Scientific Visualization, 2022, volume 14, number 4, pages 1 - 11, DOI: 10.26583/sv.14.4.01

High-speed Flow Structures Detection and Tracking in Multiple Shadow Images with Matching to CFD using Convolutional Neural Networks

I.A. Doroshchenko[1,A], I.A. Znamenskaya[2,A], N.N. Sysoev[3,A], A.E. Lutsky[4,B]

[A] Lomonosov Moscow State University
[B] Keldysh Institute of Applied Mathematics RAS

[1] ORCID: 0000-0002-0488-0020, doroshenko.igor@physics.msu.ru
[2] ORCID: 0000-0001-6362-9496, znamen@phys.msu.ru
[3] ORCID: 0000-0002-1162-7680, nn.sysoev@physics.msu.ru
[4] ORCID: 0000-0002-4442-0571, allutsky@yandex.ru

Abstract

Shadowgraph imaging has been widely used to study flow fields in experimental fluid dynamics. Nowadays high-speed cameras allow to obtain millions of frames per second. Thus, it is not possible to analyze and process such large data sets manually and automatic image processing software is required. In the present study a software for automatic flow structures detection and tracking was developed based on the convolutional neural network (the network architecture is based on the YOLOv2 algorithm). Auto ML techniques were used to automatically tune model and hyperparameters and speed-up model development and training process. The neural network was trained to detect shock waves, thermal plumes, and solid particles in the flow with high precision. We successfully tested out software on high-speed shadowgraph recordings of gas flow in shock tube with shock wave Mach number M = 2-4.5. Also, we performed CFD to simulate the same flow. In recent decades, the amount of data in numerical simulations has grown significantly due to the growth in performance of computers. Thus, machine learning is also required to process large arrays of CFD results. We developed another ML tool for experimental and simulated by CFD shadowgraph images matching. Our algorithm is based on the VGG16 deep neural network for feature vector extraction and k-nearest neighbors algorithm for finding the most similar images based on the cosine similarity. We successfully applied our algorithm to automatically find the corresponding experimental shadowgraph image for each CFD image of the flow in shock tube with a rectangular obstacle in the flow channel.

Keywords: High-speed shadowgraphy, shock wave, convolutional neural network, YOLOv2, VGG16, Auto ML, object detection, CFD, k-nearest neighbors, cosine similarity.

1. Introduction

Flow visualization is the visualization of data obtained in fluid dynamics experiments or in computational fluid dynamics (CFD). Schlieren and shadowgraph techniques have been widely used for studying gas flows in experimental fluid dynamics since the 19th century [1, 2]. Both methods are based on the light refraction phenomena. If the light ray passes through region of inhomogeneity, it will refract proportional to the gradient of the refractive index of the medium. The refractive index, in turn, is proportional to the density of the medium in accordance with the well-known Gladstone-Dale equation [3]: $n - 1 = G(\lambda)\rho$, where n is the refractive index of the fluid, ρ is its density, $G(\lambda)$ is the Gladstone-Dale constant and λ is the wavelength of light.

Nowadays with the development of high-speed digital equipment and growing computational power of computers the size of flow data becomes larger and manual analysis becomes very complicated and time consuming. However, machine learning and computer vision approaches can handle this problem. Typical machine learning process consists of 5 stages [4]: (1) formulating a problem, (2) collecting and preparing data for the model, (3) choosing an architecture to represent the model, (4) designing a loss function to minimize error, (5) selecting an optimization algorithm for model training. In the past few years deep learning methods have been started to apply in fluid dynamics [4, 5] and flow visualization [6, 5]. Liu et al. [6] first published a detailed review on the use of deep learning in flow visualization. They identify 3 main areas of research: Data Management, Feature Extraction, and Interactive Analysis. Data management includes such tasks as high-resolution data production from low-resolution information [7, 8], flow data downscaling, flow field reconstruction [9]. Automatic flow feature extraction is also being developed. Deep learning methods are being used for vortex [10] and shock [11, 12] features extraction. Berenjkoub et al. [13] successfully tested CNN, Resnet, Unet and IVD-Net architectures for vortex detection. Beck et al. [14] considered shock detection problem as image processing problem. Morimoto et al. [15] proposed a method to fill missing regions on the PIV images by using supervised machine learning. Training data included artificial particle images as input data and velocity vectors as output. Output vectors were obtained by a direct numerical simulation. The obtained model was successfully applied to fill missing parts of the experimental PIV images. It was found that the neural network can obtain finer and hidden flow structures, which cannot be resolved with the cross-correlation method. Ubald et al. [16] proposed a method based on the Gaussian process model for extracting quantitative information from schlieren images. A pair of schlieren images with the knife-edge at horizontal and vertical orientations was used to extract density values from the images.

It is often possible to extract quantitative physical information from schlieren and shadowgraph images using simpler computer vision methods without resorting to machine learning. Edge detection methods are widely used for shock detection. Canny algorithm [17] showed the most accurate results in edge detection on shadowgraph and schlieren images [18, 19]. Background image subtraction in frequency domain is often used to improve image quality and reduce noise level [19]. In our previous studies, we first used the Hough transform together with Canny edge detection for more accurate recognition of shock waves and automatic oblique shock angle calculation in the supersonic gas flow [20, 21].

