Prediction of confined flow field around a circular cylinder and its force based on convolution neural network

YONG ZHAO, XUEHUI JIANG ZHAO, YANG MENG
College of Naval Architecture and Ocean Engineering, Dalian Maritime University, Dalian, 116026 China

Corresponding author, Associate Professor Yong Zhao, E-mail: fluid@dlmu.edu.cn

This work is supported by the National Natural Science Foundation of China (No. 51679021) and China Scholarship Council (No.201906575002).

ABSTRACT In this paper, a large number of numerical simulations of the external confined flow field around a circular cylinder is conducted by using the open source library OpenLB. The boundary information of the flow field is defined by Euclidean distance function. Then, massive data, including the velocity field and the drag and lift coefficients of the cylinder, are obtained and used for model training. Subsequently, two convolutional neural network models are presented. The input ends of the two models are the boundary information of the flow field, and the output ends are the velocity field and the force on the cylinder respectively. Once the two models are established, they are used to predict the velocity and force respectively. The predicted results are in good agreement with the real values. It turns out that the convolution neural network models can be used to predict the flow field and force with complex boundaries, and it has the advantages of high accuracy and efficiency in practical engineering applications.

INDEX TERMS Convolutional neural network, deep learning, computational fluid dynamics, OpenLB.

VOLUME XX, 2021

I. INTRODUCTION
Flow around a circular cylinder is a classical problem in fluid mechanics, which has many important engineering applications. Many research studies have been carried out regarding this problem in the past decades, such as computational fluid dynamics (CFD) method based on Navier-Stokes equation [1], Boltzmann equation [2], particle image velocimetry [3]. In recent years, with the wide application of computational fluid dynamics, a large number of data have been accumulated. On the other hand, with the accumulation of big data, machine learning models, especially deep learning algorithms, are developing rapidly, and have been successfully applied in many fields. Thus, it is feasible to apply deep learning to fluid dynamics [4-8]. This modeling paradigm of exploring physical mechanisms with deep learning models is called data-driven model [9,10]. As one of the most promising deep learning models, the convolutional neural network (CNN) has wide successful applications [11-13], especially in image classification [14], recognition [15] since it was first proposed by Lecun [16]. For the combination of CNN model and CFD method, some papers has been published recently [17-19]. Jin et al. [20] used CNN to establish the mapping relationship between the pressure fluctuations on the cylinder and the velocity field around the cylinder using the CFD dataset. The trained model is then tested over various Reynolds numbers. The predictions of this model agree well with the CFD results. Bhatnagar et al. [21] adopted the CNN to approximate the flow field around airfoil using the CFD data from Reynolds Averaged Navier–Stokes (RANS) flow solutions. A model based on convolutional neural networks (CNNs) is proposed for flow field predictions by Bhatnagar et al. [21]. The results show that it’s possible to study the impact of the airfoil shape and operating conditions on the aerodynamic forces and the flow field in near-real time. Murata et al. [22] presented a nonlinear model, named decomposing CNN auto coder to extract the future of flow field around the circular cylinder. However, these works are the prediction of flow around a circular cylinder in an unbounded flow field. The existence of the external boundary will increase the complexity of the flow field. In order to investigate the bounded flow around the circular cylinder, we propose a deep convolution neural network model which is capable of predicting the flow field and force on the cylinder.

The structure of this article is as follows: Sec. II
illustrates the bounded flow around a circular cylinder calculated by CFD solver OpenLB. The target for identifying boundary information is realized by Euclidean distance function. Then, in Sec. III, the CNN model structure is introduced. Two CNN models, named as CNN-FORCE and CNN-FORCE, are proposed to predict the force of the cylinder and the velocity field around the cylinder, respectively. Section IV evaluates the general performance of the proposed models. Finally, a summary is provided in Sec. V.

