An Image-Based Ai Model For Micro-Flow Field Prediction During Resin Transfer Molding

Jean, Jimmy G.; Broggi, Guillaume; Caglar, Baris

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the 21st European Conference on Composite Materials

AN IMAGE-BASED AI MODEL FOR MICRO-FLOW FIELD PREDICTION DURING RESIN TRANSFER MOLDING

Jimmy G. Jean[1], Guillaume Broggi[2] and Baris Caglar[3]

[1]Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft 2629HS, the Netherlands
Email: J.G.Jean@tudelft.nl
[2]Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft 2629HS, the Netherlands
Email: G.Broggi@tudelft.nl
[3]Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft 2629HS, the Netherlands
Email: B.Caglar@tudelft.nl

Keywords: Resin Transfer Molding, Machine Learning, Composite Manufacturing

Abstract

Multiple phenomena occurring at the microscopic scale affect the final mechanical performance of composite parts manufactured through processes involving impregnation of dry fibers, such as resin transfer molding. Formation of fiber-poor areas in specific locations or air entrapment within the resin are issues that commonly arise during the impregnation. Such challenges have motivated the use of numerical simulations to understand the manufacturing processes better and to optimize the process design. However, the limitation imposed by their computational cost has encouraged the use of machine learning (ML) to replace them. Thus far, the state of the art has focused on predicting the permeability of fiber-reinforced microstructures. We expand the limits by proposing an ML-based surrogate for microscale steady-state velocity prediction of a fluid flowing through a fibrous microstructure. This model, inspired by the U-net architecture, takes as input the image representation of fiber-reinforced composite microstructures. It subsequently outputs the resin velocity field around the fibers based on prescribed boundary conditions. Those results are further used to estimate the permeability of the microstructures, thus encompassing previous works. We describe in this work the computational pipeline of our approach, starting from generation of the ground truth data to the optimization of the U-Net hyperparameters.

1. Introduction

The resin transfer molding (RTM) process, through which a composite part is manufactured by impregnating a fiber preform placed in a mold with liquid resin, offers many advantages. It enables the production of composite parts with very complex shapes. It also makes achievable composite parts with good surface appearance in relatively fast cycle times [1]. However, there are many transport issues that affect the process and, by extension, the mechanical performance of the final products. The flow of resin through fiber reinforcements is driven by a combination of viscous forces and capillary forces [2-4]. This dual-scale nature of the flow sometimes results in air entrapment [5] during RTM. These entrapped air bubbles then translate into microstructural defects in the composite material after the resin polymerization reaction during curing [6]. Various factors affect the permeability of fibrous preforms, which include the architecture and stacking of the fabrics [2,7], the deformation of the preform [8].

Owing to these challenges, numerical simulations [9] have long been used as a tool to better understand flow characteristics in microstructures and gain insight into RTM. However, one of their drawbacks is their computational cost [2]. Because of this hurdle, machine learning has been increasing in popularity as a replacement for numerical simulations. It has been well demonstrated that deep neural networks can virtually approximate any function [10]. Moreover, they can significantly increase efficiency [11], compared with numerical simulations. In their state-of-the-art work, Caglar et al. [11] trained a convolutional neural network (CNN) to predict the permeability of 2D microstructures. By additionally using a circuit analogy, they were able to predict the permeability of 3D microstructures with tortuous fibers, which resulted in speedups of two orders of magnitude. In the context of multiscale simulations, microscale permeability predictions are useful for the central areas of rovings where fibers are densely packed. However, only predicting a homogenized permeability is inadequate in resin-rich areas, such as in-between fiber bundles. A more reliable approach is to utilize microscale flow predictions to instead compute mesoscale permeability. To the best of our knowledge, there is currently no work in the literature using machine learning to predict the resin flow in fibrous microstructures. Hence, in this work, we introduce a neural network model trained to predict the resin flow field in microstructures from their image representations. Its architecture is inspired by the U-Net [12], originally designed for biomedical image segmentation. The results show the neural network model's ability to predict with high accuracy the flow field in a wide variety of microstructures.

