

ARTICLE

Ensembles of lightweight U-Net-based models with EfficientNet encoders for semantic seismic data segmentation

Alexandre Gonçalves Silva^{1*}, André Tavares da Silva², Rafael de Santiago¹, and Mauro Roisenberg¹¹Department of Informatics and Statistics, School of Technology, Federal University of Santa Catarina, Florianópolis, Santa Catarina, Brazil²Department of Computer Science, Center for Technological Sciences, Santa Catarina State University, Joinville, Santa Catarina, Brazil

Abstract

Automated seismic facies segmentation can improve the consistency and efficiency of subsurface interpretation, but complex stratigraphic boundaries and imbalanced facies remain challenging. This work evaluates a reproducible pipeline for multiclass seismic facies segmentation based on simple probability-level ensembles of three U-Net-derived architectures that share an EfficientNet-B4 encoder: UNet++, CMUNeXt, and MultiResUNet. The models were trained and tested on the real field-acquired Netherlands F3 seismic benchmark, which contains six interpreted facies. During inference, each network produces class-wise softmax probability maps, and the ensemble averages these maps before pixel-wise classification. The individual models achieved mean intersection over union (mIoU) values of 0.710, 0.744, and 0.757, respectively. The CMUNeXt–MultiResUNet ensemble reached an mIoU of 0.762, while the three-model ensemble achieved the best overall balance among the proposed configurations, with mIoU = 0.766, frequency-weighted intersection over union (FWIoU) = 0.883, pixel accuracy (PA) = 0.934, and mean class accuracy (MCA) = 0.860. In a single-run ablation, disabling all training-time augmentation while retaining the same fixed normalization reduced the three-model ensemble mIoU from 0.766 to 0.719; augmentation improved PA, MCA, FWIoU, and mIoU for every evaluated model and ensemble. Relative to published F3 results, the proposed ensemble improves several global metrics over multiple baselines but does not surpass every prior method. These findings show that low-complexity ensemble averaging can exploit complementary class-wise behavior without introducing a separate meta-learner, providing a competitive and computationally practical baseline for automated seismic interpretation.

***Corresponding author:**
Alexandre Gonçalves Silva
(alexandre.goncalves.silva@ufsc.br)

Citation: Silva AG, Silva AT, Santiago R, Roisenberg M. Ensembles of lightweight U-Net-based models with EfficientNet encoders for semantic seismic data segmentation. *J Seismic Explor*. doi: 10.36922/JSE026260116

Received: June 25, 2026

Revised: August 29, 2026

Accepted: August 31, 2026

Published online: September 16, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Keywords: Seismic segmentation; UNet++; CMUNeXt; MultiResUNet; EfficientNet-B4; Ensemble learning; Stratigraphic analysis

1. Introduction

Seismic facies interpretation supports reservoir characterization by delineating stratigraphic units and other structures from seismic reflection data. Manual interpretation is time-consuming, relies heavily on expert experience, and can vary

across interpreters. Automated semantic segmentation can improve consistency and throughput, but practical methods must also preserve thin boundaries, recognize underrepresented facies, and remain computationally feasible for large seismic volumes.

Early deep-learning studies established the feasibility of supervised seismic facies classification and the importance of representative benchmarks. Alaudah *et al.*¹ released the annotated Netherlands F3 benchmark and standardized section-based and patch-based evaluation. Zhang *et al.*² demonstrated that convolutional representations can learn diagnostic seismic patterns, while Zhang *et al.*³ extended the supervised paradigm through a large, knowledge-guided benchmark assembled from curated field data, synthetic samples, and generative modeling. Together, these studies shifted the field from isolated demonstrations toward reproducible data-driven interpretation.

Subsequent work broadened the input representations and segmentation strategies. Trindade and Roisenberg⁴ combined orthogonal two-dimensional views and geological heuristics to approximate three-dimensional reasoning at lower computational cost. Reis⁵ showed that seismic attributes can complement amplitude information. Kaur *et al.*⁶ compared DeepLabV3+ and generative adversarial segmentation, reporting a trade-off between sharper class boundaries and greater spatial continuity. These results indicate that architectural design, input representation, and spatial regularity must be considered jointly.

A second research trajectory addresses scarce labels and domain shift. Nasim *et al.*⁷ combined an encoder–decoder network with CORAL-based domain adaptation between the F3 and Penobscot datasets. Mustafa and AlRegib⁸ used deep-autoencoder representations within an active-learning loop to prioritize informative samples for expert annotation. These approaches reduce the dependence on dense labels, although they introduce additional training stages and selection mechanisms.

Architecture development has progressed from modified U-Net models to transformers, self-supervised pretraining, and automated network design. Tolstaya and Egorov⁹ combined an EfficientNet-backed U-Net with domain-specific warping, total-variation regularization, and pseudolabeling. Wang *et al.*¹⁰ used a SegFormer-based encoder–decoder with hypercolumn fusion, and De Miranda¹¹ investigated masked-autoencoder pretraining for vision transformers. More recently, Gao *et al.*¹² used differentiable architecture search to discover a seismic-specific network on F3. These methods can improve performance, but their additional modules, pretraining stages, or search procedures may complicate

implementation and comparison.

Recent studies further broadened the F3 segmentation landscape. Guo *et al.*¹³ coupled super-resolution reconstruction with semantic segmentation, Li *et al.*¹⁴ reduced label dependence through self-supervised learning, and Tian *et al.*¹⁵ introduced click-guided interactive segmentation with a hybrid U-shaped transformer. In parallel, Canguçu *et al.*¹⁶ proposed a standardized multivolume benchmark, while Gutierrez *et al.*¹⁷ showed that controlled evaluation can change the apparent ranking of convolutional neural network- and transformer-based models. Consequently, comparisons across studies should be interpreted as contextual rather than as proof of universal superiority, particularly when training splits, augmentation policies, or test protocols differ.

Against this background, a practical gap remains: the literature has extensively optimized individual architectures, but there is less evidence on whether simple ensembles of complementary, moderately sized U-Net variants can provide robust gains without a learned fusion module or an expensive search procedure. This study addresses this question by holding the encoder family and evaluation dataset fixed while varying the decoder design and combining class-probability outputs by arithmetic averaging.

The main contributions are: (i) a common EfficientNet-B4-based implementation of UNet++, CMUNeXt, and MultiResUNet for six-class seismic segmentation; (ii) a controlled comparison of the individual models and all pairwise and three-model averaging ensembles; (iii) an analysis of ensemble complementarity using pixel agreement and Jensen–Shannon divergence; (iv) a single-run ablation comparing full training-time augmentation with normalization-only training; and (v) an explicit comparison of performance and computational cost on the F3 benchmark.

The method is evaluated on field-acquired seismic data rather than on a purely synthetic image collection. However, the evidence is limited to the public F3 block; applicability to other surveys, acquisition conditions, and geological settings requires external validation. The remainder of the paper describes the dataset, preprocessing, architectures, loss function, training and inference procedures, evaluation metrics, results, limitations, and conclusion.

2. Materials and methods

2.1. Dataset

The Netherlands F3 dataset¹⁸ originates from field-acquired three-dimensional seismic data in Block F3 of the southern

North Sea. The benchmark distributed by Alaudah *et al.*¹ was selected because its public annotations, predefined partitions, and broad use enable comparison with earlier work. The arrays used in this study comprise a training block of $401 \times 701 \times 255$ samples and the held-out test block of $200 \times 601 \times 255$ samples, where the dimensions correspond to the inline, crossline, and sample axes. All 401 labeled inline sections from the training block, each containing $701 \text{ traces} \times 255 \text{ samples}$, were used for model fitting, with no separate validation subset. For evaluation, the test block was sliced along both orthogonal orientations, yielding 200 inline sections of $601 \text{ traces} \times 255 \text{ samples}$ and 601 crossline sections of $200 \text{ traces} \times 255 \text{ samples}$, for a total of 801 two-dimensional test images. Thus, the 801 images are complete orthogonal sections extracted from the same held-out three-dimensional block, rather than additional patches, augmented samples, or independent seismic sections. The corresponding masks encode six interpreted facies: Upper North Sea (N. S.), Middle N. S., Lower N. S., Rijnland/Chalk, Scruff, and Zechstein. [Figure 1](#) shows one test section and its reference interpretation. Although the seismic amplitudes are real measurements, the labels are expert-derived interpretations and therefore remain subject to interpretation uncertainty.

