doi 10.26089/NumMet.v24r217

Analysis of the experimental flow shadowgraph images by computer vision methods

Igor A. Doroshchenko
Lomonosov Moscow State University, Faculty of Physics, Moscow, Russia
ORCID: 0000-0002-0488-0020, e-mail: doroshenko.igor@physics.msu.ru

Abstract: In this study, two examples of physical experiment automation using computer vision and deep learning techniques are considered. The first of them involves the use of classical computer vision techniques to detect and track the oblique shock wave on the experimental shadowgraph images. This was achieved using Canny edge detection and Hough transform, which allowed to obtain the line equation corresponding to the oblique shock wave. By automatically calculating the angle of this wave for each frame in the video, the process of extracting quantitative information from flow visualizations was significantly accelerated. In the second example, a convolutional neural network was trained to identify four classes of objects on the shadowgraph images, namely vertical shock waves, bow shocks, plumes, and opaque particles in the flow. The custom object detection model is based on the up-to-date YOLOv8 architecture. To realize this task, a dataset of 1493 labeled shadowgraph images was collected. The model showed excellent performance during the learning process, with model precision and mAP50 scores exceeding 0.9. It was successfully applied to detect objects on the shadowgraph images, demonstrating the potential of deep learning techniques for automating the processing of flow visualizations. Overall, this study highlights the significant benefits of combining classical computer vision algorithms with deep learning techniques in the automation of physical experiments. However, classical algorithms demand the writing additional code to extract the required information. The deep neural networks can perform this task automatically, provided that a well-annotated dataset is available. This approach offers a promising avenue for accelerating the analysis of flow visualizations and the extraction of quantitative information in physical experiments.

Keywords: computer vision, digital image processing, object detection, convolutional neural networks, flow visualization, shock wave, bow shock.

Acknowledgements: This study was supported by the Russian Science Foundation (Grant No. 22–79–00054). Author thanks prof. I. A. Znamenskaya and prof. A. E. Lutsky for the help in conducting experiments and analyzing the results.

For citation: I. A. Doroshchenko, “Analysis of the experimental flow shadowgraph images by computer vision methods,” Numerical Methods and Programming. **24** (2), 231–242 (2023). doi 10.26089/NumMet.v24r217.

2023, **24** (2), 231–242. doi 10.26089/NumMet.v24r217

Анализ экспериментальных теневых изображений течений методами компьютерного зрения

И. А. Дорощенко

Московский государственный университет имени М. В. Ломоносова,
физический факультет, Москва, Российская Федерация
ORCID: 0000-0002-0488-0020, e-mail: doroshenko.igor@physics.msu.ru

Аннотация: В данном исследовании рассматриваются два примера автоматизации физических экспериментов с использованием методов компьютерного зрения и глубокого обучения. В первом из них были применены классические алгоритмы компьютерного зрения для обнаружения и отслеживания косого скачка уплотнения на экспериментальных теневых изображениях: метод выделения границ Кэнни и преобразование Хафа. Получив уравнение прямой, соответствующей косому скачку уплотнения, угол ее наклона автоматически рассчитывался для каждого кадра видео. Во втором примере была обучена сверточная нейронная сеть для определения четырех классов объектов на теневых изображениях: вертикальных ударных волн, головных ударных волн, термиков и непрозрачных частиц в потоке. Модель основана на современной архитектуре YOLOv8. Для реализации этой задачи был создан набор данных из 1493 размеченных теневых изображений. Модель показала хорошие метрики в процессе обучения: точность модели и оценка mAP50 превысили 0,9. Она была успешно применена для обнаружения объектов на теневых изображениях. Было продемонстрировано, что применение классических алгоритмов компьютерного зрения и глубокого обучения может значительно ускорить обработку визуализаций потоков и извлечение количественной информации. Однако классические алгоритмы обычно не могут использоваться напрямую и требуют от исследователя написания дополнительного кода для извлечения необходимой информации. Глубокие нейронные сети могут выполнить эту задачу автоматически, и единственное, что требуется, это разметка и сбор большего набора данных.

Ключевые слова: компьютерное зрение, цифровая обработка изображений, детектирование объектов, сверточные нейронные сети, визуализация течений, ударная волна, головная ударная волна.

Благодарности: Исследование выполнено при финансовой поддержке РНФ (грант № 22–79–00054). Автор выражает благодарность проф. И. А. Знаменской и проф. А. Е. Луцкому за помощь в проведении экспериментов и анализе результатов.

Для цитирования: Дорощенко И.А. Анализ экспериментальных теневых изображений течений методами компьютерного зрения // Вычислительные методы и программирование. 2023. **24**, № 2. 231–242. doi 10.26089/NumMet.v24r217.

1. Introduction.

1.1. Deep learning and computer vision applications. Computer Vision (CV) is a field of artificial intelligence that enables users to extract information from visual sources of information. CV algorithms began to be developed and implemented in the 1960s. It has been widely used in various business and scientific applications. CV became a much more powerful tool with the advent of modern deep learning models in the 2010s. A major breakthrough in computer vision was made in 2012, when the AlexNet [1] model outperformed its competitors by 10% on the ImageNet dataset. The use of CV is most in demand in industries such as transportation, medicine, manufacturing, construction, agriculture, retail, etc. In the transportation industry, it is used in the development of self-driving cars, pedestrian detection, parking occupancy detection, traffic analysis and road condition monitoring. In medicine, CV is applied for X-ray and MRI image analysis, cancer detection, etc. In manufacturing, this technology is often employed for defect detection, text and barcode reading, and product assembly. CV is also widely utilized in scientific applications. For instance, it helps to improve image quality, detect special objects on images, highlight certain objects, remove noise, apply different filters. CV is used to solve tasks including: image thresholding, morphological transformations, image gradient calculation, edge, contour detection, image histogram equalisation, template matching, different shape detection, image segmentation, background subtraction, feature and corner detection, etc. Deep neural networks allow to significantly improve the solution of such CV tasks as:

- Image classification.
- Image classification with localization.
- Object detection.
- Object segmentation.
- Image colorization.
- Image reconstruction.

