جامعة خليفة
Khalifa University

Deep Learning-Based Semantic Segmentation of Carbonate Rock Textures in 3D X-ray Microtomography Images

Senior Research Project

Lina Taha 100063400

Faculty Supervisors:
Dr. Mohamed Soufiane Jouini*
Dr. Aymen Laadhari

Khalifa University

Date of Submission: April 22, 2026

Abstract

This report investigates deep learning methods for semantic segmentation of petrographic textures related to carbonate rock imaging. Carbonate rocks contain highly heterogeneous pore systems and texture distributions that strongly affect properties such as porosity and permeability. Manual analysis of X-ray microtomography (μCT) images is time-consuming and subjective, which has encouraged the development of automated segmentation methods.

Two semantic segmentation architectures were implemented and compared: U-Net and Attention U-Net. The study used nine grayscale petrographic texture images integrated into the Randen texture segmentation framework, where each texture corresponds to a distinct class label. A predefined mask specifies the spatial regions of these classes, and a composite image is generated by assigning each texture to its corresponding region. The mask therefore serves as the ground-truth segmentation map used to evaluate the model predictions. Training samples were generated as 32×32 labeled patches derived from the texture images integrated into the Randen framework, while evaluation was performed on a composite image with known ground-truth segmentation. Systematic experiments were conducted by varying the optimizer, the number of training epochs, and the network depth.

The results show that the best U-Net configuration achieved an accuracy of 0.9093 using a 15-layer architecture with Adam and 24 epochs, while the best Attention U-Net configuration achieved an accuracy of 0.8920 using a 15-layer architecture with SGDM and 50 epochs. Timing experiments also showed that U-Net required substantially less training time. Overall, the findings suggest that encoder–decoder deep learning models are capable of effectively segmenting petrographic textures, with U-Net offering the best trade-off between accuracy and computational efficiency in this study.

Keywords: Texture analysis, carbonate rocks, μCT imaging, semantic segmentation, deep learning, U-Net, Attention U-Net, convolutional neural networks

Contents

1 Introduction

1.1 Background and Motivation

Carbonate rocks play a key role in industrial and environmental applications, including hydrocarbon reservoir characterization, groundwater studies, and carbon storage. Their petrophysical properties are strongly controlled by pore structure and heterogeneity. In carbonate reservoirs, pore systems are highly heterogeneous and exhibit complex, multi-scale geometries, which makes their characterization challenging using conventional methods [1].

Three-dimensional X-ray microtomography (μCT) provides a non-destructive way to image rock samples at high resolution and allows internal structures to be studied in detail [2]. This makes μCT a valuable tool for understanding carbonate heterogeneity and for relating image-derived properties to physical rock behavior. However, manual segmentation of textures in these images is time-consuming, subjective, and difficult to scale. As a result, automated semantic segmentation has become an important research direction.

Deep learning methods are especially attractive for this kind of task because they can learn texture representations directly from data instead of depending only on handcrafted descriptors [3]. Encoder-decoder segmentation models are particularly useful because they combine contextual understanding with pixel-level localization [3]. In this project, U-Net and Attention U-Net were investigated for supervised semantic segmentation of texture images [3, 4].

1.2 Problem Statement

The goal of this project is to perform semantic segmentation of texture regions in two-dimensional images related to carbonate rock texture analysis. Formally, given an input image

$$I \in \mathbb{R}^{H \times W},$$

the objective is to learn a function

$$f : \mathbb{R}^{H \times W} \to \mathbb{Z}^{H \times W}$$

that predicts a segmentation map

$$S = f(I),$$

where each pixel $S(i, j)$ is assigned a discrete class label corresponding to a texture class.

In this work, the segmentation framework was built using nine petrographic texture images integrated into the Randen setup. The project then compared U-Net and Attention U-Net under multiple training configurations in order to identify the best-performing model. The dataset construction and training workflow were based on an

adapted Randen-style benchmark framework [5].

1.3 Research Objectives

The main objectives of this project are:

1. To investigate supervised deep learning approaches for multi-class texture segmentation.
2. To implement and analyze U-Net and Attention U-Net architectures for petrographic texture segmentation.
3. To construct a controlled training and evaluation framework based on nine grayscale texture images integrated into a Randen-style benchmark.
4. To study the influence of key training parameters, including optimizer type, number of epochs, and network depth.
5. To compare model performance using quantitative metrics (accuracy, precision, recall, F1-score), computational metrics (training and inference time), and qualitative segmentation results.

2 Literature Review

2.1 Traditional Texture Analysis Methods

2.1.1 Statistical Approaches

Gray-Level Co-occurrence Matrix (GLCM) is one of the classical statistical methods used for texture analysis. It captures the frequency of intensity pairs occurring at a specific spatial offset and direction [6]. From this matrix, several features can be derived, such as contrast, energy, and homogeneity. These descriptors can be useful for regular textures, but they may be less effective when the texture is highly heterogeneous or non-stationary, as is often the case in carbonate rocks.

Local Binary Patterns (LBP) describe local texture by comparing each pixel to its surrounding neighbors and encoding the result as a binary pattern [7]. LBP is simple and computationally efficient, but it mainly captures local micro-patterns and may not fully represent larger spatial texture structures.

2.1.2 Spectral Approaches

Gabor filters are widely applied in texture analysis because they capture information in both the spatial and frequency domains [8]. They are useful for extracting features across different scales and orientations, but their performance depends heavily on the selection

of filter parameters and can be limited when textures show large variations in direction and scale.

Wavelet-based methods provide multi-resolution analysis and can capture texture characteristics across different scales [9, 10]. These methods are more flexible than some classical statistical descriptors, but they still depend on manual feature design and may not adapt as effectively as learned deep representations.

2.1.3 Limitations of Traditional Methods

Although traditional texture analysis methods remain useful, they have several limitations for semantic segmentation of complex geological textures:

- They often rely on handcrafted features.
- They may not handle non-stationary textures well.
- Their ability to represent multi-scale spatial context can be limited.
- They do not naturally perform end-to-end pixel-wise segmentation.

These limitations motivate the use of deep learning models that can learn hierarchical and task-specific texture features directly from the training data.

2.2 Deep Learning for Texture Segmentation

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) learn image features through trainable convolution kernels. A convolution operation can be written as

$$O(i,j) = (I * K)(i,j) = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} I(i+m, j+n) \cdot K(m,n),$$

where I is the input image, K is the filter, and O is the output feature map [11, 12]. CNNs are especially useful for texture-related tasks because they can learn multiple levels of representation, from simple edges and local patterns to more abstract spatial structures. Nonlinear activation functions such as ReLU allow these learned representations to model more complex texture patterns, while pooling operations help aggregate contextual information across larger spatial regions.

