A Machine Learning Strategy for Race-Tracking Detection During Manufacturing of Composites by Liquid Moulding

Joaquín Fernández-León1,3 · Keayvan Keramati1,2 · David Garoz2 · Luis Baumela3 · Carlos Miguel4 · Carlos González1,2

Received: 8 January 2022 / Accepted: 11 April 2022 / Published online: 16 May 2022 © The Author(s) 2022

Abstract
This work presents a supervised machine learning (ML) model to detect race-tracking disturbances during the liquid moulding manufacturing of structural composites. Race-tracking is generated by unexpected resin channels at mould edges that may induce dry spots and porosity formation. The ML model uses the pressure signals recorded by a sensor network as input, providing a classification of the race-tracking event from a set of possible scenarios, and a subsequent variable regression for their position, size and strength. Such a model is based on the residual network (ResNet), a well-known artificial intelligence architecture that makes use of convolutional neural networks for image recognition. Training of the ML classifier and regressors was carried out with the aid of a synthetically generated simulation data set obtained throughout computational fluid dynamics simulations. The time evolution of the pressure sensors was used as grey-level images, or footprints, as inputs to the ResNet ML. The trained model was able to recognise the presence of race-tracking channels from the pressure data yielding good accuracy in terms of label prediction as well as position, size and strength. The model correlation was carried out with a set of injection experiments performed with a constant thickness closed mould containing induced race-tracking channels. The ability of ML models to provide an approximation to the inverse problem, relating the pressure sensor distortions to the cause of such events, is analysed and discussed.

Keywords Machine learning · Mould filling simulations · Composite materials · Liquid moulding

Introduction
Fibre-reinforced polymer composites (PMCs) are nowadays widely used in applications that require lightweight materials such as those found in the aerospace, automotive, energy, health-care sectors and, among others, sports [1, 2]. However, at present, the dependence of processing on external factors or disturbances is still remarkable in some strategic areas such as aeronautics, aerospace or the automotive sector which means that the number of parts to be inspected and either rejected or reworked produces a considerable impact on the final cost distribution. With that in mind, the cumulative and recurrent costs for the non-implementation of the appropriate diagnosis manufacturing policies that could allow recovery from failure will produce, without any doubt, negative impacts on the competitiveness of the industrial sector in the future.

During the last decade, enormous advances in high-performance computer modelling have enabled concurrent material design, optimisation and a way of manufacturing
through virtual processing (VP) within an integrated computational materials engineering (ICME) framework [3–5]. This is especially relevant in the case of manufacturing of structural composites as the material performance is strongly path dependent. There is a need to incorporate such knowledge from the early stages of the design phase. The future of ICME and, in particular VP, involves a concurrent and seamless hybridisation with data-science and artificial intelligence (AI) in a new modelling paradigm [6]. Such a paradigm will be dedicated to predicting the evolution of material properties through the entire production chain, including processing disturbances, enabling right-first-time concepts and zero-waste production.

Within the family of manufacturing methods for structural composites, liquid moulding (LM), and more precisely, resin transfer moulding (RTM) emerged as the out-of-autoclave technique that best provides high-quality parts at optimised costs. The RTM process starts with a dry fabric preform which is draped and placed into a mould for its impregnation. The RTM process starts with a dry fabric preform which is draped and placed into a mould for its impregnation with a liquid resin driven by a pressure gradient [7, 8]. After the part is totally impregnated, the resin is cured, normally by the action of heat sources until the part becomes solid and can be demoulded. However, one of the major drawbacks associated with RTM arises from the inherent uncertainty as regards the flow patterns produced during impregnation, which are seriously affected by processing disturbances that lead in the end to dry spots and void formation. Such dry spots are generated by unexpected resin channels at mould edges, known as race-tracking channels [9, 10], resulting in non-homogeneous resin flow diverted directly to the outlet gates. In addition, resin velocity variations, mainly attributed to local values of permeability and viscosity, may induce void formation due to unbalanced capillary and viscous forces in textile reinforcements. In the last two decades, efforts have been made in the field to propose automated systems that perform corrective actions after detection of manufacturing disturbances in RTM. The concept of online filling in RTM developed by Advani and co-researchers is an example of such a philosophy [11–14]. These strategies rely on smart use of sensors for flow and pressure detection [15, 16] distributed over the surface of the mould.

AI is opening revolutionary opportunities across engineering disciplines such as continuum mechanics [17] and fluid mechanics, [18] as well as manufacturing [19, 20]. However, the permeation of practical knowledge in manufacturing is still limited and applications are at an initial stage [21–23]. Focusing on structural composites, it is worth mentioning some of the latest contributions in the area [24–27]. Iglesias et al. [24] used Bayesian inference concepts in combination with the inversion method to address the problem of fabric permeability and uncertainties in RTM by using pressure sensors. Successful applications to detect local variations in fabric porosity and permeability in RTM were presented in [26], although the predictions strongly rely on the geometrical approximation of the fluid front position through flow detectors. The detection of regions of dissimilar permeability based on signals recorded by pressure sensors was demonstrated in [25]. These authors used a convolutional neural network (CNN), operating with pressure footprints influenced by the presence of a dissimilar permeability region. CNN regression was able to predict the presence, position, extension and strength of a dissimilar material region based on knowledge gained through the use of synthetic data sets. A different approach was presented recently in [27]. These authors divided a rectangular RTM mould into equal-size rectangular regions through using the synthetic pressure signals obtained by simulation for the detection of permeability dissimilar regions inducing voids. The classification problem was approximated by a gradient-boosting method that provided reasonable results. However, from an industrial perspective, the number of sensors used in this work is still unrealistic and minimisation of use without losing information becomes mandatory.

Machine learning (ML) is a paradigm within AI, which is focused on solving tasks without being explicitly programmed for it. It is the data that allow ML models to begin to learn the specifics of the problem and, throughout trial and error, the model ends understanding the patterns hidden in data. Learning from inputs and outputs can be considered a gradual task continuously improving by experience gains. Somehow, this paradigm imitates the way humans learn. The main purpose of this work is to provide a first experimental exploration of ML methods to detect automatically flow disturbances caused by the presence of race-tracking channels in RTM. The detection capabilities of the model fall on the analysis of pressure changes recorded by a distributed network of only five sensors. The model was developed to address the problem of the presence of race-tracking regions in a squared flat RTM mould. The experimental system was intentionally kept simple with the purpose of exploring ML capacities for diagnosis in manufacturing.

