

OPEN

# A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network

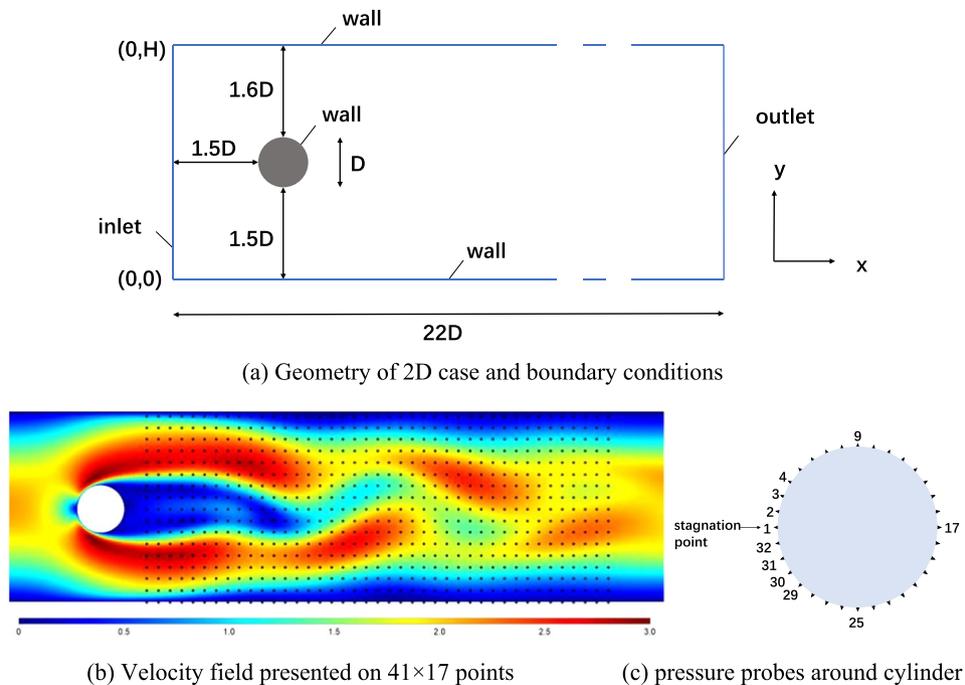
Shuran Ye<sup>1,2</sup>, Zhen Zhang<sup>1,2</sup>, Xudong Song<sup>3</sup>, Yiwei Wang<sup>1,2\*</sup>, Yaosong Chen<sup>3</sup> & Chenguang Huang<sup>1,2</sup>

In a myriad of engineering situations, we often hope to establish a model which can acquire load conditions around structures through flow features detection. A data-driven method is developed to predict the pressure on a cylinder from velocity distributions in its wake flow. The proposed deep learning neural network is constituted with convolutional layers and fully-connected layers: The convolutional layers can process the velocity information by features extraction, which are gathered by the fully-connected layers to obtain the pressure coefficients. By comparing the output data of the typical network with Computational Fluid Dynamics (CFD) results as reference values, it suggests that the present convolutional neural network (CNN) is able to predict the pressure coefficient in the vicinity of the trained Reynolds numbers with various inlet flow profiles and achieves a high overall precision. Moreover, a transfer learning approach is adopted to preserve the feature detection ability by keeping the parameters in the convolutional layers unchanged while shifting parameters in the fully-connected layers. Further results show that this transfer learning network has nearly the same precision while significantly lower cost. The active prospects of convolutional neural network in fluid mechanics have also been demonstrated, which can inspire more kinds of loads prediction in the future.

In the research field of fluid mechanics, the loads modeling of flow field has always been an important issue in complicated flow field. Fortunately, the emerging deep learning methods could provide more ideas for it. The big data era and the significantly improved computing power have laid a good foundation for the machine learning as well as its related applications. Therefore, the provided massive labeled flow field data makes it promising to expand the machine learning method to fluid mechanics. Early attempts were made mainly focusing on turbulence modeling. Tracey *et al.*<sup>1</sup> used neural networks with a single hidden layer to simulate terms in Spalart-Allmaras Reynolds Averaged Navier-Stokes (RANS) model, which suggests the possibility of neural networks in turbulence modeling. Ling *et al.*<sup>2</sup> proposed a multiplicative layer embedded with Galilean invariance to predict the Reynolds stress anisotropy tensor. This neural network architecture provides improved accuracy in Reynolds stress modeling than their previous work using random forest. Wang *et al.*<sup>3</sup> reconstructed the discrepancies in the RANS-simulated Reynolds stress using high-fidelity Direct Numerical Simulation (DNS) data based on machine learning methods. This method offered a different way to interpret the improved prediction.

