# MoTUNet: A MobileNetV2-Transformer U-Net for Water Body Segmentation

Kimsay Pov[1], Tara Kit[1], Tae-Kyung Kim[2], Youngsun Han[3*]
[1]Master's student, Dept. of AI Convergence, Pukyong National University
[2]Assistant Professor, Dept. of Management Information Systems, Chungbuk National University
[3]Professor, Dept. of AI Convergence, Pukyong National University

# MoTUNet: 수계 분할을 위한 MobileNetV2-Transformer U-Net

뽀워 김사이[1], 킷다라[1], 김태경[2], 한영선[3*]
[1]국립부경대학교 인공지능융합학과 석사과정, [2]충북대학교 경영정보학과 조교수, [3]국립부경대학교 인공지능융합학과 교수

**Abstract** Efficient real-time water body segmentation is crucial for applications such as flood detection, but balancing accuracy and inference efficiency remains challenging. In this paper, we propose MoTUNet (MobileNetV2-Transformer U-Net), designed to optimize both accuracy and inference speed for water body segmentation. Its performance is evaluated against several popular segmentation models such as U-Net, DeepLabV3+, PSPNet, PAN, and LinkNet. All models use MobileNetV2 as an encoder to reduce computational complexity while preserving feature extraction, and the Kaggle RIWA dataset is used for training and evaluation. The key metrics include Intersection over Union (IoU), precision, recall, F1-score, frames per second (FPS), and the average inference latency. Our results show that U-Net and DeepLabV3+ achieve the highest accuracy, while PSPNet is the most efficient in terms of FPS. MoTUNet provides an optimal balance by being 97.20% and 64.49% faster than U-Net and DeepLabV3+ at a 512×512 input size, and 81.71% and 58.18% faster at a 256×256 input size, while maintaining competitive segmentation accuracy.

**Key Words :** Water body segmentation, Flood monitoring, Deep learning, Convolutional neural network, Transformer-based decoder.

**요 약** 수자원 감시나 홍수 탐지와 같은 응용 분야에서 실시간 수계 분할은 필수적이지만, 정확도와 추론 효율성의 균형을 맞추는 것은 여전히 도전적인 과제로 남아있다. 본 논문에서 우리는 수계 분할의 정확도와 속도를 모두 최적화하기 위한 MoTUNet, 즉 MobileNetV2-Tranformer U-Net,을 제안하고 U-Net, DeepLabV3+, PSPNet, PAN, LinkNet과 같은 다양한 딥러닝 모델과 비교하여 성능을 평가하였다. 모든 모델은 특징 추출 능력을 유지하면서 계산 복잡도를 줄이기 위해 MobileNetV2를 인코더로 사용하며, Kaggle RIWA 데이터셋을 훈련 및 평가에 활용하였다. 주요 평가 지표로는 교집합 비율(IoU), 정밀도, 재현율, F1-점수, 초당 프레임 수(FPS), 평균 추론 지연 시간을 포함한다. 실험 결과에 따르면 U-Net과 DeepLabV3+가 가장 높은 정확도를 달성했으며, PSPNet은 FPS 측면에서 가장 효율적인 것으로 나타났다. MoTUNet은 512×512 입력 크기에서 U-Net과 DeepLabV3+보다 각각 97.20%, 64.49% 더 빠르고, 256×256 입력 크기에서는 81.71%, 58.18% 더 빠르면서도 높은 분할 정확도를 유지하며 최적의 균형을 제공한다.

**주제어 :** 수계 분할, 홍수 모니터링, 딥러닝, 합성곱 신경망, 트랜스포머 기반 디코더

*이 논문은 부경대학교 자율창의학술연구비(2023년)에 의하여 연구되었음.
*교신저자 : 한영선(youngsun@pknu.ac.kr)
접수일: 2025년 03월 06일 수정일: 2025년 03월 24일 심사완료일: 2025년 04월 08일

## 1. INTRODUCTION

Deep Learning (DL) [1] has revolutionized computer vision by enabling models to learn complex patterns from data, leading to significant advancements in image classification, object detection, and segmentation. DL-based segmentation models have demonstrated remarkable progress, particularly in applications requiring precise boundary detection and real-time inference.

Convolutional neural networks (CNNs) [2] serve as the backbone for many semantic segmentation models [3,4], allowing hierarchical feature extraction to improve spatial and contextual understanding. The choice of CNN encoder varies based on the specific application, as different architectures offer trade-offs between computational efficiency and representational power. One efficient backbone is MobileNetV2 [5], which reduces computational complexity while maintaining strong feature extraction capabilities.

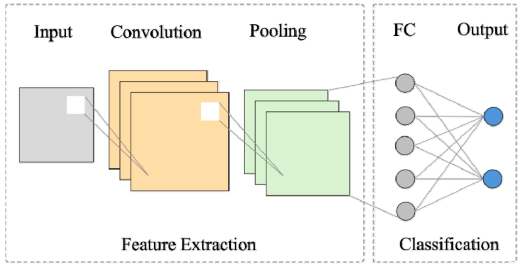
Popular segmentation models such as U-Net [6,7] and DeepLabV3+ [8,9] achieve high segmentation accuracy by capturing intricate details of input images. However, their computational cost makes them less suitable for real-time applications. In contrast, Pyramid Scene Parsing Network (PSPNet) [10,11], Pyramid Attention Network (PAN) [12], and LinkNet [13] offer faster inference but often compromise segmentation accuracy.