2.2.2 U-Net

U-Net is a type of encoder–decoder model that was first proposed for biomedical image segmentation tasks [3]. In this architecture, the encoder gradually captures more abstract features by applying convolution and pooling operations, while the decoder works to

rebuild the segmentation output through upsampling. One of the key strengths of U-Net is its use of skip connections, which help preserve detailed spatial information by passing it directly from the encoder to the decoder layers.

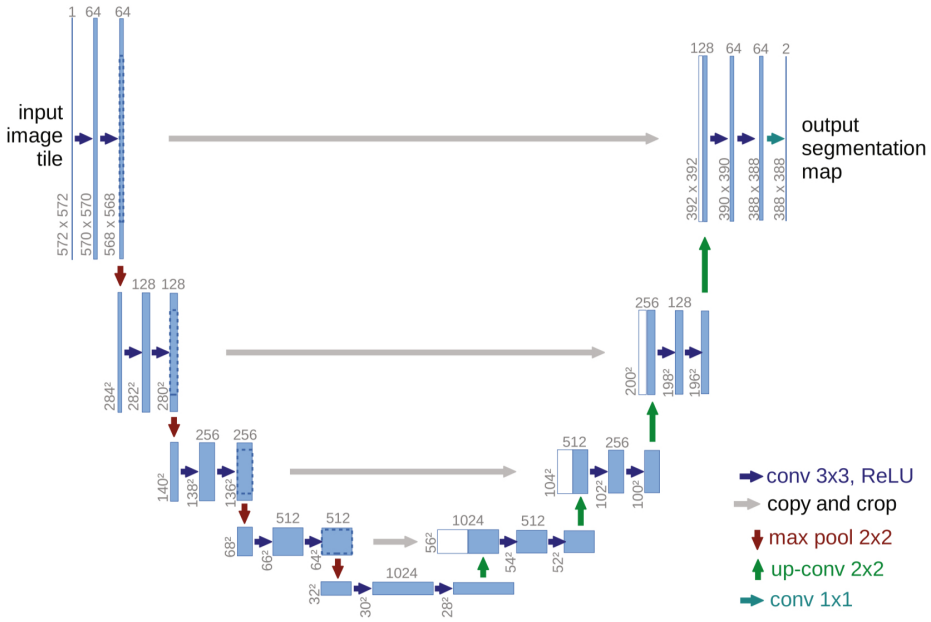


Figure 1: U-Net architecture from Ronneberger et al. [3]. The contracting path captures context through downsampling, while the expanding path reconstructs the segmentation map and skip connections preserve spatial detail.

This can be represented as

$$x^{(l)}_{\text{decoder}} = \text{concat}\left(\text{upsample}(x^{(l+1)}_{\text{decoder}}), x^{(l)}_{\text{encoder}}\right).$$

These skip connections help preserve boundaries and local detail, which is important in texture segmentation.

2.2.3 Attention U-Net

Attention U-Net extends the original U-Net framework by integrating attention gates into its skip connections [4]. Rather than passing all encoder features straight to the decoder, the network selectively focuses on the most important regions while reducing the influence of less relevant information.

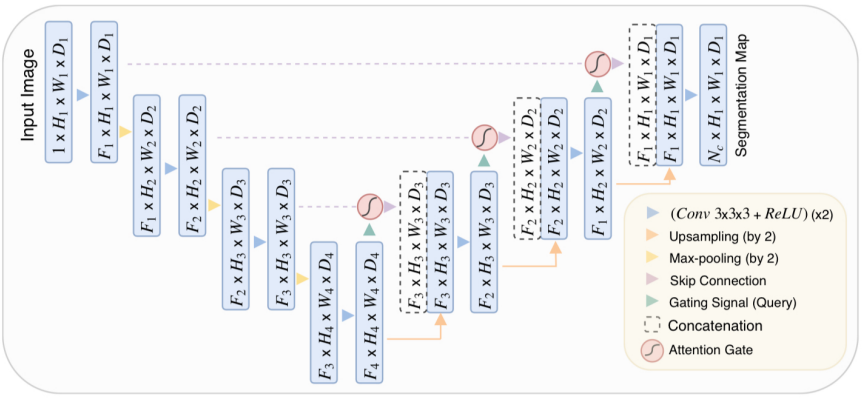


Figure 2: Attention U-Net architecture overview from Oktay et al. [4]. Attention gates are inserted into skip connections in order to filter encoder features before they are combined with decoder features.

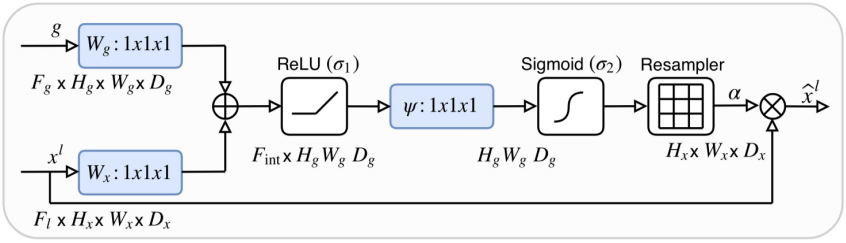


Figure 3: Attention gate mechanism from Oktay et al. [4]. The gate combines encoder features and a decoder gating signal to generate attention coefficients used to weight the transmitted features.

The attention computation can be written as

$$q_{\text{att}} = \psi^T \left(\sigma_1(W_x^T x + W_g^T g + b_g)\right) + b_\psi,$$

$$\alpha = \sigma_2(q_{\text{att}}),$$

where x is the encoder feature map, g is the gating signal, and α is the attention coefficient. The weighted feature map is then

$$\hat{x} = \alpha \odot x.$$

Therefore, the main difference between U-Net and Attention U-Net is that standard U-Net forwards encoder features directly through skip connections, whereas Attention U-Net first weights these features through attention gates so that more relevant regions are emphasized before decoding. This mechanism helps the model reduce the influence of less relevant responses and focus more strongly on informative structures in the image.

2.2.4 Loss Function and Optimization

For semantic segmentation, categorical cross-entropy is a common choice:

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_{n,k}\log(\hat{y}_{n,k}),$$

where $y_{n,k}$ is the ground-truth label and $\hat{y}_{n,k}$ is the predicted class probability [13].

Categorical cross-entropy is commonly used since it directly evaluates how different the predicted probability distribution is from the actual class distribution. In semantic segmentation tasks, every pixel is classified into one of multiple categories, with the model producing a probability for each class. This loss function applies a stronger penalty when the model is confidently incorrect, which helps improve how well the predicted probabilities are calibrated during training. It is particularly effective for multi-class classification problems and pairs naturally with the softmax activation function, making it easy to incorporate into convolutional neural network models [13].

