

Deep-PIV: a new framework of PIV using deep learning techniques

Shengze Cai¹, Jiaming Liang¹, Shichao Zhou¹, Qi Gao¹,
Chao Xu^{1*}, Runjie Wei², Steve Wereley³, Jae-Sung Kwon⁴

¹ Zhejiang University, Hangzhou, China

² MicroVec., Inc, Beijing, China

³ Purdue University, West Lafayette, IN, USA

⁴ Incheon National University, Incheon, South Korea

* Corresponding author: Chao Xu (cxu@zju.edu.cn)

Abstract

In this work, we investigate a framework for performing deep learning-based particle image velocimetry (Deep-PIV). A convolutional neural network (CNN) is presented to extract motion fields from planar particle images. In order to implement supervised learning strategy, an artificial dataset including consecutive particle images and the corresponding ground-truth motions is produced for network training. Various image settings and various motion conditions are considered for the dataset generation. The trained CNN model receives an image pair as input and predicts a dense velocity field (one velocity vector for one pixel). A number of experiments including the synthetic turbulent flow, the laboratory jet flow in a tunnel and the electrothermal flow in a microfluidic chip are assessed in this paper to demonstrate the performance of the estimator. The proposed Deep-PIV shows potential to become a promising PIV technique with high accuracy and efficiency.

1 Introduction

Fluid visualization plays an increasingly important role in experimental fluid mechanics, as it helps the researchers to get a deeper insight into the complex flow phenomena. Among the flow visualization technologies, particle image velocimetry (PIV) (Raffel et al., 2007; Adrian and Westerweel, 2011) is the most commonly-used method, which can provide non-intrusive quantitative measurement of the velocity fields. One possible way to construct a PIV experiment is as follows. Small particles are firstly seeded into the flow medium. A laser sheet illuminates the flow domain to activate the particles, then successive images with fluorescent particles are recorded by a camera. Eventually, velocity vectors of the investigated flow are obtained by analysing the consecutive images. Besides PIV hardware system development, how to estimate the motion fields from an image pair or sequence, remains the key problem and main challenge in this area.

There are two main developed techniques for extraction of velocity fields from particle images: cross-correlation (CC) and optical flow (OF) methods. Correlation-based methods seek the displacement vector by searching the maximum of the cross-correlation between two interrogation windows of an image pair (Westerweel, 1997). This kind of technique provides sparse motion field and usually requires proper post-processing steps, such as outlier detection and spline interpolation (Westerweel and Scarano, 2005). It has been developed for thirty years and implemented in some commercial softwares (e.g., LaVision) due to the simplicity and efficiency. Another popular technique is optical flow computation, which is studied intensively in computer vision community. Optical flow computing relies on the optimization of an objective function and provides dense motion field for the whole image (Horn and Schunck, 1981). It attracts the researchers in experimental fluid mechanics since the formulation is easier to cope with priori knowledge of physics and the result contains more detailed structures of the flow (Heitz et al., 2010). However, optical flow approaches are generally time-consuming as it requires optimization process. Comparison between the OF and CC methods in performing particle image velocimetry has been presented by Liu et al. (2015).

With the development of deep learning techniques (Lecun et al., 2015) and the success of convolutional neural networks (CNN) (Krizhevsky et al., 2012), designing a deep neural network for motion estimation problem becomes a promising direction. In the PIV community, a proof-of-concept on this topic is proposed in Rabault et al. (2017). The authors designed modern artificial neural networks to perform end-to-end PIV for the first time and also provided detailed discussion, which inspires other researchers to involve in this study (Lee et al., 2017; Cai et al., 2019). The philosophy of the network models in Rabault et al. (2017) is similar to cross-correlation methods: the input of the models is a combination of two patches from the consecutive images in a size of 32×32 pixels and the output includes a displacement vector of the patch center. On the contrary, Cai et al. (2019) present the use of a deep neural network, which is called FlowNetS (Dosovitskiy et al., 2015), for extracting the dense velocity field (one vector for one pixel). The proposed CNN is trained on an artificial PIV dataset. The experimental results in Cai et al. (2019) indicate that the proposed model can provide good estimation with very high efficiency, which is promising for the application of PIV. However, the model does not outperform the classical variational optical flow method with respect to the accuracy.

In this paper, following the idea of Cai et al. (2019), we develop a deep learning estimator based on a newly-proposed CNN model called LiteFlowNet (Hui et al., 2018). LiteFlowNet is a successor of FlowNetS and is reportedly accurate, efficient and in small model size. The PIV dataset is used to train the parameters of the neural network. In this paper, we also setup several assessments to evaluate the performance of the CNN model. As illustrated by the experimental results, the trained model is likely to become a state-of-the-art estimator for PIV experiments. We should outline that this article is an extension of Cai et al. (2019). The identical framework of designing a deep learning PIV is proposed in Cai et al. (2019). However, we test a novel CNN model here and provide new assessments on some typical applications.