To address this challenge, we propose MoTUNet, a U-Net variant that integrates MobileNetV2 as the encoder and a Transformer-based decoder [14] to balance accuracy and computational efficiency. To ensure a fair comparison, we configured all baseline models using the same encoder while preserving their conventional decoder.

The main contribution of this study is to evaluate and demonstrate MoTUNet's performance and robustness in water segmentation tasks, which are crucial for real-time environmental monitoring and disaster detection.

## 2. Background

This section presents the required knowledge to understand the paper, including an overview of CNNs and the segmentation model.




[Fig. 1] Architecture of convolutional neural networks, adopted from [2].

### 2.1 Convolutional Neural Network

CNNs [2] are unique neural networks for grid-like data, such as images. CNNs excel in image classification, object detection, and semantic segmentation by learning spatial hierarchies of features through convolutional layers.

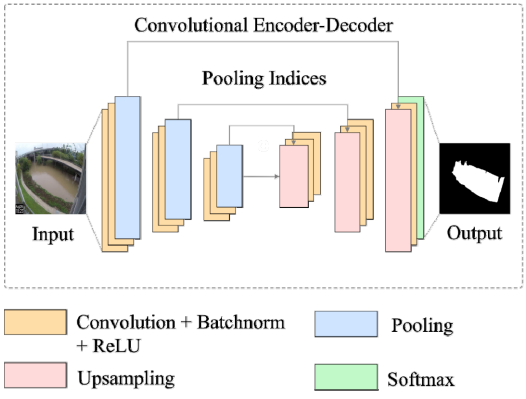
As shown in Figure 1, a CNN consists of key components. Convolutional layers apply filters to input images, extracting features like edges and textures, allowing the network to learn spatial hierarchies. A non-linear activation function, such as the Rectified Linear Unit (ReLU), follows each convolution to introduce non-linearity and enhance pattern recognition. Pooling layers, such as max or average pooling, and downsample feature maps reduce computational complexity and provide translational invariance.

Fully connected layers map extracted features to outputs, such as classification labels or segmentation masks. The output layer in segmentation tasks generates pixel-wise predictions, producing a segmentation mask where each pixel is assigned a class label.

### 2.2 Semantic Segmentation

Semantic segmentation [3] is a fundamental

task in computer vision that aims to assign a class label to each pixel in an image, enabling a more detailed understanding of scene structure compared to image classification, which assigns a single label to the entire image. By learning spatial and contextual relationships, segmentation models generate pixel-wise prediction maps that help distinguish different objects or regions within an image. This capability is instrumental in medical imaging, autonomous driving, environmental monitoring, and flood detection.




[Fig. 2] Overview of the segmentation model architecture, adopted from [4].

Modern segmentation models are primarily built on CNNs and employ an encoder-decoder architecture to extract hierarchical features and reconstruct fine-grained segmentation masks. The encoder is responsible for feature extraction through convolution and pooling layers, often utilizing pre-trained backbones such as ResNet or MobileNet to enhance feature representation. The decoder then reconstructs the segmentation mask by progressively upsampling feature maps, employing techniques like bilinear upsampling, global attention, or transformer-based decoders to refine the segmentation boundaries. Figure 2 illustrates the overall structure of a typical segmentation model, highlighting its encoder-decoder framework.

## 3. Related Works

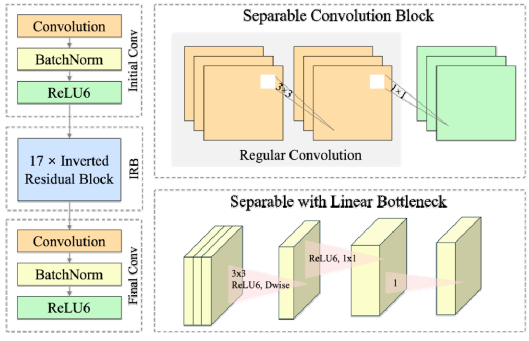
This section reviews widely used deep learning models for semantic segmentation, focusing on their architecture, strengths, and limitations in terms of accuracy and inference speed. All baseline models in this study adopt MobileNetV2 as their encoder for computational efficiency, while preserving their conventional decoders.

### 3.1 MobileNetV2

MobileNetV2 [5], is a lightweight architecture that extends the original MobileNet by introducing Inverted Residual Blocks (IRBs) with a linear bottleneck. Each IRB expands the features to a higher-dimensional space before projecting them back to a lower-dimensional representation without non-linear activation, effectively preserving essential spatial information while minimizing parameters and computational cost.

As illustrated in Figure 3, MobileNetV2 is composed of an initial stem convolutional layer, followed by a series of IRBs, and concludes with a final convolutional layer.

The lightweight and efficient design of MobileNetV2 has been widely applied in various computer vision tasks, such as image classification, object detection, and particularly semantic segmentation, where it serves as an effective encoder.




[Fig. 3] Architecture of MobileNetV2 model, adopted from [5].

### 3.2 U-Net

U-Net, introduced by Ronneberger et al. [6], is a CNN-based segmentation model specifically designed for biomedical tasks. It adopts a symmetric encoder-decoder structure, commonly known as a "U-shape" architecture. The encoder (contracting path) captures high-level contextual features through repeated convolution and downsampling operations. Meanwhile, the decoder (expanding path) restores spatial resolution by upsampling and combines it with corresponding encoder features via skip connections, allowing precise localization. This architecture enables dense pixel-wise prediction for segmentation tasks.

