A CNN-based vortex identification method

Article *in* Journal of Visualization · October 2018

DOI: 10.1007/s12650-018-0523-1

Deng Liang
Computational Aerodynamics Institute China Aerodynamics Research & Develop…
37 PUBLICATIONS **372** CITATIONS

SEE PROFILE

Jie Liu
Shandong University of Science and Technology
52 PUBLICATIONS **435** CITATIONS

SEE PROFILE

J Vis (2019) 22:65–78
https://doi.org/10.1007/s12650-018-0523-1

REGULAR PAPER

Liang Deng · Yueqing Wang · Yang Liu · Fang Wang · Sikun Li · Jie Liu

A CNN-based vortex identification method

Received: 20 July 2018 / Accepted: 7 October 2018 / Published online: 22 October 2018

Abstract Vortex identification and visualization are important for understanding the underlying physical mechanism of the flow field and have been intensively studied recently. Local vortex identification methods could provide results in a rapid way, but they require the choice of a suitable criterion and threshold, which leads to poor robustness. Global vortex identification methods could obtain reliable results, while they require considerable user input and are computationally intractable for large-scale data sets. To address the problems described above, we present a novel vortex identification method based on the convolutional neural network (CNN). The proposed method integrates the advantages of both the local and global vortex identification methods to achieve higher precision and recall efficiently. In specific, the proposed method firstly obtains the labels of all grid points using a global and objective vortex identification method and then samples local patches around each point in the velocity field as the inputs of CNN. After that it trains the CNN to decide whether the central points of these patches belong to vortices. By this way, our method converts the vortex identification task to a binary classification problem, which could detect vortices quickly from the flow field in an objective and robust way. Extensive experimental results demonstrate the efficacy of our proposed method, and we expect this method can replace or supplement existing traditional methods.

Keywords Vortex identification · CNN · Unsteady flow field

1 Introduction

Vortex is a common and complex flow phenomenon, which plays a decisive role in mixing of fluid, energy transport, chemical reaction and combustion, aerodynamic noise generation and emission and shear stress changes (Jeong and Hussain 1995). As accurate identification and visualization of vortices are important for studying the law and mechanism of the flow field, diverse vortex identification methods based on associated

L. Deng (✉) · Y. Liu · S. Li · J. Liu
College of Computer, National University of Defense Technology, Changsha, China
E-mail: dengliang11@nudt.edu.cn

Y. Liu
E-mail: liuyang777@nudt.edu.cn

S. Li
E-mail: sikunli@126.com

J. Liu
E-mail: liujie@nudt.edu.cn

L. Deng · Y. Wang · Y. Liu · F. Wang
Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang, China
E-mail: yqwang2013@163.com

implicit vortex definition have been developed. These methods can be roughly classified into two categories: local methods and global methods (Jiang et al. 2005).

The local methods are typically based on physical properties of flow field and are conceptually easy to explain, such as 'locally high vorticity,' 'locally low pressure,' 'vorticity overtakes deformation,'. They only use the local flow information to compute some criterions like the Q-criterion (Hunt 1987), the Ω-criterion (Liu et al. 2016), the Δ-criterion (Chong et al. 1990) and the λ_2-criterion (Jeong and Hussain 1995), therefore obtaining the results efficiently. However, in practical applications, the local methods require to choose a suitable threshold carefully to produce effective results. Even so, there are many false positives and false negatives in the results (Wu et al. 2005).

Different from the local methods, the global methods are generally based on global topological properties of the flow field, such as winding angle (Sadarjoen et al. 2002), elliptic Objective Eulerian Coherent Structures (OECSs) (Serra and Haller 2016), Instantaneous Vorticity Deviation (IVD) (Haller et al. 2015). In these methods, the global flow information is used to detect vortical regions. Due to the objectiveness and robustness, these global methods are always used to verify the accuracy of the identification results. However, these methods are computationally intensive, so taking more time compared with local methods. Meanwhile, they need significant user intervention to obtain robust results. Therefore, it is difficult to apply these methods to identify vortical structures for large-scale data sets.

Since the local and global methods have their advantages and disadvantages, it is worth to combine advantages from both methods. There are a few works making efforts in this direction (Zhang et al. 2014; Biswas et al. 2015), but the generality and scalability of these works are poor. The appearance of convolutional neural network (CNN) provides a better way to solve these problems (Simard et al. 2003).

CNN is a class of machine learning algorithms which has made breakthrough innovation in the object recognition recently. Due to the locality of the convolutional operation and the translational invariance, CNN could effectively resist recognition difficulties such as translation, deformation and rotation. For these reasons, we believe that CNN is well suited for the task of vortex identification. Nevertheless, adapting CNN to the physical flow field still has some challenges. The first challenge is to obtain the labeled training data. As we know, CNN is a typical supervised learning algorithm, which requires a large amount of labeled data to train the network. However, it is difficult to get lots of labeled data in the vortex identification stage because there is no a rigorous mathematical definition of a vortex. The second challenge is to ensure the generality and scalability of the trained CNN models. In the field of computational fluid dynamics (CFD), the flow fields have diverse scales and shapes, which mean they cannot be directly regarded as the inputs of CNN model that needs inputs with a fixed size. Making the CNN effective on the flow fields with different scales and shapes is a challenge.

To address these challenges, this paper presents a robust and reliable vortex identification method based on CNN. In specific, we use a global method based on an objective vortex definition to get the labels of each flow field point. By this way, we could get $N_x \times N_y \times N_z$ labeled data on each flow field, where N_x, N_y and N_z are the number of points in each spatial dimension of the flow field. Meanwhile, we sample $N_{fx} \times N_{fy} \times N_{fz}$ neighborhood points around each point as its inputs by considering the local property of CFD computation, where N_{fx}, N_{fy} and N_{fz} represent the number of local-feature maps in each spatial dimension of the local receptive field. After that we could use these $N_x \times N_y \times N_z$ labeled patches to train the CNN. More importantly, we only use the velocity information of the flow field, which reduces the computation cost of the derived variables, consequently we can obtain the identified results in a rapid way. In addition, to improve the ability of our method in processing flow fields with different scales and shapes, we design the network for points rather than the entire flow fields. By this way, the CNN gets unrelated to the size and shape of the flow field and would further exhibit good generality and scalability.

The contributions of this work include:

- We present a novel vortex identification method based on CNN. By using the local flow information and the global attributes of the labels, our method could integrate the advantages of traditional global and local vortex identification methods.
- The proposed method utilizes CNN to learn a representation of the flow data and classifies points in flow field into the class 'vortex' or the class 'non-vortex.' By doing this, we convert the vortex identification task to a binary classification problem based on CNN, which can improve the generality and scalability of our method.
- Experimental results show that our method could detect vortices quickly from the flow field in an objective and robust way compared with traditional methods.

