Article

Machine Learning for Aiding Blood Flow Velocity Estimation Based on Angiography

Swati Padhee[1], Mark Johnson [2], Hang Yi [2], Tanvi Banerjee [1] and Zifeng Yang [2,*]

[1] Department of Computer Science and Engineering, Wright State University, Dayton, OH 45435, USA; padhee.2@wright.edu (S.P.); tanvi.banerjee@wright.edu (T. B.)

[2] Department of Mechanical and Materials Engineering, Wright State University, Dayton, OH 45435, USA; mark.johnson@wright.edu (M. J.); hang.yi@wright.edu (H. Y.); zifeng.yang@wright.edu (Z. Y.);

* Correspondence: zifeng.yang@wright.edu; Tel.: +1-937-775-5091

Abstract: Detailed flow distributions in vascular systems are the key to identifying hemodynamic risk factors for the development and progression of vascular diseases. Although computational fluid dynamics (CFD) has been widely used in bioengineering research on hemodynamics predictions, not only are high-fidelity CFD simulations time-consuming and computing-expensive, but also not friendly to clinical applications due to the difficulty of comprehensive numerical calculations. Machine learning (ML) algorithums to estimate the flow field in vascular systems based on the angiographic images of the blood flow using existed diagnostic tools are emerging as a new pathway to facilitate the mapping of hemodynamics. In present work, the dye injection in a water flow was simulated as an analogy of the contrast perfusion in blood flow using CFD. In the simulation, the light passes through the flow field and generates projective images, as an analogy of X-ray imaging. The simulations provide both the ground truth velocity field and the projective images of the flow with dye patterns. A rough velocity field was estimated using the optical flow method (OFM) based on projective images. ML algorithums are then trained using the ground truth CFD data and the OFM velocity estimation as the input. Finally, the interpretable (logistic regression) and deep (neural networks, convolutional neural networks, long short term memory) machine learning models are validated by using parallel *in vitro* experiments on the same flow setup. The validation results showed that the employed ML model significantly reduced the error rate from 53.5% to 2.5% in average for the v-velocity estimation.

Keywords: machine learning (ML); convolutional neural networks (CNN); computational fluid dynamics (CFD); optical flow method (OFM); dye perfusion

1. Introduction

In developed countries, the leading causes of mortality and morbidity are diseases in the vascular system [1]. In recent years, blood flow/shear stress has gained a lot of interest as a complementary explanation for plaque formation [2,3], aneurysm initiation, growth, and rupture [4]. The role of blood flow in the development of atherosclerosis is based on the observation that vascular inflammation and plaques are distributed near side branches or arterials stenosis, where blood flow is non-uniform, and at the lesser curvature of bends where the blood flow rate is relatively low [5,6]. The effect of blood flow on the vessel wall is through shearing and stressing forces, which influences the behavior of endothelial cells, including morphological adaptations and physiological changes.

The methods for estimation of wall shear stress (WSS) can be performed by acquiring three-dimensional (3D) reconstruction of vessel volume using either invasive modalities such as intravascular ultrasound and invasive coronary angiography, or less-invasive techniques, including computed tomography angiography and cardiac magnetic resonance angiography with the application of numerical methods to calculate flow within the reconstructed arterial model for solving fluid dynamics [7,8]. The numerical methods are known as computational fluid dynamics (CFD) techniques. 3D reconstruction of the patient-specific vascular models with subsequent CFD simulations is being increasingly used to predict the nature of blood flow in the human vascular system [9-11].

The merit of CFD lies in its efficiency and relatively low cost in the fully detailed flow simulation and in developing or optimizing fluid mechanical devices or system designs [12]. The governing equations of fluid dynamics, called the Navier-Stokes equations, can be computed to obtain the detailed flow field and pressure distribution. In order to simulate natural vascular blood flow, a domain of interest must be defined, and boundary conditions need to be specified. The isolation and generation of boundary conditions are among the most significant challenges in the integration of CFD for assessing the physiologic significance of artery disease [13]. However, (i) the reconstruction of the vascular model using the medical images, (ii) the accuracy of the reconstructed geometry, and further (iii) the computation resources required for computation still create barriers to the wide application of this methodology for clinical use.

On the other hand, *in vivo* quantifying hemodynamics during the diagnostic imaging process has drawn interest in recent years. To determine hemodynamic characteristics, a high spatial resolution of the blood flow velocity distribution is needed, especially for quantifying the WSS. Medical professionals and researchers have used digital subtraction angiography (DSA) for estimating the blood flow velocity distribution using intensity-based algorithms, such as the optical flow method (OFM) [14-18]. From the DSA images, the movement of the contrast pattern is calculated, representing the movement of the local blood flow. However, the accuracy suffers due to the method in which these images are acquired [19]. DSA images involve x-rays that transmit through the patient and produce a contrast projection onto the scintillator plane. The intensity variation is based on how much X-ray energy passes through the nonuniform medium, which results from varying attenuation through muscle, bone, soft tissue, blood, and contrast agent [20]. DSA images are generally acquired by transmitting light through a 3D domain and projected onto a 2D image. Without isolating a 2D plane within the flow, the images are essentially integrated representations of the entire flow field. Therefore, an error is introduced into the calculations, especially when the measurements are inferred as measurements at the center plane of the domain [8].

Kawaji et al.[21], calculated blood velocity by determining displacement over consecutive images with a known elapsed time between frames. The uncertainty in the velocity profiles begin to increase when the end point becomes unclear, but for simple flows the relative error can be as low as 2.5% [21]. In our earlier work, Yang and Johnson [19] have conducted research with dye visualization specific to pulsatile flow through a tube, where OFM underestimated the center plane velocity by 16%-24% in the central region and by about 29%-43% in the outer region compared to PIV. To increase the accuracy of calculating the velocity distribution, this research proposes the use of machine learning to correct velocity measurements from OFM.