Data augmentation was applied only to the training set in the primary experiments. Paired spatial transformations were applied identically to each seismic image and its label mask, whereas intensity transformations were restricted to the seismic image. The full pipeline included random cropping, additive Gaussian noise, Gaussian blur, and modest brightness and contrast perturbations. These operations preserve the principal stratigraphic geometry while exposing the models to plausible local variations in sampling, noise, and amplitude. In the ablation condition, all augmentation operators were disabled, and the training images underwent only the fixed normalization described below. Both conditions used the same 401-section training partition, no separate validation partition, the same 200-epoch schedule and evaluation metrics, and the same 801-image held-out test evaluation set described above. Representative samples from the full augmentation pipeline are shown in [Figure 2](#).

For an 8-bit input I , preprocessing first scales amplitudes to $I_s = I / 255$ and then standardizes them as $I_{\text{norm}} = (I_s - \mu) / \sigma$. The implementation uses the fixed scalar parameters $\mu = 0.449$ and $\sigma = 0.226$ for the single grayscale channel. The same values were used in the normalization-only and full-augmentation conditions. The test set was normalized with these fixed values and was not augmented.

2.2. Deep learning architectures

U-Net is widely used for semantic segmentation because

its encoder–decoder topology combines multiscale context with spatial detail propagated through skip connections. These properties are also relevant to seismic sections, in which thin interfaces and laterally continuous structures must be recovered from noisy texture. This study evaluates three complementary U-Net-derived decoders (UNet++, CMUNeXt, and MultiResUNet) individually and in probability-level ensembles.

2.2.1. Loss function

For each supervised output, the training objective combines three complementary components: focal loss, Dice loss, and a boundary-aware loss. Let $z \in \mathbb{R}^{C \times H \times W}$ denote the logits, $p = \text{SOFTMAX}(z)$ the class-probability map, $y \in \{0, \dots, C-1\}^{H \times W}$ the discrete ground-truth label mask, and $t = \text{ONEHOT}(y) \in \{0, 1\}^{C \times H \times W}$ its one-hot representation. Pixel-wise cross-entropy $\mathcal{L}_{CE}^{(n)}(z, y)$ provides the basis for the focal term; the Dice term compares p with t , and the boundary term compares the Sobel-derived edges of each predicted class-probability map p_c with those of the corresponding one-hot target map t_c , averaged over the C classes. The loss for one prediction head is defined as **Equation 1**:

$$\mathcal{L}_{\text{single}}(z, y) = \underbrace{\frac{1}{N} \sum_{n=1}^N \left(1 - e^{-\mathcal{L}_{CE}^{(n)}}\right)^2 \mathcal{L}_{CE}^{(n)}}_{\mathcal{L}_{\text{focal}}} + 0.5 \underbrace{\left[\frac{1 - 2 \sum p t + \epsilon}{\sum p + \sum t + \epsilon} \right]}_{\mathcal{L}_{\text{Dice}}} + 0.1 \underbrace{\frac{1}{C} \sum_{c=0}^{C-1} \|S_x * p_c - S_x * t_c\|_1}_{\mathcal{L}_{\text{boundary}}} \quad (1)$$

where N is the number of pixels, $\epsilon = 10^{-6}$ is a numerical-stability constant, and S_x denotes the horizontal Sobel kernel. **Equation 2** contains four weighted head losses because each architecture exposes four deep-supervision prediction heads at progressively refined decoder levels; it does not introduce a fourth loss component. For head logits $z_1, \dots, z_4$, the corresponding probabilities were $p_i = \text{SOFTMAX}(z_i)$, while the same discrete label mask y and one-hot target t were used for every head. The total loss used $w_i = (1, 0.5, 0.25, 0.125)$. The largest weight was assigned to the final high-resolution output, while geometrically decreasing auxiliary weights preserved gradient flow to earlier decoder stages. These weights were fixed before evaluation and were not tuned on the test set.

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^4 w_i \mathcal{L}_{\text{single}}(z_i, y), w_i = [1.0, 0.5, 0.25, 0.125] \quad (2)$$

2.2.2. UNet++

UNet++ extends the original U-Net through nested, densely connected skip pathways between encoder and decoder stages.¹⁹ Instead of connecting each encoder level directly to a single decoder block, intermediate decoder nodes progressively reduce the semantic gap between low-level spatial features and high-level contextual features. In the adopted configuration, the encoder produces

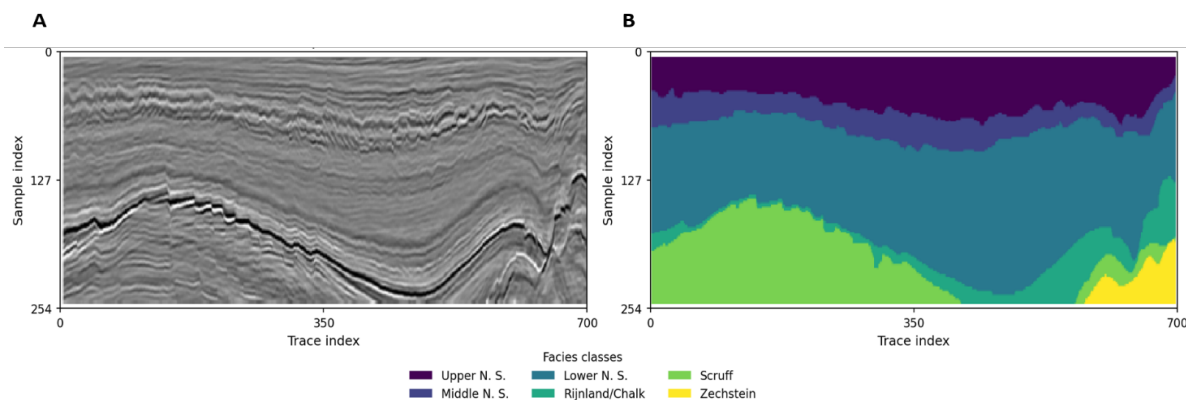


Figure 1. Field-acquired seismic section from the F3 test set and its corresponding ground-truth facies interpretation. (A) Seismic amplitude section. (B) Ground-truth facies.

Abbreviation: N. S.: North Sea.

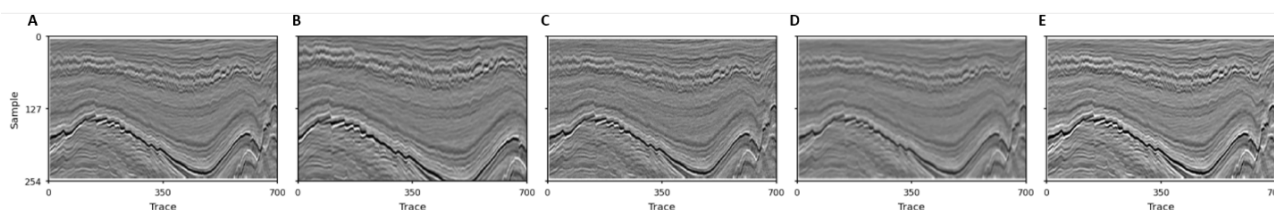


Figure 2. Representative outputs of the training-only augmentation pipeline applied to a field-acquired F3 seismic section. (A) Original. (B) Random crop. (C) Gaussian noise. (D) Gaussian blur. (E) Brightness/contrast.

multiscale feature maps, and the decoder reconstructs high-resolution representations through repeated upsampling, concatenation, and fusion. Figure 3 illustrates this decoding grid. Auxiliary heads attached to multiple high-resolution decoder nodes provide the four outputs used by the deep-supervision loss in Equation 2.

