Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Engineering Applications of Artificial Intelligence

Enhancing multi-physics modelling with deep learning: Predicting permeability through structural discontinuities

Amanzhol Kubeyev *

Heriot-Watt University, Edinburgh, EH14 4AS, United Kingdom
Concr, Babraham Research Campus, Cambridge, CB22 3AT, United Kingdom

ARTICLE INFO

Keywords:
Deep learning
Modelling
Multi-physics
Convolutional neural networks
Fluid mechanics

ABSTRACT

Numerical modelling of complex structural discontinuities such as fractures poses a computational challenge, as it involves solving multi-scale and multi-physics phenomena and simulating various processes, including solid, fluid, thermal and chemical interactions. To overcome the limitations of long computation times, simplifications or conceptualizations are often required. However, in multi-physics modelling, it is desirable to obtain predictions of certain parameters without making simplifications. In this study, a data-driven deep learning approach is presented that predicts physical permeability parameters through discontinuities with complicated geometries based on digital images. Images of fractures were generated from a digitalized rough fracture surface of subsurface rock. Permeability was calculated using the Stokes equation and Finite Volume discretization for training and testing purposes. Two cases were analyzed: when the fluid velocity field of the fracture was provided to the CNN for training, and a more challenging case when it was not. Results show that deep learning can accurately predict permeability without fluid velocity information. Besides, the model generalizes well, providing accurate predictions of permeability for fractures with significantly different roughness parameters. In conclusion, this approach can reduce computation time during multi-physics modelling and can be used to predict continuous physical permeability values from an image of a fracture with a complex surface.

1. Introduction

Multi-physics numerical modelling, particularly fracture simulation, is computationally intensive (Andrianov and Nick, 2021). Fracture modelling typically requires multi-scale and multi-physics approaches, incorporating solid mechanics, fluid mechanics, chemical interactions, and thermal processes. Each individual physics component usually requires multiple simulation steps.

Fracture permeability is typically calculated numerically or through laboratory measurement. The latter is time-consuming, expensive, and may not be readily available. In many numerical studies on solid–fluid mechanics coupling, the simple parallel-plates approximation (Witherspoon et al., 1980) is widely used. However, this approximation overestimates permeability or differs from experimental results, as shown in laboratory analysis. Advanced methods (Wang et al., 2016; Zou et al., 2017), such as Lattice Boltzmann or Navier–Stokes equation, resolve fluid flow through the fracture surface but fail to consider surface mechanical deformation under stress. Pyrak-Nolte and Morris (2000) and Pyrak-Nolte and Nolte (2016) had to simplify the modelling by assuming cylindrical shapes for fracture asperities or representing the fracture surface as a network of pipes, but these representations require further testing.

Despite conducting experiments on both solid and fluid mechanics of fractures, Xiong et al. (2011) numerically modelled only the fluid mechanics part using Navier–Stokes. Watanabe et al. (2008) similarly modelled fluid mechanics using Navier–Stokes, but did not model solid mechanics, assuming no deformation occurs during fracture penetration. Huang et al. (2018) modelled fluid mechanics by discretizing the Reynolds equation, but modelled solid mechanics with an analytical model based on bi-dilation. Overall, limiting assumptions or simplifying concepts have to be made to perform computationally demanding multi-physics modelling.

Recent research by Kubeyev et al. (2022) has successfully addressed the solid deformation component of multi-physics problems through the use of a first principles contact mechanics approach. This method avoids making simplifying assumptions, such as assuming specific shapes for fracture asperities. The fluid flow aspect is modelled using the Stokes equation (Curle and Davies, 1968; Richter et al., 2016), which calculates permeability. Because of these, computational time remains a challenge and it would be desirable to find a novel approach to predict certain aspects of multi-physics without the need for extensive numerical simulations.

* Correspondence to: Concr, Babraham Research Campus, Cambridge, CB22 3AT, United Kingdom.
E-mail address: aidan@concr.co.

https://doi.org/10.1016/j.engappai.2023.106562
Received 2 September 2022; Received in revised form 2 February 2023; Accepted 29 May 2023
Available online xxxx

Deep machine learning (Hinton et al., 2006), also known as artificial neural networks, has recently gained significant scientific and engineering interest. It is a versatile and expandable (Géron, 2019) method that can tackle complex tasks such as autonomous driving, speech recognition, handwriting recognition, and image classification. It has also been used in modelling various fields, including fluid mechanics (Brunton et al., 2020; Brenner et al., 2019; Bright et al., 2013; Milano and Koumoutsakos, 2002), numerical modelling (Andrianov and Nick, 2021; Haghighat et al., 2020; Ashworth, 2021; Srinivasan et al., 2018), and geoscience to predict earthquakes (Asencio-Cortés et al., 2018). Andrianov and Nick (2021) applied CNN to address the computational demands of multi-scale fractured reservoir simulation.

Haghighat et al. (2020) applied physics-informed neural networks (PINN) (Raissi et al., 2019) to model solid mechanics. The cost function was modified to incorporate the governing mechanics equations, such as momentum balance and constitutive law. This approach was demonstrated on linear elasticity and non-linear elastoplasticity problems, demonstrating that deep learning can converge to the true solution with limited data.

Deep learning is commonly used for classification problems, such as predicting discrete values like gender or object classification in an image. Regression problems, which involve predicting continuous values, are more challenging to address with deep learning. Usually, continuous values are transformed into discrete values (Mahendran et al., 2017) or ranges to simplify the problem as a classification. Fischer et al. (2015) has explored using CNN for regression, such as predicting the rotation angle of an inclined 2D picture. The model was trained using data-driven methods such as data augmentation and transfer learning (Torrey and Shavlik, 2010). Despite these efforts, the full orientation estimation task resulted in relatively high error. Mahendran et al. (2017) proposed using CNNs for regression to solve the 3D pose estimation problem, which involves retrieving 3D information (such as azimuth, elevation, and distance) from a 2D image. However, the results did not outperform existing methods, with some categories showing a larger error. These findings highlight the challenge of using CNNs for regression tasks.

To summarize, numerical multi-physics modelling is challenging due to the computational demands of solving various processes, such as solid mechanics, fluid dynamics, thermal physics, and chemical reactions in a coupled manner. To reduce computation times, simplifications or limiting assumptions are often necessary. Thus, it is desirable to find more efficient methods to model and predict multi-physics phenomena.

This study explores the potential of utilizing deep learning to enhance physics-based solid–fluid mechanics modelling by replacing one part of it, permeability through complex geometries. The results show that this can be done with high accuracy, without the need for fluid velocity as a key feature. The data for various fractures was generated by digitizing a subsurface rock specimen with intersecting fractures using a digital optical microscope. The computation times of the numerical simulation method were compared to those of deep learning. Although this study does not fit strictly within the category of physics-informed neural networks (PINN), it should be viewed as a separate component that can be integrated into fluid–solid numerical modelling.

2. Methods

Here, the methods used to perform deep learning modelling, calculating the permeability in rough fractures and data generation are introduced. The data-driven modelling is performed in the TensorFlow/Keras platform (Chollet, 2015), an open-source software library and application programming interface for artificial neural networks.

