



Review

Deep learning for fluid velocity field estimation: A review

Changdong Yu ^a, Xiaojun Bi ^{b,*}, Yiwei Fan ^c^a College of Information and Communication Engineering, Harbin Engineering University, Harbin, 150001, China^b College of Information and Engineering, Minzu University of China, Beijing, 100081, China^c College of Shipbuilding Engineering, Harbin Engineering University, Harbin, 150001, China

ARTICLE INFO

Keywords:

Deep learning

PIV

Fluid motion estimation

Velocity field reconstruction

Optical flow

ABSTRACT

Deep learning technique, has made tremendous progress in fluid mechanics in recent years, because of its mighty feature extraction capacity from complicated and massive fluid data. Motion estimation and analysis of fluid data is one of the significant research topics in fluid mechanics. In this paper, we provide a comprehensive review of fluid motion (i.e., velocity field) estimation methods based on deep learning. Essentially, the fluid super-resolution (SR) reconstruction task can also be regarded as an velocity field estimation from low resolution to high resolution. To this end, we mainly give a review on two topics: fluid motion estimation and later velocity field super-resolution reconstruction. Specifically, we first introduce the basic principle and component of deep learning methods. We then review and analyze deep learning based methods on fluid motion estimation. Note we mainly investigate the commonly used fluid motion estimation approach here, particle image velocimetry (PIV) algorithm, which extract velocity field from successive particle images pair in a non-contact manner. In addition, SR reconstruction methods for velocity fields based on deep learning technique are also reviewed. Eventually, we give a discussion and possible routes for the future research works. To our knowledge, this paper are the first to give a review of deep learning-based approaches for fluid velocity field estimation.

1. Introduction

The acquisition of the global velocity field (i.e., motion fields) is of great significance to research the structure of complicated fluid flows in the fluid mechanics. As a widely used velocity field estimation technique, particle image velocimetry (PIV) (Nguyen et al., 2012; Khalid et al., 2019; Fleit and Baranya, 2019), which can extract the velocity vectors of the whole field from successive particle images. Fig. 1 shows the working principle of PIV technique and is described as follows (Adrian and Westerweel, 2011). First, the small tracer particles are cast into the fluid medium to be measured, then the tracer particles in the area to be measured are illuminated with a uniform sheet of light. The illuminated tracer particles are continuously photographed by a camera, so as to obtain successive particle image pair. Finally, the corresponding algorithms are adopted to calculate the particle image pair to obtain the velocity field. Hence, how to obtain high resolution velocity field is always a core problem in the PIV estimation community.

There are two main traditional PIV estimation algorithms: cross-correlation algorithm and optical flow algorithm. Cross-correlation algorithm (Adrian, 2007) obtains the displacement vector by querying

the displacement correlation peak value of the corresponding interrogation window between the image pair. Namely, the estimated displacement vector within each interrogation window is regarded as the average velocity in that window. As a result, this correlation-based approach provides a sparse (i.e., low-resolution) velocity field. Therefore, many modifications have been made to enhance the computational accuracy and efficiency of the cross-correlation algorithm (Wereley et al., 2002; Scarano, 2001). Furthermore, many advanced post-stage velocity field processing methods, such as outlier detection (Westerweel and Scarano, 2005; Wang et al., 2018a) and spline interpolation (Astarita and Cardone, 2005; Cholevski, 2007), are put forward to further reduce error. The cross-correlation algorithm has been relatively mature after forty years of development, and it has achieved good performance in the International PIV Challenges (Stanislas et al., 2005, 2008). Essentially, the cross-correlation algorithm still cannot provide a dense velocity field at the pixel level. Optical flow algorithm (Horn and Schunck, 1981) has been a popular research direction in the computer vision community that estimates the motion field between image pairs by solving the optimal value of an objective function. In contrast to the cross-correlation approach, optical flow method can provide a dense velocity field for the whole image. Additionally, optical

* Corresponding author.

E-mail address: bixiaojun@hrbeu.edu.cn (X. Bi).

Terminology	
PIV	Particle image velocimetry
CFD	Computational fluid dynamics
CV	Computer vision
NLP	Natural language processing
AI	Artificial intelligence
ML	Machine learning
DL	Deep learning
DNN	Deep neural networks
FNN	Fully-connected neural network
DBN	Deep belief network
CNN	Convolutional neural network
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
MLP	Multi-layer perceptron, each layer of neurons is fully connected to the next layer
Receptive field	The mapping area size of the pixel of the output feature map on the original map
Feedforward network	One-way propagation, no connection and feedback between layers
SR	Super-resolution
HR	High-resolution
LR	Low-resolution
PINN	Physics-informed neural network
GAN	Generative adversarial net

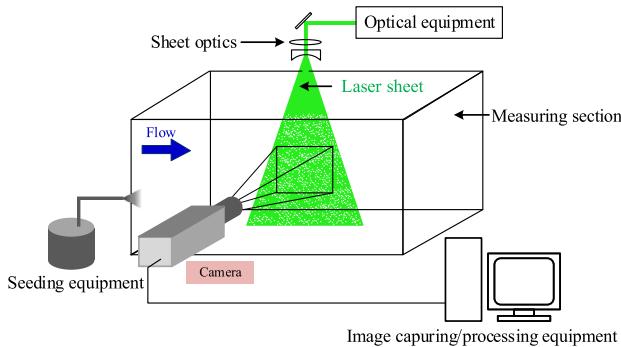


Fig. 1. Schematic diagram of the basic principle of PIV technique.

flow methods are easily embedded with prior physical constraints to make the algorithm more suitable for different fluid scenarios (Heitz et al., 2010). Therefore, optical flow algorithm has attracted many researchers in fluid mechanics to improve it and apply it to fluid motion estimation (Corpetti et al., 2006; Kapulla et al., 2011; Hua et al., 2014; Zhong et al., 2017). Although optical flow algorithms have been widely used in various fluid scenarios, there are still two obvious problems. First, the optical flow algorithm is time-consuming in the process of variational optimization. In addition, the optical flow method is sensitive to noise, especially changing illumination, which will affect the accuracy of velocity field estimation. Review articles (Heitz et al., 2010; Liu et al., 2015) give a comprehensive description of optical flow methods for fluid motion estimation and comparisons with cross-correlation algorithms. PIV-equipment

Deep learning (DL), as a significant branch of machine learning (ML), has achieved distinguished performance in different fields such as computer vision (CV) (Krizhevsky et al., 2012; Wang and Bi, 2021;

Chen et al., 2022a), speech recognition (Zhang et al., 2018a; Lee et al., 2021), natural language processing (NLP) (Strubell et al., 2019; Otter et al., 2020), etc. Deep learning has a strong ability to learn data features, and can handle fluid mechanics topics with complex nonlinear, high-dimensional, and big data characteristics. Therefore, deep learning techniques have made tremendous progress in the fluid mechanics community in recent years. Since the use of shallow convolutional neural networks for PIV estimation was proposed by Rabault et al. (2017) in 2017, deep learning-based fluid motion estimation methods have continued to emerge (Lee et al., 2017; Cai et al., 2019b). These models achieve better outcomes in terms of precision and inference speed.

In fact, the quest for high-resolution velocity fields has always been the pursuit in fluid mechanics. In addition to PIV estimation techniques to get high-resolution velocity fields, another significant computational fluid dynamics (CFD) technique can also provide high-resolution flow fields. CFD technique can simulate complicated turbulent fields with high fidelity by mesh refinement. However, the experimental simulations are expensive and time-consuming, especially as the number of spatial meshes increases. Meanwhile, for complex high-dimensional flow fields, the process of numerical modeling and solution is also very complicated. Deep learning has a powerful nonlinear function fitting ability, which is able to mine useful feature information from a large amount of fluid data. Therefore, more researchers tend to use deep learning methods to estimate high-resolution flow fields around physical models (Guo et al., 2016; Ling et al., 2016). This task is similar to super-resolution (SR) reconstruction tasks in the computer vision. The reconstruction of high-resolution (HR) velocity field from low-resolution(LR) counterpart can also be regarded as a process of velocity field estimation. It aims at estimating a HR fluid data from a low-resolution counterpart. Therefore, this task is in line with the theme of our article review.

In this article, we give a comprehensive overview of deep learning-based fluid velocity field estimation, which covers two topics: fluid motion estimation (for PIV) and velocity field SR reconstruction. We first introduce the basic theory and knowledge of deep learning, paving the way for the subsequent introduction of research topics. Deep learning based velocity field estimation approaches including optical flow learning and cross-correlation are then investigated and compared. After that, we also review the velocity field SR reconstruction approaches based on deep learning. Note that we not only describe but also analyze the advantages and disadvantages of various algorithms in the review process. Finally, we present our own conclusions and summarize the trends and challenges for future work. To our knowledge, this is the first review paper that comprehensively covers deep learning-based fluid motion estimation and velocity field reconstruction.

2. Principles of deep learning

Deep learning technology (LeCun et al., 2015; Du et al., 2016) has increasingly become a research hotspot and mainstream direction in the artificial intelligence (AI) domain. Deep learning-based architecture is a deep machine learning model that usually contains multiple layers of neural networks. In the learning process, deep learning maps the input data from low-level to high-level to a new feature space, which makes it have the characteristics of hierarchical and distributed abstraction. In this way, complicated nonlinear functions can be well fitted and high-dimensional nonlinear input data can be processed. During the development of deep learning technology, there are many different typical models such as Deep Belief Network (DBN) (Mohamed et al., 2009), Convolutional Neural Network (CNN) (Li et al., 2016), Recurrent Neural Network (RNN) (Pascanu et al., 2013), etc. For the commonly used models in the field of fluid mechanics, we mainly describe the CNN and RNN architectures in this section.

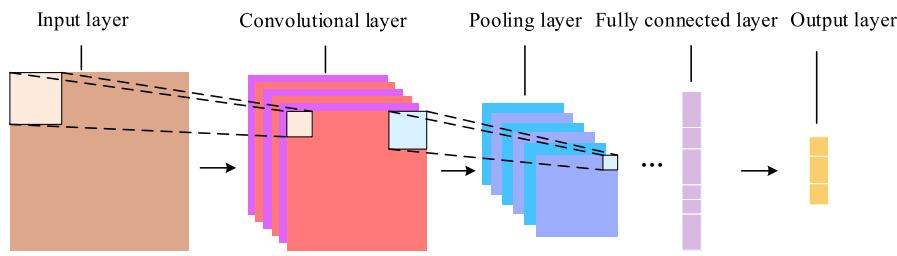


Fig. 2. Schematic diagram of a general CNN structure.

2.1. Convolutional neural network

Different from the fully-connected neural network(FNN), the CNN is a feedforward neural network with convolutional computations that is inspired by the receptive field mechanism of the cerebral cortex in biology. CNN introduces a sparse connection mechanism in the convolutional layer, which effectively avoids the loss of spatial features in the image through the local receptive field. In addition, the weight sharing mechanism and pooling operation are used to overcome the training overfitting problem caused by excessive model parameters. In summary, the three interesting operations of CNN are local receptive fields, weight sharing and pooling layers. Through the above operations, it is possible to build a deeper CNN network and effectively learn more abstract features in images, thereby enhancing the recognition and classification capabilities of the model. The general structure of CNN model (see Fig. 2) mainly consists of input layer, convolutional layer, pooling layers, fully connected layers, output layer and activation function, respectively. In addition, deconvolution layers have important applications in pixel-level upsampling processes such as image SR reconstruction and optical flow estimation tasks. We next describe the basic components of a CNN model in detail.

2.1.1. Convolutional layer

It is well known that the convolutional layer is a key part of the CNN architecture, which plays an important role in the feature extraction process. As depicted in Fig. 3(a), different from the fully connected mode (red arrow in the figure), convolutional layers are connected in a sparse manner, i.e., only a fraction of input neurons x_k are connected (solid connecting lines) to output neurons y_l . Here we assume that the input vector is denoted by $X = (x_1, x_2, \dots, x_k)^T$, and the output is denoted by $Y = (y_1, y_2, \dots, y_l)^T$, then the relationship between the output and input is described as follows:

$$Y = WX + b \quad (1)$$

where the weight matrix W represents the sparse matrix, and b represents the bias parameter. The working principle of the convolution layer is to use the convolution filter to convert the input image into a feature map and then feed it to the next layer. Specifically, as illustrated in Fig. 3(b), the convolution kernel (gray cube in the figure) of size $3 \times 3 \times 4$ shifts horizontally or vertically on the feature map (light green cube in the figure) of size $5 \times 5 \times 4$ with a step size (here $s = 1$). Then, the weight coefficient of the convolution kernel ($3 \times 3 \times 4 = 36$) is multiplied by the element of the corresponding position of the feature map, and the output result of the corresponding position is obtained by summative. The output feature map can be calculated by the following formula:

$$\begin{aligned} W_{\text{out}} &= (W_{\text{in}} - K_{\text{size}}) / S_{\text{size}} + 1 \\ H_{\text{out}} &= (H_{\text{in}} - K_{\text{size}}) / S_{\text{size}} + 1 \\ C_{\text{out}} &= N \end{aligned} \quad (2)$$

where W_{in} and H_{in} denote the width and height dimensions of the input feature map, K_{size} and S_{size} denote the convolution kernel size and convolution step size, respectively. The final output feature map

size is $W_{\text{out}} \times H_{\text{out}} \times C_{\text{out}}$, and the depth dimension C_{out} of the output feature map equals the quantity of convolution kernels (N). For simplicity, only one convolution kernel is used here as an illustration, so the depth of the obtained feature map is 1. In the model structure, the initial convolution mainly extracts low-level detail feature information, such as corners, edges, and colors. With the forward pass of features, subsequent convolutional layers are adopted to extract higher-level and richer feature information. Generally, the size and stride of the convolution kernel is able to set manually, and its parameters are obtained through learning during training process.

2.1.2. Pooling layer

Pooling layer usually follows a convolutional layer, which is used to downsample the feature maps output by the convolutional layer. This operation is able to effectively reduce network parameters, computational cost and training difficulty of the network, and avoid overfitting. In addition, the pooling operation has feature invariance, that is, it no longer cares about the specific location of the feature but cares about the existence of the feature. Pooling operations mainly contain max pooling and average pooling operation. The average pooling is to compute the average value of all elements, and use the obtained average value as the output of the pooling layer. In contrast, the output of the max pooling layer is the maximum value of all elements as the output. It can be seen from Fig. 4 that after a pooling operation with a step size of 2, the image becomes 1/4 of the original image. Through the max pooling operation, the largest representative feature of the feature map is able to be extracted, while reducing the amount of data by 75%. Therefore, the max pooling operation is widely used in the CNN model.

2.1.3. Activation function

The activation function is mainly adopted in the convolutional layer or the fully connected layer, and its main purpose is to perform nonlinear fitting on the output features, thereby increasing the nonlinear feature representation ability of the network. Different kinds of activation functions have emerged to enhance the ability of the model in a targeted manner. For instance, the Sigmoid function, Tanh function, ReLU function (Glorot et al., 2011) and Softmax function are common activation functions in convolutional neural networks (see Fig. 5).