In the present study we developed object detection neural network based on the YOLOv2 architecture [22]. It was trained to detect three classes of objects in the shadowgraph images: solid particles in the flow, vertical shocks, and thermal plumes (Figure 1). It is improved version of the neural network used in [20]. We used Auto ML tools [23] developed by Apple [24] to automatically tune the model and hyperparameters to achieve better results in object detection. We also developed ML tool that allows you to compare CFD and experimental shadowgraph images and obtain their similarity level. We applied this solution to find the corresponding experimental image to each synthetic CFD frame. Our solution is based on the VGG16 convolutional neural network [25] for feature vector extraction and k-nearest neighbors algorithm for finding the most similar images based on the cosine similarity.

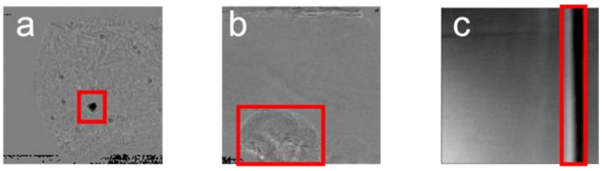

Figure 1. Examples of object classes in the training set (shadowgraph images):
a - "particle", b - "plume", c - "shock"

2. Experiments

We performed 3 types of experiments to test our ML software. First set of experiments included shadowgraph imaging of the flow in open-ended shock tube in time range from 0 to 40 ms (Figure 2). Shock tube has a rectangular cross-section of 24x48 mm^2. The driver gas is helium, the driven is air. The driven gas pressure was 1 atm (open-ended shock tube). Created shock wave (Figure 2, 1) Mach number is M = 2-4.5. A shadowgraph system was set up to visualize flow in the test section. Test section contains a set of electrodes (Figure 2, 2) to create pulsed surface and volume electrical discharges to initiate instant energy input into the flow. The side walls are quartz glass windows for optical access (Figure 2, 3).

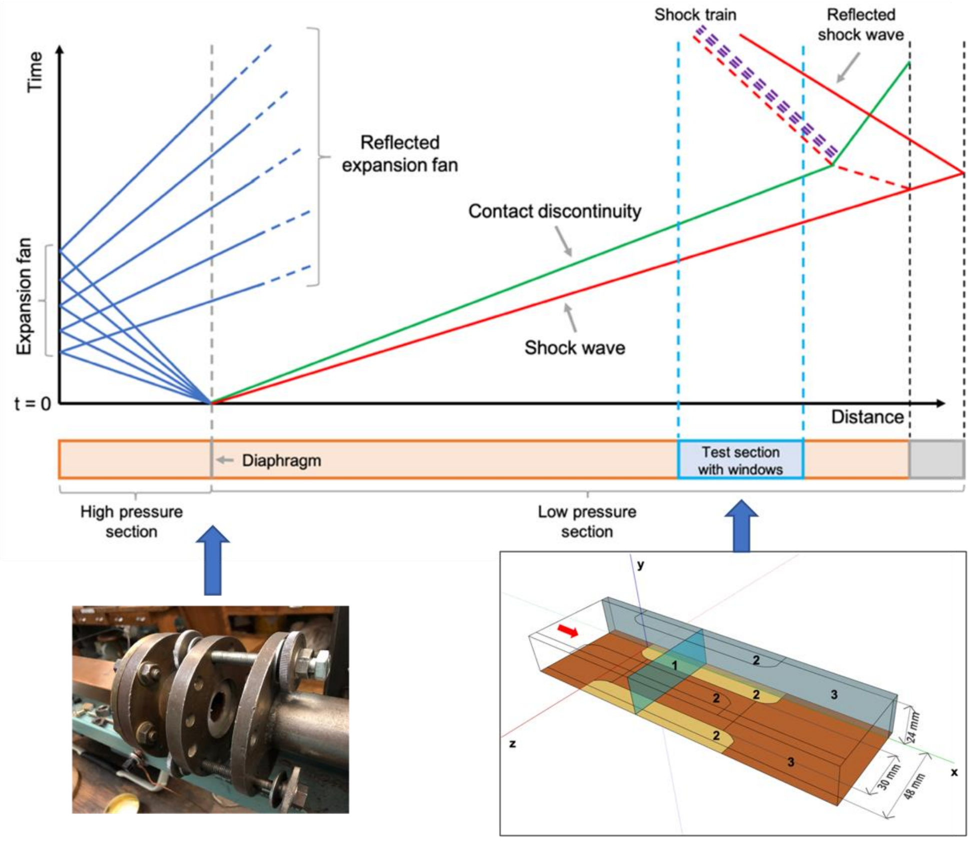

Figure 2. Shock tube scheme and x-t plot of gas flow development after diaphragm rupture

The second set of experiments was designed to test vertical shock wave detection in complex flow with horizontal and vertical shocks and high noise level. The flow was created in our test section by pulsed surface and volume electric discharges. A target for object detection was a vertical shock wave created by a pulsed cylindrical volume discharge. A detailed description of the experiments is given in [26].

The third set of experiments was designed to compare CFD and experimental frames using VGG16 convolutional neural network and automatically find the most similar experimental images to the CFD results. The experiments were carried out on the same experimental setup. A rectangular obstacle was placed on the bottom wall of the test section in a supersonic flow. At a specific point in time, an electric discharge was switched on the lower wall of the test section initiating a shock wave spreading from the discharge [27].

