Granite Journal

The University of Aberdeen Postgraduate Interdisciplinary Journal

Breaking the Boundaries: Interdisciplinary research approaches and methods

Applying Green AI Methods to Digital Rock Technology Workflows

Ciprian-Theodor Panaitescu, Kejian Wu, Yukie Tanino, Andrew Starkey (School of Engineering, University of Aberdeen)

Abstract: Digital Rock Technology comprises a set of image-based modelling techniques that analyse rocks at the pore level, gaining insight essential in industries including geological carbon dioxide sequestration, underground hydrogen storage, oil and gas, or geothermal energy. Pore Network Modelling simulates flow through a network of pores and throats representing a rock sample's void space, which results in improved efficiency and scalability compared to conventional simulation methods. We propose using semantic segmentation to identify the location and geometry of complex features such as fractures and vugs. Convolutional Neural Networks, a form of deep learning technology, can assist in automating the time-consuming process of handling extensive X-ray micro-computed tomography (micro-CT) data.

However, a solely accuracy-focused approach neglects the importance of efficiency and interpretability. This study explores the effect of deep-learning architecture choices on micro-CT semantic segmentation performance, considering computational resource efficiency and carbon footprint under the Green AI principles. The proposed novel objective function, incorporating topology and curvature information via Minkowski functionals, shows a 4% improvement in the accuracy of pre-trained models, and the study presents a strategy for selecting the optimal configuration. To address the lack of

available labelled datasets, an algorithm was developed for generating semi-synthetic image-label pairs using features from publicly available data.

Keywords: Digital Rock Technology, Computational Porous Media, Machine Learning, Deep Learning, Green AI, Image Segmentation

1 Introduction

Digital Rock Technology (DRT) uses high-resolution imaging, such as microcomputed X-ray tomography (micro-CT), and numerical simulations to study rocks at the microscale (Japperi *et al.*, 2022). DRT is crucial in geoscience and engineering applications like hydrocarbon exploitation, carbon capture and storage, geothermal energy, and underground hydrogen storage, as it enhances the understanding of rock properties and predictions of fluid flow (Japperi *et al.*, 2023). The complex pore space of real rock, for example, is often represented by a network of large pores of simple geometries (pore bodies) and straight constrictions that connect them (pore throats). Pore network modelling (PNM), a key component of DRT, simulates fluid flow in porous media through such pore networks, providing efficiency and simulation speed gains of up to two orders of magnitude. (Xiong et al. 2016, Rabbani and Babei 2019, Zhao et al. 2020).

Fractured and vuggy carbonates are challenging formations to analyse, even when using advanced imaging techniques. Therefore, they are suited to analysis using PNM because direct simulation, like finite element methods or Lattice Boltzmann, incurs high computational costs due to the multi-scale domain and complex features (Wang *et al.*, 2022; Xiong *et al.*, 2016).

Wang *et al.* (2022) identify the complex pore structure, fracture-vug interconnectivity, heterogeneity and lack of standard methodologies as main obstacles in processing images of this kind. Unlike the existing literature, our new method focuses on performing semantic segmentation of the three classes of interest (pores, vugs and fractures). This entails partitioning images with varied geometries into multiple meaningful categories by classifying the pixels belonging to each feature, even when these features intersect. To achieve this, we use a diverse database of rock samples with

varied geometries. In addition, we propose a new algorithm for creating semi-synthetic label images, hence addressing the need for an extensive benchmark dataset.

All DRT methods involve imaging and modelling (Berg et al., 2017), but for the application of Multi-feature Pore Network Modelling (MPNM), a new intermediate step is necessary: that of multiclass semantic segmentation of the porous medium. This entails obtaining the geometry and distribution of each feature rather than the conventional solid-void binarisation. This new stage bridges the image acquisition from rock samples and the DRT network construction and simulation. The workflow is illustrated in Fig. 1.

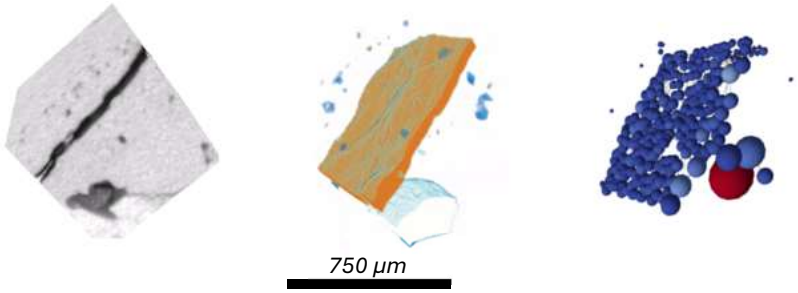

Figure 1: Illustration of the overall process: micro-CT fractured carbonate rock of size 300x300x300 extent and 2.5 µm resolution (left), segmentation (centre) followed by MPNM (right).

Artificial intelligence (AI)-based image segmentation, mainly using deep learning, has become a crucial tool in computer vision applications, ranging from medical imaging to remote sensing. However, as micro-CT imaging capability has improved to capture higher resolutions, the data volume produced is vast, prohibitively so for manual labelling. Furthermore, our particular semantic segmentation case also has additional challenges. Firstly, the grey-level values of the micro-CT Images may be very similar between the three classes of interest (pores, fractures, and vugs), making conventional thresholding techniques ineffective. Secondly, fracture and vug features may also appear at various scales, making the simple size-based analysis inefficient. This has, therefore, led to an increased reliance on statistical approaches, deep learning in particular. An example is convolutional neural networks (CNN), which have achieved highly accurate results in tackling challenging problems like the current ones (Kazak *et* al., 2021; Niu *et al.*, 2020).

CNNs depend on two interconnected pathways: the encoder, which begins with the original image and represents it in a latent space, and the decoder, which takes the latent representation and transforms it into the segmentation output. This architecture

offers significant flexibility to deep-learning models, allowing them to autonomously generate numerous features. However, interpreting these deep-learning layers can be challenging.

2 Convolutional Neural Network Methods

CNN are a versatile group of algorithms, effective in many contexts, including image segmentation for biomedical, engineering and geoscience tasks (Yeung *et al.* 2022; Wang and Zai 2023; Das *et al.* 2022; Malik *et al.* 2021; Yun *et al.* 2019), but the effect of different design decisions on the outcome varies depending on the application, and there is no agreement on the combined significance of these choices (Rex *et al.*, 2022). Therefore, the performance and resource efficiency of several advanced CNN segmentation algorithms is assessed, while considering the effect of design choices on effectiveness and segmentation frugality.

Unlike other segmentation methods, which usually rely on user-defined filters before running the algorithm, as used in Panaitescu *et al.* (2023) for a similar application, CNN automatically learns this input-label mapping based on a high volume of examples. It generally depends on an encoder-decoder method where the former successively extracts information and represents the image in a latent space. At the same time, the latter starts with the latent representation and produces the desired segmentation result.

The general model selection will be split into multiple factors of interest, enumerated below and treated in more detail in the following sections. Firstly, the network size and depth will be considered. A flow chart of the overall analysis and model selection strategy is available in Appendix 7.1. Secondly, the overall network architecture will explore three advanced options typical for biological and geological micro-CT segmentation: the U-Net, Link-Net and Feature Pyramid Network (FPN). Design options for backbones inspired by ResNet, EfficientNet and the VGG (proposed by the Visual Geometry Group) will also be considered. Further, we will discuss the impact of pre-trained models. Pre-training (also referred to as transfer learning) entails using models trained for classification tasks on an entirely different set of natural

images. The pre-training goal is to provide a more robust and informative model initialisation, effectively leveraging previous resources invested in training similar algorithms. The pre-trained model is then fine-tuned on a smaller dataset for the specific required task (Han *et al.*, 2021). Finally, two options will be considered for the objective function choice, also known as loss functions. The first option is the Focal Loss (FL) - proven in the literature to be highly effective for unbalanced datasets (where the numbers of examples in each class are highly different). The second option is a novel hybrid extension of FL that includes curvature and topology via the Minkowski functionals.

The overall architecture relies, at every level, on layers of weight parameters, which are continuously adjusted using the back-propagation technique based on training examples. The training methodology used is also essential as the dataset is split into three subsets: training, testing, and validation datasets. The testing dataset is also used when adjusting hyperparameters and selecting the best-performing model. At the same time, the last data subset aims to provide an unbiased estimate by any previous choice or test.

If a model is too complex relative to the size of the training data, it can overfit: it performs well on the training data but poorly on new data. Therefore, monitoring the training and validation losses during the training process is performed to detect overfitting (Fig. 2).

Furthermore, early stopping can help prevent overfitting by halting the training process when the validation loss increases. In this research, we have adopted this best practice and stopped the training process at the first plateau of the process, considering a patience parameter of five epochs (complete passes of the entire dataset through the algorithm).

