International Journal of Multiphase Flow 126 (2020) 103194

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow

Bubble patterns recognition using neural networks: Application to the analysis of a two-phase bubbly jet

Igor Poletaev[a,*], Mikhail P. Tokarev[b,c], Konstantin S. Pervunin[b,c]

[a] Google LLC, 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
[b] Kutateladze Institute of Thermophysics, Siberian Branch of the Russian Academy of Sciences (IT SB RAS), 1 Lavrentyev Ave., 630090, Novosibirsk, Russian Federation
[c] Department of Physics, Novosibirsk State University (NSU), 2 Pirogov Str., 630090, Novosibirsk, Russian Federation

ARTICLE INFO

Article history:
Received 1 August 2019
Revised 19 November 2019
Accepted 24 December 2019
Available online 26 December 2019

Keywords:
Computer vision
Machine learning
Neural networks
Bubbles recognition
Two-phase bubbly jet
Planar Fluorescence for Bubbles Imaging (PFBI)

ABSTRACT

Gas-liquid two-phase bubbly flows are found in different areas of science and technology such as nuclear energy, chemical industry, or piping systems. Optical diagnostics of two-phase bubbly flows with modern panoramic techniques makes it possible to capture simultaneously instantaneous characteristics of both continuous and dispersed phases with a high spatial resolution. In this paper, we introduce a novel approach based on neural networks to recognize bubble patterns in images and identify their geometric parameters. The originality of the proposed method consists in training of a neural network ensemble using synthetic images that resemble real photographs gathered in experiment. The use of neural networks in combination with automatically generated data allowed us to detect overlapping, blurred, and non-spherical bubbles in a broad range of volume gas fractions. Experiments on a turbulent bubbly jet proved that the implemented method increases the identification accuracy, reducing errors of various kinds, and lowers the processing time compared to conventional recognition methods. Furthermore, utilizing the new method of bubbles recognition, the primary physical parameters of a dispersed phase, such as bubble size distribution and local gas content, were calculated in a near-to-nozzle region of the bubbly jet. The obtained results and integral experimental parameters, especially volume gas fraction, are in good agreement with each other.

1. Introduction

Oil, gas or aerospace industries development is inevitably associated with the use of modern methods of their study. Fundamental research is required to improve existing technologies that help solve more and more challenging technical problems appearing in production. In the economy spheres mentioned above, various nonstationary two-phase bubbly flows are often encountered. Such bubbly flows can, for example, be used in petrochemical reactors – bubble columns in which bubbles are involved in the process of intensification of mass transfer between liquid and gas phases. Moreover, bubbly jets arising during the operation of some types of hydraulic equipment can lead to serious negative consequences closely related to the phenomenon of cavitation (Caupin and Herbert, 2006). From the standpoint of fundamental research of these flows, it is necessary to develop highly accurate approaches for the primary processing and analysis of experimental data the amount of which is continuously increased. Information support of experiments, including data storage and visualization, as well as data treatment with advanced algorithms, becomes a crucial element of any modern study.

To date, there are two primary classes of measurement techniques for two-phase flow diagnostics: probe-based (contact) and non-invasive (contactless) methods. The first of them makes it possible to evaluate some physical quantity (temperature, pressure or velocity) at a high frequency in a certain flow region and, thereby, derive its average and fluctuating values. The undoubtable advantage of the contact methods is the possibility of using sensors in hard-to-reach places. The second type of measurement technqiues includes most of optical approaches the development of which has been significantly accelerated along with the evolution of laser, recording and computer technologies, and favorably differ from the probe-based methods in that they do not disturb an investigated flow.

[*] Corresponding author.
E-mail address: ipoletaev@google.com (I. Poletaev).

https://doi.org/10.1016/j.ijmultiphaseflow.2019.103194

In most of existing contactless techniques for dispersed-phase analysis, for example shadow photography (Bongiovanni et al., 1997), the thickness of a measurement volume (its dimension along the line of sight of a camera) is proportional to the depth of field of an optical system used. This makes it impossible to discern and treat particles (solids, drops or bubbles) in a specific layer of a flow under consideration, excluding the rest of them. This issue is overcome in the planar fluorescence for bubbles imaging (PFBI) approach (Akhmetbekov et al., 2010; Dulin et al., 2012), probably the most effective modern technique to determine bubble parameters. It is a whole-field optical method, in which bubbles located far from the central plane of a laser-light sheet (measurement plane) become less and less visible with increasing the distance to the laser sheet , that is, the intensity of their patterns drops quickly with the distance. This technique allows obtaining high-resolution experimental data in a rectangular three-dimensional volume with the thickness of 0.5–1.5 mm.

