Neural Network for Water Body Segmentation - A quick review

Jamil Nassar and Thomas Garnier
Instructor: Dr. Danilo Comminiello
Submission Date: 23/06/2025
Exam Date: Session 1 (26/06/2025)

1 Introduction

The management of water resources is a critical issue in the context of climate change. Satellite imagery, for exemple like Sentinel-2, offers a mean to observe and analyze water bodies at large scale, and at different times. However, identifying water regions from satellite images is still a challenging task.

Semantic segmentation models, such as U-Net, have shown good results in identifying water bodies. Yet, U-Net architectures often struggle with complex patterns. To overcome these limitations, recent research has focused on improving U-Net with mechanisms like residual connections and attention modules, which can upgrade the quality of models.

1.1 Selected papers and project aim

The selected paper initially used for this project is the AER U-Net: attention-enhanced multi-scale residual U-Net structure for water body segmentation using Sentinel-2 satellite images (11). In this paper, the author tries to increase the quality of existing water body segmentation model, using U-Net architecture as a baseline, and adding residual connections and attention mechanisms in skip connections. Our goal is to replicate this results, building the latter modified U-Net architecture, in order to improve quality of the model

2 Theoretical background and key concepts

2.1 Convolutional Neural Networks (CNN)

Convolutional neural networks are architectures used in deep learning that directly work on the input data and learns from it. It is of multiple layers, including convolutional layers, pooling layers, and finally fully connected layers. The convolutional layer applies a learnable fliter, called kernel, to its input data. It enables detecting important key points in the image input. It also reduce the number of learnable parameters, and exploit spatial locality, wich is perfect for image data. Usually, after convolution layer, the model applies a batch normalization layer, which ensures stability, and a nonlinear activation function. The pooling layer helps modifying the dimension of the data. It can be max pooling or any kind of pooling which takes the input shapes and modify the resolution (Height and Weight). The fully connected layer is the one that connects the neuron to all other neurons, and is usually used at the end of the model, performing regression or classification tasks.(5)

2.2 U-Net

The U-Net model is a convolutional neural network specifically designed for image segmentation, known for its precision, robustness, and adaptability. It consists of two main parts: an encoder and a decoder.

The encoder comprises convolutional layers followed by pooling layers, progressively extracting features while reducing spatial dimensions. This process continues until the network reaches the bottleneck stage.

The decoder then works in the reverse manner, starting with upsampling layers. At each decoding stage, feature maps are concatenated with corresponding feature maps from the encoder via skip connections, preserving spatial information. These concatenated feature maps are then processed by convolutional layers to refine the segmentation output. (2; 7)

Figure 1 illustrates this architecture, clearly showing the encoder, decoder, and skip connections.

2.3 Dropout

In deep learning models, one of the challenges is to face and fight against overfiting. Dropout method is one of the current solution, proposed to solve

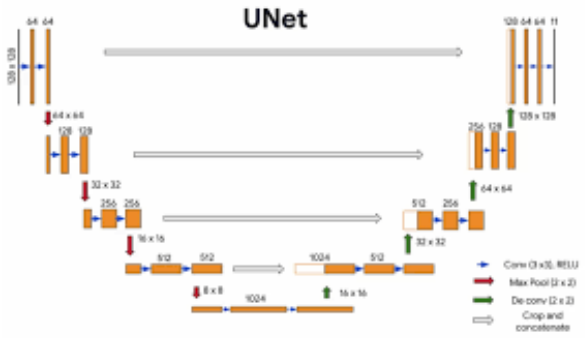

Figure 1: U-Net Architecture.(2; 7)

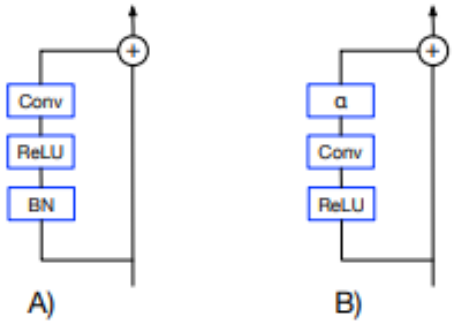

Figure 2: Residual Block (9)

this problem. It offers an effective way of mixing a huge number of distinct neural network architectures. The term "dropout" refers to dropping out units (hidden and visible) in a neural network during training. When a unit is dropped out, all of its incoming and outgoing connections are also temporarily removed from the network. The units that are dropped are chosen at random, regarding an hyperparameter p, describing the probability of keeping a neuron in the training. The study also discovered that, in comparison to training with other regularization techniques, training a network using dropout and applying this approximation averaging strategy at test time results in significantly lower generalization error on a wide range of classification problems.(8)