2 Related work

2.1 Local vortex identification methods

Given n-dimensional velocity field $\boldsymbol{u}$, the Jacobian of the velocity $\boldsymbol{J}$ is an $n \times n$ matrix that can be used for analyzing the characteristics of the flow patterns in a small zone around a given point. Many of the local vortex identification methods are based on the decomposition of the Jacobian matrix $\boldsymbol{J}$ into $\boldsymbol{J} = \boldsymbol{\Omega} + \boldsymbol{S}$, with

$$\boldsymbol{\Omega} = \frac{\boldsymbol{J} - \boldsymbol{J}^T}{2} \quad \boldsymbol{S} = \frac{\boldsymbol{J} + \boldsymbol{J}^T}{2} \tag{1}$$

where the anti-symmetric matrix $\boldsymbol{\Omega}$ is called the spin tensor and the symmetric matrix $\boldsymbol{S}$ is called the strain-rate tensor. The Q-criterion, Ω-criterion, Δ-criterion and λ_2-criterion are the most important local methods that only rely on the Jacobian $\boldsymbol{J}$. The Q-criterion of Hunt (1987) considers a connected region to be a vortex if

$$Q = \frac{1}{2}\left(\|\boldsymbol{\Omega}\|^2 - \|\boldsymbol{S}\|^2\right) > Q_{\text{thresh}} \tag{2}$$

where $\|\boldsymbol{\Omega}\|^2$ is the Euclidean norm of the spin tensor, $\|\boldsymbol{S}\|^2$ is the Euclidean norm of the strain-rate tensor, and Q_{thresh} is nominally equal to 0. Similar to the Q-criterion, Liu et al. (2016) present the Ω-criterion that defines a vortex region by

$$\Omega = \frac{\|\boldsymbol{\Omega}\|^2}{\|\boldsymbol{\Omega}\|^2 + \|\boldsymbol{S}\|^2} > \Omega_{\text{thresh}} \tag{3}$$

where Ω_{thresh} empirically is set to 0.52. The threshold 0.5 is equivalent to $Q > 0$. The Δ-criterion (Chong et al. 1990) is

$$\Delta = \left(\frac{Q}{3}\right)^3 + \left(\frac{\det\boldsymbol{J}}{2}\right)^2 > \Delta_{\text{thresh}} \tag{4}$$

where $\det\boldsymbol{J}$ is the determinant of $\boldsymbol{J}$ and Δ_{thresh} is theoretically set to 0. The condition $\Delta > 0$ shows that the Jacobian $\boldsymbol{J}$ has complex eigenvalues associated with the presence of vortical structures. Generally, $\Delta > 0$ is less restrictive than $Q > 0$, thus the Δ-criterion extracts larger vortex region (Chakraborty et al. 2005). The λ_2-criterion, introduced by Jeong and Hussain (1995), is based on the assumption that a pressure minimum should exist across a vortex core. It defines a vortex to be an area where the Hessian $(\boldsymbol{S}^2 + \boldsymbol{\Omega}^2)$ has at least two negative eigenvalues. This criterion is more reliable and effective in comparison with other local vortex identification methods (Schafhitzel et al. 2008).

All local vortex identification methods are threshold-dependent. The selection of an appropriate threshold plays a key role in determining the efficacy of a given vortex identification method. However, there is no impartial way to judge if thresholds of these local vortex identification methods are the 'best,' since the choice of the threshold requires domain knowledge of fluid mechanics and can rather be subjective (Günther and Theiselr 2018).

2.2 Global vortex identification methods

The global methods depend on considering the vortices as elliptic shapes and use closed contour approaches. Sadarjoen et al. (2002) use the winding angle method to extract vortices from the flow fields based on the geometry of streamlines. One shortcoming of the winding angle method is high computational cost, because it has to trace streamlines globally in the entire field. Another disadvantage is that this method cannot be applied into 3D flow fields. Serra and Haller (2016) define the boundaries of vortical structures (generalized elliptic OECSs) in 2D flows as outermost closed instantaneous curves for which the pointwise tangential strain rate is constant, and present an automated method for computing such curves efficiently based on the geometry of the underlying geodesic flow (Serra and Haller 2016). This approach requires significant numerical effort to compute null-geodesics of general Lorentzian metrics and is only to be used for 2D flows. Haller et al. (2015) view vortices as sets of tubular surfaces of constant intrinsic material rotation rate, and propose IVD, a threshold-independent vortex indentification technique. The IVD is defined as the absolute value of the difference between vorticity at a point and the spatially averaged vorticity of the global field:

$$IVD(\boldsymbol{x}, t) = |\boldsymbol{\omega}(\boldsymbol{x}, t) - \boldsymbol{\omega}_{\text{avg}}(\boldsymbol{x}, t)| \quad (5)$$

where $\boldsymbol{\omega}$ represents the vorticity value. By the calculation of Eq. (5), the IVD field is easily obtained. The challenge is to select the IVD-based vortex boundaries so that it encloses an impermeable material region of concentrated IVD. Algorithm 1 gives the detailed solution procedure to find objective vortex boundaries for 2D flows. For 3D flows, vortex boundaries are done by locating and connecting N parallel 2D planes. A Matlab implementation of IVD method is publicly available at: https://github.com/Hadjighasem/Lagrangian-Averaged-Vorticity-Deviation-LAVD.

Algorithm 1 IVD-based vortex boundaries for 2D flows.

```
Input:    A 2D velocity field
1: Calculate the IVD($\boldsymbol{x}$,t) with Eq.5.
2: Identify local maxima of the IVD field.
3: Use the level set method to seek vortex boundaries as outermost,
   closed contours of IVD($\boldsymbol{x}$,t) satisfying all the following:
   (a) Each vortex boundaries encircles a local maxima of IVD($\boldsymbol{x}$,t).
   (b) Each vortex boundaries has arclength exceeding the minimal
   vortex perimeter.
   (c) Each vortex boundaries has convexity deficiency less than $10^{-3}$.

Output:    Positions of vortex boundaries (closed curves) and vortex
   centers (isolated points).
```

In above global methods, the IVD is an objective technique that can be applicable in arbitrary translation and rotation reference frame. Meanwhile, the IVD has ability to detect vortical features in 2D and 3D flows more quickly and accurately (Mattia and George 2016). Therefore, this paper uses the IVD to obtain the labels of each point in the flow field.

2.3 Machine learning methods for vortex identification

Machine learning methods are getting an increasing attention in the vortex identification and visualization problems (Zhang et al. 2014; Biswas et al. 2015). These methods leverage multiple local methods to construct a more accurate and robust method. Zhang et al. (2014) employ the adaptive boosting (*AdaBoost*) Freund and Schapire (1997) method to assign different weights to four different local vortex detectors based on expert information, and obtain a reduced misclassification rate in two CFD data sets. Instead of assigning different weights to the individual detectors, Biswas et al. (2015) assign equal weights and introduce a fuzzy-based approach to combine uncertainty in the outputs of four existing local vortex detection methods. For the Tapered Cylinder data set, the error rate of Biswas et al. (2015) and *AdaBoost* are about 31 and 38%, respectively. Compared with the *AdaBoost* method, Biswas et al. (2015) can provide more robust and reliable identification results. In summary, by reducing false positives and false negatives, these methods can enhance the accuracy of vortex identification results to some degree, but increase the computation cost because of using multiple local vortex identification methods. In addition, these methods heavily depend on the labeled data of domain experts to optimize the models. What is worse, these methods are related to the size and shape of flow field, so the generality and scalability of them are poor.

