New accurate deep learning method for flow field reconstruction from tiny data

Jian Liu,[1] Zhenwei Huang,[1,*] Jinsong Zhang, [1] and Zanao Hu, [1]

State Key Laboratory of Hydroscience and Engineering, Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China

Deep learning technology can not only save the computing power and time in experimental or computational fluid dynamics but also improve the quality of fluid experimental measurements or numerical simulations. Certain numerical computing errors result in quite inaccurate simulation results. Nevertheless, the deep learning technology will be able to reconstruct the distorted results with high precision, making flow mechanism research more convenient. The novel super-resolution convolutional neural network with the u-shaped architecture (SRUnet) model is developed by combining the advantages of super-resolution convolutional neural network (SRCNN) and convolutional neural network with the u-shaped architecture (U-net) to solve the problem of turbulence flow filed image with low resolution. The SRUnet model will be validated to obtain the high precision results just with tiny datasets when it is applied in reconstructing the velocity flow field of the two-dimensional (2D) cylinder wake flow. Further studies have shown that the performance of training and testing for SRUnet model supplements more under-resolved details than the SRCNN and U-net. In comparison with the quality of the velocity flow field reconstruction of SRCNN and U-net with same hyper-parameter values, the average values of the peak signal-to-noise ratio (PSNR) are 5.35% and 14.22% higher than those of SRCNN and U-net when the low-resolution velocity field images are taken as model inputs for the testing dataset reconstruction with the SRUnet model. Meanwhile, the average values of the structural similarity index metric (SSIM) are 1.09% and 3.33% higher than those of SRCNN and U-net. When the super-low resolution flow field images are taken as model inputs for the testing dataset reconstruction with the SRUnet model, the average values of the PSNR metric are 8.31% and 2.76% higher than those of

[1] State Key Laboratory of Hydroscience and Engineering, Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China.

SRCNN and U-net. Meanwhile, the average values of the SSIM metric are 5.81% and 2.25% higher than those of SRCNN and U-net.

Nomenclature

X	= input image
Y	= output image
F	= convolution operation
$W^{(l)}, H^{(l)}, D^{(l)}$	= width, height, and length of X for layer l
k, m	= size and numbers of the convolution kernel
S	= length of the convolution kernel sliding
P	= size of padding
$\mathbb{R}^m, \mathbb{R}^n$	= space with m dimensions and n dimensions
$\mathcal{F}$	= mapping function from X to Y
Θ	= parameters of the neural network
$L(\Theta)$	= loss function of the neural network
N	= size of the training dataset
C_α, C_β, K	= pixels of the horizontal and vertical directions of the input image X , number of color channels
$\mathbb{F}$	= bicubic interpolation
$f(i, j)$	= pixel value of the feature map at coordinate point (i, j)
$S(x)$	= formula of convolution sampling
I, J	= pixel matrix of two different images
MSE	= total error between the pixels of images I and J

p, q = width and height of image resolution

MAX_I^2 = maximum pixel value of image I

$PSNR$ = performance criterion to assess the image resolution

$l(x,y), c(x,y), s(x,y)$ = matrix of luminance, contrast, structure

$\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$ = mean value of sample x, mean value of sample y, variance of sample x, variance of the samples y, covariance of samples x and y

c_1, c_2, c_3 = constant values to smoothen the formula

$SSIM$ = metric to assess the image resolution

$\boldsymbol{u}$ = velocity field of incompressible fluids

$\boldsymbol{p}$ = pressure of incompressible fluids

v = kinematic viscosity

$g(x)$ = Gaussian white noise

$G(x,y)$ = Gaussian blur kernel

I. Introduction

THE control equations and theoretical models in classical fluid dynamic theory are derived on the basis of some physical conservation laws. Earlier analytical methods encountered great difficulties in dealing with complex real engineering problems. Although computational fluid dynamics (CFD) technology has greatly reduced the difficulty of hydrodynamic analysis and design costs, many fundamental and engineering application problems are waiting to be solved. The mass data produced by numerical computation and experimental measurements of fluid dynamics are the original big data [1]. The methods by which to promote the extension of numerical computation and data processing of fluids to artificial intelligence through intelligent empowerment of data to establish a link between theory, computation, and experiment and to explore a new paradigm of fluid mechanics research have been the major topic of research nowadays [2,3,4].

The use of artificial intelligence techniques in experimental measurements and numerical simulations is a frontier and major topic of research in CFD. The utilization of deep learning techniques to construct high-fidelity flow field structures in experimental measurements and numerical simulations is an even newer research direction. Deep learning models can be used not only to fit the control equations of fluid mechanics [5] but also to solve turbulence models [6]. In fluid mechanics control equations, deep learning is supported by fluid experiments and numerical simulations. After enough samples are collected, fluid mechanics problems are solved with high accuracy by employing deep learning models based on conservation laws and gauge properties to achieve a rapid solution of the flow equations [7,8]. Conservation laws and important physical a priori information can be integrated into deep learning models to improve the accuracy of the models [9,10]. In this work, physically informed, rapidly converging, and highly accurate neural network models are developed [11,12]. Traditional turbulence modeling aims to develop mathematical correlations between time-averaged flow, spatial location, and Reynolds stress tensor or turbulent vortex viscosity. The principle of deep learning models for solving turbulence modeling is to build data-driven black-box models directly from large amounts of data [13], with neural network models as the main focus to successfully achieve coupled solutions between the models and the Navier–Stokes (N-S) equations [14,15]. Furthermore, deep learning models can be used to characterize and quantify the uncertainty of the computational results of traditional turbulence models [16,17], and data-driven models can be constructed with the deviation of the RANS model parameters and the corresponding parameters as the optimization objectives. The optimal parameters of the RANS model can be obtained by calculating the minimum value of the model optimization objective and solving for it, which improves accuracy [18]. Although these deep learning models can produce high accuracy, relying on large amounts of data as model inputs or increasing the depth and width of the neural network to fit complex mapping relationships is not the ideal modeling strategy. It would be better to simplify the input information and complete the prediction task with a smaller amount of sample data while maintaining guaranteed accuracy [19]. The neural network model must achieve high accuracy and excellent generalization performance to attain a balance between the two. No method that can offer a clear definition and generalization bound for the generalization performance of models in the field of deep learning has been established because a precise description of the scope of the application of model generalizations is difficult to formulate [20];

moreover, one can only rely on the continuous application of the model to solve different practical problems to ensure that qualitative analyses of the generalization ability of the model can be performed [21,22].

When numerical parameters are used as model inputs, numerical regressions are executed using neural network models with multilayer perceptrons (MLPs). Convolutional neural networks (CNNs) have been studied with regard to image processing problems in fluid mechanics with the widespread use of MLPs because the intrinsic mechanism of CNNs can automatically extract important feature information from images without manual selection of the extraction, which makes it much easier to build models and improve the effectiveness of the models. Particle image velocimetry techniques have been proven to capture the more intricate features of turbulent structures from particle images [23] and extract 2D velocity fields [24]. Accordingly, new methods for flow field reconstruction in fluid mechanics have been developed, along with various CNNs in deep learning techniques [25,26], where the theory and methods of image hyper-segmentation reconstruction are effective for flow field reconstruction [27,28]. Fukami et al. [29] used the hybrid downsampled skip-connection/multi-scale model (DSC/MS) to reconstruct turbulent flow fields, which have demonstrated 1000 snapshots are sufficient to recover the flow field from coarse data, although the experiment used 10000 snapshots to training the machine learning model. Liu et al. [30] proposed a multiple temporal path CNN model (MTPC) to train open-source flow field image data to reconstruct the flow field with high fidelity from the perspective of image super-resolution. 1000 snapshots were also required to reconstruct high-fidelity local regions in complex turbulence from the extremely coarse and blurred flow field images, demonstrating the effectiveness of the super-resolution reconstruction model, since if there are less than 1000 snapshots, the model will not be convergent to reconstruct the flow field completely and clearly. Although both adopted the image super-resolution technique to rebuild the flow field, they only reconstructed a local region of the flow field. The model of smooth local reconstruction cannot be applied to the reconstruction of the flow field as a whole because the pixel features of the small local areas of the flow field are easier to extract. By contrast, the pixel features of the entire flow field are more challenging to extract, due to their large size.

Among the areas where computer vision is being applied, image super-resolution reconstruction has been a major research topic in the field of intelligent medicine or video analysis; hence, the deep learning model is considered to be migrated to the flow field reconstruction of turbulent flows in fluid mechanics. The first successful application of CNNs to reconstruct low-resolution images was the SRCNN model proposed by Dong et al. [31]; the average peak signal-to-noise ratio (PSNR) metric tested can reach 32.45. New hyper-segmentation models, such as DRCN [32], LapSRN [33], SRGAN [34], and EDSR [35], were later proposed due to the insufficient convolutional depth of SRCNN, which leads to the problem of small perceptual fields. In addition to frameworks for the application of image super-resolution reconstruction, models for applied image segmentation may also be adopted. For example, the popular neural network U-net [36] can extract image features with high precision using the structure of skip-connection; hence, the RUnet model can be applied to image super-segmentation [37]. The model is an upgrade on the existing image-segmentation network U-net, which is more suited to the task of image super-segmentation. However, some feature information is inevitably lost because the structure of skip-connection uses feature fusion by cropping the feature map. Therefore, feature fusion must be modified in the structure of skip-connection to enhance the model's ability to extract features.