First, the definition formula of Sigmoid is as follows:

$$s(x) = 1 / (1 + e^{-x}) \quad (3)$$

The function image of Sigmoid function is shown in Fig. 5(a). The Sigmoid function is a continuous monotonous curve that maps the input value to the interval from 0 to 1, so it is often used in the probability prediction problem of binary classification. In addition, the Sigmoid function has the problem of easy gradient disappearance during training. For the Tanh activation function, it is depicted in Fig. 5(b), and its description formula is defined as follows:

$$t(x) = (1 - e^{-2x}) (1 + e^{-2x}) \quad (4)$$

The Tanh function is a hyperbolic tangent function, which converges faster than the Sigmoid function. However, both functions have the

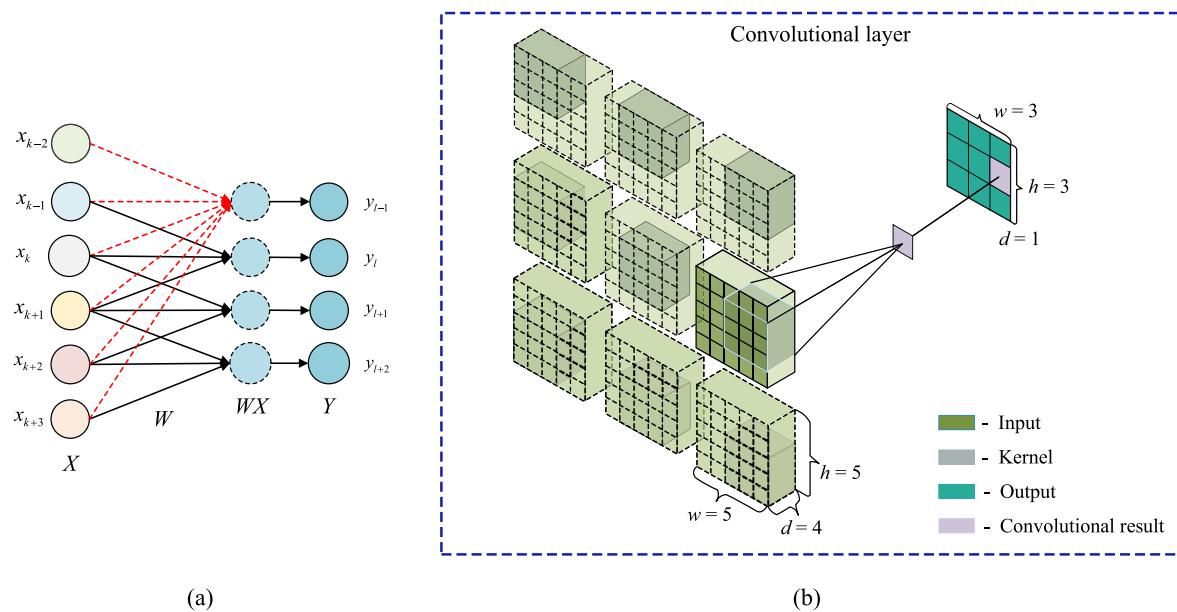


Fig. 3. Schematic diagram of connections between convolutional layers (a) and convolution operation in CNN (b).

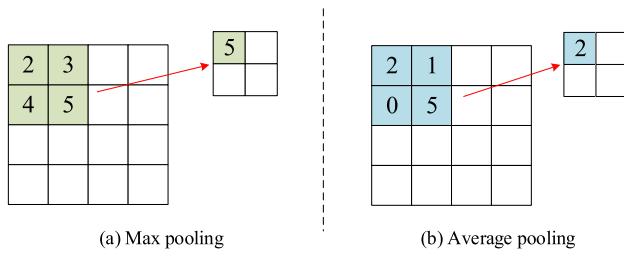


Fig. 4. Schematic diagram of max pooling (a) and average pooling (b).

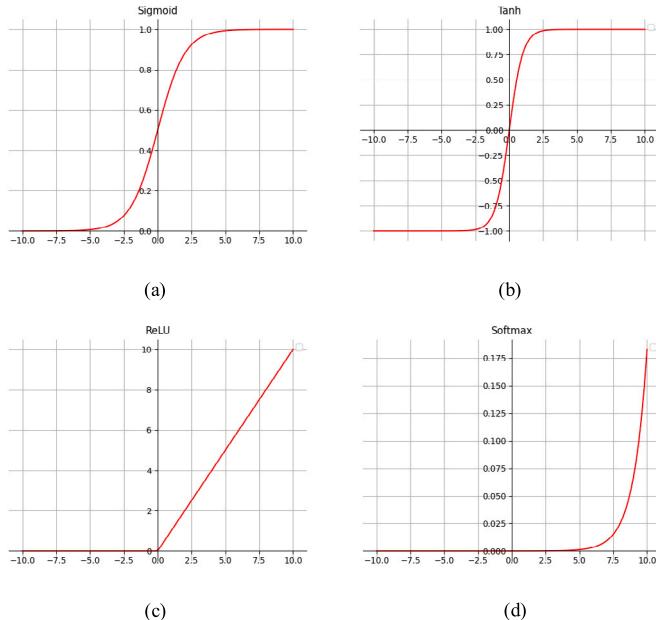


Fig. 5. Schematic diagram of different activation functions: Sigmoid function (a), Tanh function (b), ReLU function (c) and Softmax function (d).

problem that the gradient is easy to disappear. Compared with the first two activation functions, the application of the ReLU function is more

popular, and its description formula is described as follows:

$$r(x) = \max(x, 0) \quad (5)$$

Fig. 5(c) represents that the value of function is 0 when x is less than 0, and it is a one-dimensional linear function when x is greater than 0, keeping the gradient at a constant value. This method effectively solves the gradient disappearance of Sigmoid and Tanh functions. At the same time, it also has the advantages of strong sparsity and easier convergence. The function image of Softmax function is shown in Fig. 5(d), which is described as follows:

$$m(x_i) = \exp(x_i) / \sum_j \exp(x_j) \quad (6)$$

where i denotes the label of the classification, and $m(x_i)$ denotes the probability of one of the categories. The Softmax function is actually a normalization function, which is often used in the output layer of the feedforward neural network, mapping the output value to a probability between 0 and 1, and is mainly used as an activation function in multi-classification problems.

2.1.4. Fully connected layer

In the CNN structure, after highly abstract modeling of the two-dimensional information of the input image through the convolution and pooling layers, it is necessary to further reduce the dimensionality of the feature map using the fully connected layer. In this way, deep two-dimensional features are mapped to a one-dimensional sample space. Essentially, the fully connected layer is also a convolution calculation, which performs convolution operations on the input multi-channel feature maps and accumulates the convolution results (see Fig. 6). During the convolution operation, it is necessary to expand the calculation feature matrix in front of the fully connected layer into a one-dimensional vector, which is adopted as the input data for the fully connected layer calculation. The essence of a fully connected layer is a linear function, which cannot effectively model nonlinear features, so multiple fully connected layers are usually used in neural networks. Fig. 6 shows the schematic diagram of the fully connected operation. Here, two fully connected layers are designed to flatten the input feature map from the pooling layer into a one-dimensional vector, and then complete a feature weighting through the first fully connected layer ((FC1) layer and ReLU activation. Then the result of the first feature weighting is passed through the FC2 layer and

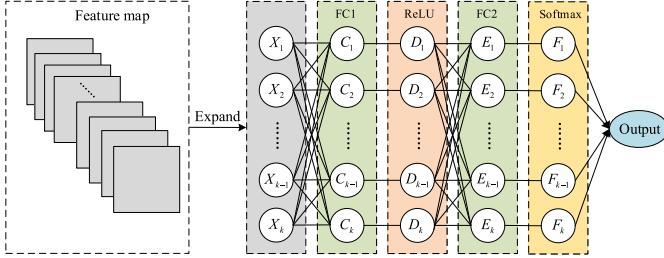


Fig. 6. Schematic diagram of the fully connected operation.

Softmax activation function to complete the second feature weighting, and finally the prediction result is output.

Taking the input to the FC1 layer as an example, the operation process is shown in the following formula:

$$\begin{aligned} C_1 &= X_1 \times H_{11} + X_2 \times H_{12} + \dots + X_k \times H_{1k} + b_1 \\ C_2 &= X_1 \times H_{21} + X_2 \times H_{22} + \dots + X_k \times H_{2k} + b_2 \\ &\dots \\ C_k &= X_1 \times H_{k1} + X_2 \times H_{k2} + \dots + X_k \times H_{kk} + b_k \end{aligned} \quad (7)$$

We convert Eq. (7) into a matrix, which is defined as follows:

$$\begin{pmatrix} C_1 \\ \dots \\ C_k \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & \dots & H_{1k} \\ \dots & \dots & \dots & \dots \\ H_{k1} & H_{1k} & \dots & H_{kk} \end{pmatrix} * \begin{pmatrix} X_1 \\ \dots \\ X_k \end{pmatrix} + \begin{pmatrix} b_1 \\ \dots \\ b_k \end{pmatrix} \quad (8)$$

where $X_1 \sim X_k$ and $C_1 \sim C_k$ denote the input and output, $H_{11} \sim H_{kk}$ represents the weight of the fully connected layer, and $b_1 \sim b_k$ represents the bias of the fully connected layer. Eq. (8) further proves that the calculation method of the fully connected layer is a convolution calculation. In addition, although the fully connected layer generally only accounts for 10%–20% of the model, its parameter volume accounts for more than 80% of the total.

2.1.5. Deconvolution layer

The deconvolution (transposed-convolutional) (Zeiler et al., 2010) is a variant of the convolution operation that essentially inverts the forward and backward computations of convolution. Fig. 7 presents the calculation process of a 3×3 convolution kernel moving on a 4×4 feature map with a stride of 1. In the forward pass of the model, the convolution operation reduces the size of the feature map layer by layer. Conversely, the deconvolution operation increases the feature map size layer by layer until it returns to the original image size. Therefore, the deconvolution operation is similar to an upsampling operation. However, unlike the traditional upsampling method (e.g., bilinear upsampling), the convolution kernel parameters of deconvolution can be learned from training, which allows the model to learn better feature representation capabilities in the data.

For the fluid velocity field estimation task, it is essentially a process of pursuing a high-resolution velocity field. For PIV estimation, the model extracts feature from low-dimensional particle image pairs and finally upsamples to a high-resolution velocity field. Similarly, the SR reconstruction of the flow field is also a process of estimating the HR velocity field from the LR velocity field data. Therefore, the deconvolution operation is of great significance in the fluid velocity field estimation task.

2.1.6. Parameter optimization

Once the neural network model is designed and determined, its parameters w need to be trained and optimized on the corresponding dataset. Assuming that the sample of the training set is p and the corresponding label is q , the training process of model learning is equal to addressing an optimization problem,

$$w = \arg \min_w L(F(p, w)), q \quad (9)$$

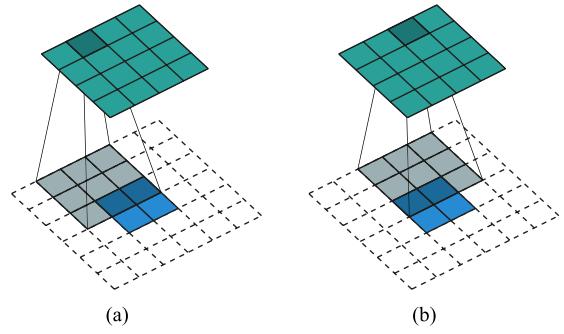


Fig. 7. Schematic diagram of deconvolution computation process.
Source: Reprinted from Dumoulin and Visin (2016).

where F denotes the network function, L denotes the loss function. Once the training process is completed, w is fixed to the optimal value. Since 2012, CNNs have attracted much attention from researchers, and many classic model structures have emerged, such as LeNet (Le-Cun et al., 1998), Alex-Net (Krizhevsky et al., 2012), ZFNet (Zeiler and Fergus, 2014), VGGNet (Simonyan and Zisserman, 2014) and GoogleNet (Szegedy et al., 2015). Nowadays, CNNs are still an active topic applied to many different tasks.

2.2. Recurrent neural network

When feedforward networks such as convolutional neural networks process input data, adjacent input samples are independent of each other in the process of feature learning. Yet, when the input data is sequential information with dependencies, the CNN model cannot effectively learn the temporal correlation features between adjacent samples. Instead, RNN introduce self-recurrent operations in their own neurons, so that the output of each time is not only related to the input at present, but related to the input at previous timestamps. Fig. 8(a) shows schematic diagram of the internal structure of an RNN neuron. It can be seen that it learns an implicit representation of the input sequence through an internal recurrent structure. When given an input sequence $x_t = (x_1, x_2, \dots, x_n)$, the recurrent update process of RNN hidden neurons is as follows:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (10)$$

where W_{xh} is the weight input to the variable x_t at this moment, W_{hh} is the neuron state h_{t-1} at the previous moment as the weight input at this moment. b_h is the bias value, and f is the activation function, which usually adopts the Tanh function. Further, the network output y_t at the current time t can be obtained by the following iterative formula:

$$y_t = W_{hy}h_t + b_y \quad (11)$$

where W_{hy} is the network weight when h_t is used as input, and b_y is the corresponding bias value.

Although RNN can handle time series data, there is still a serious problem about gradient vanishing in the process of training. As a result, RNN can only have short-term memory and cannot effectively process long-term time series data. In order to address these issues, some excellent variants appear immediately, e.g., representative Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014). These networks add information storage units to the structure of the recurrent neural network, which makes the network have stronger memory capabilities than RNNs.

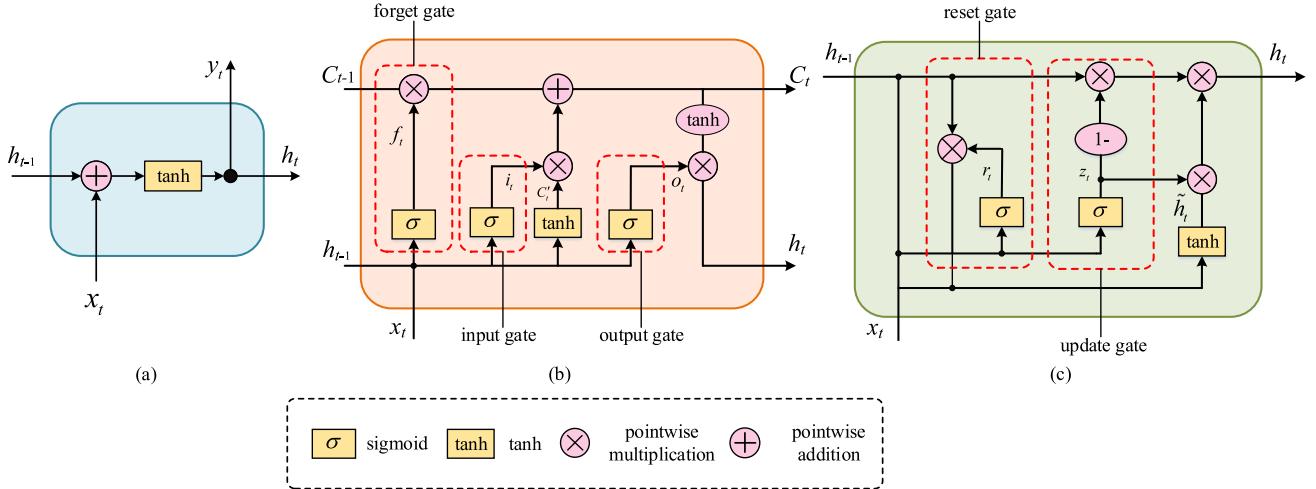


Fig. 8. Schematic diagram of the RNN structure (a), the LSTM structure (b) and GRU structure (c). Where , represents the input and C_t denotes the long-term state of the unit.

2.2.1. LSTM

Different from the basic structure of RNN, LSTM uses three “gates” to constrain the state and output at different times, i.e., input gate, output gate and forget gate. LSTM combines short-term memory with long-term memory through a gate structure, which can effectively alleviate the problem of gradient disappearance. The gate structure is a fully connected layer that uses a bitwise multiplication operation, and its activation function is a sigmoid function. The sigmoid function will output a value between 0 and 1 to represent the amount of sample information that can transmit the gate at the current time. 0 represents that no feature information can be transmitted, and 1 represents that all feature information can be transmitted. The gate structure can be shown in Fig. 8(b), and its specific calculation process is described by the following formula, which can be divided into three parts:

(1) Forget gate is used to forget information, discarding some unwanted information from the long-term state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (12)$$

where W_f denotes the weight matrix of the forget gate, and b_f denotes the bias value. The input of the forget gate f_t is h_{t-1} and x_t , and the output value is between 0 and 1, which is multiplied by each corresponding position element in the previous long-term state C_{t-1} . The operation of multiplying with 0 here means forgetting the information, and 1 The operation of multiplying represents receiving this information.