In order to determine the parameters of bubbly flows, such as the mean velocity and turbulent characteristics of the continuous phase, the mean velocity and local gas concentration along with their fluctuations of the dispersed phase, specific software is needed that can extract information from PFBI data on the exact positions of bubbles in space and their sizes. These data can be used later, for example, by the particle image velocimetry (PIV) or particle tracking velocimetry (PTV) algorithms (Bröder and Sommerfeld, 2003; Lindken and Merzkirch, 2002). To find and identify objects in images, there exist many traditional approaches based on the extraction of "manually configurable" features, for example SIFT descriptors (Lowe, 2004) or histograms of oriented gradients (Dalal and Triggs, 2005). Another example is the study by Liu et al., 2016), where bubble detection and calculation of their sizes from experimental images were performed using the ImageJ analysis software.

The next paper byAkhmetbekov et al., 2010) proposes the following two approaches of bubble recognition, which solve the aforementioned problem to some degree of accuracy. The first technique is based on the search for bubbles by outlining their boundaries in images and then searching for related areas to determine geometric centers and sized of bubbles. The second technique uses the correlation method, where a scanned image part is compared with a pre-generated mask image, whereby the presence or absence of a bubble in the image is determined.

Both methods have a low quality of overlapping bubbles identification, because in a real experiment, bubble images, due to a variety of different effects, can be significantly distorted. Additionally, we can point out a high computational cost inherent to the correlation analysis, although it can partially identify overlapping bubbles as separate objects. Another issue is related to a lot of adjustable parameters required by both methods to operate properly. These parameters are highly coupled with a particular kind of experimental data.

In order to scale up the use of bubble recognition methods and speed up experiments, new approaches are required that will be able to overcome the above disadvantages.

In the context of this study, instead of using traditional approaches for the classification of objects in the images, to achieve the goal, we introduce a new technique based on the use of artificial neural networks (NNs). Thanks to the computational technologies, which have rapidly developed in recent decades, and the tremendous growth of available data, it has become possible to utilize NNs for a variety of applications such as image classification or object detection (Szegedy et al., 2013). Nowadays, convolutional neural networks (CNNs) are one of the most successful examples of machine learning applications, because they allow the achievement of state-of-the-art performance in solving different image processing problems.

CNNs are a part of the general concept of “deep learning" (Goodfellow et al., 2016). Deep learning approaches may be formulated as a set of machine learning technologies that are designed to extract and comprehend low- and high-level abstractions in data automatically. This approach provides a necessary flexibility in experimental data analysis, which implies that the algorithms are adaptable and simple as much as possible for configuration and use. This, in turn, allows, for example, building of an effective and robust classifier for recognizing bubble patterns in images of two-phase bubbly flow with minimal interventions from outside.

It is worthwhile mentioning that there are only several kinds of research (Ohmi, 2008; Sapkota and Ohmi, 2009, 2008) in which the authors used NNs for such a physical experiment analysis. These studies refer to modification of the PTV algorithm by NNs for increasing the efficiency and universality of the PTV method. In the recent paper by Ilonen et al., 2018), various methods for bubble detection and size estimation were compared. It was concluded that the best choice for an off-line bubbles image analysis was the application of modern CNNs.

Summarizing the above, the focus of this paper is on developing a software based on NNs and its quality evaluation. This software should be easily adaptable to experimental data for various types of turbulent bubbly flows and applicable to a wide range of volume gas contents. Furthermore, the recognition algorithms have to be capable of identifying different bubble patterns in raw images, regardless of their size, shape, intensity profile, background, and degree of overlap.

2. Neural network architectures

Since the beginning of the 90s to the present, the theory of NNs has experienced a real boom in the number of scientific publications and various applications, ranging from financial to medical fields. Moreover, NNs in some areas, such as machine translation (Johnson et al., 2017), actually reached the human level, and solving problems such as speech recognition, have long surpassed human results (Amodei et al., 2016).

Currently there are varying NN architectures used for image recognition. Despite the differences in structure, the basic principles of their operation are identical - they require labeled data to be trained on (see Section 3). After running the training procedure, NN parameters are adjusted to perform the classification task. Thus, a NN acquires the ability to correctly process previously unknown input data.