2.1. Deep machine learning

Machine learning is the process of calculating the association between the two parts—provided input data and output, and calculating parameters of a system using limited observations (Cherkassky and Mulier, 2007; Brunton et al., 2020). The approximations made during the learning process are fundamentally stochastic. The learning process is the minimization of a risk functional (Brunton et al., 2020)

$$R_{\boldsymbol{w}} = \int L\left[\boldsymbol{y}, \boldsymbol{\phi}\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}\right)\right] p\left(\boldsymbol{x}, \boldsymbol{y}\right) d\boldsymbol{x} d\boldsymbol{y} \tag{1}$$

where $\boldsymbol{x}$ is the inputs, images in this study, $\boldsymbol{y}$ is the output or also called a target, fracture permeability in our case, p is the probability distribution, $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$ defines the machine learning architecture. The latter is the way and schematics of connecting multiple neurons in a specific way that reflects knowledge about the problem and type of data (Brunton et al., 2020). The weights parameters are w, and L is the loss function, an objective function that quantifies the distance between real and predicted values.

Training of a neural network is the process of finding the best set of weights for the parameters in the layers of a neural network, so as there is a minimum error between the model predicted values and the true values of a target. Here, the mean square error (MSE) is used to quantify the loss (error)

$$L_{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{2}$$

where, $\hat{y}_i$ is model predicted values, y_i is the true values, and m is the number of data points. The target in this study is fracture permeability, hence training means finding the minimum loss between the actual permeability and the predicted by the data-driven model.

Here, three types of losses are analysed: training loss, cross-validation loss and testing loss. In addition to the MSE, a mean absolute percentage error (MAPE) is used to assess the test data due to simplicity. To minimize the loss, an adaptive Adam optimizer is used, which is a stochastic optimization algorithm for first-order gradient-based optimization (Kingma and Ba, 2014).

Here, a specific type of deep learning—Convolutional Neural Network (CNN) is used. It evolved from investigating the visual cortex of a brain Hubel (1959) and Hubel and Wiesel (1959) and from understanding that many neurons in the visual cortex have a small local receptive field, reacting only to the limited visual region (Géron, 2019). The idea was further developed (LeCun et al., 1998) into the CNN by introducing a neural network architecture to recognize hand-written numbers. It applies different filters to the original image, which helps to identify features to extract information during the learning process. These features help to relate the input to the target. Despite much recent success in the application, a solid theory of CNN is not available, and there is no clear understanding of how and why it works (Gu et al., 2018). Bronstein et al. (2021) suggests that CNNs are successful because of the preservation of structures under transformations i.e. symmetry.

2.1.1. CNN architectures and training

CNN architecture is the design and sequence of putting separate neural network layers together. The layers include an input layer which takes an image of a fracture, an output layer to predict a continuous permeability value, and middle layers which consist of convolutional layers, pooling layers, activation functions and normalization layers.

Two CNN architectures were used in this study. First, a sequential architecture consisting of the four blocks, each having a set of layers placed sequentially: convolutional 2D layer, activation layer, (batch) normalization and a pooling layer (Fig. A.12). These are attached to the flattening, dense, activation and finally dense layers outputting a continuous permeability value. The second architecture is a residual network *ResNet50* (He et al., 2016; Dwivedi, 2019) shown in Fig. A.13, which is one of the most successful architectures as it won the ImageNet challenge in 2015.

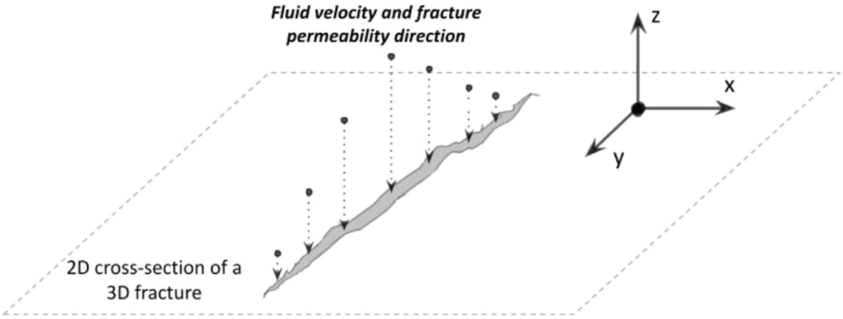

Fluid velocity and fracture permeability direction
z
x
y
2D cross-section of a 3D fracture

Fig. 1. A concept of a 2D fracture that was used in this study. Fracture permeability is calculated perpendicular to the 2D cross-section, i.e. through the 2D rough fracture opening. The cross-section is a slice from a 3D fracture.

A limited amount of hyperparameter tuning was performed on both architectures. Learning rate, batch size, type of activations functions, design and sequence of the first architecture was performed. However, better performance is obtained by switching to *ResNet50*.

2.2. Stokes equation for fracture permeability calculation

In this study, the fracture is a 2D opening with two rough surfaces, located in impermeable media, and the fluid flow through the opening only. The fracture permeability is investigated through a 2D space opening, calculating it perpendicular to the fracture cross-section as shown in Fig. 1. Permeability through this rough fracture can be accurately calculated using the Stokes equation for arbitrary 2D cross-sections (Curle and Davies, 1968; Richter et al., 2016). The assumptions are— incompressible, Newtonian fluid, laminar flow, and impermeable surrounding around the fracture opening. The Stokes equation is the Laplace equation

$$\nabla^2 U = 0 \tag{3}$$

where

$$U = u + \frac{dP}{dz}\frac{(x^2+y^2)}{4\mu}, \tag{4}$$

where u is the fluid velocity in m/s, P is the pressure (Pa) perpendicular to the 2D cross-section and parallel to the z-axis, μ is the dynamic viscosity in Pa s. x and y are coordinates on the plane. The no-flow boundary conditions around the fracture edges

$$u_{bc} = 0 \text{ at } \Gamma, \tag{5}$$

$$U_{bc} = \frac{dP}{dz}\frac{(x_{bc}^2+y_{bc}^2)}{4\mu}. \tag{6}$$

Eq. (3) is solved using discretization of the single-phase incompressible flow-pressure solver of MRST (Lie, 2014) and assumed pressure gradient and boundary conditions. The selection of the gradient does not impact the permeability calculation and taken to be $\frac{dP}{dz} = 10000$ Pa m^{-1}. Once fluid velocity is calculated using Eqs. (3)–(6), rough fracture permeability can be calculated using Darcy's law

$$Q = \int_{\Omega} u\,dA = -\frac{k_{f\perp}A}{\mu}\frac{dP}{dz}, \tag{7}$$

where $k_{f\perp}$ is the fracture permeability through the 2D cross-section, in m^2 converted to Darcy. A is the flow area through the fracture opening in m^2, Q is the volumetric flow rate in m^3/s. Ω denotes the domain for the fluid flow, which in this 2D study is equivalent to the fracture opening. The above equation is discretized using the finite volume method (FVM), and the fracture permeability is obtained by

$$k_{f\perp} = -\frac{\mu \Sigma_{i=1}^{n}(u_i \Delta A_i)}{\Sigma_{i=1}^{n}(\Delta A_i)\frac{dP}{dz}}. \tag{8}$$

where n is the number of grid cells, ΔA_i is the flow area of a cell i. This permeability is a target for deep learning to learn. It should not be confused with the permeability from the matrix to fracture as matrix is impermeable.

