Available online at www.CivileJournal.org

Civil Engineering Journal

(E-ISSN: 2476-3055; ISSN: 2676-6957)

Vol. 11, No. 11, November, 2025

Evaluation of Flood Inundation Image Detection Performance Using Deep Learning

Arief A. Soebroto [1*], Lily M. Limantara [2], Ery Suhartanto [2], Moh. Sholichin [2], Fatwa Ramdani [3], Turniningtyas A. Rachmawati [4]

[1] *Department of Informatics, Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia.*

[2] *Department of Water Resources Engineering, Faculty of Engineering, Universitas Brawijaya, Malang, Indonesia.*

[3] *Graduate School of Humanities and Social Sciences, University of Tsukuba, Tsukuba 305-8577, Japan.*

[4] *Department of Urban and Regional Planning, Faculty of Engineering, Universitas Brawijaya, Malang, Indonesia.*

Received 08 July 2025; Revised 11 October 2025; Accepted 15 October 2025; Published 01 November 2025

Abstract

Floods are the most frequently occurring natural disasters, significantly impacting the environment and society. As part of natural disaster mitigation, the impacts could be reduced through predictive techniques using deep learning for semantic segmentation of inundation images. Therefore, this research aims to evaluate the performance of deep learning architectures in segmenting inundation images using the Flood Segmentation dataset, which comprised 290 aerial images. The following segmentation architectures, U-Net, SegNet, and LinkNet, were compared using backbones such as MobileNet, ResNet, EfficientNet, and VGG, as well as optimizers including Adam, SGD, AdaDelta, and RMSProp. Performance was assessed using Intersection over Union (IoU) score, precision, F1-score, recall, and accuracy metrics. The results showed that U-Net achieved the highest performance with IoU, precision, F1-score, recall, and accuracy of 0.767, 0.862, 0.866, 0.876, and 0.899, respectively. Regarding the backbones, MobileNet excelled with IoU, precision, F1-score, recall, and accuracy of 0.764, 0.866, 0.865, 0.869, and 0.898, respectively. The Adam optimizer outperformed others, yielding IoU, precision, F1-score, recall, and accuracy of 0.712, 0.807, 0.824, 0.873, and 0.843. In conclusion, the combination of U-Net with MobileNet backbone and Adam optimizer was the most effective architecture for flood inundation image segmentation, offering a robust foundation for prediction systems.

Keywords: Deep Learning; Disaster Mitigation; Flood; Image Segmentation; Natural Disaster.

1. Introduction

Floods are frequently occurring natural disasters, which have a significant impact on the environment and society. This condition causes huge amounts of water to cover terrestrial areas, thereby destroying agricultural land and important infrastructure, displacing human populations, and disrupting economic activities, as well as leading to epidemics and deaths [1]. Several efforts have been implemented through disaster mitigation to prevent or reduce the associated risks and impacts [2, 3]. In this context, flood prediction plays a significant role in terms of enabling decision-makers to adopt preventive action and respond rapidly [4]. The process entailed the use of an image segmentation method to identify vulnerable areas [5], detected by separating the image into various parts based on certain characteristics [6]. Semantic segmentation allows pixel-level annotation, compared to object detection or classification that provides bounding boxes

* Corresponding author: ariefas@ub.ac.id

http://dx.doi.org/10.28991/CEJ-2025-011-11-08

or class labels per image. This pixel-wise precision accurately delineates the extent of inundated areas and supports localized decision-making during flood mapping. The approach is commonly applied to separate objects such as land, water, and buildings, providing essential information for identifying vulnerable regions [7]. Moreover, the identification of these objects requires semantic image segmentation, which is responsible for the division into separate areas with distinctive interpretations [8, 9].

Semantic segmentation refers to the process of separating images based on pixels, labelled in line with certain object classes or purposes [10]. This approach is essential in the development of image processing to provide a deeper understanding of the visual content [11]. The application of semantic segmentation can be carried out using deep learning-based methods to produce exceptional performances [12]. Deep learning has the advantage of capturing complex feature patterns and performing extraction automatically [13, 14]. Additionally, it provides detailed information on the flood inundation image in the resulting abstract features. Deep learning architecture used in image segmentation included SegNet, U-Net, and LinkNet [15, 16]. Regarding this perspective, U-Net produced accurate segmentation due to the deep encoding-decoding feature [17]. LinkNet plays an efficient role in carrying out image segmentation [18], with SegNet showing spatial information relevant for understanding the context of the affected areas [19].

Based on the description above, this research aims to evaluate flood inundation image segmentation to significantly contribute to the development of related prediction systems. Performance comparison was carried out between deep learning architectures such as U-Net, LinkNet, and SegNet to identify the most optimal. Additionally, the results provided valuable information for decision-makers and relevant teams in terms of selecting the most effective architecture for improving accuracy and response to flood, including serving as a reference for future prediction.

2. Research Methodology

The development of image segmentation for flood inundation was conducted to enable the accurate identification of the affected areas, as well as support timely disaster response. In addition, semantic segmentation provided pixel-level classification, offering a more granular view of inundation zones compared to traditional object detection. This precision supported decision-making related to the mitigation process and recovery planning by facilitating localized analysis of flood extents and patterns.

The flow diagram for flood inundation segmentation is shown in Figure 1, with the initial stage focused on the collection of data from various relevant sources, including visual sensing systems. The collected information was subjected to preprocessing, as well as resizing to 512×512 pixels. This was aimed to ensure consistency with model input and data augmentation through horizontal flipping and brightness and contrast adjustment, including spatial transformations such as random shifting, scaling, and rotation to increase information diversity. Furthermore, the pixel intensities were normalized using ImageNet standards to achieve consistent data distribution. The preprocessed information was fed into deep learning models using three popular segmentation architectures, namely U-Net, SegNet, and LinkNet, combined with the following backbones: MobileNet, ResNet, or EfficientNet to balance accuracy and computational efficiency. The training process enabled the models to learn the relationship between input images and flood masks by capturing relevant spatial and contextual patterns. The results of the predicted segmentation were evaluated using separate test data, and the performance was measured with standard metrics. It included Intersection over Union (IoU), precision, recall, and F1-Score, derived from the confusion matrix.

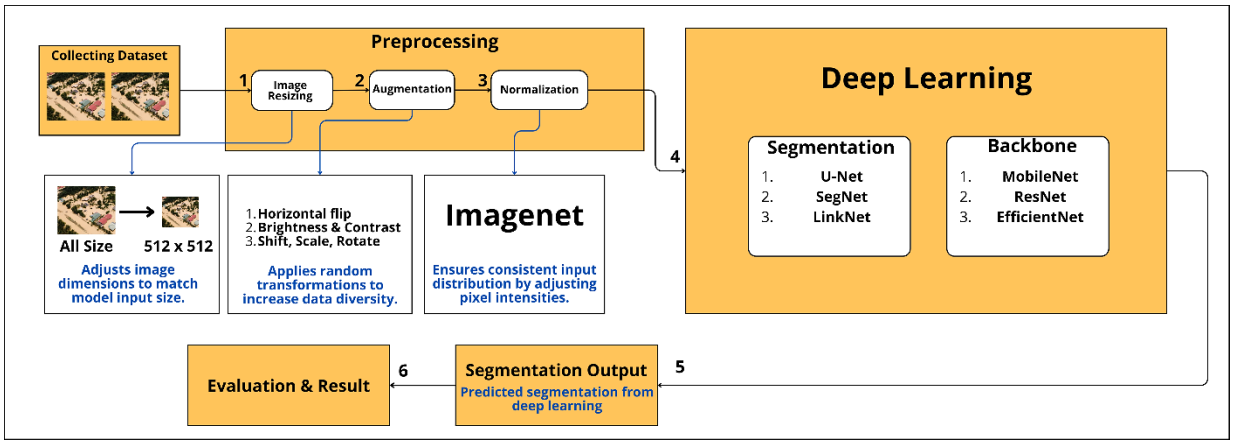


Figure 1. Flood Segmentation Flowchart

2.1. Deep Learning

This stage focused on performing image segmentation by adopting several well-established convolutional neural network architectures. The process started with the training of the flood inundation image dataset using three widely

identified models, namely U-Net, SegNet, and LinkNet. Additionally, each architecture research contributed distinct strengths to the segmentation task, as shown in Table 1.