2.2.3. CMUNeXt and MultiResUNet

Figure 4 shows the common five-level encoder and four-stage decoder topology used for CMUNeXt and MultiResUNet. At each stage, the deeper feature map is upsampled, concatenated with the matching encoder feature, and processed by a model-specific refinement block. In the CMUNeXt-style decoder,²⁰ bilinear upsampling is followed by depthwise separable convolution and a FusionConv module with channel attention and dropout. In the MultiResUNet-style decoder,²¹ the concatenated representation is processed by a MultiResBlock that combines sequential multiscale paths with a 1×1 shortcut projection, normalization, activation, channel attention, and dropout. The two models therefore share the same macro-topology and supervision strategy but differ in how they refine fused encoder-decoder features.

2.3. Training algorithm

Algorithm 1 summarizes the model-independent training procedure. The same loop is used for each architecture and includes data loading, optional paired augmentation, multiclass-label construction, mini-batch optimization, deep-supervision-aware loss computation, and checkpoint storage. For the normalization-only ablation, the augmentation block was bypassed while all subsequent operations remained unchanged.

All models used a one-channel input and six output logits followed by a softmax operation. Training used Adam with a base learning rate of 10^{-4} , a batch size of 4, and 200 epochs. UNet++ and MultiResUNet used a base width of 32 filters, whereas CMUNeXt used a lightweight hierarchical channel configuration. The architectures were selected for complementary decoder behavior: UNet++ emphasizes nested multiscale fusion, MultiResUNet emphasizes multiresolution refinement, and CMUNeXt emphasizes efficient depthwise operations and skip fusion.¹⁹⁻²³ Performance was evaluated using pixel accuracy (PA), class accuracy, mean class accuracy (MCA), mean intersection over union (mIoU), and frequency-weighted intersection over union (FWIoU).

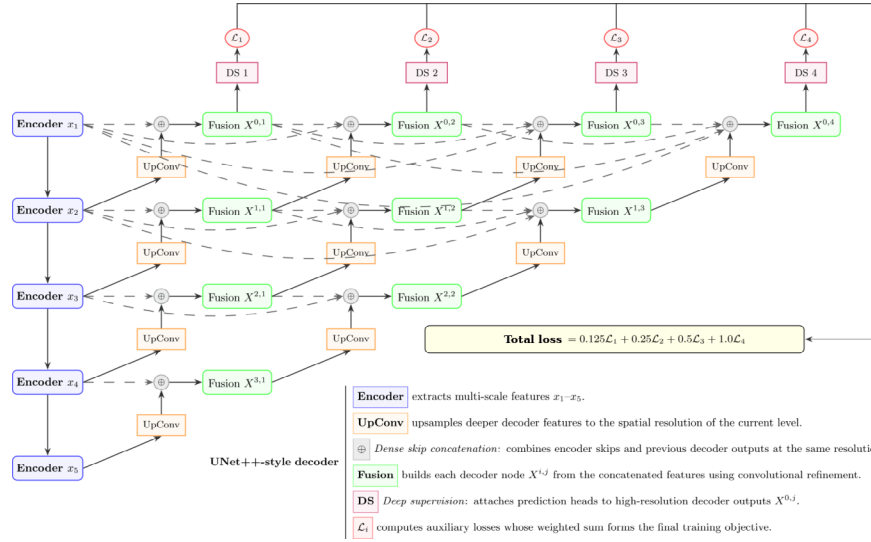


Figure 3. UNet++ decoder with nested dense skip connections and four deep-supervision outputs. Multiscale encoder features are progressively upsampled, concatenated, and fused to form decoder nodes.

Algorithm 1. Training procedure for multiclass seismic facies segmentation

Require: Training images I , class-specific mask directories $M_1, \dots, M_6$, model type A , number of epochs E

Ensure: Trained model

Define training transformations, and build the training dataset

For each image i in the training set do

 Read the grayscale input image

 Pad the image to a spatial size compatible with the network

 Construct the multiclass label map from the class-specific masks

 Apply the same transformations to image and label

End for

Create a shuffled mini-batch data loader

Instantiate the selected segmentation model f_A

Freeze Batch Normalization layers

Initialize the Adam optimizer

For epoch $\leftarrow 1$ to E do

 Set the model to training mode and freeze Batch Normalization layers again

 Initialize cumulative training loss

 For each mini-batch (x, y) of inputs and discrete labels do

 Compute head logits $z_1, \dots, z_4 \leftarrow f_A(x)$

 Compute probabilities $p_i \leftarrow \text{SOFTMAX}(z_i)$ for $i = 1, \dots, 4$

 Set $t \leftarrow \text{ONEHOT}(y)$, then compute the weighted total loss in Equation (II)

 Reset gradients, backpropagate, and update model parameters

 Accumulate the mini-batch loss

 End for

End for

Save the trained model

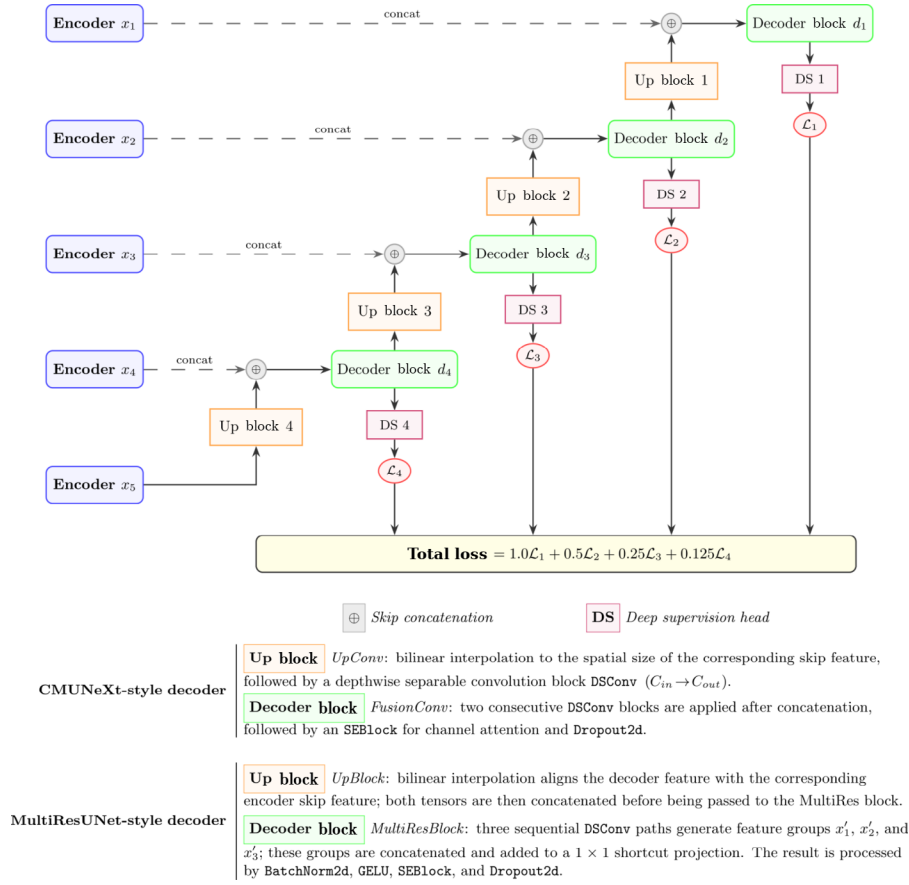


Figure 4. Encoder-decoder topology shared by the CMUNeXt and MultiResUNet models. Encoder features are progressively upsampled, concatenated through skip connections, and refined by model-specific decoder blocks; four auxiliary heads provide deep supervision.