(2) Input gate is used to determine new memory information and store new information in the long-term state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (13)$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (14)$$

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \quad (15)$$

where W_i and W_c still represent the corresponding weight matrix, b_i and b_c represent the bias value. This part can be divided into three operations. First, the input x_t at the current moment and the hidden state h_{t-1} at the previous moment are adopted as the input of the input gate, and the output value i_t is controlled within the range of 0 to 1 through the Sigmoid activation function. The second operation uses the Tanh activation function to create a new candidate state C'_t , as described in Eq. (14). Finally, Eq. (15) is used to add the candidate state C'_t to the current long-term state C_t .

(3) Finally, the state information h_t is output, and its calculation process is described as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (16)$$

$$h_t = o_t \times \tanh(C_t) \quad (17)$$

The input x_t at the current time and the previous hidden state h_{t-1} are adopted as the input of the current Sigmoid activation function to calculate the output o_t . The output o_t is then multiplied by the long-term state C_t to get the current hidden state h_t .

In summary, the forget gate constrains the amount of sample information that can be passed from a previous time to the current cell state. The input gate determines the amount of sample information that the input can save to the current unit. The output gate determines how much information a cell state can output to the current state output value. Afterwards, LSTM was improved and generalized for various applications (Mirza and Cossan, 2018; Koutnik et al., 2014).

2.2.2. GRU

GRU is improved on the basis of LSTM, and it has better effect than LSTM while simplifying the structure of LSTM. Compared with the three-gate structure of the LSTM, GRU simplifies it to two gates: update gate and reset gate. The former is designed to control the amount of information transmitted from the previous moment to the current state, and the latter is adopted to control the amount of information forgotten at the previous moment. As illustrated in Fig. 8(c), x_t is the input data, h_t is the GRU output state. r_t and z_t are the reset gate and the update gate, respectively. These two gates together handle the computation from state h_{t-1} to h_t , and the specific formula is described as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (18)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (19)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (20)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (21)$$

where W_z , W_r and W denote the weight matrices of the update gate, reset gate and hidden state respectively. Based on GRU, some researchers have improved and further improved its performance (Zhao et al., 2017; Che et al., 2018).

3. Review of fluid motion estimation

The fluid motion estimation methods based on deep learning are developed along with deep learning technology. The conceptual study of deep learning for fluid motion estimation could date back to 1990s (Teo et al., 1991; Cenedese et al., 1992; Hassan and Philip, 1997; Grant and Pan, 1997). However, at the time neural networks with only a few layers were adopted to process only part of the PIV tasks such as feature extraction and process optimization. In fact, early work just

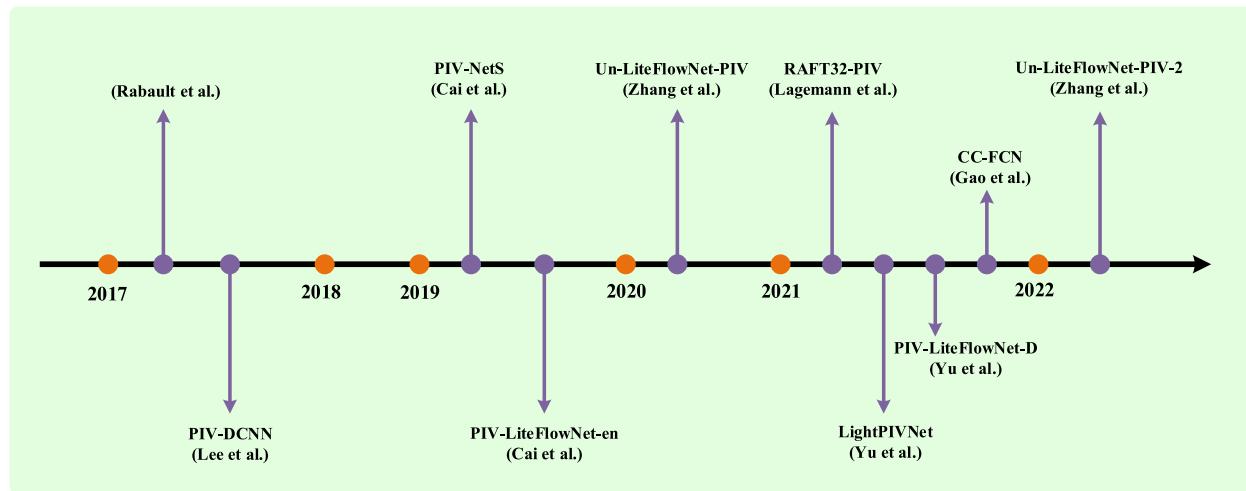


Fig. 9. A temporal overview of a deep learning-based fluid motion estimation models (for PIV).

used an artificial neural network for pattern recognition of particle trajectories (Hassan and Philip, 1997; Grant and Pan, 1997). On the other hand, limited by the computing power of hardware systems, deep learning technology also fell into a low period of development at that time. But this has laid a certain theoretical foundation for the subsequent study of fluid motion estimation based on deep learning.

Krizhevsky et al. (2012) put forward deep neural networks for image classification tasks in 2012 and achieved the best performance. Deep learning technology has once again attracted the interest and research of researchers in various fields. In 2017, Rabault et al. (2017) put forward a convolutional neural network (CNN) that truly applies deep learning methods to estimate velocity field. Likewise in 2017, Kutz (2017) gave a short review about deep learning methods in fluid dynamics and point out that it is just a matter of time before deep learning achieves outstanding performance in the complex turbulence modeling. Meanwhile, they also suggest building open challenge datasets in the fluid community similar to ImageNet (Deng et al., 2009) to fairly compare approaches.

Since then, deep learning methods for fluid velocity field estimation have gradually emerged. Fig. 9 presents the development process of fluid velocity field estimation methods based on deep learning in recent years. The design of these network models is inseparable from the inspiration of traditional algorithms. The ideas of optical flow learning and cross-correlation learning have important guiding significance for fluid motion estimation. Next, we first review deep optical flow learning and cross-correlation learning methods for velocity field estimation. Subsequently, we give consideration and analysis on the applicability of deep learning methods.

3.1. Optical learning

3.1.1. A short overview of deep optical flow learning

Optical flow estimation has always been a research hotspot in the field of computer vision (CV), which is mainly used for motion estimation and analysis of objects. With the maturity of deep learning, the deep optical flow learning method has completely surpassed the performance of traditional optical flow algorithms (Dosovitskiy et al., 2015; Ilg et al., 2017). Just as the traditional optical flow algorithm has aroused the interest of experimental mechanics researchers, the deep learning optical flow approach has also attracted the attention of researchers. The research on velocity field estimation based on deep optical flow learning has progressed along with the development of optical flow learning models. Fig. 10 presents the evolution of representative deep learning optical flow networks. According to the literature review (Zhai et al., 2021) for optical flow approaches, we can divide

the existing optical flow models into three types: U-Net, spatial pyramid network and recurrent iteration network.

U-Net

Dosovitskiy et al. (2015) first propose to use CNNs to estimate optical flow fields in 2015, which presents two networks namely FlowNetS and FlowNetC. Both architectures are based on the encoder-decoder structure (i.e., U-Net), which consists of encoder and decoder parts. Fig. 11 exhibits the architecture of FlowNetS and its encoder part consists of consecutive convolution layers to extract deep feature maps. Because the stride of some layers is 2, the size of the feature map will decrease as the network develops deeper. The decoder part of FlowNetS is composed of a series of deconvolution layers that recover the size of feature maps. Finally, the model outputs fine optical flow field at the 1/4 resolution of the original input. Different from the structure of FlowNetS, the input of FlowNetC is divided into two branches, and each branch consists of 3 convolutional layers to extract features of image pairs. The outputs of these two branches are then fed to the correlation layers to compute the feature matching cost. The decoder part of FlowNetS and FlowNetC is the same. After that, based on this U-Net architecture, many variants appeared successively (Ilg et al., 2017; Yu et al., 2016; Lai et al., 2017). For example, FlowNet2 (Ilg et al., 2017) model is putforward to further enhance the accuracy for optical flow estimation. However, the improvement of the accuracy is obtained by stacking sub-networks, which results in a large and time-consuming model. Meanwhile, the structure of U-Net is regarded as a general approximation method and lacks the exploration of optical flow theory.

Spatial pyramid

The optical flow model SPyNet based on spatial pyramid was firstly proposed by Ranjan and Black (2017). The SPyNet is a coarse-to-fine spatial pyramid architecture to predict optical flow at different resolutions. Although the SPyNet model is smaller, the estimation accuracy is not as good as FlowNet2. The SPyNet estimates large motion on coarse layers and warps the second image towards the first using the upsampled flow from the previous level. Therefore, only the remaining flow for each level needs to be calculated. In order to further deal with the large displacement and reduce the model parameters, based on (Ranjan and Black, 2017), two representative pyramid structures PWCNet (Sun et al., 2018) and LiteFlowNet (Hui et al., 2018a) are proposed simultaneously in 2018. This architecture adopts feature warping operation instead of image warping under different scale. Meanwhile, it also adopts cost volume layer to compute the matching cost on each pyramidal layer. The two networks achieve excellent results on different optical flow datasets. In the following period, many optical flow networks (Wang et al., 2018b; Liu et al., 2019; Yang and Ramanan,

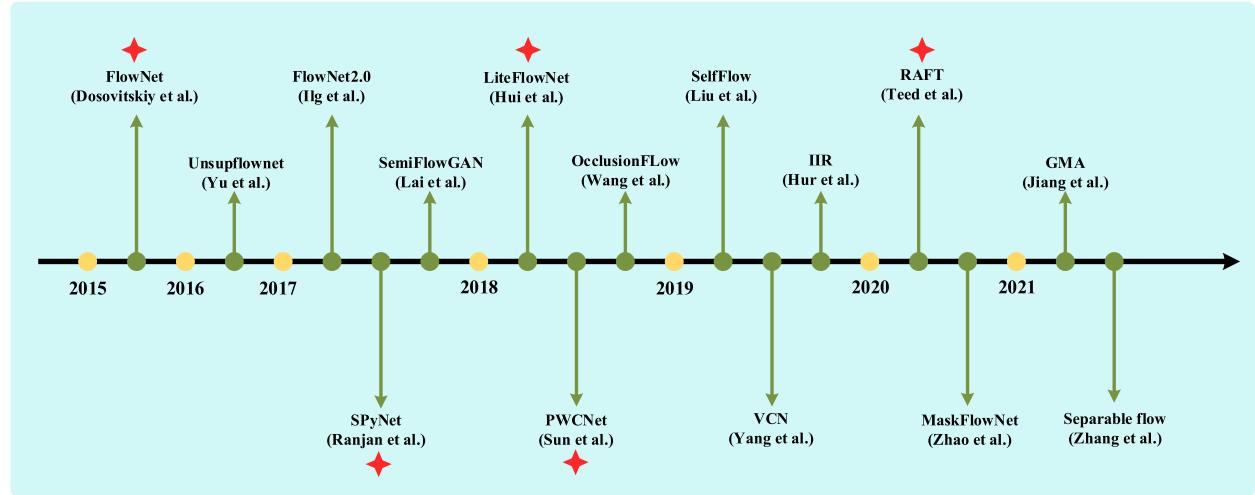


Fig. 10. A temporal overview of deep learning optical flow models, where the pink stars represent milestone approaches.

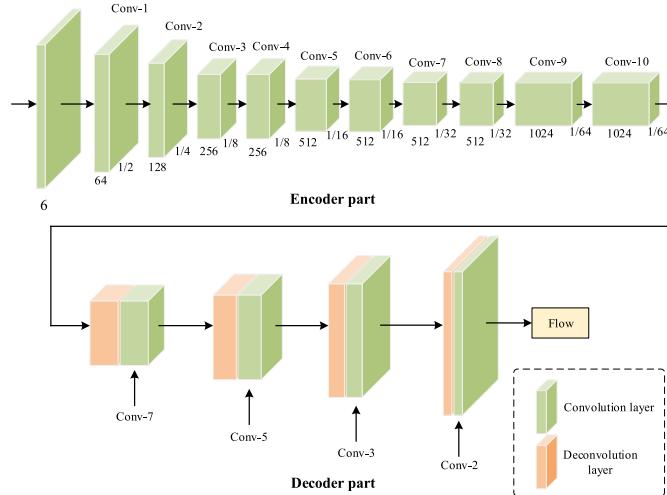


Fig. 11. The architecture of FlowNetS (Dosovitskiy et al., 2015).

2019; Hur and Roth, 2019) are still improved based on this spatial pyramid structure.

Recurrent iteration

The coarse-to-fine refine strategy of spatial pyramid structure effectively improves the estimation accuracy. But the refinement of optical flow is limited by the number of pyramid levels. Some related works adopted iterative refinement methods to enhance results on optical flow. In 2020, a novel optical flow architecture named recurrent all-pairs field transforms (RAFT) (Teed and Deng, 2020) is proposed and becomes a new milestone approach. As shown in Fig. 12, RAFT first extracts the features of the image pair using the feature encoder. The context encoder has the same structure as the feature encoder to extract the context features of the first image, which provides more semantic features for the subsequent flow inference to improve accuracy. Then, a 4D($W \times H \times W \times H$) correlation layer that builds a correlation volume to compute the matching of the corresponding feature vectors. Finally, a recurrent update operator based Conv-GRU block is utilized to update and refine optical flow. It is worth mentioning that the number of iterations of the update operator of RAFT can be selected in training and testing until the model evaluates to a satisfactory effect. The design idea also imitates the optimization process of the traditional variational algorithm. RAFT reaches optimal estimation performance

with high efficiency in running time, training speed and model size. It has become a new milestone architecture, which has attracted many researchers to improve and enhance it (Jiang et al., 2021; Zhang et al., 2021).

3.1.2. Optical flow learning for velocity field estimation Supervised learning

Since traditional optical flow algorithms play an important role in fluid motion estimation, it is conceivable that deep optical flow learning also has attractive prospect. Estimating fluid motion fields from particle image pairs can also be considered as an image processing issue in the CV area. Cai et al. (2019b) first modified the optical flow model FlowNetS (Dosovitskiy et al., 2015) for velocity field estimation. The original FlowNetS use the interpolation approaches to improve the flow field into full resolution. However, it finds that the interpolation method ignores the small-scale vortex structure information of the flow field, which is critical for complex flow fields. Hence, two more deconvolutional layers are integrated into the final stage of the model. In this way, a dense high-resolution velocity field (pixel-level) can be obtained. In addition, considering the supervised learning method, the corresponding dataset needs to be established to train the models. The generated PIV dataset contains various flow fields and has strong generalization. Up to our knowledge, this dataset is a public PIV dataset which can be a benchmark for comparing different deep learning algorithms. The main process of generating the PIV dataset is as follows. First, the particle image generator (PIG) is adopted to generate particle image (Raffel et al., 2007) and a particle can be defined by a two-dimensional Gaussian function:

$$I_p(x, y) = I_0 \exp \left[\frac{-(x - x_0)^2 - (y - y_0)^2}{(1/8)d_p^2} \right], \quad (22)$$

where I_0 , d_p and (x_0, y_0) represent the intensity, diameter and center position of the particle, respectively. Additionally, the particle seeding density determines the particle quantity in the observed domain. Therefore, different control parameters can determine a unique particle image. The generated image then moves symmetrically following the flow motion to get an image pair. For different flow patterns, it can be extracted in different ways such as computational fluid dynamics (CFD) and open source addresses (e.g., 2D DNS-turbulence flow Carlier, 2005, surface quasi-geostrophic (SQG) model of sea flow Resseguier et al., 2017 and the Johns Hopkins Turbulence Databases (JHTBD) Li et al., 2008). The components of the dataset are illustrated in Table 1.