In this study, a classical computer vision approach for object detection and deep learning based neural network are used. The images containing objects are obtained in the physical experiment — high-speed flow visualizations using the shadowgraph technique.

1.2. Flow visualization. Since the 19th century, the visualization of density fields in transparent media has been accomplished mainly using schlieren and shadowgraph techniques [2, 3], both of which rely on the refraction phenomenon. High-speed flow recording using these techniques is commonly employed to examine unsteady gas flows, including shock waves, contact discontinuities and other flow structures. Modern digital high-speed cameras can capture videos with extremely high frame rates up to 10 000 000 fps, making it essential to create software systems capable of recognizing and measuring flow structures automatically without any handwork. Currently, the most promising methods to solve the problem involve digital image processing with various edge detection, segmentation, image filtering, template matching, object recognition algorithms [4, 5] and machine learning as well [6, 7].

1.3. Classical computer vision methods. One of the most important CV tools for simple image processing is edge detection. Several edge detection algorithms are appropriate for shock wave detection, such as Prewitt, Roberts, Sobel, and Canny [8]. As it was shown in work [9], the Canny algorithm is the most suitable for schlieren and shadowgraph image processing. In papers [5, 10], authors applied custom shock detection method based on the edge detection, image segmentation, feature detection and tracking and line detection. The bilateral filter was used to remove image noise preserving edges. The detection success rate was 94–97% depending on the object type. To measure shock stand-off distance from a model within a supersonic wind tunnel, a shock wave detection software utilizing various edge detection algorithms was developed in work [8]. Laplacian of Gaussian (LoG) and Canny methods provided the most precise results, particularly for image sequences with high noise levels. A software for detecting and tracking shock waves was elaborated in [9]. A modified version of the Canny algorithm was developed there, including background image subtraction in the frequency domain and additional filtering before edge detection to enhance image quality. The Canny edge detection method was successfully applied to CFD (Computational fluid dynamics) solutions by replacing pixel brightness with the pressure values within the CFD grid [11]. Authors of the work [12] analyzed the edge-enhanced schlieren images of the mixing process within an ejector using the Canny algorithm for image processing. The results of this study showed that the line density increased in the turbulent mixing region. The non-mixed length along the ejector was estimated by computing the ratio of the vertical density of lines at a particular location x to the maximum vertical line density across the entire ejector. There are papers devoted to CV applications not only for the experimental images, but also for the CFD fields of flow parameters. Edge detection based shock detectors may be created for the synthetic schlieren images calculated from the CFD density field [13, 14].

Researchers often need to not only identify objects in an image, but also estimate the flow velocity. The Particle Image Velocimetry (PIV) method has proven to be effective, involving seeding the flow with tracer particles and measuring their displacements between two consecutive frames [15]. The method is based on calculating cross-correlation. A variety of open source and commercial software has been developed for PIV image processing and experiment automation [16]. However, there are the non-seeded methods for measuring flow velocities from images of flow structures as well. Authors of paper [17] developed a method for flow velocity estimation. This method is a minimization-based procedure where two terms in the cost function have been designed for the images under study. The data term was deduced from the physical dependence between the luminance function and the flow density gradient. The vast majority of approaches to measuring the seedless flow velocity are based on the cross-correlation algorithms. The non-seeded methods based on the cross-correlation are known as Schlieren Image Velocimetry (SIV). The simplest way to perform SIV is to use commercial or open-source PIV software [18]. Authors of manuscript [19] used the open source PIV software to measure flow velocity of the heated air spreading from the object under study. Usually, flow structures differ significantly from the particles used for flow seeding. Therefore, PIV software is not always suitable, and it becomes necessary to write custom scripts for image processing. As it was concluded in paper [20], the existing PIV software packages do not perform well when processing schlieren images, but custom-written normalized cross correlation code gives better results. To implement the normalized cross-correlation, the function `normxcorr2` from MATLAB or the function `matchTemplate` from Python OpenCV library may be used. To improve velocity measurement resolution, different image upscaling methods such as a bicubic interpolation may be applied [21].

1.4. Machine learning and deep learning methods. Over the past decade, deep learning methods have been increasingly utilized in the fields of fluid dynamics and flow visualization, as demonstrated in a number of papers. For instance, authors of work [6] published a detailed review on the applications of deep learning in flow visualization. The convolutional neural networks such as Resnet, Unet, and IVD-Net were effectively employed by authors of manuscript [22] to detect vortices. Meanwhile, Beck et al. framed flow structure detection as an image processing problem in paper [23]. Morimoto et al. [24] proposed a method for filling missing regions on PIV images through supervised machine learning using artificial images with particles as input data and velocity vectors as output. This model successfully identified hidden flow structures that were undetectable by traditional methods. Additionally, Ubald et al. [25] presented a Gaussian process model-based approach for extracting quantitative flow information from schlieren images. Various deep learning techniques were also explored for shock wave detection in a number of other papers. A deep learning approach for identification of shock locations in large tensor field datasets generated in process of two- and three-dimensional simulations of turbulent combustion were presented in [26]. The developed neural network approximated the numerical schlieren values more efficiently than the direct density gradient calculation. The different machine learning techniques to extract quantitative information from the schlieren images of the plasma channel in the tenuous vapor were used in work [27]. The neural network capable of classifying vortex wakes behind an oscillating airfoil was developed in paper [28]. The neural network distinguished 3 wake types: $2S$, $2P + 2S$ and $2P + 4S$. In work [29], authors proposed a two-level scheme for estimating speed change on the airplane surface. This scheme includes the extraction of shock features from the schlieren images and the classification of feature vectors. Each class represents one case of speed change.

The computer vision and the deep learning applications for schadowgraph images processing will be considered in the present study. The classical computer vision methods such as Canny edge detection, Hough transform are used here for shock waves detection. Also, a deep neural network based on the YOLOv8 (You Only Look Once) architecture is developed and used for object detection on the shadowgraph images. All the shadowgraph images used in the study were obtained in the experiments at Faculty of Physics, Lomonosov Moscow State University.