Three optimizers were considered in this work:

- **SGDM**, which uses momentum to improve convergence [14]
- **RMSprop**, which adapts learning rates based on recent gradient magnitudes [15]
- **Adam**, which combines momentum and adaptive learning rate ideas [16]

2.2.5 Evaluation Metric

The main performance measure used in this project was pixel-wise accuracy:

$$\text{Accuracy} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}.$$

Pixel-wise accuracy was used as the primary metric for comparing the different model configurations in the main experiments. In addition, precision, recall, and F1-score were computed for the selected best configurations in the rerun stage in order to provide a more detailed evaluation of segmentation performance.

2.3 Reported Performance of Traditional Methods

2.3.1 Statistical Methods: GLCM and LBP

Previous studies show that statistical texture descriptors can provide useful segmentation and classification performance, but their results depend strongly on the type of image, the selected features, and the classifier used. For Gray-Level Co-occurrence Matrix (GLCM), Karabag et al. reported an average misclassification of 33.23% on the six Randen benchmark composites, which was substantially worse than the best U-Net configuration in the

same study [5]. In another study, GLCM-based texture features were used for crop classification from high-resolution UAV imagery, where the use of GLCM features improved classification accuracy by 13.65% compared to using grayscale images alone [17]. In addition, overall classification accuracies of up to 90.91% were achieved when GLCM features were combined with machine learning classifiers such as Random Forest, demonstrating the effectiveness of texture-based descriptors in distinguishing visually similar classes.

For Local Binary Patterns (LBP), Karabag et al. reported an average misclassification of 12.36% on the same Randen composites, which was much better than co-occurrence matrices but still worse than the best U-Net result [5]. LBP-based approaches also remain competitive in image analysis applications, particularly in face recognition tasks. Ahonen et al. demonstrated that Local Binary Pattern Histogram (LBPH) representations achieve strong performance on the FERET database by capturing local texture information through spatially enhanced histograms [18]. Subsequent studies showed that applying Linear Discriminant Analysis (LDA) to LBP histograms further improves recognition performance compared to using histogram similarity measures alone, highlighting the importance of combining LBP features with appropriate classifiers [18].

Experimental results reported in the literature also indicate that LBP-based methods can achieve high recognition rates, often exceeding 80%–90% depending on the dataset and configuration, and that performance improves when multi-scale or region-based representations are used. For example, mean recognition rates of approximately 83.1% to 86.4% were reported when varying the number of local regions in LBP-based methods on the FERET dataset [18]. These results show that LBP descriptors are effective at capturing discriminative local patterns, particularly when spatial information is incorporated.

These results explain why both GLCM and LBP remain important reference methods in texture analysis. However, they also share a fundamental limitation: both rely on handcrafted descriptors whose effectiveness depends on manually selected parameters, including neighborhood structures, spatial offsets, region sizes, and encoding strategies. As a result, their performance can degrade when texture characteristics vary across regions, scales, or imaging conditions. This limitation motivates the use of deep learning approaches, which can learn hierarchical and task-specific features directly from data.

2.3.2 Spectral Methods: Gabor and Wavelet-Based Approaches

Spectral methods are also widely used because they represent texture in terms of spatial-frequency content. Gabor filters have produced strong results in several studies. In histopathological brain tumor subtype classification, a Gabor-filter-based texture analysis approach achieved a highest reported classification accuracy of 95% when Gabor-filter energy was combined with a fractal signature feature vector [19]. In the same study, Gabor filter energy features alone achieved a total classification accuracy of 89.38%, demonstrating that Gabor-based descriptors can effectively capture discriminative texture

patterns in complex medical images [19].

In addition, Gabor filters have also been used in deep learning settings. In Gabor-Net, incorporating a Gabor-constrained first layer into a convolutional neural network produced up to 6% higher accuracy than a standard CNN on the Dogs vs Cats dataset, while also improving convergence speed during training [20]. This shows that Gabor-based representations remain useful even when integrated into modern learned models.

Wavelet-based approaches have also shown strong performance because of their multi-resolution representation of image content. In an early and influential study, wavelet packet signatures achieved perfect classification (100% accuracy) on 25 natural textures using a simple two-layer network classifier [21]. This result demonstrates the strong discriminative capability of wavelet-based representations for texture classification.

More advanced wavelet-based feature extraction techniques have shown better performance by capturing finer details within frequency channels. For instance, Yu and Kamarthi found that a cluster-based wavelet feature extraction approach improved overall classification accuracy compared to traditional wavelet methods, as it was able to derive more informative features from the distribution of wavelet coefficients [22].

However, both Gabor and wavelet-based methods still require manual feature design and parameter choices, such as filter banks, decomposition levels, and feature aggregation strategies. As a result, their performance depends heavily on parameter choices and prior domain knowledge. This is one of the main reasons deep learning methods are attractive in the present study: they can learn feature hierarchies automatically instead of depending on handcrafted frequency-domain design.

2.4 Reported Performance of U-Net and Attention U-Net

2.4.1 Reported U-Net Results

The selection of U-Net as the baseline model in this project is strongly supported by prior research. First, U-Net was originally developed for biomedical image segmentation and achieved state-of-the-art results in the ISBI 2015 cell tracking challenge, with the ability to segment a 512×512 image in under one second on a GPU [3]. Second, in a Randen-style benchmark study by Karabag et al., U-Net produced the best overall performance, recording an average misclassification rate of 8.50

Third, U-Net has consistently delivered strong performance in medical image segmentation tasks. For example, in a recent study on cerebral infarction segmentation, a standard U-Net achieved a Dice score of 0.8947, a mean IoU of 0.8798, and a pixel accuracy of 0.9963, surpassing the more complex U-Net3+ model despite its simpler architecture [23].

Taken together, these studies suggest that U-Net is a strong baseline because it preserves spatial detail through skip connections while remaining computationally efficient. This literature also supports the interpretation of the present study, where the standard

U-Net achieved the best overall performance among the tested configurations.

2.4.2 Reported Attention U-Net and Attention-Based U-Net Results

Attention U-Net was selected because prior work suggests that attention mechanisms can improve segmentation when the model must focus selectively on relevant regions. In the original study by Oktay et al., Attention U-Net improved Dice from 0.814 to 0.840 and recall from 0.806 to 0.841 on the CT-150 dataset relative to baseline U-Net [4]. On the TCIA-82 dataset, Dice improved from 0.815 to 0.821, and in the fine-tuned setting from 0.820 to 0.831 [4]. These results show that attention gates can provide measurable gains when difficult target structures need to be localized more precisely.