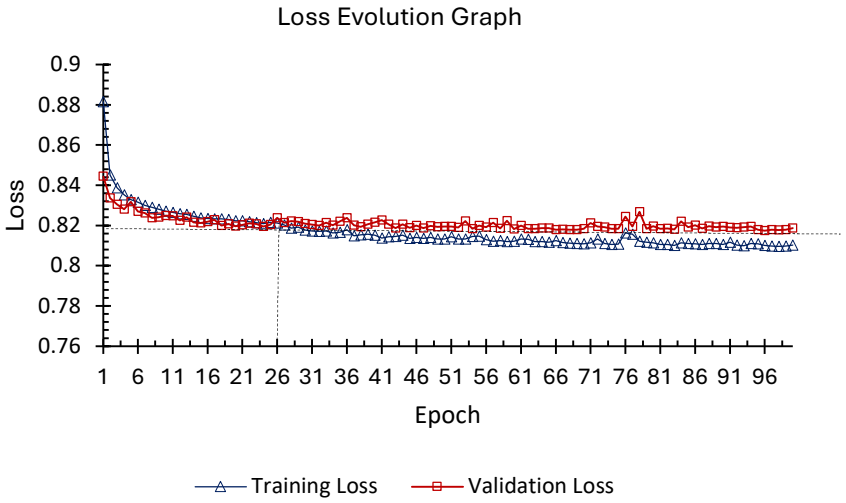

Figure 2: Typical classification loss (or error) evolution profiles on training and validation datasets: after a certain point (indicated by the dotted line), the algorithm can start overfitting, improving the training metric by memorisation and decreasing performance on un-seen, generalisable data.

2.1 Architecture Design

The U-Net, shown in Fig. 3, is the most popular deep learning architecture used for micro-CT image labelling, created initially with medical micro-CT in mind (Ronneberger *et al.*, 2015). It has an architecture similar to an auto-encoder with an encoding path that decreases the size of the latent layers whilst increasing their number and a decoding path that achieves the opposite, hence synthesising and reconstructing images of the same shape as the input, which has been transformed inside the latent space. In addition, information from previous layers is concatenated (or added, as in the case of the Link-net variation, as first proposed by Chaurasia and Culurciello in 2017) from the encoder path to the decoder layers of the same size.

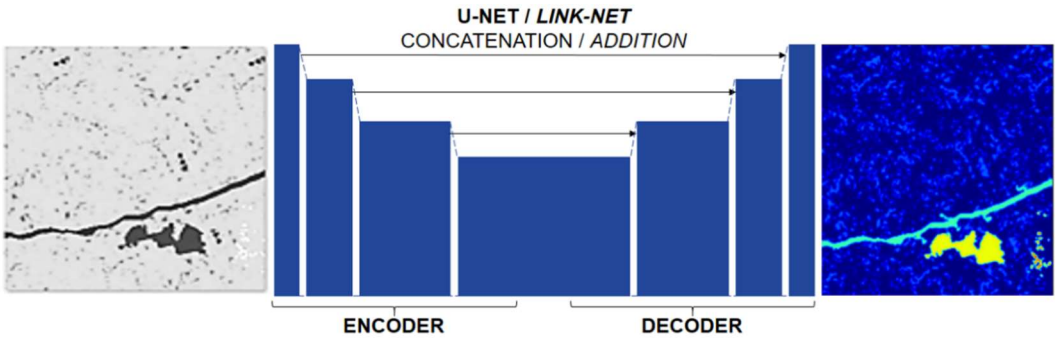

Figure 3: The U-Net and Link-Net Design showing the encoder-decoder path and the cross-connections. The input image is a typical greyscale view (256 levels), and the output has only five levels: matrix, pores, fracture, vugs, and secondary mineralogy. The image is of a typical vuggy fractured carbonate reservoir rock of 256 by 256 extent and 2.5 μm resolution.

The U-Net has seen continuous use since its proposal, including micro-CT segmentation, property prediction of porous media, denoising, and others (Wand and Za, 2023; Liu *et al.*, 2020). The Link-Net has also been used continuously, including geological segmentation examples by authors such as Malik (2022) or Das *et al.* (2022).

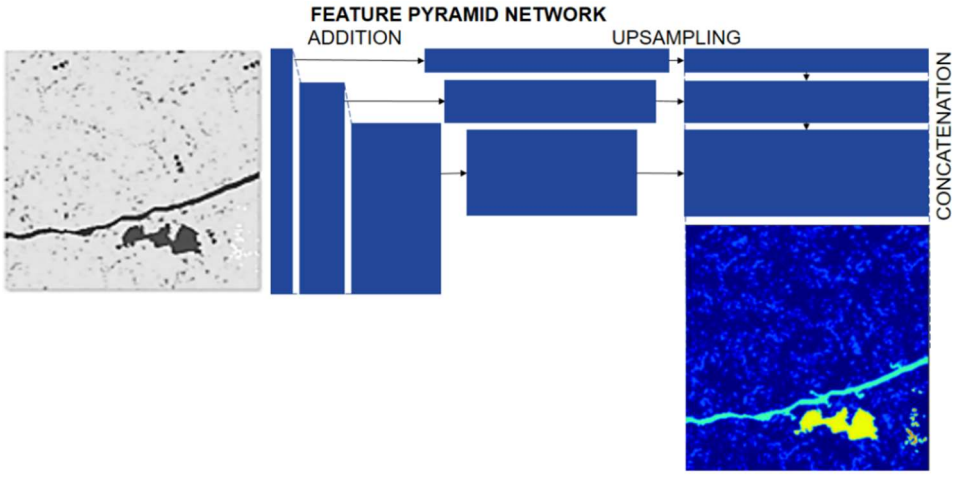

Figure 4: Feature Pyramid Network (FPN) architecture schematic, showing images of rock of 256 by 256 extent and 2.5 μm resolution.

The last architecture considered is the FPN, first proposed for object detection by Lin *et al.* (2017). Like the previous two architectures, it is commonly used in segmenting biomedical micro-CT but has seen comparatively less usage in geoscience applications. One example of its use in the latter fields includes its application to the segmentation of seismic profiles (Li *et al.*, 2020).

The architecture uses a bottom-up and top-down pathway to automatically generate a feature pyramid, not predefined by the user. The feature pyramid consists of several thousand representations with the same number of pixels but represents the image at different scales. It is thus able to learn cross-scale abstractions more directly. The feature pyramid replaces the encoder-decoder component of U-Net and Link-Net, as shown in Fig. 4.

2.2 Backbone Design

This research defines the backbone as the potentially pre-trained set of connected convolutional neural network layers that do not include the classification head. It efficiently extracts data representations via feature maps. The feature maps are

subsequently transmitted to the segmentation head, consisting of convolutional layers, up-sampling layers, and other specialised layers, aiming to generate the ultimate segmentation output. In the current research, three backbones are analysed (VGG, ResNet and EfficientNet); they were first proposed by Simonyan and Zisserman (2014), He *et al.* (2016), and Tan and Le (2019), respectively. The Imagenet pre-trained weights for all three are publicly available.

VGG and ResNet focus on small-size convolution operations, hence emphasising short-range interactions (though by using depths, longer-range correlation can also be learned). In addition, ResNet introduces the concept of residual connection, illustrated in Fig. 5, which confers additional learning flexibility to CNN.

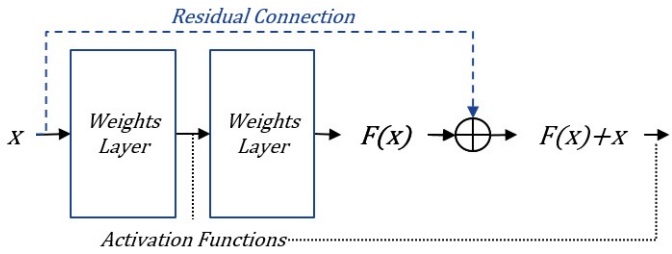

Figure 5: Residual connection concept

It also helps alleviate the common 'vanishing gradient' issue, improving the training process. Finally, EfficientNet uses a compound model scaling optimised using grid search. This preliminary filter selection concept allows it to achieve state-of-the-art accuracy on many tasks with a shallower design. The compound scaling feature also allows EfficientNet to use convolutional filters of varied aspect ratios.

2.3 Objective Function Design

The objective function computes the error incurred during prediction and is used to adjust the network weights. Several studies (Yeung *et al.*, 2022; Ma *et al.*, 2021; Lin *et al.*, 2017; Yun *et al.*, 2019) have shown that objective function design can significantly impact the model's ability. Here we use the Focal Loss (FL), shown in Eq. 1, and a novel loss function we named the Minkowski Difference Loss (MDL), whose mathematical formulation is shown in Eq. 2-5. FL is an objective function especially suitable for unbalanced datasets, like the one presented in these examples. It is a dynamically

scaled version of the cross entropy metric, using the following formula: γ is the focusing parameter, and p_t is the class probability. γ reduces the loss for well-classified cases, focusing more on the other classes (Lin *et al.*, 2017)

$$FL(p_t) = -(1-p_t)^{\gamma}\, log(p_t) \quad (1)$$

Porous media research has shown that the topography and area are crucial in characterising rock samples, traditionally using measures such as the Euler Number or the Betti Numbers (Blunt, 2017). Recently, even more focus has been placed on the ability to use the Minkowski functionals to characterise porous media (for a 2D image, the first three, denoted as M0, M1, and M2, measuring surface area, curvature and topology, shown mathematically in Eq. 2-6, respectively). In addition, new research has shown their ability to characterise rocks (Armstrong et al., 2018; Nair et al., 2021). However, these parameters are not used in general computer vision applications, but they inspired our current approaches. Therefore they have been included to explicitly focus on connectivity and treat it as a learning objective.

