

Research Article

Visual Explainable Convolutional Neural Network for Aerodynamic Coefficient Prediction

Yanxuan Zhao,¹ Chengwen Zhong,¹ Fang Wang ,² and Yueqing Wang ²

¹School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

²China Aerodynamics Research and Development Center, Mianyang 621050, China

Correspondence should be addressed to Yueqing Wang; yqwang2013@163.com

Received 30 April 2022; Revised 2 October 2022; Accepted 18 October 2022; Published 1 December 2022

Academic Editor: Giacomo V. Iungo

Copyright © 2022 Yanxuan Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, aerodynamic performance analysis has been widely studied due to its importance in aircraft design. Most works adopted computational fluid dynamics (CFD) simulation to compute the aerodynamic forces, which is time consuming. To reduce the simulation time, several works proposed to use deep learning model as the surrogate model of CFD simulation. However, the explainability of deep learning models is poor and has been widely criticized, which limits the further development of deep learning in aerodynamic performance analysis. In this paper, a novel neural network is proposed to predict the aerodynamic forces of airfoils. To improve the explainability, the circular padding is proposed to replace traditional zero padding in the convolutional layers. Moreover, the saliency map of the predicted aerodynamic force on the input airfoil is shown in a more intuitive way. In this manner, the influence of different parts of airfoil on the final aerodynamic force can be easily analyzed. Extensive experiments on different data sets show that our work is efficient and effective. Most importantly, these results explain the potential relationship between the airfoil and the aerodynamic force.

1. Introduction

Airfoil design and optimization have been extensively studied recently because they are crucial in aircraft design. In this field, the explainability between airfoils and their aerodynamic forces is very important because it can provide the design experience and extract the potential design rules. However, most shape design and optimization methods [1–3] did not consider the explainability. Several methods [4–7] tried to study the relationship from a large number of airfoils and the corresponding aerodynamic forces through statistical methods. Nevertheless, they can only get the general optimization direction and cannot guide researchers on how to adjust a specific airfoil to get better aerodynamic performance. Moreover, these methods usually assumed that there is a linear relationship between the input and output, which leads to some nonlinear relationships that cannot be expressed and further adversely affects the accuracy of optimization design.

With the rapid development of deep learning technology, some researchers used deep learning technology to

improve efficiency [8–16]. These methods usually predict the aerodynamic forces using deep learning models and then analyze the sensitivity between the input airfoils and the aerodynamic forces. By observation, we find some issues still exist with these methods.

First, these approaches adopted the airfoil shape as their input, but their shape representations were not accurate enough. Some methods used images to represent the airfoils [8–10]. In this manner, they translated the aerodynamic force prediction task to a computer vision task, so they can make full use of the convolutional neural network (CNN) achievements in the field of computer vision. However, the image representation of the airfoil will lead to several problems: (i) when the resolution of airfoil images is small, the airfoil shape cannot be accurately described, which will bring difficulties for feature extraction and further decrease the accuracy of aerodynamic performance prediction; (ii) there are lots of blank pixels in the airfoil images, which will lead to a lot of useless calculations. In fact, the image representation is similar to the uniform Cartesian grid, and the coordinate of airfoil is similar to the body-fitted grid. The body-

fitted grid can obtain more accurate results than the Cartesian grid in computational fluid dynamics (CFD) simulation. Therefore, we use coordinate representation for predicting aerodynamic forces instead of image representation. There are several works [16–21] that adopted parameterization methods like class-shape function transformation (CST) [22], orthogonal basis function method [23], and the Hicks-Henne [24] to encode the airfoils. These methods can reduce the input dimension and alleviate the curse of dimensionality in optimization problems, but they cannot distinguish between airfoils with small deformations due to the information loss after dimensionality reduction. Several works took the space coordinates as the inputs. They fixed the X coordinate and only used Y coordinate to represent the airfoils [9, 12]. However, for different airfoils, the parts with sharp gradient changes may be different, so using only Y coordinate and keeping the X coordinate unchanged is difficult to describe these different parts accurately.

Second, the explainability of the existing methods is poor. Most of the related works did not consider explainability at all and only regard the deep learning method as a surrogate model of the CFD simulation. Some works regarded the optimization problem as the searching problem [18–20] and searched the globally optimal airfoil shape from a huge design space, which ignored the explainability between input and output. There is not too much information about the optimization problem itself (only relying on the information of the objective function sampling) to design the algorithm, so the explainability is poor. There are several works that tried to explain the mechanism of the deep learning models in computer vision area [14, 15], but these methods cannot be directly transferred into the CFD field.

Third, most methods did not consider the physical characteristics [8–10]. Some methods directly converted the input coordinates as images, which discarded the topology information of the original data. Other methods used real coordinates to represent the input shape, but they ignored the boundary conditions. In the CFD simulation, the points near the boundary of the shape need to be specially processed, so the intelligent methods also need to consider the boundary conditions carefully.

To address the above issues, we design a convolutional neural network to predict the aerodynamic forces and provide a novel way to optimize the airfoil shape. In our work, we take the XY coordinates of the airfoils as the inputs, which can keep the topology of the airfoils and capture the gradient change of the shape. To keep the boundary conditions and improve the explainability of the model, we use the circular padding to replace the zero padding in convolutional layers. Finally, we design the input-output explainable module based on class activation mapping (CAM), which explains the influence between the airfoil shape and the aerodynamic forces. In this manner, we can present the significance of aerodynamic influence on airfoil coordinates more intuitively. Moreover, the explainable module can help us to optimize the airfoil at the point level, which means we can change the points of an airfoil and observe how this change influences the aerodynamic performance of the airfoil immediately.

Experimental results show that this work can predict aerodynamic forces efficiently, visualize the saliency more directly, and optimize the shape at the coordinate level with different inflow conditions.

The contributions of this paper are summarized as follows:

- (i) A convolutional neural network is designed, which can be easily visualized and obtain aerodynamic forces efficiently and accurately
- (ii) A more reasonable explanation for why circular padding mode is better than zero padding mode by using kinetic analysis is given
- (iii) The input-output explainability module based on class activation mapping is designed, which can explain the local mapping relationships more directly from only integral quantities and may help researchers to further optimize the airfoil shape

2. Related Work

Deep learning technology has been widely applied to solve CFD problems due to its powerful learning ability, through solving the Navier-Stokes (N-S) equations [25], predicting flow field [26], inverse designing [27], identifying flow field [28], modifying turbulence models [29], improving flow field resolution [30], generating mesh [31], shape optimization [32], and so on [33–35]. In this section, we will introduce the related works that applied deep learning in the prediction of aerodynamic performance and aircraft design optimization. These related works can be roughly grouped into three parts.

The first part is the prediction of aerodynamic parameters by deep learning methods. Zhang et al. directly used convolutional neural network (CNN) to predict the aerodynamic parameters of airfoil and output the aerodynamic parameters by taking the image of airfoil as input [8]. Sekar et al. proposed a convolutional neural network method to encode the airfoil image. Then, the encoded airfoil, working condition, and position coordinates are input into the multi-layer perceptron to predict the flow results at current coordinates [26]. However, as mentioned before, when the resolution of airfoil images is too small, the airfoil shape cannot be accurately described, as shown in Figure 1. Wu et al. adopted the generative adversarial networks (GANs) to predict the flow field [17, 36]. Some works tried to add the boundary conditions into their models. Thuerey et al. proposed the method of using U-net to predict the flow field. The working conditions, boundary conditions, and airfoil are regarded as different channels of the input image [37]. To adapt different kinds of grids, Kashefi et al. proposed a deep learning framework based on point cloud to predict the flow field [13].