Instead of using DSA images directly, we used Cradle CFD Software (Cradle Co., Ltd., Japan), to simulate dye propagating through water, where the results were then used to generate a projection-based image similar to DSA. The pixel intensity was determined through a discretized form of the light attenuation equation or Beer-Lambert Law [22]. Since DSA images can subtract visible structures prior to contrast injection, such as bone and tissue, the simulation of dye in water is analogous to DSA images. Here the dye injection simulates the perfusion of iodine-contrast agent, and water simulates the blood flow. We recognize that water is a Newtonian fluid whereas blood is non-Newtonian; however, this research is interested in the feasibility of machine learning (ML) algorithms to correct OFM results from projection-based images such as DSA. Choosing CFD to simulate the dye perfusion in water over experimental dye projection images is to easily obtain an accurate 3D velocity field for training the ML algorithms and allows error analysis of the OFM and predicted values. Additionally the simulation images avoid errors associated with experimental imaging process. Therefore, the error associated with the OFM estimation would be a result of the 3D compression effect onto a 2D image. A three-step ML framework for flow field estimation is established as shown in Figure 1: 1) Simulate dye perfusion in a water flow, generate projective images, as well as provide the flow velocity estimation from OFM and the groundtruch velocity from CFD; 2) train ML models using the CFD and OFM data; and 3) validate the ML models using parallel *in vitro* experiments.

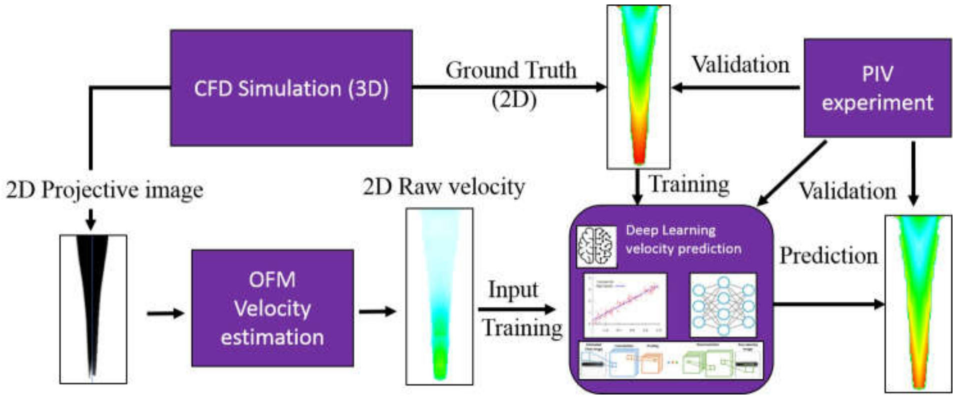


Figure 1. Flow chart of the machine learning framework for velocity estimation on the dye perfusion.

2. Materials and Methods

2.1. CFD Simulated Data

Cradle SC/Tetra 2021 (Software Cradle Co., Ltd., Japan) was used to simulate the dye injection into a laminar water flow to obtain the accurate 3D velocity field and obviate experimental uncertainties related to pixel intensity. The accurate velocity field from CFD serves as the ground truth for the supervised training of the ML model. The simulation was conducted within a 1 × 1 in square cross-sectional tube with a length of 1-inch with the dye inlet located in the center. To minimize computational costs, only a fourth of the domain was simulated therefore the simulation dimensions were 0.5 × 0.5 in by 1 in. For our experiments, water was used in the simulation, which reduces the continuity and momentum equations with the assumption of incompressible and Newtonian fluid. To capture the dye diffusion throughout the domain, the species equation was enabled. The governing equations can be written in tensor form as:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

$$\rho\frac{\partial u_i}{\partial t} + \rho\frac{\partial(u_j u_i)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial}{\partial x_j}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] + \rho g_i \quad (2)$$

$$\rho\frac{\partial \phi}{\partial t} + \rho\frac{\partial u_i \phi}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\rho D\frac{\partial \phi}{\partial x_i}\right] \quad (3)$$

where u_i represents the water/dye velocity, ρ is the density of water and dye which were identical to the measurements in the experiments, p is pressure, μ is viscosity, g_i is gravity, ϕ is the dye concentration which is identified a scalar value 1.0 in the dye injection inlet, D is the dye diffusion coefficient. The coefficient of diffusion was set at 10^{-9} m^2/s as the typical value for dye is between 10^{-10} and 10^{-9} m^2/s [23]. A set of unstructured tetrahedral meshes was generated using the assembled meshing package in Cradle SC/Tetra 2021. A mesh independent study for the quarter model was conducted, and the final mesh with elements of 16,850,005 was adopted for the current study. The mesh size near the wall and in the dye injection region was refined with y+ < 1, where y+ is the dimensionless wall distance.

Four different transient simulations were created where the dye concentration and pattern were varied for each simulation. The first simulation injected dye at a constant rate of 0.005 m/s with a dye radius of 0.37 mm initially and gradually increased the radius throughout time until the radius reached 2.0 mm, which is the same situation as the parallel experiment. Second simulation had intermittent dye injections where dye was injected for 0.1 secs then stopped for 0.25 secs, and this process would repeat continuously with a period of 0.35 secs. The diameter of the dye was held constant at 2.0 mm for the second simulation. For the third simulation, the dye was injected intermittently in the same pattern as the second simulation, except the dye concentration was no longer uniform at the inlet. The distribution of the dye concentration was represented with a cosine function where the dye concentration at the center was 1.0, and near a radius of 2.0 mm the concentration was 0.54. The fourth simulation involved a uniform dye concentration in the radial direction but varied in a sinusoidal pattern with time. The dye distribution equations are expressed for the third and fourth simulation, respectively, as

$$\phi(x) = \phi_o \cos(\omega x) \tag{4}$$

$$\phi(t) = \cos\left(\frac{4\pi}{3}t\right) \tag{5}$$

In equation (4), ϕ represents the dye concentration, ϕ_o represents a constant concentration value of ϕ_o= 1.00, ω is the angular frequency of ω = 500 1/m. Equation (5) provides the transient cosine wave for the dye concentration, where $\frac{4\pi}{3}$ provides a time period of 1.5 seconds and t is the time in seconds. These four variations were carried out to represent the impact of different perfusion patterns of the dye on the estimated velocity accuracy. Results from the simulation were exported either every 0.2 seconds or 0.3 seconds (in reference to the simulation time), and additional results were exported 0.02 seconds after the 0.2 or 0.3 second interval. This short time step was necessary for OFM to calculate the velocity between two successive images, as the nature of the OFM algorithm requires the movement of the pattern to be less than 1 pixel to maintain a decent accuracy. Overall, the four cases provided 106 exported result files which would lead to 106 image files or 53 data files after being analyzed with OFM.