2 PIV via Deep Learning Techniques

2.1 PIV training dataset

Training the neural networks by supervised strategies requires data with ground-truth to optimize the model parameters. However, it is difficult to obtain the accurate velocity fields from experimental PIV data. We have to generate a synthetic dataset for CNN training. Following the general way to build a PIV dataset representing the fluid flow, we first generate a particle image as well as a flow motion pattern, then move the positions of the particles symmetrically by the flow motion to obtain an image pair.

To generate the particle images, one can follow the general idea (Raffel et al., 2007) that one particle can be described by a two-dimensional (2D) Gaussian function:

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

where I_0 is the peak intensity in the Gaussian center, d_p denotes the particle diameter and (x_0, y_0) the position of the particle. Note that I_0 , d_p and (x_0, y_0) can be diverse from each particle. Furthermore, let us define the particle seeding density of the image as ρ (unit: particle per pixel, ppp), which is a global parameter for an image and affects the particle number in the observed domain. With specified values of I_0 , d_p , (x_0, y_0) and ρ , a typical particle image can be produced. Parameters for defining an image can be randomly selected in a proper range, as shown in Table 1. Figure 1 displays three examples of particle image produced by different parameters. All the images are in the resolution of 256×256 pixels.

As for the flow motion, we extract the velocity fields from computational fluid dynamics (CFD). On the one hand, we simulate several flow benchmarks such as uniform flow (denoted by Uniform), backward stepping flow (Back-step) and the flow over a circular cylinder (Cylinder) with various configurations (e.g.,

Table 1: The ranges of parameters for defining a synthetic particle image.

Parameter	Range	Unit
Seeding density ρ	0.05 – 0.1	ppp
Particle diameter d_p	1 – 4	pixel
Peak intensity I_0	200 – 255	grey value

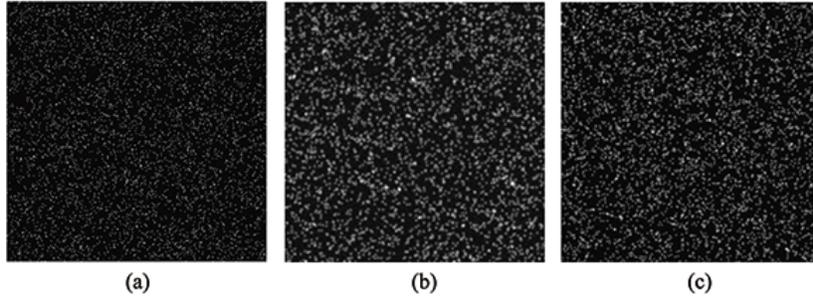


Figure 1: Particle images generated by PIG with different parameters: (a) seeding density $\rho = 0.078$ ppp, particle diameter $d_p = 1.31$ pixel, (b) $\rho = 0.051$ ppp, $d_p = 3.68$ pixel, (c) $\rho = 0.069$ ppp, $d_p = 2.81$ pixel.

Table 2: Descriptions of the motion fields for PIV neural network training.

Case name	Description	Condition	Quantity
Uniform	Uniform flow	$ \mathbf{dx} \in [0, 5]$	1000
		Re = 800	600
Back-step	Backward stepping flow	Re = 1000	600
		Re = 1200	1000
		Re = 1500	1000
Cylinder	Flow over a circular cylinder	Re = 40	50
		Re = 150	500
		Re = 200	500
		Re = 300	500
		Re = 400	500
DNS-turbulence	A homogeneous and isotropic turbulence flow	/	2000
SQG	Sea surface flow driven by SQG model	/	1500
JHTDB-channel	Channel flow provided by JHTDB	/	1600
JHTDB-mhd1024	Forced MHD turbulence provided by JHTDB	/	800
JHTDB-isotropic1024	Forced isotropic turbulence provided by JHTDB	/	2000

Reynold number). On the other hand, there are some online available CFD data that can also produce a number of flow structures. For instance, a 2D turbulent flow motion (termed DNS-turbulence) and a surface quasi-geostrophic (SQG) model of sea flow are provided in Carlier (2005) and Resseguier et al. (2017), respectively. And the Johns Hopkins Turbulence Databases (JHTBD) which includes different flow motions can be found in Li et al. (2008). The particle displacements of the whole dataset range from 0 to 10 pixels. We have generated a PIV dataset with more than 13K items (where each item consists of an image pair and a ground-truth motion field). Only around 500 items of the data are used for validation/testing, others are used for network training.