The primary focus of this manuscript is to describe the general ML methodology and correlate it with data collected from the experiments. The basic idea distilled is to demonstrate the possibility of training ML models with only synthetic data extracted from simulations, using them for classification and regression purposes without performing an extensive and costly experimental campaign. Training ML models using only empirical data is prohibitive in many manufacturing systems, so efforts are made to use virtual processing information as much as possible. Future improvements in the ML model will be related with the inclusion of a new portfolio of race-tracking types and/or new processing disturbances scenarios such as non-homogeneous fibre volume fraction, deformations induced in the fabric textile while preforming among others.
The general description of the experimental set-up and the RTM tests with induced race-tracking channels is presented in Sect. 2. The systematic generation of a fully-based simulation data set corresponding to the physical problem is presented in Sect. 3 while the general architecture of the ML model, training and deployment is described in Sect. 4. A general discussion on the accuracy of the model for synthetic and experimental data is carried out in Sect. 5.2. Lastly, some remarks are made and conclusions drawn in Sect. 6.

Materials and Experiments

Resin injection experiments were carried out by using a $600 \times 300 \text{ mm}^2$ closed mould as shown in Fig. 1a. The upper half of the mould was manufactured with transparent polymethyl methacrylate (PMMA) providing optical access to its interior while the other half was steel-made. In order to avoid excessive deflections of the transparent PMMA cover due to the injected fluid pressure, the mould was externally reinforced with a set of four steel beams helping to maintain a constant cavity thickness of 3 mm. Injection experiments were performed by imposing an overall uni-axial flow between the inlet and outlet gates. To this end, a single injection point was placed in the middle of one of the 300 mm edges to guide the flow inside a triangular channel to approximately half of the total mould length. The RTM system was kept as simple as possible to serve as a database for the correlation of the models. The race-tracking channels strongly distorted the fluid flow, providing a double correlation benchmark in which the physical model (Darcy’s flow) and the ML classifier-regressors are validated.

The geometry of such a channel was created simply by using vacuum bag sealant tape. The outlet gate was placed at the opposite 300 mm edge and connected to a resin trap open to the atmosphere. The effective fabric dimensions inside the mould were $L = 300 \times 300 \text{ mm}^2$, as shown in Fig. 1b. The two mould halves were closed with 16 screws generating a uniform clamping pressure on the mould cavity. A silicone seal surrounding the mould contour was used to prevent any fluid leaks during the injection.

The inlet was connected to a pressure pot in which fluid pressure was controlled externally with a compressed-air line. Fluid pressure in the panel was measured with five pressure transducers (Omega PX61V0-100AV). Four of them, namely $P_1$ to $P_4$, were distributed equally in a uniform square array while a fifth sensor, $P_0$, was placed at the fabric entrance, Fig. 1b. The transducers were screwed into the transparent PMMA cover through specific cylindrical cavities machined on it, exposing the testing sensor area to the fluid action without direct contact with the fabric. An additional sensor was used to control the electro-valve activity of the pressure pot, avoiding line pressure disturbances during the tests. All the tests were performed by applying a constant pressure at the injection pot of $P_{in} = 0.7$ bar. The pressure sensor signals were recorded by using a Catman Quantum HBM data acquisition system. A video camera installed on the top of the mould was used to track the flow front evolution for direct comparison with the simulations.

A blend of corn-syrup with water at normal concentrations of $\approx 70 - 30\%$ was used as an injection fluid for all the experiments. The blends were first degassed for 20 min by using a vacuum container. The viscosity of the fluid was measured individually and immediately before the experiments were performed by means of a rotational viscosimeter (Fungilab) with L2 type spindle at 150 rpm. No shear strain rate effects were taken into account, assuming that the fluid behaves as a Newtonian one does. The viscosity values measured ranged between 130 and 160 mPa·s for all the experiments corresponding to slight variations in the blend as well as the laboratory daily temperature.

The fabric lay-up involved stacking four plies of a polyethylene terephthalate (PET) non-woven felt of 280 g/ m$^2$ areal mass with a volume fraction of reinforcement of $\approx 30\%$. Such a combination of the aforementioned corn syrup blend with the PET non-woven felt provided a good
optical contrast for fluid front visualisation purposes. Race-tracking was induced by cutting with a blade lateral channels along the flow direction of \(\approx 10\ mm\) width. Special care was taken to prevent leaks on regions with non-race-tracking edges in which resin flow may divert. In this work, four types of race-tracking types were studied as shown in Fig. 2a. In all the cases, race-tracking channels were introduced at the edges parallel to the flow direction. Type 1 and Type 2 can be considered the same race-tracking type but flipped with respect to the flow direction axis. Type 3 race-tracking contains the two channels starting from the fabric entrance. Type 4 was generated by producing centred channels totally disconnected from the entrance and exit of the panel. A set of four snap-shots corresponding to one case with the Type 3 race-tracking configuration is plotted in Fig. 2b. The upper and lower length of the race-tracking channels was set to \(l_1=0.075\ m\) and \(l_2=0.225\ m\), respectively, for this case. As expected, the fluid progressed faster through the channels which produced a clear deviation of the flow front from the theoretical linear unidirectional case.

The pressure signals corresponding to the five cases analysed (no race-tracking, Type 1, Type 2, Type 3 and Type 4) are presented in Fig. 3. The existence of the sensor \(P_0\) at the fabric entrance requires a specific comment. In all the cases analysed, the controlled pressure at the injection pot was \(\approx 0.7\ bars\) which differed considerably with the readings of sensor \(P_0\) at the fabric entrance. A significant pressure drop of \(\approx 30\%\) was attributed to viscous flow and local losses produced along the flow path from the pot to the fabric entrance making necessary its direct measurement. In addition, the pressure recorded by sensor \(P_0\) was not constant in time and depended on the fabric resistance to flow. This has a considerable effect on the filling time which diverges from the theoretical value obtained under the constant pressure assumption. In this case, by assuming a linear pressure gradient over the length \(L\) of the fabric, and assuming a constant inlet pressure of \(p_{\text{max}}\), the filling time can be obtained by the direct integration of the Darcy equation. Such an integration yields the well-known relation \(t_f^0 = \mu L^2/(2Kp_{\text{max}})\), where \(\mu\) and \(K\) stand for the fluid viscosity and fabric permeability. However, this expression is no longer valid if the inlet pressure is not held in time. If the pressure evolution at sensor \(P_0\) is given by \(p_0(t) = p_{\text{max}}f(t)\), then integration of the Darcy equation is \(\int_0^t f(t)\ dt = \mu x^2(t)/(2Kp_{\text{max}})\).