Above all, more study needs to be done to figure out the possibilities machine learning owns in fluid mechanics, not only on turbulence modeling, but on flow feature study as well. Success on ImageNet Classification with deep convolutional neural networks makes the convolutional neural network popular for its fewer connections and parameters<sup>4</sup>. Umetani and Bickel<sup>5</sup> proposed a data driven method to predict the fluid flow around three-dimensional objects. A novel PolyCube maps-based parametrization was adopted to instantly calculate the nonlinear response of the flow using a Gaussian process regression. Ströfer *et al.*<sup>6</sup> used convolutional neural

<sup>1</sup>Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China. <sup>2</sup>School of Engineering Science, University of Chinese Academy of Sciences, Beijing, 100049, China. <sup>3</sup>College of Engineering, Peking University, Beijing, 100871, China. \*email: [wangyw@imech.ac.cn](mailto:wangyw@imech.ac.cn)



**Figure 1.** Flow geometry and probes distribution in the flow around a cylinder. (a) Geometry of 2D case and boundary conditions. (b) Velocity field presented on  $41 \times 17$  points in the wake flow of the cylinder. (c) Indexes and locations of pressure probes around cylinder.

network to identify the features in fluid flow. This attempt received good results, which provides a general method to flow feature identification study even for distinguishing between similar ones. Jin *et al.*<sup>7</sup> also used the convolutional network to predict the velocity field around the cylinder. This method can detect the invariant features of the small translations in the temporal dimension of pressure fluctuations on the cylinder. Liu *et al.*<sup>8</sup> proposed a shock wave detection method based on CNN, which has better performance on detection result than the first step to introduce deep learning into shock detection method with less time consumption than the traditional approach. These papers proved that the convolutional neural network have the critical advantages on feature detection in fluid mechanics.

Therefore, we began to focus on the relationship between the flow structure and the concerned flow statistics. The convolutional neural network is constructed in hope of finding the mapping relation between them. Flow around a cylinder is taken as the example to discuss the proposal. This type of flow is a common flow structure owning certain complexity as well. Based on the classic non-uniform flow field, the network between the flow structure and the flow concerned amount is constructed. This CNN-based deep learning method links the flow field to the pressure on cylinder by an intermediate variable which presents the feature learned from the flow field.

The rest of this paper is described as follows: In Sect.2, we describe flow data and the configuration of the CNN. The results of the network and the evaluation are presented in Sect.3, and the explanation of the convolution operation is discussed in Sect.4. In Sect.5, transfer learning is introduced to improve the network efficiency. Finally, Sect.6 concludes the paper.

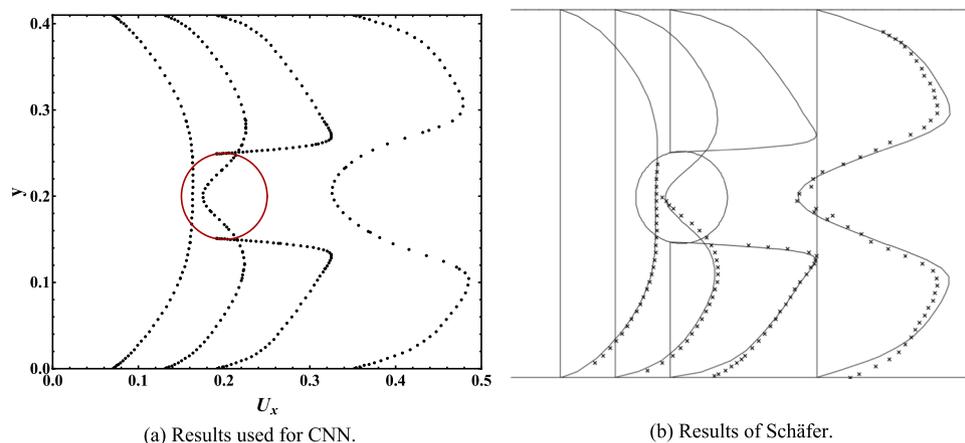
## Methodology

**Flow modeling & data acquisition.** The flow around a cylinder has always been a typical project in fluid mechanics, due to its critical influence in engineering problems. When the Reynolds number increases, the wake flow passing the cylinder turns into an unsymmetrical state, generally presented as vortex shedding alternately, which will cause the fluctuating pressures on the surface of the cylinder<sup>9</sup>.

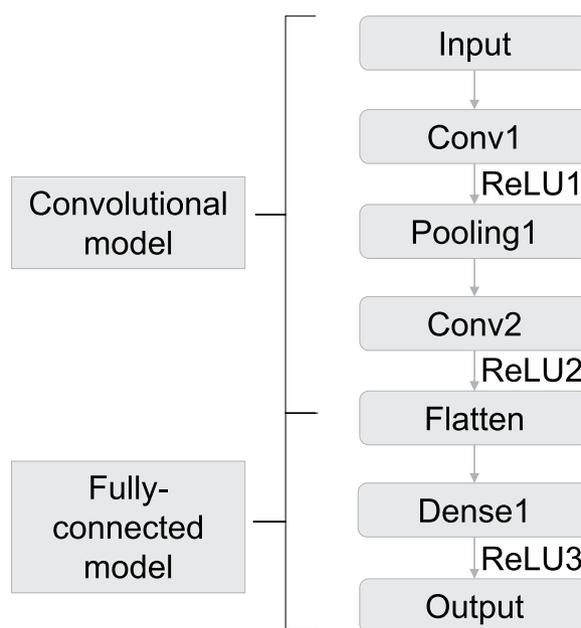
Therefore, we focus on the flow around a circular cylinder whose pressure on the surface changes along with the velocity field. For the flow around a cylinder with a circular cross-section, the two-dimensional case is adequate to express flow features<sup>10</sup>. The flow geometry along with boundary conditions are shown in Fig. 1a. The profile of inlet horizontal velocity is set as a parabolic function:  $u(y) = \frac{4u_m y(H-y)}{H^2}$ . And the vertical component of velocity at inlet is set as zero.