The Euler equations model describing the motion of an ideal compressible fluid was used numerically solved by the finite volume method. In this work we are interested in the discontinuities' dynamics and energy input time and configuration. So the viscosity effect can be neglected. Energy input during 200 ns time interval was simulated using q component in the energy conservation equation. Shadow images comparison between experiment (left) and numerical simulation (right) was made (Figure 3).

Figure 3 shows 3D sketch of the process and a comparison of the synthetic shadowgraph images obtained by CFD and the shadowgraph images obtained in experiments. The developed ML model was trained to automatically find such pairs of similar CFD and experimental images.

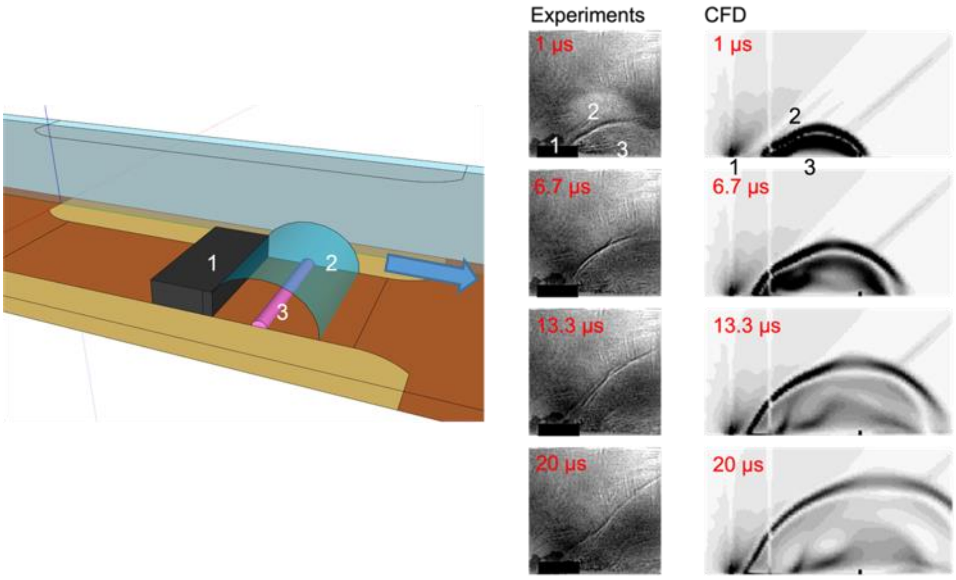

Figure 3. Shock wave spreading from the pulsed surface electrical discharge in the supersonic flow near the obstacle: CFD and experimental shadowgraph images comparison. 1 – solid obstacle; 2 – discharge-induced blast wave; 3 – discharge channel (energy input region).

3. Fluid structures detection using CNN

We trained a CNN based on the well-known YOLOv2 architecture to detect three types of objects on shadowgraph images: shocks, plumes and particles (Figure 1). We used Auto ML [23] tools to optimize CNN model and hyperparameters. Create ML (v. 3.0) utility developed by Apple was used to automate and simplify model setup and training [24]. Object detection mode was selected in the Create ML utility (CNN with modified YOLOv2 architecture). 9000 learning iterations, auto batch size and 13x13 pixels grid size were selected as the initial settings. We prepared data for training and testing based on the set of 844 objects in total: 472 shock wave images, 196 plume images and 176 particle images (Figure 4). It also includes a configuration file with the locations and sizes of objects bounding boxes and their classes.

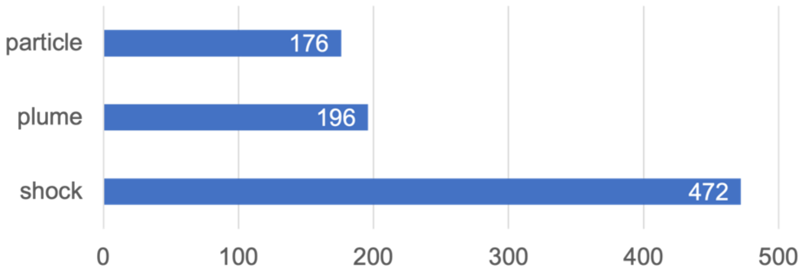

Figure 4. Distribution of objects for training

The supervised learning process included 9000 iterations. The graph in Figure 5 characterizes the effectiveness of training (loss vs epoch dependency). Final loss value on the last iteration was 0.665, which is quite good (generally, values between 0 and 1 are considered appropriate). Testing the model showed the following results: 92% correct answers for the training set and 88% for the test set.

The input for the model was an image of size 416x416 pixels (color image). Intersection over union threshold (IoU) was set to 0.45. Confidence threshold was set to 0.25. Output included two multidimensional arrays: confidence (boxes × class confidence) and coordinates (boxes × [x, y, width, height]).