The definitions apply to a smooth geometry (X) with a boundary δX. Legand *et al.* (2011) provided a theoretical background for computing the Minkowski function based on digital images. The theory and the mathematical formulation of Eq. 2-4 come from Boelnes and Tchelpi (2021). Therefore we propose a new metric (Eq. 5) that adds a new factor to the Focal Loss that incorporates these terms, hence introducing metrics with enhanced physical relevance compared to exclusively image-based measures. The MDL has the following formula (Eq. 6), where the *M* relative errors, ds is an area unit element, dc is a circumference unit element, and *R* is the local curvature radius.

$$M_0(X) = \int_X ds \quad (2)$$

$$M_1(X) = \frac{1}{2\pi}\int_{\delta X} dc \quad (3)$$

$$M_2(X) = \frac{1}{2\pi^2}\int_{\delta X} \frac{1}{R}\, dc \quad (4)$$

$$M_{j,error} = \frac{M_j(X_{ground\ truth}) - M_j(X_{predicted})}{M_j(X_{ground\ tr}\quad)},\ j \in \{0,1,2\} \quad (5)$$

$$MDL = FL + M_{0,error} + M_{1,error} + M_{2,error} \quad (6)$$

3 Materials and Methods

3.1 Hardware and Software Requirements

Algorithm tuning is performed using cloud high-performance computing resources to allow fast and reliable resource allocation on a global scale and leverage new processor technology (high-performance Graphics and Tensor Processing Units (GPU and TPU) available in the Google platform. The resource use will be benchmarked on a final training performed on a local computer with Intel® Core™i9-10885H CPU and eight cores. The CNNs were implemented in Python, using the TensorFlow library and the Segmentation Models Library (Iakubovskii, 2019) as a consistent starting point. The experimental design and factor analysis were performed using the JMP statistical software. Including Minkowski functionals in the objective function leverages an implementation by Boelnes and Tchelpi (2021).

3.2 Semi-synthetic Image Synthesis and Generation Algorithm

Due to the need for a much larger dataset than readily available, this project created a semi-synthetic training dataset using fractures from samples that can be confidently and automatically segmented. Databases of 2D 256x256 pixels segmented features were created separately from a total of 10 separate images. The advantage of this approach is that each feature is correctly and fully labelled, and they can be reassembled to create fully labelled combined training datasets that emulate samples with pores, vugs, and macro-fractures. The re-combination algorithm pseudocode is presented below; the original dataset samples for the semi-synthetic training images are shown in Fig. 1. Once a database of appropriate semi-synthetic labelled images has been created, as specified above, enough data should be available for a deep learning algorithm to be applied (typically at least 3000-5000 images, in our case creating a total of 3600). The algorithm's description can be found in Appendix 7.2.

3.3 Green Artificial Intelligence

Green AI evaluates algorithms holistically based on accuracy, efficiency, and explainability. It optimises AI algorithm design with a balanced focus on resource use and classification performance. Efficiency analysis traditionally considers memory and computational speed, but we also examine energy consumption and environmental impact, such as carbon footprint, as formulated in Anthony *et al.* (2020) and Budennyy *et al.* (2023).

Recent studies reveal that training large AI models can have a significant impact: a single training of the largest current AI models emits approximately 300 tons of carbon dioxide Dhar (2020). Furthermore, the current trend in model size shows a 300,000 times increase over just the six years before 2019, as noted by Schwartz et al. (2019). This trend appears linear over the six years monitored; however, the performance uplift provided by the larger models is not linear, yielding diminishing returns after a certain point. The models investigated in this article are smaller but still large enough (up to 7 million parameters) to account for significant emissions. Moreover, some model selection and design adjustments can disproportionately impact resource use.

Computational and environmental resource frugality were measured using the method by Anthony *et al.* (2020), with the CO_2 footprint quoted for Aberdeen, UK, whose variable values are seen in Fig. 6. The energy output is proportional to the number of operations (FLOPS) and carbon footprint, with computations performed under the same setup and conditions. The Green AI analysis focuses on performance-resource use optimisation while acknowledging similar explainability challenges across all deep convolutional network architectures.

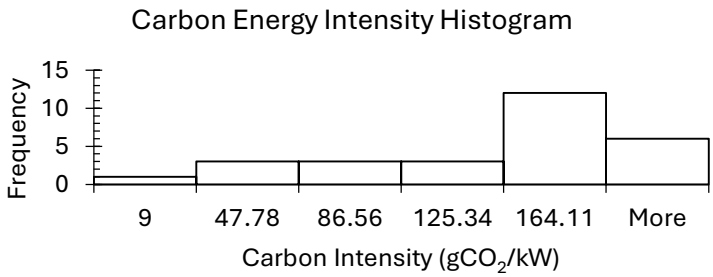

Figure 6: Carbon Intensity registered during the training period presented with a bin size of 38.8 gCO_2/kW. The computed mean and standard deviation are 115.8 gCO_2/kW and 58.5, respectively.

3.4 Design of experiments

The design of a deep learning system is treated somewhat subjectively and experimentally, where there are no clear guidelines for optimal design. Whilst there are some patterns and generally accepted principles, there is currently no consensus on general design methodologies and best practices. Several challenges have hindered the creation of such standards, including the following: (A) A high degree of domain specificity. The best approaches in some areas of computer vision can perform sub-optimally in others. However, some successful approaches can be adjusted for application to problems with similar characteristics such as some tomography analysis techniques are often successfully shared between soil science, geoscience and biomedical science. (B) The high complexity and probabilistic nature of algorithms make it difficult for humans to understand the decision-making process and predict the effect of changes in all parameters. (C) The tendency to over-fit when trained on small datasets makes data curation and acquisition a critical step for the successful application of a system. (D) Finally, the number of hyper-parameter present in models and the diversity of hyper-parameter tuning strategies, some of which emphasise search speed in the hyper-parameter space over absolute optimisation and this makes the comparison between different studies more difficult.

To address these challenges, this study will employ a Design of Experiments approach in order to determine the relative importance of four factors of interest (overall model architecture, residual block design, transfer learning, and objective function choice), having three, three, two and two levels, respectively. In order to systematically and efficiently obtain insight, a resolution IV fractional factorial design was proposed. The optimal design planning ensures orthogonality allowing the investigation of effects and first-order interactions without confounding, which is the primary interest of this study, in only twenty runs. Further confounding only occurring second and higher-order interactions). This is particularly important in maintaining high resource efficiency, as each training run takes several days to complete. The design can be found in Appendix 7.3.

The factors of interest for the current research are transfer learning, described as where the weights have been initialised using the values provided by other researchers investigating different classification problems using natural images from the Imagenet database, the overall network design (U-Net, Link-Net, FPN), residual block design (ResNet, EfficientNet, Inception), and objective function (Focal Loss, and Minkowski Functional Difference).

4 Results and Analysis

The analysis will consider the model performance, measured primarily by the Intersection-Over-Union (IOU) and F1 metrics (equally weighting the reciprocal of type 1 and type 2 errors to form a balanced metric). Precision and Recall (or type 1 and 2 errors) are also quoted for each experiment; however, the F1 score offers a more balanced view, combining both error types. Finally, IOU (or Jaccard Score) overlaps the prediction and ground truth, measuring the relative ratio of their intersectional area divided by the entire estimated area.

4.1 Multi-factor Analysis of Segmentation Performance

In general, larger deep learning models perform better than shallower versions. Further, this article considers the impact of model depth and the number of parameters. Given the objective of utilising pre-trained models as a focal point, the model's design had to align with the available pre-trained configurations. Consequently, achieving a perfectly constant number of model parameters for our initial analysis proved challenging, even though the aim was to make the selection as similar as possible. To counteract this, this research checked the dependence of precision accuracy on the parameter plane, and, as shown in Fig. 7, for our selection of parameters, the relatively small variation in network size has a negligible effect.

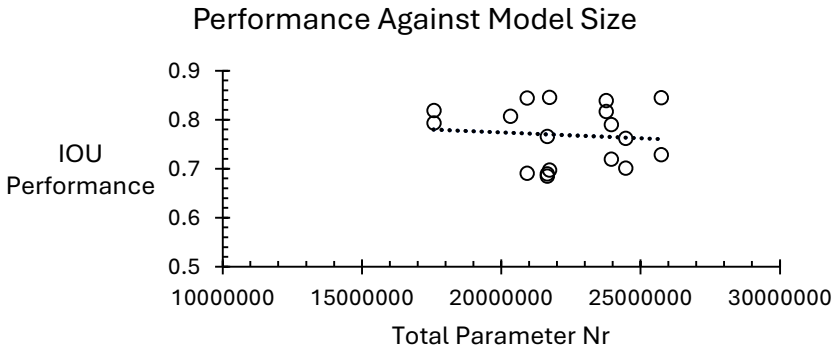

Figure 7: Performance against model size for our experimental design, showing the lack of expected size performance proportionality in our selection, hence blocking this effect.