2.4 Residual Blocks

Residual blocks address the vanishing gradient problem. In order to create an identity mapping, the residual block adds a skip connection, also known as a shortcut connection, which enables a layer's input to be added straight to the output without passing through one or more convolutional and normalization layers. This straightforward change has become a crucial part of modern deep learning designs and facilitates the training of very deep networks. The main formula of the residual block is $r(x) = f(x) + x$ Where: • x is the input to the residual block. • f(x) is the function applied to the input x which usually consists of one or more convolutional / normalization layers. • r(x): The output of the residual block As illustrated in the picture below, we present an example of residual block implementation.

Remark: Each residual block aligns the input channels to match the output using a 1 × 1 convolution. This design facilitates better gradient flow and ensures that critical information is not lost as the network depth increases.(9)

2.5 Attention Block

The attention function is defined as a mapping from a query and a set of key-value pairs to an output, where the query, keys, values, and output are all transformed vectors of the input. The result is a weighted sum of the values, with each value's weight determined by the query's compatibility function with the relevant key. There exists two types of attention: scaled dot-product Attention and Multi-Head Attention

2.5.1 Scaled Dot-Product Attention

The input is the queries, keys and values. It performs then the dot product between the queries and the keys. After that, it applies a softmax function to the product obtained divided by $\sqrt{d_k}$ (where d_k is the dimension of the keys). Subsequently, we multiply the result of the softmax by the values V

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

2.5.2 Multi-Head Attention

When this function is applied to a batch of elements, this layer calculates the attention function independently for every element of the batch. Instead of applying a single attention function on high-dimensional queries, keys, and values (with dimension d_model), the model projects them multiple times into smaller dimensions (d_k and d_v) using learned linear transformations. These projections create multiple attention heads. These two types are illustrated in the picture below

The attention block refines skip connections by focusing on relevant spatial regions. This attention map highlights important regions, which are multiplied with the skip connection to refine the input for the decoder. (10)

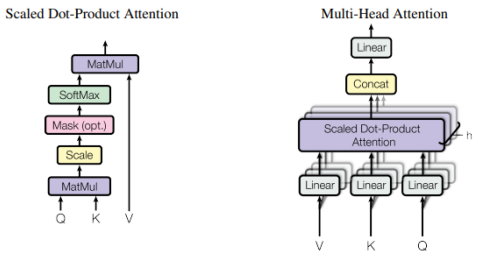

Figure 3: Multi Head Attention

2.6 Attention-enhanced multi-scale residual U-Net (AER U-Net)

The AER U-Net model builds upon the classical U-Net architecture by incorporating residual connections and attention gates within the skip connections. The residual blocks help mitigate the vanishing gradient problem and improve convergence during training. Attention gates, conversely, allow the network to selectively focus on relevant regions within the skip connections, enhancing feature refinement.

Like U-Net, the AER U-Net has an encoder-decoder structure with skip connections linking corresponding stages. However, each block in the network is enhanced with residual layers and attention gates, which are placed along the decoder path to refine the features passed from the encoder.

Figure 4 illustrates the AER U-Net architecture, highlighting these residual and attention components integrated within the standard U-Net framework.

Despite these architectural improvements, our implementation of the AER U-Net did not consistently outperform the baseline U-Net, suggesting the need for further research into more efficient or better-tuned attention mechanisms.

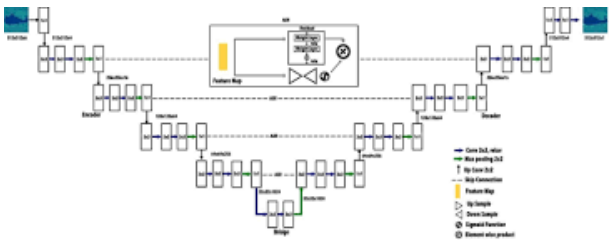

Figure 4: AER U-Net architecture showing residual connections and attention gates incorporated within the U-Net framework.