The velocity field of a 2D cylindrical bypass flow is studied in this work. Moreover, SRUnet models with better performance are proposed in a comprehensive analysis of the advantages and disadvantages of the application of SRCNN and U-net models applied to image hyper-segmentation, which can overcome the problems of small SRCNN perceptual field and asymmetric U-net feature map fusion under the same parameters. The three models are separately applied to accurate open-source arithmetic data,38 and the high accuracy of the SRUnet model for the entire flow field image reconstruction is verified. Finally, it is demonstrated that the SRUnet model achieves a certain generalized level of performance while satisfying the accuracy criterion by reconstructing the hydrofoil velocity field obtained from the CFD calculations. The SRUnet model produces a balance between accuracy and generalization performance, both of which are high. In addition, in tests, 201 and 24 snapshots were required for the two examples of cylindrical flow and winged type, respectively, whereas previous studies [29,30] have required about 1000. Therefore, this model is less dependent on the amount of data. The proposed SRUnet model can serve as the foundation for the study of high-precision agent models in CFD.

II. Methodology

A. Details of CNN

A CNN is a deep neural network that has excellent performance in a variety of fields, such as computer vision and speech recognition. Manipulation of the convolutional layer is considered to be a fundamental CNN operation. The function of this layer is to implement the transformation for the input image, which can automatically complete feature abstraction by performing a convolution operation on an input image with convolution kernels. A convolution kernel, also known as a convolution matrix or convolution mask, is a small matrix that has a height and width smaller than the image to be convolved. The dot product of the convolution kernel and the image is computed at each spatial location, and the sliding length is called stride S. The image and convolution kernel should be altered using a discrete 3D formula, as follows:

$$Y(u,v,p)=(X*F)(u,v,p)=\sum_i\sum_j\sum_k X(i,j,k)F(u-i,v-j,p-k) \tag{1}$$

where X is the input feature map, F is the convolution kernel, and Y is the output feature map after the convolution operation. In Figure 1, the size of the output feature map Y after convolution is presented as follows when the dimension of the input image X of a layer l in CNN is $W^{(l)}\times H^{(l)}\times D^{(l)}$, the size and number of the convolution kernel are $k\times k$ and m, respectively, the stride is denoted as S, and the padding is denoted as P:

$$W^{(l+1)}=\frac{W^{(l)}-k+2P}{S}+1 \ , \tag{2}$$

$$H^{(l+1)}=\frac{H^{(l)}-k+2P}{S}+1 \ , \tag{3}$$

$$D^{(l+1)}=m \ . \tag{4}$$

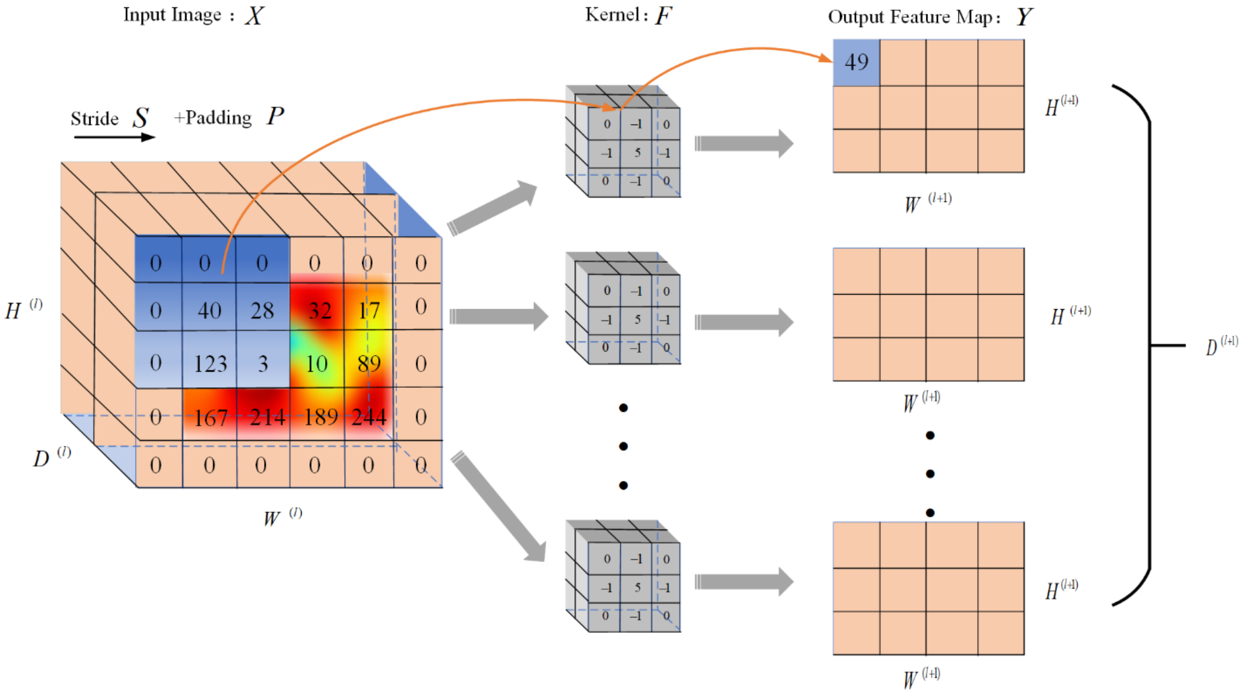
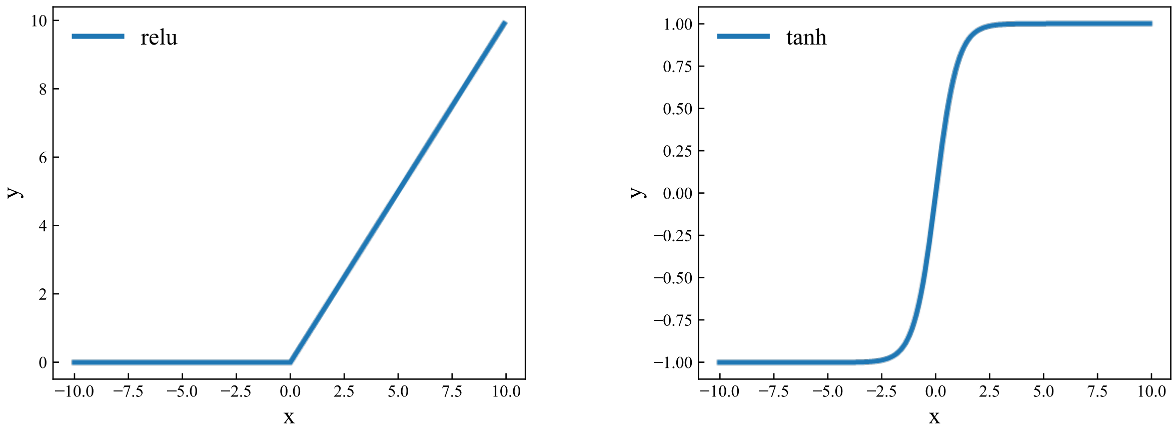


Figure 1. Convolutional layer

In addition, output feature map Y should be nonlinearized by means of an activation function that can increase the degree of the nonlinearization of output as the final component of the convolutional layer before it moves to the subsequent layer. The activation function in the convolutional layer is either selected as a rectified linear unit (ReLu) or as a hyperbolic tangent function (Tanh) (Fig. 2).



(a) ReLu function

(b) Tanh function

Figure 2. Activation function

The pooling layer is another commonly used component in CNNs, and it is generally utilized to imitate the human visual system for the reduction of data dimensionality to depict an image with high-dimensional features. The following are the three most important purposes for adopting a pooling layer: simplify the density of information and decrease its redundancy, enhance the scale and rotation invariances of the CNN model, and avoid or reduce overfitting. The two universal methods for the pooling layer are maximum pooling and average pooling. In the maximum pooling method, the result of every region of the image originates from the maximum value in the corresponding region during the forward propagation in the neural network. During backward propagation, the gradients are only effective for the maximum value of forward propagation, while those of the other positions become zero. The most significant advantage of the maximum pooling method is its capacity to learn the edge and texture structure of the image. In the average pooling method, the result of every region of the image originates from the average value in the corresponding region, reducing the bias of the estimated mean and improving the robustness of the CNN model.

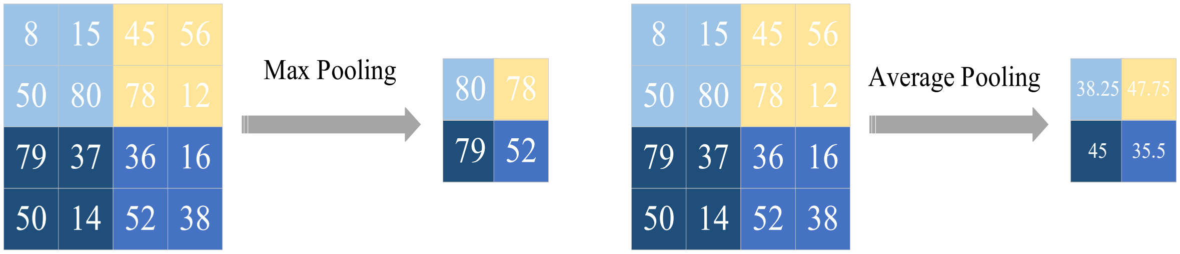


Figure 3. Pooling layer

B. Proposed Model

CNN is excellent at fitting the mapping of high-dimensional nonlinearization. The input of the CNN model can be given by low-resolution flow field images $X \in \mathbb{R}^m$. Meanwhile, the output can be provided by high-resolution flow field images $Y \in \mathbb{R}^n$. The objective of the CNN model is to search for the best mapping function $\mathcal{F}$ to satisfy $\mathcal{F}(X;\Theta) \approx Y$ as accurately as possible. Searching for the best mapping fucntion means finding the best parameters Θ to minimize the value of loss function $L(\Theta)$ by updating the parameters Θ during the forward computation and backward propagation. The loss function $L(\Theta)$ is defined as follows:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{F}(X_i;\Theta) - Y_i\right\|^2 , \tag{5}$$

where N is the size of the training dataset. The loss function is the error between the predicted and the reference images, which is also known as the mean squared error (MSE), and it can improve the model prediction quality.