The relative importance of each factor in the design experiment has been obtained by linear modelling, firstly analysing the IOU measure followed by the F1 analysis. The top 5 factors for IOU identified, presented in order of importance, were: Backbone and transfer learning interaction, Backbone and Objective functions interaction (for VGG), followed by the Backbone selection, and the Transfer Learning decision. The importance of the factor interactions is presented in Fig. 8.

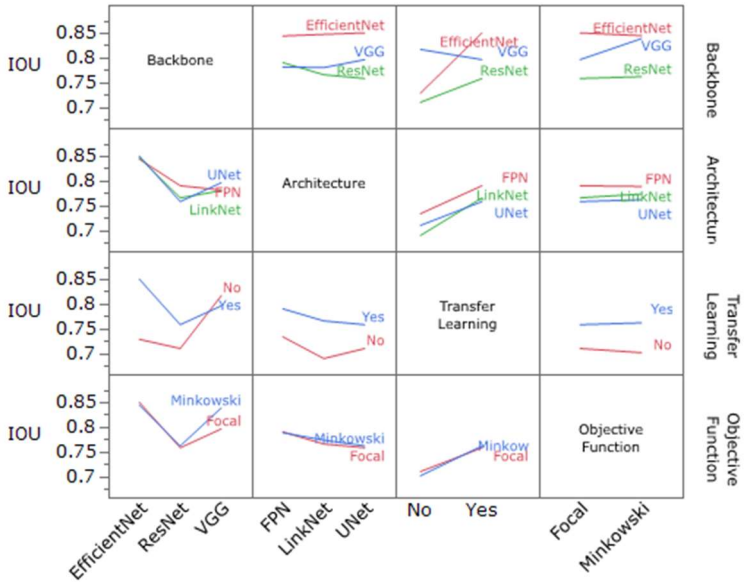

Figure 8: Interaction profiles: cross-over of lines further highlights the effect of second-order interaction terms.

It can be concluded that the backbone selection and usage of transfer learning are the most critical choices on an individual basis, and the new objective function produces, on average relatively reduced impact, though still statistically significant.

However, the new objective function yields significant improvement when combined with other parameter combination - the novel Minkowski function produces a 4% improvement in IOU scores over the Focal Loss when only considering transfer learning experiments, as shown in Fig. 9. The top 3 IOU scores were 0.845, 0.844, and 0.844, respectively. They all used EfficientNet as a residual block backbone and started with pre-trained weights based on Imagenet. Moreover, the top two used the new Minkowski Difference objective function. Each of the three, however, uses a different overall architecture.

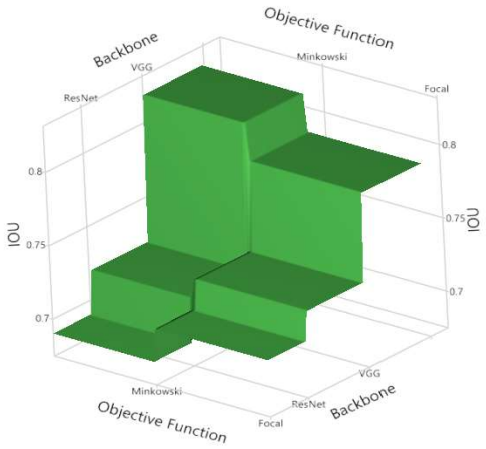

Figure 9: the Effect of backbone and objective function on model performance, showing their complex interaction.

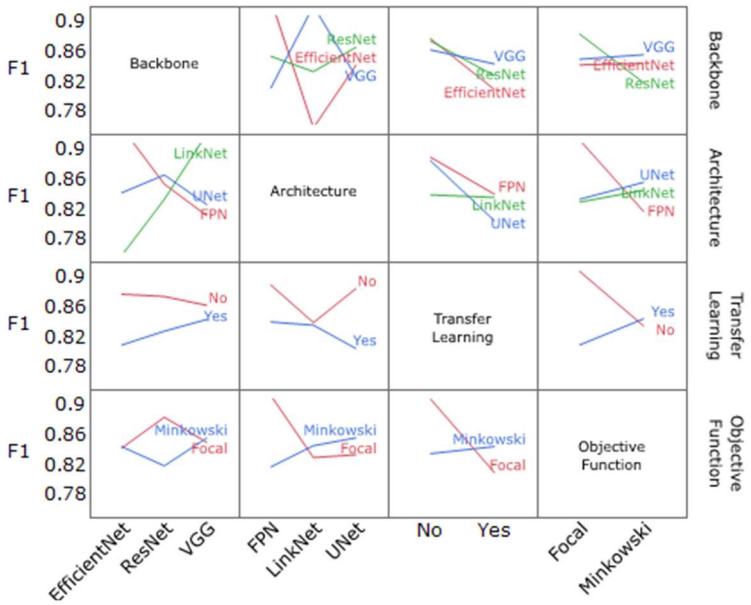

Figure 10: Interaction profiles using F1 as the metric of interest.

Interestingly, one result of this research is the behaviour of the novel Minkowski Difference objective function: it improves performance by at least 2% compared to Focal Loss for all architectures that have started with pre-trained weights, with a particularly distinct improvement in IOU scores. However, for models where the weights were initialised randomly (not leveraging transfer learning), the new objective function decreases performance by as much as 3%. This kind of behaviour is likely due to the improved ability of the new function to model our specific dataset; however, the pre-trained weights help avoid early overfitting.

The analysis was repeated using F1 as the overall performance metric. The general findings are similar to the IOU case, as shown in Fig. 10, though some differences were also observed. The top 5 factors influencing the F1 result identified were: Backbone and Architecture interaction (combination of all Architectures with EfficientNet and VGG), Architecture and Objective functions interaction, followed by the Transfer Learning and Objective function interactions. Interestingly, the most important factors explaining the F1 score do not even include the individual factors and instead are all focused on the selection of multi-factor combinations.

Regarding F1, the top 3 performing architectures again use EfficientNet as the backbone, achieving scores in excess of 90%. All three top scores also used transfer learning, with the two employing the new objective function.

To further increase our understanding of the analysis, we have also split the F1 scores into the Precision and Recall scores (known as type 1 and 2 errors). Here, again, the EfficientNet backbone and the Transfer learning strategy dominate. However, here the new objective function has a significant potential uplift (up to 3.5% with negligible potential lowering accuracy in all combinations). On the other hand, its effect is much more varied on Recall (on average, having 0 effects as it can both aid performance - if combined with Imagenet weights or otherwise hinder it by the same amount).

4.2 Analysis of the Model Size on Performance Accuracy

As suggested throughout the previous analyses, the effect of the model size on the predictive capability has also been examined in more detail. Two skilful models were

investigated: (A) FPN – EfficientNet – Pretrained – Focal loss and (B) UNet – EfficientNet – Pretrained – Minkowski Difference Loss. The most direct and consistent way to change the model size is by adjusting the backbone. 3 backbone sizes were selected from the available pre-trained models: EfficientNet-b1, EfficientNet-b4 and EfficientNet-b7. The alternative method was the configuration examined in the designed experiment above.

The analysis concludes that of the problem and dataset at hand, further increase of the model size only achieves marginal returns. When increased by more than an order of magnitude, the model only achieves at most 1.8% improvement in IOU score and 1.1% improvement in F1 score, therefore having a minor impact than the objective function choice as well as the backbone-architecture selection. The performance metric most affected by the model size is Precision (by up to 2.2%). However, the Recall has the opposite effect, albeit smaller in magnitude. The Performance variation with parameter space size is visualised using bubble charts, which also capture the size effect on resource use which is not directly proportional.

Deeper networks can achieve better accuracy but only have marginal returns, though a more linear relationship can be seen rather than the exponential (straight line on the semi-log scale) (Fig. 11).

From this size-efficiency-performance analysis, it can be concluded that, for this problem, it is more important to focus on other parameters rather than depth alone. Optimising the architecture-backbone-objective function match provides a more significant performance uplift than the increase in size, and performing this analysis is also less expensive regarding computation and environmental resources. While, by definition, increasing the mode size, especially in terms of depth (nr of layers), can never decrease performance given the available dataset is sufficient, after a certain point, these gains plateau: despite significant size increase in the model, the improvement in prediction performance is modest, as highlighted in Fig. 12.