2.3. Subsurface core with fracture

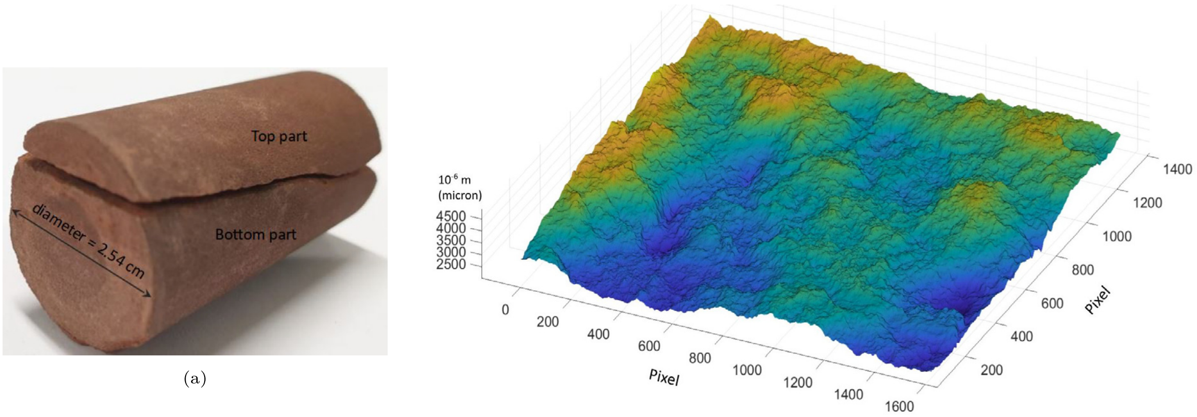
The fracture surface image was generated using a Keyence VHXTM-6000 digital optical microscope (Phillips et al., 2021; Kubeyev et al., 2022; Keyence, 2017) on the rock sample shown in Fig. 2(a). The microscope uses photogrammetry to obtain 3D coordinates and surface irregularity.

The specimen was taken from a natural CO$_2$ reservoir at the Green River, Utah. The formation is 50 m thick, complex, consisting of three lithofacies (Kampman et al., 2013): (a) interbedded, unfossiliferous red and grey shale and bedded gypsum (b) red and grey claystone/siltstone and (c) fine-grained sandstone. The fracture surface data was cleaned from the noise manually, a digital image of part of the surface is shown in Fig. 2(b).

2.4. Fracture images data generation

The present section outlines the generation of a dataset for deep learning. The dataset consists of images of rough fractures as input and fracture permeability as the target output. The conceptual workflow for the generation of fracture images is illustrated in Fig. 3 and is summarized in the following steps:

1. **Region selection:** The two regions in the top and bottom surfaces of the core are selected such that they do not have noise.
2. **Random slicing:** 2D slices are randomly selected (coloured in red) from the selected 3D regions. Using a microscope resolution of $2.25 * 10^{-6}$ m, pixels are converted to lengths. All slices have the same lengths.
3. **Connecting top and bottom slices**. Each pair of slices is brought together and connected to form a closed polygon. The polygon is the only area open to fluid flow, i.e., no flow in the solid matrix. The closed polygon represents a rough fracture.
4. **Fracture discretization:** The fracture opening is discretized to solve the Stokes equation using the Finite Volume Method. Numerical computations are performed to determine the fluid velocity distribution in 2D, and then a single permeability value for the whole fracture opening is obtained.
5. **Creating an image:** For each fracture, a 1280 × 1280 pixel image is created by converting the figure to a JPEG image. Each fracture now has a fluid velocity distribution populated by solving the Stokes equation. The images are then coarsened to 128 × 128 pixels to reduce learning time.
6. **Creating a dataset:** The above steps are repeated to create a dataset of 20,000 images with fractures and corresponding target permeability.
7. **Fracture centering:** An object-centering algorithm is applied to move the fracture to the centre of an image for aesthetics. A visual quality check is performed. It is believed that CNN has invariance to translation, but this was not investigated in this study.



(a)

(b)

Fig. 2. Carmel mudrock core sample with the traversing fracture (a) and digitalized 3D surface of a natural fracture of the Carmel mudstone core (b).

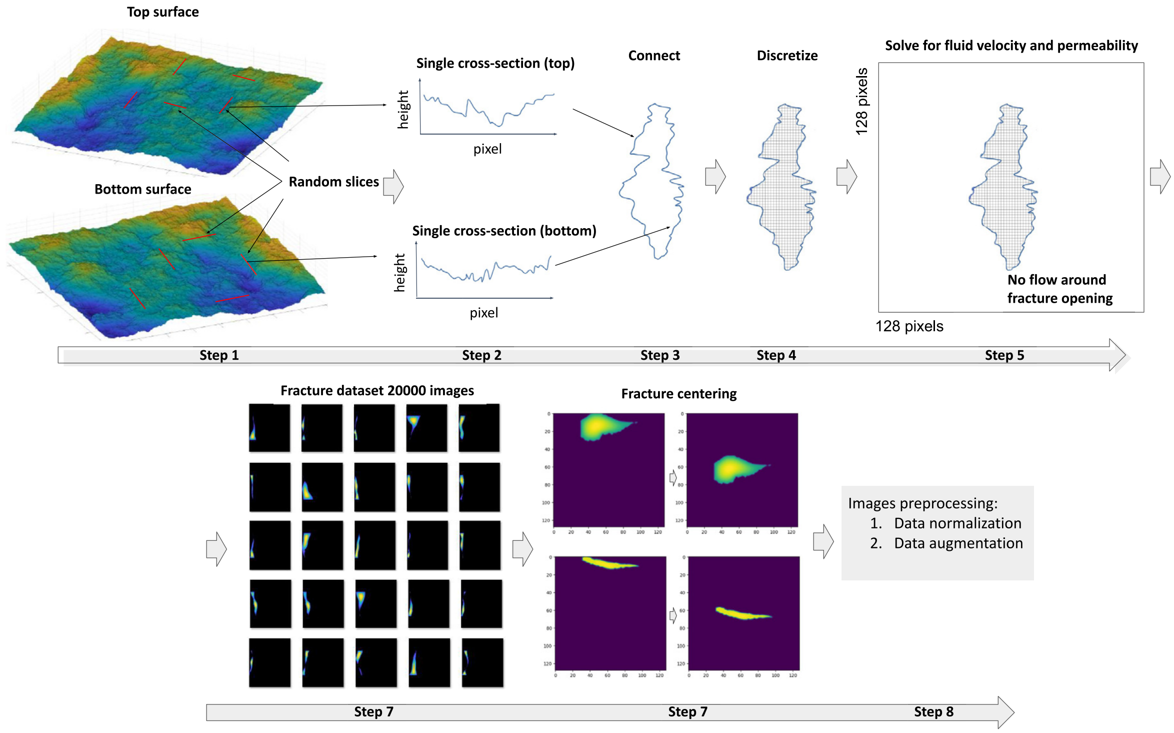


Fig. 3. Step-by-step fractures dataset generation (conceptual workflow).