The SRCNN model was created to address the problem of capturing the details of bioimages. We try to apply the model in flow field reconstruction, such as reconstructing the velocity field image of the 2D cylinder wake flow. The principle of the SRCNN model is described as follows: The input image X is first magnified by twofold by bicubic interpolation [39]. Then, the new image will go through three convolution layers, including convolution kernels (the sizes of the kernels are 9×9, 5×5, and 5×5). The dimension of the final output image Y will remain consistent with the input image X after bicubic interpolation. For instance, if $m = C_\alpha \times C_\beta \times K$, then $n = (2C_\alpha) \times (2C_\beta) \times K$, where C_α and C_β represent the pixels of the horizontal and vertical directions of the input image X , respectively; and K means the number of color channels ($K = 1$ indicates that the image is filled with gray pixels, and $K = 3$ denotes that the image is filled with three channel colors [RGB , red, green, and blue]).

The proposed U-net model is mainly used to develop the image segmentation field, which can enhance the model's ability to segment the boundaries of images and improve the model's accuracy in recognizing images. The principle of the U-net model is described as follows. The input image X is transformed through a series of downsampling operations that include convolution and maximum pooling at first on the left side, where four blocks are present, each with two effective convolutional layers and a maximum pooling layer. Then the number of feature maps double after every downsampling operation, and the size of the feature maps is halved. Finally, the feature maps obtained are 32×32 on the left side. The size of the output feature maps of the left side after an effective convolution at the bottom layer is 28×28. Thereafter, the new output feature maps are transformed by a series of upsampling operations that include merging the results of the upconvolution (transpose convolution) on the right side and the feature maps passed from the left side each with its respective layer; four blocks are present, each with two effective upconvolutional layers and the operation merging with feature maps by cropping. The number of feature maps is cut in half after every upsampling operation, and the size of the feature maps doubles. The size of feature maps after the upconvolution on the right side is not equal to that on the left side; hence, the feature maps from the left side of the U-net model must be cut and cropped so that they can merge. Furthermore, the last layer must be the softmax layer to output the label of the image to ensure that the U-net model can accomplish the image segmentation. Generally, the softmax layer is often used in completing classification tasks. However, the problem of super-resolution reconstruction belongs to the domain of regression tasks so we decide to cancel the softmax layer during the process of designing neural network framework here. Therefore, the U-net model cancels the softmax layer in the original U-net model to obtain the output image Y with the same size as the input image X.

In this work, the principle of the SRUnet model, which was proposed by combining the SRCNN and U-net models, is described as follows: The input image X is first magnified by twofold by bicubic interpolation. Then, the feature of the new image will be abstracted by two convolution layers, including convolution kernels (the sizes of the kernels are 9×9 and 5×5). Finally, high-resolution output image Y is obtained by the improved U-net architecture. We mainly improve the way of U-net feature fusion and perform the fusion of two different sizes of features by bicubic interpolation instead of simple edge cropping operation. The principle of feature fusion with bicubic interpolation is illustrated as follows:

$$\mathbb{F}(i+v,j+u)=\sum_{row}\sum_{col}f(i+row,j+col)\cdot S(row-v)\cdot S(col-u)\,, \tag{6}$$

where i and j denote the coordinates of an input image, $f(i,j)$ denotes the feature maps passed from the left side, row and col denote the deviation between current coordinates (i,j) and others in $f(i,j)$, v and u denote the deviation between the current coordinates (i,j) and the coordinates after interpolation, and $\mathbb{F}(i+v,j+u)$ represents the feature map after interpolation. The formula of convolution sampling is denoted as $S(x)$

$$S(x)=\begin{cases}1-(a+3)x^2+(a+2)|x|^3, & 0\le|x|\le1\\ -4a+8a|x|-5ax^2+a|x|^3, & 1<|x|\le2\end{cases}, \tag{7}$$

where the different values of a can decide the use of different spline functions. For example, when $a=-0.5$, $S(x)$ is general spline function called triple sampling function and given by

$$S(x)=\begin{cases}1-2.5x^2+1.5|x|^3, & 0\le|x|\le1\\ -2-4|x|+2.5x^2-0.5|x|^3, & 1<|x|\le2\end{cases}. \tag{8}$$

The size of the feature map on the left side of SRUnet can be made consistent that on the right side by implementing the above-mentioned bicubic interpolation. The feature information of the low-level feature map will not be lost.

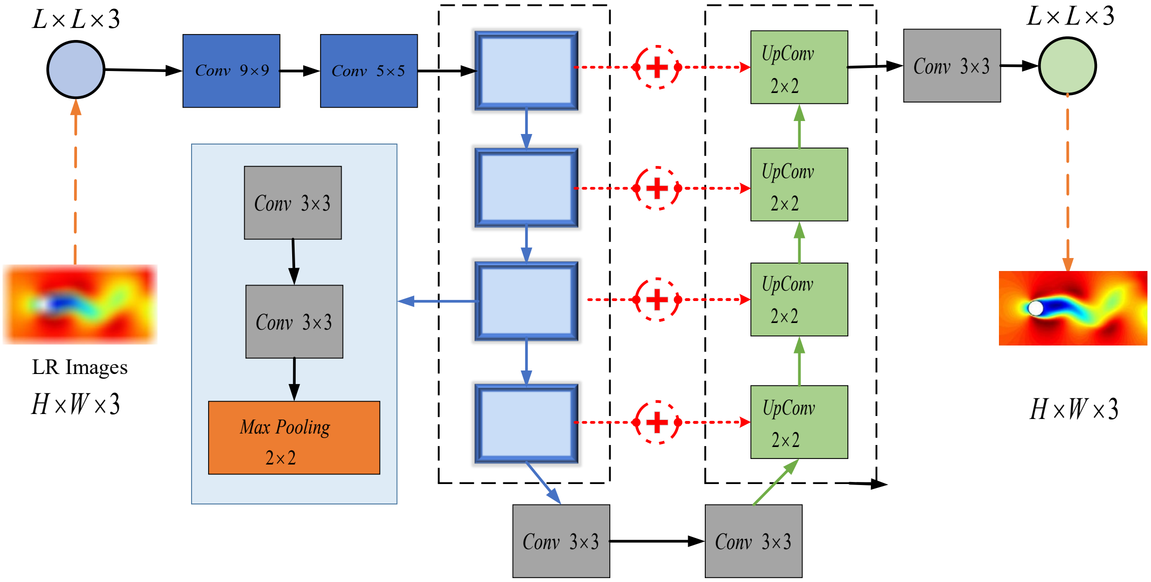


Figure 4. Frame of the SRUnet model

The low-resolution image is first scaled up to the target size for the SRCNN model and nonlinearized by three convolution layers. The final result will be the high-resolution image. The accuracy of the model reconstruction is not enough to reconstruct the details of the flow field because the simple neural architecture only trains the single scale factor, and the perceptron area is smaller. The U-net model mainly is applied to the image segmentation field, and its shape presents a symmetric structure like “U”. The left side of the U-net model is assembled by convolution blocks. Meanwhile, the right side is assembled by up-sampling block. The feature maps computed by every convolution block will connect the output of the corresponding up-sampling block, which can merge the high-level and low-level features into the final feature maps and improve the accuracy of model prediction by merging feature maps with different scales. The feature map needs to be copied and cropped before fusion, and the form of the cropped edges is asymmetric to the feature map. Accordingly, the effect is not as good as using the bicubic convolution interpolation method.

A new CNN named as SRUnet is formed by embedding the feature fusion method of U-net into the network structure of SRCNN after the above-mentioned improvement. The architecture of SRUnet increases the perceptual field of the convolutional layer of the network and efficiently solves the problem of asymmetric feature maps of the U-net network structure, which can provide some theoretical interpretability.

C. Performance Metrics

PSNR is a prevalent performance criterion used to assess the image resolution, and it is often applied in the engineering project containing maximum signal and background noise. PSNR is a standard reference metric to measure the similarity between the image after processing and the original image, whose unit is db, because a great difference exists between the suppressing image and the original image. For example, the calculation formula of the PSNR metric for images I and J is as follows:

$$MSE = \frac{1}{pq}\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} \| [I(i,j) - J(i,j)] \|^2, \tag{9}$$

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_I^2}{MSE}), \tag{10}$$

where $I(i,j)$ and $J(i,j)$ represent the pixel value at some coordinates (i,j) of images I and J; p and q are the width and height of image resolution, respectively; MSE denotes the total error between the pixels of images I and J; and MAX_I^2 is the maximum pixel value of image I, where the pixel values are defined as the binary of B bits to calculate the MAX_I^2 by the formula defined as $MAX_I = 2^B - 1$. Although PSNR is widely used as an evaluation metric for computer vision tasks, such as super-resolution, denoising, and deblurring, it sometimes presents completely opposite results to human vision according to a large number of experimental results. Hence, the structural similarity index metric (SSIM) was subsequently proposed.