The second part is the inverse design. When researchers have a clear requirement on the aerodynamic force distribution, they can design models to deduce the shape inversely by specifying the aerodynamic force distribution. Sekar et al. considered images of the pressure distribution on

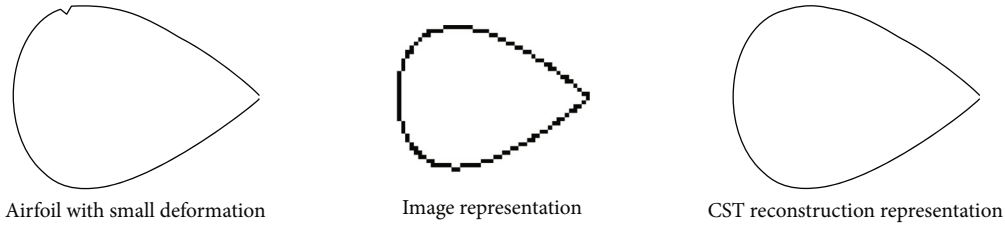


FIGURE 1: Airfoil with small deformation, its image representation, and CST reconstruction representation.

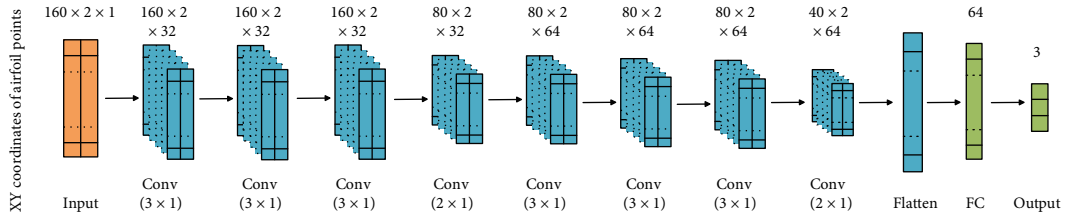


FIGURE 2: Basic skeleton of convolutional neural network.

airfoil as the inputs of their proposed CNN models and predicted the coordinates of the airfoils. Wang et al. used the conditional autoencoder as the generator of the GANs and took the pressure and surface Mach number distributions on airfoil as the input to generate the corresponding airfoil shape [12].

The last part is the intelligent optimization methods. These works include traditional intelligent optimization algorithms and data-driven optimization algorithms. In the traditional algorithms, the best shape usually is searched from a big shape space whose aerodynamic forces meet the requirement. To reduce the search space, some dimension reduction methods are often adopted to reduce the input size [16]. The multiobjective optimization method for high-lift airfoils is proposed [18]. On account of this idea, a surrogate model based on radial basis function is used to determine the location of new online test data [19, 20]. Tao and Sun proposed to use a multilayer perceptron (MLP) as surrogate model to carry out the robust optimization of aerodynamic shape [21]. Data-driven optimization algorithms consist of two categories: statistical methods and machine-learning methods. Liem et al. used probability density diagram of aerodynamic parameters to reduce the drag of airfoil [4]. Achour et al. divided the distribution of lift coefficient (Cl) and drag coefficient (Cd) of airfoil into four parts and used conditional autoencoder to predict the airfoil which meets the requirements of aerodynamic parameters [38]. Proper orthogonal decomposition (POD) can reduce data dimension and improve computational efficiency [5–7]. Li et al. proposed a method to express the constraints of data-driven optimization of airfoil. By using Gaussian mixture model for analysis, the constraints of optimization are well expressed [39]. They also parameterized the airfoil by using the geometric filtering method based on deep learning [32]. Jin et al. proposed a data-driven evolutionary algorithm, which used data to initialize the surrogate model and then optimize airfoil online [40]. Li and Zhang take advantage of the strong nonlinear fitting ability of neural network to

TABLE 1: The architectures of the proposed neural network.

CNN
Input 160×2
3×1 conv. 32 swish
3×1 conv. 32 swish
3×1 conv. 32 swish
2×1 conv. 32 stride (2)
3×1 conv. 64 swish
3×1 conv. 64 swish
3×1 conv. 64 swish
2×1 conv. 32 stride (2)
Full connected 64 swish
Linear 3
Airfoil output 3 (Cl, Cd, Cl/Cd)

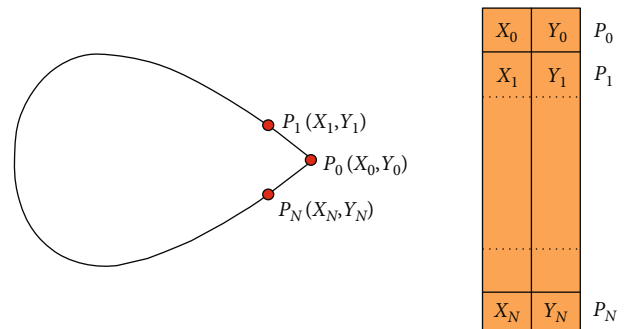


FIGURE 3: The airfoil curve and the input coordinate vector.

initialize surrogate model [41]. Wen et al. proposed a new initialization method, which used genetic algorithm to initialize the neural network as surrogate model and optimize the turbine airfoil [42].

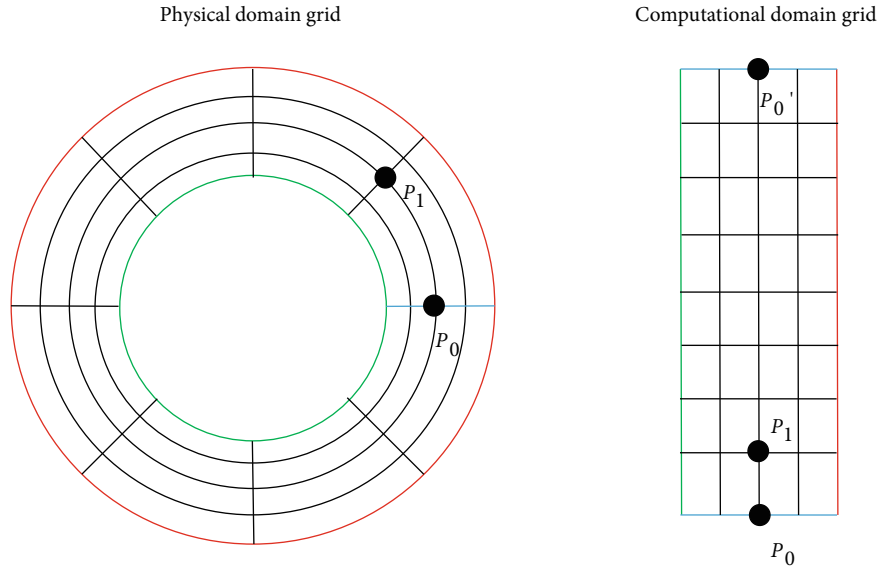


FIGURE 4: The physical grid (a) and computational grid (b). The corresponding boundary edges of two domains are indicated by the same color.