2.7 Attention Augmented Convolutional Networks

Attention-Aware Convolution is a mechanism designed to enrich convolutional feature maps with contextual information via self-attention operations. It differs from classical convolution that applies static kernels. Indeed, AA convolution refines its outputs by computing attention across spatial positions or channels.

Usually, implementing AA models means projecting the output of a CNN layer into query, key, and value representations. These representations are then used to compute an attention map, typically via scaled dot-product attention. The resulting attention-weighted features are aggregated and combined with the original convolutional output, either through addition (residual) or concatenation. (3)

2.8 Squeeze-and-Excitation Networks

Squeeze-and-Excitation (SE) Networks, introduced by Hu et al. (CVPR 2018) (4), is a model based on a channel attention mechanism that recalibrates feature responses based on their global importance. Traditional convolutional layers treat each channel equally, but SE modules learn to prioritize more informative channels dynamically.

The SE block operates in two stages: the squeeze step applies global average pooling : we obtain a channel descriptor vector. The excitation step then uses a small two-layer feedforward network to learn non-linear inter-channel dependencies and produce channel-wise attention weights. These weights are applied to the original feature maps via multiplication : it gives more weights to useful channels, and diminish less relevant ones.

In our architecture, SE modules are integrated inside each convolutional block of the baseline U-Net model.

2.9 Evaluation Metrics

Evaluation metrics are the quantitative measures that evaluate the performance of the model and the excellence of the segmentation. There are multiples types of evaluation metrices such as the accuracy, precision, IoU(intersection over union),, recall and F1 Score (11)

2.9.1 Accuracy

It is calculated by performing the fraction of the correct predictions out of the total predictions made (11)

$$Accuracy = \frac{Tp+Tn}{Tp+Tn+Fp+Fn}$$

Where:

- Tp: True Positive
- Tn: True Negative
- Fp: False Positive
- Fn: False Negative

2.9.2 Recall

It measures the proportion of the positive instances from the total count of positive instances. (How much the model is good in detecting) (11)

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

2.9.3 Precision

It calculates the ratio of the true positive prediction to the count of all positive predictions made. (How much the model is precise in detetcting)(11)

$$Precision = \frac{Tp}{Tp+Fp}$$

2.9.4 F1 score

Precision focuses on the accuracy of positive predictions, emphasizing how well the model avoids false positives. Recall focuses on the ability of the model to capture all relevant positive instances, emphasizing how well the model avoids false negatives. The F1 score, which is the harmonic mean of precision and recall, is a metric that combines both measures (11)

$$F1 = 2 * \frac{Recall*precision}{Recall+precision}$$

2.9.5 Intersection over Union (IoU)

It measures the overlap between the predicted segmentation mask and the ground truth mask.(11)

$$\text{IoU} = \frac{|\text{Prediction} \cap \text{Ground Truth}|}{|\text{Prediction} \cup \text{Ground Truth}|}$$

3 Methodology

Initially, we implemented the U-Net model as a baseline, and three models introducing residual connections and/or attention gates inside the skip connections, called AER U-Net, AE U-Net, and R U-Net. However, as detailed in the results section, the performance of these models was below expectations. To address this, we explored two additional architectures that further incorporate attention and residual connections in a different way. Specifically, we enhanced the convolutional blocks themselves by embedding Attention Augmentation (AA) and Squeeze-and-Excitation (SE) modules, experimenting with the presence of residual connections and/or attention mechanisms within the SE blocks. These modifications aim to refine feature extraction and improve model convergence.

4 Implementation details

We used the Satellite Images of Water Bodies dataset, publicly available on Kaggle (Satellite Images of Water Bodies). This dataset is made of 7,000 pairs of RGB satellite images and corresponding binary masks indicating water presence. The binary masks label water pixels as 1 and background pixels as 0, making this a binary segmentation task. Prior to training, images were normalized and masks converted to binary tensors. The dataset was split into training (80%) and validation (20%) sets.