SSIM is a comprehensive measure of the image described by three indicators: luminance, contrast, and structure, whose calculation formulas are defined as follows:

$$l(x,y)=\frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1}, \tag{11}$$

$$c(x,y)=\frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2}, \tag{12}$$

$$s(x,y)=\frac{\sigma_{xy}+c_3}{\sigma_x\sigma_y++c_3}, \tag{13}$$

where μ_x is the mean value of sample x ; μ_y is the mean value of sample y ; σ_x^2 is the variance of sample x ; σ_y^2 is the variance of the samples y ; σ_{xy} is the covariance of samples x and y ; c_1, c_2, and c_3 represent the constant values that were used to smoothen the formula and prevent dividing zero, whose values are as follows according to engineering experience:

$$c_1=(k_1L)^2,\ \ c_2=(k_2L)^2,\ \ c_3=c_2/2, \tag{14}$$

where L is the maximum value in the range of pixels, which is denoted as 2^B-1, in which $k_1=0.01$, and $k_2=0.03$. Let

$$SSIM=[l(x,y)^\alpha\cdot c(x,y)^\beta\cdot s(x,y)^\gamma]. \tag{15}$$

Then, the expressions of $l(x,y)$, $c(x,y)$, and $s(x,y)$ above are substituted. Let the values of α , β , and γ be equal to one. Finally, the calculation formula of the SSIM metric between samples x and y can be deduced as follows:

$$SSIM=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}. \tag{16}$$

The range of the SSIM metric value is [0,1]. If the SSIM metric value is larger, then the distortion or bias of the image will be smaller. The SSIM metric is more accurate to use as a measure of model performance in image denoising and evaluating image similarity compared with the PSNR metric. Furthermore, the SSIM metric can better represent the similarity of visual perception of the two images.

PSNR and SSIM are now highly representative metrics in image super-resolution reconstruction studies. Hence, we mainly use these two metrics in this work to evaluate the merits of the subsequent algorithm simulation results and the ability of the proposed deep learning model to reconstruct the velocity field.

D. Dataset

The velocity field image dataset used in this work is the open-source flow field data obtained by Raissi et al. [38] via CFD numerical simulation of a 2D cylindrical flow around the example. The governing equation is the N–S equation defined as follows:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{17}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{u} \bullet \nabla \boldsymbol{u} - \nabla \boldsymbol{p} + \nu \nabla^2 \boldsymbol{u} \tag{18}$$

The N–S equations describe the flow of incompressible fluids. Variable $\boldsymbol{u}$ represents the velocity field, $\boldsymbol{p}$ indicates the pressure, and ν is the kinematic viscosity.

The velocity field image dataset obtained by visualizing the numerical data with a total number of 201 groups is divided into training, validation, and testing datasets for training, validating, and testing the model, respectively. The model inputs are the standard reference velocity field images (Fig. 5a), merged with Gaussian noise to simulate the errors in the experiment at first. Specifically, each pixel value of the image is added with Gaussian white noise, defined as follows:

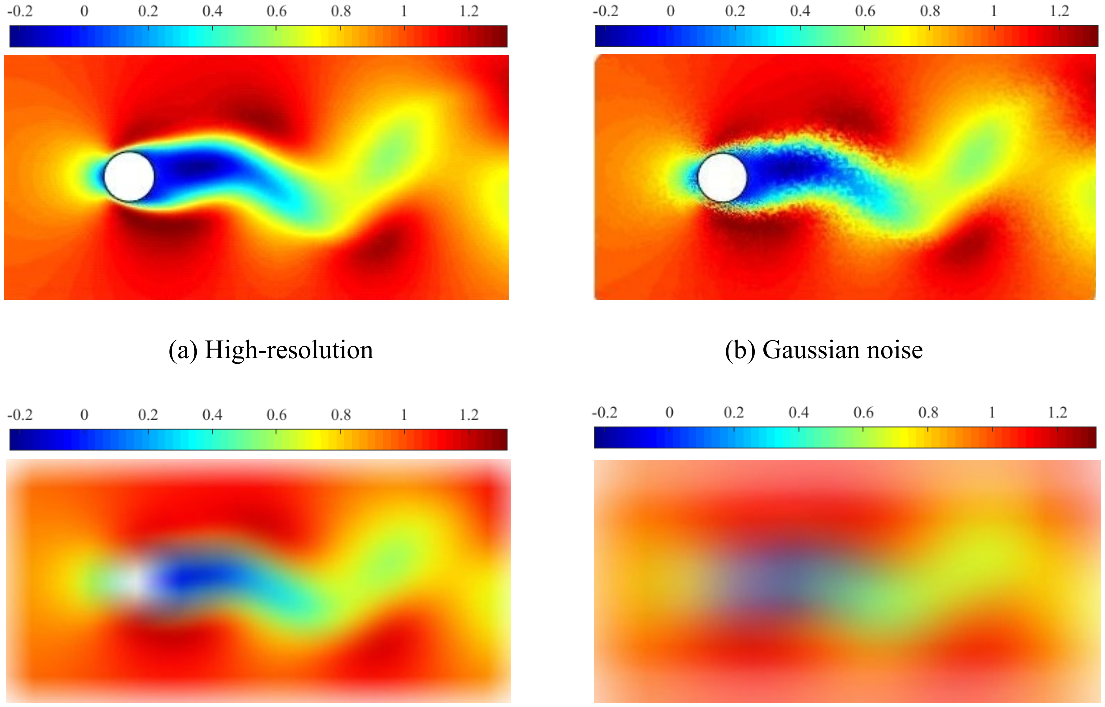
$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{(-\frac{(x-\mu)^2}{2\sigma^2})} \tag{19}$$

Then the velocity field images in Fig. 5(a) and (b), with errors generated by adding Gaussian white noise, are applied to the Gaussian blur algorithm to degrade their resolution to obtain low-resolution velocity field images of fluid. The implementation of the Gaussian blur algorithm is performed mainly by convoluting the image pixel matrix with the Gaussian blur kernel to generate the weight matrix first and then producing the Gaussian blur value (in Eq. 20, $\sigma = 30$) with each position in the image by summing the pixel points with the weight matrix. Accordingly, the final blurred images are obtained as model inputs.

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{(-\frac{(x^2+y^2)}{2\sigma^2})} \tag{20}$$

Two diverse resolution images are generated using two different Gaussian kernels of varying sizes (33×33 and 77×77), defined as low-resolution and super-low-resolution images, respectively (Fig. 5c and d). The SRCNN, U-net, and SRUnet models are used to predict the input data, and the output is a high-resolution and high-fidelity velocity field image. As shown in Fig. 5(c) and (d), the general structure of the velocity field can be vaguely seen at lower resolution, while the velocity field in the super-low-resolution case is completely unrecognizable. Deep learning technology should be used to reconstruct the velocity field because it is difficult to analyze and study when it has high-level distortion that prevents the details from being clearly perceived.

As shown in Fig. 5(a), the velocity field of the cylindrical bypass flow forms a continuous wake over time. The wake is gradually shed, which can reflect changes in the velocity field of the cylindrical winding flow. Thus, considerable time-variant velocity field image data are obtained, which can be used to give different inputs to the deep learning model.



(a) High-resolution

(b) Gaussian noise

(c) Low-resolution

(d) Super-resolution

Figure 5. Velocity field component image of the 2D cylindrical bypass flow

The velocity field image dataset of fluid is divided into training, validation, and testing datasets in the proportion of 80%, 20%, and 20% in the case. We would like to implement the bicubic interpolation to make the input image X of the SRCNN and SRUnet models consistent with the input image size of the U-net model after interpolation, which is $128 \times 128 \times 3$, to more validly compare the training effect of three models. Meanwhile, the hyper-parameter setting, gradient descent algorithm, and activation function of the three deep learning models are consistent (Table 1).

Table. 1. Parameter setting of the three models

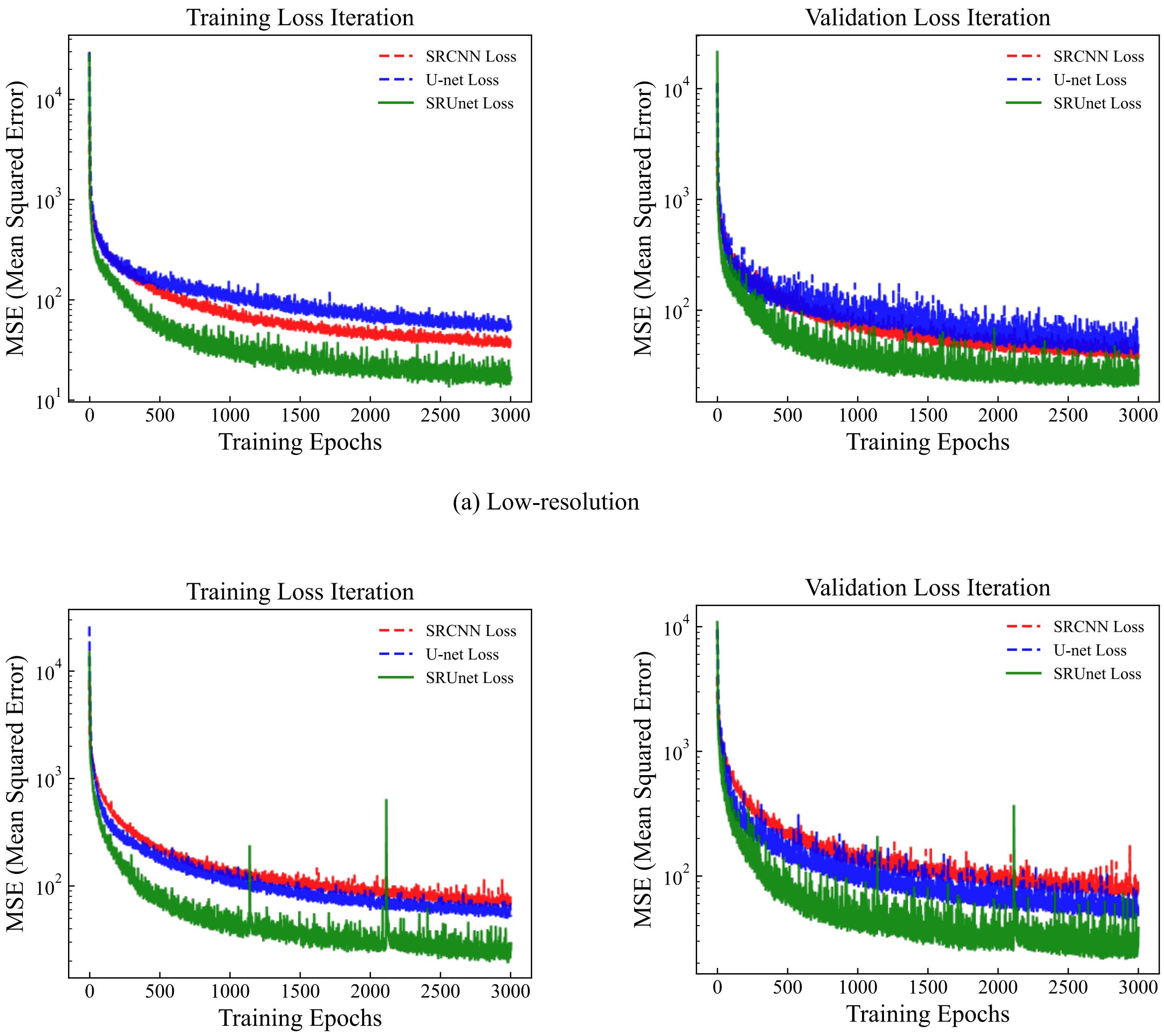
Sets	Value or Function
Learning Rate	0.001
Batch Size	8
Epochs	3000
Gradient Decent Algorithm	“Adam”
Activation Function of Last Layer	“Linear”
Early Stopping	False