II. CFD of BOUNDED FLOW AROUND CIRCULAR CYLINDER and BOUNDARY INFORMATION RECOGNITION

The bounded flow diagram is depicted in Fig.1. The length and the width are 2.2m and 0.411m, respectively. The radius \( r \) of the cylinder is 0.05m, and the position of the cylinder center is limited to the red region. The kinematic viscosity of the fluid is 0.001m/s and the density is 1 kg/m³. The bounded flow is simulated by the open source code library OpenLB. The computational cases in this paper include 15 groups according to \( \text{Re} \) equals 10, 20, 30..., 150, respectively. Note that the flow Reynolds numbers are determined with a fixed cylinder diameter of 0.1m. In each group, the cylinder is randomly positioned 500 times, so, the total number of cases is 7500, which forms an extensive database to train the deep CNN model.

**FIGURE 1.** The diagram of bounded flow around cylinder.

Flow boundary shape is the input end of CNN, which is represented by Euclidean distance function. The Euclidean distance from each point to the cylinder boundary is defined as follows,

\[
d_{ij} = (\min \| r(x_i, y_i) - r(x_r, y_r) \|)b, \quad (1)
\]

where \( i=1, 2, \ldots, m, j=1, 2, \ldots, n \). \( m, n \) represents the dimension of the input data in the x, y direction, here are 223 and 44 in CFD, respectively. \( r \) represents the distance from the center of the bluff body to those points, \( r \) represents the geometric boundary of a bluff body. If the position of any point is in the inner or the boundary of the bluff body, then \( b=0 \). If the position of any point is outside the geometric boundary of the bluff body, then \( b=1 \).

Multi-dimensional input information including Reynolds number and the position of the cylinder relative to the boundary of the computational field. For the input end, different boundary types are represented by different numbers. For example, the velocity entry boundary is denoted as number 3, the exit boundary is denoted as 4, the upper and lower boundary is denoted as 2, the entire boundary of the computational field and the inside of the cylinder are denoted as 0, and other positions of the computational field are defined as 1. These defined values constitute a matrix representing the unique geometric information, which is illustrated in Fig.2. As described above, the neural network can distinguish the various types of boundary and the cylindrical boundary at various Reynolds numbers.

**FIGURE 2.** The flow field boundary information is used as the input end of CNN

In order to verify the accuracy of LBM numerical method in this complex flow, the stream velocity at \( t=30s \) compared to NS solver in Figure 3, in which profile position is located at two cylinder diameters behind the cylinder, where \( \text{Re}=100 \). The two curves agree well, which shows that the accuracy of LBM method is pretty good.

**FIGURE 3.** Comparison between stream velocity distribution by LBM numerical method and NS solver at \( t=30s, \text{Re}=100 \).
III. CONSTRUCTION OF CONVOLUTIONAL NEURAL NETWORK

A. THE STRUCTURE OF CONVOLUTIONAL NEURAL NETWORKS

CNN model can automatically learn and extract features from data and its generalization ability is significantly better than traditional methods. The network structure is mainly composed of input layer, convolution layer, pooling layer, full connection layer, and output layer. In this paper, two structures, named as CNN-VELOCITY and CNN-FORCE are designed for predicting the velocity field, drag coefficient, and lift coefficient, respectively.

1) CONVOLUTION LAYER, DECONVOLUTION LAYER AND ACTIVATION FUNCTION

In order to extract the different characteristics of computational field boundary, a multilayer convolution layer is introduced in this paper. In the convolution layer, the feature graph of the previous layer is convolved with the convolution kernel. The convolution kernel slides on the feature graph with a fixed step size, and the result of convolution forms the feature graph of the layer under the action of the nonlinear activation function ReLU of the layer. The formula of the convolution layer is,

\[ y_c = \text{ReLU}(\kappa \ast x_c + b_c) \]  

where \( \kappa \) represents convolution kernel, \( x_c \) is the input characteristic graph, \( \text{ReLU} \) is the nonlinear activation function and \( y_c \) represents the output of the convolution layer. The convolution kernel may not be convolved at the boundary of the feature graph because it is not divisible. We add \( \theta \) artificially in the feature graph, which means we add a zero-padding layer in the network. To visualize the output velocity field information in CNN-VELOCITY model, we set up the deconvolution layer as shown in Fig. 4. The blue and green represent the input and output respectively.

\[ h(c) = \text{ConvNN}(c) \]

\[ i(c) = \text{DeconvNN}(h(c)) \]

where \( \text{ConvNN}(.) \) represents the mapping of data to eigenvectors. \( \text{DeconvNN}(.) \) represents the mapping of eigenvectors to output data. The velocity field information with the same dimension as the input information is obtained by deconvolution.