Next, we will describe the procedure employed to generate microstructures and run the corresponding numerical simulations for ground truth data. Following that, we will introduce the architecture of the ML model used to fit the simulation data. Then, we will report different parameter tests and the corresponding performance of the fitted models. Finally, we will report permeability estimation from the predicted flow field using Darcy's law and evaluate their accuracy.

2. Methodology

2.1. Data Generation

The targeted input and output data for the ML model are as follows. As input, we consider the image representation of fiber-reinforced microstructures. As output, we consider the resulting resin flow field after the application of prescribed boundary conditions. We subsequently detail the process employed to generate such data.

Fiber-reinforced microstructures encountered in practice exhibit variations on several aspects including fiber size, volume fraction, and spatial distribution. We recognize this fact, and hence designed a tool capable of generating microstructures while reflecting this variability. We start by considering a 2D space defined by fiber diameter and volume fraction. We used three values for the fiber diameters: 7, 10, 15 µm. Meanwhile, the fiber volume fractions consist of six values, ranging from 0.2 to 0.7 with an increment of 0.1. Given a fiber diameter and volume fraction, there are limitless possibilities regarding the spatial distribution of fibers in the target microstructures. We take the following steps to account for these variations. We begin with a microstructure with a quadratic arrangement of the fibers. Then, we randomly move each fiber in an arbitrary direction and distance while enforcing non-collision between them. In addition, we enforced periodicity on the geometry of the microstructures. The result is such that we can create multiple microstructures sharing the same descriptors (i.e., fiber diameter and volume fraction). Using this procedure, we generated a total of 5000 microstructures, with an average of approximately 278 per each *fiber diameter-volume fraction* pair.

With the geometry of microstructures properly defined, we then used Gmsh [13], an open-source mesh generator, to construct corresponding meshes. We note that only the fluid domain (i.e., the area between the fibers) is meshed. To run the numerical simulations, we utilized the open-source

OpenFOAM toolbox [14], which is based on the finite volume method. More specifically, we made use of SimpleFoam, which is a steady-state solver for incompressible flow based on the SIMPLE algorithm [15]. We set the fluid flowing between the fibers to be epoxy resin with a density $\rho = 1250\ kg/m^3$ and dynamic viscosity $\mu = 0.5\ Pa \cdot s$. Regarding the boundary conditions, the resin is assumed to enter the domain from the left side and exit it at the right side. The pressure is assumed to be null at the outlet. A pressure drop of $\Delta p = 10\ kPa$ is applied across the microstructure. Cyclic boundary conditions are applied to the top and bottom extremities. It is worth mentioning that the entire pipeline from geometry definition to numerical simulations is fully automated through a Python script. Moreover, additional steps were taken to test the reliability of the numerical results by validating estimated permeabilities for perfectly hexagonal/quadratic fiber arrangements against Gebart's equations [16] using Darcy's law.