2.2 Deep neural network for motion estimation

In this section we introduce the CNN model which is based on LiteFlowNet (Hui et al., 2018). LiteFlowNet denotes a lightweight convolutional neural network for optical flow estimation. As illustrated in Figure 2, the architecture of LiteFlowNet is composed of two subnetworks: an encoder called NetC and a decoder NetE. It can be intuitively seen that NetC is specialized in feature extraction while NetE performs coarse-to-fine optical flow estimation with respect to the features.

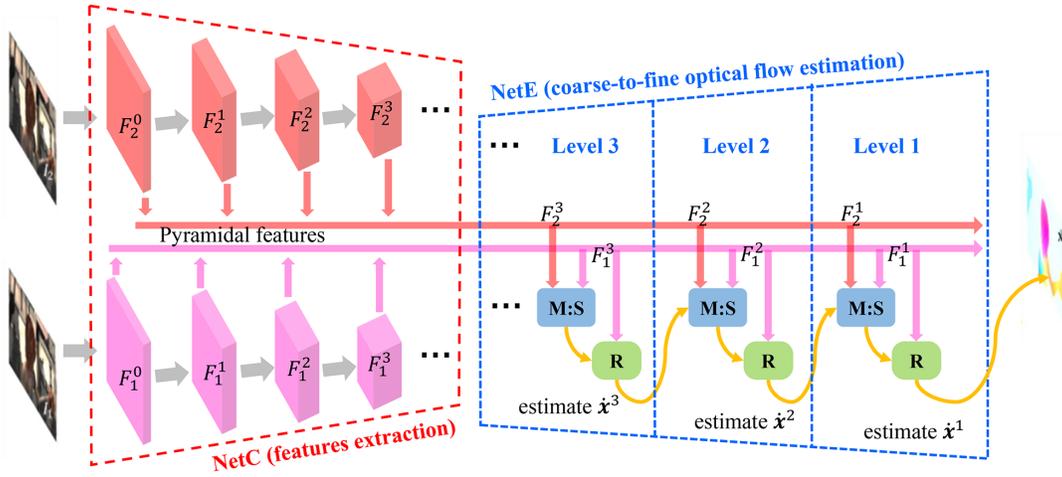


Figure 2: The network structure of LiteFlowNet Hui et al. (2018). An encoder NetC extracts pyramidal features of the input images and a decoder NetE estimates the optical flow field in a coarse-to-fine framework.

Pyramidal features extraction. NetC consists of two streams to contract the input image pair into two pyramids of multi-scale high-dimensional features, respectively. The two streams, which are composed of a set of convolutional layers with the reduction of spatial resolution by a factor of 2, share the same network structure and filter weights. As marked in Figure 2, features of different spatial resolutions are denoted by $\{F_{1,2}^k\}$ where the subscript is the image index and k represents the pyramidal level number (e.g., F^0 is the highest level with a full resolution; F^1 is the second level with a half resolution). There exists six feature levels in LiteFlowNet.

Optical flow estimation. NetE estimates the velocity field in a coarse-to-fine framework by using de-convolutional layers (Long et al., 2015). At each pyramid level (e.g., Level 2 in Figure 2), the flow field is predicted by using the features of the same resolution from NetC (e.g., F^2) as well as the estimated flow from the previous level (e.g., \hat{x} from Level 3). For example, at Level 2 this idea can be expressed simply by:

$$\hat{x}^2 = \mathbf{E}(F_1^2, F_2^2, \hat{x}^3). \quad (2)$$

The function \mathbf{E} represents a sequence of operations at each pyramid level, which can be summarized as follows. (1) Feature warping. In order to address large-displacement flows, the authors of Hui et al. (2018) propose to reduce the distance between features F_1 and F_2 by feature warping (i.e., distortion). F_2 is transferred towards F_1 by the previous flow estimate \hat{x} . In contrast to the traditional methods doing warping with respect to images, feature warping makes the network more powerful and efficient. (2) Feature matching and sub-pixel refinement (module **M:S** in Figure 2). Pixel-by-pixel matching is established by computing the correlation between feature F_1 and the warped feature \tilde{F}_2 . This results in a coarse flow estimate, which is only up to pixel-level accuracy. Then the estimated flow is fed into a module which aims to refine flow field to sub-pixel accuracy by minimizing the feature-space distance. (3) Flow regularization (module **R** in Figure 2). In order to make the estimated flow smoother and avoid outliers, LiteFlowNet uses a feature-driven local convolution to regularize the flow field. This procedure plays a similar role as the regularization term in the variational optical flow formulation. However, the regularized function is locally and adaptively constructed for individual flow patches.