2) POOLING LAYER

The pooling layer is generally set after the volume accumulation layer, and the pooling operation can be carried out according to a certain rule. Such as average pooling and maximum pooling [23]. The average pooling operation is shown in Fig. 5. The pooling layer can reduce dimension and highlight the characteristics of the data. Assuming pooling layer, then:

\[ H_y = \text{subsampling}(H_{y-1}) \]

\[ \begin{array}{cccc}
3 & 3 & 2 & 1 \\
0 & 0 & 1 & 3 \\
3 & 1 & 2 & 2 \\
2 & 0 & 0 & 2 \\
2 & 0 & 0 & 1
\end{array} \]

3) FULLY CONNECTED LAYER

The fully connected layer connects all the elements in front of the layer and converts the two-dimensional data output of the convolution layer into a one-dimensional vector, and is expressed as,

\[ y_f = w_f \ast x_f + b_f \]

where \( x_f \) is the input data of the full connection layer, \( w_f \) is the weight, and \( b_f \) is the output data of the full connection layer. The full connection layer can integrate the local information of the convolution layer or pooling layer with the distinction of categories. In order to improve the performance of the convolutional neural network, \( \text{ReLU} \) is used as the excitation function for each neuron in the full connection layer.

4) DROPPOUT LAYER

In order to improve the over-fitting phenomenon caused by the training of relatively small data sets by a complex feedforward network, this study introduced dropout layer [24]. The formula of the Dropout layer is

\[ r_f \sim \text{Bernoulli}(p), \]

\[ \tilde{y}_f = r_f \ast y_f, \]

\[ z_i^{(l+1)} = w_f^{(l)} \tilde{y}_f + b_f^{(l)}, \]

\[ y_i^{(l+1)} = f(z_i^{(l+1)}). \]

In the above equations, the Bernoulli function is to generate


a probability \( r \) vector, i.e. randomly generate a \( 0 \) or \( 1 \) vector. \( y^j \) is the input data of this layer. \( \hat{y}^j \) is to filter out the input data of some neurons.

Through the dropout layer, the sparse matrix is established by making a certain neuron stop working with probability \( p \). The updating of weights is no longer dependent on the joint action of hidden nodes with fixed relationships, which prevents some features from being effective only under other specific features, and forces the network to learn more robust features.

**B. CNN TRAINING**

1) **CNN-FORCE MODEL**

The cylinder is subjected to drag and lift force in the flow field, drag coefficient \( C_d \) and lift coefficient \( C_l \) are important indicators of the force of the cylinder. Conventional calculation of the drag and lift coefficients are calculated by\[21,\]

\[
C_l = \frac{1}{2} \rho U^2 D \int_{\tau} (\sigma \cdot n) \cdot n_1 d\tau
\]

\[
C_d = \frac{1}{2} \rho U^2 D \int_{\tau} (\sigma \cdot n) \cdot n_2 d\tau
\]

It should be noticed that due to the fluctuation the lift coefficient and drag coefficient mentioned below in this paper, are root mean square values. The CNN-FORCE model takes the boundary information of the flow around the circular cylinder as the input data and maps this data to the drag or lift coefficient by CNN, as shown in Fig.6. The internal parameters of the CNN-FORCE model are displayed in Table 1. After continuous training with a large number of datasets ranging from 10 to 150 Reynolds Numbers, the model of drag and lift coefficient exerted on the cylinder for predicting different positions and Reynolds Numbers is finally obtained.