The key advantage of U-Net lies in its ability to achieve high segmentation accuracy with very few annotated images, supported by strong data augmentation techniques such as elastic deformation. Furthermore, the model can process images of arbitrary sizes through an overlap-tile strategy, making it efficient for segmenting large biomedical images.

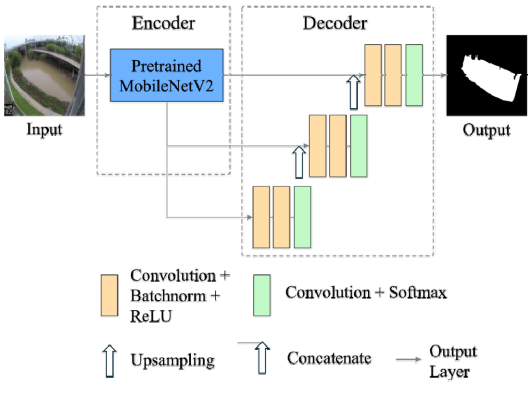
As illustrated in Figure 4, Bag et al. [7] conducted a study utilizing the U-Net architecture with a MobileNetV2 encoder to address the limitations of the original U-Net in terms of computational cost. While the original U-Net provides high segmentation accuracy, its relatively large number of parameters results in high computational complexity, making it less suitable for resource-constrained environments.

### 3.3 DeepLabV3+

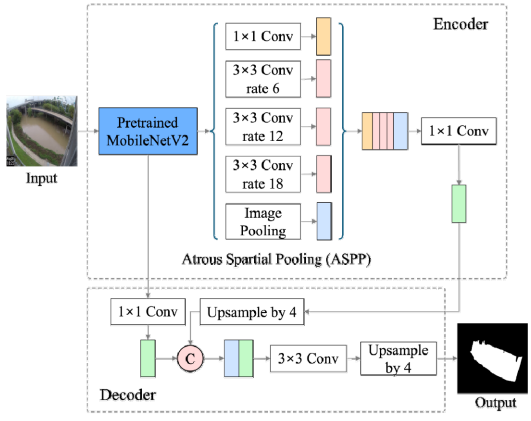
DeepLabV3+, introduced by Chen et al. [8], is an advanced semantic segmentation model designed to address two main challenges, including capturing multi-scale contextual information and preserving object boundary details. Building on DeepLabV3, which introduced Atrous Spatial Pyramid Pooling (ASPP) to extract multi-scale features using parallel atrous convolutions, DeepLabV3+ enhances this approach by adding a decoder module. This decoder progressively upsamples encoder features and then fuses them with low-level features, improving the segmentation accuracy, especially along object boundaries.

To reduce computational cost while maintaining high accuracy, DeepLabV3+ adopts depthwise separable convolutions within both ASPP and decoder modules, making it highly suitable for applications requiring efficient inference.

Figure 5 illustrates the architecture of the MobileNetV2-based DeepLabV3+ model, adapted from recent work by Zhao et al. [9], which uses MobileNetV2 as the backbone to develop a real-time semantic visual-inertial SLAM system




[Fig. 4] Architecture of MobileNetV2-based U-Net model, adopted from [7].



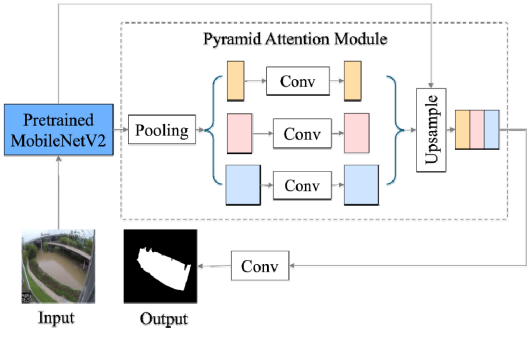

[Fig. 5] Architecture of MobileNetV2-based DeepLabV3+ model, adopted from [9].

for a dynamic environment, leveraging the pixel-wise segmentation capability of DeepLabV3+ to identify and exclude dynamic objects, improving localization robustness in challenging scenes.

### 3.4 PSPNet

PSPNet, introduced by Zhao et al. [10], is a semantic segmentation model designed to improve multi-scale feature representation through the use of a Pyramid Pooling Module (PPM), which is a key component that aggregates contextual information from multiple spatial scales by applying pooling operations with different receptive field sizes. These pooled features are then upsampled and concatenated with the original feature map, allowing the capturing of both fine-grained local details and global contextual information.

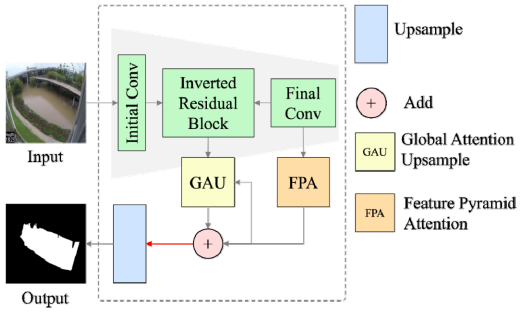
Figure 6 illustrates the architecture of MobileNetV2-based PSPNet, adapted from the recent study by Ji et al. [11], which introduced a clover dry matter prediction framework leveraging a lightweight semantic segmentation model for efficient feature extraction in agricultural analysis.