2.2. Model Architecture

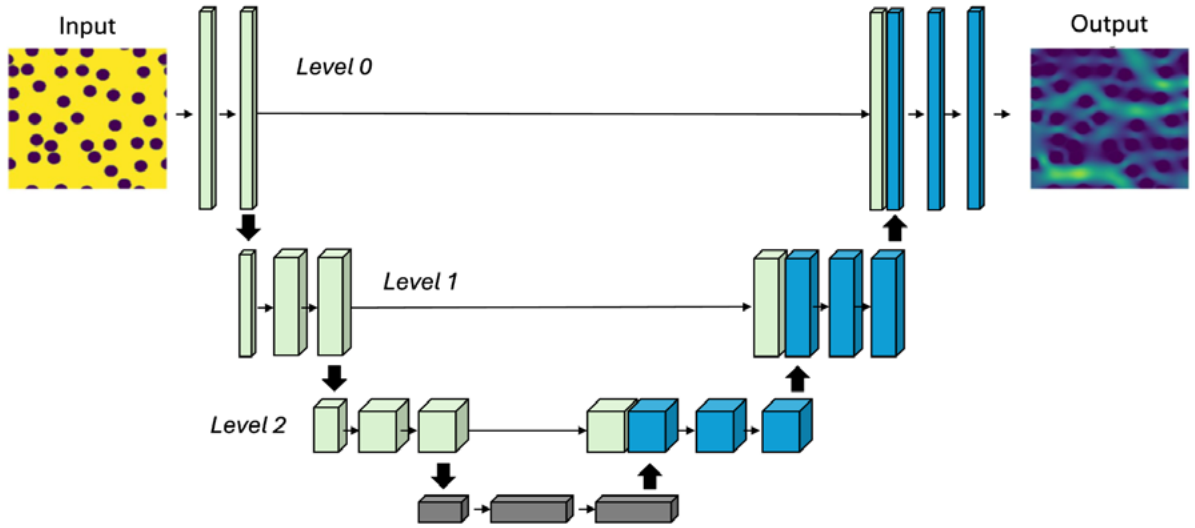

Figure 1. The architecture of the model used is based on the U-Net. It takes the image representation of microstructures as input and returns the corresponding steady-state resin flow field.

We design our neural network (Fig. 1) as a variant of the U-Net [12]. It is a neural network with encoder-decoder architecture, initially developed for the segmentation of biomedical images. In our case however, predicting velocity fields from images pertains to a regression task. We follow the original U-Net architecture in several aspects. Each level in the contracting path of the encoder consists of two convolutional blocks followed by a down-sampling operation. Each block consists of a convolution layer, followed by batch normalization and a Leaky RELU [17] activation layer. The down-sampling operation is a sequence of max pooling (2×2 kernel with a stride of 2), batch normalization, and a Leaky RELU activation layer. This operation reduces the number of channels in the preceding block by half. Each level in the expanding path of the decoder consists of an up-sampling operation followed by two convolutional blocks. The up-sampling operation is a sequence of transposed convolution (2×2 kernel with a stride of 2), batch normalization, and a Leaky RELU activation layer. This operation multiplies the number of channels in the preceding block by two. Skip connections link corresponding levels in the encoder and decoder branch. In the final step before output, an additional convolutional block is added, which performs the final mapping to the velocity magnitude at each pixel. The convolution layer in the final block uses a 1×1 kernel, while all other convolution layers use a 3×3 kernel. Finally, all the convolution layers use zero-padding.

The model takes as input the binary image (256×256 pixels) representation of microstructures, in which pixels in fiber regions take a value of 0, and 1 elsewhere. However, before feeding the input to the convolutional layers, we apply a Euclidean distance transform on it. This operation incorporates

information about the distance to the closest fibers (zero-valued) for each pixel that was originally one-valued. It has been previously shown that this pre-processing step helps to improve accuracy [18]. The model's output is the velocity magnitude field, to which we apply a cost function penalizing the departures of the predictions from target values.

2.3. Training

We implemented the neural network using the PyTorch library [19], which allowed us to take advantage of its automatic differentiation capabilities. To train the neural network, we used the following cost function. We defined the *normalized mean absolute error (NMAE)* as

$$NMAE = \frac{\frac{1}{N}\sum_{s=1}^{N}\sum_{i}\sum_{j}\left|O_{ij}^{s} - T_{ij}^{s}\right|}{\frac{1}{N}\sum_{s=1}^{N}\sum_{i}\sum_{j}\left|T_{ij}^{s}\right|} \quad (1)$$

in which O_{ij}^{s} and T_{ij}^{s} respectively refer to the predicted and target values for sample s at pixel position (i, j). N refers to the number of samples being evaluated. When training the neural network, we updated its parameters using an Adam optimizer [20] and a learning rate of 10^{-4}. Training was performed on a workstation equipped with an NVIDIA® Quadro RTX™ 6000 GPU (24 GB memory).

3. Results

3.1. Hyper-parameter Optimization

We experimented with multiple hyper-parameters to determine the most accurate network configuration. One relates to the number of channels at each level. We tried out three model configurations with different depths, which we refer to as 2D, 3D, and 6D. The number of channels after the convolution blocks at each level is listed in Table 1 for each of them.

Table 1. Number of channels after the convolution blocks at each level of the U-Net.

