 1–8, Mar. 2022, doi: 10.56705/ijodas.v3i1.42.

[46] Y. Jiao, X. Lu, P. Wu, and J. Z. Yang, "Convergence Analysis for Over-Parameterized Deep Learning," *Commun. Comput. Phys.*, vol. 36, no. 1, pp. 71–103, Jul. 2024, doi: 10.4208/cicp.OA-2023-0264.