Table 1. Architecture Research

Architecture	IoU	F1-Score
U-Net [20]	0.974	0.987
SegNet [21]	0.859	0.849
LinkNet [20]	0.975	0.987

The selection of U-Net, SegNet, and LinkNet architectures was based on previous research results that outlined the strengths and limitations of each in image segmentation tasks. U-Net was selected because it exhibited high performance, achieving IoU and F1-scores of 0.974 and 0.987, respectively, which consistently outperformed other methods in various research. SegNet was compared because, despite achieving an IoU and F1-score of 0.859 and 0.849, respectively, the architecture was widely used during initial image segmentation research, making it relevant as a baseline. The selection of LinkNet was due to the realization of performance comparable to U-Net, with IoU and F1-Score of 0.975 and 0.987. This architecture also offered better computational efficiency, leading to the attractive nature during practical applications.

U-Net, characterized by the symmetric encoder and decoder structure that allowed spatial information to be preserved in the network, is essential for generating precise segmentation outputs. Furthermore, SegNet adopted a similar encoder-decoder approach, which enhanced memory efficiency. This was realized by storing max-pooling indices during the encoding phase, including the reuse in the decoder to reconstruct spatial resolution accurately. LinkNet offered an alternative approach by introducing residual connections between the encoder and decoder, which improved the flow of gradients during training as well as reduced computational cost while maintaining reliable performance. These models were successfully applied in various environmental and remote sensing tasks, particularly in flood detection scenarios where spatial precision played a critical role. The results of the segmentation were further optimized by combining each architecture with backbone networks such as ResNet, MobileNet, EfficientNet, and VGG. The selection of these backbones was based on respective complementary capabilities, where ResNet enabled deeper feature extraction through residual learning. Additionally, MobileNet offered lightweight architecture suitable for real-time applications, and EfficientNet provided a balanced scaling strategy that enhanced both accuracy and efficiency, with VGG remaining a strong baseline model due to its simplicity and consistent output. During the training phase, a range of optimizers, including Adam, SGD, AdaDelta, and RMSProp, was explored. Hyperparameter tuning was also carried out to determine the most effective combination of architecture, backbone, and optimizer for achieving excellent performance in flood inundation image segmentation.

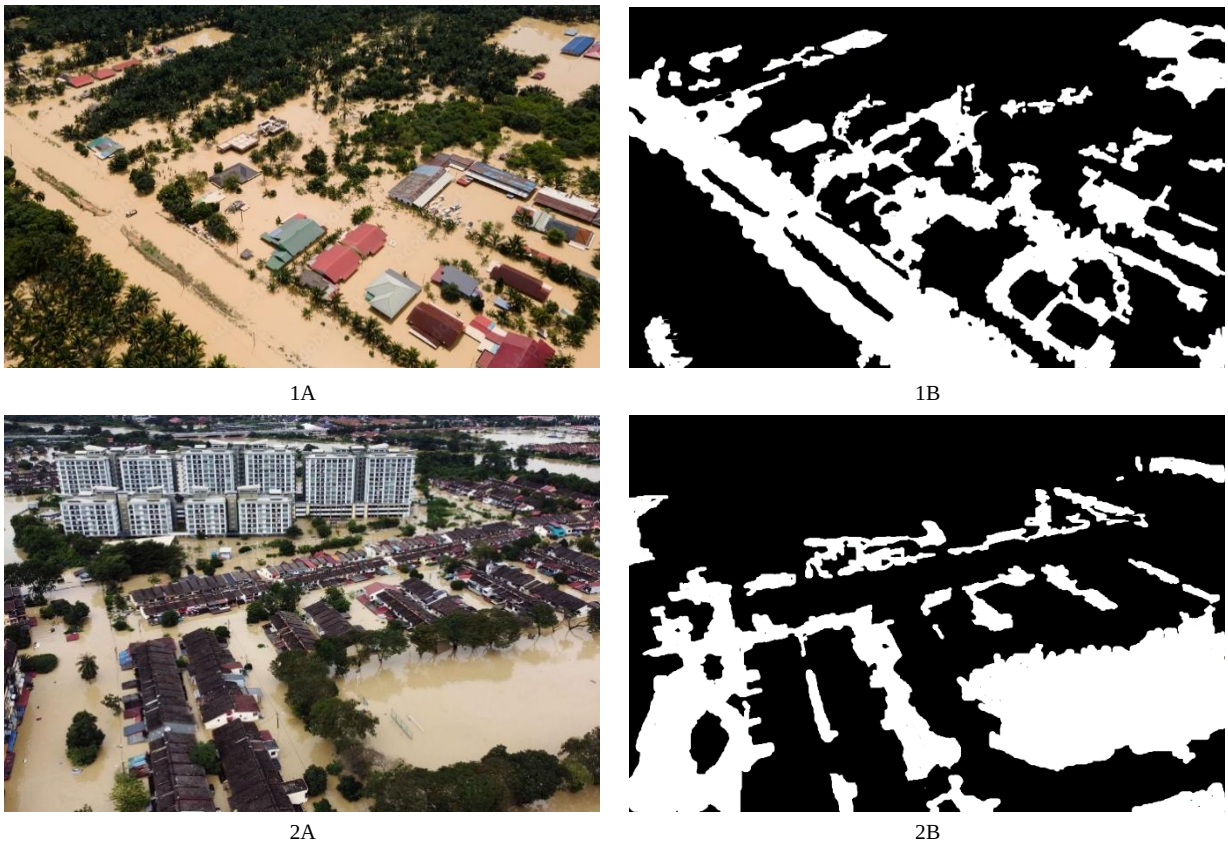
Figure 2 shows a flowchart of the general process starting with the training dataset, which consisted of 290 flood inundation images. This process was carried out by training a combination of the architecture, backbone, and optimizer for 50 epochs. Dice loss was used during training to measure the spatial overlap between predicted and ground truth masks, ensuring pixel-wise accuracy [22]. Subsequently, the models were evaluated by conducting a separate test to assess respective generalization performance. The main performance metrics included IoU Score, as well as accuracy, recall, precision, and F1-Score derived from the confusion matrix [15, 23]. These evaluation metrics were used to systematically determine the most optimal configuration of the architecture, backbone, and optimizer, thereby supporting the development of an efficient and accurate flood inundation segmentation model.

Figure 2. Deep Learning Flowchart

2.1.1. Datasets

The Flood Segmentation dataset [24], containing 290 images of inundation captured from above, was used. The dataset label used was a mask of the flood inundation image, often applied during the prediction of affected areas to improve the accuracy and response to this disaster.

Figure 3 shows the dataset of flood inundation images, comprising 1A, 2A, 3A, and 4A as the originals, while 1B, 2B, 3B, and 4B formed the segmentation images. Specifically, 1A shows an affected area with brown water submerging the building location. 1B is a segmented version of 1A, with the yellow and purple colors representing flood inundation and building locations, respectively. In addition, image 2A also shows the flood area with brown water submerging the building location. 2B is a segmented version of the image with yellow and purple colors depicting flood inundation and building locations, respectively.



1A 1B

2A 2B

Figure 3. Flood Dataset

2.1.2. U-Net

U-Net is a neural network architecture introduced in the biomedical field for developing the segmentation process by testing HeLa cell images [25]. During development, it has the potential to segment flood images [26], due to the contracting path encoder structure useful for extracting features from input data. This neural network architecture is also characterized by an expansion path decoder for restructuring images from the extracted features [27]. The encoder process was carried out using several convolution (Conv2D) layers to reduce image dimensions [28]. Additionally, the decoder process was conducted with several transposition convolution layers to increase image dimensions.

In Figure 4, the U-Net architecture started with the first encoder (Enc 1), which performed convolution, batch normalization (Batch Norm), and the ReLU activation function. These were actualized by changing dimension from 3 (RGB) to 64 and ending with downsampling to enter the next encoder layer. Meanwhile, the second encoder (Enc 2) performed convolution, batch normalization, and the ReLU activation function to change from dimension 64 to 128, ending with downsampling to enter the next layer. The third encoder (Enc 3) performed convolution, batch normalization, and the ReLU activation function to change from dimension 128 to 256 and ended with downsampling. The fourth encoder (Enc 4) performed convolution, batch normalization, and the ReLU activation function to change from dimension 256 to 512 and ended with dimension 768 entering the decoder block stage. Furthermore, the first decoder (Dec 1) block performed convolution, batch normalization, and ReLU activation function to change from dimension 768 to 256, which ended by upsampling to enter the decoder block. The second decoder (Dec 2) block performed convolution, batch normalization, and ReLU activation function to change from dimension 384 to 128, ending with upsampling to enter the next decoder block. In this context, the third decoder (Dec 3) block performed convolution, batch normalization, and the ReLU activation function to change from dimension 192 to 64 and ended with upsampling to enter the subsequent decoder block. The fourth decoder (Dec 4) block performed convolution, batch normalization, and ReLU activation function to change from dimension 128 to 32. This was followed by convolution from dimensions 32 and 16, which ended with the output of a segmentation image or map characterized by several classes of 1.