With the advent of more favorable optical flow models, Cai et al. (2019a) then put forward the PIV-LiteFlowNet-en model that is based

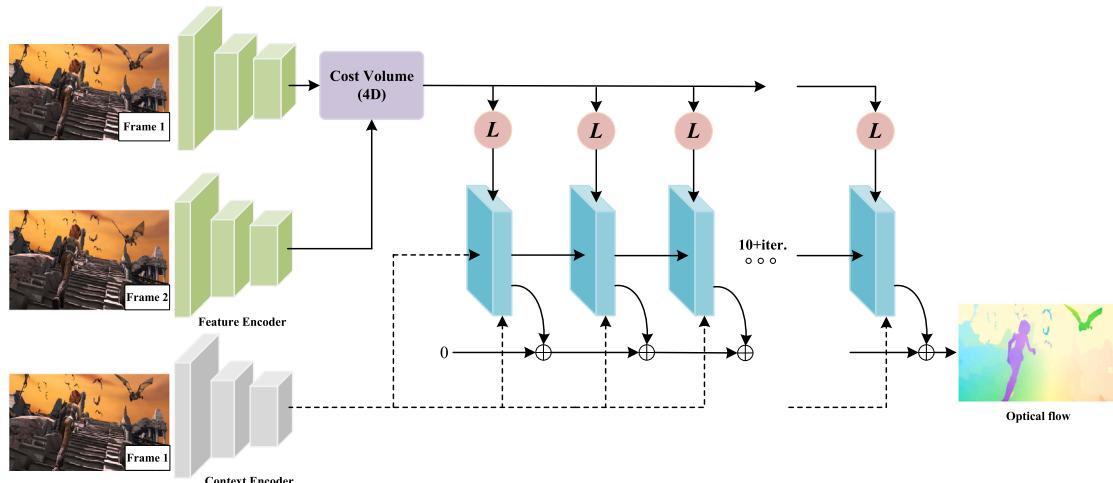


Fig. 12. The architecture of RAFT (Teed and Deng, 2020). It consists of 3 main components: a feature encoder along with context encoder, a 4D correlation layer and GRU-based update operator.

Table 1
Main descriptive details of the public PIV dataset from Cai et al. (2019b).

Case name	Main property	Parameters	Quantity
Uniform	Uniform flow field by CFD	$ dx \in [0, 5]$	1000
Back-step	Backward stepping flow by CFD	Re = 800	600
		Re = 1000	600
		Re = 1200	1000
		Re = 1500	1000
Cylinder	Flow over a circular cylinder by CFD	Re = 40	50
		Re = 150	500
		Re = 200	500
		Re = 300	500
DNS-turbulence (Carlier, 2005)	A homogeneous and isotropic turbulence flow	Re = 400	500
		–	2000
SQG (Resseguier et al., 2017)	Sea surface flow driven by SQG model	–	1500
JHTDB-channel (Li et al., 2008)	Channel flow provided by JHTDB	–	1600
JHTDB-mhd1024	Forced MHD turbulence provided by JHTDB	–	800
JHTDB-isotropic1024	Forced isotropic turbulence provided by JHTDB	–	2000

on the LiteFlowNet (Hui et al., 2018a). Similarly, special deconvolution layers are added to the model to obtain high-precision small-scale fluid information (see Fig. 13). The results show that the PIV-LiteFlowNet-en model is superior to the previous PIV-NetS model in both accuracy and efficiency. However, the increase in accuracy comes in exchange for adding layers to the model. This increases the redundancy of model parameters, which has a large room for improvement. As for LiteFlowNet, it enjoys some merits for fluid motion estimation. (1) It imitates the coarse-to-fine estimation strategy of the variational optical flow approach, which can effectively handle motions with different displacement sizes. (2) Feature warping (Brox et al., 2004) operation is adopted in each pyramid level to reduce the distance between the second image and first image. (3) Special flow regularization module can utilize image features to smooth the velocity field and decrease error vector. This process acts like a regularization term in the variational optical flow formulation. (4) Prior assumptions can be easily embedded into the multilayer loss function of the LiteFlowNet model, so researchers can couple knowledge of fluid physics to the network. Inspired by the above, based on LiteFlowNet, Yu et al. (2021b) put forward a supervised learning network to specifically solve the problem of PIV estimation in conditions of illumination variation. Brightness gradient constancy and first-order divergence-curl smoothing terms are

incorporated into the loss function of the network. Furthermore, Guo et al. (2022) presented a multi-frame velocity field estimation network, which effectively transfers and fuses flow field features between different frames, thereby further improving accuracy and computational efficiency.

With the emergence of a new milestone optical flow model RAFT, researchers began to further explore the potential of RAFT. Lagemann et al. (2021b) and Yu et al. (2021a) successively modified the RAFT to represent an excellent PIV estimator. Specifically, Lagemann et al. (2021b) crop particle images to small resolution ($32px \times 32px$) and performs feature extraction without spatial downsampling operation. By this means, the feature of the original image can be extracted and utilized effectively. Different from Lagemann et al. (2021b), Yu et al. (2021a) improve the resolution of the feature encoder from 1/8 to 1/4 by removing the residual blocks of the encoder. This operation also further compresses the model size. In essence, the improved ideas in Lagemann et al. (2021b), Yu et al. (2021a) are both to improve the ability to extract image feature information. The obvious advantage of RAFT is that it iteratively refines the velocity field using a ConvGRU-based update operator. Different from performing an iterative estimation from coarse to fine in the LiteFlowNet model, RAFT can update and iterate the velocity field multiple times during training

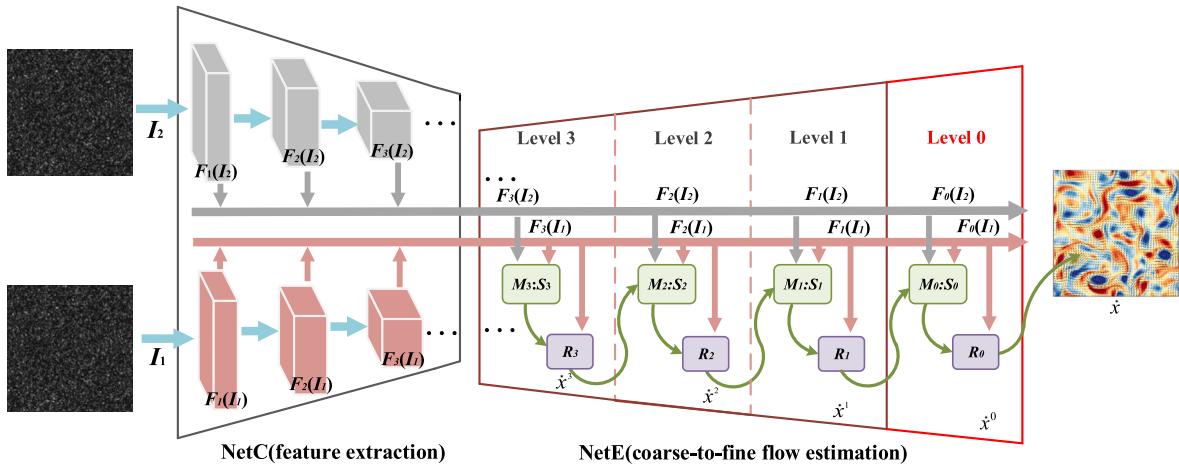


Fig. 13. The architecture of PIV-LiteFlowNet-en (Cai et al., 2019a). The flow field in the figure represents DNS-turbulence flow.

and testing until a satisfactory result is obtained. In addition, the special upsampling method (the convex combination upsampler) also increases the resolution of the flow fields to full resolution with high precision. Benefiting from the huge advantages of the RAFT architecture, these two approaches achieve the state-of-the-art PIV estimation performance. It has high accuracy not only on the PIV dataset, but also has excellent performance in the real flow field (Yu et al., 2021c).

Unsupervised learning

Supervised learning networks usually require a large amount of training data with ground truth, so the corresponding dataset usually need to be artificially generated. To avoid this problem, Zhang and Piggott (2020) proposed an unsupervised version of LiteFlowNet (called UnLiteFlowNet-PIV) inspired by Yu et al. (2016), Meister et al. (2018). The total unsupervised loss of the UnLiteFlowNet-PIV is a combination of photometric loss, flow smoothness loss and consistency loss. Concretely, the **photometric loss** is defined in terms of the difference between the two frames and corresponding forward-backward warped images, which is defined as follows:

$$L_P(I_1, I_2, \mathbf{F}^f, \mathbf{F}^b) = \sum_{\mathbf{x} \in P} \rho(I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{F}^f(\mathbf{x}))) + \rho(I_2(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{F}^b(\mathbf{x}))) \quad (23)$$

where P represents real number space, $\mathbf{x} + \mathbf{F}^f$ and $\mathbf{x} + \mathbf{F}^b$ denotes the corresponding coordinates position in the other image. Here ρ denotes the Charbonnier penalty function (Teng et al., 2005). In addition, a **second-order smooth term** is used to reduce the error vector and enhance the regularization effect, which is described as follows:

$$L_S(\mathbf{F}^f, \mathbf{F}^b) = \sum_{(s,r) \in N(x)} \sum_{x \in P} \rho(\mathbf{F}^f(s) - 2\mathbf{F}^f(x) + \mathbf{F}^f(r)) + \rho(\mathbf{F}^b(s) - 2\mathbf{F}^b(x) + \mathbf{F}^b(r)) \quad (24)$$

where N denotes a four channel filter including x, y and two diagonals, and s and r respectively denote the two pixels before and after x , more details can refer to Zhang et al. (2014). The **consistency loss** indicates that the forward and backward flow estimates should be consistent, which is described as follows:

$$L_C(\mathbf{F}^f, \mathbf{F}^b) = \sum_{x \in P} \rho(\mathbf{F}^f + \mathbf{F}^b(\mathbf{x} + \mathbf{F}^f)) + \rho(\mathbf{F}^b + \mathbf{F}^f(\mathbf{x} + \mathbf{F}^b)) \quad (25)$$

The forward flow \mathbf{F}^f should be the inverse of the backward flow $\mathbf{F}^b(\mathbf{x} + \mathbf{F}^f)$ at the corresponding pixel in the second image. Finally, experimental results show that UnLiteFlowNet-PIV can achieve competitive results compared with supervised learning methods.

Lagemann et al. (2021a) replaced the LiteFlowNet model in this framework with the RAFT model, which achieved better performance.

This is due to the optical flow architecture RAFT is superior to LiteFlowNet. Subsequently, Zhang et al. (2022) further developed the unsupervised fluid motion estimation strategy to embed more prior physical knowledge into the framework (here termed as Un-LiteFlowNet-PIV-2). This framework consist of a PDE (partial differential equation)-constrained motion predictor and a physical based corrector. This prediction-correction scheme is applied to LiteFlowNet or PWCNet architecture to further improve the estimation accuracy. Moreover, it also has great generalization ability to complex real-world fluid scenes. Overall, the aforementioned deep learning-based optical flow methods have achieved encouraging results in recent years. It can be predicted that the deep optical flow model has important application prospects and research potential in fluid motion estimation.

3.2. Cross-correlation learning

Inspired by cross-correlation algorithms, some scholars have also elaborately designed deep cross-correlation learning networks to perform end-to-end PIV estimation (Lee et al., 2017; Gao et al., 2021). Typically, Lee et al. (2017) adopt a four-level cascaded deep convolutional network called PIV-DCNN to gradually generate coarse-to-fine velocity vectors (see Fig. 14). Same as the traditional cross-correlation method, the input of the PIV-DCNN is corresponding two patches (color squares) from the successive images pair. Every sub-net can estimate a velocity vector from two patches. F1 at level 1 can be regarded as an extractor of the large displacement vector, and networks at level 2, 3 and 4 then refine the vectors by calculating the residual vector (VecRes) after central difference window offset. The performance of PIV-DCNN is competitive with the classical PIV algorithms, e.g., the window deformation iterative multi-grid (i.e., WIDIM) algorithm. However, this requires a lot of execution time due to the large number of patches that need to be recalculated. Furthermore, the velocity field output by the PIV-DCNN is a sparse and low-resolution velocity field. The proposal of method PIV-DCNN provides an important reference for velocity estimation based on deep learning.

Although deep learning-based PIV methods have showed potential, e.g., high accuracy and spatial resolution, the generalization ability and robustness of the related approaches still can be further enhanced for real applications. Gao et al. (2021) put forward a deep learning model called CC-FCN integrated with cross correlation strategy, which can fight against noise and achieve satisfactory results in practical applications. The CC-FCN network synergistically combines cross-correlation learning and fully convolutional network. Two types of calculations are used as input to the model, including the particle image pair and initial velocity field obtained by cross correlation. The embedded cross-correlation approach estimates a coarse velocity field with a large

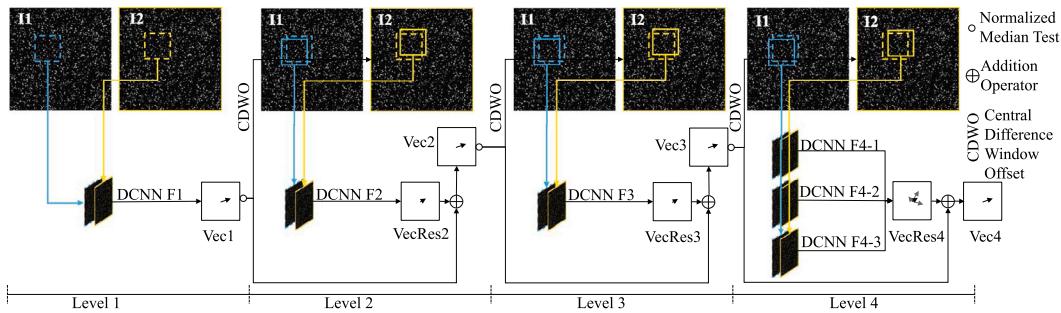


Fig. 14. Schematic diagram of the PIV-DCNN structure.

Source: Reprinted from Lee et al. (2017), with permission from Springer Nature.

interrogation window. It is well known that the execution process of cross-correlation algorithm is based on the calculation of interrogation window matching, which has strong robustness and against the effects of noise in particle images. Therefore, using this prior velocity field as a reference velocity field can contribute to enhancing the robustness of the model. The integration module then integrates these features appropriately and fed to a series of deconvolution layers to predict dense (single-pixel) velocity fields. The trained model can reasonably estimate the high-precision velocity field in the real flow fields.

Fig. 15 presents the architecture of CC-FCN and examples of velocity fields the jet flow estimated by different methods. As shown in Fig. 15(b), all methods achieve good results for velocity field estimation. To further compare the velocity field details extracted by these three approaches, a section of a typical near-wall region (labeled with a black dashed box) with rich flow structures was enlarged for observation. Note that the colored maps in the figure show the magnitude of the velocity, and two yellow dots and one white dot indicate the center of the near-wall vortex and the saddle, respectively. Cross-correlation algorithm, as a mature fluid motion estimation method, can extract flow structure with relatively high reliability. In addition, the spatial information of typical structures (i.e., two vortex centers and a saddle) captured by CC-FCN is closer to the cross-correlation benchmark method compared to the PIV-LiteFlowNet-en. Both CC-FCN and the cross-correlation method extract relatively high velocity around the vortices, while the former estimates more details of the high-speed regions. In contrast, the flow field extracted by PIV-LiteFlowNet-en has a smaller high-speed regions around the two vortices. More comparison and descriptions can refer to Gao et al. (2021). Furthermore, due to the calculation of cross-correlation algorithm and multi-layer deconvolution embedded in the CC-FCN model, the process of estimating the velocity field is relatively time-consuming.