Reynolds number ranges from 1000 to 10000, which is defined by  $Re = \bar{u}D/\nu$  with the mean velocity  $\bar{u} = 2u(H/2)/3$  and cylinder diameter  $D = 0.1$ . For each Reynolds number, flow with 300 frames are calculated where time step  $dt = 0.005$  s. Pressure coefficient  $C_p$  is chosen as the network target considering its non-dimensional property, and is defined as  $C_p = \frac{P}{\frac{1}{2} \times \rho \times \bar{u}^2}$ , where  $P$  is the local pressure and  $\rho$  is the fluid density.

To match the deep learning configuration, data we use for the network should be preprocessed. The wake flow area in range  $0.3 \leq x \leq 1.3$  and  $0 \leq y \leq 0.4$  is chosen as the velocity field area. A mesh of  $41 \times 17$  point probes is



**Figure 2.** Velocity profile of flow around a cylinder. (a) Numerical simulation results used for CNN. (b) Numerical simulation and experimental results of Schäfer.

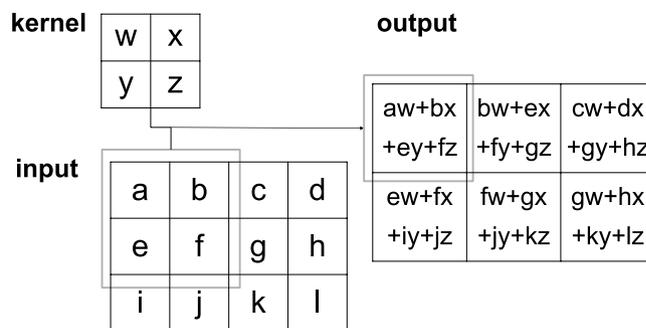


**Figure 3.** The architecture of the CNN to train the model.

created using interpolation method at the distance of 0.025 to represent the velocity field (Fig. 1b), and 32 probes are located on the cylinder surface to monitor the pressure coefficient on the cylinder, respectively (Fig. 1c).

The result of our numerical simulation in the flow around the cylinder is shown in Fig. 2 to discuss its validity. Figure 2a is the velocity profile in our numerical simulation, while Fig. 2b is the velocity profile carried out by Schäfer<sup>10</sup>. By comparing the velocity profile results around the cylinder, we find that our result shows good consistency with the Schäfer's data.

**Construction of network.** The architecture of the CNN is shown in Fig. 3. “Conv” denotes convolutional layer; “ReLU” denotes rectified linear unit; “Pooling” denotes max pooling layer; “Dense” denotes fully connected layer<sup>11</sup>. The network is composed of two main parts. The convolutional layers are an important part in a CNN. Convolution is a mathematical operation, which presented as a weighted average method (Fig. 4). By allocating the weight function as  $w$ , its common expression is presented as  $s(t) = (r * g)(t)$ . In this expression, the function  $r$  in the convolution is often referred as the input, while the function  $g$  is referred as the kernel. Considering the discrete time term as well as the two-dimensional kernel for the convolutional operation, this expression can be written as:  $s[i, j] = (I * K)(i, j) = \sum_m \sum_n I[m, n] K(i - m, j - n)$ <sup>4</sup>. Since the kernel size is far smaller than the input size, it leads to the critical improvements: sparse interactions and parameter sharing. Another improvement in efficiency called equivariant representations is achieved by pooling layer. Pooling operation makes the output of the network replaced by the overall statistical characteristics of adjacent outputs. Max pooling that gives the max value of the given location is one of the pooling operations<sup>12</sup>. In the convolutional model shown in Fig. 3, the



**Figure 4.** Convolution: the kernel size is  $2 \times 2$ .

CNN architecture is constructed considering both the classical simple network LeNet5 and the input size of our network, while the maxpooling whose pooling size is  $2 \times 2$  and stride is 1 is adopted.

Also, the fully-connected layers are a basic network form in deep learning methods, which is also known as the multilayer perceptron. Each node in one layer is connected to every node in the following layer. This type of network is inspired by the neuron model in biological neural networks, whose function is to pass information through connections. As for the neurons in neural network model, the relation is expressed as  $f(a; w; b) = a^T w + b$ , where  $w$  is the weight and  $b$  is the bias. In the network described in Fig. 3, the dense layer with 128 neurons is chosen.

In order to allow the neuron performed as nonlinear relation, activation function is added following both the convolutional layer and the fully-connected layer. The activation function in the network is Rectified Linear Unit (ReLU) whose expression is:  $f(z) = \max(0, z)$ . The results of the convolutional layer are collected by the fully-connected layer to get the final output. Hence the whole structure of the CNN is constructed.

**Training configurations.** Considering the construction of the neural network under the purpose of pressure coefficient prediction, the input of this network is supposed to be the velocity of the flow field. The velocity is presented as a three-dimensional matrix  $[x, y, \mathbf{U}]$ , where  $\mathbf{U}$  stands for  $\{u, v\}$  in the two-dimensional flow. It has been noted that all the inputs and outputs are non-dimensional. Therefore, all data needs to be preprocessed, non-dimensionalized by  $\rho$ ,  $\bar{u}$  and  $D$ .