Note that as shown in Figure 2, the estimated flow of Level 1 is the output of the network, whose spatial resolution is only half of the input images. At the end of LiteFlowNet, interpolation method is used to refine the flow field into full resolution instead of adding more de-convolutional layers inside the network. However, according to our experiments, further refinement by network training (rather than by interpolation) does improve the performance on PIV estimation. One of the reasons may be that the fluid motion contains small-scale (even sub-pixel) structures that may be ignored by interpolation methods. Therefore, we add one more level (denoted as Level 0) to the LiteFlowNet in our case. The Level 0 operates totally the same as the other Level, which can be written as Equation (2).

There are a number of convolutional layers and de-convolutional layers at each pyramid level of NetC and NetE. All the filter weights are learned by minimizing a loss function during the network training process. For LiteFlowNet, the loss function is composed of the prediction errors at different pyramid levels of NetE:

$$Loss = \sum_i \lambda_i e_i, \quad (3)$$

where i is the level index, e_i denotes the error metric between the ground-truth and the prediction and λ_i the weights of different levels. The employed network is trained on the PIV dataset by using a stochastic optimization method called Adam. We use a mini-batch of 4 image pairs for each iteration process. A CNN model with trained parameters is obtained after around 900K iterations. As a novel framework of performing particle image velocimetry, we name it as **Deep-PIV** in this article which represents a deep learning PIV estimator.

3 Experimental Evaluations

The performance of the trained CNN model (Deep-PIV) can be evaluated by a number of assessments. As the framework proposed in this article is identical to that in Cai et al. (2019), the investigation on the basic properties of the PIV estimator can be referred to Cai et al. (2019). Here we apply the novel Deep-PIV model to several typical experiments. In Sect. 3.1, a synthetic particle image sequence describing the turbulent flow is investigated. Real images of the jet flow in a water tunnel and the electrothermal (ET) flow in a microfluidic chip are assessed in Sect. 3.2 and Sect. 3.3, respectively.

3.1 Test on synthetic turbulent flow

First of all, we investigate the performance of the Deep-PIV estimator on a two-dimensional (2D) turbulent flow. The particle image sequence is provided by Carlier (2005) and generated by Direct Numerical Simulation (DNS). The dataset, demonstrating the movement of a 2D homogeneous turbulent flow with Reynolds number $Re = 3000$ and Schmidt number $Sc = 0.7$, consists of 100 images at the resolution of 256×256 pixels. An example at a typical time step is demonstrated in Figure 3(a), which illustrates the ground-truth velocity field and the vorticity map. For comparison, we introduce two traditional PIV approaches: the window deformation iterative multi-grid method (WIDIM) (Scarano, 2002) and the multi-resolution Horn & Schunck (HS) optical flow method (Ruhnau et al., 2005; Heitz et al., 2010). A multi-resolution HS algorithm with six pyramidal levels and two warping steps at each level is applied, while the WIDIM method uses two-passes implementation with a step size of 4 pixel. As we can see from the vorticity differences in Figure 3(b)-(d), the result provided by the Deep-PIV model is quite consistent with the true motion field. The HS method also provides small error even for the small-scale structures, while the WIDIM performs the worst in this case. The root mean square error (RMSE) is also computed along the whole image sequence. The results of different methods are plotted in Figure 4. As shown in the figure, the novel estimator is superior to the WIDIM method and the HS optical flow approach with smaller errors. This experiment indicates that the use of deep learning model are effective for PIV estimation, even for the turbulent flows which contain small-scale structures.

In this assessment, we also demonstrate the computational time for each estimator, as given in Table 3. Although the CNN estimator is processed with a graphics processing unit (GPU), we can see great advantage of the efficiency. The execution time of deep learning model is less than that of WIDIM method even it is converted to CPU mode. We also outline that the Deep-PIV estimator provides dense motion field while the cross-correlation method only estimates sparse velocities.

3.2 Test on jet flow

In order to demonstrate the practicality of the PIV method based on deep neural network, it is necessary to test the Deep-PIV model using real experimental data. In this subsection, particle images of jet flow are assessed. The experimental setup for performing jet flow is as follows. In the circulating water tunnel, a semicircular orifice plate is installed on the left side to form a jet flow. In addition, a step is placed in front of the orifice plate (namely on the right side of the image screen) to construct a forward stepping flow. This results in a separated vortex flow in the observed region. The configuration is a common experimental setup in fluid mechanics, which can be used to perform typical flow structures such as jet flow, vortex flow,