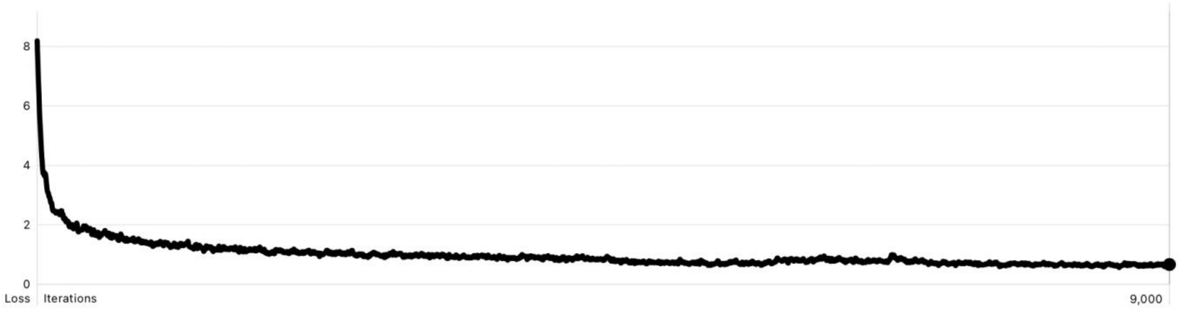

Figure 5. Effectiveness of model training: loss vs epoch

After exporting the model from the Create ML utility, we used it in our application for MacOS written in Swift programming language. In our application we used Core ML [28] and Vision [29] libraries to import the model and setup object detection workflow.

We tested our software on the high-speed shadowgraph images of the flow in the open-ended shock tube. The recording frame rate was 150 000 fps. The initial shock wave Mach number was M = 4.2. Air pressure in the open-ended low-pressure section was 1 atm. Figure 6 shows some of the examples of object detection on different stages of the flow. Red bounding boxes of the detected flow structures were rendered automatically by our software.

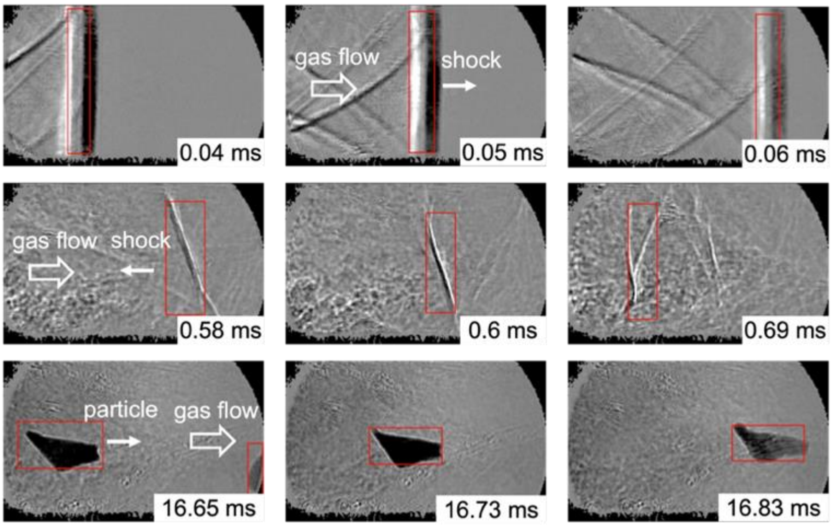

Figure 6. Flow objects detection on shadowgraph images of the flow in shock tube by the CNN. First row – initial shock wave; second row – reflected discontinuities; third row – detection of parts of the destroyed diaphragm

Video 1 contains the full image sequence.

Video 1. Flow objects detection on shadowgraph images of the flow in shock tube by the CNN

The initial shock wave created after diaphragm rupture, reflected discontinuities and particles (parts of the destroyed diaphragm) were detected automatically. Positions and sizes of the detected bounding boxes were saved to the text file for further analysis. Figure 7 shows x-t plots of the detected discontinuities in time ranges 0-6 ms and 0-1.2 ms. The initial shock wave and a few reflected discontinuities we captured. Such x-t plots allow us to estimate discontinuity velocity as $\frac{\Delta x}{\Delta t}$.

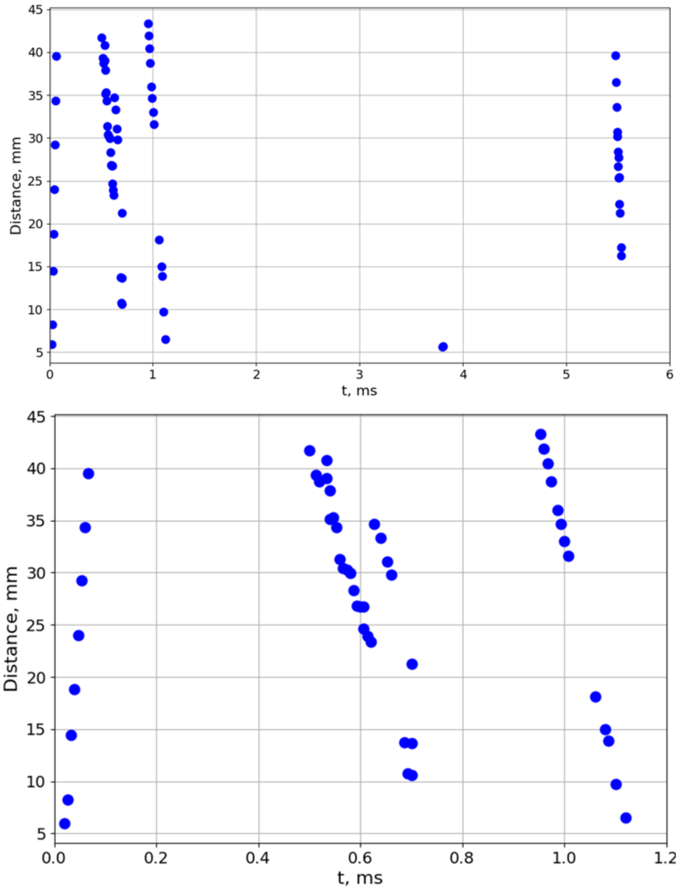

Figure 7. x-t plots of the detected by CNN discontinuities in time ranges 0-6 ms and 0-1.2 ms