[Fig. 6] Architecture of MobileNetV2-based PSPNet model, adopted from [11].

### 3.5 PAN

Pyramid Attention Network (PAN), introduced by Li et al. [12], is a semantic segmentation model designed to enhance pixel-level prediction by utilizing global contextual information.




[Fig. 7] Architecture of MobileNetV2-based PAN model, adopted from [12].

The architecture of PAN consists of two key modules, including Feature Pyramid Pooling (FPA) and Global Attention Upsample (GAU). The FPA module strengthens multi-scale feature representation through a spatial pyramid attention structure, effectively increasing the receptive field while preserving fine-grained details. Meanwhile, the GAU module utilizes global context information from high-level features to guide low-level features during upsampling, ensuring accurate localization without adding computational complexity. Additionally, its lightweight decoder design ensures faster inference compared to complex multi-stage decoders.

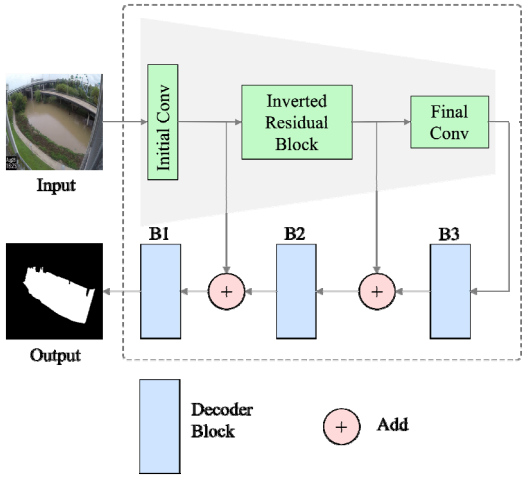
Figure 7 illustrates the architecture of the MobileNetV2-based PAN model used in this study, where MobileNetV2 serves as an efficient encoder.

### 3.6 LinkNet

LinkNet, introduced by Chaurasia et al. [13], is a semantic segmentation model designed for efficient pixel-level classification while minimizing computational cost. It adopts an encoder-decoder architecture where ResNet18 is used for lightweight yet practical feature extraction. The key characteristic of LinkNet lies in its bypass connections, directly linking the output of each encoder block to its corresponding decoder block. This design preserves spatial information lost during downsampling, ensuring accurate segmentation without relying on complex

upsampling techniques or additional parameters.

Figure 8 illustrates the architecture of the MobileNetV2-based LinkNet model used in this study, where MobileNetV2 serves as an efficient encoder.

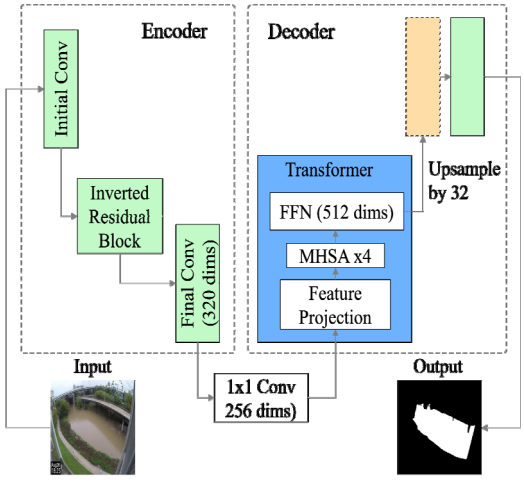



[Fig. 8] Architecture of MobileNetV2-based LinkNet model, adopted from [13].

## 4. PROPOSED METHOD

Figure 9 illustrates the overall architecture of the proposed MoTUNet, a U-Net variant that integrates a lightweight MobileNetV2 encoder with a Transformer-based decoder to balance segmentation accuracy and computational efficiency. The model is specifically designed to address the challenge of real-time water segmentation, where precise boundary detection and fast inference are essential for environmental monitoring and disaster detection systems.

MoTUNet adopts the encoder-decoder architecture, where the encoder is responsible for extracting hierarchical features from the input image. In this study, MobileNetV2 serves as the encoder due to its computational efficiency and strong feature extraction capability, achieved through the use of depthwise separable convolutions and inverted residual blocks. The encoder produces a 320-channel feature map, which is further reduced to 256 channels using a 1x1 convolution bottleneck layer to minimize computational cost before passing the features to the decoder.




[Fig. 9] Architecture of MoTUNet model.

The decoder of MoTUNet is designed based on the Transformer architecture, originally introduced by Vaswani et al. [14], which replaces recurrent operations with a self-attention mechanism, enabling parallel computation and capturing long-range dependencies. The core component of the Transformer is the Multi-Head Self-Attention (MHSA) mechanism, which calculates the attention score between query (Q), key (K), and value (V) matrices using a scaled dot-product function defined in equation (1).

$$Eq(Q,K,V) = \left\{ \frac{QK^T}{\sqrt{d_{(k)}}} \right\} \quad (1)$$

where d denotes the dimension of the key vectors. MHSA enables the model to attend to different parts of the input feature map simultaneously by splitting the input into multiple attention heads, thereby enhancing its ability to capture complex spatial relationships.