In addition to the conventional basic architecture of the multilayer perceptron (MLP) (Orbach, 1962), for computer vision, the last decade saw the increasing use of CNNs. The key idea of a CNN was first introduced by LeCun et al., 1990). The work was inspired by the principles of operation of the human visual cortex. Later, various versions of the CNNs (LeCun et al., 2010, LeCun et al., 2004) began to be used for solving different challenging computer vision issues, such as localization (Szegedy et al., 2013), segmentation (Shelhamer et al., 2017), generating sentences from images (Karpathy and Fei-Fei, 2017), object detection (Girshick et al., 2014), etc.

CNN, compared to MLP, has the following two major advantages:

- *NSpatial invariance:* Due to its architecture, CNNs don’t lose out in the process of learning information about the spatial arrangement of the input images’ details
- *Computational efficiency:* Due to distributed weights, a CNN can have a similar “depth” as a conventional perceptron but with the number of weights being an order of magnitude smaller.

With that said, in order to effectively identify bubbles in the bubbly jet images, CNNs were chosen as the base architecture.

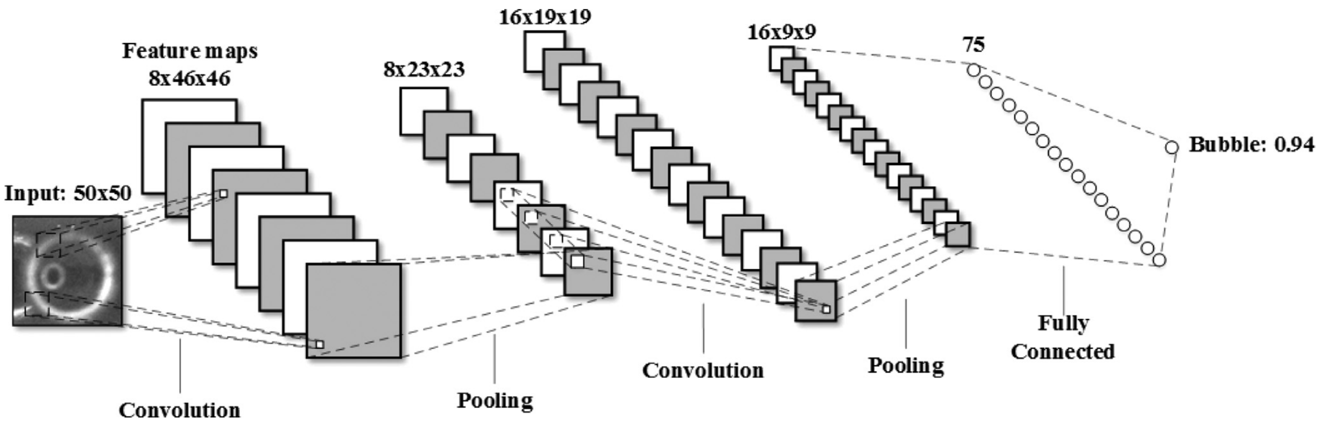

Fig. 1. Example of the CNN classifier architecture used in this study. Sizes are given for a 50 × 50 input window. The network produces the normalized probability values of the presence of a bubble in a given image.

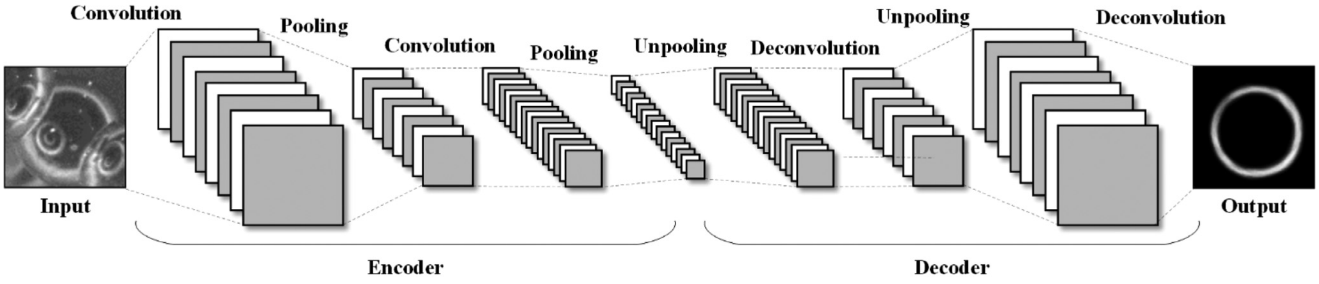

Fig. 2. Example of convolutional denoising autoencoder architecture used in this study.