Light passing through the domain was simulated to generate projective images. This was accomplished with the Beer-Lambert law [22] which describes the amount of light attenuated through a medium. By solving for the light intensity, I, out of the medium, the overall projection image can be created by dividing the domain into multiple cells and performing successive calculations of the intensity. Note the variable z represents the direction parallel to the light beam and perpendicular to the flow diretion, μ represents the attenuation coefficient, and I_o is the initial intensity prior to passing through the element,

$$\frac{dI}{dz} = -\mu I \tag{6}$$

$$I = I_o \exp(-\mu \bigtriangleup z) \tag{7}$$

For example, if a single row of elements aligned in the z-direction encounters an initial light source, the first element will decrease the intensity based on the element's attenuation coefficient (which is directly related to the dye concentration). The output intensity through the first element will then be used as the initial intensity for the second element. This process is continued until the light ray passes through all of the elements. Therefore, the intensity can be written in a discretized form as shown below.

$$I_j = I_{j-1} \exp\left(-\mu_j \bigtriangleup z_j\right) \tag{8}$$

Example of a projective image is shown below in Figure 2, where the darker regions represent locations where dye has attenuated light. The image resolution is 161x306 pixels.

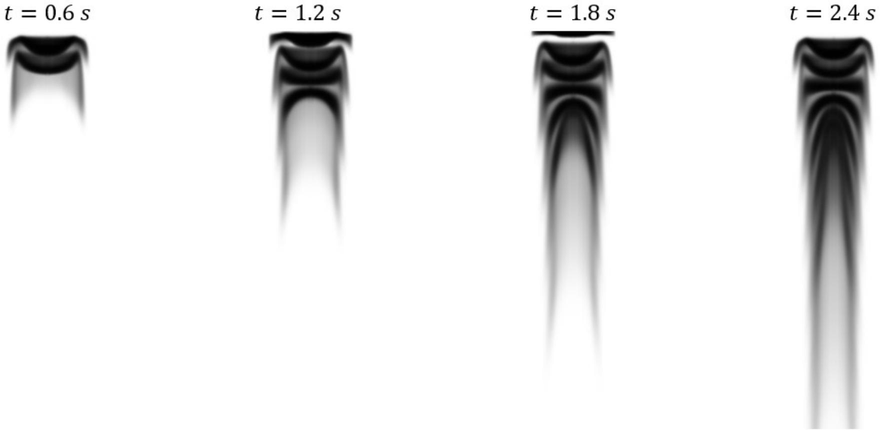


Figure 2. Projective images from case 4 which was generated from CFD results by using the Beer-Lambert law to simulate light passing through the domain.

Once projective images were generated, OFM was applied to each image pair to calculate velocity vectors at each pixel. The governing equation for OFM is an expression of a brightness constraint, which assumes the intensities are constant from one image to the next.

$$Q_t + V \cdot \nabla Q + QV \cdot V = D\nabla^2 Q \tag{9}$$

Here the intensity of pixel is represented by Q (a matrix for the entire image), the velocity vector is given by V, and D is the diffusion coefficient. Additional details can be found in our previous paper [19].

Along with the u- and v-velocity components, the intensity gradients in the x- and y-directions were also calculated. Since OFM calculates movement based on pixel intensities, the gradients are of importance as it provides insight into which vectors were calculated based on low or high intensity gradients. The velocity calculated from OFM is regarded as the test data while the CFD velocities are regarded as the ground truth data for the ML model. The CFD velocities were extracted from the center plane of the simulation as the goal is to improve the OFM results to better represent the velocities in the middle plane of the domain. Once these files were created they were then used as the training data for the ML algorithm. 423,093 simulated data points representing the flow velocity were generated for the current modeling.

2.2. Experimental Data for Validation

The data discussed above is generated based on CFD results, however the future application for this work is to apply this method directly to *in vivo* images such as DSA. Direct application to DSA images is not feasible currently because no method can accurately describe the flow field for validations. Therefore, an *in vitro* experiment analogous to DSA was created to obtain real images for testing the current method against accurate measurements from particle image velocimetry (PIV). Additionally, this provides insight into whether the ML algorithm is able to correct velocity values that incur error from the *in vitro* experiment such as non-uniform light intensity throughout the image domain. The experiment consists of a square vertical tube test section (with the same dimension as the counterpart used in the CFD simulation) with a water reservoir and contraction region directly above, as presented in Figure 3. This contraction region leads water from the reservoir to the test section through a gradual cross-sectional change to mitigate the inducing of turbulence. Downstream of the test section is a flow control valve that controls the speed of the flow when coupled with a constant water height in the water reservoir. A pump feeds water into the reservoir where a partition wall, which is slightly shorter than the reservoir walls, allows water to spill over into the adjacent reservoir to maintain a constant water level. Dye is injected in the center of the square tube through the guidance of a centering device to maintain consistent positioning during the injection process. Finally, a convex lens is positioned one focal length away from the LED bulb, where the lens' position is between the bulb and the test section, so light rays can transmit parallel through the test section thus mitigating non-uniform light intensity on the image.

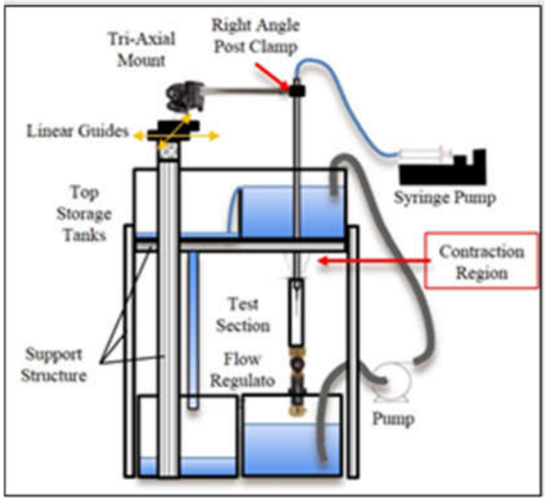
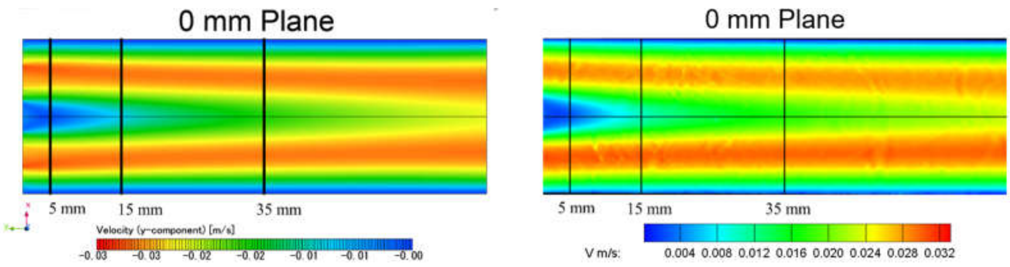


Figure 3. Schematic of the *in vitro* experiment with the contraction section, dye centering apparatus, and partitioning wall to maintain constant water level.