More recent attention-based variants of U-Net have also shown strong performance. For instance, a Residual-Attention UNet++ model achieved a Dice coefficient of 88.59

In a similar way, an attention-enhanced U-Net applied to building extraction from remote sensing images reached an overall accuracy of 97.47

These studies justify the inclusion of Attention U-Net in the present project. They show that attention-based extensions can improve segmentation quality, especially in problems involving subtle structures, scale variation, or difficult boundaries. At the same time, they also suggest a trade-off: attention mechanisms introduce additional architectural complexity and computational cost. This is consistent with the findings of the present study, where Attention U-Net was competitive but did not surpass standard U-Net.

2.5 Literature-Based Rationale for the Present Study

The reviewed literature provides a clear rationale for the design of the present study. Traditional statistical and spectral texture analysis methods remain useful and interpretable, but their performance depends heavily on handcrafted descriptors and parameter selection. This makes them less flexible for heterogeneous and multi-scale texture segmentation problems, especially when texture appearance varies across regions and boundaries.

In contrast, U-Net provides an end-to-end segmentation framework that can learn hierarchical texture features directly from data while preserving spatial detail through skip connections. Previous studies have shown that U-Net performs strongly across both benchmark texture datasets and broader image segmentation tasks, making it a suitable baseline for the present work. Attention U-Net was then selected as a meaningful extension because prior studies suggest that attention mechanisms can improve segmentation in problems involving subtle structures, complex backgrounds, or variable spatial patterns.

The literature also indicates that model performance depends not only on architecture, but also on training configuration. Factors such as optimizer choice, network depth, and number of training epochs can substantially affect segmentation results. For this reason,

the present study was designed to compare U-Net and Attention U-Net under multiple controlled training settings rather than evaluating only a single fixed configuration for each model.

Comparisons with previous studies should be interpreted carefully because datasets, segmentation targets, and evaluation metrics differ across applications. Nevertheless, the literature provides strong support for both the selection of the investigated models and the experimental design adopted in this study.

3 Methodology

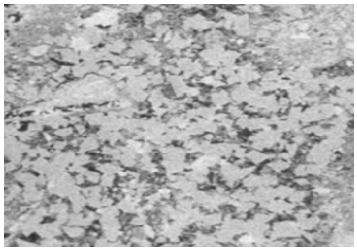
3.1 General Workflow

The implemented workflow consisted of the following main stages:

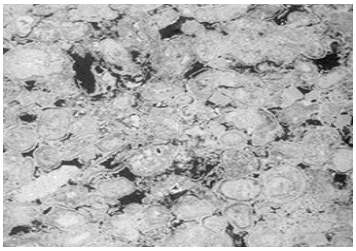
1. Preparation of nine petrographic texture images.
2. Integration of these textures into the Randen framework.
3. Construction of a composite labeled image using the class mask.
4. Generation of training image patches and label patches.
5. Training of U-Net and Attention U-Net using different settings.
6. Evaluation on the composite image.
7. Comparison of accuracy and timing results.

3.2 Texture Image Preparation

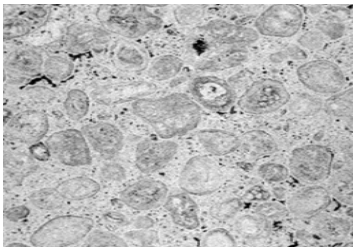
Nine texture images were used in this project. These images were then used to replace the original class textures in the Randen training structure. Since the available texture set contained nine images, the modified training texture array had nine classes.



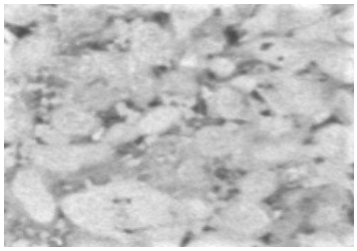
(a) Texture 1



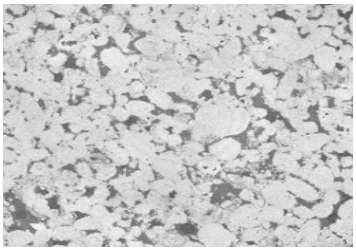
(b) Texture 2



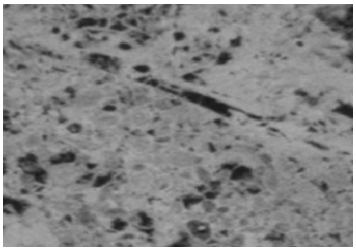
(c) Texture 3



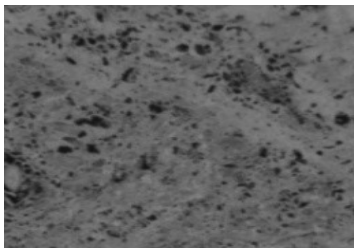
(d) Texture 4



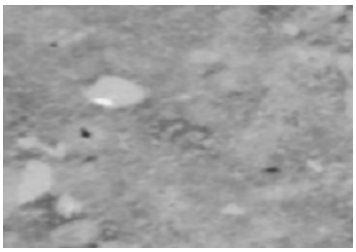
(e) Texture 5



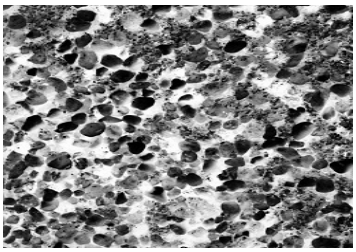
(f) Texture 6



(g) Texture 7



(h) Texture 8



(i) Texture 9

Figure 4: The nine grayscale petrographic texture images used in the study. These textures were integrated into the modified Randen framework and treated as the nine segmentation classes.