2.4. Multiclass inference

During inference, each grayscale section was padded reflectively so that its spatial dimensions were multiples of 32, normalized, converted to a tensor, and processed by the trained models. For model m , logits z_m were converted by softmax to the six-class probability map p_m , which was cropped back to the original dimensions and renormalized per pixel. The ensemble then combined these maps with non-negative weights that sum to one. Unless otherwise stated, equal weights were used. [Algorithm 2](#) details the procedure. Let $I \in \mathbb{R}^{H \times W}$, $C = 6$, $\mathcal{M} = \{m_1, m_2, m_3\}$, $z_m \in \mathbb{R}^{C \times H \times W}$, and $p_m = \text{SOFTMAX}(z_m)$ denote the per-model class probabilities. The ensemble probability $\bar{p}$ is ([Equation 3](#)):

$$\bar{p}(c, i, j) = \frac{\sum_{m \in \mathcal{M}} w_m p_m(c, i, j)}{\sum_{m \in \mathcal{M}} w_m} \quad (3)$$

where $c \in \{0, \dots, C-1\}$ and (i, j) denotes a pixel. The final discrete prediction $\hat{y}$ assigns the class with the maximum ensemble probability ([Equation 4](#)):

$$\hat{y}(i, j) = \arg \max_{c \in \{0, \dots, C-1\}} \bar{p}(c, i, j) \quad (4)$$

2.5. Evaluation metrics

Performance was evaluated with metrics commonly used for multiclass semantic segmentation. Let G_i be the set of ground-truth pixels in class i , F_i the set predicted as class i , and n_c the number of classes. The intersection $F_i \cap G_i$ contains the correctly classified pixels for class i , and $|\cdot|$ denotes set cardinality. PA is the proportion of correctly classified pixels across the complete test set ([Equation 5](#)):

$$\text{PA} = \frac{\sum_i |F_i \cap G_i|}{\sum_i |G_i|} \quad (5)$$

Class accuracy is the recall of class i , i.e., the proportion of its ground-truth pixels assigned to the correct class ([Equation 6](#)):

$$\text{CA}_i = \frac{|F_i \cap G_i|}{|G_i|} \quad (6)$$

Algorithm 2. Multiclass inference using probability-level ensemble averaging

Require: Image $I \in \mathbb{R}^{H \times W}$; trained model; weights $w_{\text{UNet++}}$, w_{CMUNeXt} , and $w_{\text{MultiResUNet}}$; number of classes $C = 6$

Ensure: Discrete prediction $\hat{y} \in \{0, \dots, C-1\}^{H \times W}$ and, optionally, ensemble probability map $\bar{p} \in \mathbb{R}^{C \times H \times W}$

Normalization and conversion to tensor

Initialize the set of loaded models $\mathcal{M} \leftarrow \emptyset$.

For each architecture $a \in \{\text{UNet++}, \text{CMUNeXt}, \text{MultiResUNet}\}$ do

Build model m_a with 1 input channel, $C = 6$ classes, and *EfficientNet-B4* encoder

Load the trained model

Load the weights into m_a with non-strict matching

Insert m_a into $\mathcal{M}$

End for

Store the original dimensions (H, W) of I

$I_p \leftarrow \text{PAD}(I)$ // *reflective padding to the borders so that its height and width become multiples of 32*

$X \leftarrow \text{NORMALIZE}(I_p)$ // *normalization, and conversion to tensor*

Initialize the list of model probabilities $\wp \leftarrow []$

For each model $m \in \mathcal{M}$ do

Run inference without gradient computation

Obtain the final high-resolution head logits $z_m \leftarrow m(X)$

Compute class probabilities: $p_m \leftarrow \text{SOFTMAX}(z_m)$

Remove the batch dimension, obtaining $p_m \in \mathbb{R}^{C \times H_p \times W_p}$

Crop the padding: $p_m \leftarrow p_m[:, 0:H, 0:W]$

Renormalize p_m per pixel so that $\sum_c p_m(c, i, j) = 1$,

$$p_m(c, i, j) \leftarrow \frac{p_m(c, i, j)}{\sum_{k=0}^{C-1} p_m(k, i, j) + 10^{-6}}$$

Add p_m to the list $\wp$

End for

Initialize $\bar{p} \leftarrow 0$ with the same dimensions as p_m

For each model $m \in \mathcal{M}$ do

Accumulate the weighted ensemble: $\bar{p} \leftarrow \bar{p} + w_m p_m$

End for

Normalize the ensemble:

$$\bar{p} \leftarrow \frac{\bar{p}}{\sum_{m \in \mathcal{M}} w_m}$$

Generate the discrete class map: $\hat{y}(i, j) \leftarrow \arg \max_c \bar{p}(c, i, j)$

If the dimensions of $\hat{y}$ differ from (H, W) then

Resize $\hat{y}$ to (H, W) using nearest-neighbor interpolation

End if

Return $\hat{y}$ and, optionally, $\bar{p}$

Mean class accuracy is the unweighted mean of the class accuracies (Equation 7):

$$\text{MCA} = \frac{1}{n_c} \sum_i \text{CA}_i \quad (7)$$

For class i , $\text{IoU}_i = |F_i \cap G_i| / (|F_i| + |G_i| - |F_i \cap G_i|)$. The reported mIoU is the unweighted mean of these class-wise values (Equation 8):

$$\text{mIoU} = \frac{1}{n_c} \sum_i \frac{|F_i \cap G_i|}{|F_i| + |G_i| - |F_i \cap G_i|} \quad (8)$$

Frequency-weighted intersection over union weights each class IoU by that class's relative frequency in the ground truth (Equation 9):

$$\text{FWIoU} = \frac{1}{\sum_k |G_k|} \sum_i |G_i| \frac{|F_i \cap G_i|}{|F_i| + |G_i| - |F_i \cap G_i|} \quad (9)$$

2.6. Ensemble strategy

Let $f_1, \dots, f_K$ denote K trained multiclass segmentation models. For an input section I , model k produces logits $z_k \in \mathbb{R}^{C \times H \times W}$, which are converted to class probabilities p_k by the channel-wise softmax function (Equation 10):

$$p_k = \text{SOFTMAX}(z_k) \quad (10)$$

The discrete prediction $\hat{y}$ was obtained from the maximum aggregated class probability; no binary threshold was used. For weighted averaging, each model was assigned a non-negative coefficient w_k , and the K coefficients sum to one. The aggregated probability map $\bar{p}$ is (Equation 11):

$$\bar{p}(I) = \sum_{k=1}^K w_k p_k(I) \quad (11)$$

The discrete multiclass prediction $\hat{y}$ is then obtained by (Equation 12):

$$\hat{y}(i, j) = \arg \max_c \bar{p}(c, i, j) \quad (12)$$

Equal weights yield the simple arithmetic-mean ensemble evaluated in this study. Four configurations were tested: the three pairwise averages and the average of all three models.

2.7. Agreement and divergence analysis

For M segmentation models, let $\hat{y}_m(p)$ be the class predicted by model m at pixel $p \in \Omega$. Agreement among all models is defined as Equation 13:

$$\text{Agreement}_{\text{all}} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \mathbb{1}[\hat{y}_1(p) = \hat{y}_2(p) = \dots = \hat{y}_M(p)] \quad (13)$$

where $\mathbb{1}[\cdot]$ is an indicator equal to 1 when all models assign

the same class to p and 0 otherwise. Pairwise agreement between models i and j is (Equation 14):

$$\text{Agreement}_{i,j} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \mathbb{1}[\hat{y}_i(p) = \hat{y}_j(p)] \quad (14)$$

The mean pairwise agreement over all distinct model pairs is (Equation 15):

$$\text{Agreement}_{\text{mean}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \text{Agreement}_{i,j} \quad (15)$$

where $\mathcal{P} = \{(i,j) \mid 1 \leq i < j \leq M\}$, $|\mathcal{P}| = M(M-1)/2$.