With the development of the CNN, there are also some works using the CNN to detect vortical structures in flow field (Lguensat et al. 2017; Ströfer et al. 2018; Franz et al. 2018; Bin and Yi 2018). Lguensat et al. (2017) develop EddyNet a deep neural network architecture for pixel-wise classification of oceanic eddies. The global accuracy of their proposed method is always higher than 88% for Sea Surface Height (SSH) maps provided by the Copernicus Marine and Environment Monitoring Service (CMEMS). However, EddNet directly regards the whole visualization images of the flow field as their inputs, which discards the characteristics of the original flow field. Based on the object detection R-CNN method by (Girshick et al. 2014; Ren et al. 2015), Ströfer et al. (2018) present Fluid R-CNN, and adapt it to the problem of identifying the horseshoe vortex in a 3D flow over a wing-body junction. The Fluid R-CNN can identify most of the horseshoe vortex in downstream of the airfoil, but also generates a large number of false positives and false negatives around the airfoil and in the wall near the inlet, respectively. Moreover, this work is high computational complexity because it requires the ten Galilean invariant inputs of each point in the flow field. What is worse, the method is restricted to the uniform mesh with single-connected domains, which leads to the generality lackness. Franz et al. (2018) present a mesoscale ocean eddy detection and tracking framework based on a CNN with the Kanade–Lucas–Tomasi (KLT) (Lucas and Kanade 1981) feature tracker. Although this work can obtain reliables and robust identified results on global sea level anomaly maps (SLA) data using the output of a local eddy detection method as training data for the CNN, they are currently not precise enough to be used. Bin and Yi (2018) provide a universal feature visualization method by using CNN to recognize and locate features. The precision ratio of the CNN method is always more than 96% for 2D vortex feature sample data. However, this approach needs many user interventions to select the feature sample data by domain-expert depending on requirments which leads to subjective identified results.

In our work, we obtain the labeled training data based on an objective and global vortex identification method and only use the velocity components as the inputs of the CNN. Besides, In order to solve the generality and scalability, we design the network for points rather than the whole flow field.

3 Our proposed method

The CNN-based vortex identification method we proposed includes two parts: the pre-processing part and the network part, as illustrated in Fig. 1. The pre-processing part provides the data inputs of the second part, and the network part, named Vortex-Net, trains a CNN model and uses it to recognize vortical structures in the flow field.

3.1 Pre-processing

In this section, we would introduce the pre-processing part of our method. This part includes four steps: transforming mesh, processing boundary, obtaining labels and sampling patches.

Firstly, we transform the non-uniform mesh in the physical plane into the uniform mesh in the computational plane, as shown in Fig. 2. The uniform mesh can be directly expressed as a rectangular array, and each mesh point has associated location information and velocity components in the Cartesian coordinate system. Therefore, we can easily sample data with a fixed size in the computational plane without paying attention to the original flow field.

Secondly, we process the boundary of the mesh. The Vortex-Net uses local patch of each point in the flow field as its representation, but the points close to the boundary have no enough neighbors to get the local patch. To solve this problem, we pad the mesh by combining properties of the original flow field. As shown in Fig. 2, the points close to the left and right boundary are interior points, so we use the velocity components of left points to pad the right side, and pad the velocity components of right points to the left side. The upper boundary is the surface of the object, which can be padded by the mirror way. The lower boundary is the far-field boundary, which would not have major effects on the results, and we use the zero value to pad the lower boundary. After boundary processing, the size of flow field is

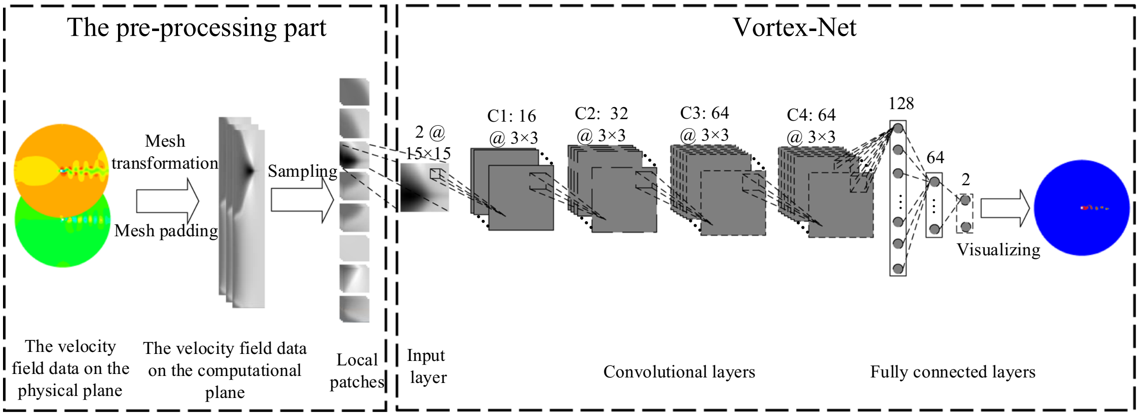


Fig. 1 The architecture of our proposed method. For clarity of illustration, the velocity field is only shown as having two physical dimensions

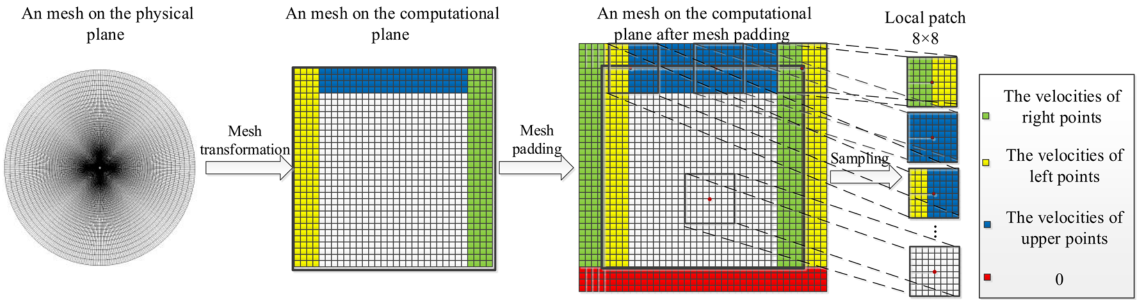

An mesh on the physical plane
Mesh transformation
An mesh on the computational plane
Mesh padding
An mesh on the computational plane after mesh padding
Sampling
Local patch 8×8
🟩 The velocities of right points
🟨 The velocities of left points
🟦 The velocities of upper points
🟥 0

Fig. 2 A detailed description of the pre-processing part

$(2n + N_i) \times (2n + N_j) \times (2n + N_k)$, where n is the number of padding layers and N_i, N_j, N_k are the number of points in each spatial dimension of original flow field.

Thirdly, we use a global vortex method named IVD to label all points in the flow field, where ‘0’ represents the non-vortex point and ‘1’ represents the vortex point. Through this way, we introduce the global information into our method.