2. The datasets.

2.1. Schadowgraph images for classical computer vision analysis. In this work, a set of two thousand consecutive monochrome shadowgraph images taken at the recording frame rate of 150 000–300 000 fps was used for classical computer vision analysis. They contain oblique shock waves generated by the small obstacle in a supersonic flow in a rectangular channel of the shock tube. The angle of the shock depends on the flow velocity: the higher the velocity, the smaller the angle. If the flow velocity is close to the speed of sound, the oblique shock angle is close to 90°. The purpose of the computer vision code is to accurately measure the bow shock angle for each of the given images.

2.2. The dataset for deep neural network training. The dataset for deep neural network training contains a collection of shadowgraph images of different flow structures. The dataset initially included 623 shadowgraph images with 4 object classes: “bowShock”, “particle”, “plume”, “shock” and “background”. Objects of each class were annotated: the objects’ bounding box coordinates and classes were stored and provided with the images. The original 623 images contained 963 annotations: 506 for the “shock” class, 196 for the “plume” class, 176 for the “particle” class and 85 for the “bowShock” class. A median image ratio was 226×91 px.

Initial dataset was augmented to increase images count for model training. The new versions of each image from the training set were created by blurring (up to 1 px) and adding artificial noise (up to 5% of the image pixels). The noise augmentation was chosen to help the model be more resilient to camera artifacts. The blur augmentation included a random Gaussian blur implementation to the image to help the model be more

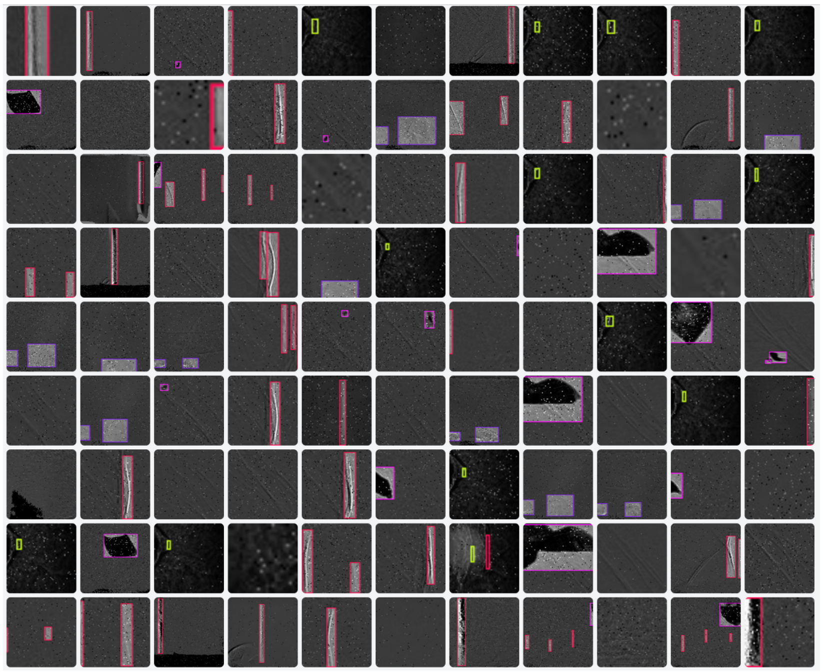

Figure 1. Random image samples from the shadowgraph images dataset

resilient to camera focus. Thus, the extended data set contained 1493 images. Some examples of the dataset are shown in Figure 1. The dataset is available as open source under the Apache 2.0 licence [30].

3. Methods.

3.1. Classical computer vision methods: Edge detection and Hough transform. Classical CV tools have been widely used in various scientific applications until the advent of machine learning based approaches. They can capture low-level features of images, making it possible to extract some quantitative information about the image. Typical shadowgraph images contain bright and dark brightness patterns corresponding to the second derivatives of the refractive index of the fluid. Shock wave shadowgraph images contain a bright band that follows the dark band due to light refraction. Such brightness patterns are well suited to edge detection algorithms.

Many CV algorithms, including edge detection, are based on the convolutional operation:

$$(f * g)(t) = \int\limits_{-\infty}^{\infty} f(u)g(t-u)\,du. \tag{1}$$

For images, convolution is the process of adding each image element to its neighboring elements (pixels), multiplied by certain coefficients (weights). It is used to highlight the characteristic structures of images, such as edges and corners. Here f is a source image and g is a kernel containing a discrete number of pixels. Thus, the convolution operation for images can be defined by the following way:

$$f(x,y) * g(x,y) = \sum_{n=0}^{columns} \sum_{m=0}^{rows} g(n,m) * f(x-n, y-m). \tag{2}$$

For example, if the Sobel operator is used as the kernel g, it is possible to implement an edge detection based on the image gradient calculation. It operates with two 3×3 kernels. They are convolved with the original image to compute approximations of the derivatives — one for horizontal changes G_x and one for vertical changes G_y:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * I, \qquad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I, \tag{3}$$

where I is a source image, G_x and G_y are image gradients.

In our study, we apply Canny edge detection and Hough transofrm to detect an oblique shock wave on the shadowgraph images and to calculate its angle. To implement the algorithm, we use the Python OpenCV library.

The Canny edge detection is an advanced algorithm for edge detection. It includes four main image processing steps: the noise reduction with a low-pass Gaussian filter, the image gradient calculation, the non-maximum suppression and the hysteresis thresholding. To realize Canny edge detection, the `Canny` method from the OpenCV library is used. The minimum and maximum thresholds are set to 150 and 300, respectively, to eliminate noise and detect only strong edges on the images, which could correspond to the shock wave.

Additionally, the Hough transform is applied to find the line equations corresponding to the shock waves. The edge images are considered as an input to the Hough transform. The main idea of the Hough transform is to convert the problem of detecting collinear points to the problem of finding concurrent curves in the Hough space $(r,\,\theta)$. To describe the Hough space, one can represent the equation of a straight line using the distance from the origin to the closest point on the straight line (r) and the angle between the horizontal axis and the line connecting the origin with this closest point (θ). Then, the equation of the line has the following form:

$$r = x\cos\theta + y\sin\theta. \tag{4}$$

Consider a single point in the plane. A set of all straight lines passing through this point corresponds to a sinusoid on the Hough plane $(r,\,\theta)$ that is unique to that point. A set of two or more points forming a straight line will create sinusoids that intersect at the point $(r,\,\theta)$ for the considered line. To implement the algorithm, the `HoughLinesP` method from the OpenCV library is used. To accurately detect the straight lines corresponding to the shock waves on the edge images, the distance r and angle θ are fixed with an accuracy of 1 and $\pi/180$ respectively. Moreover, the threshold, minimum line length and maximum line gap parameters are adjusted for our shadowgraph images to detect lines as accurately as possible.