In the *in vitro* experiment, the water is analogous to blood in DSA images and the dye provides contrast similar to the contrast agent injected in blood vessels. Also, the LED light is comparable to the x-ray source since both transmit through the domain and project contrast agent perfusion onto a 2D plane. A LaVision PIV system consisted of a Nd: Yag laser with a wavelength of 532 nm with a maximum pulse energy of 300 mJ, and a programmanble timing unit (PTU), and a 14 bit CCD camera with a resolution of 1392 × 1040 pixel (66 × 49 mm) was employed to conduct the PIV measurements. The laser was shaped into a laser sheet of 1 mm in the measurement region to illuminat the flow field. The PTU synchronized the laser and the camera to obtain the particle images with a time delay of 33 ms between images. The flow was seeded with fluorescent red polyethylene microspheres with a diameter rangine from 10 to 45 μm. The fluorescent emission light from the partile has a wavelength of 610 nm. A long pass filter with a cut-off wavelength of 600 nm was mounted on the camera length to filter out the laser light and only leave the emission light passing through. This fluorescent imaing can effectively remove the light reflection noise from the wall. DaVis 8.0 was used to process the images to obtain 1000 instantaneous velocity distributions with a frame rate of 30 Hz. A multi-pass cross correlation analysis was conducted in DaVis 8.0 with a 32 × 32 pixel interrogation region with 50% overlap, then a second pass with an 8 × 8 pixel window size with 50% overlap. The final velocity vector resolution is 5.2 vectors/mm. The uncertainty in the velocity measurements is estimated to be about 1% of the magnitude. A total of 5,071 flow velocity points for OFM velocity and PIV velocity were generated for evaluation of current ML algorithms.

The CFD simulation of the velocity distribution is compared with the PIV measurements in the middle cut plane, as presented in Figure. 4. The overall difference in the velocity distribution between the CFD simulation and the PIV measurement is less than 5% of the velocity magnitude, which essentially validated the accuracy of the current simulation results.



(a) (b)

Figure 4. Velocity distribution comparision in the middle plane of the tube: (a) CFD simulation; (b) PIV experiment.

2.3. Machine Learning Models

The following sections detail predicting the accurate velocity field with supervised neural network training. Three deep machine learning models with varying numbers of parameters and depths were tested. *Regression* is a predictive modeling task that involves predicting a numerical output given some input. It is different from classification tasks that involve predicting a class label. Typically, a regression task involves predicting a single numeric value. Although, some tasks require predicting more than one numeric value. These tasks are referred to as multiple-output regression or multi-output regression for short. In multi-output regression, two or more outputs are required for each input sample, and the outputs are required simultaneously. The assumption is that the outputs are a function of the inputs, and u- and v-components of the velocity are correlated. Since OFM calculates a velocity vector for every point in the image, even when the dye is not present (pseudo vectors were calculated in regions without dye), specific regions with the presence of dye were selected for the training and test data. The images generated were 8-bit grayscale images where pixel intensity values are assigned between 0 and 255. The presence of dye in a pixel location decreases the intensity value for that location. Based on observation, pixel values of 80 or less (after considering the image noise) represent regions where dye is dominant and thus represent data points used in the training data or test data sets. This criterion was applied to all of the OFM output results, and each valid data point compiles into a single file. Multiple data points correspond with the same XY-coordinates as a result of multiple time instances containing dye at that point. Therefore all of the duplicates except one are used for training, and the other duplicate was used in the test set. For instance, if the coordinate point (x, y) = (0.001, 0.0253) is near the dye inlet, then there were multiple OFM data sets that contain dye at that coordinate point. Assuming 10 duplicates, one data point was selected randomly for the test set while the remaining 9 points were compiled into the training data. This random selection provided unbiased data for the test set to determine if the ML algorithm was able to predict values accurately.

2.3.1. LASSO Regression Model

A multivariate (multi-input, multi-output) regression model with the L1 penalty was designed, commonly known as LASSO [24]. We selected this model since it achieves sparsity in the estimated model by setting the regression coefficients for features to zero for those features that don't affect the output or target values. We treat this ML model as our baseline model.

2.3.2. Multi Layer Perceptron Model

A multilayer perceptron (MLP) model is defined for the multi-output regression task defined in the previous section. Each sample has 4 inputs and 2 outputs, therefore, the network requires an input layer that expects 4 inputs in the first hidden layer and 2 nodes in the output layer. The ReLU activation function was used in the hidden layer which has

32 nodes, chosen after some trial and error. The model was fitted using two loss functions and the Adam version of stochastic gradient descent.

2.3.3. Convolutional Neural Networks (CNN) for Multi-Output Regression

CNNs have proven great capability of learning important features from images at the pixel level in order to make useful predictions for both classification and regression problems [25]. Another advantage of this approach, compared to conventional fully-connected layer networks, lies in the fact that convolutions provide weight-sharing and sparse connectivity. These properties enable more efficient memory usage to learn the necessary information needed to create a surrogate model. The surrogate model then reconstructs an approximation of the entire velocity field from a given set of boundary conditions. Standard CNN has several convolutional layers followed by fully-connected layers (layers where each hidden node is connected to every hidden node in the preceding and subsequent layer), ending with a classification softmax layer (that generates the final classification output). Adapting a classification CNN architecture to regression consists in removing the softmax layer and replacing it with a fully connected regression layer with linear or sigmoid activation. Linear activation means that the transfer function is a straight line. Thus the activation is proportional to the input and not confined to a specific range. Three convolutional layers were added with ReLU activation and a max pool layer after the first convolutional layer.

2.3.4. Long Short-term Memory Model

Long short-term memory (LSTM) is a type of Recurrent Neural Network (RNN), i.e. a multi-layer NN. The LSTM architecture was originally introduced by Hochreiter and Schmidhuber [26] with the purpose of overcoming the vanishing or exploding gradients problem. In a network of n hidden layers, n derivatives are multiplied together. If the derivatives are large then the gradient will increase exponentially as we propagate down the model until they eventually explode, known as the problem of exploding gradient. Alternatively, if the derivatives are small then the gradient will decrease exponentially as we propagate through the model until it eventually vanishes, known as the vanishing gradient problem. LSTM allows flowgates, i.e., the input gate, the forget gate, the control gate, and the output gate, as shown in Figure. 4. In Figure 4, the input gate, the forget gate, the control gate, and the output gate are denoted by i_t, f_t, c_t, and o_t , respectively. The details of these four gates are enlightened below.