Since the Transformer lacks inherent positional awareness, positional encodings are added to the

input embeddings to retain spatial information. These positional encodings are generated using sinusoidal functions, allowing the model to distinguish the relative and absolute positions of tokens within the sequence. This mechanism enables the Transformer decoder to model both local and global contexts effectively.

In the MoTUNet decoder, the extracted feature maps from the encoder are projected into a sequence format suitable for Transformer processing. The sequence passes through two Transformer decoder blocks, where each block consists of an MHSA layer with four attention heads and a Feed-Forward Network (FFN) with 512 hidden dimensions. These blocks refine the spatial relationships of the feature representation and enhance contextual awareness, particularly around object boundaries. After processing, the output sequence is reshaped back to the spatial domain and passed through a final 1x1 convolutional layer to generate the segmentation logits. Finally, the decoder output is upsampled by a factor of 2 to restore the original input resolution, producing the final segmentation mask.

The proposed MoTUNet offers several advantages over conventional segmentation models. The integration of MobileNetV2 ensures efficient feature extraction with a significantly reduced number of parameters, making the model suitable for real-time applications on resource-constrained devices. Meanwhile, the Transformer-based decoder enhances the model's ability to capture global contextual information and refine boundary details, which are essential for accurate water segmentation in complex natural scenes. Overall, MoTUNet achieves a desirable trade-off between accuracy and inference speed, making it a promising solution for practical deployment in real-time environmental monitoring and flood detection systems.

## 5. Performance Evaluation

In this section, we present the experimental setup and evaluation result.

〈Table 1〉 Number of train, validation, and test samples.

| No | Dataset | Train | Validation | Test |
|---|---|---|---|---|
| 1 | Kaggle RIWA | 1142 | 166 | 323 |

〈Table 2〉 Number of parameters and size of each segmentation model.

| No | Model | Parameters (M) | Size (MB) |
|---|---|---|---|
| 1 | U-Net | 6.6 | 26.516 |
| 2 | DeepLabV3+ | 4.4 | 17.514 |
| 3 | PSPNet | 2.3 | 9.053 |
| 4 | PAN | 2.4 | 9.666 |
| 5 | LinkNet | 4.3 | 17.279 |
| 6 | MoTUNet | 3.0 | 12.057 |

### 5.1 Dataset

In this study, we utilized an established benchmark dataset called RIWA [15] from Kaggle for a comprehensive experiment and analysis of baseline DL models and our custom model. This dataset is publicly available on Kaggle and contains water-related images for segmentation tasks. Table 1 presents the number of training, validation, and testing samples based on its original split.

### 5.2 Training Configuration

Two training configurations were used in our experiments to evaluate the model's performance and learning behavior. In Configuration 1, the model was trained using the Adam optimizer with a fixed learning rate of 1e-5, a batch size of 32, and 50 training epochs. No learning rate scheduler was applied in this setting.

In Configuration 2, the Adam optimizer was also used, but with an initial learning rate of

1e-3 and a weight decay of 1e-4. The batch size and number of epochs were kept the same at 32 and 50, respectively. Additionally, a ReduceLROnPlateau scheduler was employed with a reduction factor of 0.5, a patience of 5 epochs, and a minimum learning rate of 1e-6.

Both configurations were trained on an NVIDIA GeForce RTX 4090 with 24GB of RAM. The second configuration was introduced to address the limitations observed in Configuration 1, improving model's learning performance and prediction accuracy.

### 5.3 Model Configuration

All the baseline models and MoTUNet are configured to use MobileNetV2 as an encoder to ensure computational efficiency. The original decoder of each baseline model is preserved, while MoTUNet adopts a transformer-based decoder for improved performance.

The number of parameters and model size illustrated in Table 2 are obtained from our experimental implementation using the PyTorch Lightning framework. These values were automatically computed during the model summary, reflecting the practical configuration used in our experiments.

### 5.4 Evaluation Criteria

The evaluation is conducted using segmentation metrics, including IoU, Precision, Recall, and F1-score, along with inference metrics such as FPS and average latency per frame.

IoU measures the overlap between the predicted and ground truth as defined in equation (2).

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

Precision is the ratio of correctly predicted positive pixels to the total predicted positives, as defined in equation (3).

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Recall is the ratio of correctly predicted positive pixels to the total of actual positives defined in equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

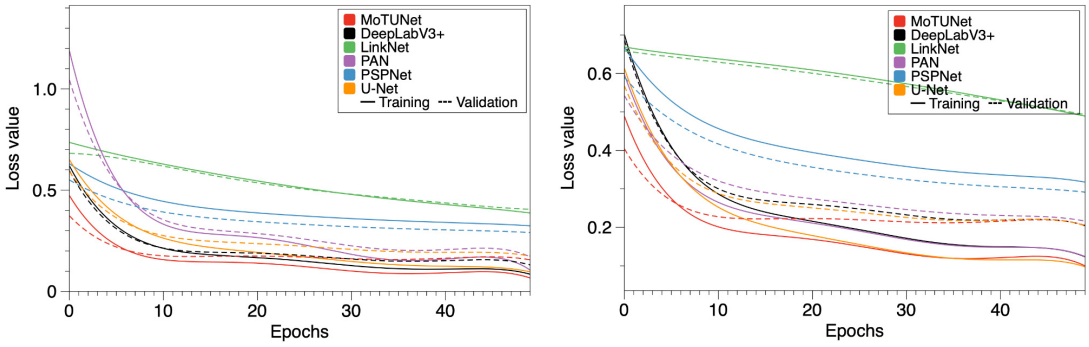
F1-score is the harmonic mean of Precision and Recall as defined in equation (5).