“Particle” class objects were also captured (parts of the destroyed diaphragm). The corresponding x-t plot (Figure 8) allows us to roughly estimate flow velocity during the time range from 0 to 40 ms from particle velocities if we suppose that particles are following the flow. Each particle velocity can be calculated as $\frac{\Delta x}{\Delta t}$ for the corresponding x-t dependency. It is clearly visible that flow velocity gradually decreases. 25 ms after the passage of the initial shock wave velocity oscillations were detected.

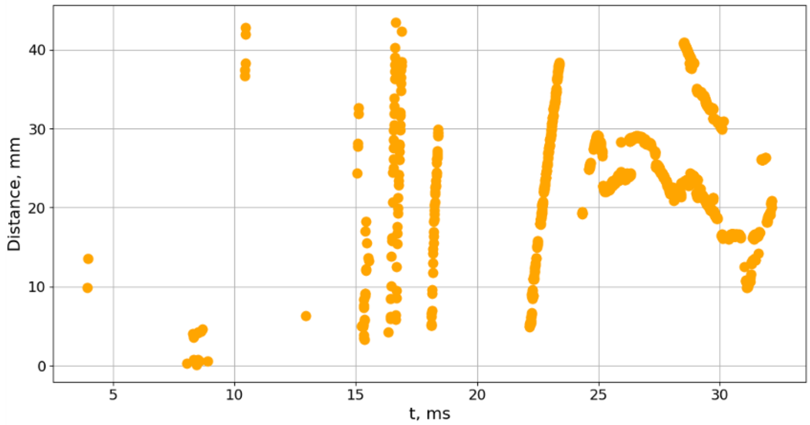

Figure 8. x-t plots of the detected by CNN particles in time range 0-40 ms

We performed another set of experiments with the high-speed shadowgraph imaging of the flow created by the complex pulsed electrical discharge arrangement. Created by the discharges flow contains three shocks: two horizontal – spreading from the surface discharges and one vertical – spreading from the cylindrical discharge in the volume of the test section. The vertical shock wave Mach number is M = 1.6. The flow contains a lot of inhomogeneities and noise, which complicates the recognition of objects by the neural network. Despite this fact, the created neural network showed high resistance to noise and made it possible to accurately recognize all vertical discontinuities in the flow (Figure 9). The first two rows (Figure 9) contain shock wave (1) capture spreading from the pulsed cylindrical electric discharge. It moves from right to left. The third row contains images of the reflected shock waves detection (2). They move from left to right. Horizontal shock waves and different flow inhomogeneities don't spoil object detection accuracy.

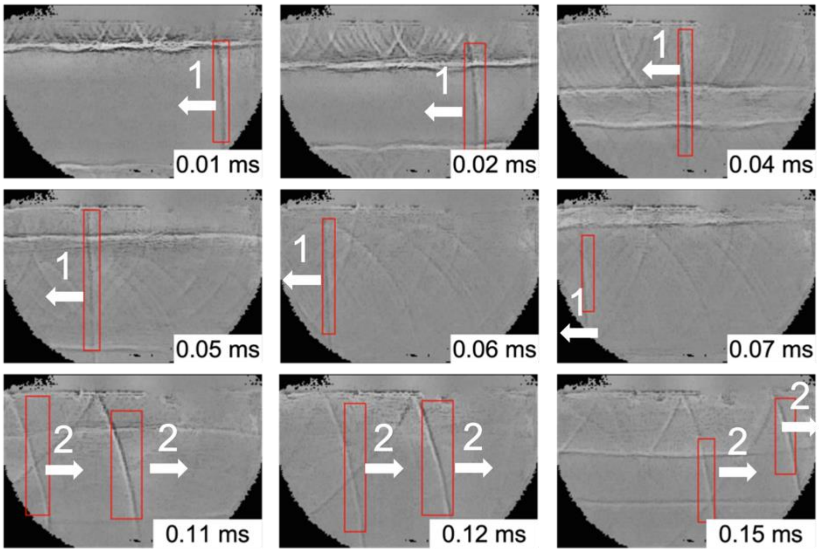

Figure 9. Flow structures detection on the shadowgraph images of the complex post-discharge flow by CNN.

Figure 10 shows shock wave detection results from full data set corresponding to Figure 9. High accuracy of object detection with high background noise level was demonstrated.