3.3. Applicability analysis

We conduct a comparison of deep learning-based fluid motion estimation methods on the public PIV dataset in this subsection. In addition, we also give a modest analysis and discussion on the applicability and computing costs of these methods. We first summarize test results of deep learning methods on the public PIV dataset. For fair comparison, we uniformly use the Averaged Endpoint Error (AEE) metric to evaluate the performance of the methods. The AEE can be defined as the L_2 distance between the estimated flow \mathbf{F}_e and the ground truth flow \mathbf{F}_g , which is described as follows:

$$\text{AEE} = \|\mathbf{F}_e - \mathbf{F}_g\|_2 \quad (26)$$

Table 2 presents the estimation results of different methods on the public PIV dataset. For easy distinction, UnLite-FlowNet-PIV-2 represents the new unsupervised framework proposed by Zhang et al. (2022). It can be seen that the deep learning-based methods achieve

superior performance for different flow fields compared to traditional algorithms. Estimation accuracy are continuously improved as better variants emerge. The supervised learning method has achieved high-precision velocity field estimation results, and the variant based on the optical flow model RAFT can achieve more outstanding prediction results than the LiteFlowNet. Similarly, in the unsupervised PIV estimation framework, the effect of embedding the RAFT model It is also better than embedding the LiteFlowNet. This proves that the new milestone optical flow model RAFT has stronger feature representation ability in the fluid estimation task and can become the new basic architecture for PIV estimation. Furthermore, the RAFT has fewer parameters and inference time compared to the LiteFlowNet. For example, the LightPIVNet model proposed by Yu et al. (2021a) has a parameter size of only 3.725M, which further compresses the RAFT parameter structure. Although the evaluation results of unsupervised learning methods on the training data set are slightly lower than that of supervised learning methods, they integrate the loss of prior knowledge of fluid physics in the training. This enables the model to have strong robustness and generalization ability in the face of unknown flow fields. Similarly, the CC-FCN model embedded with cross-correlation learning also has good generalization ability.

In overall, the current deep learning methods can achieve high estimation accuracy in the PIV dataset, and its calculation error is already very low. This is due to the powerful feature learning and representation capabilities of deep learning technique. Optical flow methods occupy half of the PIV estimation methods, which also proves the advantage of optical flow learning for fluid motion estimation. At present, the optical flow model RAFT has great advantages in inference time, parameters and accuracy. But it is worth noting that the 4D cost volume of the RAFT consume more memory during the calculation process. With the advent of better optical flow variants, more applicable fluid motion estimator will emerge.

In addition, compared with the RAFT, the LiteFlowNet and PWCNet models based on the multi-level pyramid structure can be more easily embed the prior physical constraints of the fluid, which is important for the application in specific flow scenarios. This is still an interesting direction in the future to embed more specific prior fluid knowledge into optical flow learning networks.

4. Review of velocity field reconstruction

In this part, we first present a short review of related super-resolution (SR) deep learning methods. Then, the description of fluid super resolution reconstruction is carried out around the three aspects of CNNs and generative adversarial nets(GANs) and physics-informed neural networks (PINNs). Finally, we also give a applicability analysis of the fluid SR reconstruction methods.

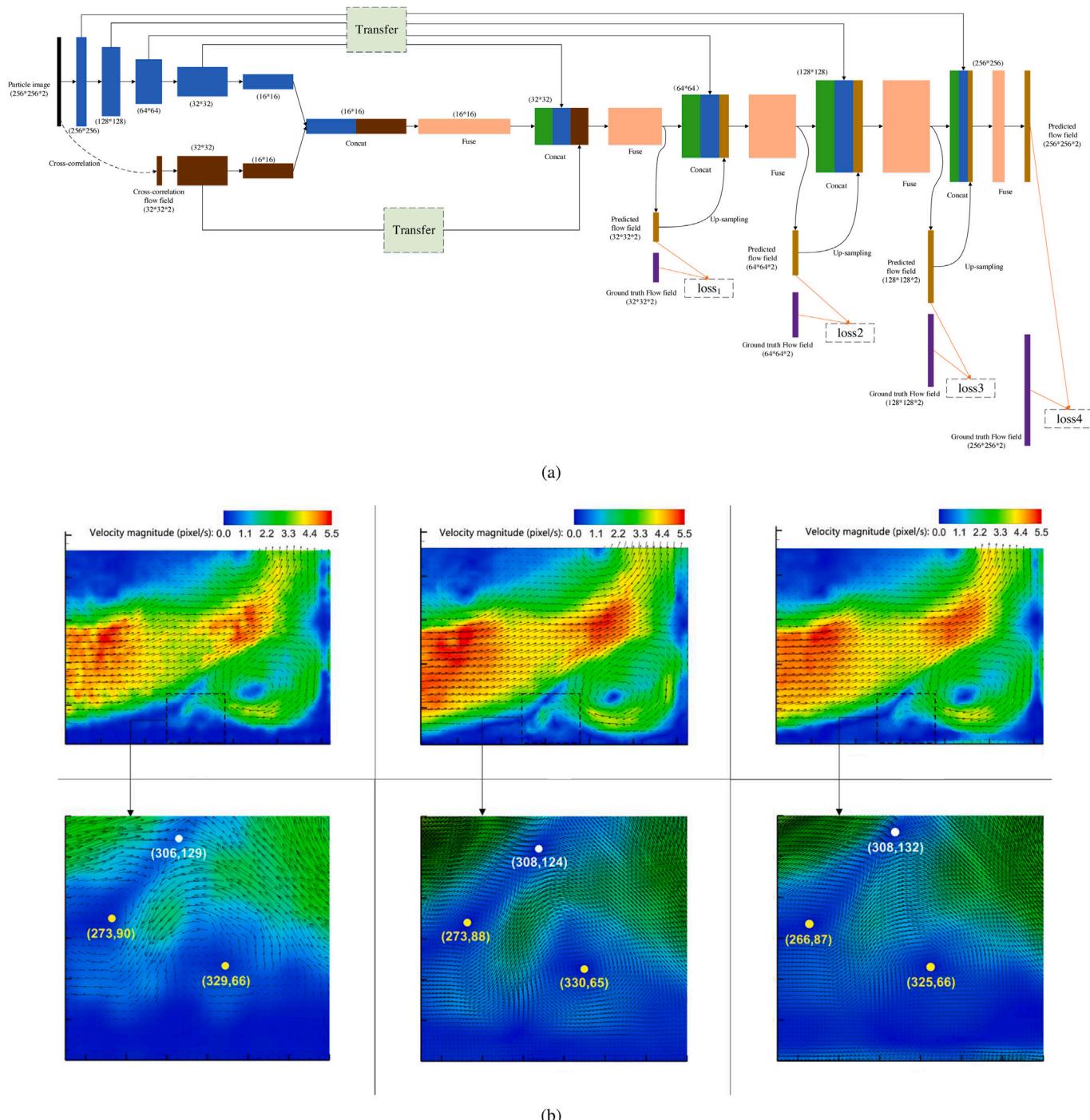


Fig. 15. Schematic diagram of the structure of the CC-FCN model (a) and comparison of jet velocity field estimated by different methods (left column: cross correlation method, middle: CC-FCN method and right: PIV-LiteFlowNet-en method).

Source: Reprinted from Gao et al. (2021), with the permission of AIP Publishing.

4.1. An overview of SR methods

As the name implies, SR technique represents the reconstruction of corresponding high-resolution (HR) data from low-resolution (LR) data. SR technique has extensive applications in the areas of satellite images, medical imaging and monitoring equipment, etc. The super-resolution convolutional neural network (SRCNN) is the pioneering work of deep learning in SR reconstruction (Dong et al., 2014). As shown in Fig. 16, the network structure of SRCNN is relatively simple and just contains three convolutional layers. It first adopts bicubic interpolation to enlarge the LR image to the target size, then fits

the nonlinear mapping through the three-layer convolutional network, and eventually outputs the HR image. Three-layer convolution can be interpreted as three stages: image feature extraction, feature nonlinear mapping and final reconstruction.

Since then, super-resolution methods based on deep learning have been studied and developed intensively. Anwar et al. (2020) have evaluated the single image SR methods on the benchmark dataset, and summarized and classified the SR methods, which can be divided into 9 categories: containing linear, residual, recursive, progressive, densely connected, multi-branch, attention-based, multiple degradation and adversarial design (see Fig. 17). For a linear network, it has a simple

Table 2

The Averaged Endpoint Error (AEE) of various methods for the public PIV-Dataset (Cai et al., 2019b). Note that the error unit is set to pixels per 100 pixels for easier comparison.

Methods	Back-step	Cylinder	JHTDB-channel	DNS-turbulence	SQG
WIDIM (Scarano, 2001)	3.4	8.3	8.4	30.4	45.7
HS (Horn and Schunck, 1981)	4.5	6.9	6.9	13.5	15.6
PIV-DCNN (Lee et al., 2017)	4.9	7.8	11.7	33.4	47.9
PIV-NetS-noRef (Jiang et al., 2021)	13.9	23.7	23.7	52.5	52.5
PIV-NetS (Jiang et al., 2021)	7.2	15.5	15.5	28.2	29.4
PIV-LiteFlowNet (Li et al., 2008)	5.6	10.4	10.4	19.6	20.2
PIV-LiteFlowNet-en (Li et al., 2008)	3.3	7.5	7.5	12.2	12.6
PIV-RAFT (Yu et al., 2021a)	2.1	3.1	12.6	10.7	17.9
LightPIVNet (Yu et al., 2021a)	1.3	2.4	5.8	7.7	9.9
RAFT256-PIV (Lagemann et al., 2021b)	1.6	1.4	13.7	9.3	11.7
RAFT32-PIV (Lagemann et al., 2021b)	0.4	1.8	1.1	2.8	2.1
CC-FCN (Gao et al., 2021)	3.4	3.3	13.5	10.5	22.5
UnLiteFlowNet-PIV (Zhang and Piggott, 2020)	10.1	7.8	9.6	13.5	19.7
UnLiteFlowNet-PIV-2 (Zhang et al., 2022)	9.4	6.9	8.4	15.0	17.3
UnPwcNet-PIV (Zhang et al., 2022)	8.2	7.1	13.4	21.5	25.2
URAF-PIV (Lagemann et al., 2021a)	6.5	6.6	8.1	12.5	13.2

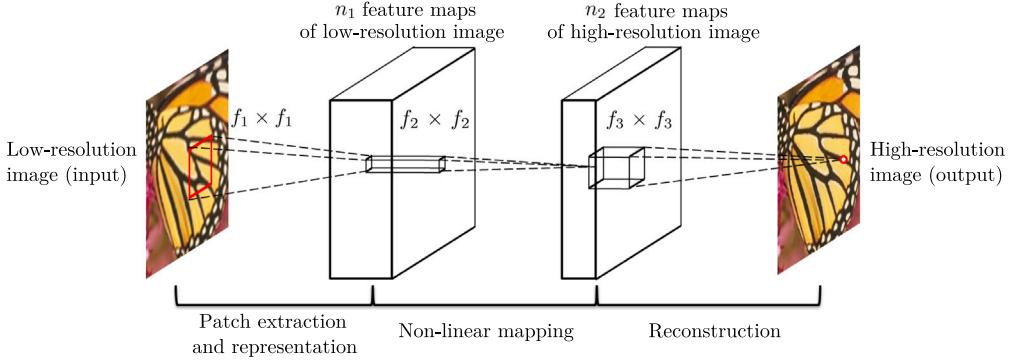


Fig. 16. Schematic diagram of the SRCNN structure (Dong et al., 2014).

structure consisting of a single path or multiple paths without any skip connections. As for such a network design, several convolutional layers are stacked, and input features from the initial layer are fed to subsequent layers in order. The linear networks operate in different ways of upsampling, i.e. early upsampling or late upsampling. For example, DnCNN (Zhang et al., 2017) directly learns to predict high-frequency residuals rather than potentially super-resolution images. Similar to SRCNN, the architecture of DnCNN is very simple because it only stacks several convolutional layers. However, its performance heavily depends on the precision of noise estimation. In addition, DnCNN is relatively computationally expensive due to the batch normalization (BN) operation after each convolutional layer. Compared to linear networks, residual learning adopts skip connections to avoid vanishing gradients and makes it feasible to design very deep networks (He et al., 2016). Lim et al. (2017) put forward an enhanced deep super-resolution network (EDSR), introduced the residual block to expand the depth of the model to further improve performance, and removed unnecessary BN layers of the previous network to stabilize training process. Subsequently, Yu et al. (2018) improved the EDSR by removing the redundant convolution layer and splitting the residual body into two parts for feature fusion, thus further improving the SR performance. Furthermore, Tai et al. (2017) proposed a deep CNN architecture in which 52 convolutional layers are employed, and called deep recurrent residual network (DRRN). Specifically, the model adopts global and local residual learning to solve the problem of training very deep networks, and uses recursive learning to increase the depth without increasing the model parameters. To extract richer feature information of images, the design ideas of dense connection and multi-branch are

introduced to SR networks such as SRDenseNet (Tong et al., 2017), RDN (Zhang et al., 2018c) and IDN (Hui et al., 2018b).

The above network model considers that the spatial location and feature channel are unified in the process of feature extraction. However, not all features are necessary for the SR tasks. The attention mechanism technique can effectively help the model focus on more important local features. Anwar and Barnes (2020) introduce the Densely Residual Laplacian Network (DRLN) for the SR image reconstruction task (see Fig. 18). The novel design of the DRLN model is to propose the densely connected residual units and Laplacian attention mechanism. Specifically, the residual architecture consists of the dense residual Laplacian module (DRLM) hierarchically in a cascaded manner. Then, the learned features are weighted and fused using Laplacian attention in each module to adaptively learn features of different scales. Empirical results also indicate that the DRLN method has excellent performance in terms of vision and accuracy. For multiple degradation processing networks, only bicubic degradation is considered, which may not be a feasible assumption in practical applications because multiple degradations may occur simultaneously. Many existing CNN-based SR approaches assume bicubic downsampling of LR images from HR images. However, when the actual degradation does not satisfy this assumption, it will inevitably lead to poor performance. To address these issues, Zhang et al. (2018b) present a general framework with dimensional stretching strategies. This framework utilizes a single convolutional SR models to take as input two important factors of the SR degradation process, namely the blur kernel and the noise level. Therefore, this method can solve multiple and even spatially varying degradations, which greatly enhances the practicability. Similarly, the super resolution multiple degenerate network (SRMD) (Zhang

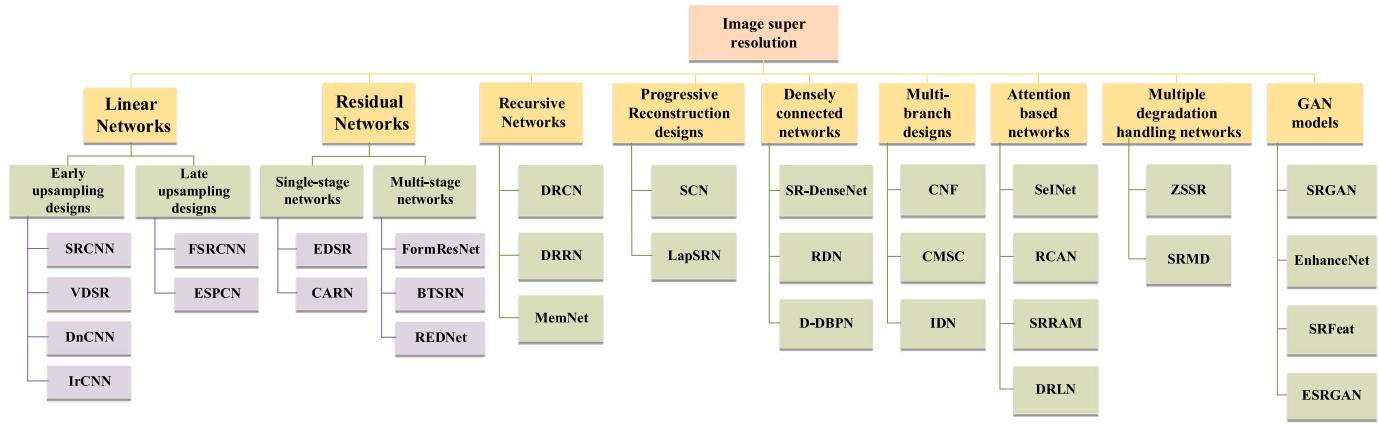


Fig. 17. Schematic diagram of the classification of existing single image SR approaches.