For probabilistic outputs, let $P_i(p)$ and $P_j(p)$ be the softmax probability vectors predicted by models i and j at pixel p . Their Jensen–Shannon divergence²⁴ is (Equation 16):

$$\text{JSD}_{i,j}(p) = \frac{1}{2} D_{\text{KL}}(P_i(p) \parallel M_{i,j}(p)) + \frac{1}{2} D_{\text{KL}}(P_j(p) \parallel M_{i,j}(p)) \quad (16)$$

where $M_{i,j}(p) = \frac{1}{2}(P_i(p) + P_j(p))$ and DKL is the Kullback–Leibler divergence.²⁵ The image-level Jensen–Shannon divergence is obtained by averaging over pixels (Equation 17):

$$\text{JSD}_{i,j} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \text{JSD}_{i,j}(p) \quad (17)$$

Finally, the mean Jensen–Shannon divergence across all model pairs is (Equation 18):

$$\text{JSD}_{\text{mean}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \text{JSD}_{i,j} \quad (18)$$

where $\mathcal{P} = \{(i,j) \mid 1 \leq i < j \leq M\}$, $|\mathcal{P}| = M(M-1)/2$.

3. Results

All three models converged stably over 200 epochs in the full-augmentation condition. UNet++ reduced its training loss from 0.5266 to 0.0411 and reached a minimum of 0.0392 at epoch 93. MultiResUNet decreased from 1.0026 to 0.0884, with a minimum of 0.0857 at epoch 93. CMUNeXt decreased from 1.3803 to 0.0968 and reached a minimum of 0.0941 at epoch 98. Figure 5 presents the corresponding loss and mIoU trajectories. The augmentation effect is evaluated separately in Section 3.1 and Table 1.

The models used EfficientNet-B4²⁶ as the encoder. Computational measurements were obtained for 512×512 inputs and 200 training epochs on an Intel Core Ultra 9 275HX processor (2.70 GHz), 64 GB RAM, and an NVIDIA GeForce RTX 5070 Laptop GPU with 8 GB memory. UNet++ had the largest computational footprint: 31.87 giga-floating-point operations per second (GFLOPs) and 20.81 million parameters. Its total training time was

Table 1. Single-run ablation of training-time data augmentation.

Configuration	Training processing	PA	MCA	FWIoU	mIoU
UNet++	Normalization only	0.897	0.761	0.818	0.690
UNet++	Full augmentation	0.925	0.862	0.873	0.710
CMUNeXt	Normalization only	0.912	0.804	0.846	0.702
CMUNeXt	Full augmentation	0.930	0.855	0.876	0.744
MultiResUNet	Normalization only	0.900	0.771	0.823	0.691
MultiResUNet	Full augmentation	0.925	0.815	0.866	0.757
Ensemble 1	Normalization only	0.907	0.788	0.835	0.714
Ensemble 1	Full augmentation	0.934	0.869	0.884	0.743
Ensemble 2	Normalization only	0.902	0.773	0.825	0.706
Ensemble 2	Full augmentation	0.929	0.834	0.874	0.742
Ensemble 3	Normalization only	0.910	0.797	0.841	0.710
Ensemble 3	Full augmentation	0.930	0.837	0.875	0.762
Ensemble 4	Normalization only	0.912	0.796	0.842	0.719
Ensemble 4	Full augmentation	0.934	0.860	0.883	0.766

Note: Each configuration is reported with normalization only and with the full training augmentation pipeline.

Abbreviations: FWIoU: Frequency-weighted intersection over union; MCA: Mean class accuracy; mIoU: Mean intersection over union; PA: Pixel accuracy.

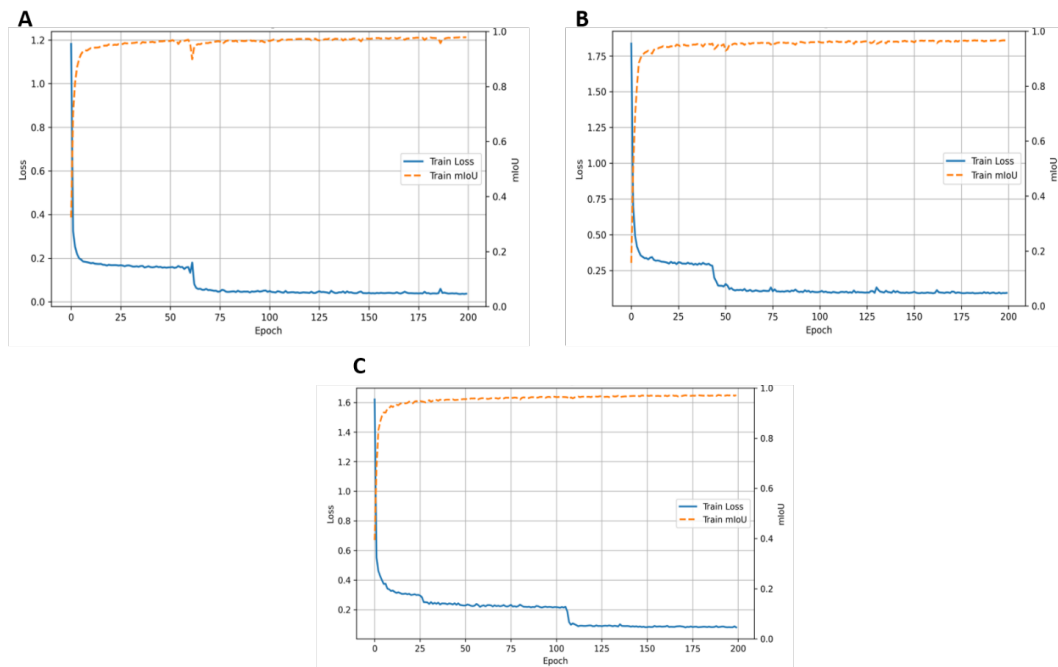


Figure 5. Training performance of the three segmentation models. Loss and mean intersection over union (mIoU) curves are shown over 200 epochs for (i) UNet++, (ii) CMUNeXt, and (iii) MultiResUNet.

3303.04 seconds (55.05 minutes), corresponding to 16.37 ± 1.22 seconds per epoch. CMUNeXt required 13.33 GFLOPs and 18.77 million parameters. This represents approximately 58% fewer floating-point operations than the largest individual model. Training required 2965.36 seconds (49.42 minutes), or 14.71 ± 1.07 seconds per epoch, reducing total training time by approximately 10.2%. MultiResUNet was the least expensive configuration, with 10.79 GFLOPs and 17.75 million parameters. Training required 2846.70 seconds (47.44 minutes), or 14.12 ± 0.30 seconds per epoch, approximately 13.8% faster than UNet++.

These measurements show that CMUNeXt and MultiResUNet reduce computational cost relative to UNet++ while retaining competitive segmentation performance. Future work should measure inference time and peak memory before making deployment-specific claims.

Table 2 reports published F3 results together with the proposed models. No proposed configuration dominates every metric or every facies. Ensemble 4 achieved the highest mIoU among the proposed methods (0.766), while Ensemble 1 showed the highest proposed MCA (0.869) and FWIoU (0.884). Tolstaya and Egorov⁹ remain stronger on the principal global metrics, and several literature methods lead individual classes. Table 3 therefore reports signed differences between Ensemble 4 and each literature result, making both gains and deficits explicit.

The comparison with Tolstaya and Egorov⁹ uses their augmentation-based setting without pseudolabeling. Their method optimizes a single EfficientNet-backed U-Net using geological warping, whereas the present study examines architectural diversity through arithmetic probability averaging. Because published studies differ in partitions, preprocessing, augmentation, and reporting, the deltas in Table 3 indicate contextual differences rather than controlled head-to-head superiority.