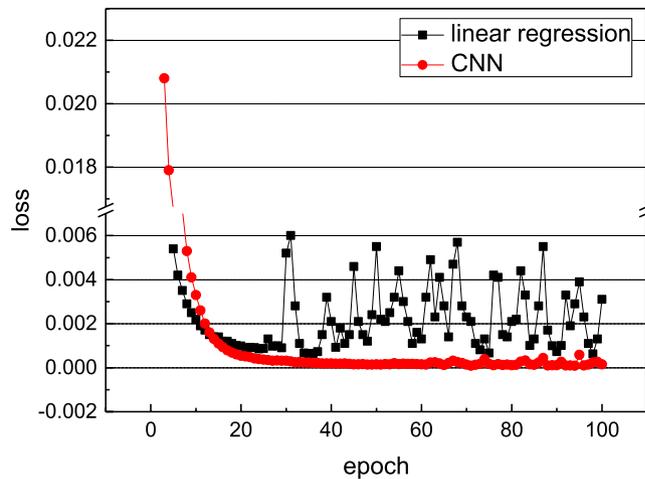
The loss function is a non-negative real-valued function which measures the deviation between the predicted value and the CFD results<sup>13</sup>. The loss function is chosen as the root mean square error (RMSE). It presented as  $RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{n}}$ , where  $y_i$  is the CFD results and  $\tilde{y}_i$  is the output of the network<sup>14</sup>. Thus this training process is converted into an unconstrained optimization problem, finding the proper variables which lead to the minimum value of the loss function. Gradient descent algorithms are basic methods to attain the optimized value and Adam algorithm is one of the most universal methods for its effectiveness. It is an algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments. The learning rate of the Adam algorithm is 0.001 and the beta value is  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . Monitoring the continually decreasing RMSE, the neural network gradually approaches to the real mapping relation<sup>15</sup>. Noted that the mini-batch training is used whose batch size used for training is 64. The CNN training process is carried out with TensorFlow, an end-to-end open source platform for machine learning<sup>16</sup>.

## Result of Pressure Modeling with The Typical CNN Network

**Result on basic data.** The training set is the flow around cylinder cases at Reynolds number equaling to 1000, 3000, 5000, 7000, and 9000, while the testing set is the flow around cylinder at Reynolds number equals to 2000, 4000, 6000, 8000, and 10000. The data is acquired under the condition described in Section 2.1. The same cylinder was used while the different Reynolds number is achieved with different viscosity coefficient. Noted that the  $Re = 10000$  is an extrapolate testing data set which can better illustrate the feasibility range of this method. For each Reynolds number, 300 frames are calculated where  $dt = 0.005$  s. Therefore, the size of training samples and the testing samples are both 1500.

With the iteration of the losses and the weights in the network, this network gets closer to the real mapping relation. By back propagation to the weights in each layer, the loss function which stands for the RMSE drops rapidly. As shown in Fig. 5, the training error for entire dataset decreases with the increasing number of training epoch. When the epoch reaches to 50, the loss of this network estimates to  $2 \times 10^{-4}$ . It means the root mean square error between the real pressure coefficient and the predicted pressure coefficient is  $2 \times 10^{-4}$ . To better understand the CNN model which is not only a physics informed neural network, but has improved performance as well, a comparison with linear regression is carried out. The linear regression is constructed by the dense layer in the network without using the activation function. Noted that the number of the layers will not change the fact that it is still a linear regression model. We train this model in the same way as we train our CNN model. Figure 5 shows the comparison of the loss between the CNN model and the linear regression model. Noted that the pressure coefficients range normally in  $(-3, 3)$  as they are non-dimensionalized by the mean velocity  $\bar{u}$ .

Now we focus on its trained network on the behavior on testing data. When the flow condition changes, the adaptive ability of the network is discussed. The Reynolds number is changed in testing data. By assessing



**Figure 5.** Convergence of the CNN and the linear regression model.

	Reynolds number	RMSE for CNN	RMSE for linear regression
Train data	1000, 3000, 5000, 7000, 9000	$2.5 \times 10^{-4}$	$9.0 \times 10^{-4}$
Test data	2000, 4000, 6000, 8000, 10000	$3.4 \times 10^{-3}$	$1.4 \times 10^{-2}$

**Table 1.** The RMSE on train data and test data.

the input of the test data, Table 1 gives the overall RMSE between the predicted pressure coefficient and the CFD-calculated pressure coefficient in CNN model, which is around  $3.4 \times 10^{-3}$ . The CNN model has better performance on RMSE than the linear regression on both train data and the test data. The CNN performance on  $Re = 10000$  is shown in Fig. 6. As shown in picture, CNN predicted results show good agreements with the CFD results as reference values in the test data set.