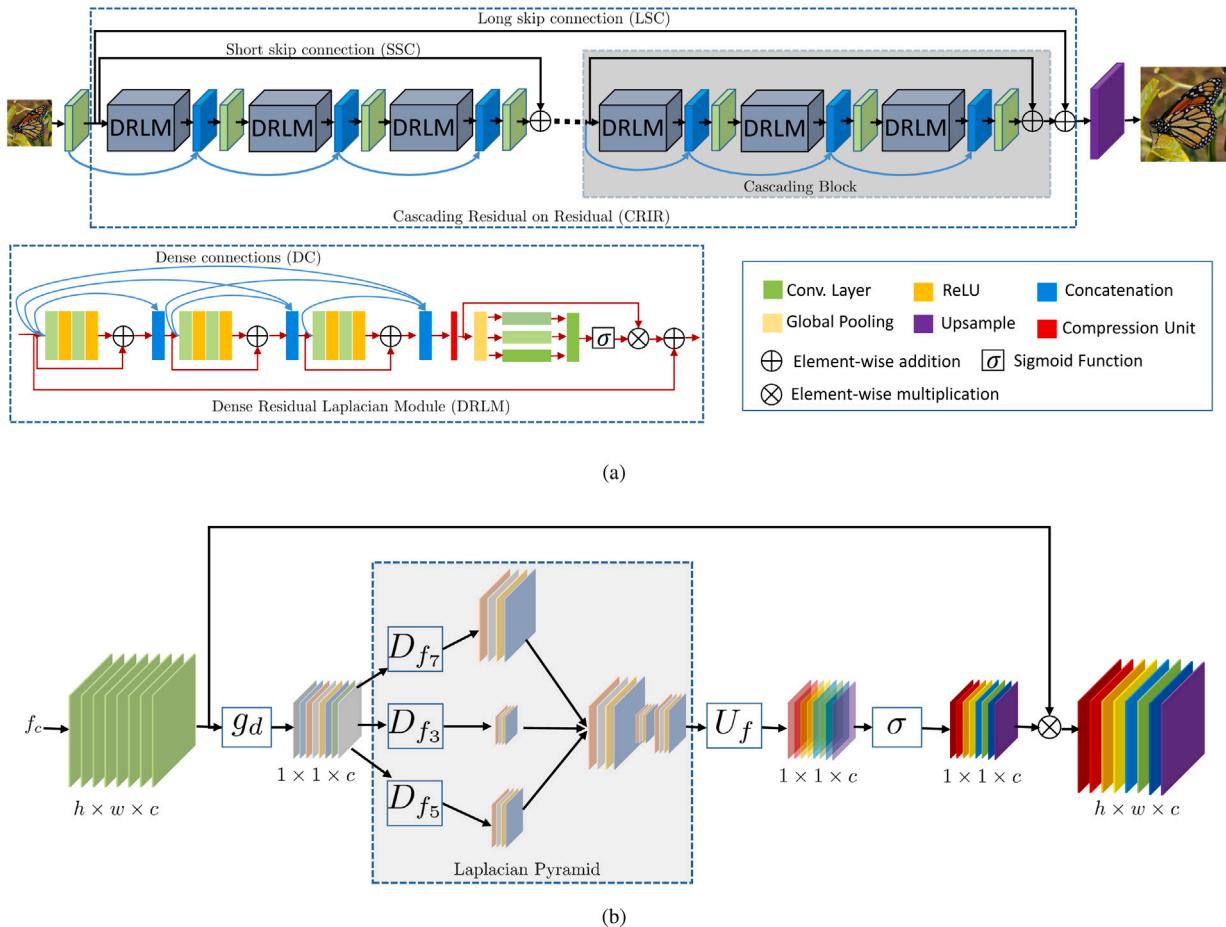


Fig. 18. Schematic diagram of the DRLN structure (a) Anwar and Barnes (2020) and Laplacian attention module (b).

et al., 2018d) employed a scheme of connecting LR images and their degradation maps.

In addition, the use of generative adversarial networks (GANs) to perform SR tasks is also an interesting and promising research direction. In addition, the physics-informed neural network (PINN) (Raissi et al., 2019) is a novel neural network proposed recently, which has shown great advantages in the field of flow fields reconstruction. Next, we give a review of the velocity fields reconstruction from 4 aspects: CNN-based methods, GAN-based methods, RNN-based methods and PINN-based methods.

4.2. CNN-based methods

Inspired by machine learning, Fukami et al. (2019) develop two deep learning models, namely convolutional neural network and the hybrid downsampled skip-connection/ multi-scale (called DSC-MS) models. Both of them are applied to test two-dimensional cylinder wake flows. The results show that it has superior ability to reconstruct turbulent and laminar flow from LR flow fields data. Meanwhile, it pioneered the task of flow field reconstruction using deep learning methods. Subsequently, Liu et al. (2020) also propose two deep learning networks to address the SR reconstruction of turbulent flows from LR

coarse flow field data. One is the static convolutional neural network (SCNN) based on the SRCNN (Dong et al., 2014). The other one is the novel multiple temporal paths convolutional neural network (termed MTPC) that takes a time series of velocity fields as an input and then output a HR flow field. The MTPC simultaneously considers and integrates the spatial and temporal information of the flow field, which can enhance the reconstruction of the fine-scale structure of turbulence. Fig. 19 shows the SR results of different methods for isotropic turbulence fields at low resolution $r = 4$. It can be seen that the velocity field estimated by the bicubic interpolation is smooth because only low-frequency information can be captured. In contrast, the SRCNN and MTPC achieve good reconstruction results that are closer to the direct numerical simulation (DNS) result, but the latter can estimate more details and reconstruct finer structures than SRCNN. In other words, the MTPC estimates more high-frequency content from the low-resolution velocity field. The author analyzes that this is because the MTPC effectively integrates the extra spatio-temporal information of the flow fields.

Since the high-resolution data is a broad pursuit in physics and engineering, researchers have further applied deep learning SR techniques to various flow fields reconstruction tasks. For instance, Kong et al. (2020) put forward a multipath SR convolutional neural network (MPSRC) to achieve super-resolution temperature field reconstruction. The MPSRC achieves better reconstruction results with lower error and higher peak signal-to-noise ratio (PSNR). Although this method has made great progress, there are still some challenges. For example, when the magnification factor is large, the super-resolution reconstruction performance degrades to a certain extent. Ferdian et al. (2020) proposed the 4DFlowNet to produce noise-free SR 4D flow MRI data. Based on Fukami et al. (2019), Fukami et al. (2021) successively construct a deep learning-based spatio-temporal network to reconstruct turbulence fields. Kong et al. (2021) used simple convolutional layers to construct a multi-path model architecture to reconstruct the low-resolution supersonic flow fields. When the sampling factor is low ($r = 4$), the model can reconstruct a clear background wave. But when the sampling factor is large ($r = 8$), the reconstructed shock structure is completely lost, and the reconstruction result is seriously distorted. Recently, Chen et al. (2022b) proposed a multi-branch fusion convolutional neural network (called MBFCNN) to reconstruct the flow field in a supersonic combustor and achieved great reconstruction results. The MBFCNN model can predict a rich information source for the evolution of the wave system structure under the self-ignition conditions of the hydrogen-fueled scramjet and greatly improves the detection precision. Deng et al. (2022) subsequently developed a dual-branch network based on a multi-head attention mechanism to reconstruct the flow field schlieren image in a supersonic combustor, and results demonstrate that the model can effectively reconstruct the basic wave system structure of a complicated flow field.

It can be seen that the general CNN-based SR techniques improve the accuracy by building a temporal multi-branch network structure. This is because flow field data has multi-scale spatial and temporal characteristics (Pope and Pope, 2000). However, the network designed by this idea also increases the complexity of the model and ignores the computational efficiency. Inspired by single-image SR reconstruction methods in the field of computer vision, Bi et al. (2022) put forward a multi-scale integration network (named FlowSRNet) to reconstruct the HR flow fields. Considering the multi-scale spatial characteristics of the fluid flows, a lightweight multi-scale aggregation block (LMAB) is carefully designed inspired by Gao et al. (2019), which includes a parallel cascading architecture and feature aggregation module. Furthermore, a corresponding SR dataset is build to train and verify the proposed approach, which contains a variety of fluid flows. The results demonstrate that the FlowSRNet model achieves outstanding SR performance for various flow fields. Meanwhile, the FlowSRNet has the advantage of being lightweight, and its parameters are only 0.432M when stacking two LMABs in the backbone network architecture (see Fig. 20).

4.3. GAN-based methods

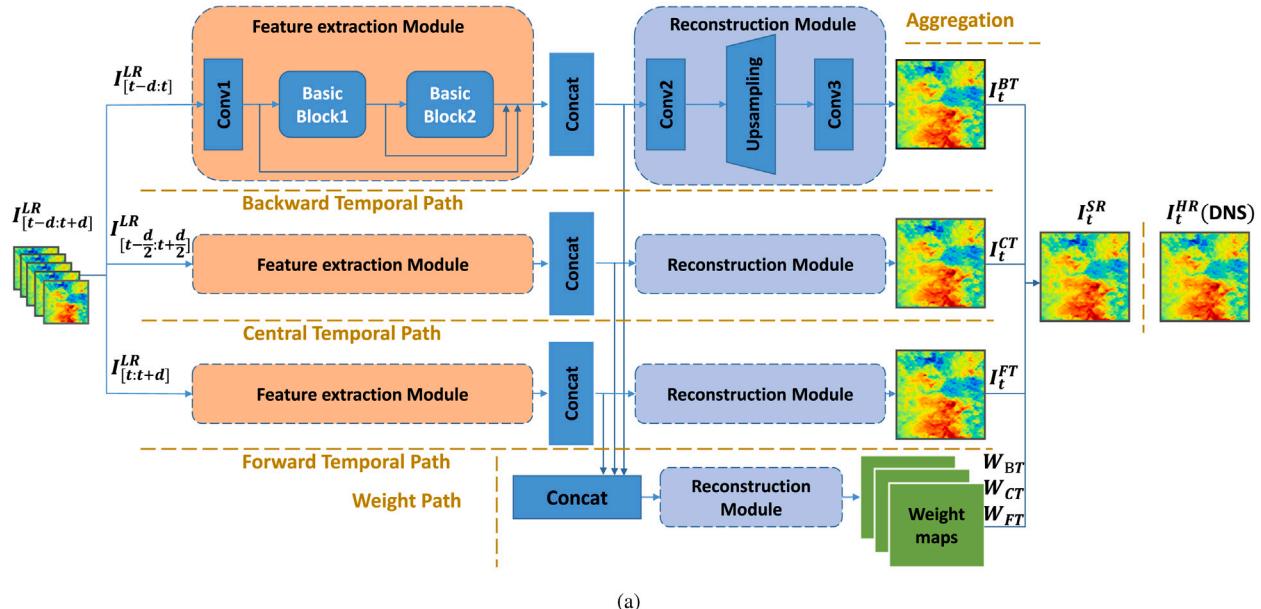
Generative adversarial net (GAN) (Goodfellow et al., 2014) is composed of two networks: one is a generator network (G) and the other is a discriminator (D) (see Fig. 21). The final destination of GAN is to learn a high-quality generator and GAN can obtain high quality generator by adding discriminator. Concretely, the generator aims to generate images during the training process that are as realistic as possible so that the discriminator cannot judge the authenticity of this picture. The purpose of discriminator in the training process is to identify the true or false of this picture as much as possible. Therefore, generator wants to maximize the error rate of discriminator, and discriminator wants to minimize the error rate. The two confront with each other and improve in competition together. In theory, this relationship can reach an equilibrium point, which is the Nash equilibrium. In other words, when the probability of discriminating the picture generated by generator as real data is 0.5, that is, the current discriminator can no longer distinguish the true or false pictures generated by the generator. Then the purpose of the generator also achieves the purpose of mixing the fake with the real. The optimization objective function of GAN is as follows:

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)} [\log(D(x))] + E_{z \sim P_z(z)} [\log(1 - D(G(z)))] \quad (27)$$

where z is a noisy sample of the prior probability distribution P_z , and x is a real sample following a specific distribution P_{data} . The generator works to minimize the target loss function during training, while the discriminator maximizes the target loss function. For the training schedule, one side is fixed during the training process, the parameters of the other network are updated, and the training is carried out alternately and iteratively. Finally, the samples generated by the generator are closer to the real data. Super-resolution techniques based on generative adversarial networks, such as SRGAN (Ledig et al., 2017) and ESRGAN (Wang et al., 2018c), also achieved wonderful performance in image super-resolution tasks.

Because of the novelty of GAN, some researchers improved GAN and applied it to flow fields super-resolution reconstruction. In the TempoGAN network proposed by Xie et al. (2018), two aspects are considered in the discriminator to discriminate space and time respectively. This method can generate more detailed real and time-consistent physical quantities of flow field. Xu et al. (2020) proposed an algorithm for data-driven 3D super-resolution that increases spatial resolution by twofold along each spatial direction. The approach, known as 3D-Superresolution Generative Adversarial Network (3D-SRGAN), constructs a generator and a discriminator network to study topographic information and at a given LR counterpart to infer high-resolution 3D turbulent flame structure. Lee et al. (2018) proposed a deep learning GAN network method for simulating small-scale features of turbulent flow. This method is novel in processing 3D convolution to achieve 3D structure prediction and predicting accurate solutions with less computational cost. Deng et al. (2019) develop SR reconstruction methods from LR flow fields using GAN-based deep learning frameworks SRGAN (Ledig et al., 2017) and ESRGAN (Wang et al., 2018c). The analysis of reconstructed instantaneous flow fields and spatial correlation shows that both models can accurately reconstruct high spatial resolution flow field in complicated flow structures.