The proportionality between injected length and time, \(t\propto x(t)\), is no longer fulfilled and the filling time is delayed to \(t_f = \chi t_f^0\). The factor \(\chi\) depends on the shape of pressure build-up function \(f(t)\). Therefore, when \(f(t) = 1\), the classical constant pressure case is recovered with \(\chi = 1\).

The experimental characterisation of the permeability of the fabric \(K\) in the absence of race-tracking channels was determined first, see Fig. 3a. In this case, the flow progressed homogeneously and the readings of sensors \(P_1-P_2\) and \(P_3-P_4\) were similar in time and lay within the experimental scatter. The direct measurement of the filling time as a function of the injected length, see \(x(t)\) length in Fig. 1b, was used for the permeability characterisation. To this end, three tests without race-tracking channels were carried out. The integral of the experimental function \(f(t)\) obtained from sensor \(P_0\) reading as \(\int_0^t f(t)\ dt\) is then represented against the term \(\mu x^2(t)/(2p_{\text{max}})\), with the slope of the relationship being the permeability of the fabric \(K\). A Bayesian marginalisation of said slope, the permeability \(K\), is computed by means of the EMCEE Python implementation of the affine invariant

![Fig. 2](https://example.com/fig2.jpg)  
**Fig. 2** a Race-tracking configurations used in this work and their relative location within the mould, b flow front evolution during injection with race-tracking type three corresponding to induced channel lengths of \(l_1 = 0.075\ m\) and \(l_2 = 0.225\ m\)
Markov chain Monte Carlo (MCMC) ensemble [28]. The probability density function is then obtained, providing the averaged value of $K$ and the 5% and 95% confidence percentiles as $K = 4.6515 \pm 0.0974 \pm 0.0994 \times 10^{-10} \text{ m}^2$.

The pressure evolution curves for the remaining cases analysed are gathered in Fig. 3b to e. The distortion produced by the presence of the race-tracking channels was evident with effects on the maximum pressure, arrival times and pressure build-up. This information is stored in a non-dimensional space in the form of a pressure footprint image as shown in Fig. 4. The footprint is defined as a pixel-based image of $n \times 5$ size in which the vertical axis of the array is the non-dimensional time $\hat{t} = t/t_0$ and the horizontal axis the specific sensor number ($P_0$ to $P_4$). The pixel intensity is assigned to the pressure value $\hat{p} = p/p_{\text{max}}$. The time size of the images was set to $n = 150$, with this value being justified in Sect. 4.
Mould Filling Model

The physical model used in this work is based on the resolution of the Darcy equation for the fluid flow through a porous medium. The Darcy equation establishes a linear relation between the average fluid velocity through the fiber preform $\mathbf{v}(x, t)$ and the pressure gradient $\nabla p(x, t)$, with the proportionality factor being related to the fabric permeability tensor $\mathbf{K}(x)$ and the fluid viscosity $\mu$ as

$$\mathbf{v}(x, t) = -\frac{\mathbf{K}(x)}{\mu} \nabla p(x, t). \quad (1)$$

In this equation, $x$ and $t$ stand for, respectively, the position of a given point in the fabric and the time. Assuming flow continuity, $\nabla \cdot \mathbf{v}(x, t) = 0$, the governing equation for the pressure field can be obtained as

$$\nabla \cdot \left( -\frac{\mathbf{K}(x)}{\mu} \nabla p(x, t) \right) = 0. \quad (2)$$

Initial ($t = 0$) and boundary conditions should be given to determine the evolution of the pressure and velocity fields during the time. Such a problem is normally defined as a moving boundary problem because the flow front position $\Gamma(x, t)$ evolves during the time until the preform is completely filled. Normally, if the position of the flow front is known for a given time $t$, the pressure and velocity fields are determined by standard finite element modelling. Once such information is acquired, an update of the flow front position for time $t + \Delta t$ can be obtained. Several numerical techniques were developed in the past to solve such problems in liquid moulding manufacturing of composites. For instance, the finite element/control volume method uses regions associated with every node to update information regarding the filling factors through the use of the flow rates obtained with the velocity fields. Other ways to simulate mould filling processes are based on the direct solving of the two-phase flow problem by using the Navier–Stokes equations for incompressible fluids. In this case, the continuity equation $\nabla \cdot \rho\mathbf{v}(x, t) = 0$ is accompanied by the linear momentum equation reading as

$$\rho \frac{d\mathbf{v}}{dt} + \nabla \cdot (\rho\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{S}. \quad (3)$$

where $\mathbf{S}$ is known as the Darcy-Forchheimer sink term

$$\mathbf{S} = -\left( \mu \mathbf{D} + \frac{1}{2} \rho |\mathbf{v}| |\mathbf{F}| \right) \mathbf{v} \quad (4)$$

and where $\mathbf{D}$ and $\mathbf{F}$ stand for material parameters. The second term in Eq. (4) is related to inertial effects which are negligible for the case of liquid moulding of composites under a low Reynolds number flow. In this situation, the inertial terms related to the velocity can be neglected, recovering the standard Darcy equation with the factor $\mathbf{D}$ being the inverse permeability of the fabric. In two-phase flow, the interface between the two fluids, namely resin and air in liquid moulding, is tracked by means of the volume of fluid (VOF) approach by using $\alpha$ as a phase variable. This variable ranges, respectively, between $\alpha = 1$ and $\alpha = 0$ for the resin and air fluids. Within this approach, the $\alpha$ variable is continuously updated during simulation time by using the advection equation given by

Fig. 4 Pressure footprints for: a no race-tracking, b race-tracking Type 1, c race-tracking Type 2, d race-tracking Type 3, e race-tracking Type 4. Each image is formed by $150 \times 5$ pixels corresponding to five sensors ($P_0$ to $P_4$) and $n = 150$ temporal divisions.
\( \frac{da}{dt} + \nabla \cdot (a \mathbf{v}) = 0 \) \hspace{1cm} (5)

OpenFoam (Open-Source Field Operation And Manipulation) [29] is a free open-source computational fluid dynamics (CFD) software that can be used to solve the problems related to filling in the liquid moulding of composites. OpenFoam includes interFoam a solver for two-phase incompressible, isothermal and immiscible fluids, tracking interfaces with the VOF method. Details and performance of the algorithms used to track interfaces, as well as pressure and velocity solvers for two-phase flow, can be found in [30].