$$F1 - score = \frac{2 \times Precision \times Recall}{\mathrm{Precision} + Recall} \tag{5}$$

### 5.5 Performance Results

The performance of the segmentation models was evaluated based on accuracy and inference efficiency for two different input sizes and training configurations. Figure 10 and Table 3 illustrate the loss curves and performance results under Configuration 1, which uses a fixed learning rate and reflects the raw performance of each model. Meanwhile, Figure 11 and Table 4 present the results of Configuration 2, where a learning rate scheduler was applied to enable models to learn and converge optimally while reducing overfitting.

Based on the results, MoTUNet demonstrates comparable segmentation accuracy to other popular models, particularly U-Net, DeepLabV3+, PAN, and LinkNet, while achieving faster inference speed. Although PSPNet remains the fastest model in terms of inference, MoTUNet surpasses PSPNet in segmentation accuracy, making it a more balanced and effective model for real-time applications where both accuracy and efficiency are essential. Table 5 illustrates the average inference speed improvement and reduction of MoTUNet under both configurations against other segmentation models.

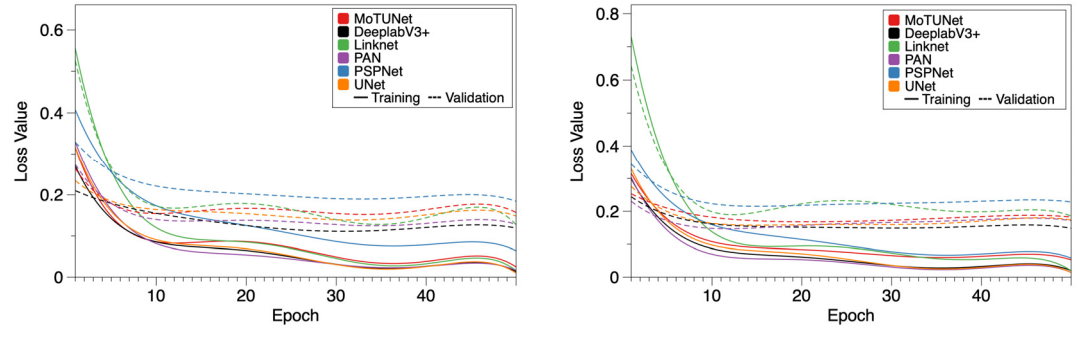



(a) 512×512 images (b) 256×256 images

[Fig. 10] Training and validation loss curves under Configuration 1.

〈Table 3〉 Performance comparison of segmentation models under Configuration 1.

| No | Model | Input Size | IoU (%) | Precision (%) | Recall (%) | F1 (%) | Avg. Latency (ms) | FPS |
|---|---|---|---|---|---|---|---|---|
| 1 | U-Net | 256×256 | 89.59 | 95.33 | 93.82 | 94.57 | 16.79 | 59.55 |
| | | 512×512 | 92.27 | 96.10 | 95.93 | 96.01 | 64.40 | 15.52 |
| 2 | DeepLabV3+ | 256×256 | 88.87 | 93.90 | 94.55 | 94.22 | 14.55 | 68.68 |
| | | 512×512 | 92.06 | 96.06 | 95.73 | 95.90 | 53.83 | 18.57 |
| 3 | PSPNet | 256×256 | 84.47 | 89.07 | 94.25 | 91.59 | 8.10 | 123.3 |
| | | 512×512 | 83.93 | 88.78 | 93.91 | 91.27 | 28.29 | 35.33 |
| 4 | PAN | 256×256 | 89.08 | 93.28 | 95.05 | 94.16 | 11.76 | 84.9 |
| | | 512×512 | 89.52 | 93.49 | 95.55 | 94.50 | 42.31 | 23.63 |
| 5 | LinkNet | 256×256 | 76.10 | 80.19 | 94.02 | 86.56 | 12.70 | 78.6 |
| | | 512×512 | 83.48 | 86.80 | 95.66 | 91.02 | 46.33 | 21.58 |
| 6 | MoTUNet | 256×256 | 88.61 | 93.99 | 93.91 | 93.95 | 9.24 | 108.2 |
| | | 512×512 | 90.91 | 94.73 | 95.78 | 95.25 | 32.75 | 30.53 |




(a) 512×512 images (b) 256×256 images

[Fig. 11] Training and validation loss curves under Configuration 2.