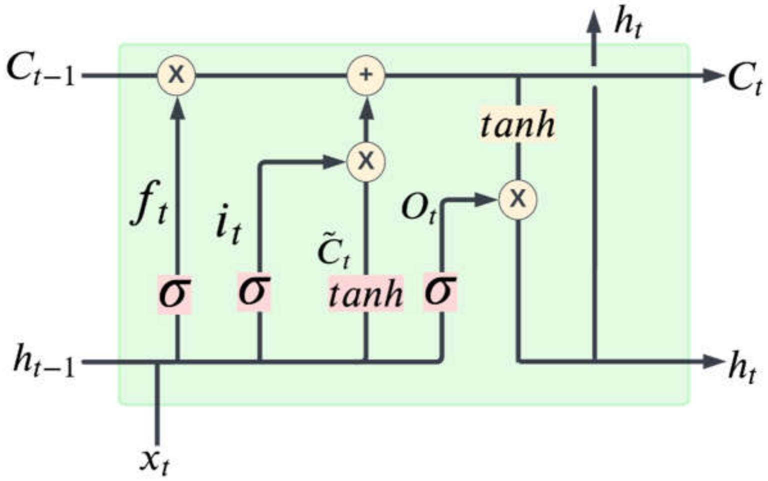


Figure 5. Structure of our Long-short-term memory (LSTM) cell at time *t*.

The input gate, forget gate and control gate are expressed as:

$$i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i) \tag{10}$$

$$f_t = \sigma\left(w_f * [h_{t-1}, x_t] + b_f\right) \tag{11}$$

$$\widetilde{c_t} = \tanh(w_c * [h_{t-1}, x_t] + b_c) \tag{12}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{13}$$

The input gate decides which information can be transferred from the earlier cell to the current cell as shown in equation (10). The forget gate is defined by equation (11), and it is used to store the information from the input of previous memory or otherwise. The control gate controls the update of the cell and it is defined in equation (13). To update the hidden layer (h_{t-1}) and update the output, finally the output gate is used which is given by the following equation (14):

$$\begin{gathered} O_t = \sigma(w_c * [h_{t-1}, x_t] + b_o) \\ h_t = O_t * \tanh(c_t) \end{gathered} \tag{14}$$

In the above equations, x_t is the input, w represents the corresponding weight matrix of input, b_o is the corresponding bias of input, C_{t-1} is the previous block memory, C_t represents the current block memory, $\widetilde{c_t}$ as shown in equation (12) is the vector of new candidate values updated with the tanh layer, h_{t-1} represents the previous block output, h_t represents the current block output. Furthermore, tanh is the hyperbolic tangent function that is used to scale the values in the range of -1 to 1, and σ is the sigmoid activation function, which gives the output between 0 and 1. We designed a sequential model containing LSTM layers with ReLU activations, a dense output layer, and an Adam optimizer with two regression loss functions. We set the input dimension in the first layer and the output dimension in the last layer of the model followed by a regression layer.

2.4. Loss Functions

Conventional regression loss functions are metrics-inspired losses, namely the Mean Absolute Error (MAE), and Mean Squared Error (MSE), defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - g_i| \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - g_i)^2 \tag{15}$$

where predicted (resp. ground truth) values are denoted p_i (resp. g_i). We report both the loss functions, as advocated in [27, 28]. The metrics to evaluate the results are Mean Absolute Error (MAE). We repeated 10-fold cross-validation with 10 folds and three repeats training the model with Adam optimizer. Models are implemented with Keras and PyTorch on NVIDIA Tesla T4 GPU.

3. Results and Discussion

In this section, the models discussed in providing efficient approximations of flow velocity were evaluated. In experiments, four regression models and two loss functions, the Mean Absolute Error (MAE), and the Mean Squared Error (MSE) were compared. Table 1 shows an average mean absolute error value across 10 runs of experiments. Hyper-parameter optimization using gridsearch with a ten-cross-validation technique is applied to optimize the number of hidden layers, hidden neurons, and the batch size. The number of hidden layers is optimized ranging from 1 to 4 and the number of hidden neurons is optimized ranging from 4 to 20 with an increment of 2. The batch sizes used in parameter tuning are 32, 64, and 128. After applying hyper-parameter optimization, the best hyper-parameters are obtained, there is one hidden layer with 10 hidden neurons, and the batch size is 32. A dropout rate of 0.02 is applied to avoid overfitting after the first hidden layer. Each of the MLP, CNN, and LSTM architectures were run for 100, 500, 1000, and 2000 epochs. The best results were obtained with 100 epochs for MLP with an execution time of 4 mins per 10 epochs and 500 epochs for both CNN and LSTM, as with a higher number of epochs, the average performance got worse.

Table 1. Performance of regression models in terms of mean absolute error (mae) for two different loss functions: MSE, MAE.

	LASSO	MLP	CNN	LSTM
Validation with simulated test data				
MAE	2e-4†	2e-4*	4e-4*	4e-4*
MSE	3e-8†	5e-8*	7e-8†	8e-8*
Validation with *in vitro* experimental data				
MAE	3e-3†	5e-4*	6e-4†	6e-4*
MSE	5e-7†	4e-8*	5e-8*	8e-8†

* indicates p-value < 0.01, † indicates p-value < 0.05

Multiple regression algorithms were adopted to provide models capable of predicting the velocity field. The deep learning models (MLP, CNN, and LSTM) outperformed the baseline LASSO model. The hypothesis that the relationship between the 2D projection images and the initial and final velocity components can be represented as a multivariate regression problem with improved accuracy has been demonstrated in Table 1. Significant differences across the three deep learning models were not observed. All of the reported models were statistically significant with a p-value of <0.01 or 0.05. However, the MLP model was computationally less expensive and hence, a quantitative analysis of the results in comparison with the baseline OFM and with the standard CFD approach is presented in Figure 6 and Figure 7 for comparisons of both u- and v-components velocity contours. The velocity distribution (both u-component and v-component) can be used to analyze flow characteristics such as the wall shear stress, which is a key parameter in the study of the pathophysiology of vascular disease.