Finally, the normalization method is shown as Eq. (6)

$$\begin{aligned} u &= \frac{(u_0 - u_{\min})}{(u_{\max} - u_{\min})}, \\ v &= \frac{(v_0 - v_{\min})}{(v_{\max} - v_{\min})}, \\ w &= \frac{(w_0 - w_{\min})}{(w_{\max} - w_{\min})} \end{aligned} \tag{6}$$

where u, v and w are the normalized velocity components of each points, u_0, v_0 and w_0 are the velocity components of each points in the original velocity field, $u_{\min}$, $u_{\max}$, $v_{\min}$, $v_{\max}$, $w_{\min}$, $w_{\max}$ are the maximum and minimum of the velocity components of each points in the original velocity field, respectively. Then, we sample local patches around each points of the normalized velocity field, and take these patches and labels as the inputs of Vortex-Net, which makes the method suitable for different scales and shapes of the flow field. Moreover, by using global labels and local patches to train the network, our method could obtain an approximate result as global methods using the local information in the testing stage.

3.2 Vortex-Net

Following the pre-processing part, each point must be classified. The classification task is done by the second part of our method, called Vortex-Net, which trains a CNN model to differentiate non-vortex points and vortex points. In this section, we will first present the structure of our Vortex-Net. The structure must be flexible enough to make the classification task possible and simple enough to keep the number of training parameters and computational cost low. Then, we will give a brief introduction of Vortex-Net training and testing stage. In the Vortex-Net training stage, the optimal values for the trainable parameters are learned by the labeled data. The vortex identification results are obtained through the well-trained CNN model in the Vortex-Net testing stage.

3.2.1 The structure of Vortex-Net

The structure of our Vortex-Net includes a single input layer, four convolutional layers and three fully connected layers, as shown in Fig. 1. For clarity of illustration, we show an input with only two velocity components in the figure.

The first layer is the input layer which corresponds to a patch in the flow field. The size of this input layer is 2 × 15 × 15, where 2 is the number of channels and 15 represents the size of local patches. The second to the fifth layer are convolutional layers. The number of feature maps of these layers is 16, 32, 64 and 64, respectively. All of these convolutional layers use 3 × 3 convolutional kernels. These kernels need to be determined through the Vortex-Net training stage. Moreover, they are the same for all spatial locations, which enforces translational invariance and greatly reduces the number of trainable parameters. The activation function used in these convolutional layers is rectified linear unit (ReLU), as shown in Eq. (7).

$$y = \max(0, x) \tag{7}$$

where y is the output of the activation function, and x means the output of the convolutional layer. The sixth to the eighth layer are the fully connected layer, which have 128, 64 and 2 neurons, respectively. The final fully connected layer is the output layer which classifies the input as belonging to the class 'vortex' or to the class 'non-vortex.' In our Vortex-Net, the output layer is a *softmax* layer, and its size is equal to the number of categories being considered. Each neuron in a *softmax* layer represents a probability of the input belonging to one category, with the sum of all the outputs equal to one. The output of the *softmax* neuron is given by

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}} \tag{8}$$

$$z_j = \sum_{k=1}^{N_{\text{prev}}} (w_{k,j} x_k) + b \tag{9}$$

where m means the number of classes, z_j represents the output of the last layer, N_{prev} is the number of neurons of previous layer, x_k and $w_{k,j}$ are the output and weight from the kth neuron of the previous layer to the jth neuron of the *softmax* layer.

3.2.2 Vortex-Net training

After obtaining the labeled data in the pre-processing stage, we train the Vortex-Net to optimize the Vortex-Net parameters. In this study, we use *Adam* (Kingma and Ba 2014) and back-propagation with the cross-entropy cost function to train the network. The cross-entropy cost function is given by

$$C = -\frac{1}{n} \sum_{x} [y \ln a + (1 - y) \ln(1 - a)] \tag{10}$$

where y and a are the predicted and true output of the neuron, x is the number of training samples, and n is the total number of training samples. *Adam* can obtain individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients of C to minimize the *loss* function. Back-propagation uses the chain rule to iteratively compute the gradients for each layer, starting from the output layer.

In the testing stage, we first fix the network and directly regard the testing patches as the inputs. Then, the vortex identification results are obtained by using the inputs to pass through the well-trained Vortex-Net. Finally, we assemble the vortex identification results into the entire flow domain for visualization.

4 Experimental results and discussions

In this section, we compare our method with other eight methods, including four popular local vortex identification methods, two conventional machine learning algorithms, one deep learning technique and the IVD method. The local methods are the Q-criterion, Ω-criterion, Δ-criterion and λ_2-criterion. All these methods need to select a proper threshold to produce a relatively good result. In our experiment, we choose the threshold of each local approach by matching the identified result with that of IVD. The conventional machine learning algorithms are k-nearest neighbors (KNN) (Cover and Hart 1967) and *AdaBoost*. The number of neighbors in KNN is set by using the cross-validation method. The *AdaBoost* is a parameter-free algorithm. Another deep learning technique is Multi-Layer Perceptron (MLP) which is a feed forward neural network model. As shown in Fig. 3. The MLP used in this paper contains four fully connected layers, which can also be divided into five layers, i.e., the input layer I, three hidden layer H_1, H_2, H_3, and the output layer O. In our experiment, we only need to replace the Vortex-Net with the structure of MLP due to the highly modular of our proposed method. Thus, the size of input is the same as our approach for the input layer I and the actual number of neuron nodes is 450. The number of neuron nodes of three hidden layers is 256, 128

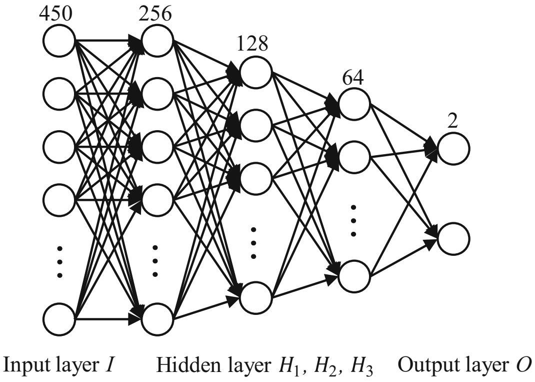


Fig. 3 The structure of MLP

and 64, respectively. For the output layer O, the number of nodes equals the actual number of classes, which is 2.

In this paper, we use three classical metrics to measure the performance of each method, including the precision, the recall and the execution time. The precision is the fraction of identified vortical points that are correct, while the recall is the fraction of the true vortical points that are successfully detected. They are calculated using Eqs. (11) and (12):

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{12}$$

where FP, FN, TP and TN represent the number of false positives, false negatives, true positives and true negatives, respectively. When analyzing the execution time, we neglect the training cost of Vortex-Net because the trained model can be used in the future recognition process.

We evaluate each method on several 2D and 3D flow cases detailed in Table 1. The 2D flow cases include: the cylinder flow field, the plate flow field, the square flow field and the triangle flow field. The 3D flow cases consist of the shockwave vortex(SV) interaction flow field (Zhang et al. 2009) with different size.