The running environment of the three models is Windows 10, and the processor is i7-10750H, CPU 2.60 Hz, and the type of GPU is NVIDIA GeForce 1650Ti. The framework that is used to construct models is the TensorFlow-GPU framework, which incorporates many operators and provides great convenience for all types of deep learning models.

III. Results

A. Training

The goal of deep learning model training is to gain relative experience, such that the model can rapidly and accurately reconstruct the velocity field. Fig. 6(a) and (b) demonstrate that the training and validation losses are smaller and converge more rapidly than SRCNN and U-net when the model inputs are velocity field images with low-resolution or super-low resolution. This indicates that the training performance of SRUnet is better than those of the SRCNN and U-net models. Although the MSE metric eventually converges to approximately 10, which is relatively higher than that of the usual regression task, it is only a computational criterion used for model convergence and does not reflect how well the model performs. The metrics used to evaluate the model strengths and weaknesses in this work are PSNR and SSIM.



(a) Low-resolution

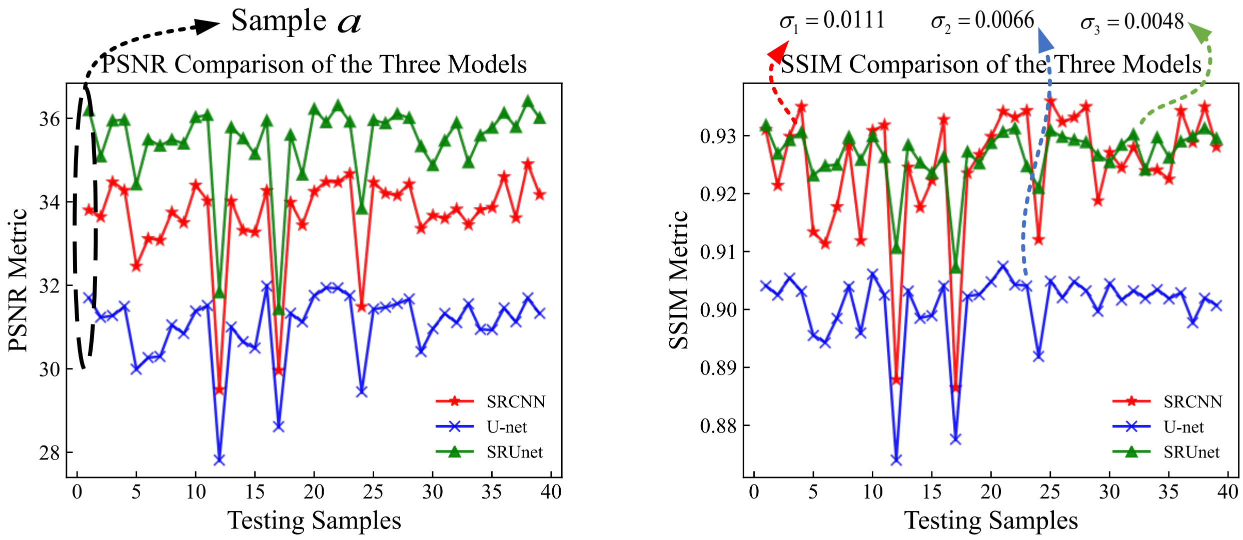
(b) Super-low resolution

Figure 6. Iterative training process of loss function (including the loss value of training and validation)

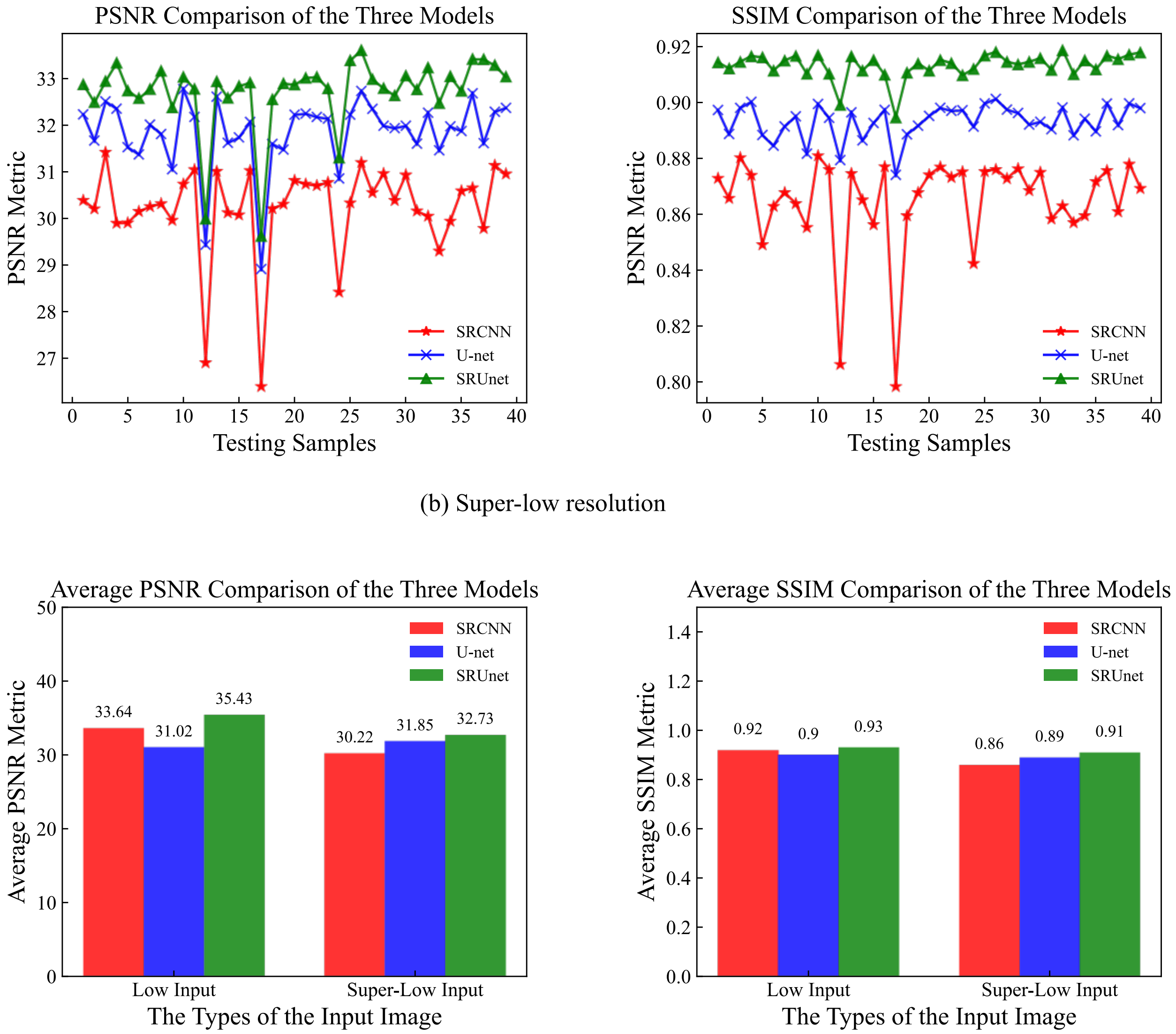
Although the convergence process of the SRCNN and U-net models is also relatively fast, and the overall convergence speed and the MSE of the final convergence of both are comparatively close to each other, they are worse than the SRUnet model. Hence, the SRUnet model based on two models can be used to improve the small sensory field of SRCNN and the complex feature fusion method of the U-net model to a certain extent, thereby overcoming the defects of not being able to fully learn the velocity field features.