In the sections below, respective CNNs will be designated, according to their architecture's parameters. For instance, *CNN_8_16_75* is the convolutional network architecture (Fig. 1), where the values 8 and 16 denote the number of convolutional kernels (LeCun et al., 1998) used in the first and second convolutional layers, respectively, whereas "75″ corresponds to the size of the penultimate fully-connected neurons layer.

However, even if the NN correctly detects a bubble in the image, this does not always mean that its geometric characteristics can be easily determined. Very often the bubble is overlapping by other bubbles or its outlines are entirely blurred. To overcome this problem, denoising autoencoders were used in this study. The following section briefly describes their operation.

2.1. Denoising autoencoders

Autoencoding is a data compression algorithm where the compression and decompression functions are data-specific, lossy and learned automatically from examples, rather than engineered by a human. Additionally, in almost all contexts where the term autoencoder is used, the compression and decompression functions are implemented with NNs. Today, two interesting practical applications of autoencoders are data denoising and dimensionality reduction for data visualization. With appropriate dimensionality and sparsity constraints, autoencoders can learn data projections that are better than the PCA or other basic techniques (Smith, 2002).

In our study, we used both MLP and CNN-based autoencoders to solve the denoising problem (Baldi, 2012; Vincent et al., 2008). Models were trained on synthetic images and their clear shapes, as shown in Fig. 2. The autoencoder aim is to extract only central and focused bubble shapes from the images and suppress surrounding noise and particles.

The advantage of using autoencoders over other similar-purpose filters [for example, the non-linear median filter with Sobel operator (Bröder and Sommerfeld, 2003)] is that we do not need to adjust processing parameters manually; an autoencoder learns how to deal with the data during the training stage. Moreover, its main distinguishing feature is that it leaves only the central, bright and focused bubble in the image, even if it contains several bubbles (see Fig. 11 and Section 5.2). Thus, this makes the autoencoder a universal tool, replacing other digital filters that can be used for this problem.

3. Image generation for training neural networks

As described in Section 2, training samples are needed in order to train a NN. From the sample of experimental images shown in Fig. 11, it can be seen that in order to determine the characteristics of the two-phase flow, it is necessary to detect preferably all bubbles and determine their parameters. Following this, it is necessary to carefully remove all detected bubbles in the image and calculate the properties of the velocity field of the continuous phase, with the help of either the PIV or PTV method. For this purpose, all image sections need to be processed successively. In order to perform that, the respective image processing algorithms divide the experimental images into a set of overlapping square windows with equal side sizes (denoted by parameter a) and an equal value of the overlapping between them (denoted by parameter s). The size of these square windows depends on the average bubble radius; in our case, it was within the range of 40 up to 80 pixels. Moreover, due to the phenomenon of overlapping bubbles and for the algorithm resolution improvement, the value of the s parameter should be lower with the increase of the gas actual content β_{exp} in the pipe (see Section 6).

Each image should be divided in such a way that can be effectively processed by the NN. The task of the NN is to determine the presence of a bubble inside a particular window as well as to calculate its geometrical characteristics (see Section 4). Thus, we need

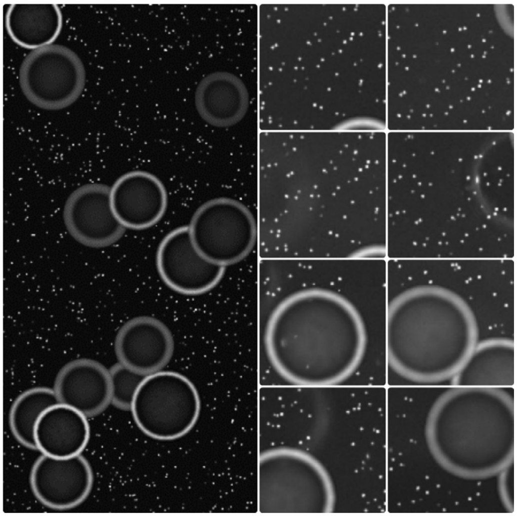

Fig. 3. Examples of the generated synthetic images of the virtual bubbly flow with the gas content value of 0.5%. The left image is the source large synthetic image. The images on the right are negative (upper) and positive (lower) training images respectively.