3.1. Data-augmentation ablation

To isolate the contribution of training-time augmentation, the three architectures were also trained with every augmentation operator disabled. Only the fixed single-channel normalization ($\mu = 0.449$, $\sigma = 0.226$) was retained. The seven individual and ensemble configurations were evaluated on the same held-out test block, sliced into 200 inline and 601 crossline sections (801 test images in total); equal probability-averaging weights were used for the four ensembles in both conditions. Augmentation improved every global metric for all seven configurations. Across the individual models and ensembles, the absolute gains were 0.018–0.028 for PA, 0.040–0.101 for MCA, 0.030–0.055

for FWIoU, and 0.020–0.066 for mIoU. MultiResUNet showed the largest mIoU increase (0.691 to 0.757; +0.066). For Ensemble 4, augmentation increased PA from 0.912 to 0.934, MCA from 0.796 to 0.860, FWIoU from 0.842 to 0.883, and mIoU from 0.719 to 0.766. The consistent direction of change supports retaining the augmentation pipeline, but the single-run design precludes claims of statistical significance.

Figure 6 compares a representative field-acquired section, its interpretation, the predictions of the three individual models, and Ensemble 4. The figure shows trace and sample indices for the seismic and segmentation profiles. The upper units were consistently recovered, whereas the transition among Rijnland/Chalk, Scruff, and Zechstein exhibited more diffuse probability responses and local overlap. Ensemble 4 reduced some isolated inconsistencies while preserving the principal stratigraphic geometry.

The ensemble members showed strong agreement in their discrete predictions. All three models assigned the same class to 0.9201 of the evaluated pixels, and the mean pairwise agreement was 0.9446. CMUNeXt and MultiResUNet had the highest pairwise agreement (0.9540), followed by UNet++–CMUNeXt (0.9440) and UNet++–MultiResUNet (0.9357). The models are therefore largely coherent but not identical.

At the probability level, the mean pairwise Jensen–Shannon divergence was 0.0409. CMUNeXt and MultiResUNet had the smallest divergence (0.0328), consistent with their high pixel agreement. UNet++ and MultiResUNet had the largest divergence (0.0490), suggesting that this pair contributes the greatest probabilistic diversity. Moderate diversity can benefit averaging near ambiguous facies boundaries, although agreement and divergence alone do not establish a causal ensemble gain.

Figure 7 compares the row-normalized confusion matrices for all four ensemble configurations. The three Upper N. S. units were consistently recognized: diagonal rates ranged from 0.98 to 0.99 for Upper N. S., 0.90 to 0.91 for Middle N. S., and 0.98 to 0.99 for Lower N. S. Rijnland/Chalk is also stable, with diagonal values of 0.80 to 0.81. The main differences occur for Scruff and Zechstein. Scruff accuracy ranged from 0.67 to 0.70, with Ensemble 1 highest at 0.70, whereas Zechstein ranged from 0.64 to 0.84, with Ensemble 1 highest at 0.84 and Ensemble 4 second at 0.79.

Across configurations, Scruff was most often confused with Rijnland/Chalk (0.16 to 0.17) and Lower N. S. (0.09 to 0.13). Zechstein was frequently assigned to Lower N. S. by Ensembles 2 and 3 (0.24 and 0.23); this error decreases

Table 2. Comparison with published results on the Netherlands F3 benchmark

Model	PA	Class accuracy						MCA	FWIoU	mIoU
		Zechstein	Scruff	Rijnland/ Chalk	Lower N. S.	Middle N. S.	Upper N. S.			
Alaudah <i>et al.</i> ¹	0.905	0.602	0.674	0.772	0.941	0.938	0.974	0.817	0.832	–
Trindade & Roisenberg ⁴	0.893	0.69	0.49	0.67	0.96	0.92	0.98	0.785	0.814	–
Reis ⁵	0.912	0.618	0.595	0.737	0.981	0.898	0.976	0.801	0.857	–
Nasim <i>et al.</i> ⁷	0.85	0.84	0.35	0.73	0.94	0.84	0.95	0.78	0.77	0.62
Tolstaya & Egorov ⁹	0.94	0.97	0.96	0.97	0.85	0.76	0.95	0.91	0.9	0.79
Wang <i>et al.</i> ¹⁰	0.907	–	–	–	–	–	–	0.852	0.836	–
De Miranda ¹¹	0.938	0.874	0.782	0.764	0.977	0.93	0.977	0.884	0.893	0.7766
Gutierrez <i>et al.</i> ¹⁷	0.9338	–	–	–	–	–	–	0.8828	–	0.7875
^a UNet++ (f_1)	0.925	0.830	0.710	0.780	0.970	0.907	0.975	0.862	0.873	0.710
^a CMUNeXt (f_2)	0.930	0.802	0.679	0.795	0.986	0.891	0.981	0.855	0.876	0.744
^a MultiResUNet (f_3)	0.925	0.551	0.646	0.813	0.985	0.908	0.987	0.815	0.866	0.757
^a Ensemble 1 (f_1+f_2)/2	0.934	0.841	0.703	0.800	0.982	0.908	0.983	0.869	0.884	0.743
^a Ensemble 2 (f_1+f_3)/2	0.929	0.635	0.682	0.808	0.981	0.912	0.987	0.834	0.874	0.742
^a Ensemble 3 (f_2+f_3)/2	0.930	0.665	0.674	0.808	0.987	0.904	0.985	0.837	0.875	0.762
^a Ensemble 4 ($f_1+f_2+f_3$)/3	0.934	0.785	0.687	0.806	0.985	0.911	0.987	0.860	0.883	0.766

Note: ^aModels evaluated in this study using the full-augmentation condition.

Abbreviations: FWIoU: Frequency-weighted intersection over union; MCA: Mean class accuracy; mIoU: Mean intersection over union; N.S.: North Sea; PA: Pixel accuracy.

Table 3. Signed metric differences between Ensemble 4 and published F3 results (Ensemble 4 minus reference). Positive values favor Ensemble 4; em dashes indicate unavailable metrics.

Reference	Δ PA	Δ MCA	Δ FWIoU	Δ mIoU
Alaudah <i>et al.</i> ¹	+0.029	+0.043	+0.051	–
Trindade & Roisenberg ⁴	+0.041	+0.075	+0.069	–
Reis ⁵	+0.022	+0.059	+0.026	–
Nasim <i>et al.</i> ⁷	+0.084	+0.080	+0.113	+0.146
Tolstaya & Egorov ⁹	–0.006	–0.050	–0.017	–0.024
Wang <i>et al.</i> ¹⁰	+0.027	+0.008	+0.047	–
De Miranda ¹¹	–0.004	–0.024	–0.010	–0.011
Gutierrez <i>et al.</i> ¹⁷	+0.000	–0.023	–	–0.022

Note: These differences are descriptive because the cited studies do not necessarily use identical partitions, preprocessing, augmentation, or model-selection protocols. Positive values favor Ensemble 4.

Abbreviations: FWIoU: Frequency-weighted intersection over union; MCA: Mean class accuracy; mIoU: Mean intersection over union; PA: Pixel accuracy.

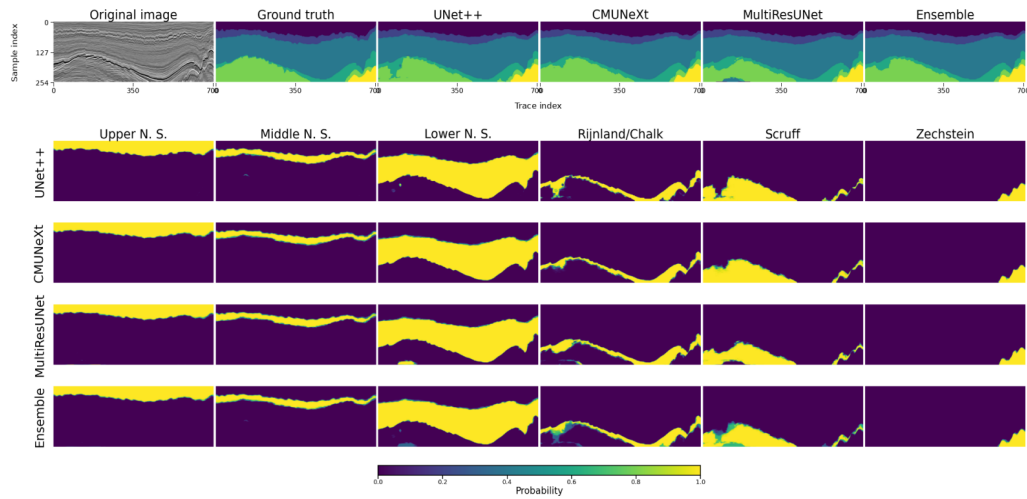


Figure 6. Representative segmentation outputs and class-probability maps for UNet++, CMUNeXt, MultiResUNet, and Ensemble 4. Coordinate axes report trace and sample indices. Abbreviation: N. S.: North Sea.