3.2. Deep learning methods: convolutional neural network for object detection. Since the early 2010s, the rapid development of the neural networks and deep learning based object detection methods has begun. New efficient convolutional neural network architectures for object detection and segmentation have emerged: AlexNet, YOLO, SSD, RCNN, etc. Thus, the use of deep learning has made it possible to replace traditional computer vision methods, which require writing large amounts of code to solve highly specialized problems, with neural networks. Deep neural networks create the rules for object recognition automatically, without human intervention, saving a lot of time for the researchers. The only tasks that the researcher has to solve are the collection of the data set and the setting of the model hyperparameters. In the present study, I have collected a data set of the shadowgraph images containing different flow structures and trained a deep neural network based on the modern YOLOv8 [31] architecture for the object detection.

All images in the dataset are divided into 3 groups for model training, validation and testing: 1300 images for the training set, 125 images for the validation set and 63 images for the testing set.

YOLO is a popular object detection model that is designed to find and locate objects in an image with high accuracy and speed. Unlike traditional models that rely on region proposals and sliding windows to detect objects, YOLO applies a single neural network to the entire image, dividing it into a grid of cells and predicting the class probabilities and bounding boxes for each object present in each cell. This results in a real-time object detection system that can process images at a very high speed, making it suitable for use in applications like autonomous driving, security, and robotics. YOLO has undergone several iterations, with YOLOv8 being the latest and most advanced version of the model, featuring improvements in accuracy, speed, and versatility. YOLOv8 is object detection and image segmentation model. It can be trained on large datasets. It supports a wide range of hardware platforms, from CPUs to GPUs. The model was implemented in a Python environment. The model preparation process, from training to inference based on new, previously unseen images, includes the following stages:

- YOLOv8 installation.
- Preparing a custom dataset for model training.
- Model training using transfer learning.
- Model validation.
- Inference with the developed model.

Since the dataset is relatively small, then in order to improve the accuracy of the model, the transfer learning is applied. The transfer learning is a machine learning technique that involves taking a pre-trained model developed for one task and adapting it for another related task. Instead of training a new model from scratch for a specific task, transfer learning leverages the knowledge already encoded in the pre-trained model, which has learned from a large amount of data, and fine-tunes it on a smaller dataset for the new task. This approach can significantly reduce the amount of data and computation required to achieve good performance on new task and often leads to better results compared to training a new model from scratch. The model was pre-trained on the COCO dataset. The training process of the object detection model includes 50 epochs. The following metrics are used to evaluate the model: F_1-confidence, precision-confidence, precision-recall and recall-confidence. The *precision* defines the quality of a positive prediction made by the model (see equation (5) below). The *recall* is calculated as the ratio of the number of positive samples correctly classified as positive to the total number of positive samples (see (6)). And the F_1-score combines the *precision* and *recall* scores of a model. It is defined as the harmonic mean of *precision* and *recall* (7). The value of F_1-score ranges between 0 and 1, with a higher value indicating better performance. The F_1-score is a useful metric when the dataset is imbalanced, and there are significantly more instances of one class than the other. It takes into account both false positives and false negatives. The expressions for the discussed metrics are:

$$\mathit{precision} = \frac{TP}{TP + FP}, \tag{5}$$

$$\mathit{recall} = \frac{TP}{TP + FN}, \tag{6}$$

$$F_1 = 2 \cdot \frac{\mathit{precision} \cdot \mathit{recall}}{\mathit{precision} + \mathit{recall}}, \tag{7}$$

where TP is a True Positive, FP is a False Positive and FN is a False Negative.

A number of other metrics are also assessed: *box_loss*, *cls_loss*, *mAP_0.5* and *mAP_0.5:0.95*. The *box_loss* is a bounding box regression loss (Mean Squared Error). The *cls_loss* is a classification loss (Cross Entropy). The metric *mAP_0.5* represents the mean Average Precision (mAP) score when using an IoU (Intersection over Union) threshold of 0.5. On the other hand, *mAP_0.5:0.95* calculates the average mAP across multiple IoU thresholds, ranging from 0.5 to 0.95.

To better evaluate the model, a confusion matrix for each of the object classes is used as well. A confusion matrix is a table utilized in object detection to estimate the performance of the model by comparing the predicted class labels against the true class labels of objects in an image. The matrix is structured as a table with rows representing the true class labels and columns representing the predicted class labels. The cells in the matrix represent the number of objects in the image that are predicted to belong to a certain class and actually belong to that class, as well as the number of objects that are predicted to belong to a certain class but actually belong to a different class. The confusion matrix provides insights into the model's accuracy, precision, recall, and F_1 score for each class and helps in identifying areas of improvement for the object detection model.

4. Results.

4.1. Shock wave detection using the edge detection and Hough transform. To track the oblique shock wave position and its angle on the shadowgraph images, the Canny edge detection and Hough transform were applied. Figure 2 shows a schematic diagram of the algorithm. It includes the background image subtraction in the frequency domain, image noise reduction, Canny edge detection, Hough transform for line detection, and then application of some filters and limits on line length and angle to find the oblique shock wave: the line should be the longest of the detected ones and its angle should vary from 0 to 90°. The algorithm was automatically applied to each frame of the video, which contained several thousand frames.

Figure 3 demonstrates some examples of the oblique shock wave detection. The case $t = 0$ corresponds to the moment when the initial shock wave reaches the recording region. The supersonic flow following the initial shock wave produces oblique shocks at $t > 0$. This oblique wave was accurately detected. The temporal evolution of the oblique shock angle yields important physical information about the flow in the shock tube. One can clearly see that the shock angle increases with time, indicating flow deceleration. The physical results have been discussed in detail previously in works [32, 33].