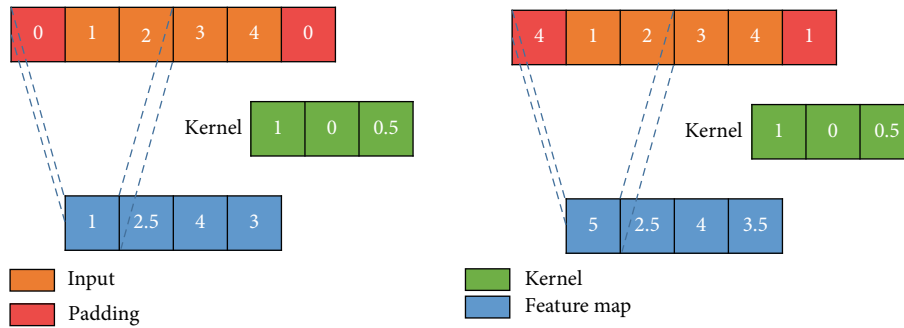


FIGURE 5: (a) Zero padding and its receptive field. (b) Circular padding and their convolutional results.

Overall, deep learning has been widely applied in aerodynamic parameter prediction and shape optimization. Although these methods can achieve good results, they still have some problems, for example, the shape representation, the explainability, and the physical characteristics.

3. Our Approach

3.1. Network Structure. In our work, we design a convolutional neural network to replace the equation solver in the CFD method and to predict the aerodynamic coefficients directly. The basic network structure is depicted in Figure 2, and the detail of each layer is given in Table 1. The input of this neural network is XY coordinates of the airfoil while the output is C_l and C_d . The first dimension of the kernel size of our network is set to 3 according to experience, and the second dimension is set to 1 because we try to maintain the spatial relationship of XY coordinates. In our network, swish activation function [43] is used, convergence criteria commonly used in CFD, and mean relative error is used as loss function. For explainability, bias will not

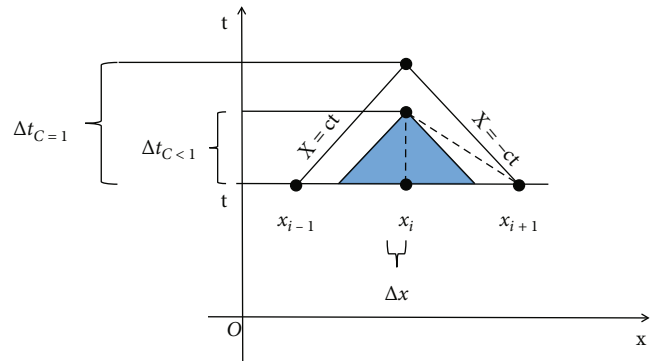


FIGURE 6: When zero padding is use, the point $i - 1$ will be set to 0, so the Courant number is an infinite number. The blue part in the figure is the domain of difference equation. The area enclosed by dotted line represents the domain of zero padding.

be used for the final linear layer. Adam optimizer was selected with a learning rate of $3e - 4$, and tenfold cross-validation was used.

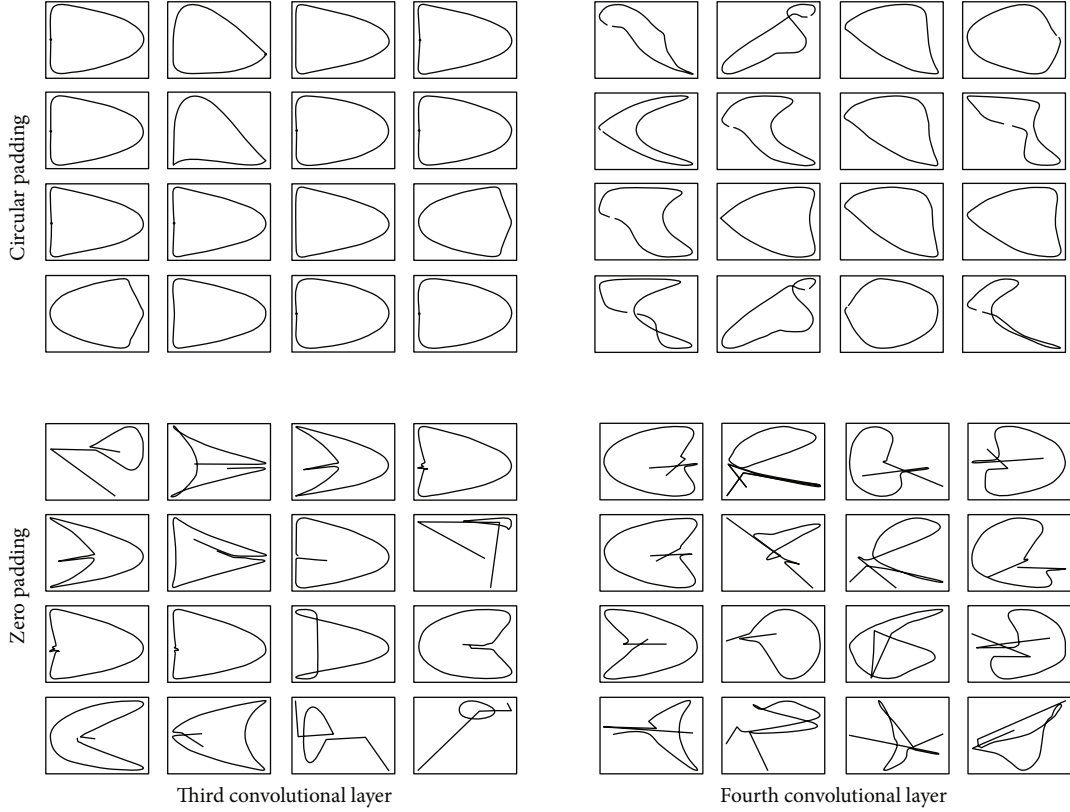


FIGURE 7: Feature maps of several convolutional layers with different padding modes.

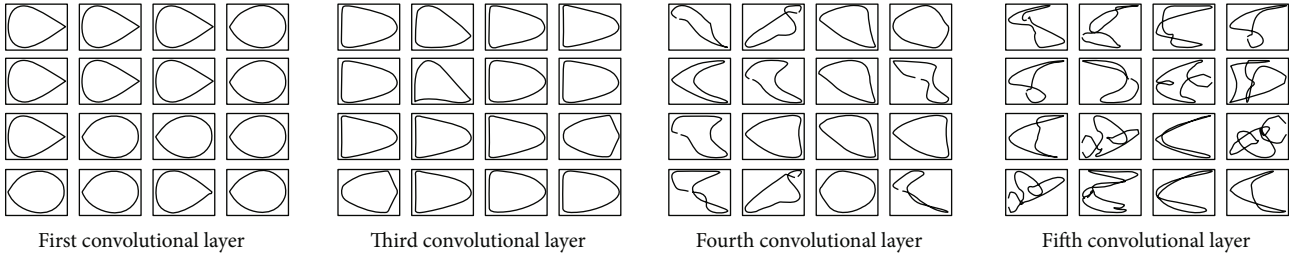


FIGURE 8: Feature maps of different convolutional layers. All these layers use circular padding.

$$\text{Loss} = \left| \frac{\text{prediction} - \text{ground truth}}{\text{ground truth}} \right|. \quad (1)$$

In our network, zero padding is replaced by the circular padding in each convolutional layer. The circular padding will be introduced in the following.