B. Testing

The purpose of deep learning model testing is to verify the generalization capability of the model to predict unseen input data (satisfying the condition of independent homogeneous distribution) to achieve validation of the feasibility of the model application. The testing results of three models with two different input datasets are shown in Fig. 4(a) and (b), with two metrics, namely, PSNR and SSIM. The proportion of the test that is set in the total dataset is 20%. As shown in Fig. 7(a), the PSNR metric of the SRUnet model is greater than those of the SRCNN and U-net models when low-resolution flow field images are used as model inputs. However, the right side indicates that the SSIM metrics predicted by SRUnet for some samples are not greater than those of SRCNN and U-net, while the standard deviations of the three numerical distributions are 0.0111, 0.0066, and 0.0048, respectively. The smaller standard deviation represents a more stable and robust model. The standard deviation of the samples predicted by our proposed SRUnet model is 0.0048, and the mean obtained for SSIM is greater than those of SRCNN and U-net (Fig. 7c). Thus, the SRUnet model is more stable and less dependent on the samples. As shown in Fig. 7(b), the PSNR metric of the SRUnet is greater and more stable than those of the SRCNN and U-net models when the super-resolution flow field images are model inputs. This indicates that the robustness of the SRUnet model is stronger than those of the SRCNN and U-net models. A comparison between Fig. 7(a) and (b) demonstrates that the precision of the velocity flow reconstruction with the super-low resolution is lower than that with low resolution. In summary, the testing performance of the SRUnet model is better than those of the SRCNN and U-net models.



(a) Low-resolution



(b) Super-low resolution

(c) Distribution of PSNR and SSIM of total testing dataset

Figure 7. Distribution of PSNR and SSIM of the total testing dataset

As shown in Fig. 7(c), the average values of the PSNR metric are 5.35% and 14.22% higher than those of SRCNN and U-net when the low-resolution flow field images are taken as the model inputs to test the dataset reconstruction with the SRUnet model. The average values for the SSIM metric are 1.09% and 3.33% higher than those of SRCNN and U-net. The average values for the PSNR metric are 8.31% and 2.76% higher than those of the SRCNN and U-net models when the super-low resolution flow field images are taken as the model inputs for testing dataset reconstruction with SRUnet model. The average values for SSIM are 5.81% and 2.25% higher than those of SRCNN and U-net.

Sample a is selected from the reconstruction results of the testing dataset to create the visualization (Fig. 8). When the low-resolution flow field images are taken as the model inputs, the PSNR metric of the SRUnet model is 37.45, approximately 11.56% higher than the SRCNN model and 18.12% higher than the U-net model. The SSIM metric of the SRUnet model is 0.937, which is approximately 1.52% higher than the SRCNN model and 3.65% higher than the U-net model. When the super-low-resolution flow field images are taken as the model inputs, the PSNR metric of the SRUnet model is 32.877, which is approximately 8.19% higher than the SRCNN model and 2.01% higher than the U-net model. The SSIM metric of the SRUnet model is 0.914, which is approximately 4.70% higher than the SRCNN model and 2.35% higher than the U-net model. Fig. 8 demonstrates that the velocity field image of the SRUnet model reconstruction is clearer than those of the SRCNN and U-net models. Although no obvious differences can be observed in some regions for those three models, the metrics quantify the evaluation of the reconstruction consequences from a holistic perspective. A comparison between PSNR and SSIM metrics indicates that the accuracy of the SRUnet model is higher, and the quality of reconstructing the velocity field image is closer to the reference image.

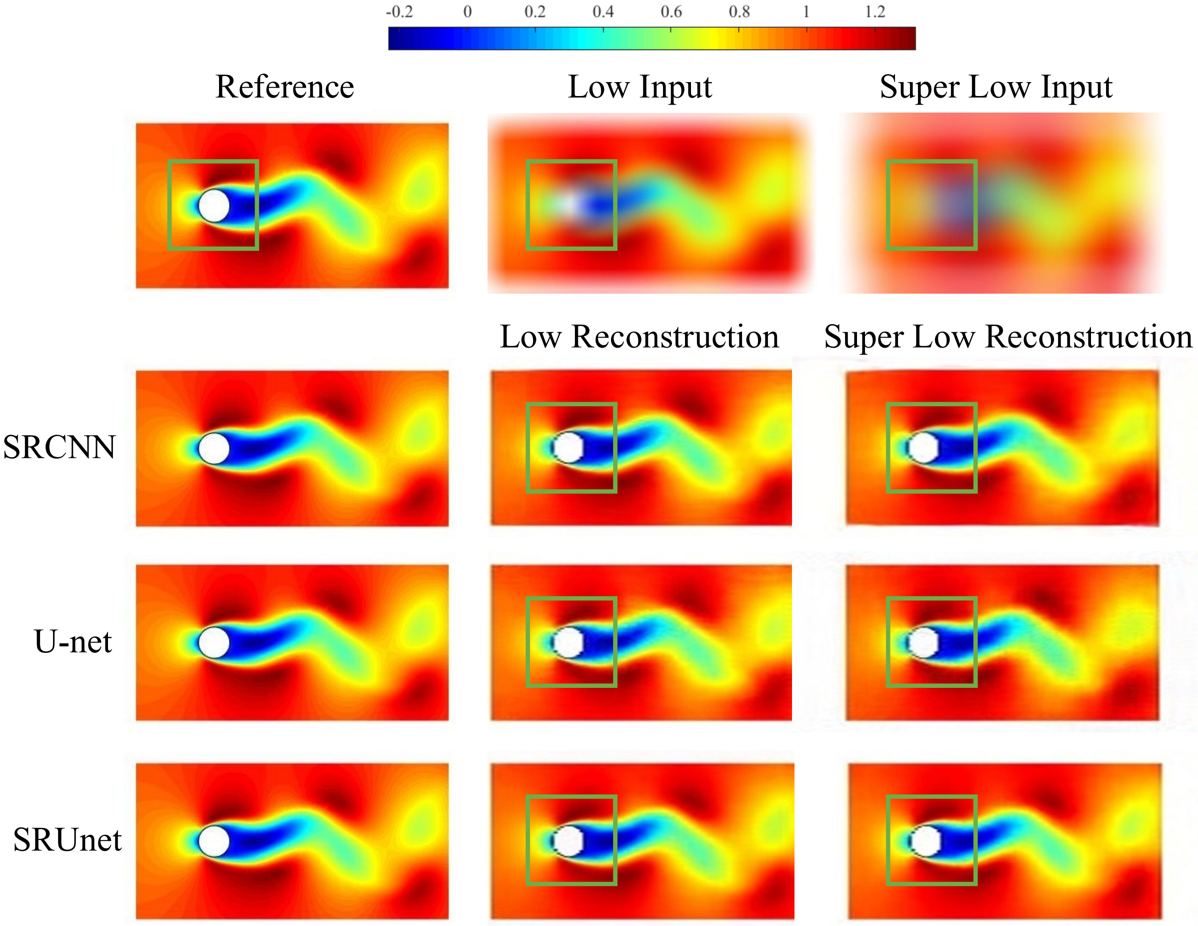


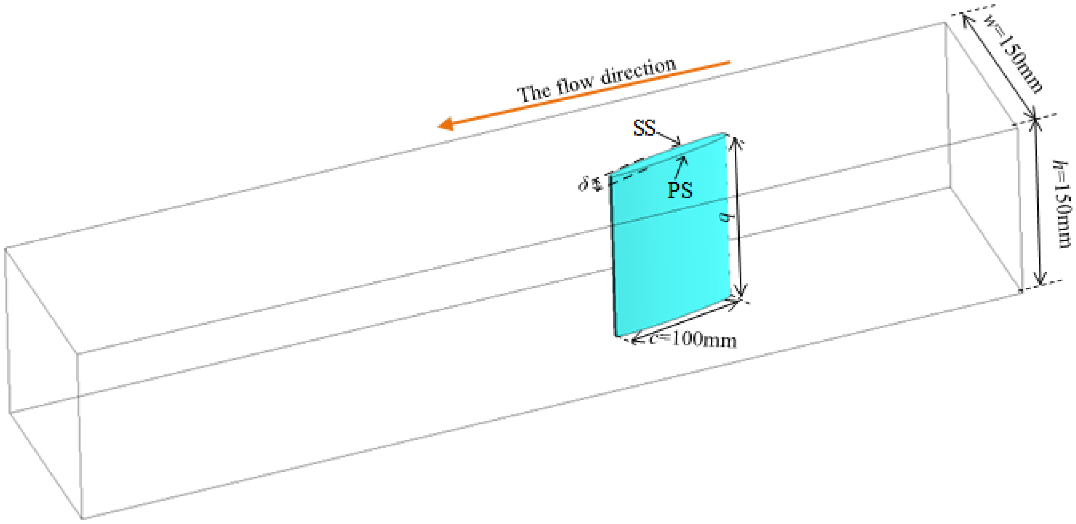
Figure 8. Reconstruction results of the velocity field image of sample a from the testing dataset

In this velocity cloud diagram, the fluid bypasses the cylinder and generates a low velocity circulating flow region on its back side. Under the Reynolds number condition, the cylinder bypass generates a Karman vortex street (i.e., a periodic vortex structure shape fluctuating back and forth on the back side of the cylinder). At this moment, the circulating flow region on the back side of the cylinder is deflected to one side and is about to generate a vortex structure rotating toward this side. Although no significant difference was observed in the reconstructed lower resolution velocity field for three models when reconstructing the low-resolution velocity field, the vortex structure reconstructed by the SRUnet model is significantly clearer than the reconstructed results of the two models, namely, SRCNN and U-net, when reconstructing the ultra-low resolution velocity field, as shown by the green box markers in Fig. 8.