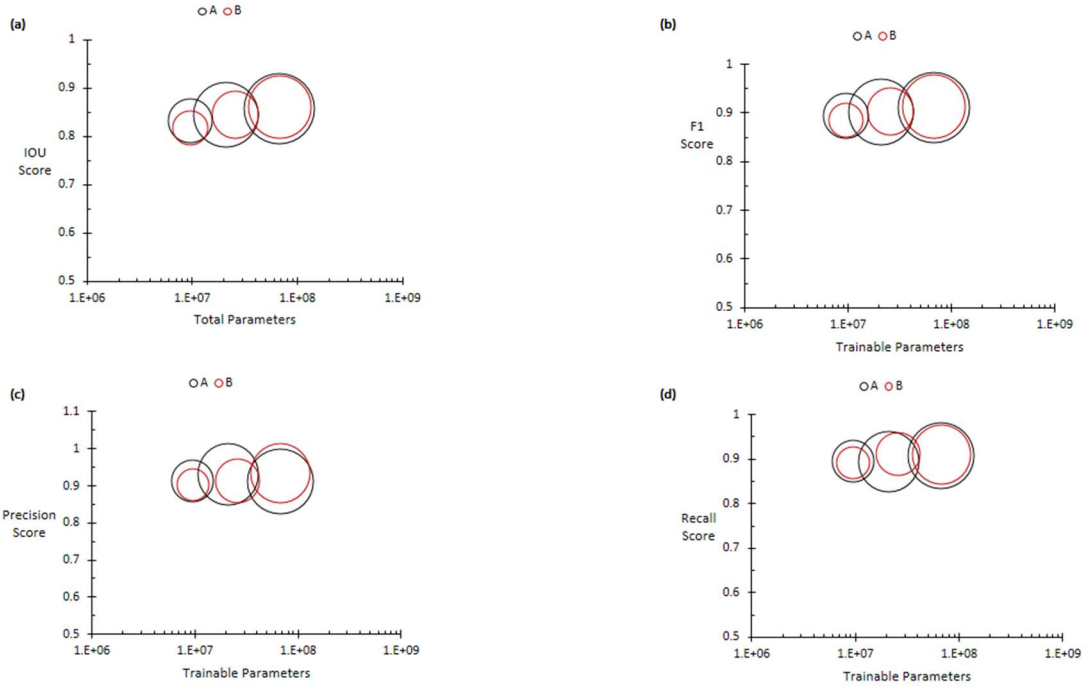

Figure 11: Model performance, measured by (a) IOU, (b) F1, (c) Precision, and (d) Recall, against network size, quantified by the number of trainable parameters and plotted using a base ten semi-log scale. The bubble area is directly proportional to FLOP usage or energy consumption per 100 epochs.

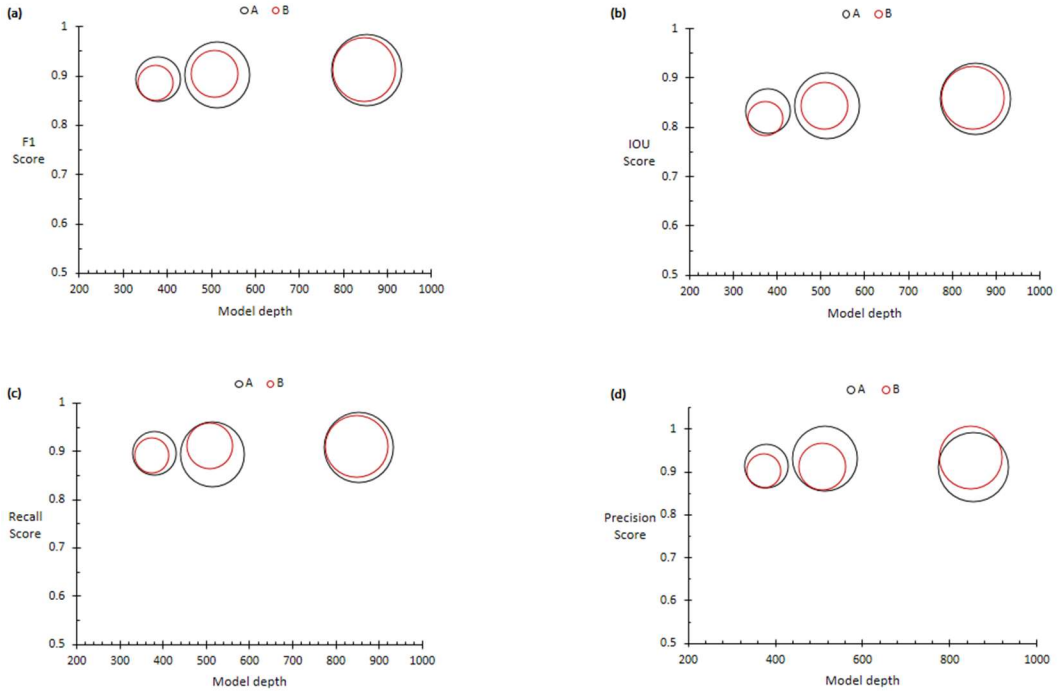

Figure 12: Model performance, measured by (a) F1, (b) IOU, (c) Precision and (d) Recall, against network size, quantified by the number of hidden layers (equivalent to network depth). The bubble area is directly proportional to FLOP usage or energy consumption per 100 epochs.

4.3 Multi-factor Analysis of Algorithmic Resource Use

The effect of the same primary factors for the resource efficiency analysis, except whether the model was pre-trained, was investigated. As shown above, this factor is important for predictive performance, however, it is not essential for energy consumption or resource use, so it was eliminated to simplify the analysis. The analysis is based on fitting a linear model with categorical inputs (Architecture, Backbone, Objective Function) and a continuous output (Algorithm resource consumption, estimated using the kW/100 epochs as a measure). The model fitted achieved an R2 score of 0.96 and a P-Value of 0.0036, indicating the results' statistical significance.

Unlike the Intersection-Over-Union and F1 Score analysis, the resource use analysis can determine that the selection of the main factor of the overall architecture is the most critical determinant of resource use, as seen in Table 1 below. The choice of backbone blocks follows it, however, the two top factors are relatively independent (their interaction being the most negligible factor to the model, also achieving a P-Value higher than 0.51). Therefore, the performance prediction of models is far simpler than their performance on the given task relying heavily on the network's overall design.

Table 1: Energy Use Modelling Results

Factor	P-Value	Logworth
Architecture	*0.00047*	*3.324*
Backbone	*0.00137*	*2.863*
Architecture - Objective Function Interaction	0.08425	1.074
Objective Function	0.23124	0.636
Backbone - Objective Function Interaction	0.43500	0.362
Architecture - Backbone Interaction	0.52148	0.283

To offer a more precise overview, the interaction profiles have also been plotted (Fig. 13), showing how the choice of FPN significantly increases resource requirements

across the board and the selection of ResNet for backbone reduces the resource requirements. The choice of objective function between Focal and Minkowski is also shown to have relatively little effect.

The top three architectures for resource use all use the ResNet backbone with LinkNet architecture combination. The difference in resource use is significant, with the most efficient networks achieving, on average, 0.65, 0.83 and 0.87 kW/100 epoch, and the three most resource intensive requiring 3.24, 3.04, and 2.62. The most resource-intensive algorithms use the FPN architecture, which was shown to be highly accurate. This comparison highlights the much more significant differences in resource usage compared to accuracy, with the lightest model requiring only 20% of the most expensive model's resources.

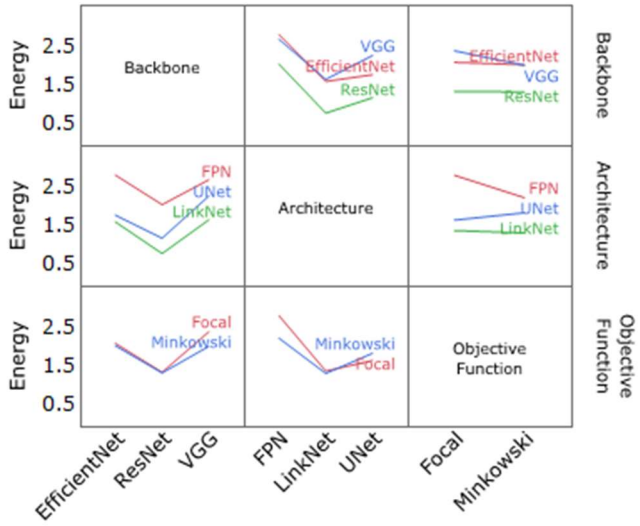

Figure 13: Interaction profiles using local energy usage as the metric of resource intensity.

Therefore, the top-performing model (FPN, EfficientNet, Minkowski) ranks third worse in resource requirements, though not the most resource-intensive, with 2.62 kW/100 epochs. The second best performer in terms of IOU and F1 measures, however, places roughly average, with 1.89 kW/100 epochs consumption. The average of all models studied being 1.80 kW/100 epochs, therefore, comparing the top two models on these criteria, the top model results in 0.071% and 0.19% improvement in IOU and F1 scores over the second-best performing model. However this also produced 45.6% increase in resource use.

5 Conclusion

In conclusion, this study aimed to achieve a resolution IV experimental design configuration that helped identify the main factors affecting the process and the outcome of the matrix-pore-fracture-vug semantic segmentation of rock micro-CT samples. It also helped to discern their first-order interaction and analyse the performance of the applied segmentation model.

The segmentation performance analysis showed that larger deep learning models performed only marginally better than baseline configuration (20-25 million parameters). It was found that the most significant choices on an individual basis are backbone selection and transfer learning. Additionally, the novel Minkowski Difference objective function improves performance by at least 2% compared to Focal Loss for all architectures that had started with pre-trained weights. The improvement was particularly distinct in IOU scores.