**Machine Learning Model**

**Data Set Generation**

OpenFoam provided the massive synthetic data required for the ML model. Given that obtaining such a data set size from only experiments is prohibitive, they were replaced by numerical simulations. Computations were carried out in a non-dimensional space by dividing length, time and pressure by \( L, T, p_{\text{max}} \), respectively, so that the model could be used for different panel size \( L \) fabrics, permeability \( K \) and fluid viscosity \( \mu \). Macroscopic unidirectional flow was induced in a unit square fabric of \( 1 \times 1 \times 1/100 \) by imposing a time-dependent pressure boundary condition at the inlet \((x = 0)\) following \( p_{\text{in}}(t) = 1 - e^{-\gamma t} \) while the outlet pressure was kept at \( \hat{p} = 0 \) \((x = 1)\). The time-evolution parameter \( \gamma \) controls the pressure build-up in sensor \( P_{\text{in}} \) in non-dimensional time. The pressure build-up at the inlet was introduced to follow experimental observations being the exponential function with a single parameter \( \gamma \) good enough to capture the general evolution. However, additional terms can be added to the model without loss of generality. Slip-free conditions were applied to the remaining faces of the mould. Race-tracking channels were modelled as dissimilar regions of width \( 1/30 \) and length \( l_1 \) and \( l_2 \) with isotropic permeability given by \( K_{\text{inh}} = K/\beta \). \( K \) is the permeability of the bulk fabric while \( \beta = K_{\text{inh}}/K \) the race-tracking strength. A uniform brick cell discretisation of the domain was used with in-plane dimensions \( 1/100 \) and a single cell in the mould through-the-thickness direction. The models contained a total of 10000 brick cells which were judged fine enough to capture accurately the flow front position evolution during time, being the effect of discretisation negligible at the pressure sensor points. Viscosity \( \mu \) and permeability \( K \) were arbitrary chosen in this approach to provide \( \rho_{\text{ff}} = 1 \) for the non-dimensional space.

A four non-dimensional tuple formed by \( X = [l_1, l_2, \beta, \gamma] \) defines the characteristics of an individual non-dimensional OpenFoam case used for the synthetic data set generation. The \( X \) variable space was uniformly covered by means of the Latin hypercube sampling technique by means of the PyDOE software package [31]. All the regression variables were assumed to follow uniform probability distributions: \( U_{l_1} = U_{l_2} = U(0.05, 0.95) \), \( U_{\beta} = U(10, 2000) \) and \( U_{\gamma} = U(2, 25) \). The computation time was kept constant for the simulations and equal to \( t_f = \chi r_f^0 \) with \( \chi = 1.5 \). The latter value of \( \chi = 1.5 \) corresponds to the resolution of the integral equation \( \int_0^\chi f(\tilde{t}) \, d\tilde{t} = 1 \) for \( t_f \) with \( f(\tilde{t}) = 1 - e^{-\gamma \tilde{t}} \) and \( \gamma = \min(U_{\gamma}) \). This number reflects the influence of the pressure build-up on the filling time delaying it \( \approx 50% \) in the worst situation (corresponding to \( \gamma = 2 \)). As a consequence, the size of the footprint images was set to \( n = 150 \times 5 \) with 150 pressure points distributed in the non-dimensional time \( \tilde{t} = [0, 1.5] \).

The set of simulations with different four-tuples \( X = [l_1, l_2, \beta, \gamma] \) for each of the types of race-tracking, as well as without it, was generated with OpenFoam. The automation of the computational process was carried out with PyFoam, a Python library that manipulates and control OpenFoam running cases. Instances of the problem corresponding to each of the four-tuples were generated by modifying the input parameters of the topoSetDict and controlDict dictionaries. Once an individual simulation was finished (approximately in 180s in a standard computer), the pressure probes data at the position sensor \( p_0(t) \) of the experiments were stored. However, the particular computational time may significantly increase for other complicated geometry and boundary conditions problems. The process was followed with a new job submission until the whole data set was created. A total of 4000 OpenFoam simulations were run to serve as a database for model benchmarking. Data sets were serialised into a Python pickle, providing easy access for subsequent ML training tasks.

**Machine Learning Classifier-Regressor Structure**

ML scripts were built by using the end-to-end open-source platform PyTorch [32]. In this work, among those available in the published literature, a kernel architecture made with the deep learning Residual Network (ResNet) [33] was selected. Similar results with other top-performing DL models, such as DenseNet [34] or EfficientNet [35], could be obtained; however, the comparative performance analysis is beyond the scope of this work. The ML model proposed here is a combination of two basic and subsequent operations. First, a classifier that predicts the race-tracking labels (no race-tracking, Type 1, Type 2, Type 3, Type 4) as a function of the pressure footprints and, second, a regressor that provides an approximation for the four-tuple.
ResNet is developed by mimicking pyramidal cells existing in the cerebral cortex with skip connections jumping over convolutional layers. Convolutional networks present some degree of inconvenience with regard to back-propagation and gradient vanishing. This happens when the depth of the network is considerably large, meaning that the back-propagation of gradients shrink to zero after several training epochs. As a result, network weights are not updated properly and learning is not produced. In order to address this, the ResNet architecture introduced skip connections between layers. The structure of the ResNet model used in this work is presented in Fig. 5. It is composed of three subsequent block operations from the input images of the pressure footprints: convolution $\text{Conv2D}$, batch normalisation $\text{BN}$ and activation with a Rectified Linear Unit $\text{ReLU}$:

- **Convolution2D ($\text{Conv2D}$)**. This corresponds to an image operation based on the application of a given set of filters to enhance specific features of the image. The 2D convolutions of an individual image may be obtained, namely $\mathbf{B}$, from an input image, namely $\mathbf{A}$, by applying the kernel function $\mathbf{F}$, as

$$\mathbf{B}(i,j) = \mathbf{F} \ast \mathbf{A} = \sum_k \sum_l F_{k,l} A_{i-k,j-l}$$  \hspace{1cm} (6)

where $\mathbf{F}$ stands for the filter applied and $\ast$ the convolution operator. In this work, a total of 17 convolutions with kernel-size between $1 \times 1$ and $3 \times 3$ size were applied in the convolution, using $\text{stride}=1$ and $\text{stride}=2$ (which informs about the step pixel displacement through the image) and $\text{padding}=1$ to avoid image trimming layer on the image edges.