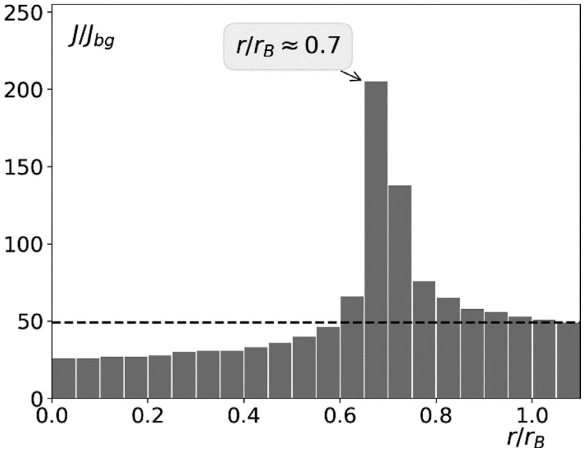

Fig. 4. Simulated brightness distribution of the intensities of a bubble image. Calculations are made for the water-air medium. The brightness is averaged on the interval $[-\frac{h}{2}, \frac{h}{2}]$, $h = 1$ mm inside the square area of 60 x 60 pixels. The bubble rim, as noted in the histogram, has a maximum brightness at a distance of about 70% of its real radius.

to obtain the respective training data to train such NNs where the input is a square image, with or without bubbles, and output is the digital representation of the information about this bubble (or bubbles) or its absence. This section describes the way to prepare this training data, as was done for this study.

Besides, it is important to note the following empirical fact: The more training samples and the more varied they are, the better the algorithm generalizes the learned information. It increases the overall NN performance in the processing of the experimental images. Unfortunately, for obvious reasons, hand-crafted labeling of the required amount of data (several tens of thousands of images) was too laborious and impossible in practice. Therefore, in the study, we leveraged synthetically generated images, which modelled the real experimental data. Artificially generated images were used to train the NNs to recognize bubbles, find their geometric centers and clean up the patterns found due to background noise. An example of an approach of artificial image synthesis with realistic bubble images can be found in Fu and Liu (2019). The next section describes the data generation approach in detail.

3.1. Synthetic training data generation

During the creation of the training data, the appearance of a bubble inside a square image was simulated, which floated inside a virtual two-phase flow with given gas content parameters (Fig. 3).

Images of the bubbles were drawn as ellipses with specified coordinates of their geometric centers as well as with the given values of the semi-axes and the rotation angle of their semi-major axis, with respect to the normal bubbly flow direction. The uniform random distribution was used to generate the position of the geometric center of the bubbles. Normal distributions were used to generate the corresponding semi-axis sizes and rotation angle values. Thus, the entire area of the created images was utilized to place the bubbles. Cases were also simulated when the center of the generated bubble was outside the current image. This approach was extremely important in providing the NN with information on as many possible real-life situations.

Physically meaningful intensity distribution of the rim of a bubble (Akhmetbekov et al., 2010) was added to the black background and superimposed with Gaussian noise and particle images, which were generated using a specific software that considers various physical conditions such as general illumination, concentration of tracers, their reflectivity, etc. To impart a greater similarity to real data, 200 square images of the background without bubbles were taken from real data. These images were used as random masks that were added to the generated examples, with a multiplier randomly sampled in the interval [0.01, 0.5] for each generated image. Such a combined approach allowed the generation of a large number of dissimilar examples to train the NNs.

3.2. Physical background

In order to simplify the task for NNs, it is necessary to model the training data as realistically as possible. Therefore, the specific intensity distribution profile of a bubble rim was generated based on the physical model described by Dulin et al., 2012). It was shown that rings corresponding to bubble images have roughly a 30% smaller diameter relative to the original diameter of bubbles (Fig. 4).

This theoretical model is well confirmed experimentally. In further detail, the results of the measurements of the intensity profiles of a single pop-up bubble can be found in the studies of Akhmetbekov et al., 2010) or Pervunin and Timoshevskiy (2016).