4.2. Deep neural network for flow structures detection. YOLOv8 based model was trained for flow structures detection. Figure 4 shows model metrics during the training process: F_1-confidence, precision-confidence, precision-recall and recall-confidence curves.

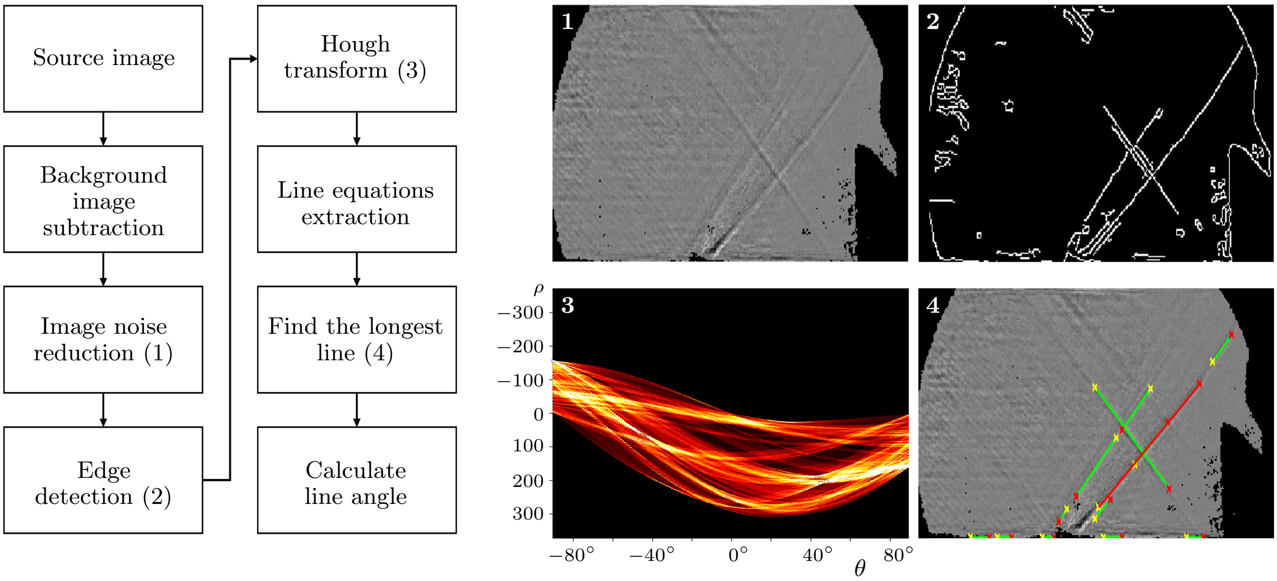

Figure 2. Algorithm for detecting oblique shock waves using the edge detection and Hough transform (left) and examples of some stages of image processing (right)

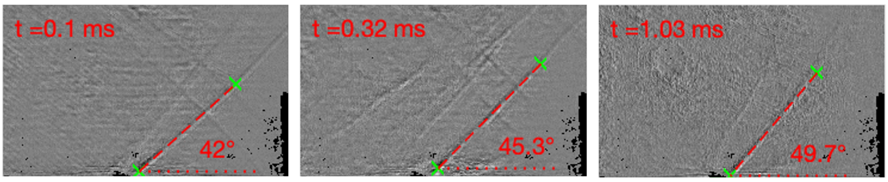

Figure 3. Automatic oblique shock wave detection and its angle calculation using Canny edge detection and Hough transform examples