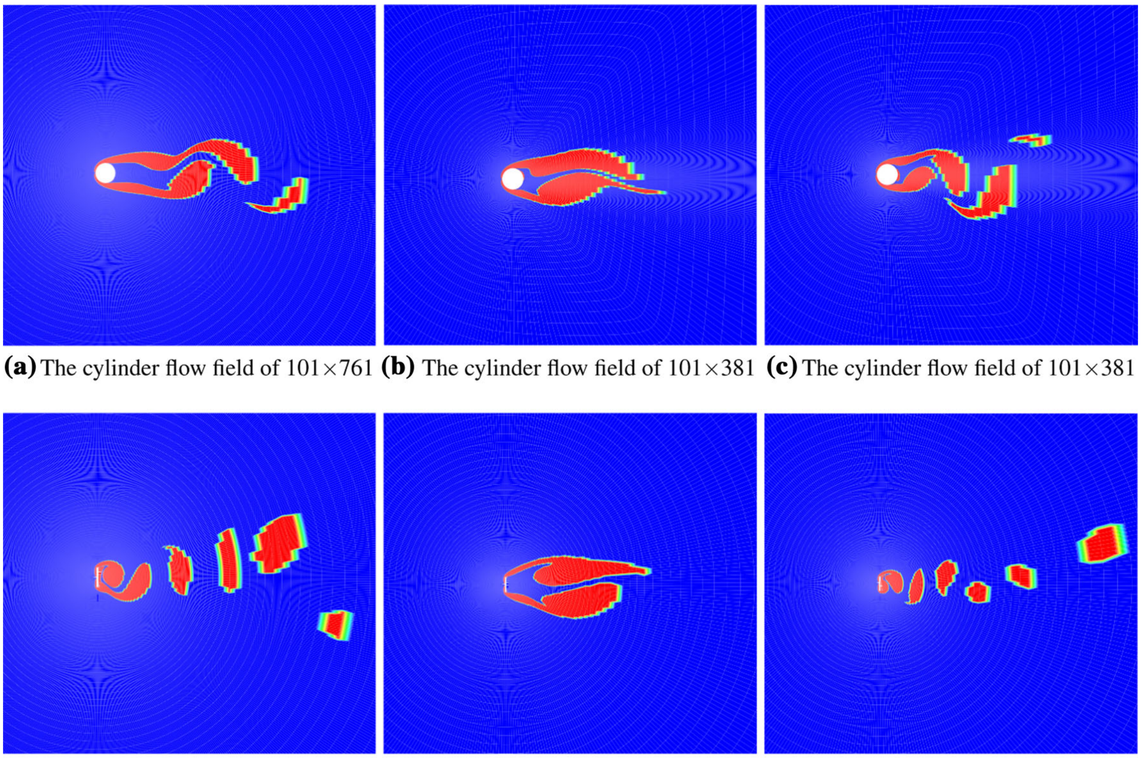
The implementations for local methods and the IVD method are realized in *Matlab*. Our method, the conventional machine learning algorithms and MLP are implemented in *Python*. All these nine methods run in a desktop computer which has an Intel Xeon CPU E5-2650v3 @2.30 GHz with 64 GB memory.

4.1 2D flow cases

This subsection involves identifying the vortex in 2D flow cases. The Vortex-Net is trained using six flow fields: the cylinder flow field with 101 × 761 grid size, the cylinder flow field with 101 × 381 grid size, the cylinder flow field with 101 × 381 grid size on different time step, the plate flow field with 101 × 761 grid size, the plate flow field with 101 × 381 grid size, and the plate flow field with 101 × 381 grid size on different time step. The total number of labeled data is $3.0 \times 10^5 (2 \times (101 \times 761 + 101 \times$

Table 1 The flow cases used in our experiment

Name	Grid size
2D	
Cylinder	101 × 381
Cylinder	101 × 761
Plate	101 × 381
Plate	101 × 761
Square	101 × 381
Triangle	101 × 761
3D	
SV-81	81 × 81 × 81
SV-161	161 × 81 × 81



(a) The cylinder flow field of 101×761 **(b)** The cylinder flow field of 101×381 **(c)** The cylinder flow field of 101×381

(d) The plate flow field of 101×761 **(e)** The plate flow field of 101×381 **(f)** The plate flow field of 101×381

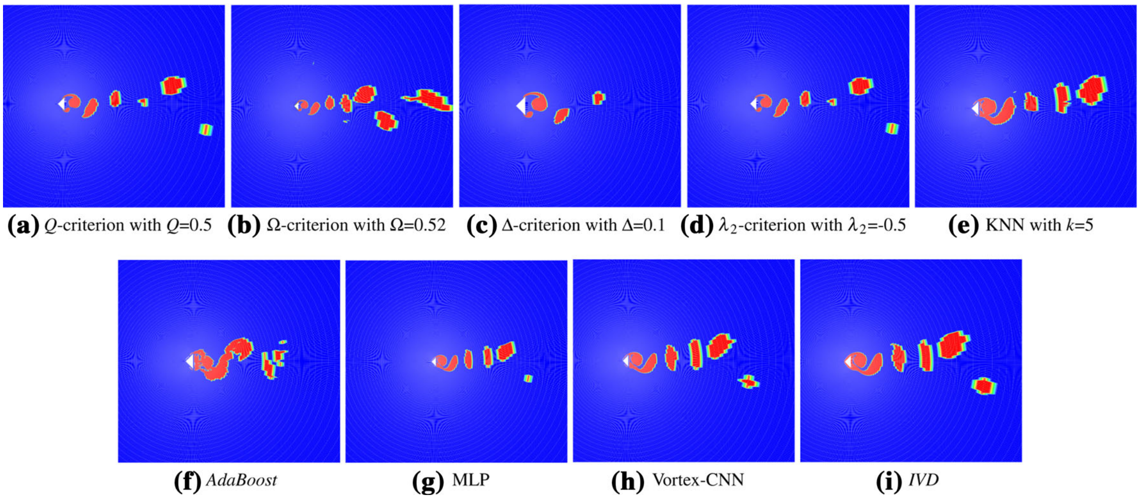
Fig. 4 Visual results for the 2D training cases. Color map with red and blue denote the vortex region and the non-vortex region. **b**, **c** The vortical structures in the same cylinder flow field on different time step. **e**, **f** The vortical structures in the same plate flow field on different time step

381 + 101 × 381)) and the labeled vortex regions for each of the training cases are shown in Fig. 4. In order to demonstrate the generality and scalability of our method, it is an intuitive idea to test this approach on the flow field with different sizes and shapes. Limited by space, here our propose approach is only tested on two new different flows, including the square flow field with 101 × 381 grid size and the triangle flow field with 101 × 761 grid size. Then, this method is tested on the same flow with different grid size in Sect. 4.2.