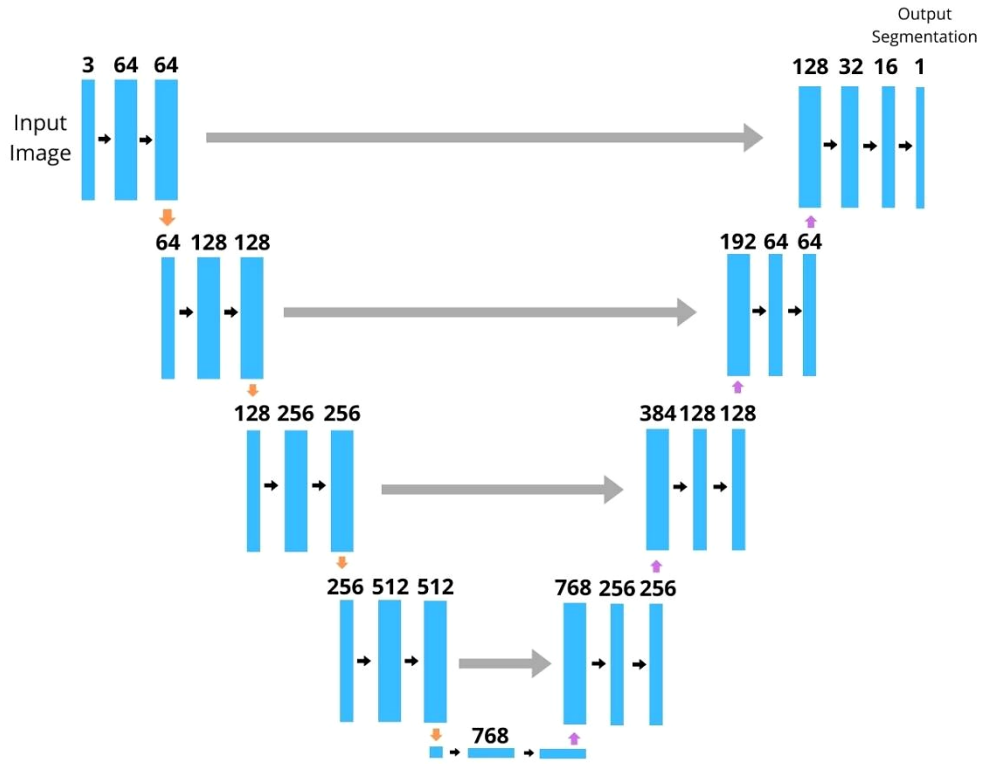


Figure 4. U-Net Architecture

2.1.3. SegNet

SegNet refers to a neural network architecture, introduced for developing image segmentation by testing pictures of roads, buildings, vehicles, and pedestrians, including floods [29, 30]. This architecture has an encoder structure for extracting features from input data. Additionally, the decoder is used to rebuild the image by carrying out convolution and max pooling. The decoder process was also conducted by upsampling to return the image to the original size [31].

In Figure 5, the architecture starts with Enc 1 and Enc 2, which are performed twice and consist of convolution, ReLu activation function, and batch normalization. Both end with Max Pooling and are connected to the fifth (Dec 5) and fourth decoders (Dec 4) to share feature representation information. Enc 3, Enc 4, and Enc 5 continue by carrying out three processes consisting of convolution, ReLu activation function, and batch normalization, which end with Max Pooling. Enc 3, 4, and 5 are connected to Dec 3, Dec 2, and Dec 1 to share feature representation information, performed after Enc 5. Dec 1, Dec 2, and Dec 3 perform upsampling, as well as three processes comprising convolution, ReLu activation function, and batch normalization. Furthermore, Dec 4 and Dec 5 perform upsampling and two processes consisting of convolution, ReLu activation function, and batch normalization. In Dec 5, the SegNet architecture ends with a Sigmoid to carry out map segmentation for several classes.

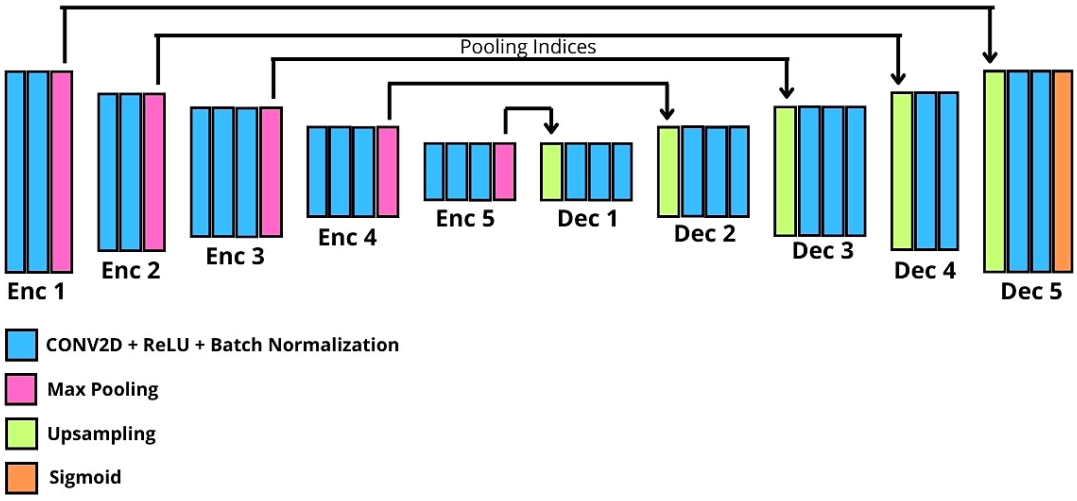


Figure 5. SegNet Architecture

2.1.4. LinkNet

LinkNet is a neural network architecture, introduced for developing image segmentation by testing cityscape images [32]. It can also be used to segment flood images and has a downsampling encoder structure for performing convolution, including max pooling [33, 34]. The decoder process was carried out by upsampling aimed to return the image to the original size [35].

Figure 6 shows that this architecture starts with the first encoder stage (Encoder Block 1), comprising convolution, batch normalization, and two layers of ReLU activation aimed at changing the dimensions from 3 (RGB) to 64. The downsample is connected to the upsample in the last decoder section (Decoder Block 4) in order to share spatial information. Meanwhile, the second encoder stage (Encoder Block 2) begins by carrying out convolution, batch normalization, and two layers of ReLU activation to change the dimensions from 64 to 128. The downsample is connected to the upsample in the third decoder section (Decoder Block 3) to share spatial information. The third encoder stage (Encoder Block 3) begins by carrying out convolution, batch normalization, and three layers of ReLU activation to change the dimensions from 128 to 256. In this perspective, the downsample is connected to the upsample in the second decoder section (Decoder Block 2) to share spatial information. The fourth encoder stage (Encoder Block 4) begins by performing convolution, batch normalization, and three layers of ReLU activation to change the dimensions from 256 to 512. The downsample is connected to the upsample in the first decoder section (Decoder Block 1) to share spatial information. The existing dimensions after the fourth encoder stage (Encoder Block 4) represent feature representations. This is followed by the first decoder (Decoder Block 1), realized by entering spatial information from the encoder and performing upsample, convolution, batch normalization, and ReLU to change dimensions from 512 to 256. The second decoder stage (Decoder Block 2) inputs spatial information from the encoder and performs upsample, transposed convolution, batch normalization, and ReLU to change dimensions from 256 to 128. The third decoder stage (Decoder Block 3) inputs spatial information from the encoder and performs upsample, transposed convolution, batch normalization, and ReLU to change dimensions from 128 to 64. Furthermore, the fourth decoder stage (Decoder Block 4) is realized by inputting spatial information from the encoder and performing upsample, transposed convolution, batch normalization, including ReLU with the same dimension output. It also consists of spatial information from the encoder and performs upsample, convolution, batch normalization, and ReLU to change dimensions from 64 to 32. The final segmentation output stage (Segmentation Output) is carried out by convolution, changing the dimensions from 32 to 1 as several related map classes.