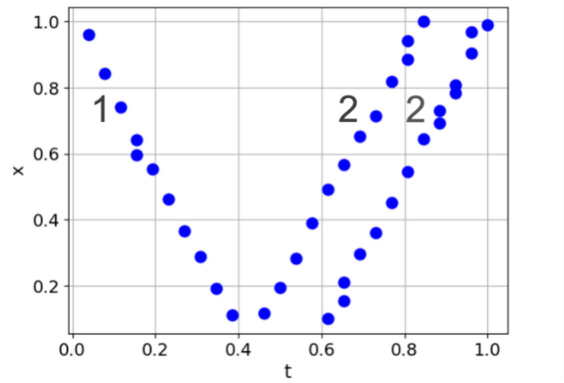

Figure 10. Shock wave detection with high background noise level based on the CNN. Axis values are normalized.

4. Experimental and synthetic CFD shadowgraph images matching using deep learning

An important task in fluid dynamics is the comparison of experiment and CFD results. Experimental shadowgraph images are often compared with the CFD results to test the calculations. We applied deep learning approach to automatically compare CFD and experimental shadowgraph images and find pairs of images for the same flow stage. Figure 11 shows some examples of input and output images.

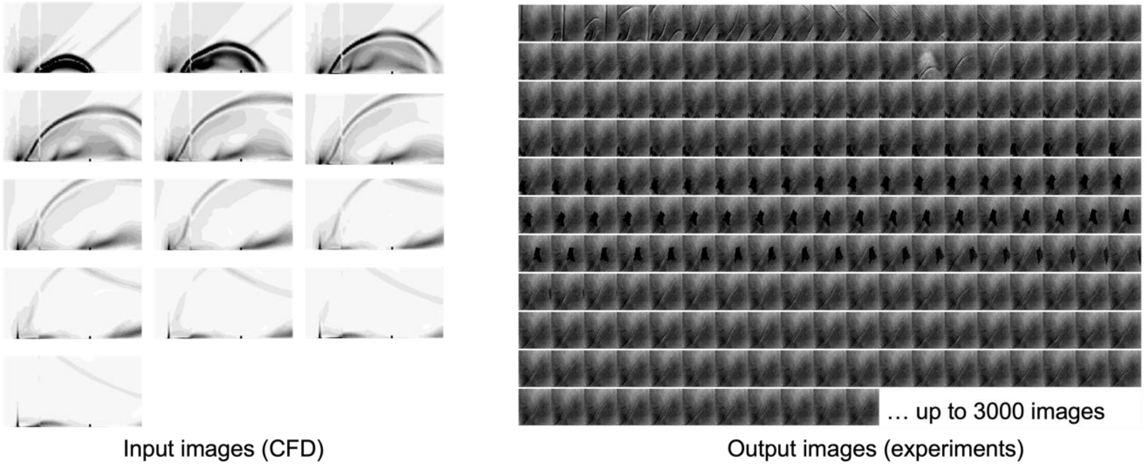

Figure 11. CFD and experimental images for matching.

Our objective was to create ML model that finds the top k images that are most similar to an input image. In our solution we used VGG16 [25] convolutional neural network architecture to convert an image to a vector of size 4096 elements. After that, we used k-nearest neighbors algorithm to find similar images to the input image based on the cosine similarity between vectors. Cosine similarity is defined for two vectors **A** and **B** as follows:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

where i is a vector component index.

First, we fed 224x224 pixels size shadowgraph images to the input of the VGG16 neural network. We used 3000 experimental images for matching. We obtained vectors of size 4096 elements as an output for each image. We used VGG16 implementation from Keras library [30]. The code was written in Python.

The second step included similar images search using k-nearest neighbors algorithm and cosine similarity of the vectors. The input on the second step was a new CFD image and the output was an array of k similar shadowgraph images, processed on the previous step, and the corresponding similarity score.

Figure 12 shows some examples of similar images matching for k = 2 and the corresponding similarity scores. Our model successfully identifies up to 50% of the image pairs at k = 1 (prediction with the highest similarity score) and up to 98% at k = 7. Thus, for each of the given CFD images, our model will search through 3000 experimental images, and it is guaranteed to find a suitable experimental one in a sample of 7 images.

Although the search for similar pairs of images does not work as well as if it were done by a human, nevertheless, the developed solution can narrow the search area from several thousand to less than 10 images. Thus, our solution can be considered as a recommender system for researchers.