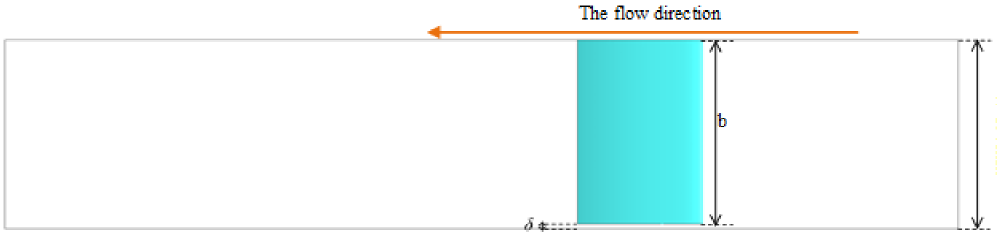
C. Model Generalization

The term generalization here means to compare the performance of deep learning models in the different examples. We can define it as "vertical generalization" while the traditional generalization can be defined as "horizontal generalization" that also be called "testing". The high predictive accuracy of the SRUnet model proposed in this work can be used not only in the case of flow around a cylinder but also for other calculation examples, verifying the strong ability of the deep learning model when used in different situations. In this work, an NACA0009 hydrofoil39,40 (Fig. 9) is assumed to have a truncated chord (c) of 0.1 m, a maximum thickness (h) of 0.0099 m, and a span (b) of 0.15 m. The angle of incidence is 10°. The fluid density is 997 kg/m3 and the inlet pressure is 1 bar. We choose a reconstruction of the velocity field in the longitudinal section (x = 0.075 m) of the NACA0009 hydrofoil with a tip clearance to observe whether the proposed SRUnet model can reconstruct its velocity field with high precision using tiny data.

STAR-CCM+ 16.02 is used in this work. The SST k–ω turbulence model with y+ wall treatment is used to simulate the unsteady flow field, following previous studies.40,41 A normal velocity of 10 m/s with low turbulence intensity is set as the inlet boundary condition41, and the outlet boundary is chosen as the static pressure. No-slip wall boundary conditions are applied to the hydrofoil surface and tunnel walls. The time step is 1 × 10-3 s.



(a) Isometric view of the test section



(b) Gap

Figure 9. Physical model of the NACA0009 hydrofoil

Fig. 9 shows the final obtained hydrofoil velocity field diagram showing periodicity in the steady-state moments after the aforesaid numerical simulation.

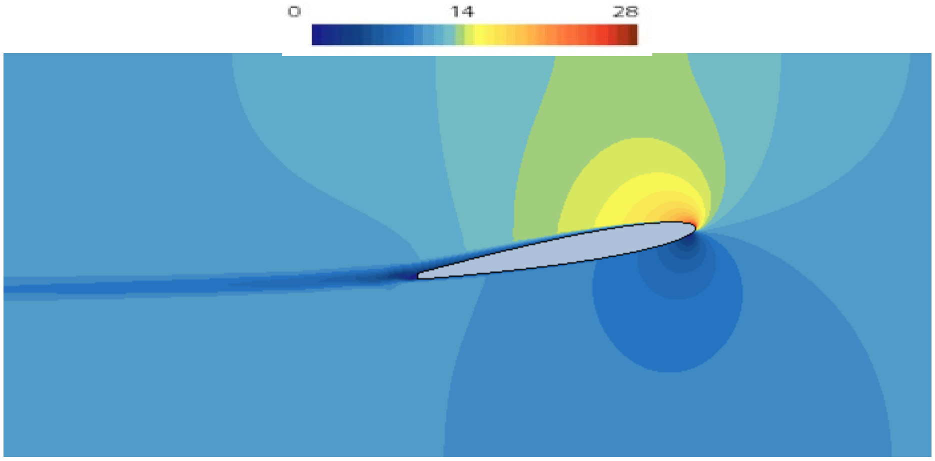
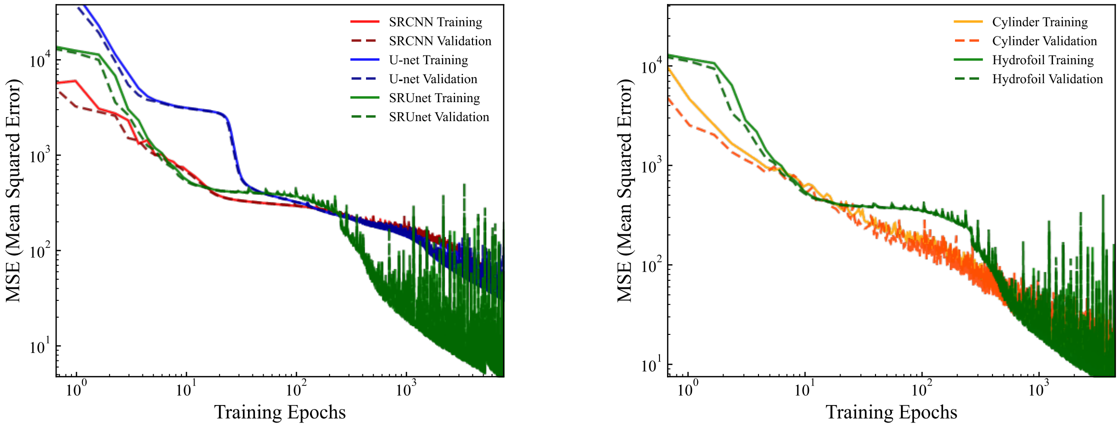


Figure 10. Velocity field of the NACA0009 hydrofoil

A hydrofoil velocity field dataset that contains 24 images is obtained by CFD numerical simulation. The same image preprocessing technique as in Section 2.4 is used to preserve the hydrofoil velocity field map after interception, remove the velocity scalar bar, and obtain a low-resolution velocity field image using a Gaussian blurring algorithm with a Gaussian kernel of 33×33, which is reconstructed using the SRUnet model. The split ratio of the training, validation, and testing datasets and the hyperparameter settings are exactly same as those in Section 2.4. During the training of the SRCNN, U-net, and SRUnet models and that of the SRUnet model on two arithmetic cases, the SRUnet model still has the best training effect of the four (i.e., fast convergence and small convergence error even on different cases). Specifically, the convergence is faster, and the convergence error is smaller. Fig. 10(b) shows that the SRUnet model is better trained for the hydrofoil velocity field than for the cylindrical bypass velocity field. Thus, the comparison between the three different models of the same arithmetic case and that between two different cases of the same model demonstrated that the training of the SRUnet model has some generalization to various arithmetic cases.



(a) (b)

Figure 11. Training

The testing dataset of the NACA0009 hydrofoil velocity field diagram is separately predicted and reconstructed using the three models. The reconstructed results are calculated to obtain the mean values of the PSNR and SSIM metrics of the testing dataset, which are compared to the results of the 2D cylindrical bypass flow example (Table 2). According to Table 2, the SRCNN model has a higher SSIM metric than the Cylinder for the two cases (n_1 and n_2 denoting the sizes of the datasets for Cylinder and Hydrofoil, respectively). The PSNR metric is lower in the Cylinder case. The U-net model has a higher PSNR metric than the Cylinder case, but the SSIM index is lower. The proposed SRUnet model in this work has improved PSNR and SSIM metrics in the Hydrofoil case compared to the Cylinder case, and both metrics of the SRUnet model are higher than the other two models in both cases. Therefore, the generalization performance of the SRCNN and U-net models for the different arithmetic cases is weak. Meanwhile, the generalization performance of the SRUnet model is strong.

Table 2. Average results of PSNR (dB) and SSIM on the cylinder and hydrofoil datasets

Eval. Mat	Case	SRCNN	U-net	SRUnet
PSNR	Cylinder ($n_1 = 201$)	33.64	31.02	**35.43**
	Hydrofoil ($n_2 = 24$)	30.88	32.38	**38.81**

SSIM	Cylinder ($n_1 = 201$)	0.92	0.90	**0.93**
	Hydrofoil ($n_2 = 24$)	0.94	0.89	**0.96**

We select a sample reconstructed result from the test set for visualization to clearly observe the reconstructed results for the three models (Fig. 11). The reconstructed velocity field map is close to the reference velocity field image of the CFD simulation. Meanwhile, the reconstructed hydrofoil velocity fields of the SRCNN and U-net models shows a large gap, which can prove the feasibility of applying the SRUnet model to the hydrofoil velocity field reconstruction from the visual perception and indicate that the SRUnet model has a certain ability for generalization in different cases.

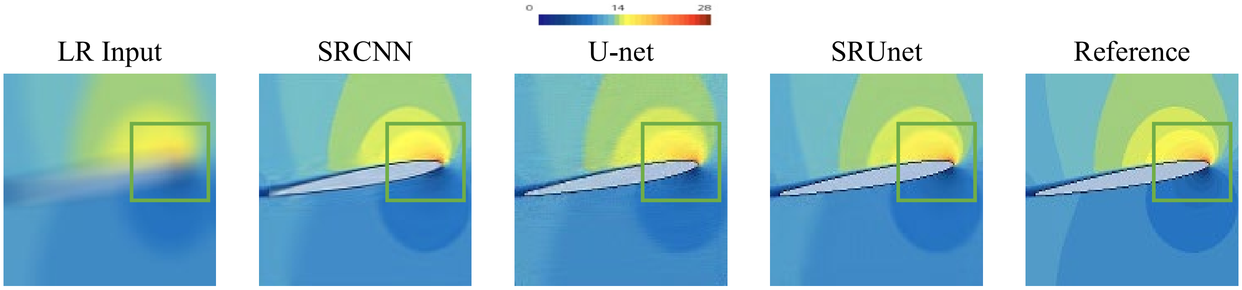


Figure 12. Results of the hydrofoil velocity field reconstruction

The velocity above the leading edge of the hydrofoil is larger, and that below is smaller. Fig. 12 shows that the velocity fields of the leading edge of the hydrofoil reconstructed by the three models are somewhat different. The flow field reconstructed by the SRUnet model, although it is different from the actual reference value in minor details, is overall similar to and clearer than the SRCNN and U-net models, indicating the superior performance of the SRUnet model.