- **Batch normalisation ($\text{BN}$)**. This seeks to alleviate the movements produced in the distributions of internal nodes of the network with the intention of accelerating training. Those movements are avoided via a normalisation step, constraining means and variances of the layer inputs and saving them during the training process. Furthermore, it reduces the need for dropout, local response normalisation and other regularisation techniques [36].

- **Rectified linear unit ($\text{ReLU}$)**. This is the activation function $g(x) = \max(0, x)$ used to introduce non-linearity between layers.

- **Average pooling ($\text{AP}$)**. This is applied to reduce the dimensions of the convolution filtered images with the purpose of obtaining more efficient and robust characteristics. The model uses a $4 \times 4$ pooling filter after the last convolution block and before the fully connected layers, taking the averaged value of the pixels in the neighbourhood as the result for a given point.

**Fig. 5** Sketch of the ResNet structure used. Each block is composed by a set of operations performed on input images (convolution, batch normalisation, average pooling, rectified linear unit and fully connected neural network). skip connections add the input signal $x$ to the transformed one by a sequence of $\text{Conv2D}+\text{BN}+\text{ReLU}+\text{Conv2D}+\text{BN}$ operations.
The footprint images of 150 × 5 are reshaped into 28 × 28, adding a padding to match the dimensions. The new reshaped images are first subjected to a first set of operations (Conv2D, BN and ReLU) to increment the number of features before using the deep residual connections. At this point, the results of this first operation set are submitted to four sequential layers which contain two blocks with internal skip connections. All the mentioned layers end with an averaging pooling AP which averages the resulting values. Lastly, a set of features is received as a result of the previous operations which feed a fully connected neural network with 128 neurons directly linked to either the class label (no race-tracking, Type 1, Type 3, Type 4) or four-tuple ($\hat{l}_1, \hat{l}_2, \hat{\beta}, \hat{\gamma}$) regression parameters.

Each of the blocks in Fig. 5 contains the aforementioned ResNet skip connections. The input skips the block layers and feeds the layer prior to the entrance in the activation function to alleviate the vanishing gradient problem [37]. By adding the input image $x$ to the block-transformed value with Conv2D+BN+ReLU+Conv2D+BN operations, namely $h(x)$, the composed response obtained $H(x) = h(x) + x$ is less sensitive to the vanishing gradient. as $H(x)$ is close to the unity. Thus, given that the deep network with skip connections learns close to identity layers it produces an optimum configuration.

The race-tracking classifier and each of the regressors used were trained individually on an NVIDIA RTX 3090 24 GB GPU system. Minimisation was carried out with the stochastic gradient descent (SGD) algorithm by using the cross-entropy (CE) and mean squared error (MSE) loss functions for, respectively, the optimisation process of the classifier and regressor. The first question to address is to define the number of simulations necessary to ensure model accuracy and the generalisation level. To this end, a data set containing 4000 simulations was generated following the proposed approach and, then, it was randomly divided in two parts corresponding to training and test sets following a ratio of 75/25. The training set is presented to the model to learn patterns while the test set is used to evaluate it with never-seen data while. Subsets of 100, 500, 1000, 2000 and 3000 simulations were randomly extracted from the training set validating them over the same test set to determine the model variances. The model is said to generalise when the losses corresponding to training and test data sets are close together within a numerical scatter band. Each subset was trained until asymptotic loss behaviour in CE and MSE was obtained which occurred typically after 400 epochs. The CE and MSE obtained for each of the subsets in the last 20 training epochs was recorded, with the mean value and variances being in the form of box-and-whisker plots presented in Fig. 6a, b. The CE and variance diminished strongly when the data set contained more than 500 simulations, indicating that the information embedded in the pressure footprints was enough for classification purposes. However, in the case of the regression variables, the trends were slightly different. Figure 6b shows the MSE corresponding to the race-tracking Type 1 with the different subset size. Once again, the model was trained to a highly effectively degree with a very small MSE even when the data set used involved 100 or 500 simulations. However, from the results obtained, it could be argued that generalisation occurs only with a minimum of 2000 simulation data sets. Therefore, the analyses presented in the following sections are obtained with such a data set size. Still, the data set can be reduced even to 500 per condition without significantly impacting the error. A compromise between the error obtained in the regression with the synthetic data and the experimental scatter should be given.
Machine Learning Classifier and Regressor Accuracy

Synthetic Data Machine Learning Accuracy

Although not presented here, the ResNet confusion matrix for the race-tracking types corresponds to almost perfect identity. Such a confusion matrix compares the labelling predictions of the classifier with the truth value used as input. From the ML model perspective, the classification problem of race-tracking produced no misleading results and the accuracy level was close to 100% without representative values of false positives and negative values. It should be remarked that such a level of accuracy is attained with the only information given by five sensors equally distributed over the mould surface.

The comparisons between the ground truth and the predicted values of the race-tracking regression variables are gathered in Fig. 7a to c. For the sake of simplicity, only data corresponding to the race-tracking case Type 1 is presented. Training (in red) and test (in green) data sets were included in the comparison. It should be noted that the test set corresponded to never-seen data which was not used for model training. The results for \( \hat{l}_2 \) were not included in the figure because of the similarity to those obtained for \( \hat{l}_1 \). Again, accuracy plots for the remaining race-tracking cases were omitted in the paper for simplicity. The relative regression deviations \( \frac{X_{\text{pred}} - X_{\text{truth}}}{X_{\text{max}}} \) computed by using only the test data set (green dots) were quantified by means of the probability density functions and the results plotted in Fig. 7d. The distributions followed the expected trend centred on the truth values \( X_{\text{truth}} \) with a slight skew. The \( \hat{l}_1, \hat{l}_2 \) and \( \hat{\gamma} \) distributions were narrower than the \( \beta \) one. Such distributions were used to construct the 5-95% confidence offsets to the one-to-one slope for Fig. 7a to c. Maximum deviations of \( \approx 3.6\% \), \( \approx 14.2\% \) and \( \approx 1.48\% \) were found for \( \hat{l}_1, \beta \) and \( \hat{\gamma} \) for this confidence interval. Thus, it can be assumed that the accuracy level meets the requirements, at least from the synthetic data, and the classifier and the set of regressors could be used to ascertain the race-tracking classes and regression parameters. However, the experimental accuracy should first be evaluated.