**TABLE 1** Internal parameters of CNN-FORCE model.

| Layers        | Filters number | Kernel size/Pooling size | Input size       | Output size       |
|---------------|----------------|--------------------------|------------------|-------------------|
| Conv2d_1      | 4              | 3x3                      | 44x223x2         | 44x223x4         |
| Conv2d_2      | 4              | 4                        | 44x223x4         | 44x223x4         |
| Max_pooling2d_1 | 2x2            | 2x2                      | 44x223x4         | 22x11x8          |
| Conv2d_3      | 8              | 3x3                      | 22x11x8          | 22x11x8          |
| Conv2d_4      | 8              | 3x3                      | 22x11x8          | 22x11x8          |
| Max_pooling2d_2 | 2x2            | 2x2                      | 11x5x5x8         | 11x5x5x16        |
| Conv2d_5      | 16             | 3x3                      | 11x5x5x16        | 11x5x5x16        |
| Conv2d_6      | 16             | 3x3                      | 11x5x5x16        | 11x5x5x16        |
| Max_pooling2d_3 | 2x2            | 2x2                      | 11x5x5x16        | 5x27x16          |
| Conv2d_7      | 64             | 3x3                      | 5x27x16          | 5x27x64          |
| Conv2d_8      | 64             | 3x3                      | 5x27x64          | 5x27x64          |
| Flatten_1     |                | 512                      |                  |                   |
| Dropout_1     |                | 1                        |                  |                   |

2) **CNN-VELOCITY MODEL**

Compared with the CNN-FORCE model, we need a more complex neural network structure to capture the potential features in the velocity field data to establish the mapping relationship between the boundary information of the flow around the circular cylinder and the velocity field. In this paper, the deconvolution neural network is imported to predict the velocity field information. The network structure of the CNN-VELOCITY model is shown in Fig. 7 and the internal parameters are displayed in Table 2.

The data set with Reynolds number information and boundary information is the input end of the neural network. After a large number of samples training and determining the network layer parameters, we obtain the velocity field model of flow around a circular cylinder at various positions and Reynolds numbers.
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Y. Zhao et al.: Flow and Force Prediction Exerted on Circular Cylinder Based on Convolution Neural Network

FIGURE 7. Network structure of CNN-VELOCITY model.

TABLE 2 Internal parameters of CNN-VELOCITY model.

| Layers          | Filters number | Kernel size/Pooling size | Input size  | Output size  |
|-----------------|----------------|--------------------------|-------------|--------------|
| Conv2d_1        | 32             | 3x3                      | 44x223x2    | 44x223x32    |
| Conv2d_2        | 32             | 3x3                      | 44x223x2    | 44x223x32    |
| Max_pooling2d_1 | 64             | 2x2                      | 22x111x32   | 22x111x64    |
| Conv2d_3        | 64             | 3x3                      | 22x111x64   | 22x111x64    |
| Conv2d_4        | 128            | 3x3                      | 11x55x64    | 11x55x128    |
| Conv2d_5        | 128            | 3x3                      | 11x55x128   | 11x55x128    |
| Max_pooling2d_2 | 256            | 2x2                      | 5x27x128    | 5x27x256     |
| Conv2d_7        | 256            | 3x3                      | 5x27x256    | 5x27x256     |
| Conv2d_8        | 256            | 2x2                      | 5x27x256    | 2x13x512     |
| Conv2d_9        | 512            | 3x3                      | 5x27x256    | 2x13x512     |
| Conv2d_10       | 512            | 3x3                      | 2x13x512    | 2 tensors    |
| Conv2d_transpose_1 | 256        | 2x2                      | 2 tensors   |              |
| Zero_padding2d_1 |                |                          |             |              |
| Conv2d_11       | 256            | 3x3                      | 5x27x512    | 5x27x256     |
| Conv2d_12       | 256            | 3x3                      | 5x27x256    | 2 tensors    |
| Conv2d_transpose_2 | 128           | 2x2                      | 2 tensors   |              |
| Zero_padding2d_2 |                |                          |             |              |
| Conv2d_13       | 128            | 3x3                      | 11x55x256   | 11x55x128    |
| Conv2d_14       | 128            | 3x3                      | 11x55x128   | 2 tensors    |
| Conv2d_transpose_3 | 64             | 2x2                      | 2 tensors   |              |
| Zero_padding2d_3 |                |                          |             |              |
| Conv2d_15       | 64             | 3x3                      | 22x111x128  | 22x111x64    |
| Conv2d_16       | 64             | 3x3                      | 22x111x64   | 2 tensors    |
| Conv2d_transpose_4 | 32             | 2x2                      | 2 tensors   |              |
| Zero_padding2d_4 |                |                          |             |              |
| Conv2d_17       | 32             | 3x3                      | 44x223x64   | 44x223x32    |
| Conv2d_18       | 32             | 3x3                      | 44x223x32   | 44x223x32    |
| Conv2d_19       | 3              | 1x1                      | 44x223x32   | 44x223x32    |