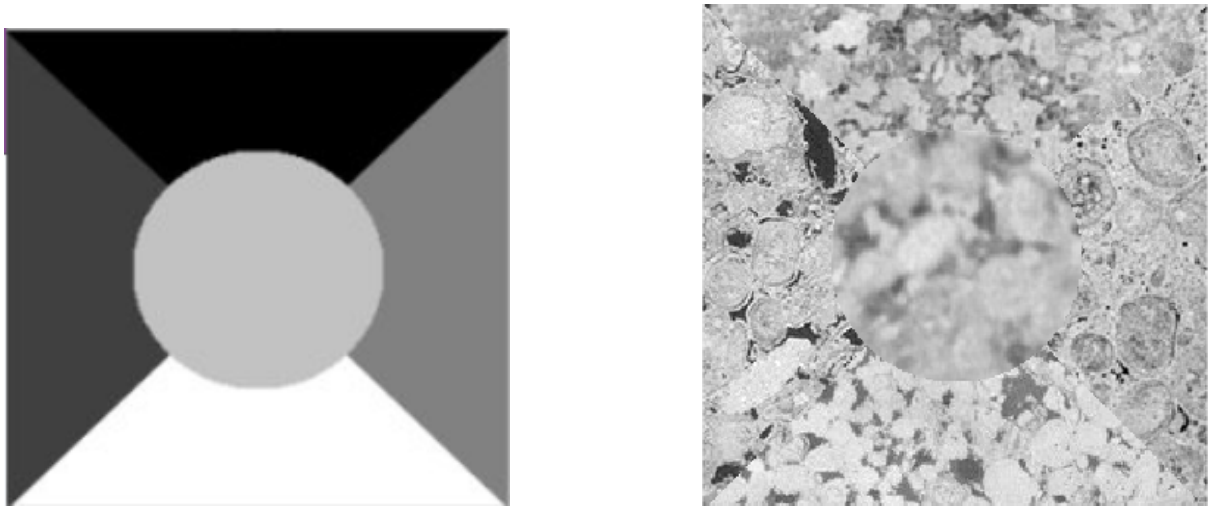
3.3 Composite Image Construction

A new composite image was created using the mask from the Randen dataset and the nine texture images. For each class c, the mask region belonging to that class was multiplied by the corresponding grayscale texture image. The final composite image was obtained by summing the class-specific contributions:

$$I_{\text{composite}} = \sum_{c=1}^{9} M_c \odot T_c,$$

where M_c denotes the binary mask for class c, T_c is the corresponding texture image, and $\odot$ denotes element-wise multiplication.

This produced a synthetic but fully labeled multi-class image suitable for controlled semantic segmentation experiments.



(a) Class mask (b) Composite image

Figure 5: Constructed segmentation target used in the experiments. The class mask defines the spatial regions of the classes, while the composite image is obtained by filling each region with its corresponding petrographic texture.

3.4 Training Patch Generation

Training data were generated from the modified texture set using patches of size 32×32. Two kinds of training samples were created:

- **Single-class patches**, where the entire patch belongs to one class
- **Two-class patches with vertical boundaries**, where the left half of the patch belongs to one class and the right half belongs to another class

These two-class patches introduce simple class boundaries into the training data, allowing the models to learn transitions between textures. However, the boundaries are limited to vertical splits, which represents a simplified boundary structure compared to more complex real-world texture interfaces.

This approach increased the diversity of the training data and exposed the models to both pure texture regions and texture boundaries. This patch-based training strategy follows the general benchmark logic used in the Randen-style texture segmentation workflow, adapted here by replacing the original benchmark textures with petrographic textures [5].

3.5 Implemented Models

3.5.1 U-Net Configurations

For U-Net, three architecture variants were evaluated:

- 15-layer U-Net

- 20-layer U-Net
- 20B U-Net

The 15-layer U-Net is the shallower configuration used in the experiments, while the 20-layer U-Net includes an additional encoder-decoder depth level, leading to more repeated convolution, pooling, and upsampling operations. The 20B U-Net is the third U-Net variant tested in this study. In implementation terms, it follows the same general deeper structure as the 20-layer model and was included as an additional comparison variant within the benchmark-style experimental design.

3.5.2 Attention U-Net Configurations

For Attention U-Net, two architecture variants were implemented:

- 15-layer Attention U-Net
- 20-layer Attention U-Net

The Attention U-Net models used encoder-decoder structures similar to U-Net, but with attention gates inserted into the skip connections. These gates combine encoder features and decoder gating signals through learnable transformations, followed by nonlinear activation and sigmoid-based weighting to generate attention coefficients. The resulting coefficients are then applied through element-wise multiplication to refine the transmitted feature maps.

Unlike the U-Net experiments, a third deeper Attention U-Net variant analogous to the 20B U-Net was not included. This was mainly due to the substantially higher training cost already observed for Attention U-Net, which made the experimental comparison more computationally demanding within the available project time.

3.6 Training Setup

The models were trained in MATLAB. The following settings were used:

- **Input size:** 32×32
- **Initial learning rate:** 10^{-3}
- **Mini-batch size:** 64
- **Shuffle:** every epoch
- **Optimizers:** SGDM, Adam, RMSprop
- **Epoch counts:** 10, 24, 50, 100

The three optimizers were selected in order to compare different gradient-based training behaviors under the same segmentation setup. SGDM was included as a classical momentum-based optimizer, RMSprop was included because it adapts the learning rate based on recent gradient magnitudes, and Adam was included because it combines adaptive learning rates with momentum. Comparing these optimizers made it possible to assess how optimization strategy influences segmentation accuracy across the tested network configurations.

Across the tested architectures, convolutional layers used 3×3 kernels to capture local texture patterns while preserving fine spatial detail. Downsampling was performed using 2×2 max-pooling layers, and decoder reconstruction was performed using transposed convolution layers for upsampling. This encoder-decoder structure allowed the models to progressively extract contextual information and then recover pixel-level segmentation output.

3.7 Dataset Split and Size

A total of 3969 image patches of size 32×32 were generated from the texture images integrated into the Randen framework and used as the training dataset. Each image patch has a corresponding pixel-wise label, resulting in the same number of labeled samples.

In this study, all generated patches were used for training, and a separate validation set was not employed. This follows the original Randen-based framework, where training is performed on generated patches and evaluation is carried out on a composite image with a known ground-truth segmentation map. Since the dataset is constructed in a controlled manner and the ground truth is explicitly defined, model performance can be directly evaluated using pixel-wise accuracy. This setup is consistent with the benchmark-style workflow used in previous MATLAB-based texture segmentation studies built on the Randen composites [5].

3.8 Experimental Design

The experiments were carried out systematically by varying architecture, optimizer, and number of epochs.

For U-Net, the combinations included:

- 3 architecture variants
- 3 optimizers
- 4 epoch settings

For Attention U-Net, the combinations included:

- 2 architecture variants

- 3 optimizers
- 4 epoch settings

3.9 Timing Measurements

After identifying the best-performing configuration for each model family from the main experiment tables, each selected best configuration was rerun separately to measure training time and inference time.

Training time is the total time required to train the network on the prepared training dataset. Inference time is the time required for a trained network to produce the segmentation prediction for the composite test image.

The selected best configurations were:

- **Best U-Net:** 15 layers, Adam, 24 epochs
- **Best Attention U-Net:** 15 layers, SGDM, 50 epochs

As the timing values were obtained from rerunning the selected configurations, the accuracy values in the timing experiment were slightly different from those in the original main result tables. This is expected in deep learning because training depends on initialization and optimization dynamics.