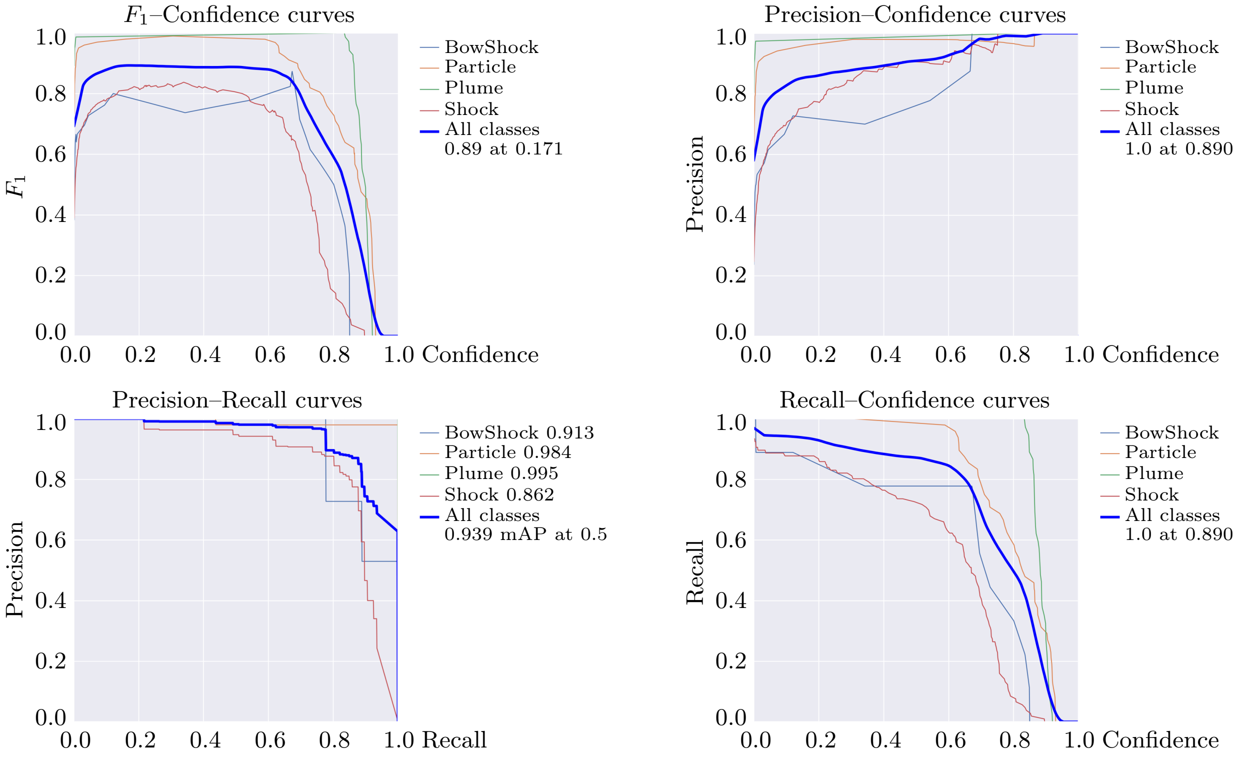

Figure 4. Key model metrics during the training process

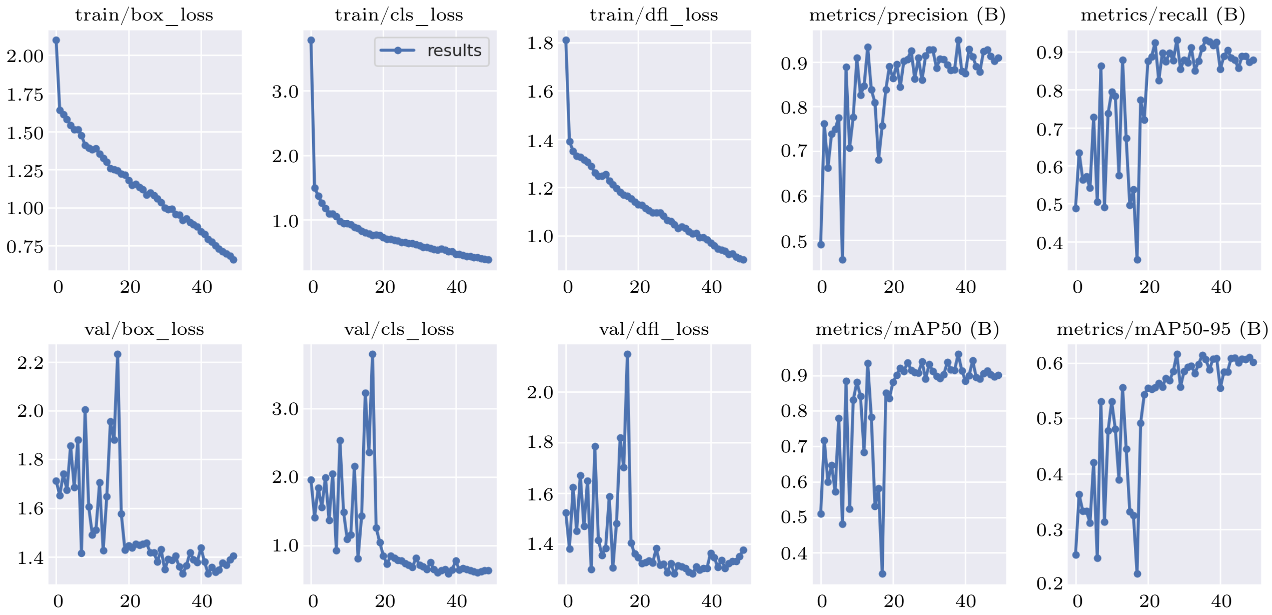

Figure 5. Model metrics during the training process

Figure 5 illustrates some additional metrics and losses: *box_loss*, *cls_loss*, *mAP_50 (B)* and *mAP_50-95 (B)*. Figure 6 demonstrates a confusion matrix for each of the object classes. The model shows quite good metrics during the learning process: model *precision*, *recall*, *mAP50* scores exceed 0.9 after 20 epochs of training. The total number of training steps was 50.

In accordance with values of the metrics and the confusion matrix, the results of the model training are quite good and allow to detect objects on the shadowgraph images almost as good as a human. The results significantly outperform our previous ones, obtained in the model based on the YOLOv2 architecture and trained on a similar dataset [32, 33].

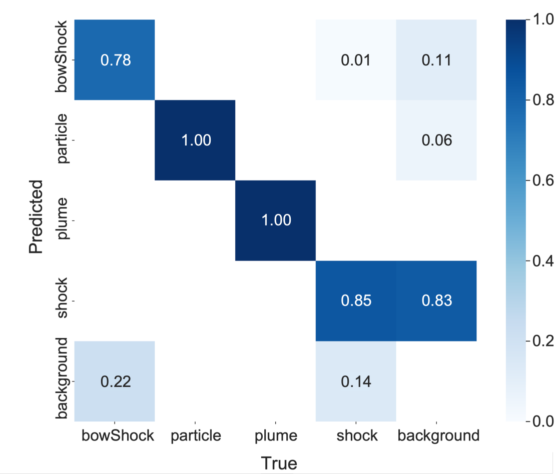

Figure 6. Confusion matrix of the model

Figure 7 shows some examples of object detection taken from the validation dataset. All classes of objects were correctly detected: particles, plumes, vertical shock waves and bow shocks. The model performed well even in such complicated cases where bow shocks and vertical shock waves were in the same small-scale image. Thus, our model can be successfully implemented for a wide range of shadowgraph images to detect flow structured on them, significantly speeding up the acquisition of new physical information from the experiment.

5. Conclusions. In the present study, two examples of automation of physical experiments by means of computer vision and deep learning techniques were described. First, a classical computer vision approach to detect and track the oblique shock wave on the experimental shadowgraph images using Canny edge detection and Hough transform was applied. After obtaining the line equation corresponding to the oblique shock wave, its angle was automatically calculated for each frame in the video. Second, a custom object detection model was trained to detect four classes of objects on the shadowgraph images: vertical shock waves, bow shocks, plumes, and opaque particles in the flow. The model is based on the modern YOLOv8 architecture. A dataset of 1493 labeled shadowgraph images for model training was collected. The model showed quite good metrics during the learning process: model *precision*, *mAP50* scores exceeded 0.9. The model was successfully applied to detect objects on the shadowgraph images.