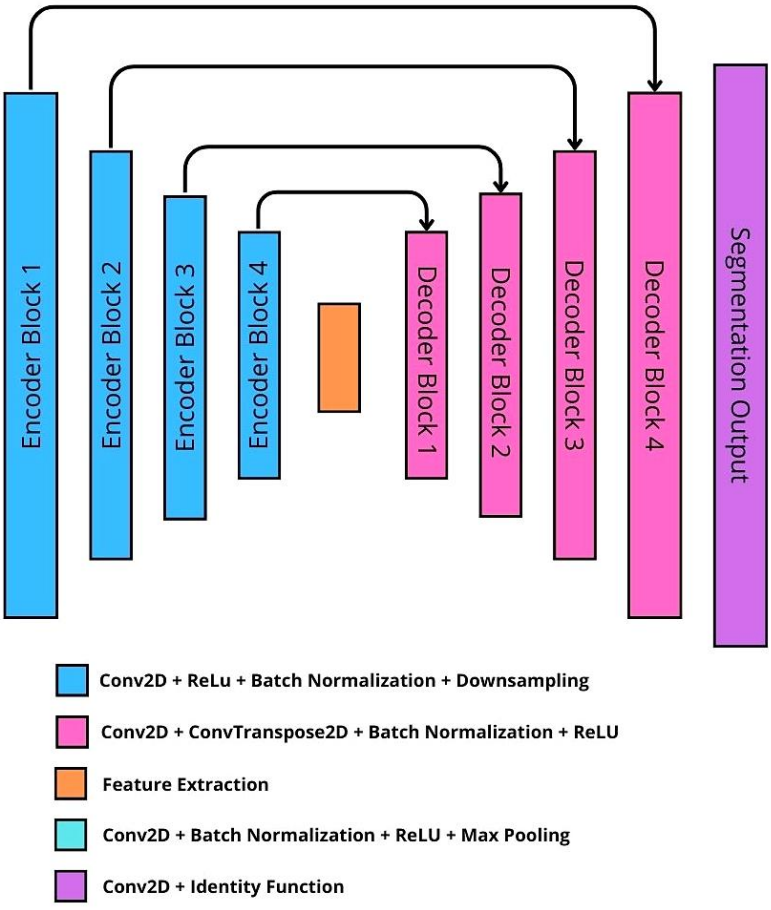


Figure 6. LinkNet Architecture

2.1.5. Backbone

Backbone is important in developing segmentation models [36], as well as implemented with various architectures, such as MobileNet, ResNet, EfficientNet, and VGG. In this context, backbone served as an encoder in segmentation architecture. A properly selected architecture improved the evaluation performance of the segmentation model.

2.1.5.1. MobileNet

MobileNet refers to a neural network architecture introduced for the development of object detection [37]. It can also serve as a backbone in image segmentation architecture due to the ability to reduce the computational load. As a result, the combination of segmentation architecture and MobileNet produces an efficient and lightweight model [38]. MobileNet has an architecture with depthwise separable convolution, capable of reducing computational load compared to regular convolution [39].

As shown in Figure 7, MobileNet started with image convolution at a 1×1 filter (Conv2D 1×1), a ReLU activation function, and batch normalization aimed to capture feature representation. Furthermore, it carried out depthwise convolution with a 3×3 filter (Depthwise Conv2D 3×3), ReLU, and batch normalization, which reduced computational parameters. The subsequent stage focused on convolution the image with a 1×1 filter, the ReLU activation function, and batch normalization. The final stage centered on performing dropout and linearity to prevent overfitting, as well as produce output in class form.

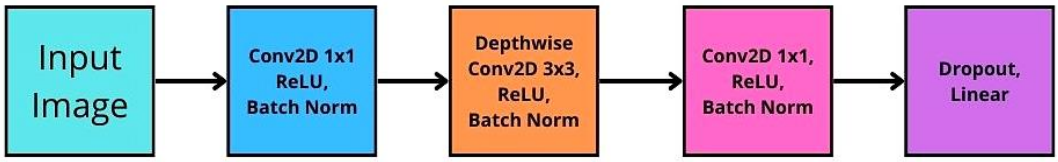


Figure 7. MobileNet Architecture

2.1.5.2. ResNet

ResNet is a neural network architecture introduced for the development of image recognition [40] and used as a backbone during segmentation because of its ability to extract more complex features. The combination of segmentation architecture and ResNet produced optimal performance. Additionally, ResNet has an architecture that uses residual learning to solve complex network problems and improve testing performance [41].

As shown in Figure 8, this architecture started with image convolution, followed by the ReLU activation function, batch normalization, and max pooling to reduce dimensions as well as capture feature representation information. Furthermore, the bottleneck block process was carried out four times (Bottleneck Blocks 1, 2, 3, and 4). Each of the bottleneck blocks contained filter image convolution measuring 1×1, 3×3, and 1×1 to reduce computation and improve model training performance. The architecture ended with a global average pool (AvgPool), including a linear link to the output.

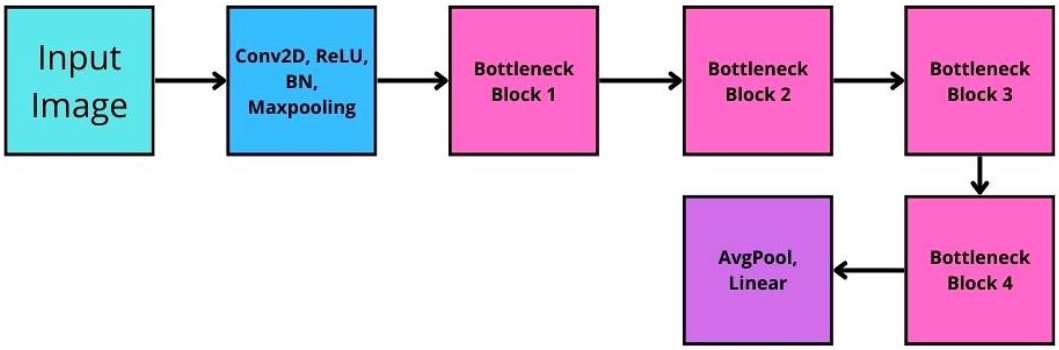


Figure 8. ResNet Architecture

2.1.5.3. EfficientNet

EfficientNet is a neural network architecture introduced for the development of image recognition [42]. It can also serve as a backbone in the segmentation process for efficiency and hierarchical representation. Therefore, the combination of segmentation and EfficientNet produced optimal performance, as well as identified features from various levels [43]. This neural network architecture adopted the concept of compound coefficient to perform feature extraction at various scale levels optimally.

Based on Figure 9, the EfficientNet architecture starts with a Conv2D Norm Activation stage, comprising image convolution, batch normalization, and ReLU activation function to capture feature representations. The next stage involves Inverted Residual Block 1, which contains depthwise convolution and a residual connection in a linear manner.

This is followed by Inverted Residual Block 2, a process that also includes depthwise convolution and a residual connection, ending linearly. The architecture then performs Inverted Residual Block 3, containing depthwise convolution and a residual connection, again ending linearly. The subsequent stage focuses on Inverted Residual Block 4, which comprises depthwise convolution and a residual connection, concluding linearly. Inverted Residual Block 5 similarly entails depthwise convolution and a residual connection, ending linearly. Additionally, Inverted Residual Block 6 is carried out with depthwise convolution and a residual connection, ending linearly. Inverted Residual Block 7 is actualized with depthwise convolution and a residual connection, also ending linearly. These processes provide flexibility in maintaining features while ensuring computational efficiency. After the Inverted Residual Blocks, the next stage of the architecture performs Conv2D Norm Activation to further capture feature representations. This is followed by global average pooling (AvgPool) and linear dropout, which connect to the optimal output while preventing overfitting.

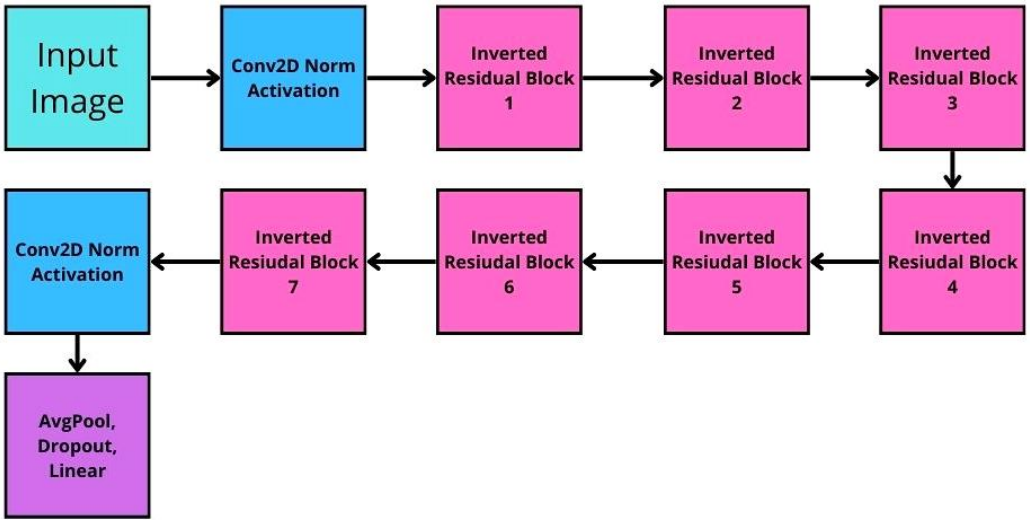


Figure 9. EfficientNet Architecture

2.1.5.4. VGG

VGG refers to a neural network architecture introduced for the development of image recognition [44]. This architecture served as a backbone in image segmentation due to the high capacity to recognize complex networks for multiscale feature extraction. Therefore, the segmentation process and VGG produced optimal segmentation performance [45]. This neural network architecture is characterized by many convolution layers and large parameters, exhibiting the ability to identify more complex features in images.