Model	*Level 0*	*Level 1*	*Level 2*	*Level 3*	*Level 4*	*Level 5*	*Level 6*
2D	64	128	256	-	-	-	-
3D	64	128	256	512	-	-	-
6D	32	64	128	256	512	1024	2048

We fitted the models on the previously described dataset, which we partitioned into training and validation data using an 80%/20% split. A batch size of 40 was used during training. The resulting loss curves are shown in Figure 2. We readily notice that the worst performance comes from the shallowest (2D) model. Increasing the depth of the model has a positive effect. The loss values decrease more quickly. Nevertheless, there is not much difference between the training loss for the 3D and 6D models, which are respectively 0.0609 and 0.0623. The biggest improvement is in the validation loss which decreased from 0.1792 (2D), to 0.1293 (3D), and then 0.0874 (6D).

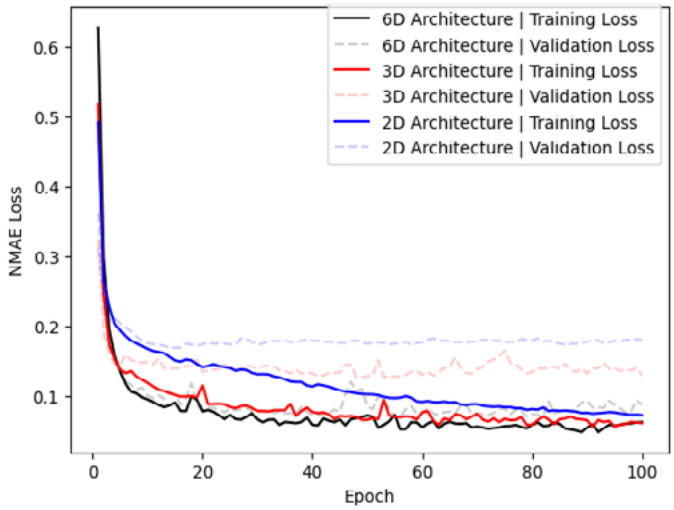

Figure 2. Training results for different model architectures. Increasing the depth of the model results in lower validation loss values.

3.2. Data Augmentation

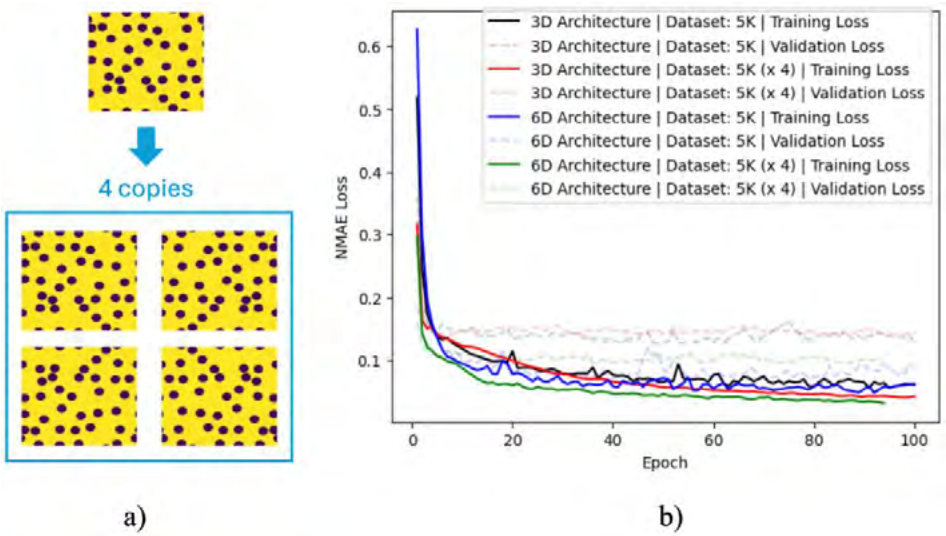

Figure 3. a) Expansion of the dataset by flipping the images in it. b) Training results for the 3D and 6D model architectures when fitted on the original dataset (5000 samples) and augmented dataset (20 000 samples).