8. **Images preprocessing:** Images are converted to greyscale and normalized from 0 to 1 by division by 255 (the maximum greyscale limit). Augmentation is performed on the dataset, as described in the next section.

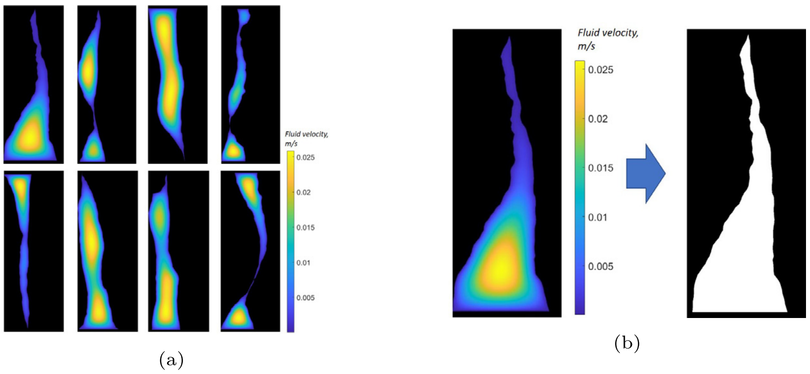
Two types of image datasets are generated based on the difficulty of the problem: easier and more difficult. In the former, fractures have a fluid velocity distribution, which acts as additional data for deep learning to train on. The velocity colour bar limits are set to be the same in all images and are identified based on the maximum of all fracture images in the dataset. Zoomed-in examples of the fracture images are shown in Fig. 4(a).

In difficult problem, the velocity distribution is removed (Fig. 4(b)), by converting pixels to possess only two values: 1 (data is present, white) and 0 (no data, black). This is done by conditioning image pixels: if the greyscale pixel is more than 0, then the pixel is set to 255, with consequent normalization to 0 or 1 by division by 255. This data requires deep learning to study fracture permeability based solely on the geometry of a fracture.

A total of 20,000 images were divided into 12,000 for training, 4000 for cross-validation, and 2000 for testing. The testing images were not used during the training process and were used to evaluate the performance of the trained model on unseen data. Additionally, 2000 extrapolation testing images were created to evaluate the model's performance on significantly different fractures. The extrapolation testing involves testing the model on data that is different from the training data, here, characterized by two roughness coefficients (Fig. 5).

2.5. Fracture roughness

Natural fractures have irregular surfaces and complex geometries. These impact the fluid flow and permeability of fractured media. Irregularities act as barriers to the flow making it more tortuous. Normally,



(a)

(b)

Fig. 4. (a) Example of eight zoomed-in fractures with fluid velocity distribution and (b) example of removing fracture fluid velocity from an image.

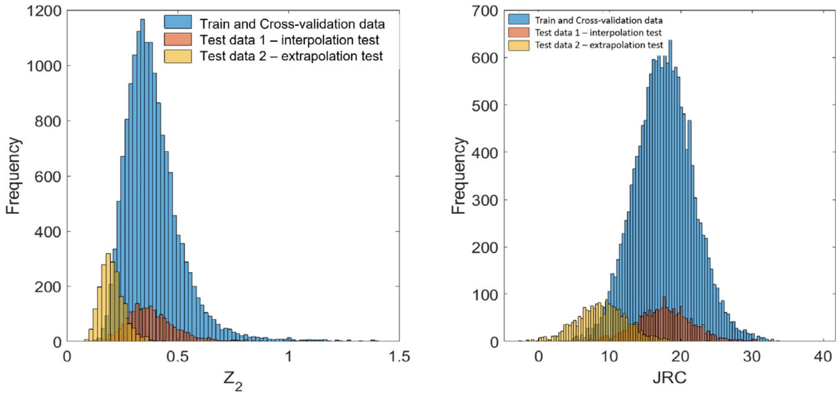


Fig. 5. Distribution of the roughness of all fractures in a dataset: train, interpolation test and extrapolation test data.

to characterize fractures and distinguish between them, a quantitative parameter measuring fracture surface irregularity is used.

Surface irregularity or roughness is defined as the local departures from planarity (Bhushan, 2000). Despite numerous ways to parametrize it, the statistical method is selected because of simplicity (Zhou et al., 2015). The scale-invariant root mean square Z_2 of the heights of fracture surface proposed by Tse and Cruden (1979) is calculated using

$$Z_2 = \left[\frac{1}{M\,(Dx)^2} \sum_{i=1}^{M} \left(x_{i+1} - x_i\right)^2 \right]^{1/2}, \tag{9}$$

where Dx is the spacing between nodes, $x_{i+1} - x_i$ is the height difference between adjacent nodes in metres, calculated along the y-axis as shown in Fig. 1. M is the number of height measurements.

Z_2 can be empirically mapped to the joint roughness coefficient (JRC), a widely used roughness characterization method using the following (Tse and Cruden, 1979) equation

$$JRC = 37.2 + 32.47 \cdot log Z_2. \tag{10}$$

The distributions of roughness parameters help in evaluating the deep learning model on the significantly different test data which has different roughness. For example, Test data 1 in Fig. 5 has similar roughness distribution as in training and validation data, thus for interpolation testing. However, Test data 2 is created for the extrapolation test, where the model is tested on fractures with considerably smaller roughness. Training data Z_2 is ~0.4, and the JRC is ~18, with standard deviations of 0.13 and 4.15. The extrapolation testing data has a JRC of ~ 10.

2.6. Data augmentation

Training of the deep learning model usually improves with more data provided, making the trained model more generalized. Data augmentation is used to expand the available dataset artificially, creating additional data instances (Géron, 2019) by duplicating and varying the existing data. Furthermore, it reduces the possibility of a model overfitting during training (Géron, 2019), making it one of the methods for implicit regularization (Hernández-García and König, 2018; Vapnik and Chervonenkis, 2015). For instance, consider the classification problem when a model needs to identify the products such as jeans, jackets or t-shirts in an image. Creating additional images of jeans by shifting or rotating the existing images increases the dataset. It also forces the machine learning model to be invariant to the position or orientation.

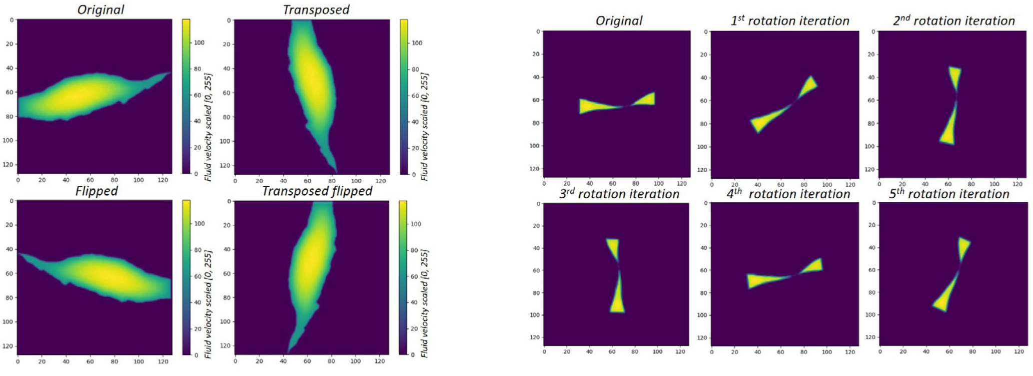
Two types of data augmentations are performed in this study. First, the manual augmentation by applying image manipulations—transposition, flipping the original image and transposition of a flipped image (Fig. 6(a)). Second, the automatic Keras/TesnorFlow augmentation method, which rotates images up to the specified 90 degrees angle (Fig. 6(b)). Augmentation is only applied to the training and cross-validation data, not to the testing data.