Figure 10 shows the VGG architecture started with 2 initial layers performing the same number of image convolutions with a 3×3 filter. This included the ReLU activation function, which ended by performing max pooling with a 2×2 filter. The subsequent stage had 3 layers by carrying out a similar number of image convolutions with a 3×3 filter, including the ReLU activation function, which ended with a 2×2 max pooling. The process aimed to perform image convolution, ReLu activation function, and max pooling to optimally capture feature representation. The next stage performed global average pooling, ReLU activation function, and linear dropout, intended to correlate the output, thereby preventing overfitting.

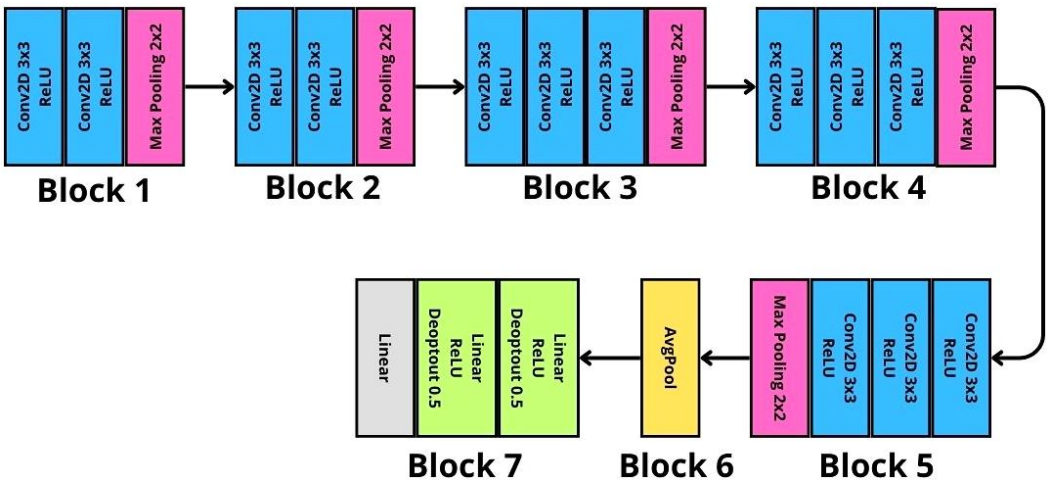


Figure 10. VGG Architecture

2.1.6. Optimizer

Optimizer is useful in deep learning development for reducing the convergence rate of training computing levels and the occurrence of overfitting, as well as increasing accuracy. It is also used to reduce the level of loss values by learning from errors through backward propagation [46]. The optimization process comprises several methods such as SGD (Stochastic Gradient Descent) [46], RMSProp [46], Adam (Adaptive Moment Estimation) [47], and AdaDelta [48].

2.1.7. Evaluation Metrics

Evaluation metrics were adopted to measure the performance of deep learning architecture, and those used in image segmentation included IoU Score, precision, F1-Score, recall, and accuracy [49]. A high evaluation metric value depicted that the results of the image segmentation evaluation produced better performance. These metrics were used to determine the level of accuracy of object identification and image balance compared to the architecture's performance. Other applications included measuring the ability of the architecture to find all objects and predict image pixels. Additionally, the precision evaluation metric was obtained using Equation 1 [49]:

$$\text{Precision} = \frac{\text{TP}}{TP+FP} \tag{1}$$

where; TP is true positive, and FP is false positive.

IoU Score evaluation metric is obtained using Equation 2 [49]:

$$\text{IoU Score} = \frac{\text{TP}}{TP+FN+FP} \tag{2}$$

where; TP is true positive, FN is false negative, and FP is false positive.

Recall evaluation metric is obtained using Equation 3 [49]:

$$\text{Recall} = \frac{\text{TP}}{TP+FN} \tag{3}$$

where; TP is true positive, and FN is false negative.

F1-Score evaluation metric is obtained using Equation 4 [49]:

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision x Recall}}{Precision+Recall} \tag{4}$$

where; Precision is Precision value (Equation 2), and Recall is Recall value (Equation 3).

Accuracy evaluation metric is obtained using Equation 5 [49]:

$$\text{Accuracy} = \frac{\text{TP+TN}}{TP+TN+FP+FN} \tag{5}$$

3. Results and Discussion

The performance evaluation of segmentation architecture was conducted to obtain the most optimal outcomes for flood inundation images. Furthermore, testing was carried out by determining IoU, accuracy, recall, precision, and F1-score [50].

In Table 2, the results of the performance evaluation showed that during U-Net segmentation, the IoU, precision, F1-score, recall, and accuracy values obtained were 0.767, 0.862, 0.866, 0.876, and 0.899, respectively. Meanwhile, for the SegNet segmentation, the IoU, precision, F1-score, recall, and accuracy values of 0.611, 0.693, 0.745, 0.879, and 0.737 were realized. LinkNet segmentation produced IoU, precision, F1-score, recall, and accuracy of 0.757, 0.864, 0.860, 0.864, and 0.895.

Table 2. Segmentation Architecture Perform

	IoU	Precision	F1-Score	Recall	Accuracy
U-Net	0.767	0.862	0.866	0.876	0.899
SegNet	0.611	0.693	0.745	0.879	0.737
LinkNet	0.757	0.864	0.860	0.864	0.895

The performance results of segmentation architecture in Figure 11 had the most optimal value in U-Net. Compared to others, U-Net produced the highest IoU, precision, F1-score, recall, and accuracy of 0.767, 0.862, 0.866, 0.876, and

0.899, respectively. The performance results showed that segmentation achieved high image balance and object identification sensitivity similar to ground truth and the most accurate pixel prediction. These results implied that U-Net ensured a high degree of image balance and also exhibited strong sensitivity in object identification, closely conforming with the ground truth as well as providing highly accurate pixel-wise predictions. This superior performance was attributed to the encoder–decoder structure enhanced with skip connections, which effectively preserved spatial details, often lost during downsampling. The retention is particularly critical in flood mapping tasks, where accurately outlining the extent of inundation at the pixel level is essential.

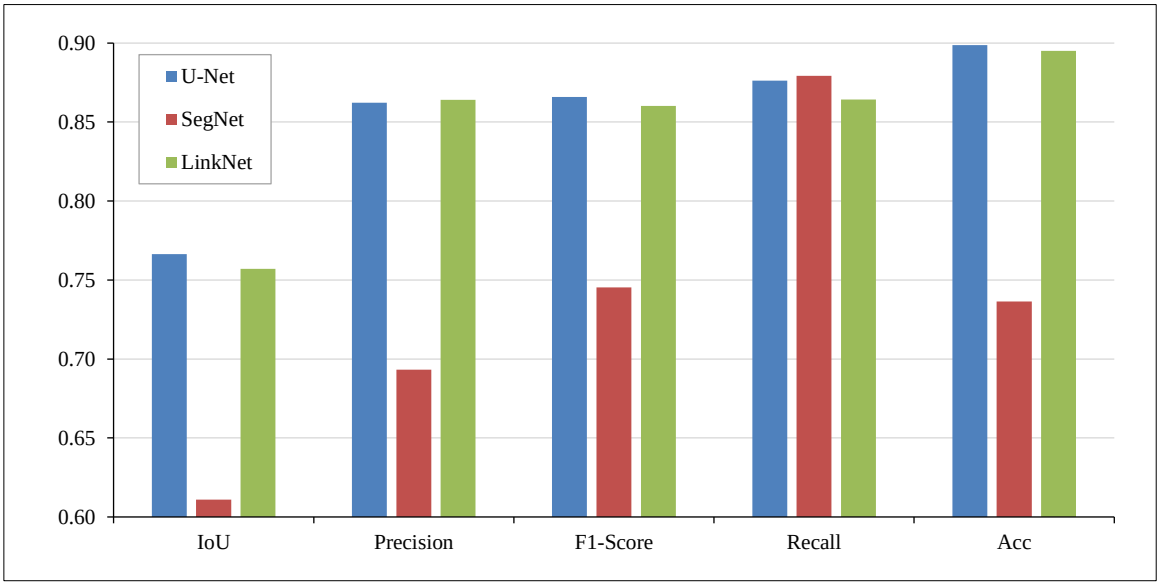


Figure 11. Segmentation Architecture Performance

Considering that the quantitative evaluation had clearly shown the superior performance of U-Net over SegNet and LinkNet, a qualitative analysis was essential to provide deeper insight into the visual quality of the segmentation results. This section focused exclusively on U-Net, which was regarded as the most optimal model. The visual output in Figure 12 was selected as a representative example to show U-Net ability to delineate flood boundaries in complex and irregular scenarios. The diagram outlined how the model captured subtle spatial variations and accurately distinguished inundated areas from surrounding terrain, reinforcing the reliability of its pixel-wise predictions in real-world conditions.