When also considering the GreenAI pillars of computational efficiency and environmental impact, the UNet – EfficientNet – Pretrained – Minkowski Difference Loss model configuration was found to be best as it offers very high segmentation accuracy and moderate energy consumption (40% lower compared to the only better-performing configuration). Hence this study prioritises classification accuracy, but analysing resource intensity also significantly reduces resource usage and carbon footprint.

Further work can be carried out comparing these models with algorithms with higher explainability. Moreover, the study investigated the applications of 2D convolutions to the analysis of 3D images, as is the currently dominant approach, but 3D convolutions could also be discussed in future research.

Overall, this study offers an analysis of deep learning convolutional neural networks applied to the problem of multi-class semantic segmentation of 3D micro-CT images. It was proposed a novel loss function that, whilst focusing on connectivity and curvature, which are essential features for porous media flow, – also improves the conventional area and pixel-based computer vision metrics. It was shown that using the Green AI principles can balance resource use and accuracy, obtaining significant

increases in efficiency for minimal accuracy losses. The results of this algorithm can be used to enhance the modelling of heterogeneous (fractured and vuggy) samples using the multiscale pore network model, offering a way to improve understanding and prediction of flow behaviour in the subsurface formations. This tool, integrated into the overall digital rock technology workflow and can offer critical insights for designing and optimising reservoir engineering projects.

6 References

1. Anthony, L.F.W., Kanding, B. and Selvan, R., (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*. https://doi.org/10.48550/arXiv.2007.03051
2. Armstrong, R.T., McClure, J.E., Robins, V., Liu, Z., Arns, C.H., Schlüter, S. and Berg, S., (2019). Porous media characterisation using Minkowski functionals: Theories, applications and future directions. *Transport in Porous Media*, 130, pp.305-335. https://doi.org/10.1007/s11242-018-1201-4
3. Berg, C.F., Lopez, O. and Berland, H., (2017). Industrial applications of digital rock technology. *Journal of Petroleum Science and Engineering*, 157, pp.131-147. https://doi.org/10.1016/j.petrol.2017.06.074
4. Blunt, M.J., (2017). *Multiphase flow in permeable media: A pore-scale perspective*. Cambridge university press. https://doi.org/10.1017/9781316145098
5. Boone, M.A., De Kock, T., Bultreys, T., De Schutter, G., Vontobel, P., Van Hoorebeke, L. and Cnudde, V., (2014). 3D mapping of water in oolithic limestone at atmospheric and vacuum saturation using X-ray micro-CT differential imaging. *Materials Characterization*, 97, pp.150–160. https://doi.org/10.1016/j.matchar.2014.08.010
6. Boelens, A.M. and Tchelepi, H.A., (2021). QuantImPy: Minkowski functionals and functions with Python. *SoftwareX*, 16, p.100823. https://doi.org/10.1016/j.softx.2021.100823
7. Budennyy, S., Lazarev, V., Zakharenko, N., Korovin, A., Plosskaya, O., Dimitrov, D., Arkhipkin, V., Oseledets, I., Barsola, I., Egorov, I. and Kosterina, A., (2022). Eco2AI: carbon emissions tracking of machine learning models as the first step towards sustainable AI. *arXiv preprint* arXiv:2208.00406. https://doi.org/10.1134/S1064562422060230
8. Bultreys, T., Stappen, J., Kock, T., Boever, W., Boone, M., Hoorebeke, L. and Cnudde, V., (2016). Investigating the relative permeability behavior of microporosity-rich carbonates and tight sandstones with multiscale pore network models. *Journal of Geophysical Research: Solid Earth*, 121(11), pp.7929-7945. https://doi.org/10.1002/2016JB013328

9. Chaurasia, A. and Culurciello, E., (2017) Linknet: Exploiting encoder representations for efficient semantic segmentation. *IEEE visual communications and image processing (VCIP)* (pp. 1-4). IEEE. https://doi.org/10.48550/arXiv.1707.03718

10. Chi, L. and Heidari, Z., (2016). Directional-Permeability Assessment in Formations with Complex Pore Geometry With a New Nuclear-Magnetic-Resonance-Based Permeability Model. *SPE Journal*, 21(04), pp.1436-1449. https://doi.org/10.2118/179734-PA

11. Das, R., Mondal, A., Chakraborty, T. and Ghosh, K., (2022). Deep neural networks for automatic grain-matrix segmentation in plane and cross-polarized sandstone photomicrographs. *Applied Intelligence*, 52(3), pp.2332–2345. https://doi.org/10.48550/arXiv.2111.07102

12. Dhar, P., (2020). The carbon impact of artificial intelligence. *Nat. Mach. Intell.*, 2(8), pp.423-425. https://doi.org/10.1038/s42256-020-0219-9

13. Rex, C.E.S., Annrose, J. and Jose, J.J., (2022). Comparative analysis of deep convolution neural network models on small scale datasets. *Optik*, 271, p.170238. https://doi.org/10.1016/j.ijleo.2022.170238

14. Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L. and Han, W., (2021). Pre-trained models: Past, present and future. *AI Open*, 2, pp.225–250. https://doi.org/10.48550/arXiv.2106.07139

15. Iakubovskii, P., (2019). Segmentation Models. *GitHub repository*. Available at: https://github.com/qubvel/segmentation_models (Accessed: February 2023).

16. Japperi, N.S., Wu, K., Starkey, A. and Panaitescu, C. (2022). Quantification of Pore Size Population and Diagenesis using Digital Rock Tools. In *Asia Petroleum Geoscience Conference and Exhibition (APGCE)* Kuala Lumpur, 28-29 November 2022. European Association of Geoscientists & Engineers. https://doi.org/10.3997/2214-4609.202270209

17. Japperi, N.S., Wu, K., Starkey, A. and Panaitescu, C., (2023). A New Method for Quantitative Diagenesis via Digital Rock Tools. *SPE Offshore Europe Conference and Exhibition*. Aberdeen, UK, 5-8 September. p. SPE-215503-MS SPE. https://doi.org/10.2118/215503-MS

18. Karpyn, Z.T., Grader, A.S. and Halleck, P.M., (2007). Visualization of fluid occupancy in a rough fracture using micro-tomography. *Journal of colloid and interface science*, 307(1), pp.181-187. https://doi.org/10.1016/j.jcis.2006.10.082

19. Kazak, A., Simonov, K. and Kulikov, V., (2021). Machine-learning-assisted segmentation of focused ion beam-scanning electron microscopy images with artifacts for improved void-space characterization of tight reservoir rocks. *SPE Journal*, 26(04), pp.1739-1758. https://doi.org/10.2118/205347-PA

20. Legland, D., Kiêu, K. and Devaux, M.F., (2007). Computation of Minkowski measures on 2D and 3D binary images. *Image Analysis & Stereology*, 26(2), pp.83-92. https://doi.org/10.5566/ias.v26.p83-92

21. Li, W., Liu, X., Liang, L., Zhang, Y., Li, X. and Xiong, J., (2020). Pore-structural characteristics of tight fractured-vuggy carbonates and its effects on the P-and S-wave velocity: a micro-CT study on full-diameter cores. *Energies*, 13(22), p.6148. https://doi.org/10.3390/en13226148

22. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988). https://doi.org/10.48550/arXiv.1708.02002

23. Liu, X., Zhang, Y., Jing, H., Wang, L. and Zhao, S., (2020). Ore image segmentation method using U-Net and Res-Unet convolutional networks. *RSC advances*, 10(16), pp.9396–9406. https://doi.org/10.1039/C9RA05877J

24. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X. and Martel, A.L., (2021). Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71, p.102035. https://doi.org/10.1016/j.media.2021.102035

25. Malik, O.A., Puasa, I. and Lai, D.T.C., (2022). Segmentation for Multi-Rock Types on Digital Outcrop Photographs Using Deep Learning Techniques. *Sensors*, 22(21), p.8086. https://doi.org/10.3390/s22218086

26. Nair, P., Mühlbauer, S., Roy, S. and Pöschel, T., (2021). Can Minkowski tensors of a simply connected porous microstructure characterize its permeability?. *Physics of Fluids*, 33(4), p.042010. https://doi.org/10.1063/5.0045701

27. Niu, Y., Mostaghimi, P., Shabaninejad, M., Swietojanski, P. and Armstrong, R.T., (2020). Digital rock segmentation for petrophysical analysis with reduced user bias using convolutional neural networks. *Water Resources Research*, *56*(2), p.e2019WR026597. https://doi.org/10.1029/2019WR026597

28. Panaitescu, C.T., Wu, K., Tanino, Y. and Starkey, A., (2023). AI Enabled Digital Rock Technology for Larger Scale Modelling of Complex Fractured Subsurface Rocks. *SPE Offshore Europe Conference and Exhibition*. Aberdeen, UK, 5-8 September. p. SPE-215499-MS. SPE. https://doi.org/10.2118/215499-MS