Our baseline model is the U-Net architecture. We implemented variants of this model, trying to replicate the AER U-Net. Specifically, the AER U-Net combines multi-scale residual blocks with attention modules, while the AE U-Net and R U-Net incorporate attention or residual blocks separately. In addition, we also built models using Attention Augmented (AA) convolution blocks, with or without residual connections inside those blocks. To end up with, we also created Squeeze-and-Excitation (SE) blocks, which recalibrate channel-wise feature responses by modeling interdependencies between channels. All models were trained on the same dataset splits, so that we can directly compair their segmentation performance.

Regarding the experimental setup : Data loaders fed batches of images and corresponding masks, and standard preprocessing (normalization, resizing) was applied. We trained each model for 10 epochs, with a dropout ration of 0.3, using the Adam optimizer with a learning rate of 1e-4. We choosed the binary cross-entropy with logits loss function to train our model. Training was performed on GPU when available, otherwise on CPU. We saved both training and validation losses and accuracies at each epoch to track performance and convergence. Moreover, and complete final evaluation of the last model was done, looking at accuracy, precision, recall, F1 score and IoU score.

5 Results and analysis

Figure 5 illustrates the evolution of validation accuracy over epochs, starting with the baseline U-Net

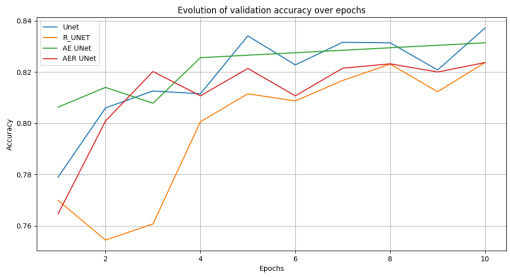

Figure 5: Evolution of validation accuracy over epochs for the AER model

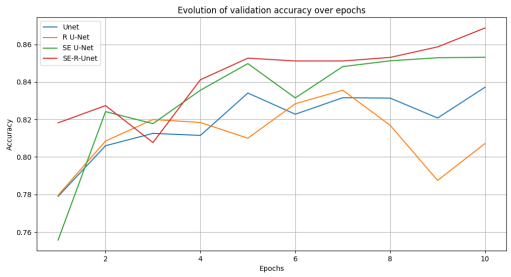

Figure 6: Evolution of validation accuracy over epochs for the SER model

Table 1: AER U-Net Replication Attempt

Model	Accuracy	Precision	Recall	F1 Score	IoU
U-Net	0.8372	0.8425	0.6434	0.7296	0.5743
R-U-Net	0.8237	0.7782	0.6764	0.7237	0.5670
AE U-Net	0.8314	0.7606	0.7387	0.7495	0.5994
AER U-Net	0.8237	0.7725	0.6855	0.7264	0.5704

Table 2: SE and Residuals Inside Convolutional Blocks

Model	Accuracy	Precision	Recall	F1 Score	IoU
U-Net	0.8372	0.8425	0.6434	0.7296	0.5743
R-U-Net	0.8072	0.7487	0.6552	0.6988	0.5371
SE U-Net	0.8532	0.8207	0.7293	0.7723	0.6290
SE-R-U-Net	0.8688	0.8593	0.7365	0.7932	0.6572

model and progressively incorporating attention and residual blocks in the R-U-Net and AE U-Net variants, culminating in the AER U-Net model. The quantitative results in Table 1 provide a detailed comparison of the performance of these different models.

We observe that the F1-score and IoU score for the AER U-Net model (0.7264 and 0.5704, respectively) are significantly below the original paper's reported values (F1 ≈ 0.946, IoU ≈ 0.947). This suggests that our replication did not fully reach the performance claimed in the literature, possibly due to limited training duration. Indeed, our models were trained for only 10 epochs compared to the 50 epochs suggested in the original article, which likely hindered full convergence.

Among the tested models, the AE U-Net shows the best balance between precision and recall, with an F1-score of 0.7495 and an IoU of 0.5994, which may indicate better generalization despite a simpler architecture.

Recall values tend to be lower across models (e.g., 0.6434 for U-Net, 0.6855 for AER U-Net), indicating a tendency to miss positive water body regions. In contrast, precision is relatively high, implying that when the model predicts water, it is usually correct, though it may under-detect some positive regions. This behavior is typical in imbalanced segmentation tasks where true positives are under-represented.