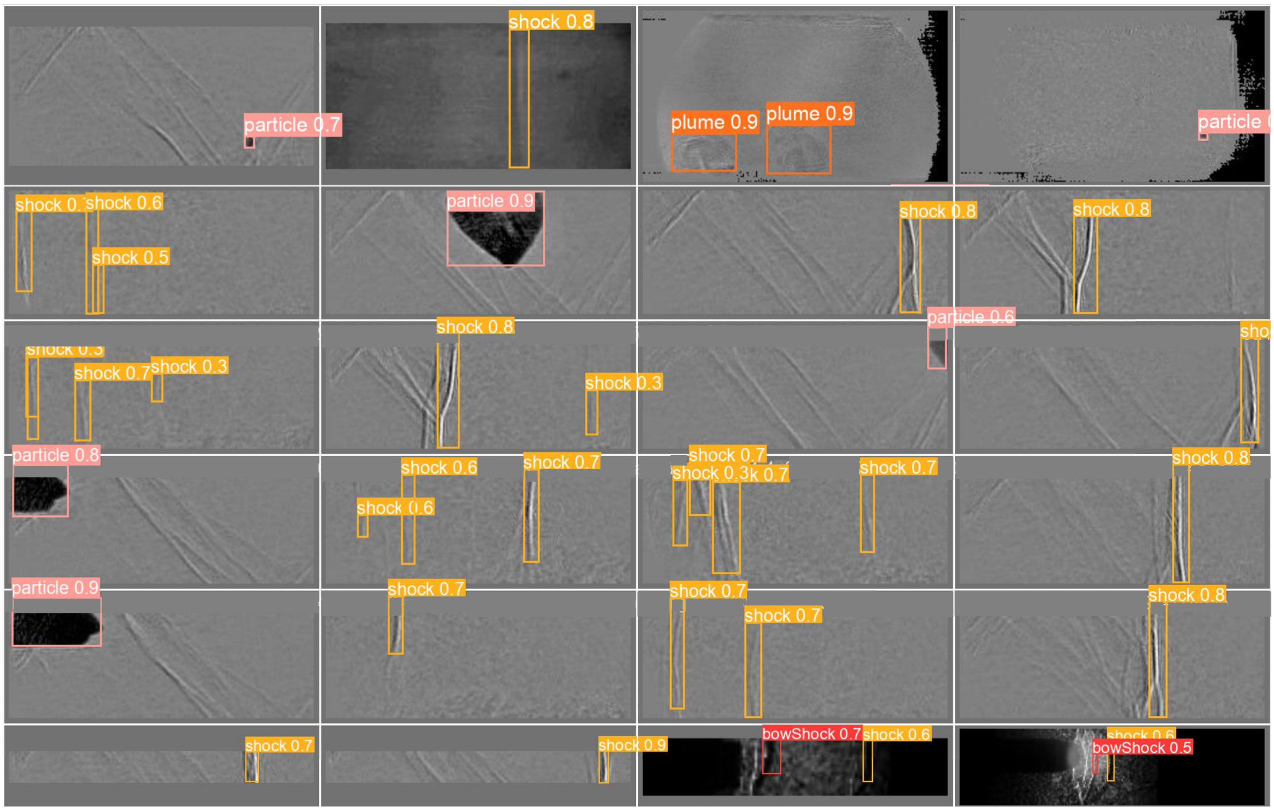

Figure 7. Object detection examples taken from the validation dataset

It was shown that the application of classical computer vision algorithms and deep learning can significantly speed up the processing of flow visualizations and the extraction of quantitative information. Classical computer algorithms usually can not be implemented directly and require the researcher to write additional code and logic to extract the required information. On the other hand, deep neural networks can do this job automatically, and the only thing that is required is a data set markup and collection.

References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Advances in Neural Information Processing Systems **25** (NIPS 2012). https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. Cited May 11, 2023.
2. J. Rienitz, “Schlieren Experiment 300 Years Ago,” Nature **254** (5498), 293–295 (1975). doi 10.1038/254293a0.
3. G. S. Settles and M. J. Hargather, “A Review of Recent Developments in Schlieren and Shadowgraph Techniques,” Meas. Sci. Technol. **28** (4), Article Number 042001 (2017). doi 10.1088/1361-6501/aa5748.
4. J. Wolfram and J. Martinez Schramm, “Pattern Recognition in High Speed Schlieren Visualization at the High Enthalpy Shock Tunnel Göttingen (HEG),” in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Springer, Berlin, 2010), Vol. 112, pp. 399–406. doi 10.1007/978-3-642-14243-7_49.
5. N. T. Smith, M. J. Lewis, and R. Chellappa, “Extraction of Oblique Structures in Noisy Schlieren Sequences Using Computer Vision Techniques,” AIAA J. **50** (5), 1145–1155 (2012). doi 10.2514/1.J051335.
6. C. Liu, R. Jiang, D. Wei, et al., “Deep Learning Approaches in Flow Visualization,” Adv. Aerodyn. **4**, Article Number 17 (2022). doi 10.1186/s42774-022-00113-1.
7. Y. Liu, Y. Lu, Y. Wang, et al., “A CNN-Based Shock Detection Method in Flow Visualization,” Comput. Fluids **184**, 1–9 (2019). doi 10.1016/j.compfluid.2019.03.022.
8. S. Cui, Y. Wang, X. Qian, and Z. Deng, “Image Processing Techniques in Shockwave Detection and Modeling,” J. Signal Inform. Process. **4** (3B), 109–113 (2013). doi 10.4236/jsip.2013.43B019.
9. G. Li, M. Burak Agir, K. Kontis, et al., “Image Processing Techniques for Shock Wave Detection and Tracking in High Speed Schlieren and Shadowgraph Systems,” J. Phys.: Conf. Ser. **1215**, Article Number 012021 (2019). doi 10.1088/1742-6596/1215/1/012021.

10. N. T. Smith, M. J. Lewis, and R. Chellappa, "Detection, Localization, and Tracking of Shock Contour Salient Points in Schlieren Sequences," AIAA J. **52** (6), 1249–1264 (2014). doi 10.2514/1.J052367.

11. T. R. Fujimoto, T. Kawasaki, and K. Kitamura, "Canny-Edge-Detection/Rankine-Hugoniot-Conditions Unified Shock Sensor for Inviscid and Viscous Flows," J. Comput. Phys. **396**, 264–279 (2019). doi 10.1016/j.jcp.2019.06.071.