FIGURE 8. Loss curves of CNN-FORCE and CNN-VELOCITY during model training.
The goal of training a convolutional neural network is to minimize the loss function. MSE is adopted as the loss function in both CNN-FORCE and CNN-VELOCITY models, and expressed as,

\[
MSE(y, y') = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i')^2
\]

where \( y_i \) is the true value of the \( i \)th data in a batch, and \( y_i' \) is the predicted value of the neural network. Error back propagation algorithm and Adam optimization algorithm are used to train the neural network. The training parameters of other networks are listed in Table 3.

| Model       | Learning rate | Batch size | Epoch |
|-------------|---------------|------------|-------|
| CNN-VELOCITY| 1e-4          | 32         | 300   |
| CNN-FORCE   | 1e-4          | 32         | 300   |

This study import a deep learning library-Keras into Tensorflow, which is a widely used machine learning platform to train CNN-FORCE and CNN-VELOCITY. Fig. 8 shows the loss curve during the training process. Due to the complex network structure of the CNN-VELOCITY model with a large number of network nodes and a large amount of data, the decrease speed of the loss curve is relatively slow. With the increase of epoch, loss gradually became stable and the training ended.

IV. RESULTS and DISCUSSION

A. CNN-FORCE RESULTS

A data set including 15 groups, \( Re=10, 20, 30, \ldots, 150 \), each group with 500 randomly distributed positions, is used to train the neural network and predict the flow around circular cylinder cases with \( Re=15, 45, 75, 105, 135 \) at random position.

Fig.9 shows the mean MSE curves of the drag coefficient and lift coefficient. The averaged MSE value keeps within 0.05, indicating high accuracy. Through the training of the neural network, the force coefficients exerted on the cylinder with different Reynolds Numbers can be predicted accurately.

![FIGURE 9. The mean MSE of the CNN-FORCE model test](image)

![FIGURE 10. Root mean square of drag coefficient (left column) and lift coefficient (right column) prediction of CNN-FORCE model and the corresponding results from LBM from top to bottom at Re=15,45,75,105,135, each scatter represents a cylinder's position.](image)

Fig.10 is the scatter diagram and correlation coefficient of drag coefficient and lift coefficient at various positions at \( Re=15, 45, 75, 105, 135 \) predicted by the CNN-FORCE model. With Reynolds number increasing, the difference
between the predicted value and true value increases slightly. However, most values of correlation coefficient R remain above 0.9, indicating a high accuracy in our research. The prediction results show that the CNN-FORCE model can capture the internal relationship between the force coefficient exerted on the cylinder and the input information. Due to the random distribution of the size and position of the cylinder, the position of the cylinder makes the drag coefficient and lift coefficient fluctuate greatly at the same Reynolds number. With the increase of Reynolds number, the fluctuation amplitude of drag coefficient and lift coefficient decreases gradually, which is in accordance with other numerical results [25].
FIGURE 11. Velocity magnitude prediction by CNN-VELOCITY model from top to bottom, left to right at Re=15, 45, 75, 105, 135. Left column is CNN model predictive results and the right column is the CFD results. The velocity magnitude is the distribution at t=30 seconds.
FIGURE 12. Absolute error distribution of CNN-VELOCITY model prediction at Re= 15, 45, 75, 105, 135.