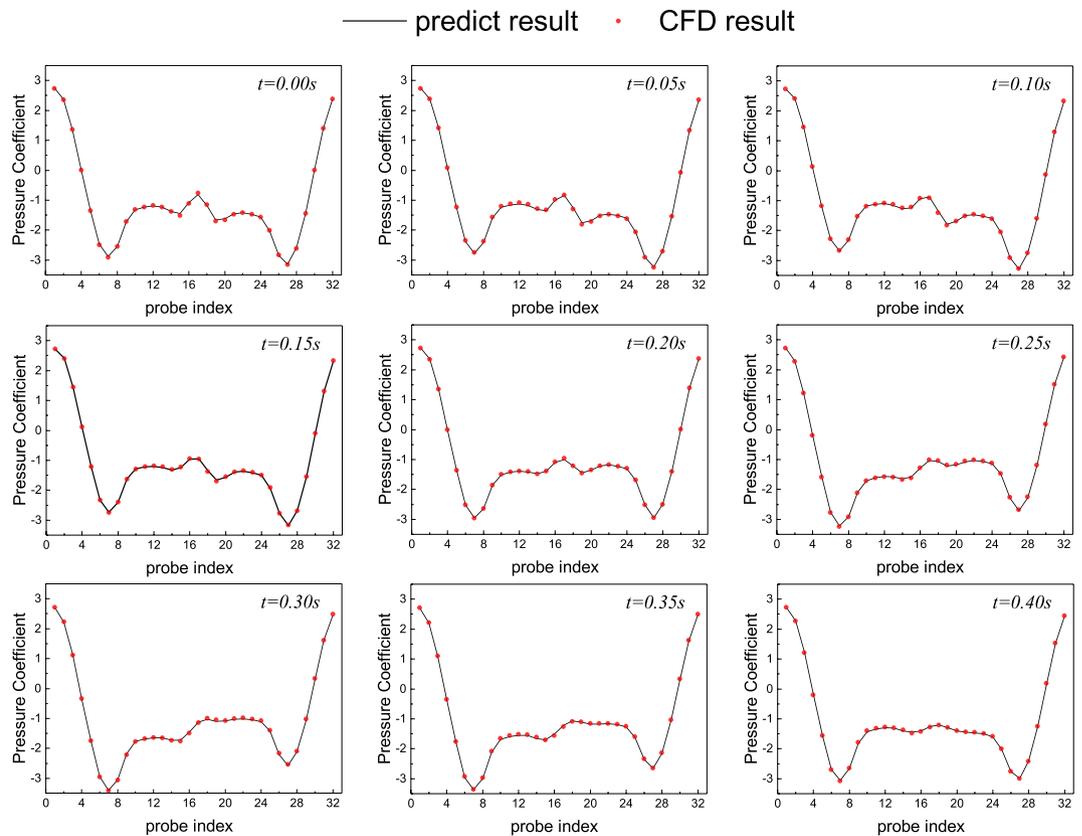
**Independence verification of sampling resolution.** In some computational fluid dynamics cases, computing requires to verify the grid independence before further study<sup>17</sup>. Inspired by this CFD computing strategy, similar verification on input data size is carried out with great importance. As for the velocity field chosen as the network input, it is necessary to ensure the size of the velocity information are representative enough to present the flow structure. Therefore, higher spatial resolution of  $341 \times 101$  is interpolated on the same case for the independence study of sampling resolution. To catch up with the larger input size of  $341 \times 101$ , more convolutional layers was constructed during this operation. The new CNN configuration with more layers is shown in Fig. 7. The two convolutional layers with one pooling layer is regarded a module, which is repeated three more times to achieve the final architecture.

From the Table 2 the presentation of different spatial resolution can be concluded. When the velocity field changes from  $41 \times 17$  points to  $341 \times 101$  points, the RMSE increases from  $2.5 \times 10^{-4}$  to  $2.8 \times 10^{-4}$ . With the increase of the velocity field points, the error of the network remains small. Though more data is added, the effect on the results of network is negligible. It can be regarded that the verification by refining the spatial resolution of data sampling is achieved while the data of  $41 \times 17$  points satisfies the requirements of calculation. Therefore, considering the cost of calculation time, the network structure with the input size of  $41 \times 17$  is also chosen for further study.

**Discussion on the feature detection mechanism of CNN.** *Interpretation of CNN by visualization.* We have mentioned that the flow feature can be extracted from the flow field information, which leads to further derivation to typical parameters as well as main description of the flow field. Now that the above results are the successful attempt to this theory is supposed to be admitted, it suggests that it is meaningful to explore the understanding of convolutional layers for further verification.

Though deep learning is normally regarded as black box for its internal algorithm invisible, convolutional neural network can be visualized. Visualization can help understanding the procedure of CNN training<sup>18</sup>. The visualization technique is to maximize the activation of the filters in different layers, presenting which patterns the data set activates in the convolution process. Therefore, it gives the insight of the features which filters are interested in. To better comprehend the feature detection process, same physical condition data source with higher spatial resolution data is adopted. The architecture of the CNN for higher spatial resolution is described in Section 3.2. Higher spatial resolution points help better describing the physical meaning of the filters and give explanation of the convolutional operation more clearly.

The first two lines in Fig. 8 shows some of the features in a fully trained model. Visualization on filters emphasizes specific features during the convolutional neural network. When velocity field entered in the input of the



**Figure 6.** Comparisons of pressure coefficient on 32 points on the cylinder between the model predictions and CFD results with Reynolds number = 10000 at different moments. The red line is regarded as the standard result which we calculated in CFD method before, and the blue line stands for the predicted results through the CNN.