3.10 Implementation and Code Basis

The implementation was carried out in MATLAB. The overall workflow follows the Randen-style texture segmentation benchmark structure used in previous comparative studies, particularly the MATLAB-based implementation discussed by Karabag et al. [5]. In that study, six composite texture images from the Randen and Husøy benchmark were segmented with U-Net configurations of varying depth, epoch number, and optimization algorithm, and the code was made publicly available through GitHub [5]. In the present project, that framework was adapted by replacing the original benchmark textures with nine petrographic texture images and by constructing a corresponding composite image and training-patch dataset.

The U-Net experiments therefore follow an adapted benchmark workflow, while the Attention U-Net implementation was developed within the same MATLAB pipeline so that both model families could be compared under consistent experimental conditions.

For reproducibility, the MATLAB scripts used for texture replacement, composite construction, patch generation, U-Net training, and Attention U-Net training are available through the project GitHub repository: https://github.com/Lina546566/carbonate-texture-segmentation

4 Results

4.1 U-Net Results

Table 1 shows the segmentation accuracy obtained for U-Net across all tested combinations.

Table 1: U-Net segmentation accuracy for Case 1

Epochs	**Layers**	**SGDM**	**Adam**	**RMSprop**
10	15	0.5954	0.7999	0.7666
10	20	0.2400	0.6973	0.5888
10	20B	0.2198	0.7765	0.5364
24	15	0.7958	**0.9093**	0.8721
24	20	0.6978	0.8379	0.8224
24	20B	0.6956	0.8395	0.6672
50	15	0.8638	0.8951	0.8836
50	20	0.7705	0.8321	0.7626
50	20B	0.7894	0.8764	0.7410
100	15	0.8845	0.9065	0.8969
100	20	0.8226	0.5922	0.6831
100	20B	0.7550	0.5583	0.7821

The best U-Net result was achieved by the 15-layer architecture trained with Adam for 24 epochs, giving an accuracy of 0.9093. The 15-layer model also remained strong at 100 epochs with Adam, reaching 0.9065, which suggests that the shallower architecture was consistently well suited to the patch size and texture characteristics used in this study.

4.2 Attention U-Net Results

Table 2 shows the segmentation accuracy obtained for Attention U-Net across the tested combinations.

Table 2: Attention U-Net segmentation accuracy for Case 1

Epochs	Layers	SGDM	Adam	RMSprop
10	15	0.6979	0.6860	0.7443
10	20	0.6433	0.4444	0.5296
24	15	0.7795	0.7499	0.8084
24	20	0.6943	0.5272	0.8120
50	15	**0.8920**	0.7771	0.3559
50	20	0.8199	0.7670	0.8838
100	15	0.8142	0.7310	0.8084
100	20	0.5835	0.3270	0.8741

The best Attention U-Net result was obtained by the 15-layer architecture trained with SGDM for 50 epochs, giving an accuracy of 0.8920. Another strong result was obtained by the 20-layer model trained with RMSprop for 50 epochs, which reached 0.8838. This indicates that Attention U-Net performance was competitive, but also more sensitive to architecture depth and optimizer choice than standard U-Net.

4.3 Best Main Experimental Results

Table 3 compares the best configuration from each model family based on the main experiment tables.

Table 3: Best configurations from the main experiment tables

Model	Layers	Optimizer	Epochs	Accuracy
U-Net	15	Adam	24	**0.9093**
Attention U-Net	15	SGDM	50	0.8920

Based on the main experimental results, U-Net achieved the highest overall accuracy in this project. This suggests that, under the present controlled texture segmentation setup, the additional complexity introduced by attention mechanisms did not provide a sufficient advantage to outperform the baseline U-Net.

4.4 Timing Results

Table 4 presents the training and inference times obtained by rerunning the selected best configuration from each model family. These reruns were performed to measure computational cost and generate additional outputs such as qualitative results and evaluation metrics.

Table 4: Timing results for the selected best configurations (rerun)

Model	Layers	Optimizer	Epochs	Accuracy	Train (s)	Infer (s)
U-Net	15	Adam	24	0.8864	1146.93	36.67
Attention U-Net	15	SGDM	50	0.7865	12951.10	33.40

For easier interpretation, the training times correspond approximately to:

- U-Net: about 19.12 minutes
- Attention U-Net: about 215.85 minutes

The results show a significant difference in training time between the two models. The Attention U-Net requires substantially more computation due to its more complex architecture and attention mechanisms. In contrast, U-Net achieves faster training while maintaining higher accuracy.

It is also important to note that the inference times are of similar magnitude for both models in the rerun experiment. This suggests that the main computational difference between the two architectures appears during training rather than during inference.

4.5 Additional Metrics and Qualitative Results

The best-performing configuration from each model family was rerun to obtain qualitative results and additional evaluation metrics, including precision, recall, and F1-score. These metrics provide a more detailed assessment of segmentation performance beyond pixel-wise accuracy.

As deep learning training is stochastic, the results obtained from these reruns differ slightly from those reported in the main experimental tables. This variation is expected due to differences in weight initialization and training dynamics, and does not indicate a change in model performance trends.

For the U-Net model, the rerun achieved an accuracy of 0.8864, with a mean precision of 0.5066, mean recall of 0.4926, and mean F1-score of 0.4971. For the Attention U-Net model, the rerun achieved an accuracy of 0.7865, with a mean precision of 0.4700, mean recall of 0.4364, and mean F1-score of 0.4447.

Table 5: Additional evaluation metrics for the selected best configurations (rerun)

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
U-Net	0.8864	0.5066	0.4926	0.4971
Attention U-Net	0.7865	0.4700	0.4364	0.4447

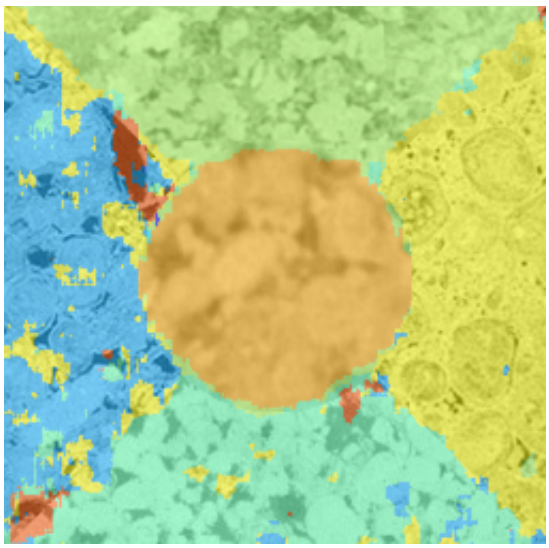
These results show that U-Net not only reaches higher accuracy but also performs better across the other evaluation metrics. At the same time, the moderate precision,

recall, and F1-score values indicate that the segmentation task is still challenging at the class level, even when pixel-level accuracy appears high. This is likely caused by confusion between visually similar texture classes and mistakes that occur around texture boundaries. The relatively higher F1-score suggests a better balance between precision and recall, meaning that U-Net is more capable of correctly detecting texture regions while reducing misclassification.