Fig. 7 a to c ResNet regression results for the parameters \( (\hat{l}_1, \hat{l}_2, \hat{\beta}, \hat{\gamma}) \) and race-tracking Type 1 (values for \( \hat{l}_2 \) were not included for the sake-of-clarity), d Probability density functions of the deviations of predicted values with respect to ground truth \( \frac{X_{\text{pred}} - X_{\text{truth}}}{X_{\text{max}}} \). Offsets to the one-to-one slope in (a to c) were included according to the 5-95% confidence interval for each of the variables.
Experimental Data Machine Learning Accuracy

The experimental values of the pressure sensors for different race-tracking cases were presented to the trained ResNet ML model to ascertain the accuracy level obtained. To this end, the experimental pressure and time were first projected to the non-dimensional space by dividing them with the corresponding values of $p_{\text{max}}$ and $l_{\text{ff}}$, respectively. The pixel grey level of the images was also projected to share the same pixel dimensions $150 \times 5$ as those used for the ResNet training set. The pixel grey level of the images corresponding to the time value of the non-dimensional pressure $\hat{p}(t)$.

Experimental footprints were first presented to the trained ResNet classifier. Again, the operation resulted in no labelling errors for the experimental data set composed by the no race-tracking, Type 1, Type 2, Type 3, Type 4 tests. The pressure footprint features extracted by the ResNet ML model were adequate for classification purposes given the pressure sensor footprints.

This latter result on the classification problem deserves additional comment. When training the ResNet model with synthetic data, the accuracy level obtained in the classification tests was very close to 100%. It should be noted that similar results can be obtained when other less sophisticated models, such as decision trees are used. However, those models fail when trying to predict the class corresponding to the experimental results. In addition, regressing the parameters of the position, size, and strength of the race-tracking in an actual situation is a challenging task requiring modern deep learning techniques such as ResNet.

After classification, each regressor was applied to the experimental results, yielding the corresponding non-dimensional four-tuple $\hat{l}_1, \hat{l}_2, \beta$ and $\gamma$. The results were back-projected to the experimental testing space, $l_1, l_2, \beta$ and $\gamma$, with the appropriate fabric dimensions $L$ and filling time $l_{\text{ff}}$. The numerical results obtained are gathered in Fig. 8, including the ResNet predictions and the truth experimental values of the race-tracking variables. Clearly, the accuracy in the regression parameters was modest as compared with the classification problem but lay within the expected range. The values of $l_1, l_2$ were compared with the imposed race-tracking lengths in Fig. 8a and b. The ResNet prediction for the parameter $\gamma$ of the exponential pressure build-up at the fabric entrance was compared with that obtained by direct fitting of the $P_0$ sensor pressure signal, Fig. 8d. Unfortunately, the race-tracking strength $\beta = K_{RT}/K$ cannot be compared with any imposed value in the test. Only crude comparisons can be made by using the theoretical equivalent race-tracking permeability ($K_{RT} = \frac{\gamma}{2}\int_{0}^{\infty} \frac{\tanh(\gamma/2h)}{P_0} \, \text{d}h$) for a rectangular duct of width $w$ and height $h$ under Stokes flow. For instance, by assuming a channel depth of $h = 3$ mm and channel widths ranging in $w = 1 - 10$ mm, the race-tracking strength $\beta$ ranges between 140 and 1300 being these factors similar to those found during the regression of the $\beta$ variable. Such a reference value is presented in Fig. 8c but only for comparison purposes.

In summary, the level of accuracy found was reasonable for regression of the variables given that they were predicted with the only information of five pressure sensors distributed on the mould surface. Of course, it could be argued that better estimates could be obtained by installing sensors at specific positions in order to gain sensitivity.

![Fig. 8](image)

ResNet predictions (orange) and truth values (green) for the four regression variables: a $l_1$, b $l_2$, c $\beta$ and d $\gamma$ and race-tracking cases. The values were back-projected to the physical variable space using panel length $L$ and filling time $l_{\text{ff}}$. © Springer
After the regressions, OpenFoam simulations were run again by using the predicted results presented in Fig. 8. The numerical values of the evolution of the sensor pressure signals are plotted in Fig. 9 and compared with the corresponding experimental results. The shape of the curves was accurately captured, and agreement includes relevant features such as the shape of pressure build-up and fluid arrival time. The degree of agreement in the case of race-tracking Type 4 was modest and the main reason was attributed to the lack of prediction of pressure build-up at sensor $P_0$. It should be remarked that such a sensor signal was assumed to follow a single-parameter exponential function ($p_0(t) = 1 - e^{-\gamma t}$ in the non-dimensional space) which, in this case, deviates from the experimental values as already noted in Fig. 8d. Better predictions could be potentially obtained by including additional parameters.

Fig. 9 ResNet predictions (discontinuous lines) and experimental results (dots) for the pressure evolution of sensors $P_0$, $P_1$, $P_2$, $P_3$, $P_4$: a no race-tracking, b Type 1 with $l_1 = 0.075$ m and $l_2 = 0.150$ m, c Type 2 with $l_1 = 0.075$ m and $l_2 = 0.225$ m, d Type 3 with $l_1 = 0.075$ m and $l_2 = 0.225$ m and e Type 4 with $l_1 = 0.150$ m and $l_2 = 0.075$ m. Dimensions according to Fig. 2a. ResNet results were back-projected to the experimental space by means of $p_{\text{max}}$ and filling time $t_0$. 

© Springer
for the description of the pressure build-up at the fabric entrance, but this analysis is considered out of the scope of the present work and will be included in future research.

The direct comparisons of the snap-shots, numerical and experimental, of the front flow evolution for different times are presented in Fig. 10a to d. The numerical results were overlaid in a transparency mode so that the direct comparison with the experiments could be carried out easily. The experimental flow front position is highlighted with a white dotted line in all the snapshots. Agreement between simulation and experimental front flow evolution was notable in shape and arrival time to specific positions bearing in mind the discrete information recorded by the sensors.