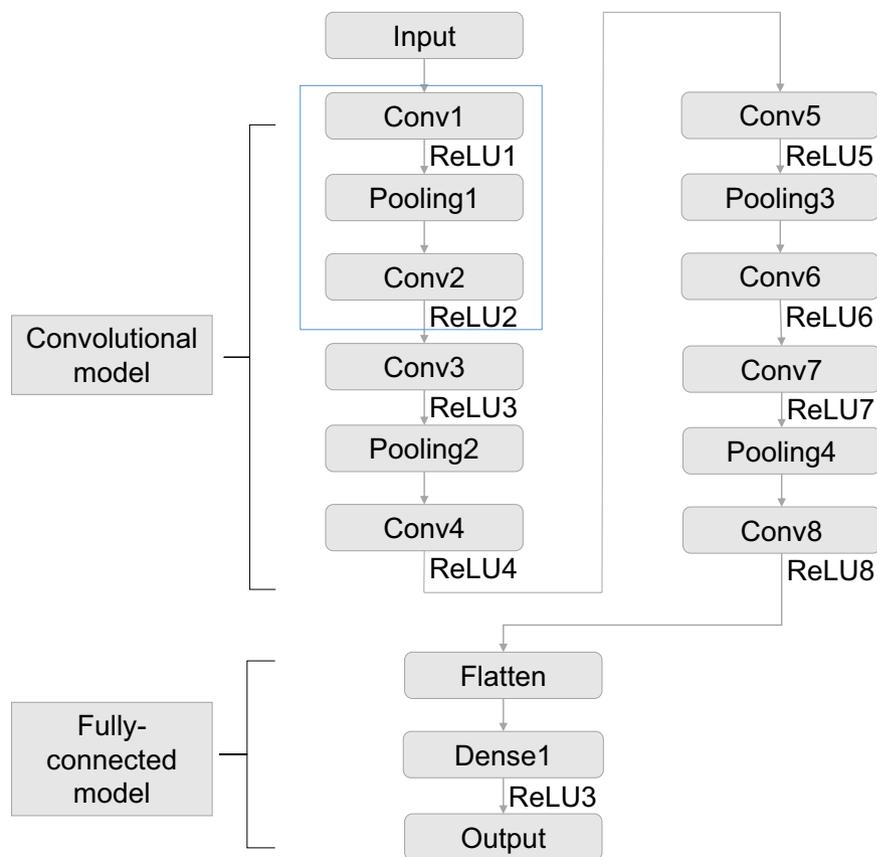
network, filters extract corresponding feature and reconstruct to the final output. Six activations of feature visualization separately belonging to 3 layers are shown in Fig. 8. Layer 2 responds to lines. Layer 4 shows variation from layer 2, and is more vortex-like structure. While layer 6 shows the edge of shedding vortex. The last line gives the output of those layer, indicating the contribution of the convolutional layer in a different way. Significant similarities between this CNN in flow field and the common CNN aiming for image recognition is confirmed. Therefore, it is concluded that CNN could detect features in flow field, and apparently is a composite process with gradually complex expression.

**Regression for reynolds number.** Owing the results in Section 3, the successful prediction in pressure coefficient among different Reynolds numbers leads us to further study in the network. Reynolds number is the key dimensionless value label for the flow state. Therefore, the same network configuration is adopted to discuss whether the neural network structure can afford the classification of flow states according to the Reynolds number.

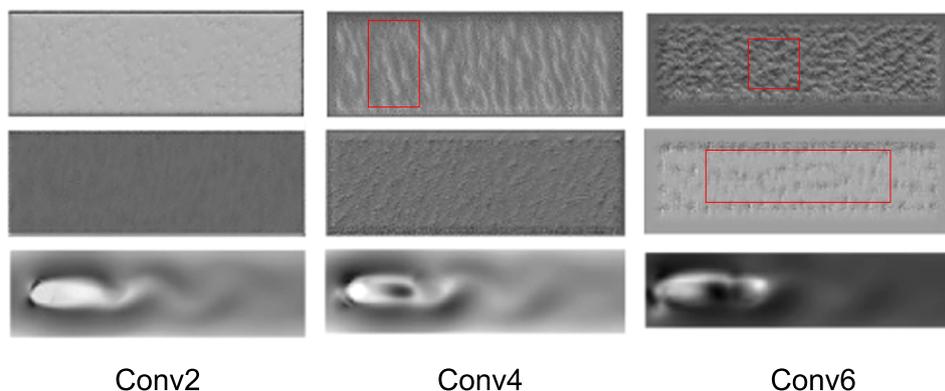
Output of a classification problem is discrete value, which is used to specify which category it belongs to<sup>19</sup>. However, it means the Reynolds number we expect the network to predict should be trained before. Though it is apparently a classification problem, for better understanding of the network we use a regression prospect to deal with it. The network has the same configuration with the previous CNN expect for the output layer, which in this case the output layer only has one neuron indicating the Reynolds number. The regression model is a function that represents the mapping between input variables and output variables. The learning of regression problem is equivalent to function fitting: a function curve is selected to fit the known data well and predict the unknown data well<sup>20</sup>. In this case, this regression will clarify the Reynolds number merely based on the flow velocity, even for those untrained Reynolds number condition.

The training set is the flow around cylinder cases at Reynolds number equals to 1000, 3000, 5000, 7000, and 9000, while the testing set is the flow around cylinder at Reynolds number equals to 2000, 4000, 6000, 8000, and 10000. For each Reynolds number, 300 sets of data are used for training and testing, which is exactly the same as the pressure coefficient prediction condition. Compared with the pressure coefficient prediction case, the change exists in the output of the network from pressure coefficient to Reynolds number.

After 100 epochs of training, the RMSE on training data is 2.1, while the testing score reaches to 8.6. Based on the Reynolds number normalization, the RMSE on both training data and testing data are decreased by 4 orders of magnitude, which means the normalized RMSE on training data and testing data is  $2.1 \times 10^{-4}$  and  $8.6 \times 10^{-4}$ , respectively. The predicted Reynolds numbers are nearly identical to the CFD results (Fig. 9), indicating