3. Results and analysis

This section presents the results of the deep learning training using two CNN architectures. First, the training results using images of the fractures with fluid velocity field (a simpler problem) are presented. Second, no velocity case results are presented. To analyse the training progress results of the loss vs. the epochs are plotted. An epoch is a value indicating the number of passes the entire training and cross-validation dataset the deep learning has completed.

3.1. Fractures with fluid velocity field

Here, training is done on the images with fluid velocity distribution as shown in Fig. 4(a). Training progressed till the learning curve start to become stable. Sequential architecture results are shown in Fig. 7, with training and validation loss (left) and results of the testing evaluation



(a)

(b)

Fig. 6. (a) Manual augmentation performed on a dataset shown on a single fracture. (b) Keras built-in augmentation by rotation to a maximum angle of 90 degrees.

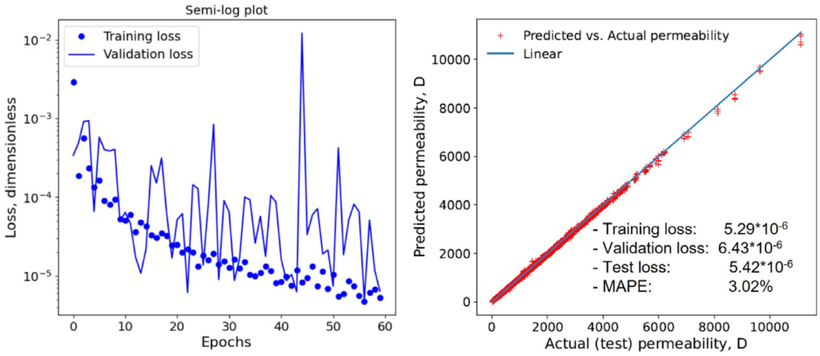


Fig. 7. Results of the sequential CNN architecture model training (*left*) and testing (*right*) on fractures with fluid velocity distribution.

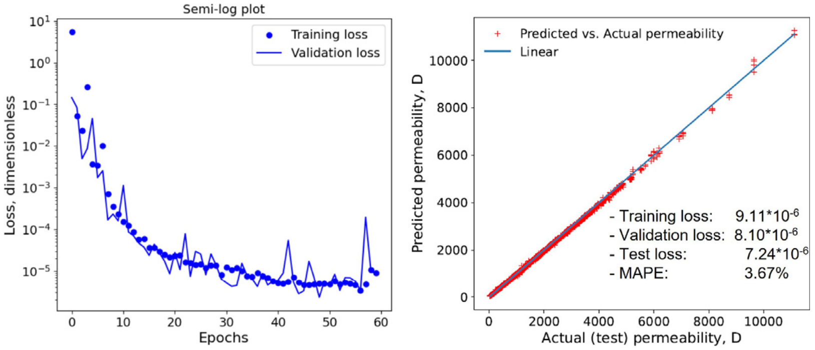


Fig. 8. Results of the *ResNet50* architecture model training (*left*) and testing (*right*) on fractures with fluid velocity distribution.

run (right). The latter shows the quality of the trained model assessed using the linear plot of actual permeability vs predicted permeability of the model. The deviation from the linear blue line indicates the mismatch from the actual true permeability.

The model experienced training as seen from the decrease of the loss functions of training and cross-validation with the increasing epochs, with ~500 times decrease. Training loss is approximately monotonic, whereas validation loss is non-monotonic. The plot indicates some degree of overfitting, which is identified by the difference between the training loss and cross-validation loss. The right plot of Fig. 7 shows evaluation metrics, i.e. how good the predictions of a model on unseen data. Overall, good results provide small 3% test data MAPE, and all losses (train, validation, test) have similar small values.

Results of the *ResNet50* architecture is shown in Fig. 8. If this is compared to sequential architecture, training and validation follow a much more similar curve during training. Hence, less overfitting is present. The testing evaluation provides good results of 3.7% MAPE, and all three losses are approximately the same. The difference in MAPE of both models can be because of the stochastic nature of the training.

3.2. Fractures without fluid velocity field

Here, CNN models are trained based on the fracture images with no fluid velocity field (Fig. 4(b)). This should be a harder problem for the deep learning regression because less information is provided for the training. Besides, this is a natural case that the model needs to identify permeability from the fracture geometry and roughness alone, without a fluid velocity field. This is because the velocity needs to be calculated numerically from the Stokes equation. The computational interest is in bypassing the requirement to solve Stokes numerically by using data-driven deep learning.

Results of the training are shown in Fig. 9 for the sequential (upper) and *ResNet50* (lower) architectures, respectively. The sequential model has larger fluctuations during training compared to *ResNet50*, similarly

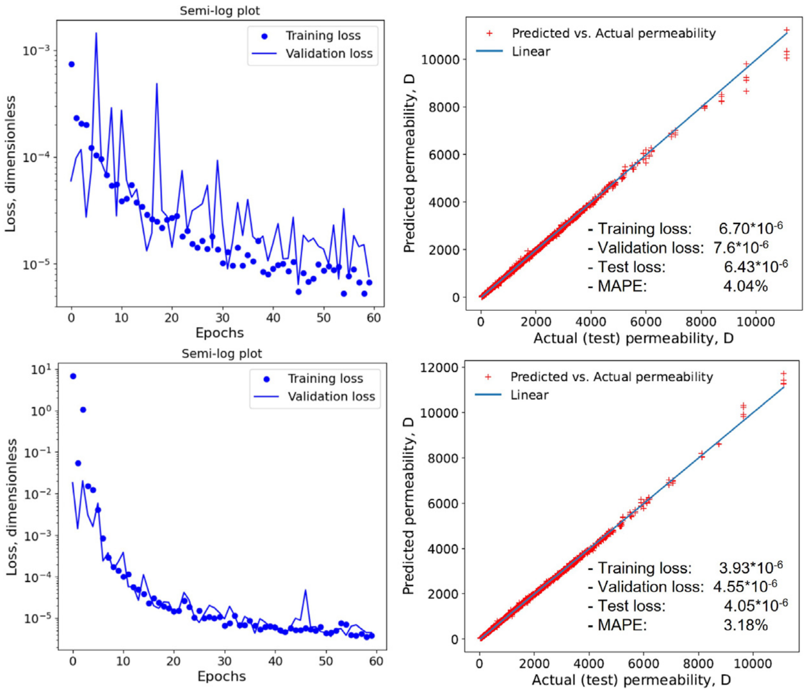


Fig. 9. Results for the deep learning training on fractures with no fluid velocity field using sequential CNN architecture (upper) and *ResNet50* (lower).

to the previous section. Testing of the large permeability values diverges noticeably from the true test permeability. However, good small MAPEs of ∼ 4 and ∼ 3% are observed in both architectures. Overall, both models show good results, which shows that deep learning can train on images without a fluid velocity field if only the image of fracture is provided. Here, *ResNet50* showed better evaluation metrics than the sequential model, with lower MAPEs and three losses.