Table 2 reports the precision, the recall and the execution time of the different methods on the triangle and square flow field. For convenience, we use 'Vortex-CNN' to represent our proposed method. In each

Table 2 The performance of different methods in the triangle flow field of 101 × 761 and the square flow field of 101 × 381

Cases	Methods	Precision (%)	Recall (%)	Execution time (s)
Triangle	Q-criterion	78.21	68.14	5.34
	Ω-criterion	80.24	77.13	6.12
	Δ-criterion	84.23	45.51	7.31
	λ_2-criterion	79.11	69.01	30.25
	KNN	94.35	89.71	69.83
	AdaBoost	89.35	83.15	2.03
	MLP	98.34	96.74	49.29
	Vortex-CNN	99.12	97.56	53.67
	IVD	100	100	345.76
Square	Q-criterion	75.16	83.35	2.76
	Ω-criterion	41.34	81.23	3.14
	Δ-criterion	91.23	34.12	3.76
	λ_2-criterion	77.15	84.03	20.15
	KNN	91.12	93.12	35.41
	AdaBoost	89.56	90.12	0.61
	MLP	95.34	96.67	24.32
	Vortex-CNN	98.12	99.24	26.51
	IVD	100	100	183.14



(a) Q-criterion with Q=0.5 **(b)** Ω-criterion with Ω=0.52 **(c)** Δ-criterion with Δ=0.1 **(d)** λ_2-criterion with λ_2=-0.5 **(e)** KNN with k=5

(f) *AdaBoost* **(g)** MLP **(h)** Vortex-CNN **(i)** *IVD*

Fig. 5 Visual results for different methods in the 101 × 761 triangle flow field. Images **a**–**d** show the identified vortex (red) of the local methods. **e**, **f** Depict the identified vortex of the conventional machine learning algorithms. **g** The identified vortex of multi-layer perceptron (MLP). **h** The identified vortex of our proposed method. **i** The identified vortex of a global approach (IVD)

comparisons except IVD, the Vortex-CNN can achieve the best precision and recall. However, it also requires more execution time than any other method. Compared with the IVD method, the Vortex-CNN can obtain 6.4× speedup on the triangle case and 6.9× speedup in the square flow field.

For the precision ratio, on the triangle case, the accuracy of Vortex-CNN is on average 18.6% higher than that of the local methods, 7.3% higher than that of the conventional machine learning methods and 0.7% higher than that of MLP. On the square case, the accuracy of Vortex-CNN exceeds that of the local methods, the conventional machine learning methods and MLP by 26.9%, 7.8% and 2.7% on average, respectively.

For the recall ratio, the Vortex-CNN is always more than 97.5%, which is superior to the MLP on all cases. In contrast to the local methods, the Vortex-CNN can obtain average recall improvement by almost 32.6 and 28.5% on these two cases. Meanwhile, the recall of our proposed method can provide an average of almost 11.1% improvement in the triangle case and 7.6% improvement in the square case using the conventional machine learning methods.

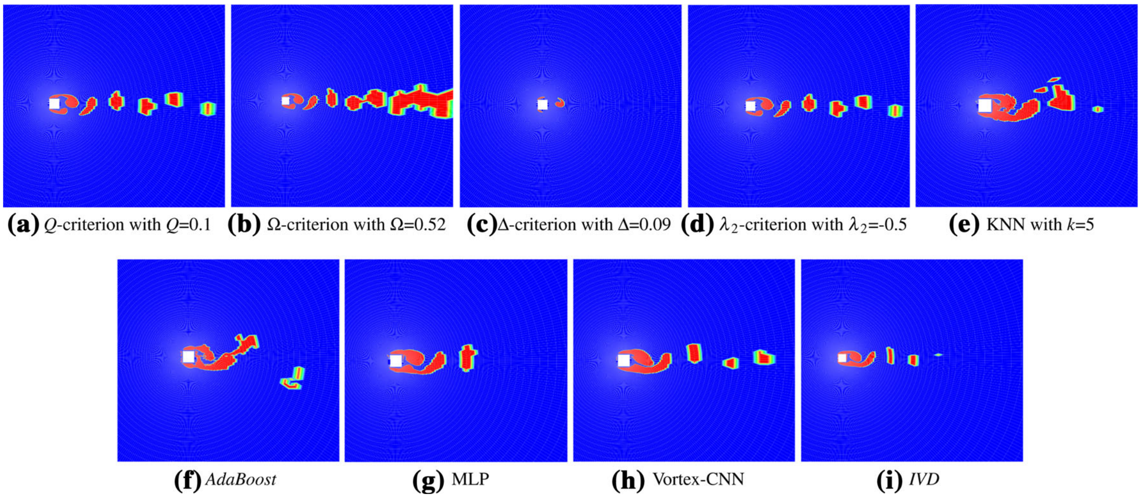
As can be seen from the comparison of the precision and recall, the local methods are difficult to obtain a high precision and recall concurrently on test cases. For example, on the square case, the precision of Δ-criterion is 91.23%, while the recall is only 34.12%, which means the Δ-criterion has lots of false negatives. By contrast, the MLP and Vortex-CNN could achieve a high precision as well as high recall at the same time.

In terms of the execution time, the Vortex-CNN runs slower than the local methods. However, all these local methods require careful choice of the threshold which is an iterative process. Therefore, the overall execution time of these local methods cannot be measured accurately. In contrast to MLP, the Vortex-CNN performs slightly worse in all case. Compared with the IVD method, the speedup ratio of Vortex-CNN can reach 6.4 in the triangle flow field and 6.9 in the square flow field.

In the CFD visualization, the precision and recall cannot reflect the flow phenomenon in detail, so we visualize the identified results, as shown in Figs. 5 and 6. There are lots of missing or false detection vortical structures for the local methods. Although the conventional machine learning methods have high precision and high recall, they cannot reflect the vortex separation in the flow field, as illustrated in Figs. 5f and 6e. For our proposed method, the visualized vortical structures are consistent with the IVD method and better than MLP. Meanwhile, the Vortex-CNN can accurately reflect vortex shedding phenomenon in the flow field.

4.2 3D flow cases

Vortex is a unique product in the 3D flow field, and the 2D flow field is usually only applied to the preliminary design to quickly view the flow mechanism. Therefore, we also add 3D cases to verify the effectiveness of our method.


(**a**) Q-criterion with Q=0.1 (**b**) Ω-criterion with Ω=0.52 (**c**) Δ-criterion with Δ=0.09 (**d**) λ_2-criterion with λ_2=-0.5 (**e**) KNN with k=5

(**f**) *AdaBoost* (**g**) MLP (**h**) Vortex-CNN (**i**) *IVD*

Fig. 6 Visual results for different methods in the 101 × 381 square flow field

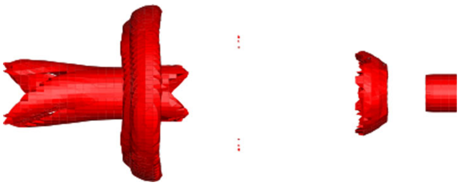


Fig. 7 The 3D training cases. For clarity of illustration, the figure only shows the labeled vortex region (red)

The Vortex-Net is trained using the shockwave vortex interaction flow field of 81 × 81 × 81. The total number of labeled data is 5.3×10^5 (81 × 81 × 81) and the labeled vortex regions for the training case is shown in Fig. 7. To testify the efficacy of our method on the same flow field, we make a series of experiments on SV which has a grid size of 81 × 81 × 81 (SV-81) on different time step and a grid size of 161 × 81 × 81 (SV-161).