3.2. Circular Padding. The airfoil shape is a closure curve in physical space. To capture the features of the curve, we consider the original coordinates as the inputs. However, this will lead to a problem: two neighbor points in the airfoil curve may not neighbor in the computational domain. For example, in Figure 3, P_0 , P_1 , and P_N are neighbors at the tail of an airfoil, but when we use coordinate vector to represent the airfoil, P_0 becomes the first point of the input vector and P_N becomes the last point of the input vector. In this man-

TABLE 2: Comparison of prediction performance.

Method	Case 1 MRE	Case 2 MRE
CNN (point Z-padding)	1.78%	1.78%
CNN (point C-padding)	1.43%	1.64%
CNN (image 32×32)	2.95%	2.41%
CNN (image 24×24)	2.96%	2.51%
CST-RBF	1.56%	2.81%

ner, neighbors in physical space are no longer neighbors in the input vector. Zero padding is the most common used padding mode. It will pad 0 around the boundary points, which is different from the physical properties of the flow field. To keep the neighbor constraint, we use the circular padding instead of the zero padding in the first convolutional layer.

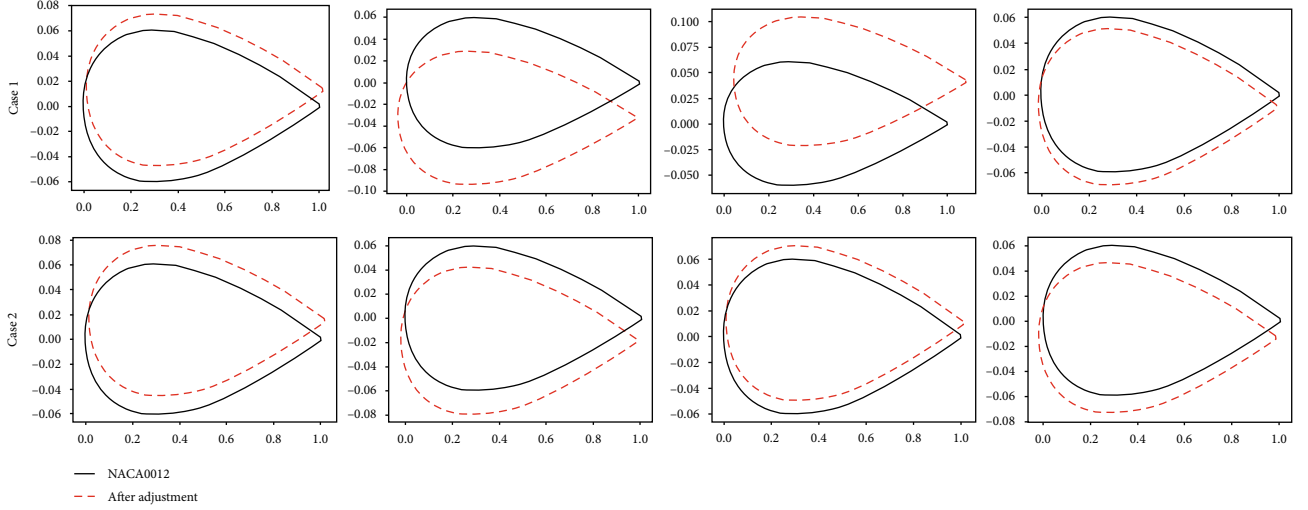


FIGURE 9: NACA 0012 airfoil and its adjustments after enlarging the output of the specific neuron in the first convolutional layer. Different columns show the results in different channels.

In other convolutional layers, we also need to keep this property. The flow field in the physical space is usually described as physical grids with different variables like velocity, pressure, and temperature. The physical grid is usually a nonuniform grid, and there is a physical distance value between two grid points. When researchers try to solve the physical equations, they must transform the nonuniform physical grid into the uniform grid in the computational space, as depicted in Figure 4. In Figure 4, the P_0 and P_1 are neighbors in the physical grid but not neighbors in the computational grid. In the CFD simulation, to keep the neighbor relationship, they copy the boundary points to the bottom of the computational grid. In our network, we can use circular padding to solve the problem.

The zero padding, circular padding, and their corresponding receptive fields are shown in Figure 5. The circular padding uses the left points to pad the right side and the right points to pad the left side. As a contrary, the zero padding only pad zeros at both sides. It can be seen that the circular padding mode can keep the neighbor relationship in the physical space.

Moreover, we find the circular padding contains a more important physical restriction. The forward propagation of neural network is similar to explicit scheme to solve differential equation [44]. Weinan proposed that each layer passed forward by the neural network can be regarded as a push forward of the numerical discretization of a dynamic system in time, and the training neural network can be regarded as finding the corresponding parameter to solve this control problem [45]. Lu et al. regard the forward process of neural network as the explicit scheme time advances of numerical solution of differential equations and used the relevant theories of numerical analysis to design the structure of neural network [46]. Zhang and Schaeffer analyzed the stability of neural network by using the relevant theory of numerical analysis [47]. Based on the above point of view, we use the relevant theories of numerical analysis to adjust the structure

of neural network. In the CFD simulation, to make sure the convergence of explicit difference scheme of hyperbolic equation, we must guarantee that the Courant number is a finite number. The Courant number is used to evaluate the time-step requirements of a transient simulation for a given grid size and flow velocity. It indicates how much the information travels across a computational grid cell in a unit of time. The CFL number (Courant number) can be computed using

$$\text{CFL} = |c| \frac{\Delta t}{\Delta x}, \quad (2)$$

where c indicates the flow velocity, Δt is a representative time step of the simulation, and Δx is the characteristic size of the grid. In a CFD simulation, we usually use the Courant number to get the stability and convergency. The stability requirements of time-integration schemes can be defined using the Courant–Friedrichs–Lewy (CFL) condition [48], and it is expressed in terms of the Courant number as

$$\text{CFL} = |c| \frac{\Delta t}{\Delta x} < C_{\max}, \quad (3)$$

where C_{\max} varies depending on the type of time integration scheme, but it is generally less than or equal to 1.0.

To guarantee the Courant number is less than C_{\max} , Δt and Δx need to be carefully selected in the CFD simulation. The proposed network is a surrogate model of the CFD simulation, so we also need to constrain these properties in our network. In the proposed network, the left and right side points can be regarded as the boundary points in the CFD simulation. When the zero padding is adopted in the network, the input will pad 0 on its boundary side. It is equivalent to set the x_{i-1} to be zero in the CFD simulation, which is shown in Figure 6. When x_{i-1} is equal to 0, the $\Delta x = 0$; then,