However, it is worth noting that training GAN is difficult and unstable, and it is prone to discriminator convergence and generator divergence, leading to model collapse. To deal with this dilemma, Wu et al. (2020) proposed a generative adversarial network embedded with statistical constraints. By enforcing covariance constraints on the training data, the network can fit the statistics of training data produced by solving the fully decomposed partial differential equations. The results show that this statistical regularization results in better performance in comparison to the standard GAN. Lin et al. (2019) proposed a novel GAN network to solve the problem that traditional GAN networks are



(a)

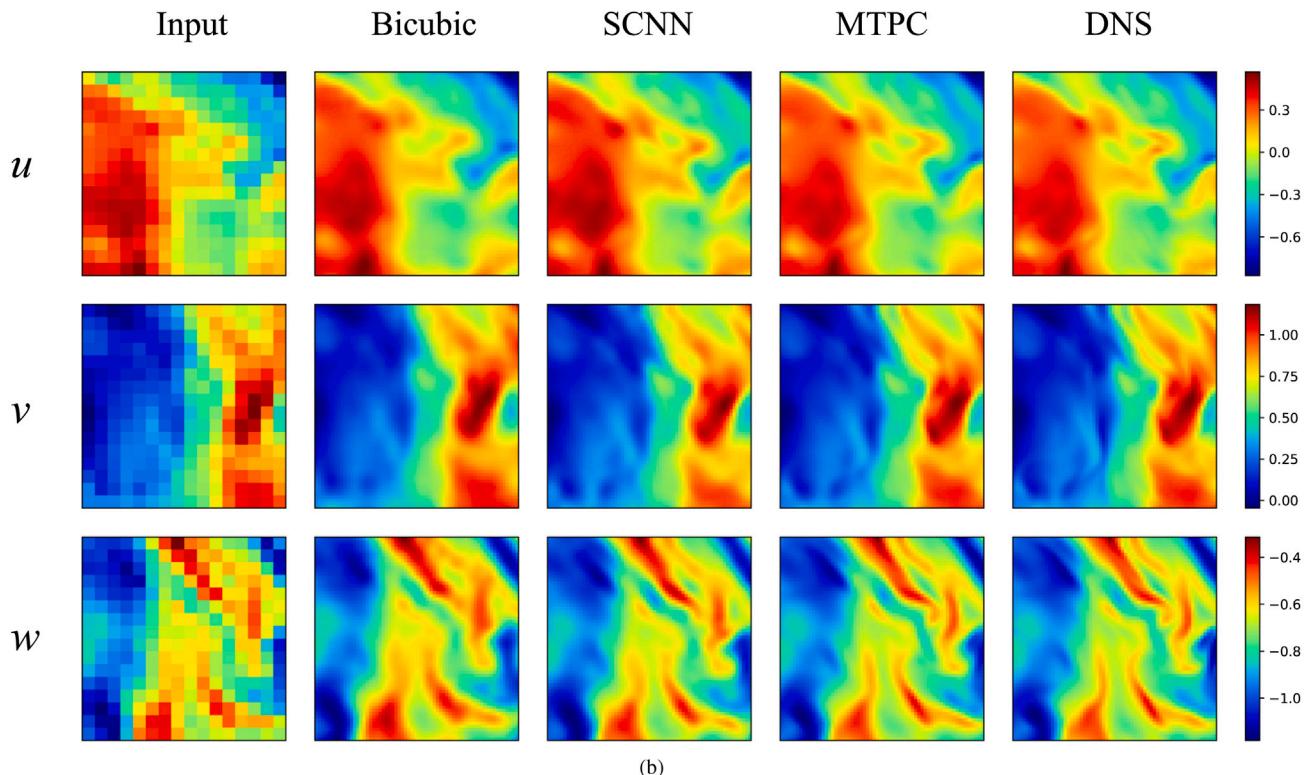


Fig. 19. Schematic diagram of the MTPC structure (Liu et al., 2020) (a) and comparison of different SR methods for the forced isotropic turbulence at $r=4$ (b).
Source: Reprinted from Liu et al. (2020), with the permission from AIP Publishing.

difficult to converge, in which the generation network no longer tries to deceive the discriminative network but seeks the matching problem of corresponding relations. Compared with traditional GAN, this method is superior in terms of iterative convergence and visual detection.

It is well known that unsteady flow has the characteristics of changes in time and space characteristics. Hence, some works have developed recurrent neural networks (e.g., LSTM) to predict the spatio-temporal characteristics of specific flow data (Pawar et al., 2019; Huang et al., 2019). The recurrent neural network itself has the advantage of extracting temporal features, which has broad application prospects in

the task of predicting the features of flow fields data (Li et al., 2021, 2022).

4.4. PINN-based methods

Both data-driven CNN-based and GAN-based methods improve SR performance by modifying the model architecture, and also rely on high-quality training data. However, the training data imply prior physical knowledge in various fluid scenarios, which is not explicitly represented by deep learning models. In recent years, a novel physics-informed neural network (PINN) (Raissi et al., 2019, 2020) is

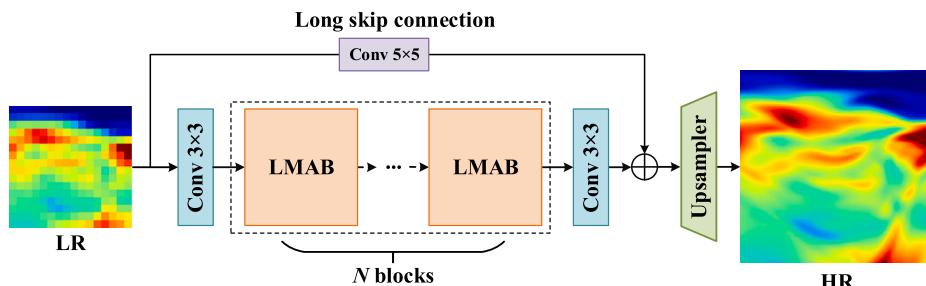


Fig. 20. Schematic diagram of the FlowSRNet structure (Bi et al., 2022).

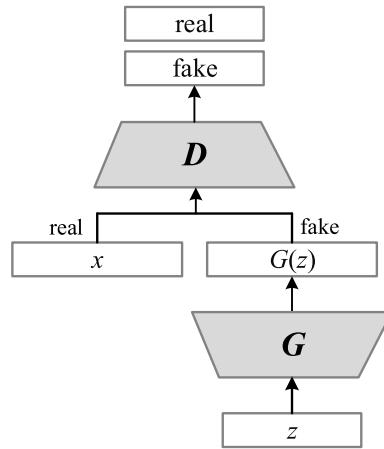


Fig. 21. Schematic diagram of the GAN structure.

proposed by combining the data-driven neural networks and physical laws. Specifically, a neural network could be thought of a general function approximator (Hornik et al., 1989), which can be composed of a fully connected neural network or a residual neural network, and the latter is implemented by embedding partial differential equations (PDEs) into the loss of the neural network using automatic differentiation. Therefore, PINNs can be easily embedded into physical constraints to accomplish the task of flow field reconstruction.

At present, PINNs have been successfully applied to reconstruct the velocity and pressure distributions from flow visualizations including the cylindrical wake flow field and the biological flow field of the intracranial aneurysm image (Raissi et al., 2020). Furthermore, the PINN can also be used to address the incompressible N-S equation, which is based on the velocity-pressure and velocity-vorticity forms (Rao et al., 2020; Jin et al., 2021). Cai et al. (2021) put forward a new technique based on PINNs to predict the full continuous 3D velocity and pressure fields from snapshots of 3D temperature fields that is obtained by the tomographic background oriented schlieren imaging. Experiments show that the PINN method can efficiently infer velocities and pressures from regression data without any information about initial or boundary conditions.

Wang et al. (2022) adopted a PINN to reconstruct dense velocity fields from sparse tomographic PIV data. The PINN not only increases the velocity resolution, but also can be used to predict the pressure field of the flow field. Fig. 22 shows the proposed PINN structure and the reconstruction results. The inputs of the PINN are space coordinates $\mathbf{X} = (x, y, z)$ and time t , and the outputs are velocity $\mathbf{U} = (u, v, w)$, and pressure p . The physical laws are described by incompressible N-S equations, which are described as follows:

$$\begin{aligned} e_1 &= u_t + uu_x + vu_y + wu_z + p_x - \frac{1}{Re} (u_{xx} + u_{yy} + w_{zz}), \\ e_2 &= v_t + uv_x + vv_y + wv_z + p_y - \frac{1}{Re} (v_{xx} + v_{yy} + v_{zz}), \\ e_3 &= w_t + uw_x + vw_y + ww_z + p_z - \frac{1}{Re} (w_{xx} + w_{yy} + w_{zz}), \\ e_4 &= u_x + v_y + w_z. \end{aligned} \quad (28)$$

where Re denotes the Reynolds number, and e_1, e_2, e_3 and e_4 are the errors of the N-S equations. In addition, the partial derivatives $\partial/\partial\mathbf{X}$ and $\partial/\partial t$ can be computed by automatic differentiation in the open source framework (e.g., TensorFlow or PyTorch). Finally, the loss function of the PINN is defined as follows:

$$L = L_{\text{data}} + \alpha L_{\text{eqns}} \quad (29)$$

where α denotes a weighting coefficient, L_{data} represents the loss between the measured data and predicted data, and L_{eqns} represents the total error of the N-S equations. Although there is still a certain gap between the reconstruction effect of PINNs and the traditional classical methods, it has become a promising data assimilation technology, which has aroused the interest of researchers.

4.5. Applicability analysis

We also give an applicability analysis of deep learning-based methods for fluid flow fields reconstruction. Deep learning techniques, such as CNNs and GANs, have many meaningful applications in flow field reconstruction tasks, showing great disadvantages. For CNN, it is relatively easy to build and improve the network structure, and has a strong feature representation ability for the flow fields. Considering the temporal feature of the flow field, building a multi-branch network is a mainstream strategy, but it also leads to a relative increase in computing time. For GAN, it can also effectively reconstruct the texture details and feature information for the flow field. However, an obvious shortcoming of GAN is that it is difficult to train and converge. Similarly, PINN also has the problem of large training cost, and the process of solving partial derivatives multiple times also increases the calculation time. In overall, researchers need to consider the characteristics of specific flow field data to select and design the deep learning model architecture. In addition, the introduction of prior physical constraints is an important trend in the integration and development of deep learning technology and fluid mechanics.

5. Conclusion

A comprehensive survey of deep learning-based methods for fluid velocity field estimation is given in this paper. Specifically, we divide methods for fluid velocity field estimation into two categories, fluid motion estimation and velocity field super-resolution reconstruction methods. First, the principle of deep learning is introduced and described in detail. Second, we mainly investigate deep learning methods for velocity field estimation from successive pair of particle images. Finally, we describe the deep learning approaches for fluid velocity field reconstruction. It is worth noting that we also report experimental results of the representative methods for a better understanding the motivation of the proposed methods. For the development of fluid velocity field estimation methods, we give the following summary and prospect.

Fluid motion estimation. Deep learning methods have achieved remarkable success in the field of fluid motion estimation because they are good at dealing with fluid data with complex nonlinearity,

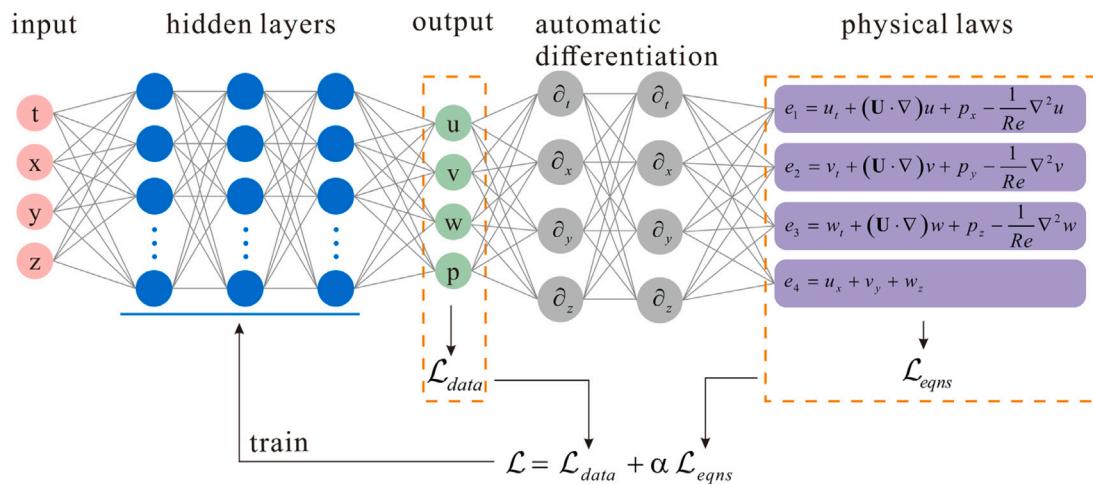


Fig. 22. Schematic diagram of the PINN structure.
Source: Reprinted from Wang et al. (2022), with the permission of AIP Publishing.

high dimensionality, and large quantities. Compared with traditional algorithms, deep learning can well solve problems such as low accuracy, low spatial resolution and computational efficiency. Yet, the following challenges still remain in the development of deep learning methods for fluid motion estimation in the future: (1) The robustness and generalization ability of the model still need to be further improved for practical applications. Some challenges inevitably exist in real images, such as severe noise and large displacement, etc. Therefore, how to effectively solve these problems is also a promising direction. (2) Learning more characteristics of the fluid velocity field, i.e., temporal and high-dimensional properties. Most of the current works are devoted to estimating the velocity field from particle image pairs while ignoring the time-resolved properties of the velocity field. It is well known that fluids have dynamic temporal properties, which are non-local in both time and space. Therefore, it is an interesting direction to obtain the time-resolved characteristics of the fluid velocity field. In addition, it is an inevitable trend to develop from estimating two-dimensional (2D) fluid motion to three-dimensional (3D) fluid motion. In the field of computer vision, 3D motion estimation models have begun to appear and develop, such as scene flow estimation using point cloud technology. Similar to the 2D optical flow model, the 3D scene flow model can be well applied to fluid estimation. But a difficulty here is how to construct a corresponding 3D dataset. We believe that in the future this problem will definitely be solved well. (3) Improving the interpretability of neural network structures. Diverse fluids have complex physical properties, and designing a more physically interpretable network architecture and embedding more prior physical knowledge can effectively enhance the performance of the model. This is a trend in the future integration of deep learning technology and fluid mechanics. (4) Developing small-sample neural network architectures. Deep learning method training requires a large amount of data. Computational fluid dynamics is limited by the high cost of data acquisition. Therefore, how to develop a small-sample neural network motion modeling method is an important research direction that can be generalized in engineering.

Velocity field SR reconstruction. Deep learning-based image SR techniques have achieved good performance. Compared with traditional interpolation methods, these networks show better performance for flow field reconstruction. The deep learning SR methods have a wide range of application tasks and scenarios for fluid reconstruction. Similar to the above, the future development of deep learning in this direction also has many directions. (1) It is also a trend to reconstruct high-dimensional (i.e., 3D or more) velocity field components. Compared to building the PIV dataset, building a high-dimensional SR dataset seems to be moderately difficult. (2) The characteristics

of different flow fields are different. It is of significance to choose a suitable neural network model for a specifical flow field reconstruction problem. (3) The establishment of standard flow field datasets is of great significance to the development of flow field reconstruction, which will contribute to the test and improvement of models. (4) Super-resolution tasks are pixel-level tasks that are time-consuming to process large volumes of fluid data. Building a lightweight model architecture has practical significance in engineering applications.

Declaration of competing 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.

Data availability

Data will be made available on request.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant 62236011.

References

- Adrian, R.J., 2007. Hairpin vortex organization in wall turbulence. *Phys. Fluids* 19 (4), 041301.
- Adrian, R.J., Westerweel, J., 2011. *Particle Image Velocimetry*, No. 30. Cambridge University Press.
- Anwar, S., Barnes, N., 2020. Densely residual laplacian super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Anwar, S., Khan, S., Barnes, N., 2020. A deep journey into super-resolution: A survey. *ACM Comput. Surv.* 53 (3), 1–34.
- Astarita, T., Cardone, G., 2005. Analysis of interpolation schemes for image deformation methods in PIV. *Exp. Fluids* 38 (2), 233–243.
- Bi, X., Liu, A., Fan, Y., Yu, C., Zhang, Z., 2022. FlowSRNet: A multi-scale integration network for super-resolution reconstruction of fluid flows. *Phys. Fluids* 34 (12), 127104.
- Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In: European Conference on Computer Vision. Springer, pp. 25–36.
- Cai, S., Liang, J., Gao, Q., Xu, C., Wei, R., 2019a. Particle image velocimetry based on a deep learning motion estimator. *IEEE Trans. Instrum. Meas.* 69 (6), 3538–3554.
- Cai, S., Wang, Z., Fuest, F., Jeon, Y.J., Gray, C., Karniadakis, G.E., 2021. Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks. *J. Fluid Mech.* 915.