**Figure 7.** The architecture of the CNN for higher spatial resolution.

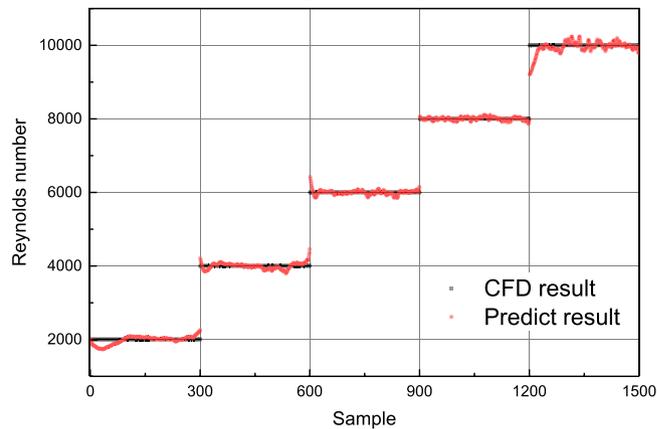


**Figure 8.** Visualizing of features in a fully trained model on layers: Conv2, Conv4, Conv6.

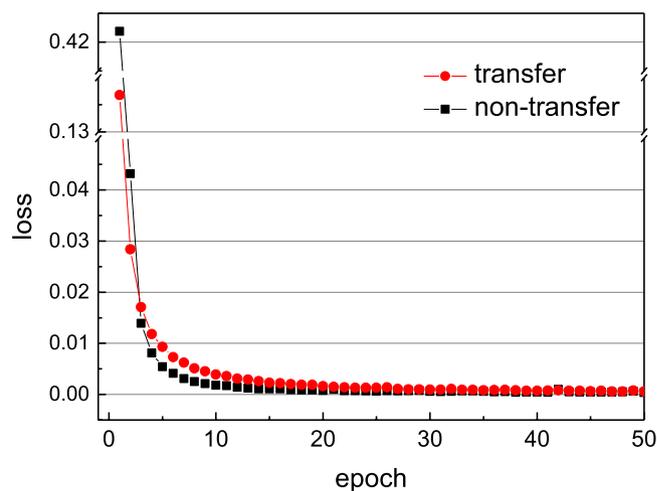
	RMSE for training data	RMSE for testing data
$41 \times 17$ points	$2.5 \times 10^{-4}$	$3.4 \times 10^{-3}$
$341 \times 101$ points	$2.8 \times 10^{-4}$	$4.2 \times 10^{-3}$

**Table 2.** The presentation on data in different spatial resolution.

that the network can make good predictions on Reynolds numbers. The result shows that the neural network structure can also figure out the Reynolds number as the main property of the flow. Combined with the pressure coefficient discussed before, we can agree on the flow features learned from the flow velocity field are with enough representativeness. The detected feature in the network could export both flow characteristics and flow property in flow field.



**Figure 9.** Comparisons of Reynolds number between the model predictions and CFD results for different data set under various Reynolds numbers.



**Figure 10.** Convergence of the CNN on non-transfer and transfer method: training error for entire test dataset decreases with the increasing number of training epoch.

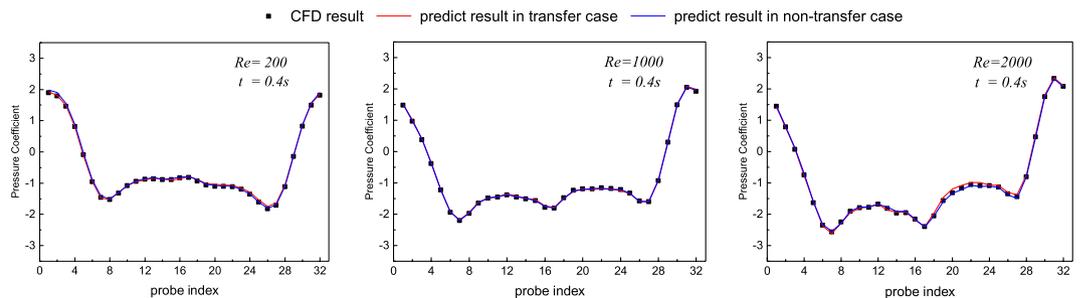
**Result on transfer learning for high efficiency.** The traditional supervised learning meets difficulties when insufficient labeled data for the task is available. In that case, we train the network with the source data and then transfer some layers to the target network<sup>21</sup>. The feature-representation-transfer approach is one of the examples. Owing the labeled data in the source domain, the feature representation can be built through the supervised learning by convolutional neural network. The key for transfer learning is to construct the good feature as well as share the parameters distributions with the target tasks. The transfer learning approach we adopted here is fine-tuning, which is achieved by freezing some of the layers while training the rest layers of the target network. The rest of the weights in the neural network adjust under the target training data set to make sure that this network can achieve better performance in the target domain.

We now focus on how transfer learning improves the computational efficiency in the flow around a cylinder case. The source domain in this transfer learning is the data set in the case of the original parabolic velocity inlet stated in Section 2.1 with Reynolds number ranging from 1000 to 10000. The target domain owns the data set under the condition of parabolic velocity inlet with an additional perturbed term defined as:  $u_t = u(y) * \left[ 1 + \sin\left(\frac{6\pi}{0.41} \times y\right)^2 \right]$ . Reynolds numbers in target domain are ranged from 100 to 2000. Admitting the convolutional layer aiming at feature representation, weights in this layer is transferred to the target model. Therefore, weights in those layers from new model are fixed as constant value, leaving the fully connected layer as the only trainable part.

This training approach could reduce the loss more quickly as shown in Fig. 10 and Table 3. The transfer learning method achieves nearly the same accuracy as the traditional one but with significantly lower time cost. Prediction of pressure coefficient on transfer learning approach is shown in Fig. 11. Comparing with the non-transfer learning approach, we can agree on that both methods are able to present the pressure coefficient. Results show that this fixed layers in the convolutional neural network aiming for the feature detection is universal in this kind flow feature recognition problem.