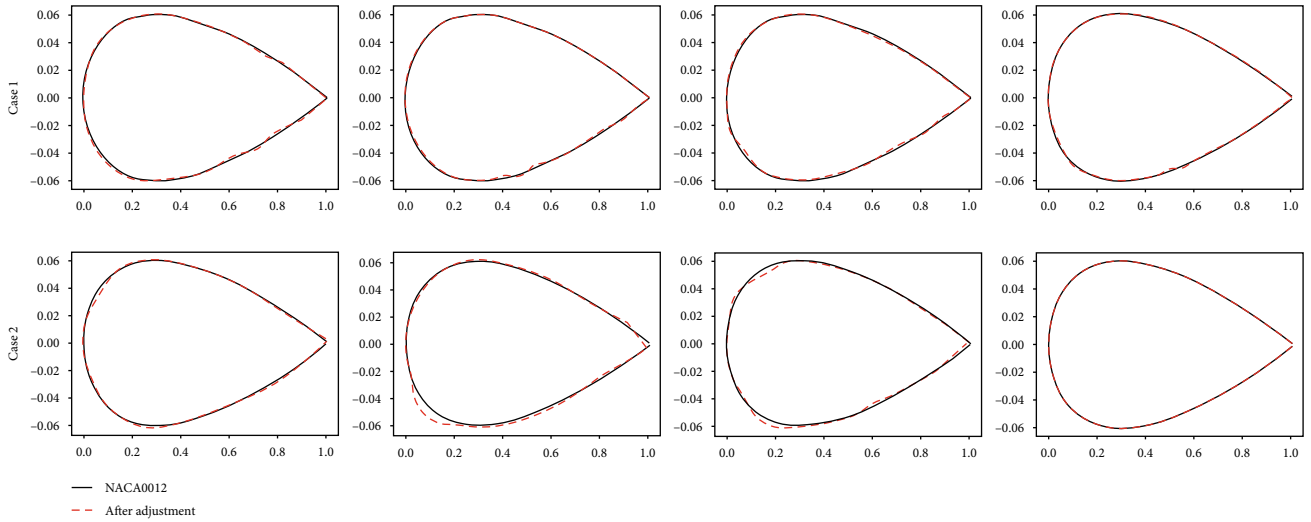


FIGURE 10: NACA 0012 airfoil and its adjustments that enlarges the output of the specific neuron in the fully connected layer. Different columns show the results of different neurons.

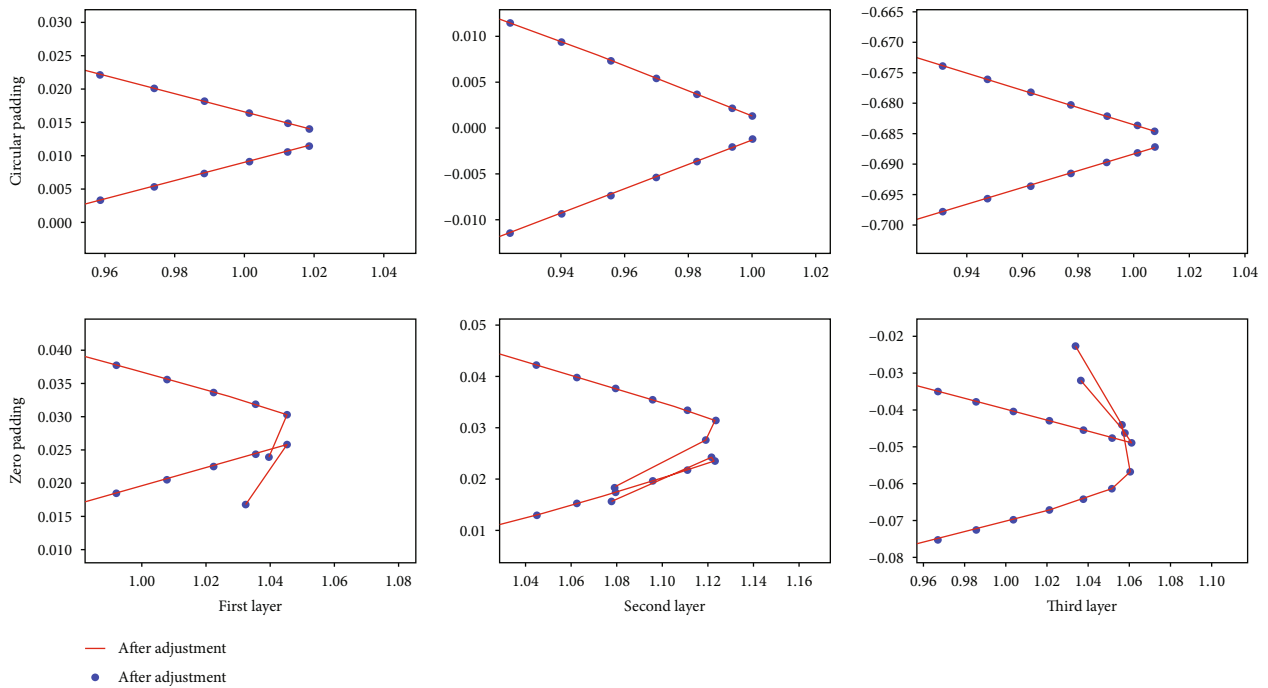


FIGURE 11: Wing tail adjustment under different padding modes.

the area size enclosed by the dotted line cannot contain the blue area, which indicates that the domain of zero padding cannot contain the domain of difference equation. The circular padding can meet the CFL restriction in all convolutional layers.

The airfoil shape curve is continuous and closed. The circular padding can keep these properties, while the zero padding cannot. Some feature maps of convolutional layers using circular padding and zero padding are shown in Figure 7. It can be seen from Figure 7 that the circular padding can make sure the output of each layer is closed and

continuous because the discrete convolution of continuous function is continuous.

From Figure 7, we can conclude that circular padding can make the output of each layer a closed curve and smoother. From Figure 8, we can conclude that low layers extract basic features while high layers can extract more abstract features.

3.3. *Gradient-Weighted Features.* The features extracted by fully-connected layers will lose some spatial information, while the features extracted by convolutional layers contain

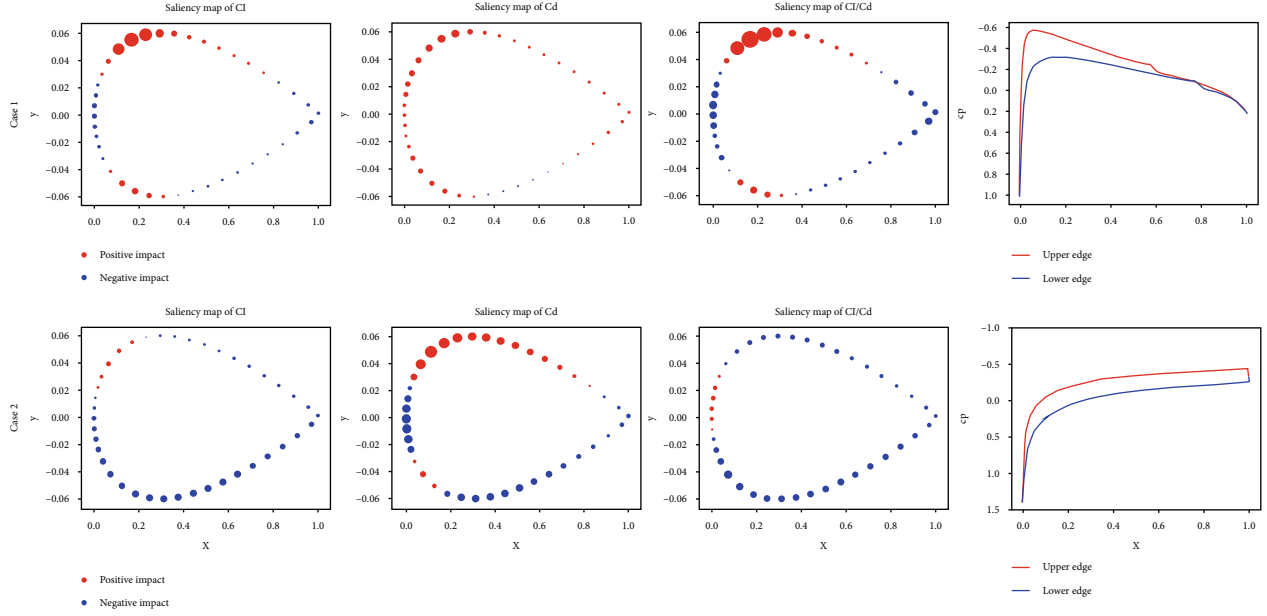


FIGURE 12: The contributions of different airfoil positions to Cl, Cd, and Cl/Cd. Red spots represent the positive effect on them and the blue spots represent the negative effect on them; the larger the spots, the greater the impact is. The first column shows the contributions of different positions to Cl, the second column shows the contributions of different positions to Cd, the third column shows the contributions of different positions to Cl/Cd, and the fourth column shows that the real Cp distribution comes from each data set.