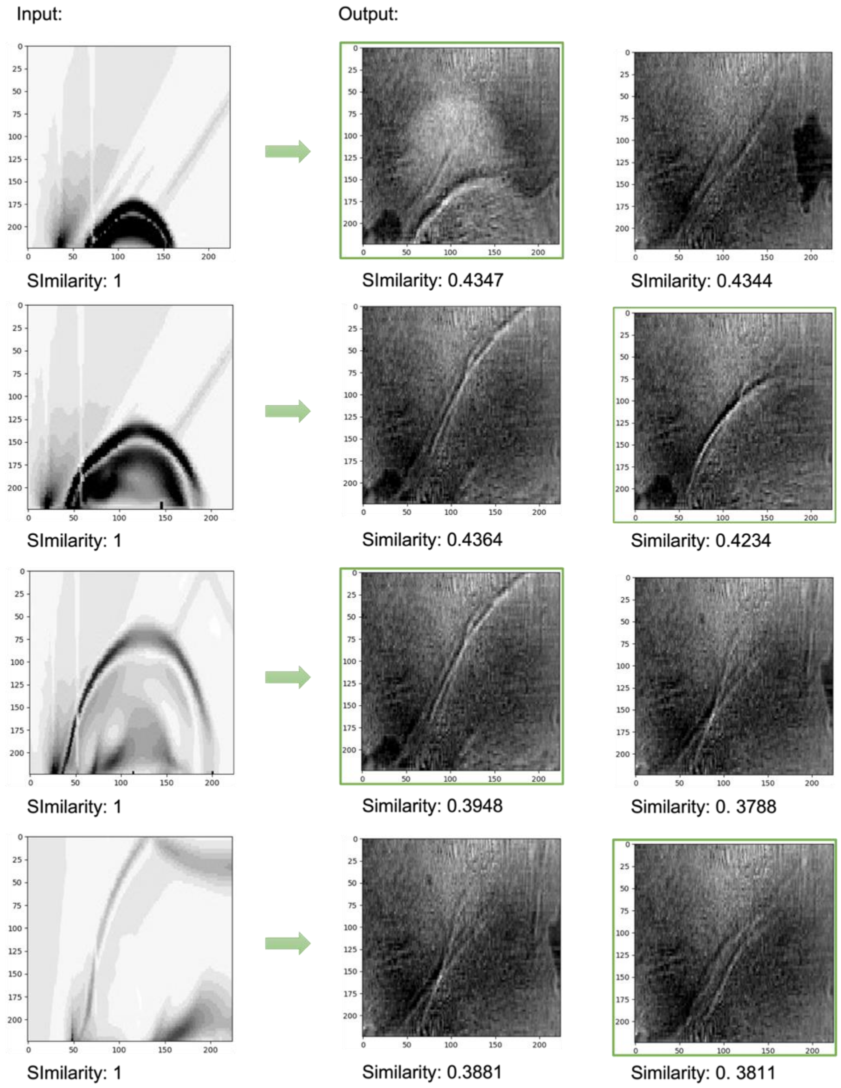

Figure 12. Similar image search using VGG16 deep neural network and k-nearest neighbors algorithm. Green bounding box indicates the correct similar image detection. Input is CFD image, output is an array of similar shadowgraph images and similarity scores.

5. Conclusions

We successfully applied deep learning methods to automate data analysis in experimental fluid dynamics and CFD. We successfully trained a CNN based on the YOLOv2 architecture to detect three types of objects on shadowgraph images: shocks, plumes and particles. Auto ML tools were used to optimize model and hyperparameters. We used the model to study flow in shock tube in time range from 0 to 40 ms. The model automatically processed large amount of shadowgraph images and tracked all discontinuities and particles in the flow. Testing the model gives the following results: 92% correct answers for training set and 88% for test set. The obtained information allowed us to obtain important physical information about the flow. Also, our model obtained excellent results when processing a more complex flow created by the pulsed electrical discharge arrangement.

We firs applied deep learning to automatically compare CFD and experimental shadowgraph images and find pairs of images for the same flow stage. Our algorithm is based on the VGG16 deep neural network for feature vector extraction and k-nearest neighbors algorithm for finding the most similar images based on the cosine similarity. Our model successfully identifies up to 50% of the image pairs at k = 1 (prediction with the highest similarity score) and up to 98% at k = 7. Thus, for each of the given CFD images, our model will search through all of the experimental images, and it is guaranteed to find a suitable experimental image in a sample of 7 results.

6. Acknowledgements

This study was supported by the Russian Science Foundation (Grant No. 22-79-00054).

7. References

[1] J. Rienitz, "Schlieren experiment 300 years ago," Nature, vol. 254, p. 293–295, 1975, doi: 10.1038/254293a0.

[2] G. S. Settles, Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media, Springer Science & Business Media, 2012.

[3] R. Goldstein, Optical systems for flow measurement: shadowgraph, schlieren, and interferometric techniques, Hemisphere Publishing Corp, 1983.

[4] S. L. Brunton, "Applying machine learning to study fluid mechanics," Acta Mechanica Sinica, vol. 37, p. 1718–1726, 2021, doi: 10.1007/s10409-021-01143-6.

[5] S. L. Brunton, B. R. Noack and P. Koumoutsakos, "Machine Learning for Fluid Mechanics," Annual Review of Fluid Mechanics , vol. 52, pp. 477-508. doi: https://doi.org/10.1146/annurev-fluid-010719-060214, 2020.

[6] C. Liu, R. Jiang, D. Wei, C. Yang, Y. Li, F. Wang and X. Yuan, "Deep learning approaches in flow visualization," Advances in Aerodynamics, vol. 4, no. 17, 2022, doi: https://doi.org/10.1186/s42774-022-00113-1.

[7] L. Guo, S. Ye, J. Han, H. Zheng, H. Gao, D. Chen, J.-X. Wang and C. Wang, "SSR-VFD: Spatial Super-Resolution for Vector Field Data Analysis and Visualization," in IEEE Pacific Visualization Symposium (PacificVis), Tianjin, 2020, doi: 10.1109/PacificVis48177.2020.8737.

[8] H. Gao, L. Sun and J.-X. Wang, "Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels," Physics of Fluids, vol. 33, no. 7, 2021, doi: 10.1063/5.0054312.