The fact that CNN can accurately identify permeability through such a complex geometry is interesting considering heterogeneity in the fluid velocity distribution based on which the permeability depends. The velocity field depends on the evolution of the boundary x and y coordinates on the 2D domain. Visually, the geometries vary considerably (Fig. 4(a)) due to randomness in surface coupling adapted in this study. It is not feasible to write analytical solutions for the Navier–Stokes equation for varying geometries. Normally, numerical methods have to be utilized.

3.3. Extrapolation testing

Here, testing and the analysis are performed on a dataset which has significantly different fractures, parametrized by roughness. The roughness distribution is noticeably different from other datasets used during training and testing, as shown in Fig. 5. Only the models trained on images without fluid velocity are used here for evaluation.

Fig. 10 shows the result of the extrapolation test. Both sequential and *ResNet50* have lower accuracy parametrized by larger MAPE values compared to the interpolation test. The MAPE values here are ∼10% and ∼8%, compared to the lower values ∼4 and ∼3% in the interpolation testing. However, because this fracture dataset has significantly different roughness from the one used during training, the overall accuracy looks high.

4. Discussion

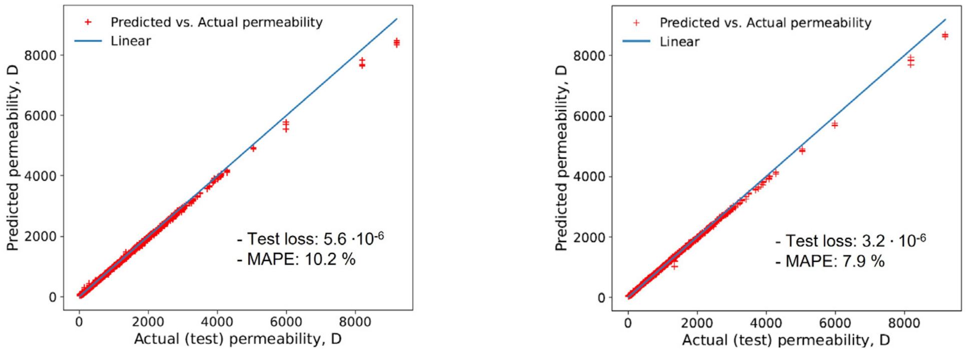
Both (a) the physics-based Stokes numerical solver and (b) the data-driven deep learning models have inherent errors. Besides, both have computation times to provide rough fracture permeability. Thus, for the fair comparison of run time it is important that the two methods have the same level of error.

The error in numerical method is related to the discretization of the domain and converges to the true solution with increasing spatial resolution. The deep learning model has an error related to the learning process and the amount of data provided. Here, a deep learning model was trained with a relatively small amount of data to high accuracy (MAPE ∼ 3%). However, the run-time of the numerical model can be reduced by decreasing the accuracy when the number of discretizing elements is lowered.

To increase error in the numerical method, the convergence study of the Stokes solver in a single fracture was performed. Results are shown in Fig. 11(a) as the relationship between the number of discretizing elements and fracture permeability. The deep learning model was trained on numerical outputs based on the discretization using 105 500 elements, which is shown as a red circle on the plot. Decreasing the number of elements leads to the increase in MAPE to ∼ 3% in the numerical model, compared to the red coloured point. Comparing both methods having the same MAPE results in the reduction of the runtime by the factor of two in the data-driven method.

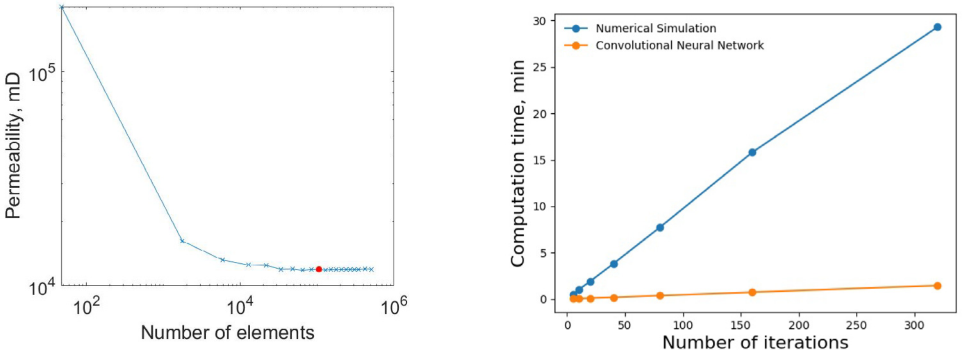
The simulation of the Stokes equation in a single 2D fracture is not particularly computationally intensive. Depending on the number of elements and the complexity of the fracture, it can take anywhere from 0.1 to 1 min to run using MATLAB 2020b and a laptop with an Intel Core i5-8250U CPU @1.60 GHz and a GeForce GTX 1050 GPU. MATLAB takes advantage of the graphics processing unit when possible. However, in the case of coupled solid–fluid mechanics simulation, as described in Kubeyev et al. (2022), computing the permeability at each step of fracture deformation becomes computationally demanding, requiring thousands of calculations, depending on the problem. Furthermore, the addition of chemical and thermal processes relevant to the fracture simulation exacerbates the situation. This study therefore aims to improve a specific aspect of fracture simulation, namely fluid mechanics, which is relevant to the overall multi-physics multi-scale modelling approach.

Fig. 11(b) compares the computation times of a numerical simulation vs. CNN. The comparison is shown at various iterations because in multi-physics the permeability needs to be calculated numerous times. At 320 iterations, CNN is twenty times quicker at providing permeability than the numerical method. Using the latter, it may



(a) (b)

Fig. 10. Results of the extrapolation testing using Test data 2, which has lower roughness. Sequential CNN architecture is shown on the left and *ResNet50* on the right.



(a) (b)

Fig. 11. (a) Convergence results of the numerical Stokes solver, fracture permeability as a function of number of elements. The red node indicates the default value used in this study. (b) Computation time per iteration: numerical simulation vs. CNN.

take approximately 1.5 h to solve for permeability thousand times. Furthermore, numerical computation time grows significantly with the number of iterations increase compared to CNN. To calculate CNN time aforementioned standard laptop GPU was used.

In a simpler problem, deep learning was trained based on fractures with the fluid velocity distribution provided. In this case, calculating permeability using numerical method is straight forward and is computationally cheaper than the data-driven method. However, deep learning is able to provide approximately the same accuracy when trained without the velocity field.