Figure 6 shows the validation accuracy progress over epochs for the SER model, while Table 2 presents a detailed comparison of models incorporating Squeeze-and-Excitation (SE) blocks and residual connections inside convolutional layers.

The SE-R-U-Net model achieves the highest validation metrics, including accuracy (0.8688), precision (0.8593), recall (0.7365), F1 score (0.7932), and IoU (0.6572). These results demonstrate that integrating residual and attention mechanisms directly inside convolutional blocks enhances segmentation performance.

The models' high precision indicates reliable positive predictions, but future work should focus on improving recall, possibly by extending training duration or using loss functions tailored for imbalanced segmentation.

Overall, these findings highlight promising directions for future research by incorporating attention and residual learning into convolutional blocks.

6 Limitations and reflections

Unfortunately, our project was limited by computational resources : we could only use 10 epochs per model, while the paper suggested 50. This prevented full convergence and impacted model performance. Additionally, the Attention Augmented (AA) convolution required too much memory. It limited its practical application even if theory explained us that it could improve performance Future work should explore longer training, alternative loss functions to better handle class imbalance, and more memory-efficient attention mechanisms to improve segmentation quality.

7 References

References

[1] Jonnala, N. S., Siraaj, S., Prastuti, Y., Chinnababu, P., Praveen Babu, B., Bansal, S., Upadhyaya, P., Prakash, K., Faruque, M. R. I., & Al-Mugren, K. S. (2023). *AER U-Net: Attention-Enhanced Multi-Scale Residual U-Net Structure for Water Body Segmentation*. Scientific Reports. https://www.nature.com/articles/s41598-025-99322-z

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. MICCAI 2015. https://arxiv.org/abs/1505.04597

[3] Bello, I., Zoph, B., Vaswani, A., Shlens, J., & Le, Q. V. (2019). *Attention Augmented Convolutional Networks*. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 3285–3294.

[4] Hu, J., Shen, L., & Sun, G. (2018). *Squeeze-and-Excitation Networks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7132–7141.

[5] N. S. Jonnala et al., « AER U-Net: attention-enhanced multi-scale residual U-Net structure for water body segmentation using Sentinel-2 satellite images », Sci. Rep., vol. 15, no 1, p. 16099, mai 2025, doi: 10.1038/s41598-025-99322-z.

[6] K. R. Reddy et R. Dhuli, « Detection of brain tumors from MR images using fuzzy thresholding and texture feature descriptor », J. Supercomput., vol. 79, no 8, p. 9288-9319, mai 2023, doi: 10.1007/s11227-022-05033-x.

[7] G. Du, X. Cao, J. Liang, X. Chen, et Y. Zhan, « Medical Image Segmentation based on U-Net: A Review », J. Imaging Sci. Technol., vol. 64, no 2, p. 020508-1-020508-12, mars 2020, doi:10.2352/J.ImagingSci.Technol.2020.64.2.020508.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, et R. Salakhutdinov, « Dropout: A Simple Way to Prevent Neural Networks from Overfitting ».

[9] S. De et S. L. Smith, « Batch Normalization Biases Residual Blocks Towards the Identity Function in Deep Networks », 10 décembre 2020, arXiv: arXiv:2002.10444. doi: 10.48550/arXiv.2002.10444.

[10] A. Vaswani et al., « Attention is All you Need ».

[11] N. S. Jonnala, N. Gupta, C. P. Vasantrao and A. K. Mishra, "BCD-Unet:A Novel Water Areas Segmentation Structure for Remote Sensing Image," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023, pp. 1320-1325, doi: 10.1109/ICICCS56967.2023.10142694.

Additional resources

- **Dataset used:** Satellite Images of Water Bodies on Kaggle.
- **Code used:** U-Net code adapted from the UNet-Water-Segmentation GitHub repository.

8 Reproducibility Instructions

To reproduce our results, please visit our GitHub repository: https://github.com/tgarnier067/Neural_Networks_for_Water_Body_Segmentation and run Notebook 13.

Make sure the dataset is properly downloaded and placed in the appropriate folders, or update the file paths in the code accordingly.

Alternatively, you can contact us (see Contact section) to request access to the preprocessed data on Google Drive. In that case, first run Notebook 2 to set up the environment, then run Notebook 13.

You will also need to select the model you want to reproduce by uncommenting the corresponding cell in the notebook.