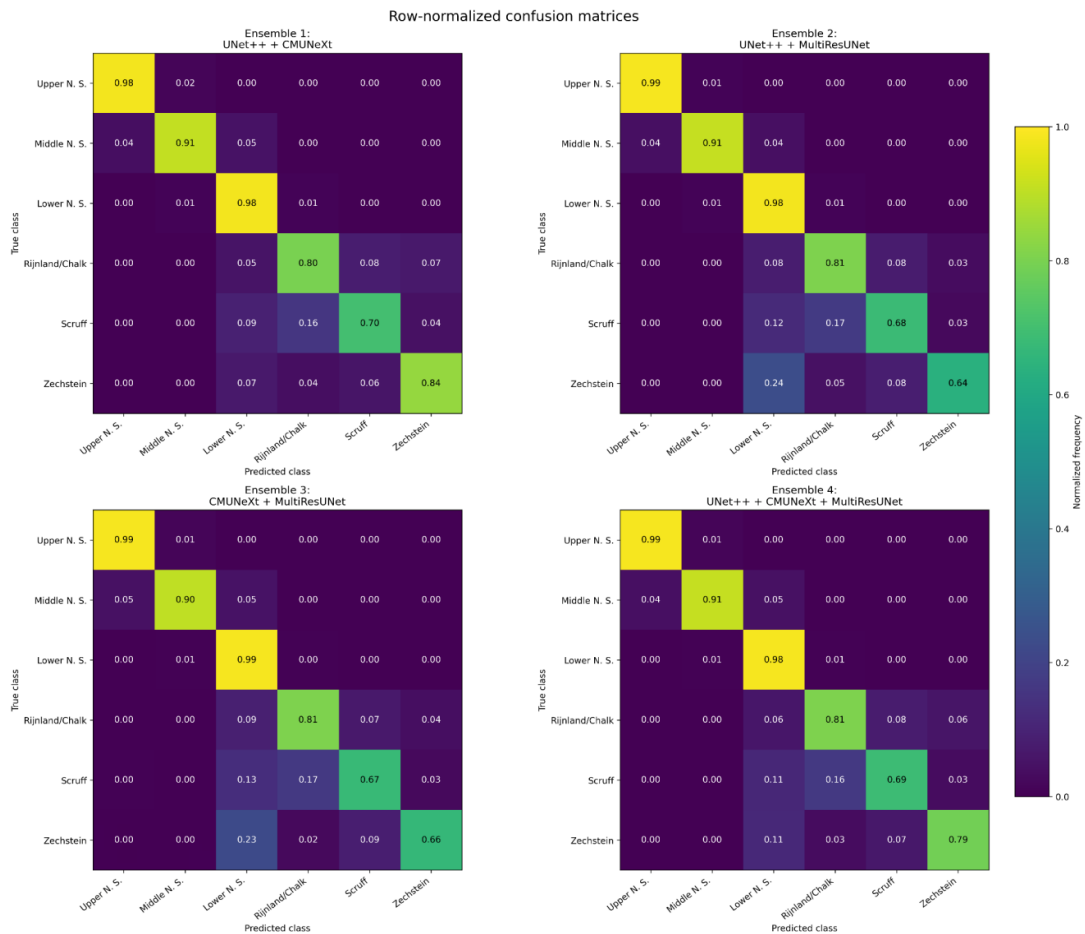


Figure 7. Row-normalized confusion matrices for Ensembles 1–4. Rows represent ground-truth classes and columns represent predicted classes; values are proportions rounded to two decimal places.

to 0.11 for Ensemble 4 and 0.07 for Ensemble 1. Ensemble 3 provided the strongest Lower N. S. diagonal (0.99), while Ensembles 2 to 4 gave the strongest Rijnland/Chalk diagonal (0.81). Thus, the four matrices corroborated the class-wise trade-offs reported in Table 2 and showed that adding all three models improved Zechstein stability relative to Ensembles 2 and 3, although Ensemble 1 remained strongest for both Scruff and Zechstein. Because entries are displayed to two decimal places, row totals may differ slightly from one because of rounding.

4. Discussion

Ensemble 4 achieved PA = 0.934, MCA = 0.860, FWIoU = 0.883, and mIoU = 0.766. These values support the central claim that arithmetic averaging of complementary U-Net-derived models can improve the balance of overlap and class-wise performance relative to the individual implementations. The claim is deliberately limited to the evaluated F3 protocol and does not imply universal state-of-the-art performance.

Among the proposed configurations, Ensemble 1 and Ensemble 4 achieved the highest PA (0.934), Ensemble 1 achieved the highest MCA and FWIoU (0.869 and 0.884), and Ensemble 4 achieved the highest mIoU (0.766). The gain from combining all three models over the best individual model is modest in mIoU (+0.009 relative to MultiResUNet), but the ensemble provides a more balanced result across the reported global metrics.

Relative to the literature, Ensemble 4 improves PA, MCA, and FWIoU over Alaudah *et al.*,¹ Trindade and Roisenberg,⁴ Nasim *et al.*,⁷ Reis,⁵ and Wang *et al.*¹⁰ It does not exceed Tolstaya and Egorov⁹ or the strongest reported values from De Miranda¹¹ and Gutierrez *et al.*¹⁷ on all available metrics. Table 3 quantifies these differences and avoids interpreting mixed evidence as categorical superiority.

The class-wise results explain why different ensembles lead to different aggregate metrics. Ensemble 1 performs best among the proposed methods for Zechstein, UNet++ for Scruff, MultiResUNet for Rijnland/Chalk, Ensemble 3 for Lower N. S., and Ensemble 2 for Middle N. S. No single model dominates every facies, supporting the use of complementary decoders. However, the high overall agreement also limits the magnitude of gains available from simple averaging.

The ablation indicates that augmentation is a material component of the reported performance rather than a neutral preprocessing choice. Improvements were observed not only for the individual networks but also after probability averaging, including a +0.047 mIoU change for

Ensemble 4. The especially large MCA gain for UNet++ (+0.101) suggests that the training transformations improve recall balance across facies, although repeated training runs are required to determine the stability of this effect.

The practical advantage of the proposed approach is simplicity: it requires no learned fusion network and reuses independently trained models. The cost is running multiple networks at inference. The individual timing results suggest that CMUNeXt and MultiResUNet are less expensive than UNet++, but ensemble-level latency and memory remain to be measured.

5. Conclusion

Simple probability averaging of three complementary EfficientNet-B4/U-Net-derived models produced a competitive result on the real F3 benchmark. Ensemble 4 achieved PA = 0.934, MCA = 0.860, FWIoU = 0.883, and mIoU = 0.766, while Ensemble 1 provided the strongest proposed MCA and FWIoU. The single-run ablation showed that full training augmentation improved all four global metrics for every individual model and ensemble; for Ensemble 4, mIoU increased from 0.719 to 0.766. The evidence supports ensemble averaging and the implemented augmentation pipeline as practical baselines, but it does not establish overall superiority to every published method or statistical significance for the ablation. The principal contribution is a controlled demonstration that decoder diversity can improve metric balance without a learned fusion stage.