Besides comparing different model architectures, we took advantage of the symmetry of the generated microstructures to augment the size of our dataset. The geometry of the microstructures exhibit periodicity both in the left/right and top/bottom sides. We quadrupled the dataset size by flipping the microstructures horizontally and vertically, as illustrated in Figure 3a. Figure 3b shows the training results for the 3D and 6D model architecture. We notice that the training losses diminish in both cases with the augmented dataset. A drop from 0.0609 to 0.0423 was observed in the training loss of the 3D model, and from 0.0623 to 0.0312 for the 6D model. However, we remark that the validation losses did not follow the same trend. The validation loss went from 0.1293 to 0.1429 for the 3D model. Meanwhile, it rose from 0.0874 to 0.0973 for the 6D model. There was a decrease in accuracy in both cases, nevertheless it was not significant. This behavior asks for further investigation.

3.3. Model Performance

3.3.1. Resin Flow Prediction

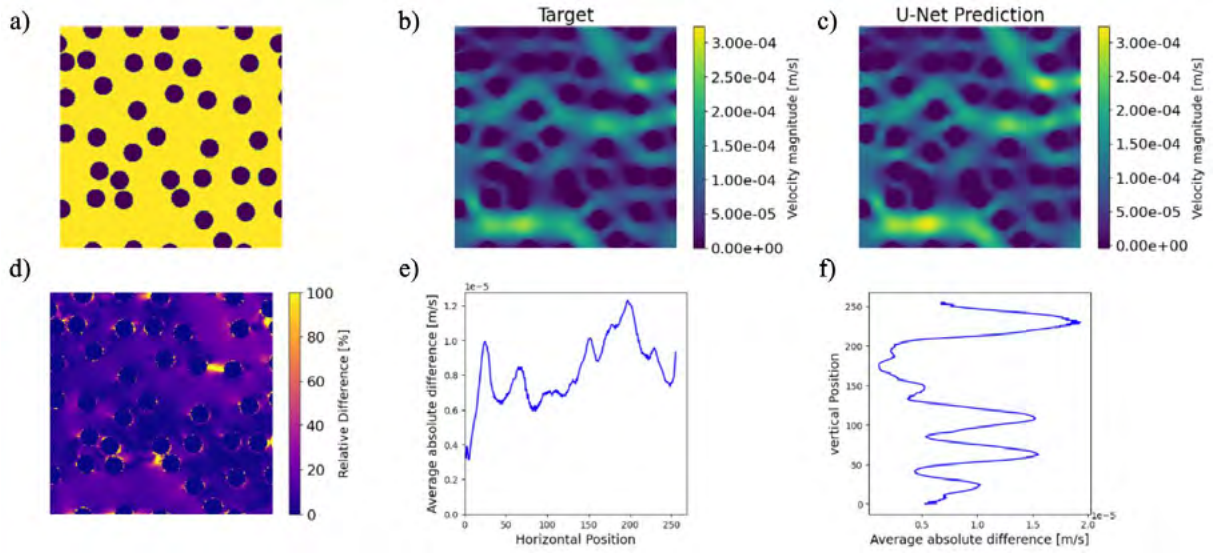

Figure 4. Prediction of the resin velocity magnitude for a microstructure with small fibers (a) in the validation set. The (c) prediction is close to the (b) target obtained via numerical simulation. d) Relative difference between the target and neural network prediction. e) Average absolute difference between pixels (taken over the vertical direction). f) Average absolute difference between pixels (taken over the horizontal direction).