sufficient spatial and semantic information. Features from lower layers to higher layers become more discriminative and abstract. Therefore, we use the features of the last convolutional layer to analyze the relationship between inputs and outputs. The partial derivative of output to feature maps of X coordinate of the last layer can be computed using

$$\frac{\partial O^t}{\partial A_x^j}, \quad (4)$$

$$\frac{\partial O^t}{\partial A_y^j}. \quad (5)$$

In Equations (4) and (5), O is the output of the neural network, t means the index of output, A_x is the feature map of X coordinate of the last pooling layer, A_y is the feature map of Y coordinate of the last pooling layer, and i, j represent the i -th point of the j -th channel. After obtaining the partial derivative, we compute an average value for each feature map after the same convolution kernel using Equation (6) and get α_j^t which represent the importance of different channels of feature map.

$$\alpha_j^t = \frac{1}{2N} \sum_i \left(\frac{\partial O^t}{\partial A_x^j} + \frac{\partial O^t}{\partial A_y^j} \right), \quad (6)$$

$$H^t = \sum_j \alpha_j^t A^j, \quad (7)$$

To study both positive and negative effects of output on input, we sum the product of α_j^t and A^j and get one saliency

TABLE 3: Comparison of prediction performance under a small amount of data.

Method	Case 1 MRE	Case 2 MRE
CNN (point C-padding)	4.65%	4.57%
CNN (image 64×64)	10.0%	5.85%
CNN (image 32×32)	9.03%	5.57%
CNN (image 24×24)	7.75%	6.72%
CST-RBF	5.75%	5.12%

map, as described in Equation (7). In this manner, the relationship between the output and the last convolution layer is obtained. As mentioned above, the last convolution layer keeps the spatial information of the input, so we can get the relationship between the input and output by resizing the H^t to the input size. Especially when H_x^t and H_y^t are needed to be summed when representing the saliency map because the X and Y coordinates together represent one position.

4. Experimental Results

4.1. Experimental Environment and Data Sets. The platform used in this work is an artificial intelligent server, which includes two Intel Xeon CPUs with a main frequency of 2.2GHz and NVIDIA® Tesla® V100 GPU.

The proposed method is evaluated using two cases. The first one is a 2D airfoil data set at low Mach number. It contains 2190 NACA 4-digit airfoils generated from XFOIL; the first digit is from 0 to 9, the second digit is from 0 to 9, and last two digits is from 03 to 27. The working conditions are

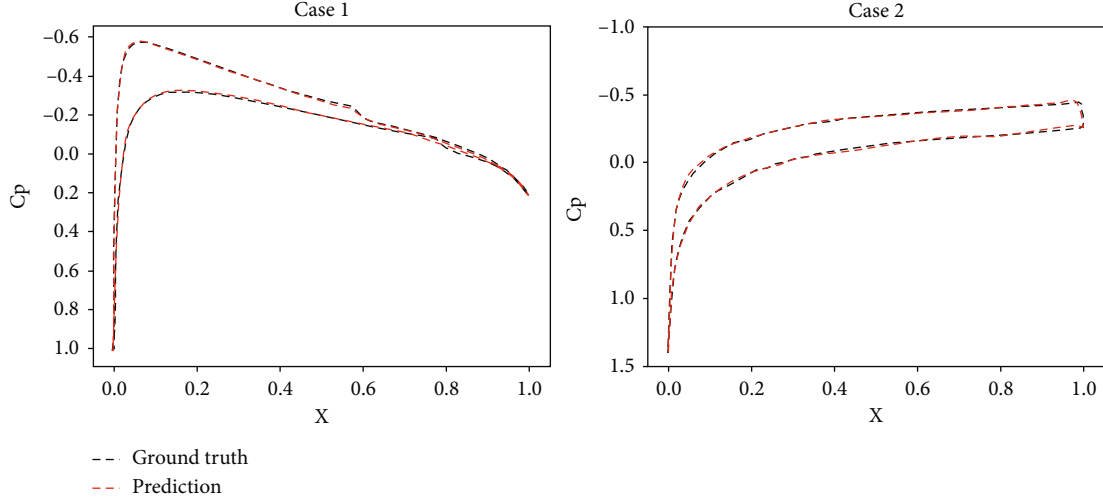


FIGURE 13: Cp prediction on NACA0012 airfoil.

$Re = 1E6$, $Ma = 0.2$, and angle of attack (AOA) = 1° . The aerodynamic forces are computed by XFOIL. 2190 airfoils are randomly divided into two parts: 1978 airfoils for training and 212 for testing. The second case is a high Mach number 2D airfoil data set. It contains 1504 airfoils downloaded from the UIUC airfoil data site. The working conditions are viscosity = $1.983E-5$ kg/ms, $Ma = 1.2$, and AOA = 3° . The results are calculated by Fluent with SST k - ω model; some classical airfoils are used to validate their results for the CFD simulation in Fluent such as NACA0012 and Rae2822. The results show that our CFD simulation is acceptable within the error range limit of error. These airfoils are randomly divided into two parts: 1354 airfoils for training and 150 airfoils for testing.

All airfoils are represented by 160 2D-coordinate points. To show the explainable visualization, the labels consist only of Cl, Cd, Cl/Cd, excluding the flow field.

4.2. Prediction Precision of Aerodynamic Forces. In this part, we evaluate the precision of the proposed method and check the effect of the circular padding. In addition, we compare the precision of the proposed network with CNN using images. Experiments on a small number of samples will be given in Appendix A.

The result of CST-RBF is the best model in CST parameters from 4 to 20 and K -means clustering center of RBF from 1 to number of train data.

It can be concluded from Table 2 that on both two cases, (i) networks using the circular padding can obtain smaller mean relative error (MRE) on testing set than networks using zero padding. (ii) Compared with CST-RBF, CNN (Point C-padding) is more accurate. (iii) CNN using images as input is inferior in accuracy than CNN using coordinates as input.

4.3. Neuron Visualization. We change the output of neurons slightly; then, get the input airfoil corresponding to the output reversely and observe the difference of the airfoil before and after the change, so as to judge the influence of the neu-

ron output change on the input. In specific, we compute the derivative of specific neuron output to original input (I_{old}), and then, we enlarge the output and use Equation (8) to obtain the corresponding new airfoil. In Equation (8), I is input, α means the step size, and S is output of the specific neuron.

$$I_{new} = I_{old} + \alpha \frac{\partial S}{\partial I_{old}}. \quad (8)$$

As can be seen from Figures 9 and 10, it is similar to neural network for image; low-level features are easy to understand while high-level features are abstract.