29. Rabbani, A. and Babaei, M., (2019). Hybrid pore-network and lattice-Boltzmann permeability modelling accelerated by machine learning. *Advances in water resources*, 126, pp.116-128. https://doi.org/10.1016/j.advwatres.2019.02.012

30. Ramos, M., Nicolas Espinoza, D., Torres-Verdín, C. and Grover, T., (2017). Use of S-wave anisotropy to quantify the onset of stress-induced microfracturing. *GEOPHYSICS*, 82(6), pp.MR201-MR212. https://library.seg.org/doi/abs/10.1190/geo2016-057

31. Ronneberger, O., Fischer, P. and Brox, T., (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing. https://doi.org/10.48550/arXiv.1505.04597

32. Simonyan, K. and Zisserman, A., (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

33. Tan, M. and Le, Q., (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR. https://doi.org/10.48550/arXiv.1905.11946

34. Wang, L., Chen, W. and Vuik, C., (2022). Hybrid-dimensional modeling for fluid flow in heterogeneous porous media using dual fracture-pore model with flux interaction of fracture–cavity network. *Journal of Natural Gas Science and Engineering*, 100, p.104450. https://doi.org/10.1016/j.jngse.2022.104450

35. Wang, F. and Zai, Y., (2023). Image segmentation and flow prediction of digital rock with U-net network. *Advances in Water Resources,* p.104384. https://doi.org/10.1016/j.advwatres.2023.104384

36. Yeung, M., Sala, E., Schönlieb, C.B. and Rundo, L., (2022). Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95, p.102026. https://doi.org/10.1016/j.compmedimag.2021.102026

37. Yun, P., Tai, L., Wang, Y., Liu, C. and Liu, M., (2019). Focal loss in 3d object detection. *IEEE Robotics and Automation Letters*, 4(2), pp.1263-1270. https://doi.org/10.48550/arXiv.1809.06065

38. Xiong, Q., Baychev, T.G. and Jivkov, A.P., (2016). Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport. *Journal of contaminant hydrology*, 192, pp.101-117. https://doi.org/10.1016/j.jconhyd.2016.07.002

39. Zhao, J., Qin, F., Derome, D., Kang, Q. and Carmeliet, J., (2020). Improved pore network models to simulate single-phase flow in porous media by coupling with lattice Boltzmann method. *Advances in Water Resources*, 145, p.103738. https://doi.org/10.1016/j.advwatres.2020.103738

7 Appendix

7.1 Deep Learning Algorithm Evaluation and Selection Flow Chart

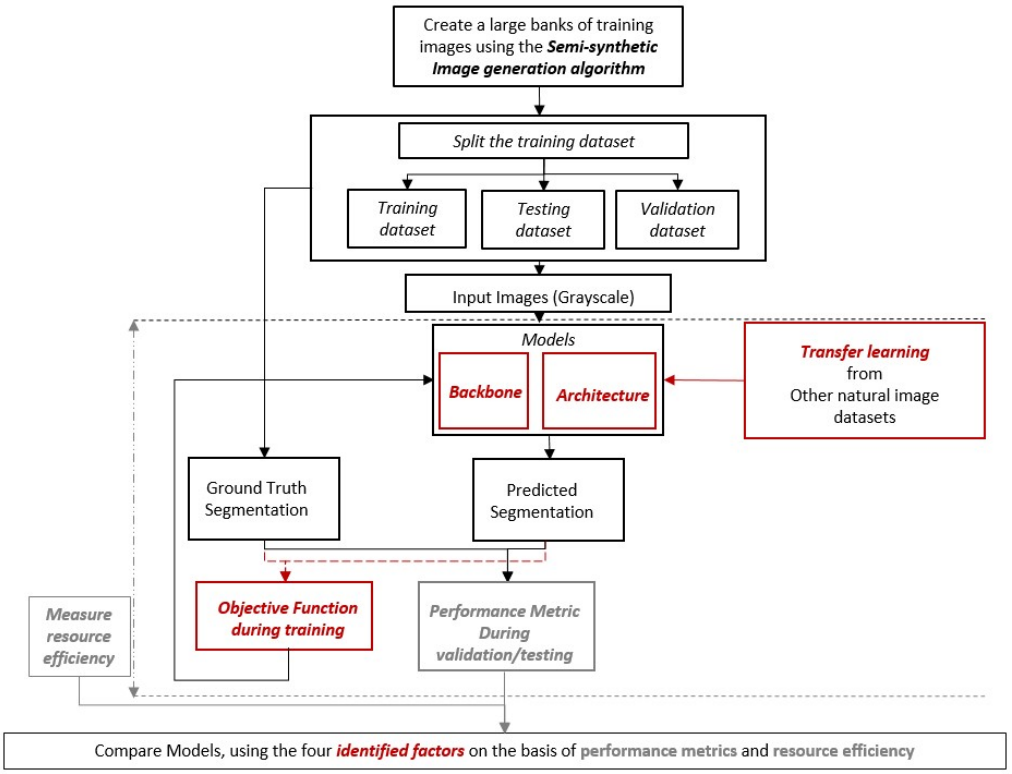

Figure 14: Model training, evaluation and selection analysis showing the design of experiments factors of interest in red and the evaluation criteria in grey.

7.2 Algorithm for Semi-synthetic Image Generation

Inputs:

Image bank containing pores, vugs, fractures and secondary deposition features

Probability of selecting a pore, vug, or fracture image: p_{pore}, p_{vug}, $p_{fracture}$

Outputs:

Semi-synthetic training images

Steps:

1. Initialise an empty list to store selected images.

2. Loop through the number of desired training images:

a. Choose a random image from the image bank based on probabilities p_{pore}, p_{vug}, and $p_{fracture}$.

b. Apply random augmentations (rotation, translation, random intensity addition, zoom and cropping) to the selected image.

c. Add the augmented image to the list of selected images.

3. For each selected image:

a. Combine the pore and matrix images and adjust the mean intensity and histogram using normalisation and matching.

b. Add the vug image geometry to the combined image and perform watershed segmentation on the pores.

c. Re-label the individual pore elements if they intersect any vug elements.

d. Add the fracture image to the segmented image and re-label any intersecting pore elements as fractures.

e. Re-label any vug elements as fractures if they meet the criteria of sharing at least 1 pixel with a fracture element and having an intersected area of at least 25% of the vug area.

f. If fractures intersect with vugs and do not meet the above criteria, some fracture pixels should be labelled as vugs using the following procedures:

i. Find the line perpendicular to the main fracture axis.

ii. Iteratively enlarge the vug area using morphological operations (dilation).

iii. Re-label the new boundary pixels if any pixels on the line perpendicular to the main fracture axis are still labelled as fractures.

g. Isolate all intersecting vugs and fractures and histogram match to remove any a-physical artefacts, such as sharp grey level boundaries at the transition between the fracture and vug void space.

h. Superimpose secondary deposition features (e.g. pyrite crystals) if they exist in the target example.

4. Return the list of semi-synthetic training images.

7.3 Design of Experiments (DOX) Model Results

Table 2: Scaled estimates for the DOX model of energy consumption against the main factors considered – nominal factors expanded to all levels.

Term	Scaled Estimate
Intercept	1.885359
Backbone[EfficientNet]	0.1951929
Backbone[ResNet]	-0.53499
Backbone[VGG]	0.3397966
Architecture[FPN]	0.6524099
Architecture[LinkNet]	-0.525158
Architecture[UNet]	-0.127252
Objective Function[Focal]	0.0765771
Objective Function[Minkowski]	-0.076577
Backbone[EfficientNet]*Architecture[FPN]	0.1016813
Backbone[EfficientNet]*Architecture[LinkNet]	0.0612648
Backbone[EfficientNet]*Architecture[UNet]	-0.162946
Backbone[ResNet]*Architecture[FPN]	0.0630872
Backbone[ResNet]*Architecture[LinkNet]	-0.031679
Backbone[ResNet]*Architecture[UNet]	-0.031408
Backbone[VGG]*Architecture[FPN]	-0.164769
Backbone[VGG]*Architecture[LinkNet]	-0.029585
Backbone[VGG]*Architecture[UNet]	0.194354
Backbone[EfficientNet]*Objective Function[Focal]	-0.044357
Backbone[EfficientNet]*Objective Function[Minkowski]	0.0443572
Backbone[ResNet]*Objective Function[Focal]	-0.068195
Backbone[ResNet]*Objective Function[Minkowski]	0.0681949
Backbone[VGG]*Objective Function[Focal]	0.112552
Backbone[VGG]*Objective Function[Minkowski]	-0.112552
Architecture[FPN]*Objective Function[Focal]	0.2159748
Architecture[FPN]*Objective Function[Minkowski]	-0.215975
Architecture[LinkNet]*Objective Function[Focal]	-0.044698
Architecture[LinkNet]*Objective Function[Minkowski]	0.0446982
Architecture[UNet]*Objective Function[Focal]	-0.171277
Architecture[UNet]*Objective Function[Minkowski]	0.1712767

Table 3: Scaled estimates for the DOX model of performance, using the IOU metric (left) and F1 metric (right) against the main factors considered – nominal factors expanded to all levels.