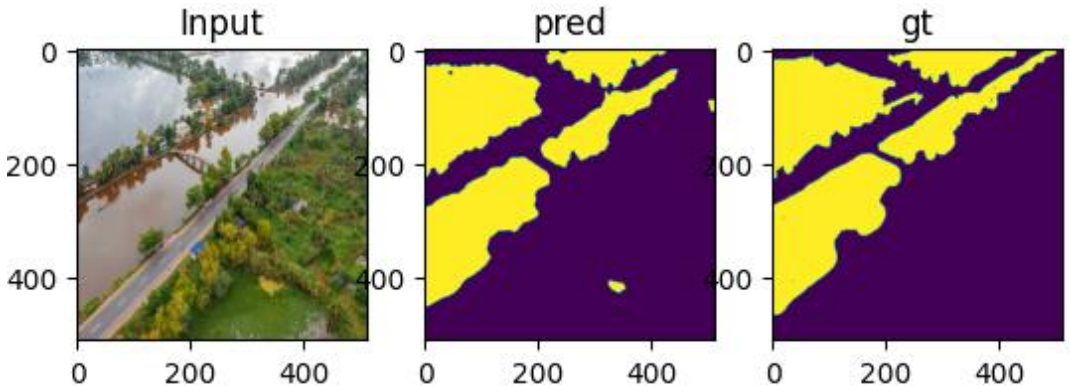


Figure 12. Sample Segmentation Result Using U-Net on a Flood-Affected Area

The segmentation output in Figure 13 was generated using U-Net architecture, which achieved the highest overall performance among all configurations tested. The U-Net model accurately delineated flood-inundated regions, with its predicted mask closely matching the ground truth as well as effectively tracing the boundaries between flooded areas and surrounding terrain. It exhibited strong spatial awareness by capturing narrow flood zones near infrastructure such as roads and vegetative areas, outlining the capacity for precise edge localization. The overall segmentation output remained coherent and visually consistent with the actual inundation pattern, while a few minor pixel-level deviations were observed. This result further validated the quantitative metrics, portraying the robustness of the model in addressing complex flood segmentation tasks.

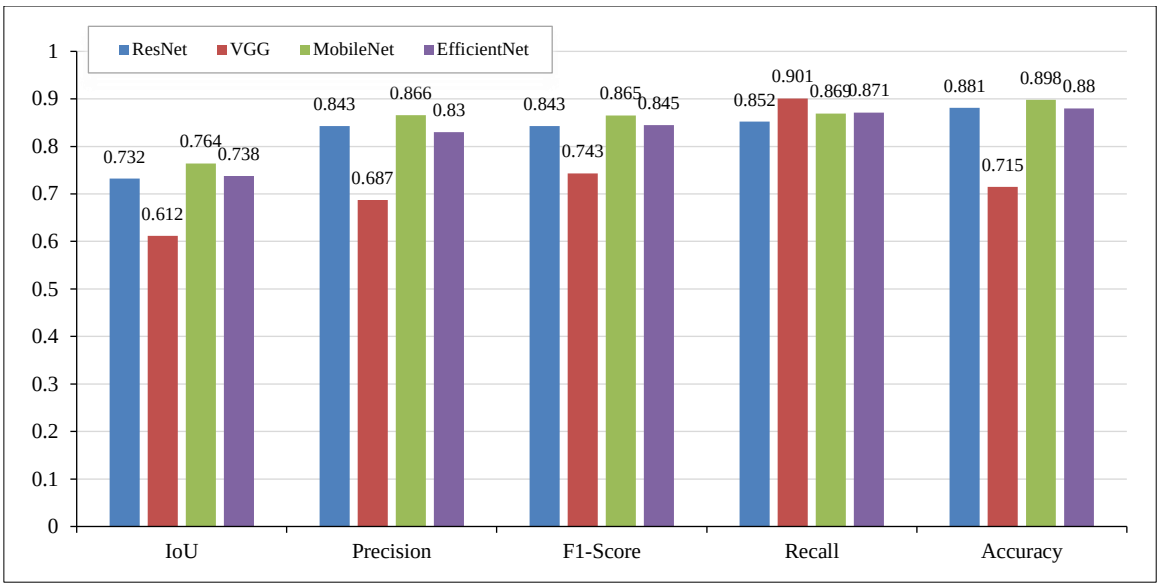


Figure 13. Backbone Architecture Performance

Performance evaluation of backbone architecture was carried out to improve the segmentation of the flood inundation image. Additionally, testing was conducted by determining IoU, accuracy value, recall, precision, and F1-score [51]. Table 3 shows that for ResNet, the IoU, precision, F1-score, recall, and accuracy values were 0.732, 0.843, 0.843, 0.852, and 0.881, respectively. VGG performance results included IoU, precision, F1-score, recall, and accuracy of 0.612, 0.687, 0.743, 0.901, and 0.715. For MobileNet, the IoU, precision, F1-score, recall, and accuracy performance results were 0.764, 0.866, 0.865, 0.869, and 0.898. Meanwhile, EfficientNet obtained IoU, precision, F1-score, recall, and accuracy performance results of 0.738, 0.830, 0.845, 0.871, and 0.880, respectively.

Table 3. Backbone Architecture Perform

	IoU	Precision	F1-Score	Recall	Accuracy
ResNet	0.732	0.843	0.843	0.852	0.881
VGG	0.612	0.687	0.743	0.901	0.715
MobileNet	0.764	0.866	0.865	0.869	0.898
EfficientNet	0.738	0.830	0.845	0.871	0.880

The results of backbone architecture in Figure 13 show the most optimal performance was obtained in MobileNet. Compared to others, the results had the highest values of IoU, precision, F1-score, recall, and accuracy of 0.764, 0.866, 0.865, 0.869, and 0.898, respectively. This superior performance was attributed to the ability of MobileNet to preserve spatial information through the structured feature extraction process. Moreover, by using depthwise convolutions, MobileNet enabled the model to capture fine-grained patterns and contextual variations essential for distinguishing flood boundaries from the surrounding landscape. These characteristics made MobileNet particularly effective for enhancing segmentation accuracy in complex visual scenes.

Optimizer performance evaluation was carried out to obtain parameter configurations that best suited the complexity of the problem in the deep learning layer, aimed to improve performance in image segmentation [52]. In addition, performance evaluation was used to determine the values of IoU, accuracy, recall, precision, and F1-score. The performance evaluation in Table 4 shows that the Adam optimizer consisted of IoU, precision, F1-score, recall, and accuracy values of 0.712, 0.807, 0.824, 0.873, and 0.843. For the SGD optimizer, the IoU, precision, F1-score, recall, and accuracy values were 0.458, 0.525, 0.620, 0.824, and 0.603. Meanwhile, the AdaDelta optimizer obtained IoU, precision, F1-score, recall, and accuracy performance results of 0.431, 0.489, 0.594, 0.803, and 0.575, respectively. For the RMSProp optimizer, IoU, precision, F1-score, recall, and accuracy performance results of 0.689, 0.786, 0.808, 0.866, and 0.829 were obtained.