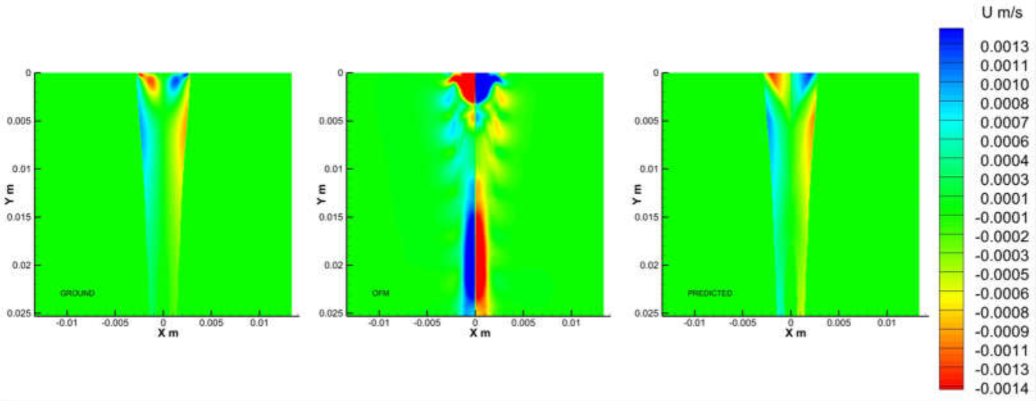


Figure 6. U-contours of the CFD ground truth (left), OFM (middle), and predicted (right) results.

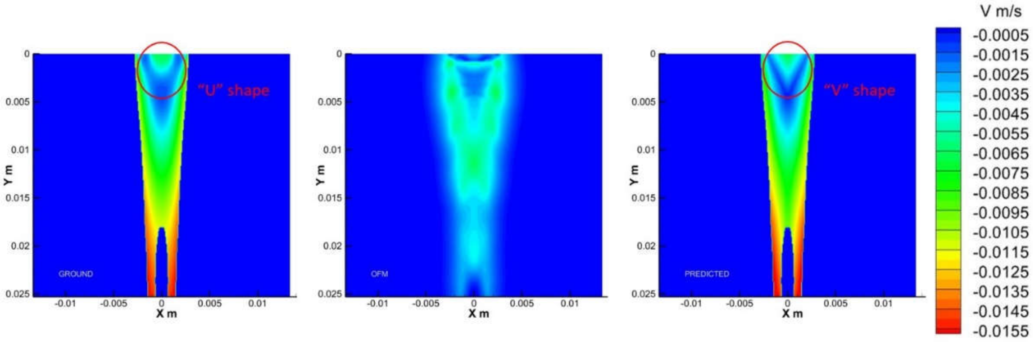


Figure 7. V-contours of the CFD ground truth (left), OFM (middle), and predicted (right) results.

As shown in Figure 6, the OFM results overestimated the u-component velocity in the central region of the domain, especially downstream, but OFM underestimated the flow between y = 0.005m and y = 0.01m. Despite these discrepancies in the u-contours between OFM and CFD, the predicted values were able to match the CFD values between y = 0.005m and 0.01m as well as further downstream. There is a discernible difference between the predicted and CFD at the dye inlet, where the predicted values overestimate near the centerline, which can be attributed by the overestimation from the OFM data. Discrepancies of the v-component velocity contours between OFM and CFD are illustrated in Figure 7, where OFM consistently underestimates the velocity especially downstream from y = 0.017m. However, the predicted values show similar contour patterns as the CFD contour for most of the regions except near the inlet. The predicted inlet has more of a "V-shape" whereas the CFD contour has more of a "U-shape" within the lower speed region due to the mixing effect between the dye and water.

Furthermore, the velocity profiles at various locations of the flow field were extracted for comparison as shown in Figure 8. Near the inlet, at y = 1.66 mm, the v-component velocity profile for the MLP predictions has trouble to match the CFD results near x = ±0.7 mm to x = ±1.5 mm, where the MLP predictions have "V-shape" in the contour as discussed earlier. Further downstream at y = 6.64 mm, the predicted velocity profile resembles the ground-truth velocity profile well with little discrepancies. These quantitative comparisons of the velocity profiles can be further examined with error analysis in Figure 9 showing a significant reduction in the percentage of error with respect to the ground truth (CFD) for the v-component velocity. Error associated with the predicted values reach as high as 13%, near the inlet, and as low as 2% around y = 0.008 m. It is interesting to observe that the error in OFM is lower at the inlet and increases further downstream, whereas the opposite is true for MLP predictions. The MLP predictions presented larger errors near the inlet as a result of larger variations in the OFM velocities caused by low intensity gradients near the inlet on the image. As the dye moving downstream, the MLP predictions can match the ground truth well. The averaged V-error of about 53.5% in the velocity estimation with OFM has been significantly reduced to 2.5% in average by the current MLP predicitons. The current pilot study is focusing on prediction of the velocity distributions. Future studies related to the wall shear stress predicitons would focus on the local velocity gradient accuracies, especially near the wall. The current ML model development focused on the 2D velocity map prediction, which limits its use for the complicated 3D flow environment, such as the blood flow in an aneurysm. Another limit is the current model can only predict the velocity, but another important flow information, pressure in the flow field, cannot be predicted. We plan to encode the Navier-Stokes equation into the modeling as recently proposed by Raissi et al [29], as well as extend the current model to predict the 3D flow feature in the future study.