Table 3 shows the precision, the recall and the execution time of the different methods on SV cases. The precision of the Vortex-CNN can reach to 98.25 and 99.01% in the SV-81 and SV-161, respectively. In specific, the precision of the Vortex-CNN increases by 15.8%, 16.8% and 1.9% on average, when compared with that of the local methods, the conventional machine learning algorithms and MLP in the SV-81. In SV-

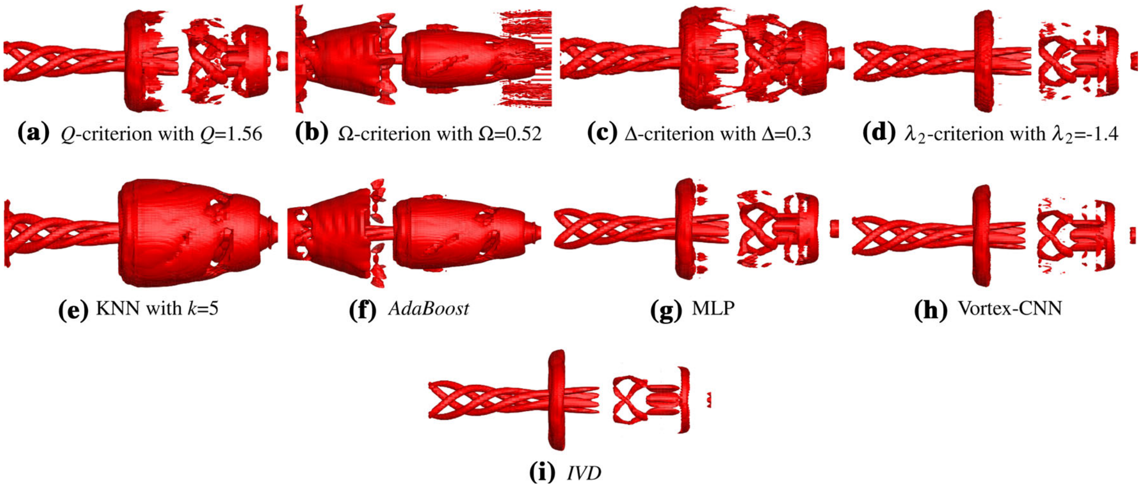
Table 3 The performance of different methods in the shockwave vortex interaction flow field

Cases	Methods	Precision (%)	Recall (%)	Execution time (s)
SV-81	Q-criterion	88.79	90.23	35.21
	Ω-criterion	71.23	87.13	40.01
	Δ-criterion	80.12	88.79	48.42
	λ_2-criterion	89.32	93.21	215.23
	KNN	90.42	94.71	487.45
	AdaBoost	72.35	88.35	15.12
	MLP	96.35	97.56	321.57
	Vortex-CNN	98.25	99.56	342.16
	IVD	100	100	2545.76
SV-161	Q-criterion	89.17	91.24	72.15
	Ω-criterion	74.34	89.23	79.48
	Δ-criterion	81.32	90.02	82.65
	λ_2-criterion	91.23	94.25	335.24
	KNN	90.74	96.31	582.03
	AdaBoost	75.23	90.41	18.31
	MLP	98.13	98.25	597.23
	Vortex-CNN	99.01	99.83	610.42
	IVD	100	100	4920.53

161, the Vortex-CNN preforms no worse which is on average 14.9% better than the local methods, 16.0% higher than the conventional machine learning algorithms and 0.8% better than MLP. In terms of the recall, the Vortex-CNN is always higher than 99.5% which is superior to the MLP on all SV cases. In contrast to the local methods, the Vortex-CNN can obtain an average of almost 9.7% and 8.6% improvement in the SV-81 and SV-161, respectively. Meanwhile, our proposed method can provide an average of almost 8.0% improvement in the SV-81 and 6.4% improvement in the SV-161 over the conventional machine learning algorithms. Taking the precision and the recall into consideration, the local methods are difficult to obtain a high precision and also a high recall in our experiment. For example, the precision of Ω-criterion is 71.23%, while the recall of it is 87.13%, which demonstrates that the Ω-criterion has lots of false positives and false negatives. However, the Vortex-CNN has a high precision as well as high recall in every case by drastically reducing the number of the false positives and the false negatives.

For the execution time, the Vortex-CNN is similar to 2D cases, which runs slower than the local methods. The execution time of the IVD method on SV-81 and SV-161 cases is 2545.76 and 4920.53 s, which is unbearable for the vortex identification task. Compared with the IVD method, our proposed method can achieve about 7.4× speedup in the SV-81 and 8.0× in the SV-161. Therefore, our method is more suitable than other traditional vortex identification methods on large-scale cases.

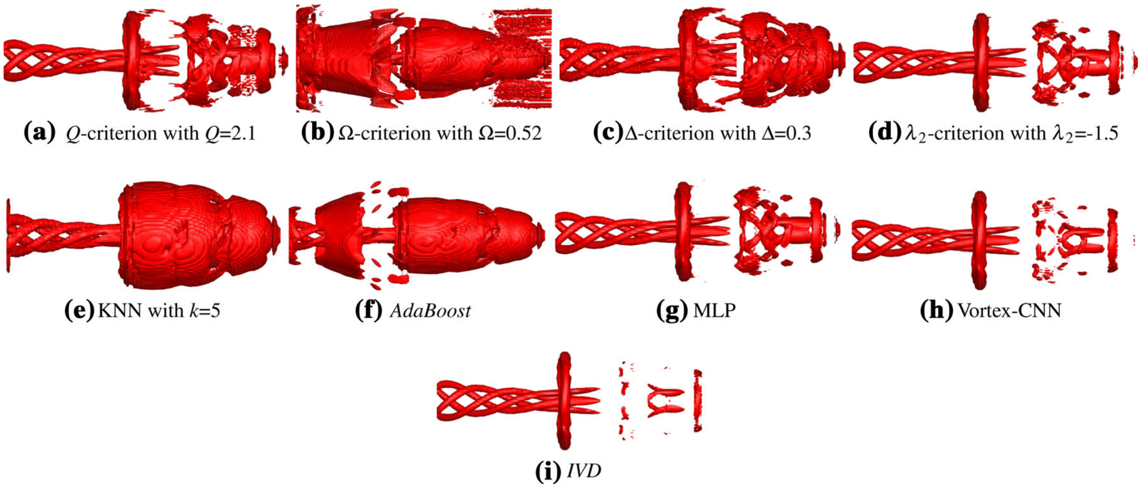
The identified vortex in the SV of different methods are shown in Figs. 8 and 9. For the local methods, there are many false positives and false negatives in the identified results. And for the conventional machine


(a) Q-criterion with Q=1.56 **(b)** Ω-criterion with Ω=0.52 **(c)** Δ-criterion with Δ=0.3 **(d)** λ_2-criterion with λ_2=-1.4

(e) KNN with k=5 **(f)** *AdaBoost* **(g)** MLP **(h)** Vortex-CNN

(i) *IVD*

Fig. 8 Visual results for different methods in the 81 × 81 × 81 shockwave vortex interaction flow field on different time steps


(a) Q-criterion with Q=2.1 **(b)** Ω-criterion with Ω=0.52 **(c)** Δ-criterion with Δ=0.3 **(d)** λ_2-criterion with λ_2=-1.5

(e) KNN with k=5 **(f)** *AdaBoost* **(g)** MLP **(h)** Vortex-CNN

(i) *IVD*

Fig. 9 Visual results for different methods in the 161 × 81 × 81 shockwave vortex interaction flow field

learning algorithms which has a high precision and high recall, but they cannot reflect the vortex separation in the flow field, as illustrated in Figs. 8f and 9e. The Vortex-CNN is able to accurately identify almost all of the vortical structures and better than MLP. Moreover, our approach can accurately reflect the flow separation in the flow field.