**Conclusions**

A new, simple and robust methodology based on ML algorithms for race-tracking and non-homogeneous flow detection is presented in this work. The presence of race-tracking channels that may induce dry spots and voids is a common processing disturbance that occurs during liquid moulding of structural composites. The presence and strength

---

**Fig. 10 a to d** Experimental and numerical snap shots of the evolution of the front flow for the race-tracking Type 1, 2, 3, 4 cases. Numerical simulations were carried out with OpenFoam using the ResNet regression parameters obtained from the experiments and are presented in Fig. 8. Results were back-projected to the experimental space with the fabric length $L$ and filling time $t_{ff}$.
of race-tracking channels were detected from the signals recorded by a discrete network of five sensors distributed over the mould surface. The readings obtained from pressure sensors were used as inputs of the ML model, delivering the race-tracking class among a set of possible scenarios (no race-tracking, Type 1, Type 3, Type 4) and regression variables for geometry (length and position) and strength. In this work, pressure sensors were preferred over standard flow sensors because of the superior information captured by the first. Flow sensors only record the time the resin passes through a given position. The pressure sensor provides, in addition, information on the resistance to flow in terms of pressure build-up.

It is often assumed that ML models require a significant amount of data for training purposes. This stringent limitation is reduced if physics-informed ML is used instead [6]. In the approach offered by this paper, it was decided to replace experiments by numerical simulations obtained through the extensive use of computational fluid dynamics.

The CFD forward model based on Darcy’s flow through porous medium permits the complete generation of a set of pressure footprints from the type of race-tracking event, size, position and strength. However, the engineering challenge involves the approximation of the inverse problem relating the distortion of the pressure footprints to its cause or manufacturing disturbance. This was addressed by using a ML model based on Residual Network (ResNet) which is one of the current outstanding convolutional network architectures used for image analysis and recognition. By applying a sequence of residual blocks operating on the input footprint images, our model was able to learn how to classify race-tracking problems among a set of different possible scenarios and provide insights into the extension and intensity of the disturbance event. The ML model was trained with 2000 synthetic data obtained through OpenFoam.

The accuracy of the ML model was also checked with both synthetic and experimental data. The correlation against synthetic unseen data not used during model training (known as test data set) was excellent in both classification and regression tasks. The confusion matrix obtained with the OpenFoam synthetic data was almost diagonal and the accuracy of the regression parameters was below ≈ 4% for all the variables with the exception of the race-tracking strength $\beta$. In this latter case, the maximum error from the test data set was ≈ 15%.

The ResNet predictions were also good when compared with experimental results. In this case, the experimental pressure footprints were used as inputs of the ML, yielding the race-tracking class and the regression parameters. The classification problem was again perfectly solved without labelling errors, although care must be taken because of the limited number of experimental tests available. The regression accuracy when compared with experimental results was modest, and errors in the prediction of the race-tracking lengths and strength of ≈ 40% were obtained in some cases. A number of factors may influence such a loss of accuracy: among them the most relevant ones are experimental scatter associated with stochastic fabric permeability and model inadequacy.

As mentioned previously, the primary purpose of the paper was to demonstrate the possibility to train a model with synthetic data generated with a standard computer fluid dynamics software, such as OpenFoam, and use it to detect critical disturbances occurring during the liquid moulding manufacturing of structural composites, such as a race-tracking situation. Thus, the challenge was to prove whether modern machine learning models, trained exclusively with synthetic data, can generalise and predict race-tracking conditions in a real experiment.

### Perspectives and Improvements

In order to overcome some of the limitations that we have encountered in the present model, the following future improvements can be introduced:

- **Data set hybridisation.** In the current situation, data sets are built exclusively by using CFD simulations. All the prediction capabilities of the model rely on accurate modelling of the physics involved in the process. However, it will be possible in the future to include experimental results by developing hybrid data sets containing experimental and simulation data. Experimental data will naturally incorporate into the ML model such inadequacies/discrepancies resulting when the current physics of impregnation does not explain all the experimental facts.

- **Optimisation of sensor positions to improve the resolution of the technique.** The pressure sensors were distributed homogeneously on the mould surface in this work. Considering that the effect of the race-tracking channels depends on the distance between the sensor and the disturbances, it is clear that sensor positions could be a matter of improvement. The paper introduces a general methodology for the analysis that permits wrapping it into optimisation algorithms that would automatically provide a solution.

- **The ResNet structure used in this work was trained and experimentally validated with only four different race-tracking scenarios which can be understood as a small set of an infinite number of possibilities. Although this is true, ML can incorporate incremental training so additional data sets will be included increasing the predictability of the model.** Such different scenarios may include other race-tracking channel events, the generation of regions with different permeability caused by local...
variations in volume fraction, deformations induced in the textile preform or uneven mould clamping pressure.

- Without the lack of generality, the present model is able to provide useful information about possible processing disturbances occurring during liquid moulding of composites by learning from pressure sensor signals. Pressure signals are immediately analysed after manufacturing providing an instant classification and regression of the race-tracking problem which can be considered a diagnosis procedure. However, the final intention will be the continuous interaction with the injection system for applying the necessary on-the-fly corrective actions to alleviate the defects induced by the presence of processing disturbances.

- Trustworthy ML. Industrial applications require the use of trusted ML. It is necessary that the ML system delivers not only an accurate and robust response in terms of classification and regression but to rely on how algorithm decisions are taken. In this respect, physics-informed ML constrains responses by aligning them with well-known physics laws [6].