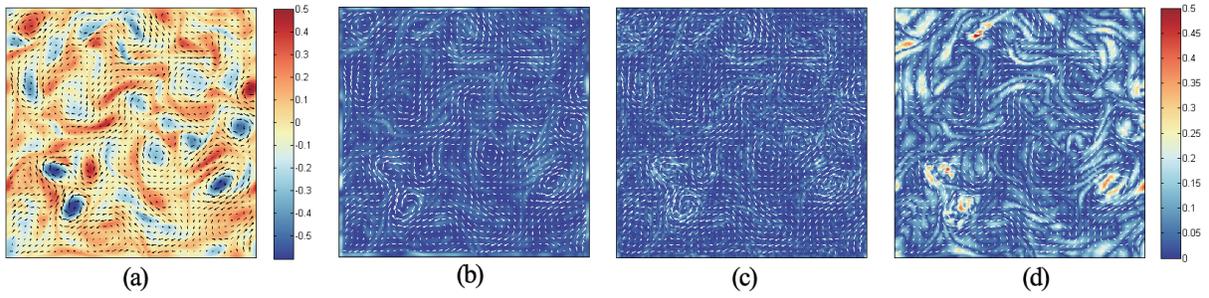


Figure 3: An example of the 2D turbulent flow. Figures show the velocity fields of (a) the ground-truth, (b) the Deep-PIV model, (c) the HS optical flow method and (d) the WIDIM method. The background color of (b)-(d) represents the absolute difference between the true vorticity and the estimated vorticity.

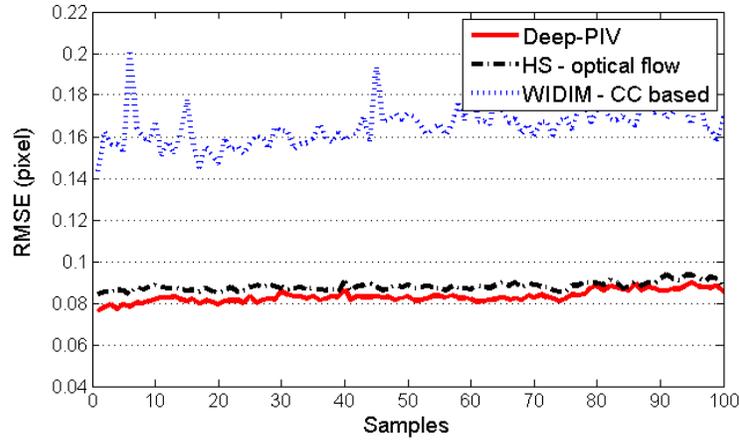


Figure 4: The RMSE of different estimators for particle image sequence of 2D turbulent flow.

stepping flow, etc. In our case, the particle images are captured by a planar PIV system. A schematic diagram of the experimental setup and an example of the particle image are illustrated in Figure 5.

For this experiment, the estimated velocity field of Deep-PIV model is compared with the result of a PIV commercial software (provided by MicroVec., Inc, Beijing). The algorithm used in the PIV commercial software is based on the advanced cross-correlation method giving a sparse velocity field, while the deep learning PIV estimator can provide dense velocity field. Figure 6 demonstrates the velocity magnitudes corresponding to the velocity fields. As shown, the estimated motion of Deep-PIV model is quite similar to the result of the PIV commercial software. As dense velocity field contains more detailed structures, it can be seen from the magnitude maps that the result of the deep neural network model is smoother than that of the correlation method. It should be noted that the real PIV images of jet flow are not included in the dataset used for network training. Therefore, the experimental result indicates that the Deep-PIV, which is trained on artificial PIV data, can be successfully applied in practical PIV application.

3.3 Test on electrothermal flow

In the following, the Deep-PIV model is applied to estimate the motion of the light-actuated electrothermal (ET) flow. The laboratory particle images assessed in this paper are proposed by Kwon and Wereley (2015). The ET flow is generated by the simultaneous application of a laser and an AC electric field in a microfluidic chip filled with deionized (DI) water. Such an experiment can help to get a more intuitive investigation of the physical phenomenon with the micro-PIV measurement.

As described in Kwon and Wereley (2015), the experimental setup for ET flow, which is illustrated in Figure 7, consists of a microfluidic chip, two microscope systems and a CCD camera. The microfluidic chip is composed of two parallel-plate electrodes and several chambers including an inlet/outlet reservoir and

Table 3: Execution time of different methods for the 2D DNS turbulent flow. N_v represents the valid vectors of the estimated fields and T denotes the average computational time for processing one image pair. The CNN estimator runs in the system with Intel Core i7-7700 CPU @3.60GHz and a graphics processing unit (NVIDIA GTX 1080 Ti).

processor	Deep-PIV GPU	HS - optical flow CPU	WIDIM - CC based CPU
N_v	256×256	256×256	61×61
T (ms)	47	2294	509