5. Conclusions

Solving multi-physics problems is usually computationally expensive, requiring specialized high-performance computers and elaborate code. This study investigates the ability of machine learning to decrease computation time in challenging multi-physics problems, specifically by investigating whether a deep learning model can quickly and accurately predict rough fracture permeability with complex geometry based on a digital image. The machine learning approach used is a convolutional neural network, which can provide results in seconds compared to numerical simulation, with a high degree of accuracy. It was found that a deep learning model can learn from images of fractures to predict permeability without the fluid velocity field. Providing the latter does not significantly impact the training results. The best results were obtained using the *ResNet50* architecture, which showed a mean absolute percentage error (MAPE) of only ∼ 3%. Furthermore, the difference between the validation loss and train loss is minor during the training, indicating good model generalization and minimal overfitting. Finally, an extrapolation test performed on fractures with significantly different roughness showed high accuracy (∼ 8% MAPE).

The potential of deep learning to predict complex physical parameters quickly and accurately suggests further investigation into its application in multi-physics problems. Future research may involve identifying other physical parameters that can be predicted using deep learning and applying it in settings involving coupled thermo-chemo-hydro-mechanical processes to reduce computation time. Besides, using machine learning in 3D fluid flow problems in porous media has the potential to reveal underlying physical processes.

CRediT authorship contribution statement

Amanzhol Kubeyev: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Funding acquisition, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Computer code availability

https://github.com/ramm777/expfracML.

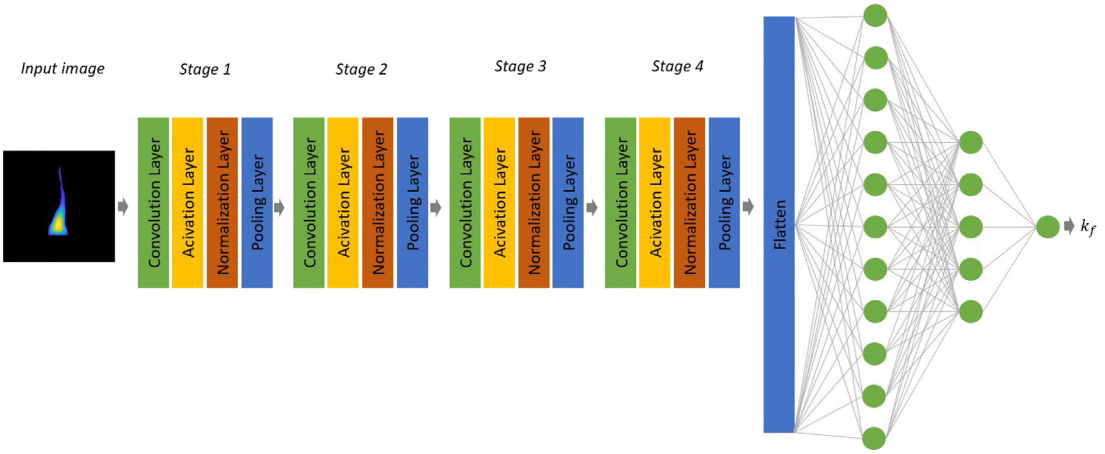


Fig. A.12. An illustrative example of the sequential CNN architecture.

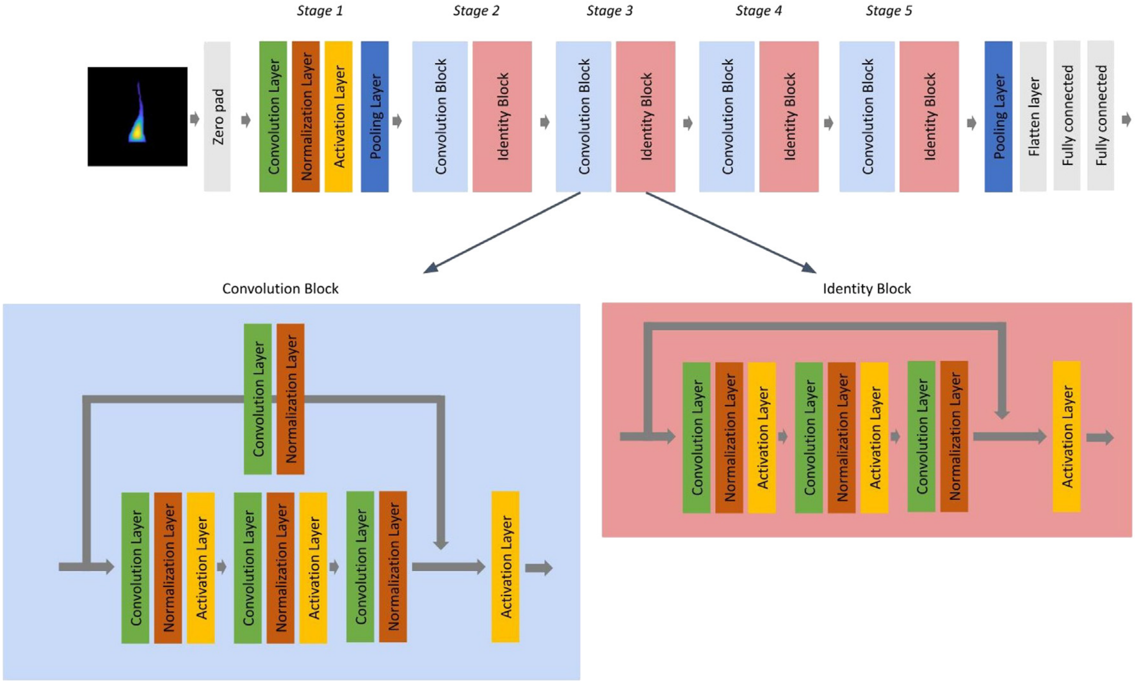


Fig. A.13. An illustrative example of *ResNet50* CNN architecture.

Appendix A. Convolutional neural network architectures

A sequential architecture is shown in Fig. A.12, and consists of four blocks, each having a set of layers placed sequentially: convolutional 2D layer, activation layer, (batch) normalization and a pooling layer. These are attached to the flattening, dense, activation and finally dense layers outputting a continuous permeability value.

The second architecture is a residual network *ResNet50* (He et al., 2016; Dwivedi, 2019) shown in Fig. A.13. The original implementation by Dwivedi (2019) was modified for the regression and used for the analysis. The architecture consists of 6 stages, in the first stage, Convolution, Batch Normalization, Activation and Pooling layers are connected sequentially. Stages 2 to 5 have the same two blocks of layers: Convolution and Identity blocks. Finally, Average Pooling, Flatten and two Fully Connected (Dense) Layer are added, with the last layer outputting a single fracture permeability value.

The reason for the good performance of *ResNet50* is using 'skip connections', which addresses the vanishing gradients problem that the deep networks usually suffer. A vanishing gradient problem is a problem encountered during training when training does not converge to find a minimum loss due to the small gradients in the neurons. If the gradients are small, the weights are not updated, thus no training occurs. Skip connections are designed to jump over layers. For instance, layers in the first (sequential) architecture (Fig. A.12) are connected sequentially, but in the *ResNet50*, the 'skip connection' allows the connection of the first stage to the fourth stage.

To implement skip connections identity and convolution blocks are used. The identity block is used where the input and output dimensions are the same. It has a main path and a shortcut path for the skip connection. The block consists of three components, where two first have Convolution, and Batch Normalization, and Activation layers. The third component has Convolution, Batch Normalization, after which a shortcut is added. Finally, there is an activation function. The convolution block is similar to the identity block, apart from, after the third component, a shortcut which has convolutional and normalization layers inside is added. This block is used when the input and output dimensions are not the same.