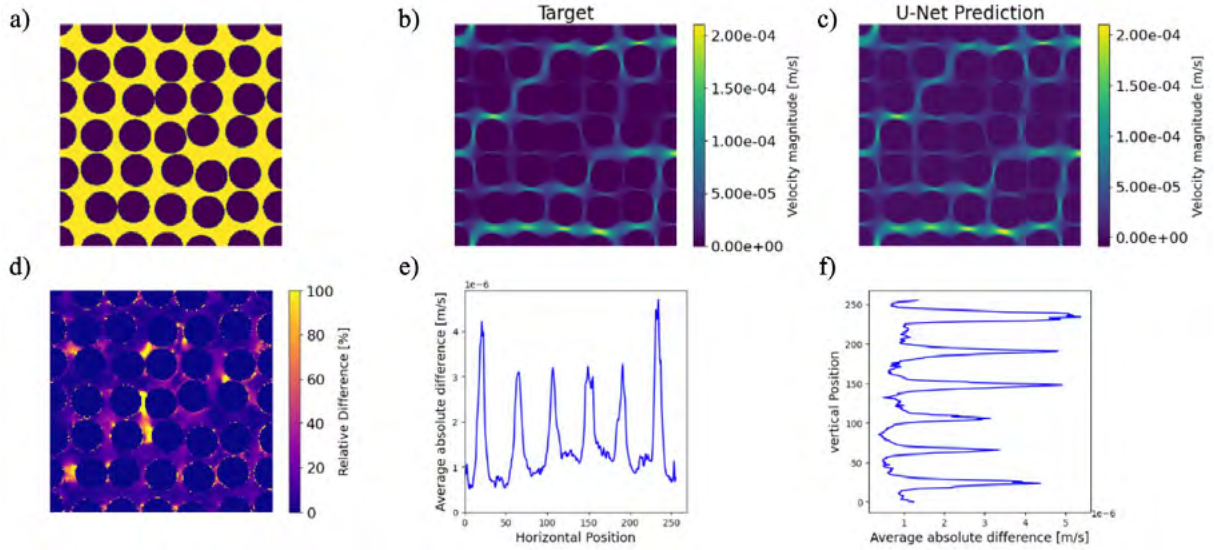

Figure 5. Prediction of the resin velocity magnitude for a microstructure with large fibers (a) in the validation set. The (c) prediction is close to the (b) target obtained via numerical simulation. d) Relative difference between the target and neural network prediction. e) Average absolute difference between pixels (taken over the vertical direction). f) Average absolute difference between pixels (taken over the horizontal direction).

Next, we illustrate the concrete performance of the model in accurately predicting the physics of resin flow in some sample microstructures. For this task, we randomly select two microstructures in the validation set, that the model did not encounter during training. The first one has low fiber volume fraction (Fig. 4a). We notice in Figure 4b-c the close similarity between predictions of the neural network and the target obtained via numerical simulation. Quantitatively, the NMAE loss for this prediction is 0.0749. For a better visualization of the similarity, we look at the relative difference between the two images (Fig. 4d). It can be observed that most of the errors are on the very low end of

the spectrum. Moreover, regions exhibiting large errors (close to/higher than 100%) correspond to areas with very low velocity. We additionally look at the average values of the absolute difference between the predicted and target pixel values, taken along the vertical (Fig. 4e) and horizontal (Fig. 4f) axes. We observed that they are 1 order of magnitude lower than the usual velocity values. Figure 5 shows results for a second microstructure with high fiber volume fraction. The NMAE loss in this case is worse: 0.0809. Similar conclusions can be made as with the first microstructure. Errors are highest in low velocity regions.

3.3.2. Permeability Estimation

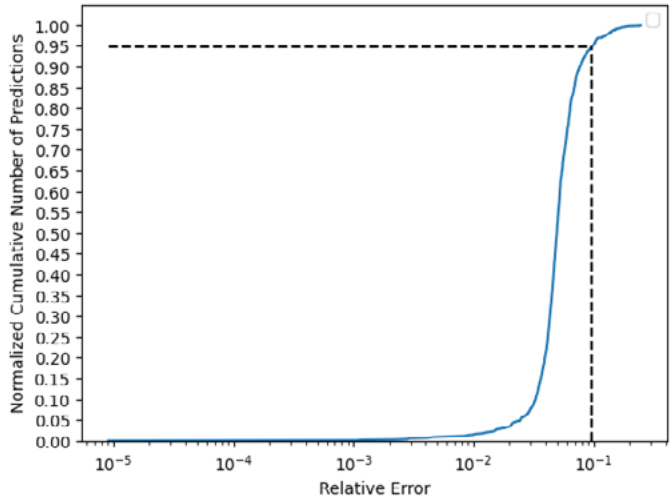

Figure 6. Cumulative distribution of the relative errors on the predicted permeabilities using Darcy's law. The permeability values are derived with high accuracy.