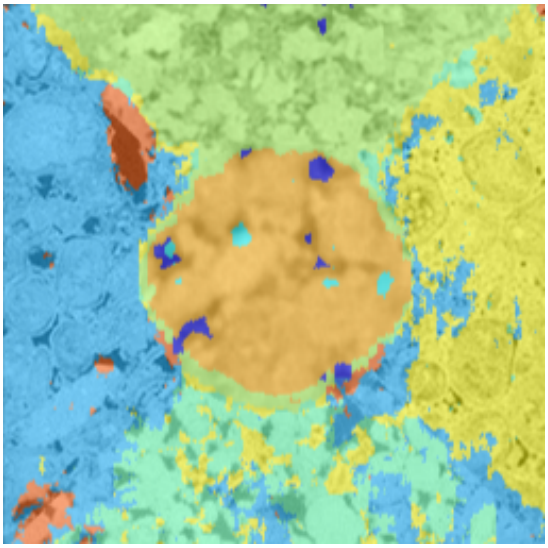
(a) U-Net correct classification map

(b) Attention U-Net correct classification map

Figure 6: Correct classification maps for the rerun best configurations. White pixels indicate correctly classified regions, while black pixels indicate misclassified regions.



(a) U-Net overlay



(b) Attention U-Net overlay

Figure 7: Overlay visualizations of predicted segmentation. These figures illustrate how each model assigns class labels across the composite image.

The qualitative results further support the quantitative findings. The U-Net model

shows a higher proportion of correctly classified regions, with fewer misclassified pixels compared to Attention U-Net. This demonstrates that U-Net provides more consistent segmentation performance for the given texture dataset.

5 Discussion

5.1 Why the Chosen Route Makes Sense

The route taken in this project was motivated by both the weaknesses of traditional texture segmentation methods and the strengths of encoder–decoder CNNs reported in previous work. Classical approaches such as co-occurrence matrices, LBP, filtering, watershed, and multiresolution sub-band filtering can produce strong results on benchmark textures, but they depend on manually designed descriptors and parameter choices [6, 7, 8, 9]. The comparative benchmark study by Karabag et al. showed that these methods remain competitive, but that a properly configured U-Net achieved the lowest average misclassification overall on the Randen composites [5]. This supports the decision to use a learned deep segmentation model rather than relying only on handcrafted descriptors.

U-Net was therefore chosen as a strong baseline because it has repeatedly shown robust performance in segmentation tasks across different domains, including biomedical imaging and benchmark texture segmentation studies [3, 5, 23]. Attention U-Net was chosen because the literature suggests that attention gates can improve performance when relevant structures are difficult to isolate, especially in cases with variable scale or weak boundaries [4, 24, 25]. In that sense, the present project was designed not only to test whether the models work, but also to evaluate whether the added attention mechanism provides a practical advantage for petrographic textures.

5.2 Effect of Network Depth

One of the clearest trends in the present experiments is that the shallower 15-layer configurations generally performed better than the deeper variants. This happened in both model families. Since the input patches were only 32×32, repeated downsampling in deeper networks quickly reduces the amount of usable spatial detail. In this situation, additional depth does not necessarily provide more useful semantic abstraction, but can instead remove discriminative local texture cues.

This interpretation is consistent with previous benchmark findings. Karabag et al. reported that U-Net performance varied substantially across depth, optimizer, and epoch settings, and that the best results were distributed across configurations rather than always favoring a deeper network [5]. In the present project, the 15-layer structure appears to preserve a better balance between contextual modeling and local texture detail.

5.3 Effect of Optimizer

Optimizer choice had a strong effect on the results. For U-Net, Adam produced the best overall configuration, which is consistent with the idea that Adam can converge efficiently in moderate-sized segmentation problems by combining adaptive learning rates with momentum-like updates [16]. In this project, that likely helped the 15-layer U-Net learn the relevant texture patterns quickly and stably.

For Attention U-Net, the best result came from SGDM at 50 epochs, while RMSprop also produced some competitive results for deeper settings. This suggests that the attention-based architecture is more sensitive to optimization dynamics. Attention gates increase architectural complexity by adding extra transformations and feature-weighting operations, so the corresponding optimization landscape may be less stable. In such cases, a more conservative optimizer such as SGDM can sometimes provide a better training trajectory. This is also compatible with the broader literature, where no single optimizer is universally best across all segmentation tasks or model variants [5].

5.4 Effect of Number of Epochs

More epochs did not always improve performance. Some configurations benefited from longer training, but others reached their best result much earlier. For example, the best U-Net result in the main experiments was obtained at 24 epochs rather than 100 epochs. This suggests that the networks learned the dominant texture patterns relatively early and that longer training sometimes led to unstable optimization or partial overfitting to patch-level details.

This behavior is again in line with benchmark work showing that the best U-Net performance on texture composites is not always obtained at the maximum epoch count [5]. Therefore, the number of epochs should be treated as a tuning parameter rather than assuming that longer training always improves segmentation.

5.5 Why U-Net Performed Better than Attention U-Net Here

The main experimental tables showed that U-Net achieved the highest overall accuracy, with a best value of 0.9093, while Attention U-Net reached 0.8920 in its best configuration. The rerun-based metrics and qualitative results also favored U-Net. There are several reasons why this outcome makes sense.

First, the segmentation task in this project is highly texture-driven and patch-based. The class differences are often local and visually strong, so the standard U-Net can already capture them effectively through its convolutional filters and skip connections. In this situation, the additional attention mechanism may not provide enough extra benefit to overcome its added complexity.

Second, attention mechanisms are especially helpful in problems where the target

structures are subtle, sparse, or heavily mixed with irrelevant background. This is exactly the kind of scenario emphasized in the original Attention U-Net work, where attention improved pancreas segmentation by refining focus on difficult anatomical targets [4]. In the current project, however, the composite is structured and the class regions are already defined by distinctive textures. Therefore, the selective gating mechanism is less critical than in medical datasets with small organs or weak tissue contrast.

Third, the timing results show the computational trade-off clearly. Attention U-Net required substantially longer training time than U-Net in the rerun, while the inference times were of similar magnitude. This indicates that the added architectural complexity did not translate into improved segmentation performance in the present setting.

Similar trade-offs have been reported in the literature. Attention-based U-Net variants improve segmentation performance by emphasizing relevant regions and suppressing background features, but they also introduce additional architectural components such as attention gates and residual units, increasing model complexity [24].

For the present controlled texture segmentation problem, this trade-off favored the standard U-Net, which achieved higher accuracy while requiring significantly less training time.