To ensure that the synthetic images are maximally similar to the real experimental data, the numerical experiment (Fig. 5) was carried out to construct a dependence of the bubble blurriness degree as a function of the distance between its geometric center and the laser sheet plane. Thus, all the bubbles generated in the images were distributed along the perpendicular axis to the measurement plane and illuminated by the same law as the bubbles in a real experiment (Fig. 3). The peculiarity of this approach is that it is possible to generate such artificial data for any experiment configuration. Namely, data generation algorithms can consider the different bubble sizes, background brightness distribution variations, total gas content values, the density of the tracer particles and optical registration magnification.

Finally, the approach described may be combined with machine learning. The generative adversarial networks (Goodfellow et al., 2014; Huang et al., 2018) can be used to generate the training data automatically. In addition to the performance increase, it will also make it possible to get rid of any manual settings. This approach is not considered in this study.

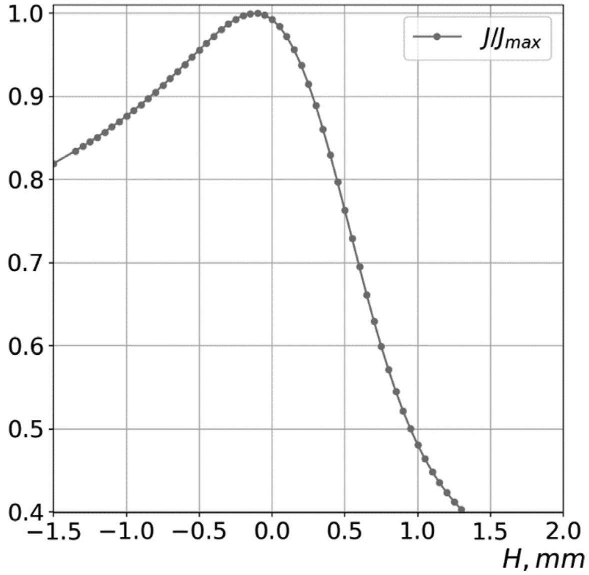

Fig. 5. Points of the calculated dependence of the maximum brightness J of the bubble ring divided by the maximum possible brightness J_{max}, when $H \approx 0$ and where H is the distance between the geometrical center of the bubble and the central plane of the laser sheet. The data are given for the case of an air bubble (real $r = 0.5$ mm) in water. H is positive when the bubble is behind the laser sheet and is negative if the bubble is located between the camera and the measurement plane.

4. Bubble recognition

To identify a bubble and determine its parameters, we propose the following algorithm, which includes the consecutive operation of the next three separate NNs:

- *Classifier*: It outputs the probability of the presence of a bubble in the processed image.
- *Centers finder*: It determines the geometric center of a bubble in a given image.
- *Autoencoder*: It clears found bubbles from the background noise (see Section 2.1).

The sections below describe how each architecture operates in detail.

4.1. Training neural networks

For all aforementioned NN architectures, as the input data, a normalized brightness map of the processed images was used (Fig. 3). Training data sets for the *classifier* and *centers finder* contained 30,000 synthetically generated square images (see Section 3) with the size $a \times a$ pixels. The value of the parameter a was equal to 50 pixels. This size was chosen because most parts of the bubbles in the experimental images can be fully fitted in such square. Input vectors for NNs (I_{11} , …, I_{aa}) are the two-dimensional array of brightness values of each pixel, $I_{lm} \leftrightarrow I(x_l, y_m)$. For the autoencoder training, we used 50,000 “noisy-clear” pairs of synthetic images (see Section 2.1).

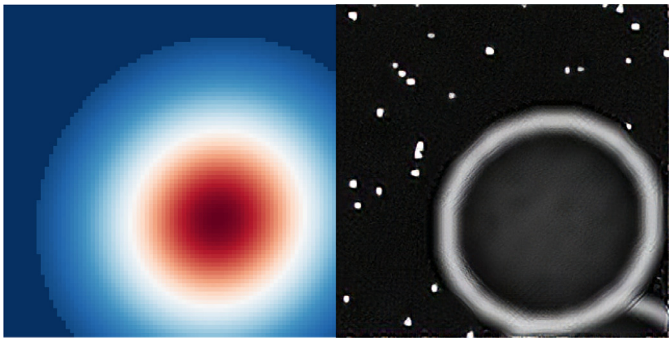

Fig. 6. An illustration of the probability distribution (1) (left) of the center of a bubble for the synthetic bubble image (right).