4.3 Discussions

The goal of this study is to address the issues of the local and global vortex detection methods by adopting CNN. The results on 2D and 3D flow cases show that the Vortex-CNN can identify vortices in a way that has higher precision as well as higher recall than the local methods and faster speed than a global method. Meanwhile, our approach is threshold-independent and can provide vortex identification results in an objective and robust way. In contrast to the conventional machine learning algorithms, the Vortex-CNN can accurately detect the vortical structures and reveal the flow phenomena in the flow fields. And the Vortex-CNN has less false positives and false negatives than MLP. More importantly, the Vortex-Net trained by one case can be applied to other cases directly, so that other cases can avoid long training and exhibit good generality and scalability.

There are however some important works needing to be improved in the future. In order to make our method more practical, we would consider the labeled data of domain experts for training. Meanwhile, the Vortex-CNN can be enhanced by using deeper CNN architecture. In the design and implementation of the Vortex-CNN, we take a highly modular approach. Thus, we can easily integrate these new improved algorithms into the Vortex-CNN.

Feature identification and tracking are two closely related research areas in the time-dependent flow fields. Feature tracking is the correspondence problem of the identified feature in the time axis. The presented method can be used directly for identifying vortical structures at individual time-steps in time-dependent flows. For describing the evolution of the vortex feature learning by Vortex-Net, we can utilize a feature tracker with sparse optical flow estimation.

5 Conclusion

In this paper, we propose a CNN-based vortex identification method and provide the structure of it. The novelty of this study is that local and global information of flow field is combined into the proposed Vortex-CNN, so it can exploit the advantages of traditional global and local vortex identification methods. In specific, the proposed method can achieve a higher precision as well as higher recall than the local methods and obtains a speedup of more than 6× over the global and objective method. In contrast to the conventional machine learning algorithms and MLP, the approach has a higher performance and achieves good visual effects.

Future work will focus on improving the performance by using a deeper CNN architecture and incorporating the method for vortex feature tracking in time-dependent flows.

Acknowledgements This work was supported in part by the National Key Research and Development Program of China (# 2016YFB0200701) and the National Natural Science Foundation of China (# 61806205, # 91530324 and # 91430218).

References

Bin T, Yi L (2018) CNN based flow field feature visualization method. Int J Perform Eng 14(3):434–444. https://doi.org/10.23940/ijpe.18.03

Biswas A, Thompson D, He W, Deng Q, Chen C-M, Shen H-W, Machiraju R, Rangarajan A (2015) An uncertainty-driven approach to vortex analysis using oracle consensus and spatial proximity. In: IEEE pacific visualization symposium, IEEE Computer Society, Los Alamitos, pp 1–8. https://doi.org/10.1109/PACIFICVIS.2015.715638

Chakraborty P, Balachandar S, Adrian RJ (2005) On the relationships between local vortex identification schemes. J Fluid Mech 535(4):189–214. https://doi.org/10.1017/S0022112005004726

Chong MS, Perry AE, Cantwell BJ (1990) A general classification of three-dimensional flow fields. Phys Fluids A 2(5):765–777. https://doi.org/10.1063/1.857730

Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27. https://doi.org/10.1109/TIT.1967.1053964

Franz K, Roscher R, Milioto A, Wenzel S, Kusche J (2018) Ocean eddy identification and tracking using neural networks. In: Computer vision and pattern recognition

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139. https://doi.org/10.1006/jcss.1997.1504

Girshick RB, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE conference on computer vision and pattern recognition proceedings of the 2014 CVPR ’14. IEEE Computer Society, Los Alamitos, pp 580–587

Günther T, Theiselr H (2018) The state of the art in vortex extraction. Comput Graph Forum 1:1–24. https://doi.org/10.1111/cgf.13319

Haller G, Hadjighasem A, Farazmand M, Huhn F (2015) Defining coherent vortices objectively from the vorticity. J Fluid Mech 795(7):136–173. https://doi.org/10.1017/jfm.2016.151

Hunt JCR (1987) Vorticity and vortex dynamics in complex turbulent flows. Trans Can Soc Mech Eng 11(1):21–35. https://doi.org/10.1139/tcsme-1987-0004

Jeong J, Hussain F (1995) On the identification of a vortex. J Fluid Mech 285(1):69–94. https://doi.org/10.1017/S0022112095000462

Jiang M, Machiraju R, Thompson D (2005) Detection and visualization of vortices. In: Visualization handbook, pp 295–309. https://doi.org/10.1016/B978-012387582-2/50016-2

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. In: Computer sciences

Lguensat R, Sun M, Fablet R, Mason E, Tandeo P, Chen G (2017) Eddynet: a deep neural network for pixel-wise classification of oceanic eddies. In: Computer vision and pattern recognition

Liu CQ, Wang YQ, Yang Y, Duan ZW (2016) New omega vortex identification method. Sci China Phys Mech Astron 59(8):684–711. https://doi.org/10.1007/s11433-016-0022-6

Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: International joint conference on artificial intelligence. Elsevier, Amsterdam, pp 674–679

Mattia S, George H (2016) Forecasting long-lived lagrangian vortices from their objective eulerian footprints. J Fluid Mech 813:436–457. https://doi.org/10.1017/jfm.2016.865

Ren S, Girshick R, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sadarjoen A, Post FH, Ma B, Banks DC, Pagendarm HG (2002) Selective visualization of vortices in hydrodynamic flows. In: Proceedings of visualization. IEEE Computer Society, Los Alamitos. pp 419–422, https://doi.org/10.1109/VISUAL.1998.745333

Schafhitzel T, Vollrath J, Gois J, Weiskopf D, Castelo A, Ertl T (2008) Topology-preserving λ_2-based vortex core line detection for flow visualization. Comput Graph Forum 27:1023–1030. https://doi.org/10.1111/j.1467-8659.2008.01238.x

Serra M, Haller G (2016) Efficient computation of null-geodesic with applications to coherent vortex detection. Proc R Soc A: Math Phys Eng Sci 473:1–18. https://doi.org/10.1098/rspa.2016.0807

Serra M, Haller G (2016) Objective eulerian coherent structures. Chaos Interdiscip J Nonlinear Sci 26(5):95–105. https://doi.org/10.1063/1.4951720

Simard PY, Steinkraus D, Platt JC (2003) Best practices for convolutional neural networks applied to visual document analysis. In: International conference on document analysis and recognition. IEEE Computer Society, Los Alamitos, pp 958–963, https://doi.org/10.1109/ICDAR.2003.1227801

Ströfer CM, Wu J, Xiao H, Paterson E (2018) Data-driven, physics-based feature extraction from fluid flow fields. In: Fluid dynamics

Wu JZ, Xiong AK, Yang YT (2005) Axial stretching and vortex definition. Phys Fluids 17(3):69–78. https://doi.org/10.1063/1.1863284

Zhang L, Deng Q, Machiraju R, Rangarajan A, Thompson D, Walters DK, Shen H (2014) Boosting techniques for physics-based vortex detection. Comput Graph Forum 33:1–12. https://doi.org/10.1111/cgf.12275

Zhang S, Zhang H, Shu CW (2009) Topological structure of shock induced vortex breakdown. J Fluid Mech 1(639):343–372. https://doi.org/10.1017/S002211200999108X