B. CNN-VELOCITY MODEL PREDICTION RESULTS
The velocity magnitude at Re=15, 45, 75, 105, 135 are predicted by the CNN-VELOCITY model, the results are shown in Fig. 11. In order to show the prediction accuracy of the flow field near the cylinder, the flow field is limited to a given range. The predicted value is in good agreement with the corresponding CFD result as shown in Fig. 12. The CNN-VELOCITY model accurately predicts the velocity field around the cylinder in the area where velocity varies violently.

Furthermore, according to the specific position of the cylinder, we choose a specific longitudinal section, defined by \( \tau = h_l / h \), where \( h_l \) is the height of the longitudinal section from the upper boundary to compare the predicted value with the actual value of the velocity amplitude. Fig. 13 shows the predicted one fits their CFD counterpart very well.
FIGURE 13. Figure (a)-(e) are the comparisons between the predicted value and the real value at the given section, defined by $\tau = h_t / h$, equals 0.75, 0.82, 0.84, 0.73 and 0.68 with Re=15,45,75,105,135. Figure (f) represents the average relative error in figure (a)-(e).
C. COMPARISON OF COMPUTING RESOURCES

Compared with the CFD method, we found that the data-driven models have great advantages in computational efficiency. Once the model is trained, the position of the cylinder and the Reynolds number can be randomly specified in the forecast stage. The CNN-FORCE and CNN-VELOCITY models can quickly predict the drag coefficients, lift coefficients and velocity field around a circular cylinder at a random location with high accuracy. The time required by OpenLB and CNN model to predict per case under the same computer resources is listed in Table 4. It can be seen that the data-driven model has a distinct advantage over the CFD approach.

| TABLE 4. CPU: INTEL XEON E5-1603 2.8GHZ RAM 8G. |  |
|--------------------------------------------------|--|
| Prediction approach | Time |
| OpenLB | 49s |
| CNN-FORCE & VELOCITY | 3s |

V. CONCLUSION

In this paper, the CFD method combined with data-driven modeling is used to predict the flow and force around a cylinder with an external confined boundary. Firstly, a large number of cases are calculated by OpenLB, and an extensive database is established for deep learning modeling. Due to different data structures, two CNN models are designed to predict the force on a cylinder and the velocity amplitude around a cylinder. Once the models are trained, they are capable of predicting the velocity and the fluid force at other positions of the cylinder at different Reynolds numbers. The results indicate that the deep learning method based on convolution neural network has the advantages of high prediction accuracy and efficiency.

It should be noted that the deep learning model needs a lot of data training. In this paper, 7500 cases are calculated to obtain big data for training. In order to save the calculation time, the selected Reynolds number of the flow around the cylinder is low. But considering the external confined boundary, the richness of the flow field is increased to a certain extent.