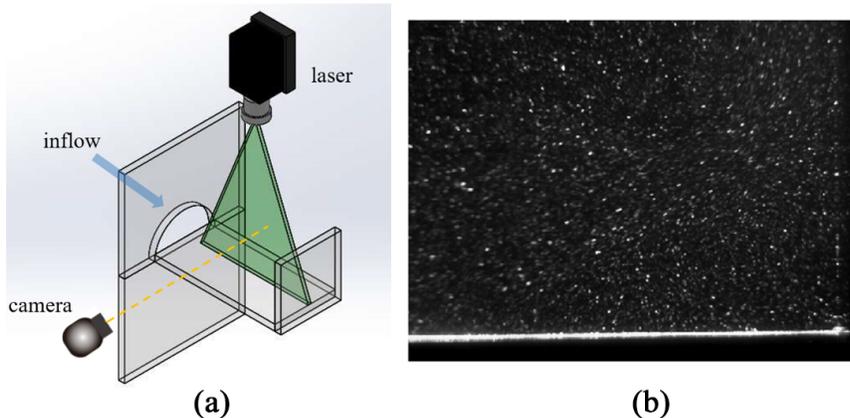


Figure 5: A sketch of the experimental setup of jet flow in a water tunnel (a) and an example of the captured particle image (b).

a microchannel. The AC electric field is produced by the parallel-plate electrodes. One of the microscope system is used to illuminate the bottom electrode and drive the light-actuated ET flow by providing a focused near-infrared Nd:YVO4 laser. The other microscope system, which provides an epi-fluorescent filter cube with a mercury-arc lamp, is applied for excitation of the red fluorescent particles. The successive particle images are captured by positioning the CCD camera to the side of the chip and placing its focal plane on the optical axis (O.A.).

The 2D imaging region was selected at the central flow cross section in the microfluidic chip. The whole image sequence contains over 200 particle images and the observed fluid motion takes the form of a rotationally symmetric toroidal vortex, as we can see from Figure 8(a). Two consecutive images are fed into the deep learning estimator and the result is taken as the average of all samples. The PIV measurement of light-actuated ET flow motion is demonstrated in Figure 8. Here we compare the estimated field with the PIV evaluation used in Kwon and Wereley (2015), which is performed building 64×64 pixel interrogation windows and applying a central difference image correction (CDIC) method (Wereley et al., 2002). As shown in Figure 8(b), the flow field estimated by Deep-PIV is similar to the one provided by advanced CC method. The vortex, which is symmetrically distributed with respect to the optical axis (O.A.), can be clearly observed. Besides, we can see from Figure 8(d) that the speed distribution on the line connecting the two vortex centers (denoted by C.L.) follows a bell curve with a maximum value at the O.A. These results, which are quite consistent with the analysis in Kwon and Wereley (2015), support the visualized axisymmetric toroidal flow structure quantitatively. From this typical experiment, we can conclude that the proposed Deep-PIV model is also effective for micro-PIV application (even though the ET flow motion is not included in the training data).

4 Conclusion

In this paper, we propose a deep learning estimator for particle image velocimetry. This estimator is developed from the LiteFlowNet Hui et al. (2018) and is able to extract dense velocity field from a particle image pair. The convolutional neural network is trained by using a synthetic PIV dataset which can be originally

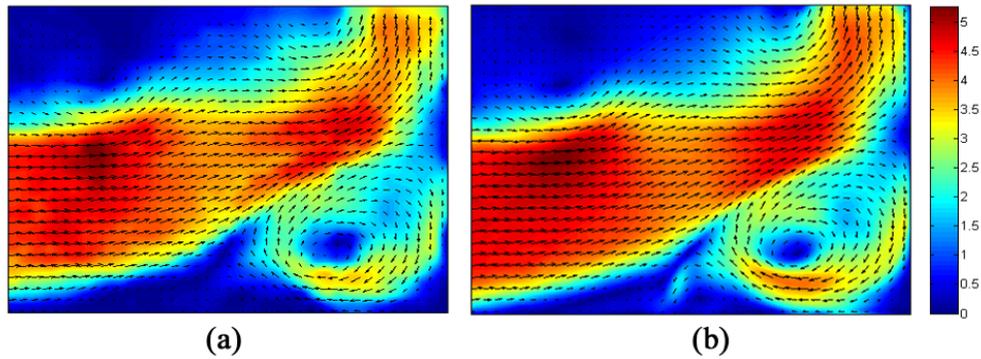


Figure 6: The velocity fields of jet flow estimated by (a) the advanced CC method and (b) the Deep-PIV model. The background color represents the magnitude of the velocity.