5.6 Texture Difficulty and Which Textures Were Easier

Another important point is that not all textures are equally difficult to segment. In the Randen-style benchmark literature, some texture arrangements are easier because the classes differ strongly in granularity, directionality, or local repetition, while others are harder because the regions remain visually similar even after equalization [5]. Karabag et al. explicitly reported that the easiest benchmark case had U-Net misclassification as low as 2.6%, while harder cases reached 17.5% even for the best U-Net configuration [5]. This indicates that texture similarity itself is a major source of segmentation difficulty.

The qualitative maps in the present report suggest a similar pattern. Regions with more distinctive texture contrast appear easier for both networks, while more visually similar areas and some class boundaries show more misclassification. This indicates that performance is not determined only by architecture, but also by how separable the texture classes are in terms of local appearance. In other words, textures that differ strongly in orientation, roughness, or granularity are easier to segment, whereas textures with closer local statistics are more difficult.

5.7 Limitations and Future Work

This work has several limitations. First, the experiments were carried out on a constructed composite image rather than a large fully annotated carbonate μCT dataset. The current setup is therefore controlled and appropriate for methodological comparison, but still simpler than a full geological deployment. Second, the textures were integrated

into a Randen-style benchmark framework, so the study emphasizes controlled texture segmentation rather than direct reservoir-scale analysis.

Third, the training setup did not include a separate validation split. This follows the benchmark-style framework, where evaluation is performed directly on a composite image with known ground truth. Although this is acceptable for the current report, future publication-oriented work should extend the design to additional cases and broader validation analysis. Finally, while pixel-wise accuracy, rerun-based precision, recall, and F1-score were reported, future work could also include IoU and Dice-based evaluation.

A natural next step is to apply the same methodology to larger and more realistic carbonate μCT datasets with expert labels. This is particularly promising because U-Net has already shown strong results in digital rock segmentation studies. For example, deep learning-based segmentation approaches have demonstrated high accuracy in distinguishing pore and matrix structures in rock images, highlighting the suitability of encoder–decoder architectures for such tasks [26]. These results suggest that the current project provides a reasonable methodological basis for future real-data applications.

6 Conclusion

This project explored the use of supervised deep learning methods for semantic segmentation of petrographic textures, focusing on U-Net and Attention U-Net architectures. A customized texture segmentation framework was developed using nine grayscale petrographic images within the Randen setup. The training dataset consisted of labeled patches of size 32×32, and a series of experiments were conducted by adjusting parameters such as network depth, optimizer choice, and number of training epochs.

In the rerun conducted for timing, qualitative assessment, and additional evaluation metrics, the chosen U-Net configuration achieved an accuracy of 0.8864, with a training time of 1146.93 seconds and an inference time of 36.67 seconds. It also recorded a mean precision of 0.5066, a mean recall of 0.4926, and a mean F1-score of 0.4971. In comparison, the selected Attention U-Net configuration reached an accuracy of 0.7865, required 12951.10 seconds for training, and had an inference time of 33.40 seconds, while also producing lower precision, recall, and F1-score values.

These results further confirm that U-Net provides better segmentation performance while requiring significantly less training time, making it more efficient and suitable for this application.

Overall, U-Net gave the strongest balance of accuracy and computational efficiency in this implementation. The results suggest that supervised encoder-decoder architectures are effective for controlled multi-class texture segmentation and that U-Net is a strong baseline for this problem. A useful next step would be to apply the same methodology to larger and more realistic carbonate μCT datasets with expert labels and to evaluate performance using additional segmentation metrics.

References

[1] M. S. Jouini, A. O. Alabere, M. Alsuwaidi, S. Morad, F. Bouchaala, and O. A. Al Jallad, "Experimental and digital investigations of heterogeneity in lower cretaceous carbonate reservoir using fractal and multifractal concepts," *Scientific Reports*, vol. 13, p. 20306, 2023.

[2] M. S. Jouini, F. Bouchaala, M. K. Riahi, M. Sassi, H. Abderrahmane, and F. Hjouj, "Multifractal analysis of reservoir rock samples using 3d x-ray micro computed tomography images," *IEEE Access*, vol. 10, pp. 1–12, 2022.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

[4] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[5] C. Karabag, J. Verhoeven, N. R. Miller, and C. C. Reyes-Aldasoro, "Texture segmentation: An objective comparison between five traditional algorithms and a deep-learning u-net architecture," *Applied Sciences*, vol. 9, no. 18, p. 3900, 2019.

[6] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[7] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[8] D. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-d gabor elementary functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 130–149, 1994.

[9] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[10] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proceedings of the International Conference on Image Processing (ICIP)*, pp. 444–447, 1995.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 28 of *JMLR Workshop and Conference Proceedings*, 2013.

[15] T. Tieleman, "Lecture 6.5 – rmsprop: Divide the gradient by a running average of its recent magnitude," 2012. Coursera: Neural Networks for Machine Learning.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (glcm) texture based crop classification using low altitude remote sensing platforms," *PeerJ Computer Science*, vol. 7, p. e536, 2021.

[18] C. H. Chan, *Multi-Scale Local Binary Pattern Histogram for Face Recognition*. PhD thesis, University of Surrey, 2008.

[19] O. S. Al-Kadi, "A gabor filter texture analysis approach for histopathological brain tumour subtype discrimination," *ISESCO Journal*, 2017.

[20] A. Alekseev and A. Bobe, "Gabornet: Gabor filters with learnable parameters in deep convolutional neural networks," *arXiv preprint arXiv:1904.13204*, 2019.

[21] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186–1193, 1993.

[22] G. Yu and S. V. Kamarthi, "A cluster-based wavelet feature extraction method and its application," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 196–202, 2010.

[23] Y. Li *et al.*, "Efficient cerebral infarction segmentation using u-net variants," *Biomedical Engineering Online*, vol. 24, p. 45, 2025.

[24] Z. Li, H. Zhang, Z. Li, and Z. Ren, "Residual-attention unet++: A nested residual-attention u-net for medical image segmentation," *Applied Sciences*, vol. 12, no. 14, p. 7149, 2022.

[25] C. Li, L. Fu, Q. Zhu, J. Zhu, Z. Fang, Y. Xie, Y. Guo, and Y. Gong, "Attention enhanced u-net for building extraction from farmland based on google and worldview-2 remote sensing images," *Remote Sensing*, vol. 13, no. 21, p. 4411, 2021.

[26] M. S. Jouini, J. S. Gomes, M. Tembely, and E. R. Ibrahim, "Upscaling strategy to simulate permeability in a carbonate sample using machine learning and 3d printing," *IEEE Access*, 2021.