When zero padding is used, the adjustments will be discontinuous and it can be seen that the discontinuous pass with forward layer through the receptive field in Figure 11.

4.4. Saliency Map. Saliency map is a map which indicates the contribution of different components to the prediction results in other words, which components are responsible for the result. In our case, the saliency map is not equal to the aerodynamic force at each point but a contribution of each point to aerodynamic force in data distribution. In order to show the relationship between input and output more intuitively, we first calculate the contribution degree of output to the last convolution layer through Equation (6) and get the contribution graph. After that, we resize the contribution graph to the input size and directly display it on the airfoil, as shown in Figure 12. The first three columns in Figure 12 show the contributions of different airfoil positions to Cl, Cd, and Cl/Cd, and the last column gives the true pressure coefficient (Cp) distribution on airfoil obtained by XFOIL or Fluent. The experiments on the saliency map of Cp will be given in Appendix B.

From these saliency maps, we can directly observe that which positions of the airfoil contribute more to the lift coefficient, drag coefficient, and lift-drag ratio. In case 1, Cp distribution shows that the lift mainly came from the front half of the airfoil, which shows that the saliency map is

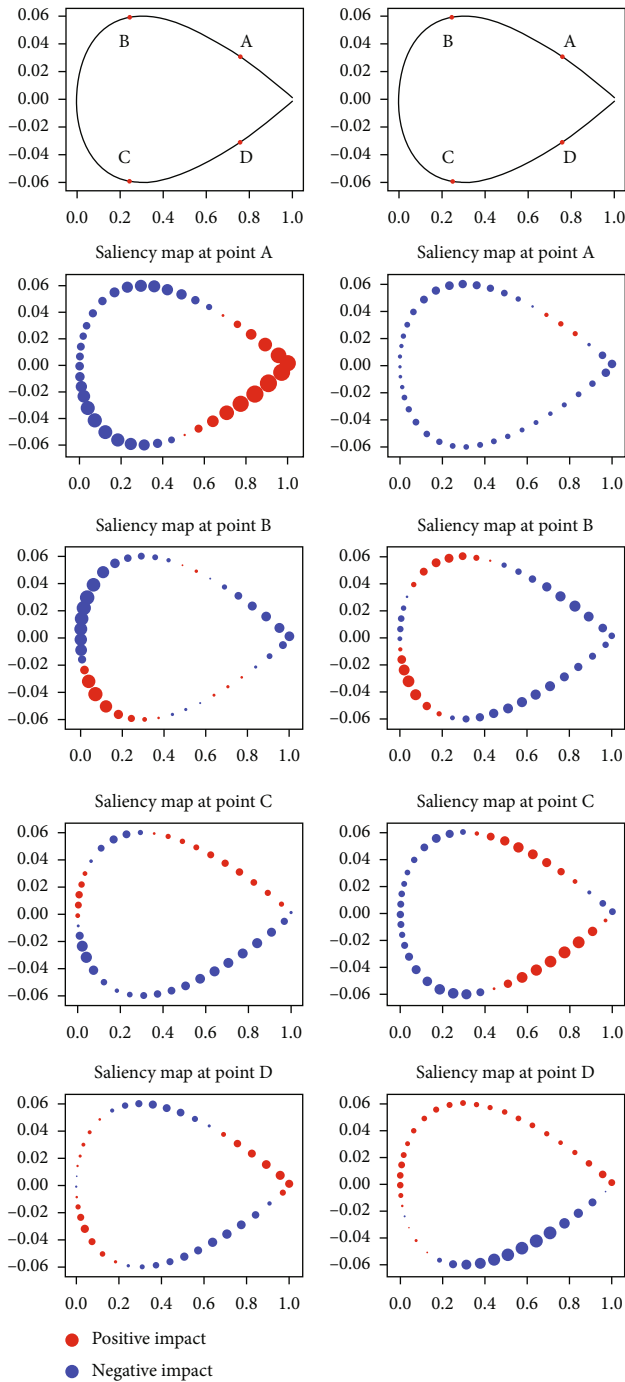


FIGURE 14: Saliency map at different points.

reasonable, while in case 2, the lift came from the whole airfoil relatively homogeneous, and saliency map of C_l has little change on x -axis. The integral quantities like C_l , C_d , and C_l/C_d can be expressed on the shape.

5. Conclusion

In this paper, we propose a novel neural network to predict the aerodynamic forces of airfoils, which uses the circular padding instead of traditional zero padding in convolutional

layers. Moreover, we show the saliency map of the predicted aerodynamic force on the input airfoils, so the influence of different airfoil parts on the aerodynamic force can be described in a more intuitive way. By using saliency map, this work intuitively shows that which position of the airfoil will have what kind of impact on the lift and drag from only integral quantity. Extensive experiments on different data sets show that our network is efficient and effective. Most importantly, our method may help researchers to further optimize the airfoil shapes. There are some limitations in our work. Firstly, we do not research on the aerodynamic forces of three-dimensional airfoils and only focus on the 2D airfoils, which are the cross-sections of real airfoils; secondly, in this work, we train different neural networks for different data sets; it is worth to study how to obtain an integrated model for different data sets. In the future, we will focus on the prediction of aerodynamic forces of three-dimensional airfoils and the integrated neural network.

Appendix

A. Experiments with a small number of samples

Performance of neural network under a small amount of data.

In cases 1 and 2, 10% of the data is used as training data (222 for case 1 and 150 for case 2), and the rest is used as test data.

In Table 3, result of CST-RBF is the best model in CST parameters from 4 to 20 and K -means clustering center of RBF from 1 to number of train data.

B. Saliency map of C_p

Similar neural network was used to predict C_p on both two cases. The prediction of C_p on NACA0012 in each case is shown in Figure 13.

The Saliency map on point ABCD which represent the front and back quarter chord length is shown in Figure 14.

Data Availability

All data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the National Natural Science Foundation of China (Grant No. 61806205), the National Numerical Wind Tunnel Project, and the 111 Project of China (Grant No. B17037).