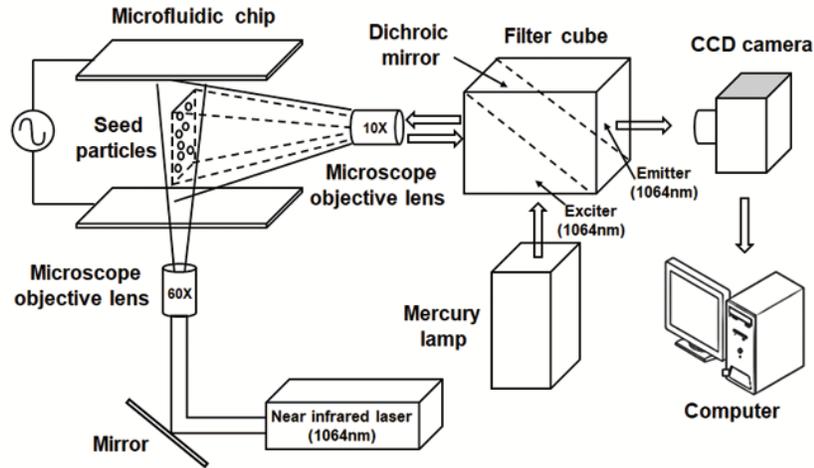


Figure 7: Schematic diagram of the experimental setup for flow visualization, PIV measurement and single dye-based LIF (laser induced fluorescence) thermometry of light-actuated ET fluid motion.

referred to Cai et al. (2019). The proposed Deep-PIV model is assessed by a number of experiments. First of all, a synthetic two-dimensional turbulent flow is used as the benchmark to evaluate the performance of the deep learning estimator. Results show that the proposed model can provide quite accurate estimations for PIV even there exists small-scale vortices which are difficult to estimate. Compared to the traditional cross-correlation and optical flow methods, it also shows great advantage of relatively high accuracy and efficiency. In addition, we apply the deep learning model to estimate the flow fields of jet flow and light-actuated electrothermal (ET) flow. The measurements in these cases are consistent with the results of PIV commercial software or the theoretical analysis in the literature. Thus we can see that although the network parameters are trained from synthetic PIV dataset, the model succeeds to extract the flow fields from real experimental particle images. Based on these assessments, we believe that deep learning can be a promising technique in conducting PIV estimation. On the one hand, the accuracy of the estimator can be further improved by designing a more powerful neural network. On the other hand, the deep learning techniques can play an important role in other applications (e.g., image reconstruction for tomographic PIV) besides the planar PIV estimation. More potentials can be investigated in the future.

Acknowledgements

This work was supported in parts by the National Natural Science Foundation of China under Grant 61473253, the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under