Future work should address these limitations by validating the method on independent seismic blocks and surveys under different acquisition conditions and geological settings, and by comparing its predictions with expert interpretations. More controlled comparisons with published approaches should employ harmonized data partitions, preprocessing pipelines, and evaluation protocols. The augmentation ablation should also be repeated across multiple random seeds to quantify run-to-run variability and report confidence intervals and statistical significance. In addition, ensemble-level inference latency and peak memory consumption should be measured to characterize the computational overhead of executing two or three networks. Finally, domain-specific deformations, uncertainty estimation, and class-balancing strategies should be investigated, particularly to improve the recognition of Scruff and Zechstein.

Acknowledgments

The authors gratefully acknowledge the technical support provided by UFSC and UDESC.

Funding

This research was funded by the project “4D Integrated Quantitative Characterization of Reservoirs” (FEESC/PETROBRAS/4600671602).

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: All authors

Data curation: Alexandre Gonçalves Silva

Formal analysis: All authors

Funding acquisition: Mauro Roisenberg

Investigation: Alexandre Gonçalves Silva

Methodology: Alexandre Gonçalves Silva

Project administration: Rafael de Santiago

Resources: Mauro Roisenberg, Rafael de Santiago

Software: Alexandre Gonçalves Silva

Supervision: Mauro Roisenberg, Rafael de Santiago

Validation: All authors

Visualization: Alexandre Gonçalves Silva

Writing—original draft: Alexandre Gonçalves Silva

Writing—review & editing: All authors

Availability of data

This study employed a publicly available facies dataset provided by Alaudah *et al.*'s “Facies Classification Benchmark” (<https://doi.org/10.5281/zenodo.3755060>). The code for the ensembles of lightweight U-Net-based models with EfficientNet encoders for semantic seismic data segmentation is available at: <https://codigos.ufsc.br/l3c/multi-label-seismic-data-segmentation>.

References

- Alaudah Y, Michałowicz P, Alfarraj M, AlRegib G. A Machine-learning Benchmark for Facies Classification. *Interpretation*. 2019;7(3):SE175-SE187. doi: 10.1190/int-2018-0249.1
- Zhang Y, Liu Y, Zhang H, Xue H. Seismic Facies Analysis Based on Deep Learning. *IEEE Geosci Remote Sens Lett*. 2020;17(7):1119-1123. doi: 10.1109/LGRS.2019.2941166
- Zhang H, Chen T, Liu Y, Zhang Y, Liu J. Automatic Seismic Facies Interpretation Using Supervised Deep Learning. *Geophysics*. 2021;86(1):IM15-IM33. doi: 10.1190/geo2019-0425.1
- Trindade EA, Roisenberg M. Multi-view 3D Seismic Facies Classifier. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. March 22-March 26, 2021; New York, NY. 2021:1003-1011. doi: 10.1145/3412841.3441976
- Reis NC. Classificação de Fácies Sísmicas Utilizando Multiatributos Sísmicos [Classification of Seismic Facies Using Seismic Multi-Attributes]. Master's Thesis. Pontifícia Universidade Católica do Rio de Janeiro; 2022. [In Portuguese] doi: 10.17771/pucrio.acad.60895
- Kaur H, Pham N, Fomel S, *et al.* A Deep Learning Framework for Seismic Facies Classification. *Interpretation*. 2023;11(1):T107-T116. doi: 10.1190/INT-2022-0048.1
- Nasim MQ, Maiti T, Srivastava A, Singh T, Mei J. Seismic Facies Analysis: A Deep Domain Adaptation Approach. *IEEE Trans Geosci Remote Sens*. 2022;60:1-16. doi: 10.1109/tgrs.2022.3151883
- Mustafa A, AlRegib G. Active Learning with Deep Autoencoders for Seismic Facies Interpretation. *Geophysics*. 2023;88(4):IM77-IM86. doi: 10.1190/geo2022-0353.1
- Tolstaya E, Egorov A. Deep learning for automated seismic facies classification. *Interpretation*. 2022;10(2):SC31-SC40. doi: 10.1190/int-2021-0140.1
- Wang Z, Wang Q, Yang Y, Liu N, Chen Y, Gao J. Seismic Facies Segmentation via a Segformer-Based Specific Encoder-Decoder-Hypercolumns Scheme. *IEEE Trans Geosci Remote Sens*. 2023;61:1-11. doi: 10.1109/TGRS.2023.3244037
- De Miranda DCB. Vision Transformers e Masked Autoencoders para Segmentação de Fácies Sísmicas [Vision Transformers and Masked Autoencoders for Seismic Facies Segmentation]. PhD Thesis. Pontifícia Universidade Católica do Rio De Janeiro; 2023. [In Portuguese] doi: 10.17771/pucrio.acad.65865
- Gao Z, Wang K, Wang Z, Gao J. Optimizing Seismic Facies Classification Through Differentiable Network Architecture Search. *IEEE Trans Geosci Remote Sens*. 2024;62:1-12. doi: 10.1109/TGRS.2024.3357929
- Guo L, Li G, Gong K. Seismic Facies Interpretation Based on Super-Resolution Learning. *IEEE Geosci Remote Sens Lett*. 2025;22:1-5. doi: 10.1109/LGRS.2025.3528958
- Li M, Yan X, Wu Q. A Self-supervised Deep Learning Framework for Seismic Facies Segmentation. *Expert Syst Appl*. 2025;288:128290. doi: 10.1016/j.eswa.2025.128290
- Tian S, Tang Y, Deng F, Luo W, Wang B, Yang S. An Interactive Segmentation-Based Method for Seismic Facies Annotation and Segmentation. *Sci Rep*. 2026;16(1):2353.

- doi: 10.1038/s41598-025-32016-8
16. Canguçu GL, Jorge LMS, Barreto GT, Silva TH, Lima LA, Caldas WS. A Principled Benchmark for Seismic Data Segmentation. *IEEE Trans Geosci Remote Sens.* 2025;63:1-9.
doi: 10.1109/TGRS.2025.3575336
 17. Gutierrez GB, Astudillo CA, Napoli OO, *et al.* On the Performance Evaluation of Deep Learning Models for Seismic Facies Segmentation. *Geophys Prospect.* 2025;73(9).
doi: 10.1111/1365-2478.70104
 18. OpendTect project with 2D & 3D Seismic Data, Acoustic Impedance, Wells, Horizons. Accessed on June 1, 2026. <https://terranubis.com/datainfo/F3-Demo-2023>
 19. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Stoyanov, D., *et al.* *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support.* DLMIA ML-CDS 2018. Lecture Notes in Computer Science, vol. 11045. Cham, Switzerland: Springer, Cham; 2018.
doi: 10.1007/978-3-030-00889-5_1
 20. Tang F, Ding J, Quan Q, Wang L, Ning C, Zhou SK. CMUNeXt: An Efficient Medical Image Segmentation Network Based on Large Kernel and Skip Fusion. *2024 IEEE International Symposium on Biomedical Imaging (ISBI).* May 27-30, 2024; Athens, Greece. 2024:1-5.
doi: 10.1109/ISBI56570.2024.10635609
 21. Ibtehaz N, Rahman MS. MultiResUNet: Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation. *Neural Networks.* 2020;121:74-87.
doi: 10.1016/j.neunet.2019.08.025
 22. Chen J, Lu Y, Yu Q, *et al.* TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv.* Preprint posted online 2021.
doi: 10.48550/arXiv.2102.04306
 23. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Lecture Notes in Computer Science.* Cham, Switzerland: Springer International Publishing; 2015:234-241.
doi: 10.1007/978-3-319-24574-4_28
 24. Lin J. Divergence Measures Based on the Shannon Entropy. *IEEE Trans Inf Theory.* 1991;37(1):145-151.
doi: 10.1109/18.61115
 25. Kullback S, Leibler RA. On Information and Sufficiency. *Ann Math Stat.* 1951;22(1):79-86.
doi: 10.1214/aoms/1177729694
 26. Tan M, Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: *Proceedings of the 36th International Conference on Machine Learning.* PMLR; 2019:6105-6114. Accessed on September 14, 2026. <https://proceedings.mlr.press/v97/tan19a.html>