[9] K. Höhlein, M. Kern, T. Hewson and R. Westermann, "A comparative study of convolutional neural network models for wind field downscaling," Meteorol Appl, vol. 27, no. 6, 2020, doi: 10.1002/met.1961.

[10] K. Franz, R. Roscher, A. Milioto, S. Wenzel and J. Kusche, "Ocean eddy identification and tracking using neural networks," in Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, 2018, doi: 10.48550/arXiv.1803.07436.

[11] M. Monfort, T. Luciani, J. Komperda, B. Ziebart, F. Mashayek and G. E. Marai, "A Deep Learning Approach to Identifying Shock Locations in Turbulent Combustion Tensor Fields," In: Schultz T, Özarslan E, Hotz I (eds). Modeling, Analysis, and Visualization of Anisotropy. Mathematics and Visualization. Springer, Cham., p. 375–392, 2017.

[12] Y. Liu, Y. Lu, Y. Wang, D. Sun, L. Deng, F. Wang and Y. Lei, "A CNN-based shock detection method in flow visualization," Comput Fluids, vol. 184, pp. 1-9, 2019, doi: 10.1016/J.COMPFLUID.2019.03.022.

[13] M. Berenjkoub, G. Chen and T. Günther, "Vortex boundary identification using convolutional neural network," in Proceedings of the 2020 IEEE Visualization Conference (VIS), Salt Lake City, 2020.

[14] A. D. Beck, J. Zeifang, A. Schwarz and D. G. Flad, "A neural network based shock detection and localization approach for discontinuous Galerkin methods," J Comput Phys, vol. 423, 2020, doi: 10.48550/arXiv.2001.08201.

[15] M. Morimoto, K. Fukami and K. Fukagata, "Experimental velocity data estimation for imperfect particle images using machine learning," Physics of Fluids, vol. 33, no. 8, 2021, doi: 10.1063/5.0060760.

[16] B. N. Ubald, P. Seshadri and A. Duncan, "Density Estimation from Schlieren Images through Machine Learning," arXiv:2201.05233, 2022, doi: 10.48550/arXiv.2201.05233.

[17] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vols. PAMI-8, no. 6, pp. 679-698, 1986.

[18] S. Cui, Y. Wang, X. Qian and Z. Deng, "Image Processing Techniques in Shockwave Detection and Modeling," Journal of Signal and Information Processing, vol. 4, pp. 109-113, 2013, doi: 10.4236/jsip.2013.43B019.

[19] G. Li, M. B. Agir, K. Kontis, T. Ukai and S. Rengarajan, "Image Processing Techniques for Shock Wave Detection and Tracking in High Speed Schlieren and Shadowgraph Systems," Journal of Physics: Conference Series, vol. 1215, 2019, doi: 10.1088/1742-6596/1215/1/012021.

[20] I. A. Znamenskaya, I. A. Doroshchenko, N. N. Sysoev and D. I. Tatarenkova, "Results of Quantitative Analysis of High-Speed Shadowgraphy of Shock Tube Flows Using Machine Vision and Machine Learning," Doklady Physics, vol. 66, pp. 93-96, 2021, doi: 10.1134/S1028335821040066.

[21] I. A. Znamenskaya and I. A. Doroshchenko, "Edge detection and machine learning for automatic flow structures detection and tracking on schlieren and shadowgraph images," Journal of Flow Visualization and Image Processing, vol. 28, no. 4, pp. 1-26, 2021, doi: 10.1615/JFlowVisImageProc.2021037690.

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," arXiv, 2016, doi: 10.48550/arXiv.1612.08242.

[23] Y.-W. Chen, Q. Song and X. Hu, "Techniques for Automated Machine Learning," arXiv, 2019, doi: 10.48550/arXiv.1907.08908.

[24] Apple, "Create ML Overview - Machine Learning - Apple Developer," [Online]. Available: https://developer.apple.com/machine-learning/create-ml/. [Accessed 9 5 2022].

[25] X. Zhang, J. Zou, K. He and J. Sun, "Accelerating Very Deep Convolutional Networks for Classification and Detection," arXiv, 2015, doi: 10.48550/arXiv.1505.06798.

[26] I. Znamenskaya, E. Koroteeva, I. Doroshchenko and N. Sysoev, "Evolution and fluid dynamic effects of pulsed column-shaped plasma," Experimental Thermal and Fluid Science, vol. 109, 2019, doi: 10.1016/j.expthermflusci.2019.109868.

[27] I. A. Znamenskaya, D. I. Dolbnya, I. E. Ivanov, T. A. Kuli-zade and N. N. Sysoev, "Pulse volume discharge behind shock wave in channel flow with obstacle," Acta Astronautica, vol. 195, pp. 493-501, 2022, doi: 10.1016/j.actaastro.2022.03.031.

[28] Apple, "Core ML | Apple Developer Documentation," [Online]. Available: https://developer.apple.com/documentation/coreml. [Accessed 9 5 2022].

[29] Apple, "Vision | Apple Developer Documentation," [Online]. Available: https://developer.apple.com/documentation/vision. [Accessed 9 5 2022].

[30] Keras, "VGG16 and VGG19," [Online]. Available: https://keras.io/api/applications/vgg/. [Accessed 11 5 2022].