〈Table 4〉 Performance comparison of segmentation models under Configuration 2.

| No | Model | Input Size | IoU (%) | Precision (%) | Recall (%) | F1 (%) | Avg. Latency (ms) | FPS |
|---|---|---|---|---|---|---|---|---|
| 1 | U-Net | 256×256 | 92.27 | 96.43 | 96.35 | 96.39 | 16.72 | 59.80 |
| | | 512×512 | 93.50 | 96.71 | 96.59 | 96.65 | 63.70 | 15.69 |
| 2 | DeepLabV3+ | 256×256 | 93.10 | 96.46 | 96.30 | 96.38 | 14.62 | 68.36 |
| | | 512×512 | 93.87 | 97.31 | 96.59 | 96.86 | 53.03 | 18.85 |
| 3 | PSPNet | 256×256 | 90.27 | 94.25 | 95.64 | 94.94 | 7.93 | 123.03 |
| | | 512×512 | 90.82 | 95.45 | 94.95 | 95.20 | 27.68 | 36.11 |
| 4 | PAN | 256×256 | 93.27 | 96.40 | 96.51 | 96.45 | 13.27 | 75.38 |
| | | 512×512 | 93.83 | 97.10 | 96.56 | 96.83 | 44.00 | 22.72 |
| 5 | LinkNet | 256×256 | 92.05 | 95.62 | 96.27 | 95.94 | 14.94 | 66.93 |
| | | 512×512 | 93.61 | 96.35 | 97.06 | 96.70 | 45.80 | 21.83 |
| 6 | MoTUNet | 256×256 | 91.83 | 95.73 | 95.62 | 95.67 | 9.20 | 108.66 |
| | | 512×512 | 93.16 | 96.77 | 96.18 | 96.47 | 32.21 | 31.04 |

〈Table 5〉 Average inference speed improvement (%) of MoTUNet compared to other segmentation models.

| No | Model | Input Size | U-Net | DeepLabV3+ | PSPNet | PAN | LinkNet |
|---|---|---|---|---|---|---|---|
| 1 | U-Net | 256×256 | 81.71 | 58.18 | -15.04 | 35.75 | 49.91 |
| | | 512×512 | 97.20 | 64.49 | -16.06 | 18.44 | 41.82 |

Figure 12 illustrates the original and predicted masks generated by each segmentation model on 10 sample images from the test dataset, demonstrating the segmentation capability of MoTUNet in comparison with other models.

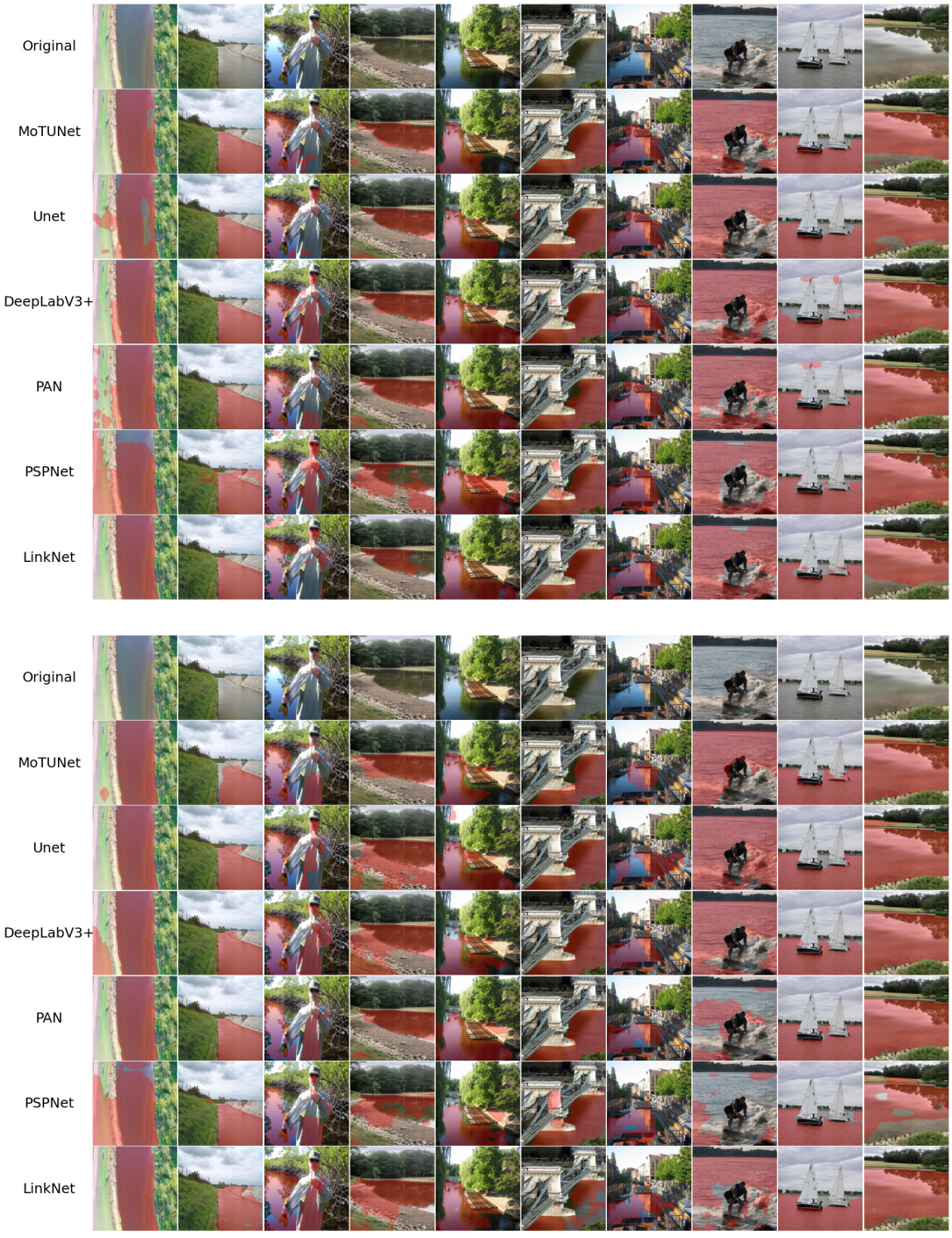
## 6. CONCLUSION

The study analyzes multiple segmentation models, including U-Net, DeepLabV3+, PSPNet, PAN, LinkNet, and the proposed Transformer-based MoTUNet with MobileNetV2 as the backbone. The results demonstrate that MoTUNet effectively balances segmentation accuracy and inference speed, making it well-suited for practical applications. While its segmentation performance is comparable to U-Net, DeepLabV3+, PAN, and LinkNet, it significantly reduces the inference latency.