Term	Scaled Estimate
Intercept	0.7739167
Backbone[EfficientNet]	0.0001167
Backbone[ResNet]	-0.034333
Backbone[VGG]	0.0342167
Architecture[FPN]	0.00315
Architecture[LinkNet]	-0.009467
Architecture[UNet]	0.0063167
Transfer Learning[No]	-0.03405
Transfer Learning[Yes]	0.03405
Objective Function[Focal]	-0.003533
Objective Function[Minkowski]	0.0035333
Backbone[EfficientNet]*Architecture[FPN]	-0.006933
Backbone[EfficientNet]*Architecture[LinkNet]	0.0038833
Backbone[EfficientNet]*Architecture[UNet]	0.00305
Backbone[ResNet]*Architecture[FPN]	0.0151667
Backbone[ResNet]*Architecture[LinkNet]	-0.001867
Backbone[ResNet]*Architecture[UNet]	-0.0133
Backbone[VGG]*Architecture[FPN]	-0.008233
Backbone[VGG]*Architecture[LinkNet]	-0.002017
Backbone[VGG]*Architecture[UNet]	0.01025
Backbone[EfficientNet]*Transfer Learning[No]	-0.036067
Backbone[EfficientNet]*Transfer Learning[Yes]	0.0360667
Backbone[ResNet]*Transfer Learning[No]	0.0008083
Backbone[ResNet]*Transfer Learning[Yes]	-0.000808
Backbone[VGG]*Transfer Learning[No]	0.0352583
Backbone[VGG]*Transfer Learning[Yes]	-0.035258
Backbone[EfficientNet]*Objective Function[Focal]	0.0096667
Backbone[EfficientNet]*Objective Function[Minkowski]	-0.009667
Backbone[ResNet]*Objective Function[Focal]	0.0048917
Backbone[ResNet]*Objective Function[Minkowski]	-0.004892
Backbone[VGG]*Objective Function[Focal]	-0.014558
Backbone[VGG]*Objective Function[Minkowski]	0.0145583
Architecture[FPN]*Transfer Learning[No]	0.0018667
Architecture[FPN]*Transfer Learning[Yes]	-0.001867
Architecture[LinkNet]*Transfer Learning[No]	-0.007908
Architecture[LinkNet]*Transfer Learning[Yes]	0.0079083
Architecture[UNet]*Transfer Learning[No]	0.0060417
Architecture[UNet]*Transfer Learning[Yes]	-0.006042
Architecture[FPN]*Objective Function[Focal]	0.0024167
Architecture[FPN]*Objective Function[Minkowski]	-0.002417
Architecture[LinkNet]*Objective Function[Focal]	-0.002208
Architecture[LinkNet]*Objective Function[Minkowski]	0.0022083
Architecture[UNet]*Objective Function[Focal]	-0.000208
Architecture[UNet]*Objective Function[Minkowski]	0.0002083
Transfer Learning[No]*Objective Function[Focal]	0.00305
Transfer Learning[No]*Objective Function[Minkowski]	-0.00305
Transfer Learning[Yes]*Objective Function[Focal]	-0.00305
Transfer Learning[Yes]*Objective Function[Minkowski]	0.00305

Term	Scaled Estimate
Intercept	0.8517681
Backbone[EfficientNet]	-0.000139
Backbone[ResNet]	-0.025956
Backbone[VGG]	0.0260944
Architecture[FPN]	0.0033861
Architecture[LinkNet]	-0.008347
Architecture[UNet]	0.0049611
Transfer Learning[No]	-0.025313
Transfer Learning[Yes]	0.0253125
Objective Function[Focal]	-0.004096
Objective Function[Minkowski]	0.0040958
Backbone[EfficientNet]*Architecture[FPN]	-0.007803
Backbone[EfficientNet]*Architecture[LinkNet]	0.0033056
Backbone[EfficientNet]*Architecture[UNet]	0.0044972
Backbone[ResNet]*Architecture[FPN]	0.0120639
Backbone[ResNet]*Architecture[LinkNet]	0.0016972
Backbone[ResNet]*Architecture[UNet]	-0.013761
Backbone[VGG]*Architecture[FPN]	-0.004261
Backbone[VGG]*Architecture[LinkNet]	-0.005003
Backbone[VGG]*Architecture[UNet]	0.0092639
Backbone[EfficientNet]*Transfer Learning[No]	-0.025913
Backbone[EfficientNet]*Transfer Learning[Yes]	0.0259125
Backbone[ResNet]*Transfer Learning[No]	-0.004125
Backbone[ResNet]*Transfer Learning[Yes]	0.004125
Backbone[VGG]*Transfer Learning[No]	0.0300375
Backbone[VGG]*Transfer Learning[Yes]	-0.030038
Backbone[EfficientNet]*Objective Function[Focal]	0.0077042
Backbone[EfficientNet]*Objective Function[Minkowski]	-0.007704
Backbone[ResNet]*Objective Function[Focal]	0.0052417
Backbone[ResNet]*Objective Function[Minkowski]	-0.005242
Backbone[VGG]*Objective Function[Focal]	-0.012946
Backbone[VGG]*Objective Function[Minkowski]	0.0129458
Architecture[FPN]*Transfer Learning[No]	0.001675
Architecture[FPN]*Transfer Learning[Yes]	-0.001675
Architecture[LinkNet]*Transfer Learning[No]	-0.008325
Architecture[LinkNet]*Transfer Learning[Yes]	0.008325
Architecture[UNet]*Transfer Learning[No]	0.00665
Architecture[UNet]*Transfer Learning[Yes]	-0.00665
Architecture[FPN]*Objective Function[Focal]	0.0021917
Architecture[FPN]*Objective Function[Minkowski]	-0.002192
Architecture[LinkNet]*Objective Function[Focal]	-0.000708
Architecture[LinkNet]*Objective Function[Minkowski]	0.0007083
Architecture[UNet]*Objective Function[Focal]	-0.001483
Architecture[UNet]*Objective Function[Minkowski]	0.0014833
Transfer Learning[No]*Objective Function[Focal]	0.0003625
Transfer Learning[No]*Objective Function[Minkowski]	-0.000363
Transfer Learning[Yes]*Objective Function[Focal]	-0.000363
Transfer Learning[Yes]*Objective Function[Minkowski]	0.0003625

Table 4: Design of experiments results for the orthogonally selected factors

Run	Backbone	Architecture	Pre-training	Objective unction	F1	IOU	Precision	Recall	Trainable Parameters	Average Energy (kW/100 epochs)	Nr. Layers	Carbon Footprint (gCO_2/100 epochs)
1	EfficientNet	FPN	No	M	0.7915	0.6904	0.819	0.8053	20,793,805	2.62	514	304
2	ResNet	FPN	Yes	F	0.872	0.79	0.87	0.875	23,917,896	2.20	205	255
3	VGG	UNet	No	F	0.8853	0.8167	0.896	0.8898	23,748,821	2.15	66	248
4	EfficientNet	UNet	Yes	M	0.9039	0.8446	0.9135	0.9113	25,608,413	1.89	508	219
5	VGG	UNet	Yes	M	0.8996	0.8388	0.9142	0.8993	23,748,821	2.43	66	282
6	ResNet	LinkNet	Yes	F	0.857	0.7655	0.8344	0.8812	21,620,408	0.64	214	74
7	EfficientNet	LinkNet	Yes	M	0.9036	0.8456	0.9214	0.9027	21,586,877	1.62	523	187
8	ResNet	FPN	No	M	0.8098	0.7197	0.867	0.8075	23,917,896	1.92	205	223
9	VGG	LinkNet	Yes	F	0.85	0.78	0.89	0.83	20,318,901	1.85	81	214
10	ResNet	UNet	No	M	0.7942	0.7012	0.8681	0.7837	24,439,384	0.95	199	110
11	VGG	LinkNet	No	M	0.8783	0.8072	0.8973	0.8853	20,318,901	1.49	81	172
12	EfficientNet	FPN	Yes	F	0.9022	0.844	0.9311	0.8936	20,793,805	3.04	514	352
13	ResNet	UNet	Yes	M	0.8405	0.7617	0.8896	0.8352	24,439,384	1.62	199	188
14	EfficientNet	UNet	No	F	0.819	0.7283	0.8593	0.8088	25,608,413	1.68	508	194
15	VGG	FPN	No	F	0.8689	0.7935	0.8921	0.876	17,574,853	3.24	66	375
16	VGG	FPN	Yes	M	0.8858	0.8187	0.9305	0.8707	17,574,853	2.18	66	252
17	ResNet	UNet	Yes	F	0.8391	0.7579	0.8879	0.8368	24,439,384	1.16	199	134
18	ResNet	LinkNet	No	M	0.7806	0.6849	0.8578	0.7754	21,620,408	0.87	214	101
19	EfficientNet	LinkNet	No	F	0.7903	0.6974	0.8441	0.7836	21,586,877	1.61	523	186
20	ResNet	LinkNet	No	F	0.7822	0.6893	0.8508	0.789	21,620,408	0.83	214	95