Acknowledgements The research leading to the results presented in this paper received funding from the Regional Government of Madrid through the research and innovation Innovation Hubs 2018 through the TEMACOM project (49.520635.9.18) and MAT4.0-CM (S2018/NMT-4381) projects. The authors would like to thank Dr. Davide Mocerino for his useful discussions throughout the work.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. González C, Vilatela J, Molina-Aldareguía J et al (2017) Structural composites for multifunctional applications: current challenges and future trends. Prog Mater Sci 89:194–251. https://doi.org/10.1016/j.pmatsci.2017.04.005
2. Llorca J, González C, Molina-Aldareguía J et al (2011) Multiscale modeling of composite materials: a roadmap towards virtual testing. Adv Mater. https://doi.org/10.1002/adma.201101683
3. Liu X, Furrer D, Kusters J, et al (2018) Vision 2040: a roadmap for integrated, multiscale modeling and simulation of materials and systems. NASA/CR—2018-219771 Report
4. Kalil T, Whitman L (2016) The materials genome initiative: the first five years
5. Furrer DU, Dimiduk DM, Cotton JD et al (2017) Making the case for a model-based definition of engineering materials. Integr Mater Manuf Innov. https://doi.org/10.1007/s40192-017-0102-7
6. Karpatne A, Atluri G, Faghmous JH et al (2017) Theory-guided data science: a new paradigm for scientific discovery from data. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2017.2720168
7. Advani S, Hsiao KT (2012) Introduction to composites and manufacturing processes. In: Advani SG, Hsiao KT (eds) Manufacturing techniques for polymer matrix composites (PMCs). Woodhead publishing series in composites science and engineering. Woodhead Publishing, pp 1–12. https://doi.org/10.1533/9780857092581.1.1. http://www.sciencedirect.com/science/article/pii/B9780857096500018
8. Rudd CD, Long AC, Kendall KN, et al (eds) (1997) Liquid moulding technologies. Woodhead Publishing. https://doi.org/10.1016/B978-1-85573-242-1.50002-7. http://www.sciencedirect.com/science/article/pii/B9781855732421500027
9. Bickerton S, Advani SG (1999) Characterization and modeling of race-tracking in liquidcomposite molding processes. Compos Sci Technol 59(15):2215–2229. https://doi.org/10.1016/S0266-3538(99)00077-9
10. Devillard M, Hsiao KT, Gokce A et al (2003) On-line characterization of bulk permeability and race-tracking during the filling stage in resin transfer molding process. J Compos Mater 37(17):1525–1541
11. Devillard M, Hsiao KT, Advani SG (2005) Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding—part II: automation and validation. Compos A Appl Sci Manuf 36(11):1581–1589
12. Lawrence JM, Hsiao KT, Don RC et al (2002) An approach to couple mold design and on-line control to manufacture complex composite parts by resin transfer molding. Compos A Appl Sci Manuf 33(7):981–990
13. Lawrence JM, Fried P, Advani SG (2005) Automated manufacturing environment to address bulk permeability variations and race tracking in resin transfer molding by redirecting flow with auxiliary gates. Compos A Appl Sci Manuf 36(8):1128–1141
14. Siddig NA, Binetruy C, Syerko E et al (2018) A new methodology for race-tracking detection and criticality in resin transfer molding process using pressure sensors. J Compos Mater 52(29):4087–4103. https://doi.org/10.1177/0021998318774829
15. Di Fratta C, Klunker F, Ermanni P (2013) A methodology for flow–front estimation in lcm processes based on pressure sensors. Compos A Appl Sci Manuf 47:1–11. https://doi.org/10.1016/j.compositesa.2012.11.008
16. Fratta CD, Koutsoukis G, Klunker F et al (2016) Fast method to monitor the flow front and control injection parameters in resin transfer molding using pressure sensors. J Compos Mater 50(21):2941–2957. https://doi.org/10.1177/0021998315614994
17. Bock FE, Aydin RC, Cyron CJ et al (2019) A review of the application of machine learning and data mining approaches in continuum materials mechanics. Front Mater. https://doi.org/10.3389/fmats.2019.00110
18. Brunton SL, Noack BR, Koumoutsakos P (2020). Machine learning for fluid mechanics. https://doi.org/10.1146/annurev-fluid-010719-060214

19. Wuest T, Weimer D, Irgens C et al (2016) Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res. https://doi.org/10.1080/21693277.2016.1192517

20. Meng L, McWilliams B, Jarosinski W et al (2020). Machine learning in additive manufacturing: a review. https://doi.org/10.1007/s11837-020-04155-y

21. Wang J, Ma Y, Zhang L et al (2018) Deep learning for smart manufacturing: methods and applications. J Manuf Syst 48:144–156 https://doi.org/10.1016/j.jmsy.2018.01.003 https://www.sciencedirect.com/science/article/pii/S0278612518300037. Special issue on smart manufacturing

22. Tao F, Qi Q, Liu A et al (2018) Data-driven smart manufacturing. J Manuf Syst 48:157–169. https://doi.org/10.1016/j.jmsy.2018.01.00

23. Wang J, Fu P, Gao RX (2019) Machine vision intelligence for product defect inspection based on deep learning and Hough transform. J Manuf Syst 51:52–60. https://doi.org/10.1016/j.jmsy.2019.03.002

24. Iglesias M, Park M, Tretyakov MV (2018) Bayesian inversion in resin transfer molding. Inverse Probl 34(10):1–49. https://doi.org/10.1088/1361-6420/aad1ce. https://arxiv.org/abs/1707.03575v2

25. González C, Fernández-León J (2020) A machine learning model to detect flow disturbances during manufacturing of composites by liquid moulding. J Compos Sci 4(2):71. https://doi.org/10.3390/jcs4020071

26. Matveev M, Endruweit A, Long A et al (2021) Bayesian inversion algorithm for estimating local variations in permeability and porosity of reinforcements using experimental data. Compos A Appl Sci Manuf 143(106):323. https://doi.org/10.1016/j.compositesa.2021.106323

27. Mendikute J, Plazaola J, Baskaran M et al (2021) Impregnation quality diagnosis in resin transfer moulding by machine learning. Compos B Eng 221(108):973. https://doi.org/10.1016/j.compositesb.2021.108973

28. Foreman-Mackey D, Hogg DW, Lang D et al (2013) emcee: The MCMC hammer. Publ Astron Soc Pac 125(925):306–312. https://doi.org/10.1086/670067

29. Weller HG, Tabor G, Jasak H et al (1998) A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 12(6):620–631. https://doi.org/10.1063/1.168744

30. Deshpande SS, Anumolu L, Trujillo MF (2012) Evaluating the performance of the two-phase flow solver interFoam. Comput Sci Discov 5(1):014–016. https://doi.org/10.1088/1749-4699/5/1/014016

31. PyDOE (2019) The experimental design package for Python. https://pythonhosted.org/pyDOE/

32. Paszke A, Gross S, Massa F, et al (2019) Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, et al (eds) Advances in neural information processing systems 32. Curran Associates, Inc., pp 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

33. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90

34. Huang G, Liu Z, Van Der Maaten L, et al (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 2261–2269. https://doi.org/10.1109/CVPR.2017.243

35. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th international conference on machine learning, pp 6105–6114

36. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167

37. Furusho Y, Ikeda K (2020) Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives. APSIPA Trans Signal Inf Process 9:e9. https://doi.org/10.1017/ATSIP.2020.7