REFERENCES

[1] Stern, F., et al. "Computational ship hydrodynamics: nowadays and way forward." International Shipbuilding Progress 60(2013):3-105.
[2] Chen, S., and Doolen, G. D., "Lattice Boltzmann method for fluid flows," Annual Review of Fluid Mechanics, vol. 30, no.1, pp. 329-364, 1998, DOI: 10.1146/annurev.fluid.30.1.329.
[3] Williamson, C. H. K., "Vortex dynamics in the cylinder wake," Annual Review of Fluid Mechanics, vol. 28, no.1, pp. 477-539, 1996, DOI: 10.1146/annurev.fluid.28.010196.002401.
[4] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," Nature, vol. 521, no.7553, pp. 436-444, May 2015, DOI: 10.1038/nature14539.
[5] Dethkordi, E. K., Goodarzi, M., and Nourbakhsh, S. H., "Optimal active control of laminar flow over a circular cylinder using Taguchi and ANN," European Journal of Mechanics - B/Fluids, vol. 67, pp. 104-115, Aug. 2018, DOI: 10.1016/j.euromechflu.2017.08.005.
[6] Ling, J., and Templeton, J., "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty," Physics of Fluids, vol. 27, no. 8, pp. 042032-94, Jul.2015, DOI: 10.1063/1.4927765.
[7] Cai, S., Zhou, S., Xu, C., and Gao, Q., "Dense motion estimation of particle images via a convolutional neural network," Experiments in Fluids, vol. 60, no. 4, pp. 73, Mar.2019, DOI: 10.1007/s00348-019-2717-2.
[8] Aminossadati, S. M., Ghasemi, B., and Kargar, A., "Computational analysis of magnetohydrodynamic natural convection in a square cavity with a thin fin," European Journal of Mechanics - B/Fluids, vol. 46, pp. 154-163, Jul. 2014, DOI: 10.1016/j.euromechfluf.2014.03.002
[9] Ling, J., Kurzawa, K., and Templeton, J., "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," Journal of Fluid Mechanics, vol. 807, pp. 155-166, Nov. 2016, DOI: 10.1017/jfm.2016.615.
[10] Ling, J., Jones, R., and Templeton, J., "Machine learning strategies for systems with invariance properties," Journal of Computational Physics, vol. 318, pp. 22-35, Aug. 2016, DOI: 10.1016/j.jcp.2016.05.003.
[11] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "ImageNet Classification with Deep Convolutional Neural Networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, May. 2017, DOI: 10.1145/3065386.
[12] Zeiler, M. D., and Fergus, R., "Visualizing and Understanding Convolutional Networks," In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Computer Vision - Eccv 2014, Pt I, Vol. 8689, pp. 818-833, 2014.
[13] Szegedy, C., Liu, W., Jia, Y. Q., Sermanet, P., Reed, S., Anguelov, D., et al., "Going Deeper with Convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), New York, Jun. 1-9, 2015.
[14] N. Liu, L. Wan, Y. Zhang, T. Zhou, H. Huo and T. Fang, "Exploiting Convolutional Neural Networks With Deeply Local Description for Remote Sensing Image Classification," in IEEE Access, vol. 6, pp. 11215-11228, 2018, DOI: 10.1109/ACCESS.2018.2798799.
[15] S. Won, "Multi-Scale CNN for Fine-Grained Image Recognition in IEEE Access," vol. 8, pp. 116663-116674, 2020, DOI: 10.1109/ACCESS.2020.3005150.
[16] Lecun, Y., "Generalization and Network Design Strategies," 1989.
[17] Morimoto, M., Fukami, K., Zhang, K. et al., "Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low dimensionalization," Theor. Comput. Fluid Dyn. 35, 633–658 (2021). https://doi.org/10.1007/s00162-021-00580-0
[18] Fukami, K., Nakamura, T., Fukagata, K., "Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data", Phys. Fluids 32, 091110 (2020)
[19] Murata, T., Fukami, K., Fukagata, K., "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics", J. Fluid Mech. 882, A13.2020.
[20] Jin, X.W., Cheng, P., Chen, W.L. and Li, H. (2018), "Prediction model of velocity field around cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder", Phys. Fluids. 30(4), 16. https://doi.org/10.1063/1.5024595
[21] Murata, T., Fukami, K., & Fukagata, K. (2020). Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. Journal of Fluid Mechanics, 882, A13. doi:10.1017/jfm.2019.822
[22] Bhathagar, S., Afsar, Y., Pan S.U., Duraisamy, K., Kaushik, S., Prediction of aerodynamic flow fields using convolutional neural networks,Computational Mechanics, Vol.64, No.2August 2019 pp. 525–545. https://doi.org/10.1007/s00446-019-01740-0
[23] Graziotin, D., and Abrahamsson, P., "A Web-based modeling tool for the SEMAT Essence theory of software engineering," Journal of Open Research Software, 1, e4, Sep.2013, DOI: 10.5334/jors.ad.
[24] Park, J., Kwon, K., and Choi, H., "Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160," ISME International Journal, vol.12, no. 6, pp. 1200-1205, 1998, DOI: 10.1007/bf02942594.
[25] Catalano, P., Wang, M., Iaccarino, G., and Moin, P. "Numerical simulation of the flow around a circular cylinder at high Reynolds numbers," International Journal of Heat and Fluid Flow, vol. 24, no. 4, pp. 463-469, Aug. 2003, DOI: 10.1016/S0142-727X(03)00061-4.