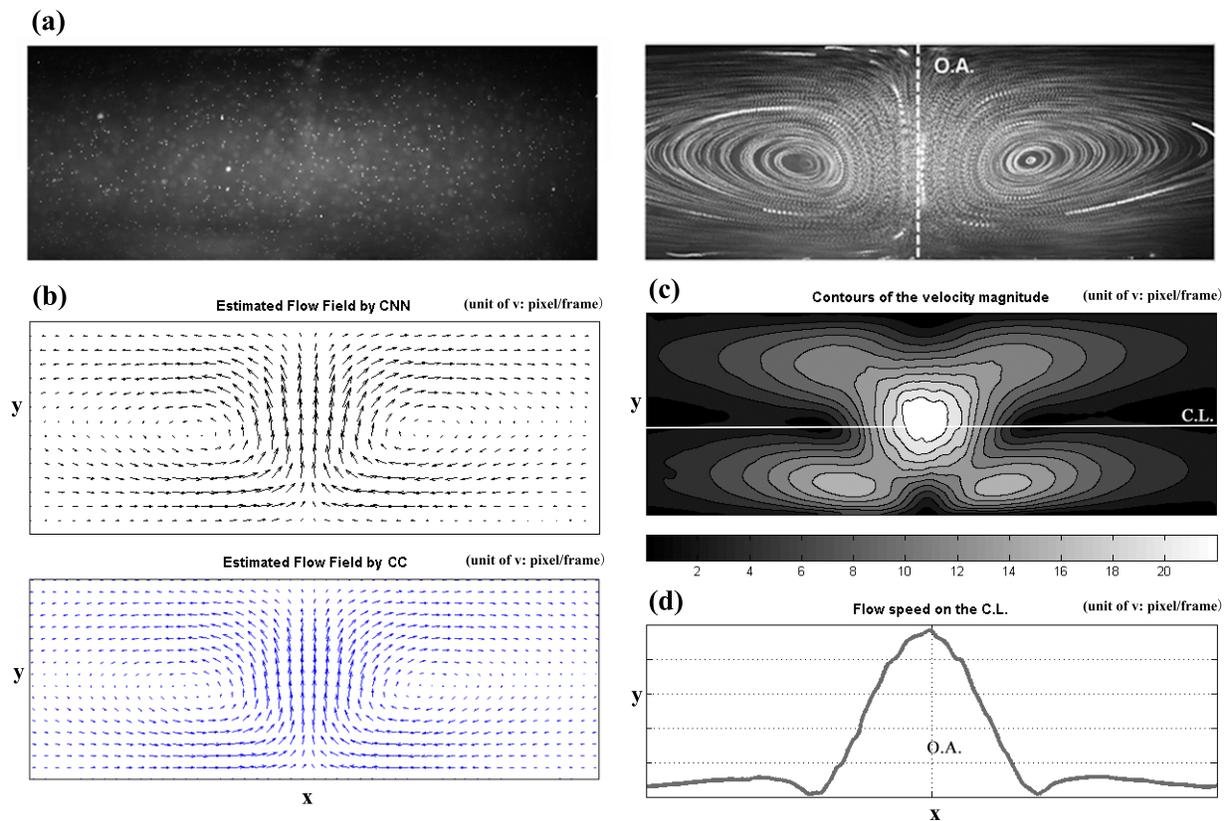


Figure 8: (a) Visualization of the ET flow: a particle image (left) and the symmetric vortex (right). (b) Flow field in the ET vortex estimated by deep learning PIV model (top) and correlation-based method (bottom). (c-d) Flow field and speed distribution in the ET vortex estimated by deep learning PIV model. In the figures, O.A. denotes the optical axis of the laser beam, and C.L. represents the line connecting the two vortex centers.

Grant 61621002 and the Fundamental Research Funds for the Central Universities under Grants 2018XZZX001-09 and 2019QNA4056.

References

- Adrian R and Westerweerd J (2011) *Particle Image Velocimetry*. Cambridge University Press
- Cai S, Zhou S, Xu C, and Gao Q (2019) Dense motion estimation of particle images via a convolutional neural network. *Experiments in Fluids* 60:73
- Carrier J (2005) Second set of fluid mechanics image sequences. *European Project Fluid image analysis and description (FLUID)* - <http://www.fluidiris.fr>
- Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van der Smagt P, Cremers D, and Brox T (2015) FlowNet: Learning optical flow with convolutional networks. in *Proceedings of the IEEE International Conference on Computer Vision*. pages 2758–2766
- Heitz D, Mémin E, and Schnörr C (2010) Variational fluid flow measurements from image sequences: synopsis and perspectives. *Experiments in Fluids* 48:369–393

- Horn B and Schunck B (1981) Determining optical flow. *Artificial Intelligence* 17:185–203
- Hui T, Tang X, and Loy C (2018) LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. pages 8981–8989
- Krizhevsky A, Sutskever I, and Hinton G (2012) Imagenet classification with deep convolutional neural networks. in *Advances in Neural Information Processing Systems*. pages 1097–1105
- Kwon JS and Wereley ST (2015) Light-actuated electrothermal microfluidic motion: experimental investigation and physical interpretation. *Microfluidics & Nanofluidics* 19:609–619
- Lecun Y, Bengio Y, and Hinton G (2015) Deep learning. *Nature* 521:436
- Lee Y, Yang H, and Yin Z (2017) PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. *Experiments in Fluids* 58:171
- Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A, and Eyink G (2008) A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence* 9
- Liu T, Merat A, Makhmalbaf H, Fajardo C, and Merati P (2015) Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images. *Experiments in Fluids* 56:166
- Long J, Shelhamer E, and Darrell T (2015) Fully convolutional networks for semantic segmentation. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3431–3440
- Rabault J, Kolaas J, and Jensen A (2017) Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Measurement Science & Technology* 28:125301
- Raffel M, Willert C, Wereley S, and Kompenhans J (2007) *Particle Image Velocimetry: A Practical Guide*. Springer
- Resseguier V, Mémin E, and Chapron B (2017) Geophysical flows under location uncertainty, Part II: Quasigeostrophic models and efficient ensemble spreading. *Geophysical & Astrophysical Fluid Dynamics* 111:177–208
- Ruhnau P, Kohlberger T, Schnörr C, and Nobach H (2005) Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids* 38:21–32
- Scarano F (2002) Iterative image deformation methods in PIV. *Measurement Science & Technology* 13:R1
- Wereley ST, Gui L, and Meinhart CD (2002) Advanced algorithms for microscale particle image velocimetry. *AIAA Journal* 40:1047–1055
- Westerweel J (1997) Fundamentals of digital particle image velocimetry. *Experiments in Fluids* 23:1379
- Westerweel J and Scarano F (2005) Universal outlier detection for PIV data. *Experiments in Fluids* 39:1096–1100