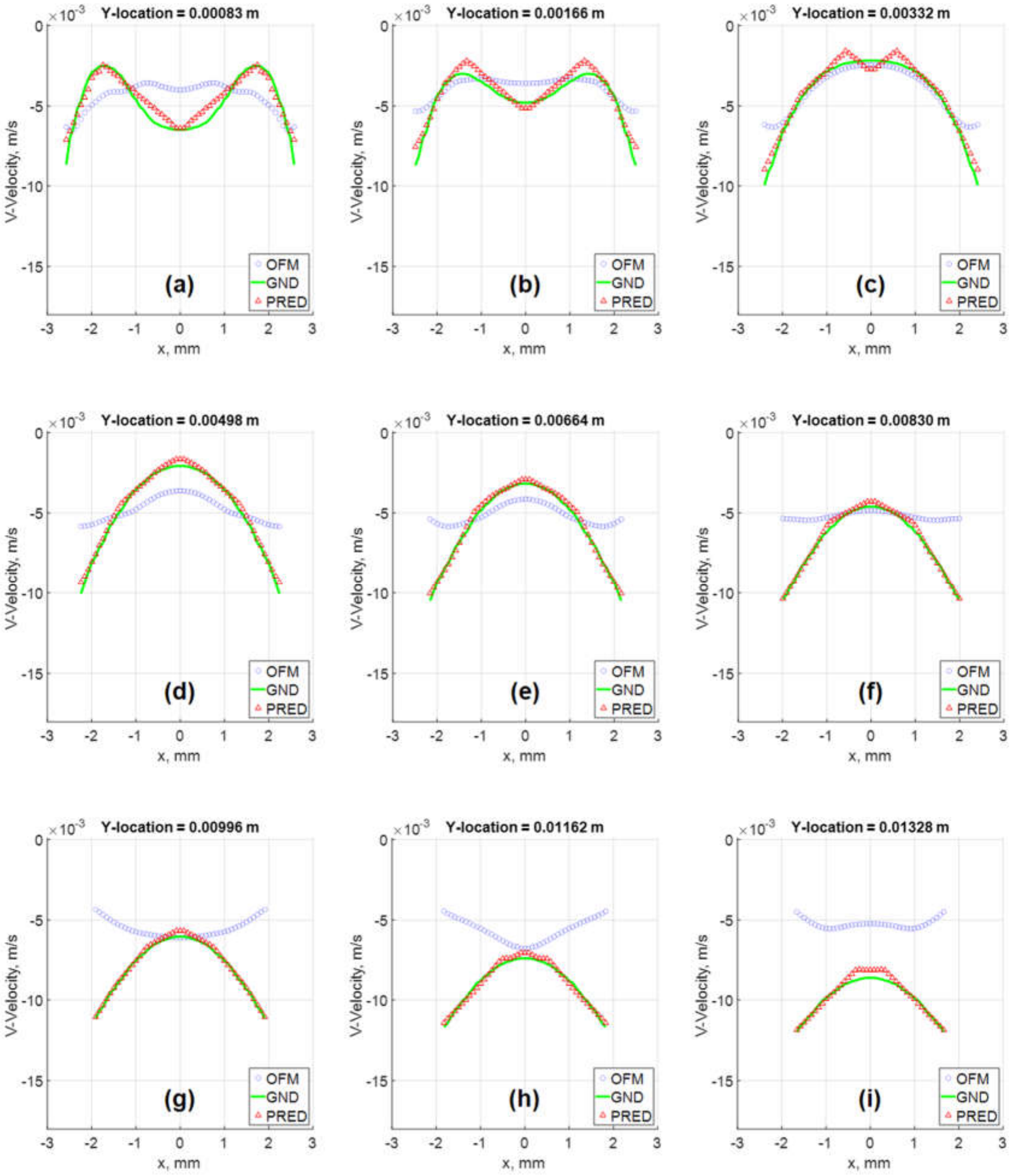


Figure 8. Velocity Vector fields comparison of the in-silico results at various Y locations: (a) Y = 0.00083 m; (b) Y = 0.00166 m; (c) Y = 0.00332 m; (d) Y = 0.00498 m; (e) Y = 0.00664 m; (f) Y = 0.00830 m; (g) Y = 0.00996 m; (h) Y = 0.01162 m; (i) Y = 0.01328 m.

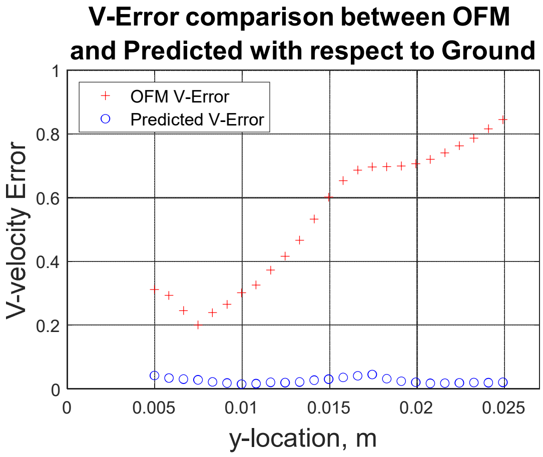


Figure 9. Percentage error of OFM estimation and MLP predicted results for v-component velocity with respect to the ground truth data at various Y-locations.

4. Conclusions

In this study, the flow velocity estimation as a multi-output regression problem was performed. The performance of four machine learning models using simulated images as well as *in vitro* experimental images were evaluated. Results showed that the performance of all the neural network architectures (MLP, CNN, and LSTM) is comparable. However, the MLP model is significantly less computationally intensive as compared to the deep learning counterparts. The LSTM model has the longest computation time and the best performance compared to CNN and MLP. Both MLP and LSTM models can be used to automate the flow velocity estimation depending on the task and taking into account the cost for computation.

The following summarizes the main contributions of this paper:

1. We presented an analysis that ML algorithms are able to correct OFM results from projection-based images significantly reducing the error rate.
2. We extended the literature by considering the interaction of u- and v-components velocity with the intensity gradients on the image in both x- and y-directions.
3. We released the data and code used in this work for reproducibility and further research in this direction.

Supplementary Materials: Figure S1: title; Table S1: title; Video S1: title.

Author Contributions: Conceptualization, Zifeng Yang.; methodology, Mark Johnson, Swati Padhee, Tanvi Banerjee, Hang Yi, Zifeng Yang; software, Swati Padhee, Mark Johnson, Hang Yi; validation, Swati Padhee, Mark Johnson; formal analysis, Swati Padhee, Mark Johnson; investigation, Swati Padhee, Mark Johnson; resources, Zifeng Yang; data curation, Swati Padhee, Mark Johnson, Hang Yi; writing—original draft preparation, Swati Padhee, Mark Johnson; writing—review and editing, Tanvi Banerjee, Zifeng Yang, Hang Yi; visualization, Swati Padhee, Mark Johnson, Zifeng Yang; supervision, Tanvi Banerjee, Zifeng Yang; project administration, Zifeng Yang; funding acquisition, Zifeng Yang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Heart, Lung, and Blood Institute grant number [No. R44HL132664] and Premier Health and Boonshoft School of Medicine Endowment Funding at Wright State University.

Data Availability Statement: Data available on request due to restrictions e.g., privacy or ethical.

Acknowledgments: This study is sponsored by the National Heart, Lung, and Blood Institute with the award No. R44HL132664 and the Premier Health and Boonshoft School of Medicine Endowment Funding atWright State University (WSU).

Conflicts of Interest: The authors have no conflict of interest to disclose.

References

1. Mathers, C.D.; Salomon, J.A.; Ezzati, M.; Begg, S.; Vander-Hoorn, S.; Lopez, A.D. *Global Burden of Disease and Risk Factors*, 1st ed.; New York, NY: Oxford University Press, USA, 2006.