Table 4. Optimizer Perform

Optimizer	IoU	Precision	F1-Score	Recall	Accuracy
Adam	0.712	0.807	0.824	0.873	0.843
SGD	0.458	0.525	0.620	0.824	0.603
AdaDelta	0.431	0.489	0.594	0.803	0.575
RMSProp	0.689	0.786	0.808	0.866	0.829

The results of optimizer performance in Figure 14 produced the most optimal performance in the Adam optimizer. Compared to others, the Adam optimizer showed the highest values of IoU, precision, F1-score, recall, and accuracy of 0.712, 0.807, 0.824, 0.873, and 0.843, respectively. The results showed that the Adam optimizer obtained high image balance and object identification sensitivity similar to the ground truth, with the most accurate pixel prediction. The superior performance was attributed to the adaptive learning rate strategy, which was dynamically adjusted for each parameter during training. Adam facilitated faster and more stable convergence, particularly in the complex and high-dimensional optimization landscapes typical of image segmentation tasks, by combining the advantages of momentum and RMS-based updates. This level of adaptability allowed the model to refine the learning process more effectively, resulting in improved generalization and a consistently accurate pixel-wise segmentation outcome.

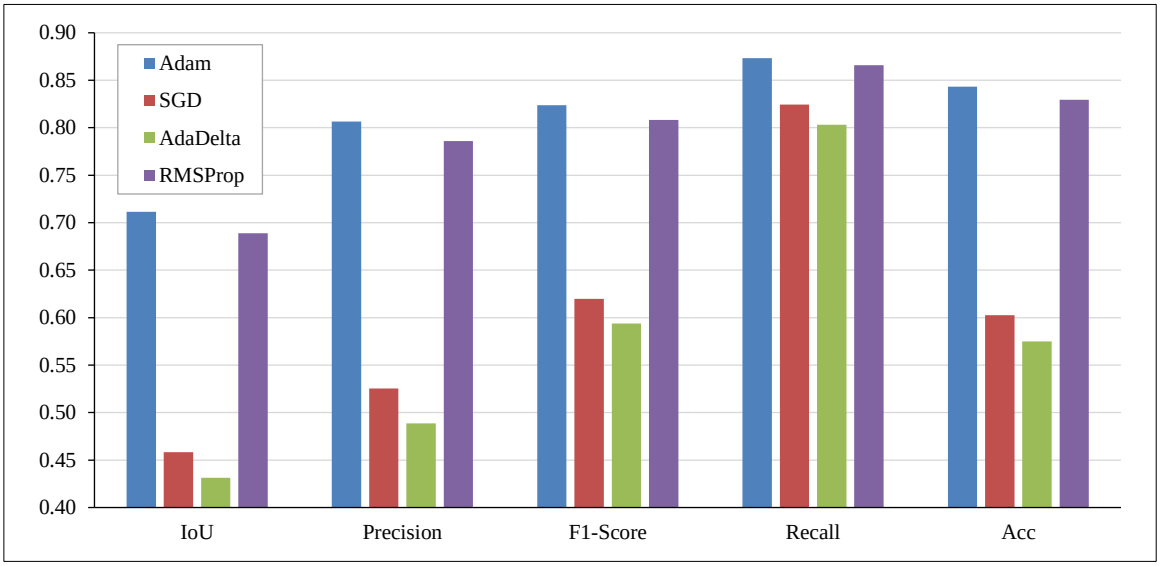


Figure 14. Optimizer Performance

4. Conclusion

In conclusion, U-Net was the most optimal in accurately identifying boundaries in flood inundation image segmentation. Based on the results, this architecture had IOU, precision, F1 score, recall, and accuracy of 0.767, 0.862, 0.866, 0.876, and 0.899, respectively. The performance of U-Net produced the most accurate image pixel prediction and effectively identified flood inundation objects. However, the selection of MobileNet as the basic architecture was the most optimal, as proven by IoU, precision, F1-score, recall, and accuracy of 0.764, 0.866, 0.865, 0.869, and 0.898, respectively. The results showed that MobileNet supported the best segmentation architecture, which successfully identified the ideal flood objects as well as predicted the optimal image pixels. The most optimal test optimizer was Adam, which produced IoU, precision, F1-score, recall, and accuracy of 0.712, 0.807, 0.824, 0.873, and 0.843. The results showed the Adam optimizer successfully identified flood inundation objects, predicted pixels, and had the best level of image balance. Its use was proven to be the most optimal hyperparameter configuration in facilitating model training. This comprehensive architecture, which combined U-Net segmentation, MobileNet, and the Adam optimizer, formed a significant foundation for addressing segmentation challenges in floods.

This research was constrained by the specificity of the datasets used and did not properly generalize across highly diverse geographical terrains or varying flood scenarios. The following factors—seasonal variation, image noise, and regional vegetation patterns—were not comprehensively assessed. Future research should enhance generalizability through domain adaptation techniques or transfer learning. Moreover, developing real-time segmentation capabilities, integrating radar or multispectral satellite imagery, and testing under extreme weather conditions could significantly support flood early warning systems and disaster response frameworks.

5. Declarations

5.1. Author Contributions

Conceptualization, A.A.S. and L.M.L.; methodology, A.A.S.; software, A.A.S.; validation, A.A.S.; formal analysis, A.A.S.; investigation, A.A.S.; resources, A.A.S. and L.M.L.; data curation, A.A.S.; writing—original draft preparation, A.A.S., L.M.L., and E.S.; writing—review and editing, E.S., M.S., F.R., and T.A.R.; visualization, E.S. and M.S. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this research are available in the article.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Acknowledgements

The authors are grateful to Faizal Karim, Sharma Khrish, and Barman, Niyar R. for providing the data used in this research.

5.5. Conflicts of Interest

The authors declare no conflict of interest.

6. References

[1] Nkwunonwo, U. C., Whitworth, M., & Baily, B. (2020). A review of the current status of flood modelling for urban flood risk management in the developing countries. Scientific African, 7. doi:10.1016/j.sciaf.2020.e00269.

[2] Qin, Y. (2020). Urban flooding mitigation techniques: A systematic review and future studies. Water (Switzerland), 12(12), 3579. doi:10.3390/w12123579.

[3] Kourtis, I. M., Tsihrintzis, V. A., & Baltas, E. (2020). A robust approach for comparing conventional and sustainable flood mitigation measures in urban basins. Journal of Environmental Management, 269. doi:10.1016/j.jenvman.2020.110822.

[4] Gessang, O. M., & Lasminto, U. (2020). The flood prediction model using Artificial Neural Network (ANN) and weather Application Programming Interface (API) as an alternative effort to flood mitigation in the Jenelata Sub-watershed. IOP Conference Series: Materials Science and Engineering, 930(1), 12080. doi:10.1088/1757-899X/930/1/012080.

[5] Nemni, E., Bullock, J., Belabbes, S., & Bromley, L. (2020). Fully convolutional neural network for rapid flood segmentation in synthetic aperture radar imagery. Remote Sensing, 12(16), 2532. doi:10.3390/RS12162532.

[6] Liu, X., Song, L., Liu, S., & Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. Sustainability (Switzerland), 13(3), 1–29. doi:10.3390/su13031224.

[7] Munawar, H. S., Hammad, A. W. A., & Waller, S. T. (2021). A review on flood management technologies related to image processing and machine learning. Automation in Construction, 132. doi:10.1016/j.autcon.2021.103916.

[8] Mo, Y., Wu, Y., Yang, X., Liu, F., & Liao, Y. (2022). Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing, 493, 626–646. doi:10.1016/j.neucom.2022.01.005.

[9] Hao, S., Zhou, Y., & Guo, Y. (2020). A Brief Survey on Semantic Segmentation with Deep Learning. Neurocomputing, 406, 302–321. doi:10.1016/j.neucom.2019.11.118.

[10] Asgari Taghanaki, S., Abhishek, K., Cohen, J. P., Cohen-Adad, J., & Hamarneh, G. (2021). Deep semantic segmentation of natural and medical images: a review. Artificial Intelligence Review, 54(1), 137–178. doi:10.1007/s10462-020-09854-1.

[11] Kar, M. K., Nath, M. K., & Neog, D. R. (2021). A Review on Progress in Semantic Image Segmentation and Its Application to Medical Images. SN Computer Science, 2(5), 397. doi:10.1007/s42979-021-00784-5.

[12] Muhadi, N. A., Abdullah, A. F., Bejo, S. K., Mahadi, M. R., & Mijic, A. (2021). Deep learning semantic segmentation for water level estimation using surveillance camera. Applied Sciences (Switzerland), 11(20), 9691. doi:10.3390/app11209691.

[13] Banan, A., Nasiri, A., & Taheri-Garavand, A. (2020). Deep learning-based appearance features extraction for automated carp species identification. Aquacultural Engineering, 89. doi:10.1016/j.aquaeng.2020.102053.

[14] Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., & Rueckert, D. (2020). Deep Learning for Cardiac Image Segmentation: A Review. Frontiers in Cardiovascular Medicine, 7. doi:10.3389/fcvm.2020.00025.