The *classifier* was trained for binary classification; it has one output, that represents the probability P of the presence of the desired bubble in the image. Thus, square images with and without bubbles ($P = 1$ or $P = 0$, respectively) were used as training examples. In this case, when more than one bubble is in the image, it is assumed that the network should provide the output, $P = 1$.

To train the *centers finder*, the following two-dimensional output map was used:

$$P(x,y) = e^{-a(x-x_0)^2+2b(x-x_0)(y-y_0)-c(y-y_0)^2} .$$
$$\{1 \; if \; image \; has \; bubble, \; 0 \; - \; otherwise\} \tag{1}$$

(1) is the Gaussian distribution that corresponds to the probability that some pixel with coordinates (x,y) is the geometric center of a given bubble with the real center in (x_0, y_0). Consequently, the output vectors $(t_1, \ldots, t_{a \cdot a})$ are a one-dimensional array of probabilities: $t_i \leftrightarrow P(x_i, y_i)$ (Fig. 6). Coefficients a, b, c in Eqn 1 are used to rotate the distribution by an angle θ, and are calculated as follows:

$$a = \frac{\cos{(\theta)}^2}{2\sigma_x^2} + \frac{\sin{(\theta)}^2}{2\sigma_y^2}; b = \frac{\sin{(2\theta)}}{4}\left(\frac{1}{\sigma_y^2} - \frac{1}{\sigma_x^2}\right);$$
$$c = \frac{\cos{(\theta)}^2}{2\sigma_y^2} + \frac{\sin{(\theta)}^2}{2\sigma_x^2} \tag{2}$$

The dispersion parameters σ_x, σ_y in Eqn 2 were chosen so that the function Eqn 1 satisfies the following condition Eqn 3:

$$P(\tilde{x}, \tilde{y}) \le 0.01 \; \forall (\tilde{x}, \tilde{y}) \in \; the \; rim \; of \; the \; drawn \; bubble \tag{3}$$

Thus, this NN was trained to determine the most probable bubble center coordinates, which correspond to the distribution maximum given by Eqn 1 . Such an approach has proved to be successful in view of the fact that bubble centers in situations with large gas content are often found under the other bubbles (Fig. 12).

4.2. Bubble parameters determination

In order to accurately calculate the gas content after the bubble is found in the image, it is necessary to determine its geometric parameters. For this problem, different approaches were probed. Experiments have shown that simple ellipse fitting algorithms operate better compared to a NN trained one to perform the same task. Despite that such algorithms function slower than NNs on average, they are usually more robust and precise.

It was also observed that the presence of noise in the images, in the form of random background intensities and intensity peaks from particle images, deteriorates the result of an assessment of the bubble parameters, regardless of the algorithm used. In order to minimize errors introduced by extraneous noise, we propose to use autoencoders described in Section 2.1. To determine parameters, images with bubbles, found and cleared by the autoencoder, were scanned by the algorithm that calculates the values of the correspondingly large and small semi-axes as well as the angle between the major axis and the line perpendicular to the jet axis for the ellipse. In other words, this algorithm tries to fit an ellipse into the considered bubble ring image most optimally.

For ellipse fitting in this study, we use the modified approach developed by Xie and Ji (2002), in which the authors suggested using the one-dimensional accumulator storing statistics of the five

Algorithm 1
Brightness enhancing algorithm.

```
Require: input image brightness map, $I$
    Compute average, minimum and maximum brightness values: $\bar{I}$, $I_{min}, I_{max}$;
    Subtract $\bar{I}$ from the image brightness: $I \leftarrow I - \bar{I}$ ;
    Scale obtained $I$ to the range $[I_{min}, I_{max}]$;
return $I$
```