References

Andrianov, N., Nick, H.M., 2021. Machine learning of dual porosity model closures from discrete fracture simulations. Adv. Water Resour. 147, 103810.

Asencio-Cortés, G., Morales-Esteban, A., Shang, X., Martínez-Álvarez, F., 2018. Earthquake prediction in california using regression algorithms and cloud-based big data infrastructure. Comput. Geosci. 115, 198–210.

Ashworth, M., 2021. Multiscale Dual-Continuum Modelling of Deformable Porous Media (Ph.D. thesis).

Bhushan, B., 2000. Surface roughness analysis and measurement techniques. In: Modern Tribology Handbook, Two Volume Set. CRC Press, pp. 79–150.

Brenner, M., Eldredge, J., Freund, J., 2019. Perspective on machine learning for advancing fluid mechanics. Phys. Rev. Fluids 4 (10), 100501.

Bright, I., Lin, G., Kutz, J.N., 2013. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. Phys. Fluids 25 (12), 127102.

Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P., 2021. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478.

Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. 52 (1), 477–508.

Cherkassky, V., Mulier, F.M., 2007. Learning from Data: Concepts, Theory, and Methods. John Wiley & Sons.

Chollet, F., 2015. Keras.

Curle, S., Davies, H., 1968. Modern Fluid Dynamics Volume 1. Van Nostrand.

Dwivedi, P., 2019. Residual networks yourself. https://github.com/priya-dwivedi/Deep-Learning/blob/master/resnet_keras/Residual_Networks_yourself.ipynb.

Fischer, P., Dosovitskiy, A., Brox, T., 2015. Image Orientation Estimation with Convolutional Networks. In: Pattern Recognition, Springer International Publishing, pp. 368–378.

Géron, A., 2019. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., 2018. Recent advances in convolutional neural networks. Pattern Recognit. 77, 354–377.

Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R., 2020. A deep learning framework for solution and discovery in solid mechanics: linear elasticity. arXiv preprint arXiv:2003.02751.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hernández-García, A., König, P., 2018. Further advantages of data augmentation on convolutional neural networks. In: International Conference on Artificial Neural Networks. Springer, pp. 95–103.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.

Huang, N., Liu, R., Jiang, Y., Li, B., Yu, L., 2018. Effects of fracture surface roughness and shear displacement on geometrical and hydraulic properties of three-dimensional crossed rock fracture models. Adv. Water Resour. 113, 30–41.

Hubel, D.H., 1959. Single unit activity in striate cortex of unrestrained cats. J. Physiol. 147 (2), 226–238.

Hubel, D.H., Wiesel, T.N., 1959. Receptive fields of single neurones in the cat's striate cortex. J. Physiol. 148 (3), 574–591.

Kampman, N., Maskell, A., Bickle, M., Evans, J., Schaller, M., Purser, G., Zhou, Z., Gattacceca, J., Peitre, E., Rochelle, C., 2013. Scientific drilling and downhole fluid sampling of a natural CO2 reservoir, Green River, Utah. Sci. Drill. 16, 33–43.

Keyence, 2017. Digital microscope VHX-6000 user's manual.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kubeyev, A., Forbes Inskip, N., Phillips, T., Zhang, Y., Maier, C., Bisdom, K., Busch, A., Doster, F., 2022. Digital image-based stress–permeability relationships of rough fractures using numerical contact mechanics and Stokes equation. Transp. Porous Media 141 (2), 295–330.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Lie, K.-A., 2014. An introduction to reservoir simulation using matlab: user guide for the matlab reservoir simulation toolbox (MRST). SINTEF ICT.

Mahendran, S., Ali, H., Vidal, R., 2017. 3D pose regression using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2174–2182.

Milano, M., Koumoutsakos, P., 2002. Neural network modeling for near wall turbulent flow. J. Comput. Phys. 182 (1), 1–26.

Phillips, T., Bultreys, T., Bisdom, K., Kampman, N., Van Offenwert, S., Mascini, A., Cnudde, V., Busch, A., 2021. A systematic investigation into the control of roughness on the flow properties of 3D-printed fractures. Water Resour. Res. 57 (4), ewrcr–25233.

Pyrak-Nolte, L., Morris, J., 2000. Single fractures under normal stress: The relation between fracture specific stiffness and fluid flow. Int. J. Rock Mech. Min. Sci. 37 (1–2), 245–262.

Pyrak-Nolte, L.J., Nolte, D.D., 2016. Approaching a universal scaling relationship between fracture stiffness and fluid flow. Nature Commun. 7 (1), 1–6.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.

Richter, C., Kotz, F., Giselbrecht, S., Helmer, D., Rapp, B.E., 2016. Numerics made easy: solving the Navier–Stokes equation for arbitrary channel cross-sections using microsoft excel. Biomed. Microdevices 18 (3), 52.

Srinivasan, G., Hyman, J.D., Osthus, D.A., Moore, B.A., O'Malley, D., Karra, S., Rougier, E., Hagberg, A.A., Hunter, A., Viswanathan, H.S., 2018. Quantifying topological uncertainty in fractured systems using graph theory and machine learning. Sci. Rep. 8 (1), 1–11.

Torrey, L., Shavlik, J., 2010. Transfer learning.

Tse, R., Cruden, D., 1979. Estimating joint roughness coefficients. In: International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, Vol. 16. Elsevier, pp. 303–307.

Vapnik, V.N., Chervonenkis, A.Y., 2015. On the uniform convergence of relative frequencies of events to their probabilities. In: Measures of Complexity. Springer, pp. 11–30.

Wang, M., Chen, Y.-F., Ma, G.-W., Zhou, J.-Q., Zhou, C.-B., 2016. Influence of surface roughness on nonlinear flow behaviors in 3D self-affine rough fractures: Lattice Boltzmann simulations. Adv. Water Resour. 96, 373–388.

Watanabe, N., Hirano, N., Tsuchiya, N., 2008. Determination of aperture structure and fluid flow in a rock fracture by high-resolution numerical modeling on the basis of a flow-through experiment under confining pressure. Water Resour. Res. 44 (6).

Witherspoon, P.A., Wang, J.S., Iwai, K., Gale, J.E., 1980. Validity of cubic law for fluid flow in a deformable rock fracture. Water Resour. Res. 16 (6), 1016–1024.

Xiong, X., Li, B., Jiang, Y., Koyama, T., Zhang, C., 2011. Experimental and numerical study of the geometrical and hydraulic characteristics of a single rock fracture during shear. Int. J. Rock Mech. Min. Sci. 48 (8), 1292–1302.

Zhou, J.-Q., Hu, S.-H., Fang, S., Chen, Y.-F., Zhou, C.-B., 2015. Nonlinear flow behavior at low Reynolds numbers through rough-walled fractures subjected to normal compressive loading. Int. J. Rock Mech. Min. Sci. 80, 202–218.

Zou, L., Jing, L., Cvetkovic, V., 2017. Shear-enhanced nonlinear flow in rough-walled rock fractures. Int. J. Rock Mech. Min. Sci. 97, 33–45.