As a further evaluation, we estimate the permeability of the microstructures from the predicted flow fields using Darcy's law. The permeability from Darcy's law is given by

$$k = \frac{\mu Q}{\Delta p} \frac{L}{A}$$

We utilize the velocity values at the inlet (first column of pixels) to estimate the flow rate Q. The pressure drop Δp and dynamic viscosity μ are known from the initial conditions. The microstructure length L and cross-sectional area A are known from the problem geometry. Figure 6 shows the cumulative distribution of the relative errors on the predicted permeabilities for microstructures in the validation set. These relative errors are computed by taking the ratio of the absolute value of the difference between predicted and target permeability to the target permeability's absolute value. We readily notice the accuracy of the predicted values. Quantitatively, 95% of the predictions have a relative error below 10%.

4. Conclusions

We have introduced a neural network capable of predicting the velocity field of fiber-reinforced microstructures given their image representation. The neural network was trained on a multitude of microstructures with varying fiber sizes, volume fraction, and spatial distribution. The results demonstrated the accuracy of the trained model. Moreover, the predicted flow fields were utilized to infer the permeability of the microstructures with remarkable precision. This work thus encompasses previous ones which could only predict the permeability of fibrous microstructures. This work also shows the potential of integrating machine learning in the process simulation for composite manufacturing.

References

[1] Advani, S. G., & Sozer, E. M. (2002). Process modeling in composites manufacturing. CRC press.
[2] Pierce, R. S. (2023, November). Challenges in permeability characterisation for modelling the manufacture of wind turbine blades. In IOP Conference Series: Materials Science and Engineering (Vol. 1293, No. 1, p. 012009). IOP Publishing.
[3] Teixidó, H., Caglar, B., & Michaud, V. (2023). Effect of wettability and textile architecture on fluid displacement and pore formation during infiltration of carbon fibrous preforms. Composites Part A: Applied Science and Manufacturing, 174, 107733.
[4] Teixidó, H., Staal, J., Caglar, B., & Michaud, V. (2022). Capillary effects in fiber reinforced polymer composite processing: a review. Frontiers in Materials, 9, 809226.
[5] Michaud, V. (2016). A review of non-saturated resin flow in liquid composite moulding processes. Transport in porous media, 115(3), 581-601.
[6] Lo, J., Anders, M., Centea, T., & Nutt, S. R. (2016). The effect of process parameters on volatile release for a benzoxazine–epoxy RTM resin. Composites Part A: Applied Science and Manufacturing, 84, 326-335.
[7] Hoes, K., Dinescu, D., Sol, H., Parnas, R. S., & Lomov, S. (2004). Study of nesting induced scatter of permeability values in layered reinforcement fabrics. Composites Part A: Applied Science and Manufacturing, 35(12), 1407-1418.
[8] Loix, F., Badel, P., Orgéas, L., Geindreau, C., & Boisse, P. (2008). Woven fabric permeability: From textile deformation to fluid flow mesoscale simulations. Composites science and Technology, 68(7-8), 1624-1630.
[9] Liu, H. L., & Hwang, W. R. (2012). Permeability prediction of fibrous porous media with complex 3D architectures. Composites Part A: Applied Science and Manufacturing, 43(11), 2030-2038.
[10] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.
[11] Caglar, B., Broggi, G., Ali, M. A., Orgéas, L., & Michaud, V. (2022). Deep learning accelerated prediction of the permeability of fibrous microstructures. Composites Part A: Applied Science and Manufacturing, 158, 106973.
[12] Ronneberger, O., Fischer, P., & 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.
[13] Geuzaine, C., & Remacle, J. F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering, 79(11), 1309-1331.
[14] Jasak, H. (2009). OpenFOAM: Open source CFD in research and industry. International Journal of Naval Architecture and Ocean Engineering, 1(2), 89-94.
[15] Patankar, S. V., & Spalding, D. B. (1983). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In Numerical prediction of flow, heat transfer, turbulence and combustion (pp. 54-73). Pergamon.
[16] Gebart, B. R. (1992). Permeability of unidirectional reinforcements for RTM. Journal of composite materials, 26(8), 1100-1133.
[17] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).
[18] Wang, Y. D., Chung, T., Armstrong, R. T., & Mostaghimi, P. (2021). ML-LBM: predicting and accelerating steady state flow simulation in porous media with convolutional neural networks. Transport in Porous Media, 138(1), 49-75.
[19] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in pytorch.
[20] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.