12. M. V. Srisha Rao and G. Jagadeesh, "Visualization and Image Processing of Compressible Flow in a Supersonic Gaseous Ejector," J. Indian Inst. Sci. **93** (1), 57–66 (2013).

13. P. V. Bulat, K. N. Volkov, and M. S. Yakovchuk, "Flow Visualization with Strong and Weak Gas Dynamic Discontinuities in Computational Fluid Dynamics," Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie). **17** (3), 245–257 (2016). doi 10.26089/NumMet.v17r323.

14. P. V. Bulat and K. N. Volkov, "Visualization of Gas Dynamics Discontinuities in Supersonic Flows Using Digital Image Processing Methods," Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie). **20** (3), 237–253 (2019). doi 10.26089/NumMet.v20r322.

15. C. Brossard, J. C. Monnier, P. Barricau, et al., "Principles and Applications of Particle Image Velocimetry," Aerospace Lab. No. 1, 1–11 (2009). https://hal.science/hal-01180587. Cited May 6, 2023.

16. E. K. Akhmetbekov, A. V. Bilsky, Yu. A. Lozhkin, et al., "Software for Experiment Management and Processing of Data Obtained by Digital Flow Visualization Techniques (ActualFlow)," Numerical Methods and Programming. (Vychislitel'nye Metody i Programmirovanie). **7** (3), 79–85 (2006).

17. E. Arnaud, E. Mémin, R. Sosa, and G. Artana, "A Fluid Motion Estimator for Schlieren Image Velocimetry," in *Lecture Notes in Computer Science* (Springer, Berlin, 2006), Vol. 3951, pp. 198–210. doi 10.1007/11744023_16.

18. M. Lawson, M. Hargather, G. Settles, et al., "Focusing-Schlieren PIV Measurements of a Supersonic Turbulent Boundary Layers," in *Proc. 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, Orlando, USA, January 5–8, 2009.* https://arc.aiaa.org/doi/abs/10.2514/6.2009-69. Cited May 5, 2023.

19. A. W. Gena, C. Voelker, and G. S. Settles, "Qualitative and Quantitative Schlieren Optical Measurement of the Human Thermal Plume," Indoor Air **30** (4), 757–766 (2020). doi 10.1111/ina.12674.

20. M. J. Hargather, M. J. Lawson, G. S. Settles, and L. M. Weinstein, "Seedless Velocimetry Measurements by Schlieren Image Velocimetry," AIAA J. **49** (3), 611–620 (2011). doi 10.2514/1.J050753.

21. M. Debella-Gilo and A. Kääb, "Sub-Pixel Precision Image Matching for Measuring Surface Displacements on Mass Movements Using Normalized Cross-Correlation," Remote Sens. Environ. **115** (1), 130–142 (2011). doi 10.1016/j.rse.2010.08.012.

22. M. Berenjkoub, G. Chen, and T. Günther, "Vortex Boundary Identification Using Convolutional Neural Network," in *Proc. 2020 IEEE Visualization Conference (VIS), Salt Lake City, USA, October 25–30, 2020* (IEEE Press, New York, 2020), pp. 261–265. doi 10.1109/VIS47514.2020.00059.

23. 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. **423**, Article Number 109824 (2020). doi 10.1016/j.jcp.2020.109824.

24. M. Morimoto, K. Fukami, and K. Fukagata, "Experimental Velocity Data Estimation for Imperfect Particle Images Using Machine Learning," Phys. Fluids. **33** (8), Article Number 087121 (2021). doi 10.1063/5.0060760.

25. B. N. Ubald, P. Seshadri, and A. Duncan, "Density Reconstruction from Schlieren Images through Bayesian Nonparametric Models," arXiv preprint: 2201.05233v3 [physics.flu-dyn] (Cornell Univ. Library, Ithaca, 2022). doi 10.48550/arXiv.2201.05233.

26. M. Monfort, T. Luciani, J. Komperda, et al., "A Deep Learning Approach to Identifying Shock Locations in Turbulent Combustion Tensor Fields," in *Modeling, Analysis, and Visualization of Anisotropy* (Springer, Cham, 2017), pp. 375–392. doi 10.1007/978-3-319-61358-1_16.

27. G. Bíró, M. Pocsai, I. F. Barna, et al., "Machine Learning Methods for Schlieren Imaging of a Plasma Channel in Tenuous Atomic Vapor," Opt. Laser Technol. **159**, Article Number 108948 (2023). doi 10.1016/j.optlastec.2022.108948.

28. B. Colvert, M. Alsalman, and E. Kanso, "Classifying Vortex Wakes Using Neural Networks," Bioinspir. Biomim. **13** (2), Article Number 025003 (2018). doi 10.1088/1748-3190/aaa787.

29. M. D. Manshadi, H. Vahdat-Nejad, M. Kazemi-Esfeh, and M. Alavi, "Speed Detection in Wind-Tunnels by Processing Schlieren Images," Int. J. Eng. **29** (7), 962–967 (2016). doi 10.5829/idosi.ije.2016.29.07a.11.

30. Shadowgraph Images. Datasets at Hugging Face. https://huggingface.co/datasets/igor3357/shadowgraph_images. Cited May 6, 2023.

31. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics. Cited May 6, 2023.

32. I. A. Znamenskaya and I. A. Doroshchenko, “Edge Detection and Machine Learning for Automatic Flow Structures Detection and Tracking on Schlieren and Shadowgraph Images,” J. Flow Vis. Image Process. **28** (4), 1–26, (2021). doi 10.1615/JFlowVisImageProc.2021037690.

33. 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,” Dokl. Akad. Nauk **497** (1), 16–20 (2021) [Dokl. Phys. **66** (4), 93–96, (2021)]. doi 10.1134/S1028335821040066.

Received
March 28, 2023

Accepted for publication
May 1, 2023

Information about the author

Igor A. Doroshchenko — Ph.D., Scientist; Lomonosov Moscow State University, Faculty of Physics, Leninskie Gory, 1-2, 119991, Moscow, Russia.