References

- [1] X. Wei, X. Wang, and S. Chen, "Research on parameterization and optimization procedure of low-Reynolds-number airfoils based on genetic algorithm and Bezier curve," *Advances in Engineering Software*, vol. 149, article 102864, 2020.
- [2] L. Daróczy, G. Janiga, and D. Thévenin, "Computational fluid dynamics based shape optimization of airfoil geometry for an H-rotor using a genetic algorithm," *Engineering Optimization*, vol. 50, no. 9, pp. 1483–1499, 2018.
- [3] J. H. Jeong and S. H. Kim, "Optimization of thick wind turbine airfoils using a genetic algorithm," *Journal of Mechanical Science and Technology*, vol. 32, no. 7, pp. 3191–3199, 2018.
- [4] R. P. Liem, J. R. R. A. Martins, and G. K. W. Kenway, "Expected drag minimization for aerodynamic design optimization based on aircraft operational data," *Aerospace Science and Technology*, vol. 63, pp. 344–362, 2017.
- [5] D. Cinquegrana and E. Iuliano, "Efficient global optimization of a transonic wing with geometric data reduction," in *35th AIAA Applied Aerodynamics Conference*, Denver, Colorado, 2017.
- [6] Y. QIU, J. Bai, N. Liu, and C. Wang, "Global aerodynamic design optimization based on data dimensionality reduction," *Chinese Journal of Aeronautics*, vol. 31, no. 4, pp. 643–659, 2018.
- [7] J. Li, M. A. Bouhlel, and J. R. R. A. Martins, "Data-based approach for fast airfoil analysis and optimization," *AIAA Journal*, vol. 57, no. 2, pp. 581–596, 2019.
- [8] Y. Zhang, W. J. Sung, and D. N. Mavris, "Application of convolutional neural network to predict airfoil lift coefficient," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, 2018.
- [9] E. Yilmaz and B. German, "A convolutional neural network approach to training predictors for airfoil performance," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, 2017.
- [10] B. Yu, L. Xie, and F. Wang, "An improved deep convolutional neural network to predict airfoil lift coefficient," *Proceedings of the International Conference on Aerospace System Science and Engineering 2019*, pp. 275–286, Springer, Singapore, 2020.
- [11] V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil using a deep convolutional neural network," *AIAA Journal*, vol. 57, no. 3, pp. 993–1003, 2019.
- [12] J. Wang, R. Li, C. He et al., "An inverse design method for supercritical airfoil based on conditional generative models," *Chinese Journal of Aeronautics*, vol. 35, 2021.
- [13] A. Kashefi, D. Rempe, and L. J. Guibas, "A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries," *Physics of Fluids*, vol. 33, no. 2, article 027104, 2021.
- [14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, pp. 818–833, Cham, Germany, 2014.
- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, Italy, 2017.
- [16] V. Sripawadkul, M. Padulo, and M. Guenov, "A comparison of airfoil shape parameterization techniques for early design optimization," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, Texas, 2010.
- [17] H. Wu, X. Liu, W. An, S. Chen, and H. Lyu, "A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils," *Computers & Fluids*, vol. 198, article 104393, 2020.
- [18] H. W. Lim and H. Kim, "Multi-objective airfoil shape optimization using an adaptive hybrid evolutionary algorithm," *Aerospace Science and Technology*, vol. 87, pp. 141–153, 2019.
- [19] M. Kanazaki, K. Tanaka, S. Jeong, and K. Yamamoto, "Multi-objective aerodynamic optimization of elements' setting for high-lift airfoil using kriging model," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2006.
- [20] Y. He, J. Sun, P. Song, X. Wang, and A. S. Usmani, "Preference-driven kriging-based multiobjective optimization method with a novel multipoint infill criterion and application to airfoil shape design," *Aerospace Science and Technology*, vol. 96, article 105555, 2020.
- [21] J. Tao and G. Sun, "Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization," *Aerospace Science and Technology*, vol. 92, pp. 722–737, 2019.
- [22] B. Kulfan and J. Bussolotti, "Fundamental" parameteric geometry representations for aircraft component shapes," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, 2006.
- [23] G. M. Robinson and A. J. Keane, "Concise orthogonal representation of supercritical airfoils," *Journal of Aircraft*, vol. 38, no. 3, pp. 580–583, 2001.
- [24] R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," *Journal of Aircraft*, vol. 15, no. 7, pp. 407–412, 1978.
- [25] Y. Du and T. A. Zaki, "Evolutional deep neural network," 2021, <https://arxiv.org/abs/2103.09959>.
- [26] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo, "Fast flow field prediction over airfoils using deep learning approach," *Physics of Fluids*, vol. 31, no. 5, article 057103, 2019.
- [27] E. Yilmaz and B. German, "A deep learning approach to an airfoil inverse design problem," in *2018 Multidisciplinary Analysis and Optimization Conference*, Atlanta, Georgia, 2018.
- [28] J. Han, J. Tao, and C. Wang, "FlowNet: a deep learning framework for clustering and selection of streamlines and stream surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 4, pp. 1732–1744, 2020.
- [29] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.
- [30] B. Liu, J. Tang, H. Huang, and X. Y. Lu, "Deep learning methods for super-resolution reconstruction of turbulent flows," *Physics of Fluids*, vol. 32, no. 2, article 025105, 2020.
- [31] Z. Zhang, Y. Wang, P. K. Jimack, and H. Wang, "Meshing net: a new mesh generation method based on deep learning," in *International Conference on Computational Science*, pp. 186–198, Cham, Germany, 2020.
- [32] J. Li and M. Zhang, "On deep-learning-based geometric filtering in aerodynamic shape optimization," *Aerospace Science and Technology*, vol. 112, article 106603, 2021.
- [33] K. T. Carlberg, A. Jameson, M. J. Kochenderfer, J. Morton, L. Peng, and F. D. Witherden, "Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning," *Journal of Computational Physics*, vol. 395, pp. 105–124, 2019.

- [34] L. Sun, H. Gao, S. Pan, and J. X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Computer Methods in Applied Mechanics and Engineering*, vol. 361, article 112732, 2020.
- [35] H. Bao, J. Feng, N. Dinh, and H. Zhang, "Deep learning interfacial momentum closures in coarse-mesh CFD two-phase flow simulation using validation data," *International Journal of Multiphase Flow*, vol. 135, article 103489, 2021.
- [36] H. Wu, X. Liu, W. An, and H. Lyu, "A generative deep learning framework for airfoil flow field prediction with sparse data," *Chinese Journal of Aeronautics*, vol. 35, 2022.
- [37] N. Thuerey, K. Weissenow, L. Prantl, and X. Hu, "Deep learning methods for Reynolds-averaged Navier-Stokes simulations of airfoil flows," *AIAA Journal*, vol. 58, no. 1, pp. 25–36, 2020.
- [38] G. Achour, W. J. Sung, O. J. Pinon-Fischer, and D. N. Mavris, "Development of a conditional generative adversarial network for airfoil shape optimization," in *AIAA Scitech 2020 Forum*, Orlando, FL, 2020.
- [39] J. Li, S. He, and J. R. R. A. Martins, "Data-driven constraint approach to ensure low-speed performance in transonic aerodynamic shape optimization," *Aerospace Science and Technology*, vol. 92, pp. 536–550, 2019.
- [40] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2019.
- [41] J. Li and M. Zhang, "Data-based approach for wing shape design optimization," *Aerospace Science and Technology*, vol. 112, article 106639, 2021.
- [42] H. Wen, S. Sang, C. Qiu, X. du, X. Zhu, and Q. Shi, "A new optimization method of wind turbine airfoil performance based on Bessel equation and GABP artificial neural network," *Energy*, vol. 187, article 116106, 2019.
- [43] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, <https://arxiv.org/abs/1710.05941>.
- [44] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [45] E. Weinan, "A proposal on machine learning via dynamical systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, pp. 1–11, 2017.
- [46] Y. Lu, A. Zhong, Q. Li, and B. Dong, "Beyond finite layer neural networks: bridging deep architectures and numerical differential equations," in *International Conference on Machine Learning*, pp. 3276–3285, 2018.
- [47] L. Zhang and H. Schaeffer, "Forward stability of ResNet and its variants," *Journal of Mathematical Imaging and Vision*, vol. 62, no. 3, pp. 328–351, 2020.
- [48] J. D. Anderson and J. Wendt, *Computational Fluid Dynamics*, McGraw-Hill, New York, 1995.