IV. Conclusions

A valid solution for an inaccurate flow field image in the study of fluid dynamics is proposed by adopting certain widely used deep learning techniques. The SRUnet model is presented to solve the problem of reconstruction of the velocity field image of 2D cylindrical bypass flow. Moreover, the model simulation with a true and accurate fluid dataset is completed based on the analysis of the advantages and disadvantages of the SRCNN model for image super-resolution and U-net model for image segmentation and the method of feature maps' fusion by modifying the architecture of U-net. The primary conclusions are summarized as follows:

(1) The deep learning model proposed in this work can reconstruct high-accuracy velocity field images by training only a small sample data of 2D cylindrical winding flow, verifying its high accuracy for velocity field image reconstruction.

(2) The SRUnet model overcomes the problems of the smaller sensory field of SRCNN and the asymmetric fusion of the U-net feature maps under the same parameters and shows better performance in training and testing than SRCNN and U-net. The SRUnet model demonstrates a relatively good reconfiguration capability in terms of convergence speed and MSE during training and PSNR and SSIM metrics during testing.

(3) The SRUnet model is not only effective for the velocity field reconstruction of cylindrical winding flow but also for non-constant velocity field reconstruction of the hydrofoil type, with higher accuracy. The results show that the SRUnet model has a certain capability for generalization across different cases.

(4) SRUnet may have some limitations. In the two input flow field images, the lower the resolution, the lower the accuracy of SRUnet for reconstructing the flow field. Thus, the SRUnet model may show large errors for extremely inaccurate flow field reconstruction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (52109108).

References

[1] Wang, Z., Luo, K., Li, D., Tan, J., and Fan, J., "Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation," *Physics of Fluids*, Vol. 30, No. 12, 2018, pp. 125101.
doi: 10.1063/1.5054835

[2] Kutz, J. N., "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, Vol. 814, 2017, pp. 1–4.
doi: 10.1017/jfm.2016.803

[3] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, Vol. 52, 2020, pp. 477–508.
doi: 1 10.1146/annurev-fluid-010719-060214

[4] Piette, J. H., Saengow, C., and Giacomin, A. J., "Hydrodynamic interaction for rigid dumbbell suspensions in steady shear flow," *Physics of Fluids*, Vol. 31, No. 5, 2019, pp. 053103.
doi: 10.1063/1.5100814

[5] Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N., “Data-driven discovery of partial differential equations,” *Science Advances*, Vol. 3, No. 4, 2017, pp. e1602614.

doi: 10.1126/sciadv.1602614

[6] Duraisamy, K., Iaccarino, G., and Xiao, H., “Turbulence modeling in the age of data,” *Annual Review of Fluid Mechanics*, Vol. 51, 2019, pp. 357–377.

doi: 10.1146/annurev-fluid-010518-040547

[7] Long, Z., Lu, Y., and Dong, B., “PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network,” *Journal of Computational Physics*, Vol. 399, 2019, pp. 108925.

doi: 10.1016/j.jcp.2019.108925

[8] Raissi, M., Perdikaris, P., and Karniadakis, G. E., “Machine learning of linear differential equations using Gaussian processes,” *Journal of Computational Physics*, Vol. 348, 2017, pp. 683–693.

doi: 10.1016/j.jcp.2017.07.050

[9] Zhang, J., Ma, W., “Data-driven discovery of governing equations for fluid dynamics based on molecular simulation,” *Journal of Fluid Mechanics*, Vol. 892, 2020, pp. A5.

doi: 10.1017/jfm.2020.184

[10] Wu, J., Xiao, H., Sun, R., and Wang, Q., “Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned,” *Journal of Fluid Mechanics*, Vol. 869, 2019, pp. 553-586.

doi: 10.1017/jfm.2019.205

[11] Pawar, S., San, O., Aksoylu, B., Rasheed, A., and Kvamsdal, T., “Physics guided machine learning using simplified theories,” *Physics of Fluids*, Vol. 33, No. 1, 2021, pp. 011701.

doi: 10.1063/5.0038929

[12] Wang, J. X., Wu, J. L., and Xiao, H., “Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data,” *Physical Review Fluids*, Vol. 2, No. 3, 2017, pp. 034603.

doi: 10.1103/physrevfluids.2.034603

[13] Zhu, L., Zhang, W., Kou, J., and Liu, Y., “Machine learning methods for turbulence modeling in subsonic flows around airfoils,” *Physics of Fluids*, Vol. 31, No. 1, 2019, pp. 015105.

doi: 10.1063/1.5061693

[14] Xie, C., Wang, J., Li, H., Wan, M., and Chen, S., “Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence,” *Physics of Fluids*, Vol. 31, No. 8, 2019, pp. 085112.

doi: 10.1063/1.5110788

[15] Jiang, C., Vinuesa, R., Chen, R., Mi, J., Laima, S., and Li, H., “An interpretable framework of data-driven turbulence modeling using deep neural networks,” *Physics of Fluids*, Vol. 33, No. 5, 2021, pp. 055133.

doi: 10.1063/5.0048909

[16] Maulik, R., San, O., Jacob, J. D., and Christopher, C., “Sub-grid scale model classification and blending through deep learning,” *Journal of Fluid Mechanics*, Vol. 870, 2019, pp. 784–812.

doi: 10.1017/jfm.2019.254

[17] Huang, J., Liu, H., and Cai, W., “Online in situ prediction of 3-D flame evolution from its history 2-D projections via deep learning,” *Journal of Fluid Mechanics*, Vol. 875, 2019.

doi: 10.1017/jfm.2019.545

[18] Geneva, N., Zabaras, N., “Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks,” *Journal of Computational Physics*, Vol. 383, 2019, pp. 125–147.

doi: 10.1016/j.jcp.2019.01.021

[19] White, C., Ushizima, D., and Farhat, C., “Neural networks predict fluid dynamics solutions from tiny datasets,” arXiv:1902.00091, Stanford University, Stanford, California, 2019 (unpublished).

[20] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O., “Understanding deep learning requires rethinking generalization,” *Communications of the ACM*, Vol. 64, No. 3, 2021, pp. 107-115.

doi: 10.1145/3446776

[21] V. Nagarajan, J. Z. Kolter, “Uniform convergence may be unable to explain generalization in deep learning,” arXiv:1902.04742, Carnegie Mellon University, Pittsburgh, PA, 2019 (unpublished).

[22] Negrea, J., Dziugaite, G. K., and Roy, D., “In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors,” *In International Conference on Machine Learning*, 2020, pp. 7263–7272.

[23] Adrian, R. J., “Particle-imaging techniques for experimental fluid mechanics,” *Annual Review of Fluid Mechanics*, Vol. 23, No. 1, 1991, pp. 261–304.

doi: 10.1146/annurev.fl.23.010191.001401

[24] Cai, S., Liang, J., Gao, Q., Xu, C., and Wei, R., “Particle image velocimetry based on a deep learning motion estimator,” *IEEE Transactions on Instrumentation and Measurement*, Vol. 69, No. 6, 2019, pp. 3538–3554.

doi: 10.1109/tim.2019.2932649

[25] Jin, X., Cheng, P., Chen, W. L., and 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," *Physics of Fluids*, Vol. 30, No. 4, 2018, pp. 047105.

doi: 10.1063/1.5024595

[26] Sekar, V., Jiang, Q., Shu, C., and Khoo, B. C., "Fast flow field prediction over airfoils using deep learning approach," *Physics of Fluids*, Vol. 31, No. 5, 2019, pp. 057103.

doi: 10.1063/1.5094943

[27] Bukka, S. R., Gupta, R., Magee, A. R., and Jaiman, R. K. "Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models," *Physics of Fluids*, Vol. 33, No. 1, 2021, pp. 013601.

doi: 10.1063/5.0030137

[28] Deng, Z., He, C., Liu, Y., and Kim, K. C., "Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework," *Physics of Fluids*, Vol. 31, No. 12, 2019, pp. 125111.

doi: 10.1063/1.5127031

[29] Fukami K, Fukagata K, and Taira K., "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics*, Vol. 870, 2019, pp. 106-120.

doi: 10.1017/jfm.2019.238

[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, 2020, pp. 025105.

doi: 10.1063/1.5140772

[31] Dong, C., Loy, C. C., He, K., and Tang, X., "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 38, No. 2, 2015, pp. 295–307.

doi: 10.1109/TPAMI.2015.2439281

[32] Kim, J., Lee, J. K., and Lee, K. M. "Accurate image super-resolution using very deep convolutional networks," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[33] Lai, W. S., Huang, J. B., Ahuja, N., and Yang, M. S., "Deep laplacian pyramid networks for fast and accurate super-resolution," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632.

[34] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W., "Photo-realistic single image super-resolution using a generative adversarial network," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.

[35] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M., "Enhanced deep residual networks for single image super-resolution," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[36] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," *In International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015, pp. 234–241.

[37] Hu, X., Naiel, M. A., Wong, A., Lamm, M., and Fieguth, P., "RUNet: A robust UNet architecture for image super-resolution," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[38] Dataset: M. Raissi, A. Yazdani, G. E. Karniadakis (2020). "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,"

[39] Keys, R., "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, Vol. 29, No. 6, 1981, 1153–1160.

doi: 10.1109/tassp.1981.1163711

[40] Huang, Z., Huang Z., and Fan, H., "Influence of C groove on energy performance and noise source of a NACA0009 hydrofoil with tip clearance," *Renewable Energy*, Vol. 159, 2020, pp. 726–735.

doi: 10.1016/j.renene.2020.05.159

[41] Guo, Q., Zhou, L., and Wang, Z., "Numerical evaluation of the clearance geometries effect on the flow field and performance of a hydrofoil," *Renewable Energy*, Vol. 99, 2016, pp. 390–397.

doi: 10.1016/j.renene.2016.06.064