[15] Saood, A., & Hatem, I. (2021). COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet. BMC Medical Imaging, 21(1), 19. doi:10.1186/s12880-020-00529-5.

[16] Akyel, C., & Arıcı, N. (2022). LinkNet-B7: Noise Removal and Lesion Segmentation in Images of Skin Cancer. Mathematics, 10(5), 736. doi:10.3390/math10050736.

[17] Sivagami, S., Chitra, P., Kailash, G. S. R., & Muralidharan, S. R. (2020). UNet Architecture Based Dental Panoramic Image Segmentation. 2020 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2020, 187–191. doi:10.1109/WiSPNET48689.2020.9198370.

[18] Gao, Y., Wang, G., Wang, G., Li, T., Zhang, S., Li, S., Zhang, Y., & Zhang, T. (2022). Two-dimensional phase unwrapping method using a refined D-LinkNet-based unscented Kalman filter. Optics and Lasers in Engineering, 152. doi:10.1016/j.optlaseng.2022.106948.

[19] Li, X., Xu, F., Liu, F., Lyu, X., Tong, Y., Xu, Z., & Zhou, J. (2023). A Synergistical Attention Model for Semantic Segmentation of Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing, 61. doi:10.1109/TGRS.2023.3243954.

[20] Quang, N. H., Nguyen, M. N., Hung, N. M., Lee, H., & Kim, G. (2025). AI-based flood mapping from high-resolution ASNARO-2 images: case study of a severe event in the Center of Vietnam. Natural Hazards, 121(15), 17647–17675. doi:10.1007/s11069-025-07485-9.

[21] Lv, S., Meng, L., Edwing, D., Xue, S., Geng, X., & Yan, X. H. (2022). High-Performance Segmentation for Flood Mapping of HISEA-1 SAR Remote Sensing Images. Remote Sensing, 14(21), 5504. doi:10.3390/rs14215504.

[22] Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., & Martel, A. L. (2021). Loss odyssey in medical image segmentation. Medical Image Analysis, 71. doi:10.1016/j.media.2021.102035.

[23] Bangaru, S. S., Wang, C., Zhou, X., & Hassan, M. (2022). Scanning electron microscopy (SEM) image segmentation for microstructure analysis of concrete using U-net convolutional neural network. Automation in Construction, 144, 104602. doi:10.1016/j.autcon.2022.104602.

[24] Karim, F., Sharma, K., & Barman, N. R., (2022). Flood Area Segmentation. Kaggle, Mountain View, United States. Available online: https://www.kaggle.com/datasets/faizalkarim/flood-area-segmentation (accessed on November 2025)

[25] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham, Switzerland. doi:10.1007/978-3-319-24574-4_28.

[26] Safavi, F., Chowdhury, T., & Rahnemoonfar, M. (2021). Comparative Study Between Real-Time and Non-Real-Time Segmentation Models on Flooding Events. 2021 IEEE International Conference on Big Data (Big Data), 4199–4207. doi:10.1109/bigdata52589.2021.9671314.

[27] Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. IEEE Access, 9, 82031–82057. doi:10.1109/ACCESS.2021.3086020.

[28] Han, L., Liang, H., Chen, H., Zhang, W., & Ge, Y. (2022). Convective Precipitation Nowcasting Using U-Net Model. IEEE Transactions on Geoscience and Remote Sensing, 60. doi:10.1109/TGRS.2021.3100847.

[29] Dong, Z., Wang, G., Amankwah, S. O. Y., Wei, X., Hu, Y., & Feng, A. (2021). Monitoring the summer flooding in the Poyang Lake area of China in 2020 based on Sentinel-1 data and multiple convolutional neural networks. International Journal of Applied Earth Observation and Geoinformation, 102. doi:10.1016/j.jag.2021.102400.

[30] Şener, A., Doğan, G., & Ergen, B. (2023). A novel convolutional neural network model with hybrid attentional atrous convolution module for detecting the areas affected by the flood. Earth Science Informatics, 17(1), 193–209. doi:10.1007/s12145-023-01155-9.

[31] Yan, Z., Su, Y., Sun, H., Yu, H., Ma, W., Chi, H., Cao, H., & Chang, Q. (2022). SegNet-based left ventricular MRI segmentation for the diagnosis of cardiac hypertrophy and myocardial infarction. Computer Methods and Programs in Biomedicine, 227. doi:10.1016/j.cmpb.2022.107197.

[32] Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. 2017 IEEE Visual Communications and Image Processing (VCIP), 8305148. doi:10.1109/vcip.2017.8305148.

[33] Yokoya, N., Yamanoi, K., He, W., Baier, G., Adriano, B., Miura, H., & Oishi, S. (2022). Breaking Limits of Remote Sensing by Deep Learning from Simulated Data for Flood and Debris-Flow Mapping. IEEE Transactions on Geoscience and Remote Sensing, 60, 1–15. doi:10.1109/tgrs.2020.3035469.

[34] Tuyen, D. N., Tuan, T. M., Son, L. H., Ngan, T. T., Giang, N. L., Thong, P. H., Van Hieu, V., Gerogiannis, V. C., Tzimos, D., & Kanavos, A. (2021). A novel approach combining particle swarm optimization and deep learning for flash flood detection from satellite images. Mathematics, 9(22). doi:10.3390/math9222846.

[35] Sahithya, G., Harshitha, K., Priya, B. V. H., & Sirisha, B. (2023, March). Comparison of Backbones for Semantic Segmentation of X-Ray Microtomography Images using Convolutional Deep Neural Architecture. 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), 15-17 March, 2023, New Delhi, India.

[36] Liu, Y., Zhang, Z., Liu, X., Wang, L., & Xia, X. (2021). Efficient image segmentation based on deep learning for mineral image classification. Advanced Powder Technology, 32(10), 3885–3903. doi:10.1016/j.apt.2021.08.038.

[37] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv Preprint, arXiv:1704.04861. doi:10.48550/arXiv.1704.04861.

[38] Sugimoto, Y., & Aono, M. (2021). Semantic Segmentation based on Extended MobileNet with FPN. 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 1–6. doi:10.1109/icaicta53211.2021.9640256.

[39] Khatri, U., & Kwon, G. R. (2024). Diagnosis of Alzheimer's disease via optimized lightweight convolution-attention and structural MRI. Computers in Biology and Medicine, 171. doi:10.1016/j.compbiomed.2024.108116.

[40] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. doi:10.1109/cvpr.2016.90.

[41] Feng, T., Liu, J., Fang, X., Wang, J., & Zhou, L. (2020). A double-branch surface detection system for armatures in vibration motors with miniature volume based on resnet-101 and FPN. Sensors (Switzerland), 20(8), 2360. doi:10.3390/s20082360.

[42] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning. PMLR, 6105-6114.

[43] Gao, F., Sa, J., Wang, Z., & Zhao, Z. (2021). Cassava Disease Detection Method Based on EfficientNet. ICSAI 2021 - 7th International Conference on Systems and Informatics, 1–6. doi:10.1109/ICSAI53574.2021.9664101.

[44] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv Preprint, arXiv:1409.1556. doi:10.48550/arXiv.1409.1556.

[45] Sultana, F., Sufian, A., & Dutta, P. (2020). Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. Knowledge-Based Systems, 201–202. doi:10.1016/j.knosys.2020.106062.

[46] Tabrizchi, H., Parvizpour, S., & Razmara, J. (2023). An Improved VGG Model for Skin Cancer Detection. Neural Processing Letters, 55(4), 3715–3732. doi:10.1007/s11063-022-10927-1.

[47] Hassan, E., Shams, M. Y., Hikal, N. A., & Elmougy, S. (2023). The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimedia Tools and Applications, 82(11), 16591–16633. doi:10.1007/s11042-022-13820-0.

[48] Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv Preprint, arXiv:1412.6980. doi:10.48550/arXiv.1412.6980.

[49] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv Preprint, arXiv:1212.5701. doi:10.48550/arXiv.1212.5701.

[50] Wang, X., Hu, Z., Shi, S., Hou, M., Xu, L., & Zhang, X. (2023). A deep learning method for optimizing semantic segmentation accuracy of remote sensing images based on improved UNet. Scientific Reports, 13(1), 7600. doi:10.1038/s41598-023-34379-2.

[51] Wang, Z., Wang, E., & Zhu, Y. (2020). Image segmentation evaluation: a survey of methods. Artificial Intelligence Review, 53(8), 5637–5674. doi:10.1007/s10462-020-09830-9.

[52] Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415, 295–316. doi:10.1016/j.neucom.2020.07.061.