2. Shaaban, A.M.; Duerinckx, A.J. Wall shear stress and early atherosclerosis: a review. *American Journal of Roentgenology* **2000**, *174*, 1657-1665.

3. Cunningham, K.S.; Gotlieb, A.I. The role of shear stress in the pathogenesis of atherosclerosis. *Lab Invest* **2005**, *85*, 9–23.

4. Ikeda, H.; Ishii, A.; Kikuchi, T.; Ando, M.; Chihara, H.; Arai, D.; Hattori, E.; Miyamoto, S. Delayed aneurysm rupture due to residual blood flow at the inflow zone of the intracranial paraclinoid internal carotid aneurysm treated with the Pipeline embolization device: Histopathological investigation. *Interv Neuroradiol* **2015**, *21*, 674-83.

5. Davies, P.F.; Polacek, D.C.; Shi, C.; Helmke, B.P. The convergence of haemody-namics, genomics, and endothelial structure in studies of the focal origin of atherosclerosis. *Biorheology* **2002,** *39*, 299–306.

6. Chaichana, T.; Sun, Z.; Jewkes, J. Computation of hemodynamics in the left coro-nary artery with variable angulations. *J Biomech* **2011,** *44,* 69–78.

7. Lee, B.K.; Kwon, H.M.; Hong, B.K.; Park, B.E.; Suh, S.H.; Cho, M.T.; et al. Hemodynamiceffects on atherosclerosis-prone coronary artery: wall shear stress/rate distri-bution and impedance phase angle in coronary and aortic circulation. *YonseiMed J* **2001**, *42*, 75–83.

8. Wong, K.K.; Sun, Z.; Tu, J.; Worthley, S.G.; Mazumdar, J.; Abbott, D. Medical imagediagnostics based on computer-aided flow analysis using magnetic resonanceimages. *Comput Med Imaging Gr* **2012**, *36*, 27–41.

9. Coskun, A.U.; Yeghiazarians, Y.; Kinlay, S.; Clark, M.E.; IIegbusi, Q.J.; Wahle, A.; et al. Reproducibility of coronary lumen, plaque, and vessel wall reconstructionand of endothelial shear stress measurements in vivo in humans. *Cathet Cardiovasc Interv* **2003**, *60*, 67–78.

10. Stone, P.H.; Coskun, A.U.; Yeghiazarians, Y.; Kinlay, S.; Popma, J.J.; Kuntz, R.E.; et al. Prediction of sites of coronary atherosclerosis progression: in vivo profilingof endothelial shear stress, lumen, and outer vessel wall characteristics topredict vascular behavior. *Curr Opin Cardiol* **2003,** *18,* 58–70.

11. Stone, P.H.; Coskun, A.U.; Kinlay, S.; Popma, J.J.; Sonka, M.; Wahle, A.; et al. Regions oflow endothelial shear stress are the sites where coronary plaque progresses and vascular remodelling occurs in humans: an in vivo serial study. *Eur HeartJ* **2007**, *28,* 5–10.

12. Lee, B.K. Computational fluid dynamics in cardiovascular disease. *Korean CircJ* **2011,** *41,* 23–30.

13. Lee, B.K.; Kwon, H.M.; Kim, D.S.; Yoon, Y.W.; Seo, J.K.; Kim, I.J.; et al. Computed numer-ical analysis of the biomechanical effects on coronary atherogenesis usinghuman hemodynamic and dimensional variables. *Yonsei Med J* **1998,** *39,* 66–74.

14. Yang, Z. Optical Flow Method for Blood Flow Velocimetry Based on DigitalX-Ray Subtraction Angiography: A Brief Review. *Research and reviews: journal of medical and health sciences* **2017**, *6*, 8–12.

15. Yang, Z.; Yu, H.; Huang, G.P.; Ludwig, B. Divergence Compensatory Optical Flow Method for Blood Velocimetry. *Journal of Biomechanical Engineering* **2017**.

16. Cornelius, N.; Kanade, T. Adapting Optical-flow to Measure Object Motion in Reflectance and X Rays Image Sequence. *ACM SIGGRAPH Computer Graphics* **1983**, *18*, 24–25.

17. Mongrain, R.; Bertrand, M.; Mailloux, G.E.; Meunier, J.; Bourassa, M.G. Obtaining blood velocity profile from coronary arteriograms via optimally controlled optical flow. *Proceedings Computers in Cardiology* **1990**, 13-16.

18. Bonnefous, O.; Pereira, V.M.; Ouared, R.; Brina, O.; Aerts, H.; Hermans, R.; et al. Quantification of arterial flow using digital subtraction angiography. *Medical Physics* **2012**, 6264–6275.

19. Yang, Z.; Johnson, M. Velocimetry based on dye visualization for a pulsatile tubing flow measurement. *Appl Opt.* **2019**, *58*, C7–C13.

20. Brody, W.R. Digital Subtraction Angiography. *IEEE Transactions on Nuclear Science* **1982**, 1176–1180.

21. Kawaji, M. Two-phase flow measurements using a photochromic dye activation technique. *Nuclear Engineering and Design* **1998**, 379–392.

22. Swinehart, D. F. The Beer-Lambert Law. *J. Chem. Educ.* **1962**, 39, 333–335.

23. Javadi, A.; Karbaschi, M.; Bastani, D.; Ferri, J.K.; Kovalchuk, V.I.; Kovalchuk, N.M.; Javadi, K.; Miller, R. Marangoni instabilities for convective mobile interfaces during drop exchange: Experimental study and CFD simulation. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **2014**, 441, 846–854.

24. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* **1996,** *58*, 67–88.

25. Ribeiro, M.D.; Rehman, A.; Ahmed, S.; Dengel, A. DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks. *arXiv preprint arXiv* **2020**, 08826.

26. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput* **1997**, *9*, 1735–1780.

27. Lathuilière, S.; Mesejo, P.; Alameda-Pineda, X.; Horaud, R. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence* **2019**.

28. Zhang, J.; Petitjean, C.; Lopez, P.; Ainouz, S. Direct estimation of fetal head circumference from ultrasound images based on regression CNN. *Medical Imaging with Deep Learning* **2020**, 121, 914-922

29. Raissi, M.; Yazdani, A.; Karniadakis, G.E.; Hidden fluid mechanics: learning velocity and pressure field from flow visualizations. *Science* **2020**, 367, 1026-1030.