Method	RMSE for train data	RMSE for test data	Time per 50 epoch*
Non-transfer	$3.4 \times 10^{-4}$	$8 \times 10^{-4}$	3615.7 s
transfer	$5.7 \times 10^{-4}$	$1.1 \times 10^{-3}$	80.3 s

**Table 3.** The presentation on non-transfer and transfer method. \*This runtime includes the 50 training epoch only. We use an Intel Core i7-8700 CPU at 3.20 GHz, 32.0 GB memory.



**Figure 11.** Pressure coefficient on 32 points on the cylinder between the model predictions and CFD results for various Reynolds numbers in transfer case and non-transfer case.

## Conclusion

In this work, we propose a convolutional neural network and apply it to the flow analysis. The trained convolutional neural network makes successful prediction on the pressure distributions on the cylinder based on the velocity field in its wake flow. It suggests that the relationship between the flow field and the pressure coefficient in different time steps and Reynolds numbers is deduced. The RMSE of the predicted results and the CFD results is less than 0.004.

This achievement suggests that the CNN could be managed to detect the flow feature. The detected flow feature is correlated with the flow parameters by fully-connected layers. Explanation of how a CNN sees a flow field is preliminarily discussed, as the function of the convolutional layers is visualized. Meanwhile, the regression of the Reynolds number using the same network structure confirmed the CNN could learn the essence of the flow feature.

Further, the generalization ability of the network is investigated by setting up a transfer learning approach of prediction for different velocity inlets and Reynolds numbers, which makes the higher efficiency on the network in both training and testing data sets. With 3 orders of magnitude speedup compared to the non-transfer method, the transfer learning nearly reaches the same accuracy.

In addition, the method we proposed is capable of providing a favorable prediction on the pressure coefficient under different conditions. An obvious next step is to extend the convolutional neural network method to bigger data set of more complex flow fields. The present method and the network system could be an effective strategy for engineering situation in fluid mechanics.

Received: 12 October 2019; Accepted: 24 February 2020;

Published online: 10 March 2020

## References

- Tracey, B., Duraisamy, K. & Alonso, J. J. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 2013*, <https://doi.org/10.2514/6.2013-259> (2013).
- Ling, J., Kurzwaski, A. & Templeton, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166 (2016).
- Wang, J. X., Wu, J. L. & Xiao, H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2**, 1–22 (2017).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 1097–1105 (Curran Associates Inc., 2012).
- Umetani, N. & Bickel, B. Learning three-dimensional flow for interactive aerodynamic design. *ACM Trans. Graph.* **37** (2018).
- Michélen Ströfer, C., Wu, J., Xiao, H. & Paterson, E. Data-Driven, Physics-Based Feature Extraction from Fluid Flow Fields. *Commun. Comput. Phys.* **25**, 625–650 (2018).
- Jin, X., Cheng, P., Chen, W. L. & Li, H. Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Phys. Fluids* **30**, 047105 (2018).
- Liu, Y. *et al.* A CNN-based shock detection method in flow visualization. *Comput. Fluids* **184**, 1–9 (2019).
- Norberg, C. Flow around a circular cylinder: Aspects of fluctuating lift. *J. Fluids Struct.* **15**, 459–469 (2001).
- Schäfer, M., Turek, S., Durst, F., Krause, E. & Rannacher, R. Benchmark Computations of Laminar Flow Around a Cylinder. in *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995* (ed. Hirschel, E. H.) 547–566 (Vieweg+Teubner Verlag, 1996), [https://doi.org/10.1007/978-3-322-89849-4\\_39](https://doi.org/10.1007/978-3-322-89849-4_39).
- Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **1409**, 1556 (2014).
- He, K., Zhang, X., Ren, S. & Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 1904–1916 (2015).

13. Zhang, Z. *et al.* Application of deep learning method to Reynolds stress models of channel flow based on reduced-order modeling of DNS data. *J. Hydrodyn.* **31**, 58–65 (2019).
14. Ye, S. *et al.* Applications of autoencoder in reduced-order modeling of flow field. *Acta Aerodyn. Sin.* **37**, 498–504 (2019).
15. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic gradient descent. *ICLR Int. Conf. Learn. Represent.* (2015).
16. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (2016).
17. Huang, H., Yang, X. & Lu, Xyun Sedimentation of an ellipsoidal particle in narrow tubes. *Phys. Fluids* **26**, 053302 (2014).
18. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53) (2014).
19. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability* 281–297 (1967).
20. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature*, <https://doi.org/10.1038/nature14539> (2015).
21. Long, M., Zhu, H., Wang, J. & Jordan, M. I. Deep transfer learning with joint adaptation networks. In *34th International Conference on Machine Learning, ICML 2017* (2017).

## Acknowledgements

This work was supported by the National Natural Science Foundation of China, grant numbers 11772340, and 11672315, the Youth Innovation Promotion Association CAS (2015015).

## Author contributions

S.Y. designed the network, performed training, analysis the result and wrote the manuscript, Z.Z. performed preprocessing, and helped in discussing the outputs of the network, X.S. prepared Figures with CFD solver, Y.W. lead the project, and reviewed the manuscript. C.H. and Y.C. gave suggestions for this research. All authors discussed extensively the results.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Y.W.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**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 license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license 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 license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020