The findings reinforce the importance of efficient segmentation architectures in water segmentation and flood detection. Integrating a Transformer-based decoder improves the model's ability to capture long-range dependencies, enhancing segmentation consistency.

While MoTUNet illustrates comparable segmentation accuracy to UNet, DeepLabV3+, PAN, and LinkNet, further studies and experiments on sophisticated augmentation will improve the performance of MoTUNet while maintaining its fast inference speed.

Future work can focus on evaluating MoTUNet's adaptability to different segmentation tasks beyond water detection. Further optimization through improved pre- and post-processing techniques or architectural refinements could enhance accuracy while maintaining inference efficiency. Testing the model on diverse datasets and real-world scenarios will also be essential for broader applicability.




[Fig. 12] Original and predicted masks of 512×512 and 256×256 images, respectively.

## REFERENCES


[1] LeCun, Yann & Bengio, Y. & Hinton, Geoffrey. (2015). Deep Learning. Nature. 521. 436-44. 10.1038/nature14539.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, 2015. [Online]. Available: https://arxiv.org/abs/1511.08458.

[3] J. Long E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431-3440.

[4] Emek Soylu, B., Guzel, M. S., Bostanci, G. E., Ekinci, F., Asuroglu, T., & Acici, K. (2023). Deep-Learning-Based Approaches for Semantic Segmentation of Natural Scene Images: A Review. Electronics, 12(12), https://doi.org/10.3390/electronics12122730.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4510-4520.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[7] B. Bag, "UNET MobileNetV2: Medical Image Segmentation Using Deep Neural Network (DNN)," JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES, Vol.18, Jan. 2023. doi:10.26782/jmcms.2023.01.00002.

[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," arXiv preprint arXiv:1802.02611, 2018. Available: https://arxiv.org/abs/1802.02611.

[9] X. Zhao, C. Wang, and M. Ang, "Real-Time Visual-Inertial Localization Using Semantic Segmentation Towards Dynamic Environments," IEEE Access, vol. PP, pp. 1-1, 2020. doi: 10.1109/ACCESS.2020.3018557.

[10] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid Scene Parsing Network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2881-2890.

[11] Y. Ji, J. Fang, and Y. Zhao, "Clover Dry Matter Predictor Based on Semantic Segmentation Network and Random Forest," Applied Sciences, vol. 13, p. 11742, 2023. doi: 10.3390/app132111742.

[12] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, arXiv:1805.10.180.

[13] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in Proc. IEEE Vis. Commun.Image Process. (VCIP), Dec. 2017, pp. 1-4.

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need.arXiv preprint arXiv:1706.03762. Available at: https://arxiv.org/abs/1706.03762.

[15] X. Blanch. F. Wagner, and A. Eltner. (2022). RIWA Dataset. [Online]. Available at: https//www.kaggle.com/dsv/4289421.



**뽀워 김사이(Kimsay Pov)** [준회원]

- 2020년 2월 : Royal University of Phnom Penh 컴퓨터과학과 (이학사)
- 2023년 9월 ~ 현재 : 부경대학교 일반대학원 인공지능융합학과 석사과정

〈관심분야〉
양자 머신 러닝, 딥러닝 소프트웨어 프레임워크

**킷 다라(Tara Kit)** [준회원]

- 2020년 2월 : Royal University of Phnom Penh 컴퓨터과학과 (이학사)
- 2024년 3월 ~ 현재 : 부경대학교 일반대학 인공지능융합학과 석사과정

〈관심분야〉
양자 머신 러닝

**김 태 경(Tae-Kyung Kim)** [정회원]

- 2005년 2월 : 충북대학교 정보산업공학과 (공학박사)
- 2010년 3월 ~ 2013년 4월 : 한국생명공학연구원 박사후연구원
- 2013년 5월 ~ 2019년 8월 : 한국 SW HRD센터 센터장
- 2020년 3월 ~ 2024년 2월 : 인천재능대학교 조교수
- 2024년 3월 ~ 현재 : 충북대학교 경영정보학과 조교수

〈관심분야〉
인공지능, 빅데이터, 양자머신러닝

한 영 선(Youngsun Han) [정회원]

- 2003년 2월 : 고려대학교 전기전자전파공학부(공학사)
- 2009년 2월 : 고려대학교 일반대학원 전자컴퓨터공학과(공학박사)
- 2009년 6월 ~ 2011년 2월 : 삼성 LSI 사업부 책임연구원
- 2011년 3월 ~ 2019년 8월 : 경일대학교 전자공학과 부교수
- 2019년 9월 ~ 현재 : 국립부경대학교 컴퓨터인공지능공학부 교수

〈관심분야〉
양자컴퓨팅, 컴파일러 설계, 양자머신러닝