- Cai, S., Zhou, S., Xu, C., Gao, Q., 2019b. Dense motion estimation of particle images via a convolutional neural network. *Exp. Fluids* 60 (4), 1–16.
- Carlier, J., 2005. Second set of fluid mechanics image sequences. European Project Fluid Image Analysis and Description (FLUID)-<http://www.fluid.irisa.fr>.
- Cenedese, A., Romano, G., Pagliajunga, A., Terlizzi, M., 1992. Neural Net for Trajectories Recognition in a Flow. Technical Report, UNIVERSITA DEGLI STUDI LA SAPIENZA ROME, ITALY.
- Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. *Sci. Rep.* 8 (1), 1–12.
- Chen, Z., Bi, X., Zhang, Y., Yue, J., Wang, H., 2022a. LightweightDeRain: learning a lightweight multi-scale high-order feedback network for single image de-raining. *Neural Comput. Appl.* 34 (7), 5431–5448.
- Chen, H., Guo, M., Tian, Y., Le, J., Zhang, H., Zhong, F., 2022b. Intelligent reconstruction of the flow field in a supersonic combustor based on deep learning. *Phys. Fluids* 34 (3), 035128.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078).
- Cholewiak, M.R., 2007. Modeling and correction of peak-locking in digital PIV. *Exp. Fluids* 42 (6), 913–922.
- Coppetti, T., Heitz, D., Arroyo, G., Mémin, E., Santa-Cruz, A., 2006. Fluid experimental flow estimation based on an optical-flow scheme. *Exp. Fluids* 40 (1), 80–97.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.
- Deng, X., Guo, M., Chen, H., Tian, Y., Le, J., Zhang, H., 2022. Dual-path flow field reconstruction for a scramjet combustor based on deep learning. *Phys. Fluids* 34 (9), 095118.
- Deng, Z., He, C., Liu, Y., Kim, K.C., 2019. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys. Fluids* 31 (12), 125111.
- Dong, C., Loy, C.C., He, K., Tang, X., 2014. Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision. Springer, pp. 184–199.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T., 2015. Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2758–2766.
- Du, X., Cai, Y., Wang, S., Zhang, L., 2016. Overview of deep learning. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation. YAC, IEEE, pp. 159–164.
- Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint [arXiv:1603.07285](https://arxiv.org/abs/1603.07285).
- Ferdian, E., Sunesiaputra, A., Dubowitz, D.J., Zhao, D., Wang, A., Cowan, B., Young, A.A., 2020. 4DFlowNet: super-resolution 4D flow MRI using deep learning and computational fluid dynamics. *Front. Phys.* 138.
- Fleit, G., Baranya, S., 2019. An improved particle image velocimetry method for efficient flow analyses. *Flow Meas. Instrum.* 69, 101619.
- Fukami, K., Fukagata, K., Taira, K., 2019. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* 870, 106–120.
- Fukami, K., Fukagata, K., Taira, K., 2021. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* 909.
- Gao, S.H., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P., 2019. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (2), 652–662.
- Gao, Q., Lin, H., Tu, H., Zhu, H., Wei, R., Zhang, G., Shao, X., 2021. A robust single-pixel particle image velocimetry based on fully convolutional networks with cross-correlation embedded. *Phys. Fluids* 33 (12), 127125.
- Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. In: JMLR Workshop and Conference Proceedings, pp. 315–323.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* 27.
- Grant, I., Pan, X., 1997. The use of neural techniques in PIV and PTV. *Meas. Sci. Technol.* 8 (12), 1399.
- Guo, C., Fan, Y., Yu, C., Han, Y., Bi, X., 2022. Time-resolved particle image velocimetry algorithm based on deep learning. *IEEE Trans. Instrum. Meas.* 71, 1–13.
- Guo, X., Li, W., Iorio, F., 2016. Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 481–490.
- Hassan, Y., Philip, O., 1997. A new artificial neural network tracking technique for particle image velocimetry. *Exp. Fluids* 23 (2), 145–154.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.
- Heitz, D., Mémin, E., Schnorr, C., 2010. Variational fluid flow measurements from image sequences: synopsis and perspectives. *Exp. Fluids* 48 (3), 369–393.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Horn, B.K., Schunck, B.G., 1981. Determining optical flow. *Artificial Intelligence* 17 (1–3), 185–203.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Netw.* 2 (5), 359–366.
- Hua, M., Bie, X., Zhang, M., Wang, W., 2014. Edge-aware gradient domain optimization framework for image filtering by local propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2838–2845.
- Huang, J., Liu, H., Cai, W., 2019. Online in situ prediction of 3-D flame evolution from its history 2-D projections via deep learning. *J. Fluid Mech.* 875.
- Hui, T.W., Tang, X., Loy, C.C., 2018a. Lifeflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8981–8989.
- Hui, Z., Wang, X., Gao, X., 2018b. Fast and accurate single image super-resolution via information distillation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 723–731.
- Hur, J., Roth, S., 2019. Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5754–5763.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T., 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2462–2470.
- Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R., 2021. Learning to estimate hidden motions with global motion aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9772–9781.
- Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 426, 109951.
- Kapulla, R., Hoang, P., Szijarto, R., Fokken, J., 2011. Parameter sensitivity of optical flow applied to PIV images. In: Proceedings of the Fachtagung “Lasermethoden in der Strömungsmesstechnik”. Ilmenau, Germany, pp. 6–8.
- Khalid, M., Pénard, L., Mémin, E., 2019. Optical flow for image-based river velocity estimation. *Flow Meas. Instrum.* 65, 110–121.
- Kong, C., Chang, J.T., Li, Y.F., Chen, R.Y., 2020. Deep learning methods for super-resolution reconstruction of temperature fields in a supersonic combustor. *AIP Adv.* 10 (11), 115021.
- Kong, C., Chang, J., Wang, Z., Li, Y., Bao, W., 2021. Data-driven super-resolution reconstruction of supersonic flow field by convolutional neural networks. *AIP Adv.* 11 (6), 065321.
- Koutnik, J., Greff, K., Gomez, F., Schmidhuber, J., 2014. A clockwork rnn. In: International Conference on Machine Learning. PMLR, pp. 1863–1871.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25.
- Kutz, J.N., 2017. Deep learning in fluid dynamics. *J. Fluid Mech.* 814, 1–4.
- Lagemann, C., Klaas, M., Schröder, W., 2021a. Unsupervised recurrent all-pairs field transforms for particle image velocimetry. In: 14th International Symposium on Particle Image Velocimetry, Vol. 1, No. 1.
- Lagemann, C., Lagemann, K., Mukherjee, S., Schröder, W., 2021b. Deep recurrent optical flow learning for particle image velocimetry data. *Nat. Mach. Intell.* 3 (7), 641–651.
- Lai, W.S., Huang, J.B., Yang, M.H., 2017. Semi-supervised learning for optical flow with generative adversarial networks. *Adv. Neural Inf. Process. Syst.* 30.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521 (7553), 436–444.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86 (11), 2278–2324.
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4681–4690.
- Lee, J., Lee, S., You, D., 2018. Deep learning approach in multi-scale prediction of turbulent mixing-layer. arXiv preprint [arXiv:1809.07021](https://arxiv.org/abs/1809.07021).
- Lee, W., Seong, J.J., Ozlu, B., Shim, B.S., Marakhimov, A., Lee, S., 2021. Biosignal sensors and deep learning-based speech recognition: A review. *Sensors* 21 (4), 1399.
- Lee, Y., Yang, H., Yin, Z., 2017. PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. *Exp. Fluids* 58 (12), 1–10.
- Li, Y., Chang, J., Wang, Z., Kong, C., 2021. An efficient deep learning framework to reconstruct the flow field sequences of the supersonic cascade channel. *Phys. Fluids* 33 (5), 056106.
- Li, Y., Hao, Z., Lei, H., 2016. Survey of convolutional neural network. *J. Comput. Appl. Math.* 36 (9), 2508.
- Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., Chen, S., Szalay, A., Eyink, G., 2008. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *J. Turbul.* 9, N31.
- Li, Y., Wang, Z., Jiang, W., Xie, Z., Kong, C., Chang, J., 2022. Research on time sequence prediction of the flow field structure of supersonic cascade channels in wide range based on artificial neural network. *Phys. Fluids* 34 (1), 016106.
- Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K., 2017. Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 136–144.
- Lin, J., Lensink, K., Haber, E., 2019. Fluid flow mass transport for generative networks. arXiv preprint [arXiv:1910.01694](https://arxiv.org/abs/1910.01694).

- Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* 807, 155–166.
- Liu, P., Lyu, M., King, I., Xu, J., 2019. Selfflow: Self-supervised learning of optical flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4571–4580.
- Liu, T., Merat, A., Makhmalbaf, M., Fajardo, C., Merati, P., 2015. Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Exp. Fluids* 56 (8), 1–23.
- Liu, B., Tang, J., Huang, H., Lu, X.Y., 2020. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* 32 (2), 025105.
- Meister, S., Hur, J., Roth, S., 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1.
- Mirza, A.H., Cosan, S., 2018. Computer network intrusion detection using sequential LSTM neural networks autoencoders. In: 2018 26th Signal Processing and Communications Applications Conference. SIU, IEEE, pp. 1–4.
- Mohamed, A.R., Dahl, G., Hinton, G., et al., 2009. Deep belief networks for phone recognition. In: Nips Workshop on Deep Learning for Speech Recognition and Related Applications, Vol. 1, No. 9. p. 39.
- Nguyen, T.D., Wells, J.C., Nguyen, C.V., 2012. Velocity measurement of near-wall flow over inclined and curved boundaries by extended interfacial particle image velocimetry. *Flow Meas. Instrum.* 23 (1), 33–39.
- Otter, D.W., Medina, J.R., Kalita, J.K., 2020. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* 32 (2), 604–624.
- Pascanu, R., Gulcehre, C., Cho, K., Bengio, Y., 2013. How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026.
- Pawar, S., Rahman, S., Vaddiraju, H., San, O., Rasheed, A., Vedula, P., 2019. A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Phys. Fluids* 31 (8), 085101.
- Pope, S.B., Pope, S.B., 2000. Turbulent Flows. Cambridge University Press.
- Rabault, J., Kolaja, J., Jensen, A., 2017. Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Meas. Sci. Technol.* 28 (12), 125301.
- Raffel, M., Willert, C.E., Wereley, S.T., Kompenhans, J., 2007. Mathematical Background of Statistical PIV Evaluation.
- Raiissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707.
- Raiissi, M., Yazdani, A., Karniadakis, G.E., 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* 367 (6481), 1026–1030.
- Ranjan, A., Black, M.J., 2017. Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4161–4170.
- Rao, C., Sun, H., Liu, Y., 2020. Physics-informed deep learning for incompressible laminar flows. *Theor. Appl. Mech. Lett.* 10 (3), 207–212.
- Resseguier, V., Mémin, E., Chapron, B., 2017. Geophysical flows under location uncertainty, part II quasi-geostrophy and efficient ensemble spreading. *Geophys. Astrophys. Fluid Dyn.* 111 (3), 177–208.
- Scarano, F., 2001. Iterative image deformation methods in PIV. *Meas. Sci. Technol.* 13 (1), R1.
- Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Stanislas, M., Okamoto, K., Kähler, C.J., Westerweel, J., 2005. Main results of the second international PIV challenge. *Exp. Fluids* 39 (2), 170–191.
- Stanislas, M., Okamoto, K., Kähler, C.J., Westerweel, J., Scarano, F., 2008. Main results of the third international PIV challenge. *Exp. Fluids* 45 (1), 27–71.
- Strubell, E., Ganesh, A., McCallum, A., 2019. Energy and policy considerations for deep learning in NLP. arXiv preprint arXiv:1906.02243.
- Sun, D., Yang, X., Liu, M.Y., Kautz, J., 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8934–8943.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.
- Tai, Y., Yang, J., Liu, X., 2017. Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3147–3155.
- Teed, Z., Deng, J., 2020. Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. Springer, pp. 402–419.
- Teng, C.H., Lai, S.H., Chen, Y.S., Hsu, W.H., 2005. Accurate optical flow computation under non-uniform brightness variations. *Comput. Vis. Image Underst.* 97 (3), 315–346.
- Teo, C., Lim, K., Hong, G., Yeo, M., 1991. A neural net approach in analyzing photograph in PIV. In: Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, pp. 1535–1538.
- Tong, T., Li, G., Liu, X., Gao, Q., 2017. Image super-resolution using dense skip connections. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4799–4807.
- Wang, H., Bi, X., 2021. Person re-identification based on graph relation learning. *Neural Process. Lett.* 53 (2), 1401–1415.
- Wang, H.P., Gao, Q., Wang, S.Z., Li, Y.H., Wang, Z.Y., Wang, J.J., 2018a. Error reduction for time-resolved PIV data based on Navier–Stokes equations. *Exp. Fluids* 59 (10), 1–21.
- Wang, H., Liu, Y., Wang, S., 2022. Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. *Phys. Fluids* 34 (1), 017116.
- Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W., 2018b. Occlusion aware unsupervised learning of optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4884–4893.
- Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C., 2018c. Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops.
- Wereley, S.T., Gui, L., Meinhart, C., 2002. Advanced algorithms for microscale particle image velocimetry. *AIAA J.* 40 (6), 1047–1055.
- Westerweel, J., Scarano, F., 2005. Universal outlier detection for PIV data. *Exp. Fluids* 39 (6), 1096–1100.
- Wu, J.L., Kashinath, K., Albert, A., Chirila, D., Xiao, H., et al., 2020. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *J. Comput. Phys.* 406, 109209.
- Xie, Y., Franz, E., Chu, M., Thuerey, N., 2018. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.* 37 (4), 1–15.
- Xu, W., Luo, W., Wang, Y., You, Y., 2020. Data-driven three-dimensional super-resolution imaging of a turbulent jet flame using a generative adversarial network. *Appl. Opt.* 59 (19), 5729–5736.
- Yang, G., Ramanan, D., 2019. Volumetric correspondence networks for optical flow. *Adv. Neural Inf. Process. Syst.* 32.
- Yu, C., Bi, X., Fan, Y., Han, Y., Kuai, Y., 2021a. LightPIVNet: An effective convolutional neural network for particle image velocimetry. *IEEE Trans. Instrum. Meas.* 70, 1–15.
- Yu, C.D., Fan, Y.-W., Bi, X.J., Han, Y., Kuai, Y.F., 2021b. Deep particle image velocimetry supervised learning under light conditions. *Flow Meas. Instrum.* 80, 102000.
- Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., Huang, T., 2018. Wide activation for efficient and accurate image super-resolution. arXiv preprint arXiv:1808.08718.
- Yu, J.J., Harley, A.W., Derpanis, K.G., 2016. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: European Conference on Computer Vision. Springer, pp. 3–10.
- Yu, C., Luo, H., Fan, Y., Bi, X., He, M., 2021c. A cascaded convolutional neural network for two-phase flow PIV of an object entering water. *IEEE Trans. Instrum. Meas.* 71, 1–10.
- Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer, pp. 818–833.
- Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R., 2010. Deconvolutional networks. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2528–2535.
- Zhai, M., Xiang, X., Lv, N., Kong, X., 2021. Optical flow and scene flow estimation: A survey. *Pattern Recognit.* 114, 107861.
- Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A.E.D., Jin, W., Schuller, B., 2018a. Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Trans. Intell. Syst. Technol.* 9 (5), 1–28.
- Zhang, C., Li, Z., Cai, R., Chao, H., Rui, Y., 2014. As-rigid-as-possible stereo under second order smoothness priors. In: European Conference on Computer Vision. Springer, pp. 112–126.
- Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y., 2018b. Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 286–301.
- Zhang, M., Piggott, M.D., 2020. Unsupervised learning of particle image velocimetry. In: International Conference on High Performance Computing. Springer, pp. 102–115.
- Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y., 2018c. Residual dense network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2472–2481.
- Zhang, M., Wang, J., Thomole, J., Piggott, M.D., 2022. Learning to estimate and refine fluid motion with physical dynamics. arXiv preprint arXiv:2206.10480.
- Zhang, F., Woodford, O.J., Prisacariu, V.A., Torr, P.H., 2021. Separable flow: Learning motion cost volumes for optical flow estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10807–10817.
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L., 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* 26 (7), 3142–3155.
- Zhang, K., Zuo, W., Zhang, L., 2018d. Learning a single convolutional super-resolution network for multiple degradations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3262–3271.
- Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., Wang, J., 2017. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans. Ind. Electron.* 65 (2), 1539–1548.
- Zhong, Q., Yang, H., Yin, Z., 2017. An optical flow algorithm based on gradient constancy assumption for PIV image processing. *Meas. Sci. Technol.* 28 (5), 055208.