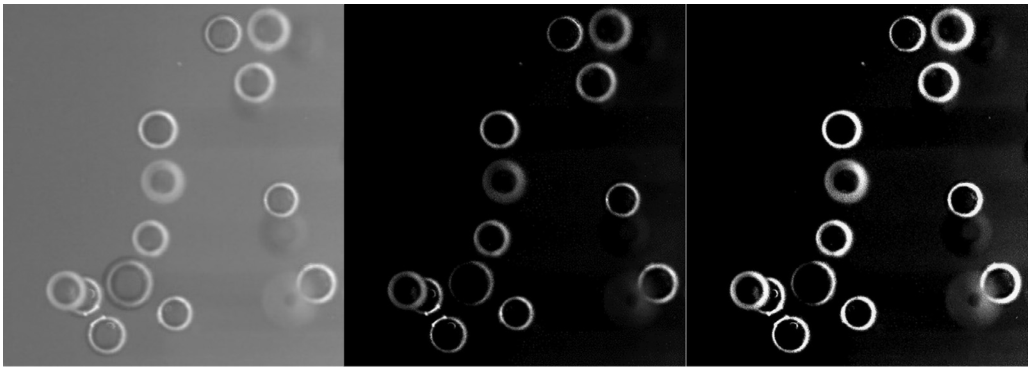

Fig. 7. From left to right the steps of the contrast enhancement procedure of the input image are presented.

parameters of the ellipse being fitted - the coordinates of its center, the size of the semi-axes and their orientation angle with respect to the line, as described above. This method was first introduced by Ho and Chen (1995). Such an approach of ellipse-fitting is significantly faster, in contrast to the time-consuming Hough transform algorithm for ellipse detection (Davies, 1989).

4.3. Experimental image processing algorithm

The first step is image preprocessing. In the course of our experiments, we found that increasing the contrast of the input experimental image provides better results (Krizhevsky et al., 2012) in terms of the bubble recognition accuracy. We enhanced the contrast as described in Algorithm 1. Example of its operation is shown in Fig. 7.

The second step uses images obtained from the previous stage, in order to identify the bubbles. A square sliding window with the side size a, moves through an experimental image with the step s. Every window is scanned by the trained *classifier*. If the NN is confident enough that the current window contains a bubble (i.e., outputted probability P is greater than some threshold t_{clf}), then the most probable coordinates of its center are determined leveraging the *centers finder*. This NN uses a fixed threshold, t_{cen}, for thresholding the output probability map. The optimum values of the aforementioned threshold parameters were defined empirically, based on the NN classification accuracy as

Algorithm 2
Bubble recognition algorithm.

```
Require: input image brightness map, $I$
Require: sliding window size, $a$
Require: sliding window moving step, $s$
Require: classifier threshold, $p_{clf}$
Require: classifier threshold, $p_{cen}$
        Initialize empty bubble_centers list;
        for sliding window $w_i$ in $I$ do
        Classifier outputs the probability $P^i$ of the bubble presence inside a given window;
        If $P > t_{clf}$, then Centers Finder returns bubble center probability distribution $p^i_{xy}$;
        If $\max p^i_{xy} > t_{cen}$ then add $(\mathrm{x}_0, \mathrm{y}_0) = \underset{(\mathrm{x},\mathrm{y}) \in \mathrm{w}_\mathrm{i}}{\mathrm{argmax}}(\mathrm{p}^\mathrm{i}_\mathrm{xy})$ to bubble_centers;
    end for
    return bubble_centers
```

Algorithm 3
Weighting clustering for bubble centers refining.

```
Require: list of bubble centers, bubble_centers
Require: nearby bubbles criteria, $R$
Initialize empty bubble_clusters list;
    Append clusters with nearby points using maximum distance $R$ criteria into bubble_clusters;
    Each cluster $k_i = \{(x_j, y_j), j = 0 \ldots |k_i|\}$ is considered to contain points of bubble $b_i$;
    for $k_i$ in bubble_clusters do
        Find maximum probability: $P^i_{max} = \max_{j=0 \ldots |k_i|} (P^j)$;
        Specify $b_i$ bubble center: $(x_i, y_i) = \frac{1}{|k_i| \cdot P^i_{max}} \cdot \sum_{j=0}^{|k_i|} [(x_j, y_j) \cdot P_j]$;
    end for
    bubble_centers $\leftarrow$ bubble_clusters;
    return bubble_centers
```

Algorithm 4
Bubble parameter determination algorithm.

```
Require: input image brightness map, $I$
Require: autoencoder input image size, $a$
Require: list of bubble centers, bubble_centers
    Initialize empty bubble_parameters list;
    for $(x_i, y_i)$ in bubble_centers do
        Select a window $w_i$ of size $a$ centered at $(x_i, y_i)$;
        Clean the window from (possibly) other bubbles: $w_i \leftarrow Autoencoder(w_i)$;
        Run parameter determination procedure on $w_i$: $p_i = (a_i, b_i, \theta_i)$;
        Append $p_i$ to bubble_parameters;
    end for
    return bubble_parameters
```

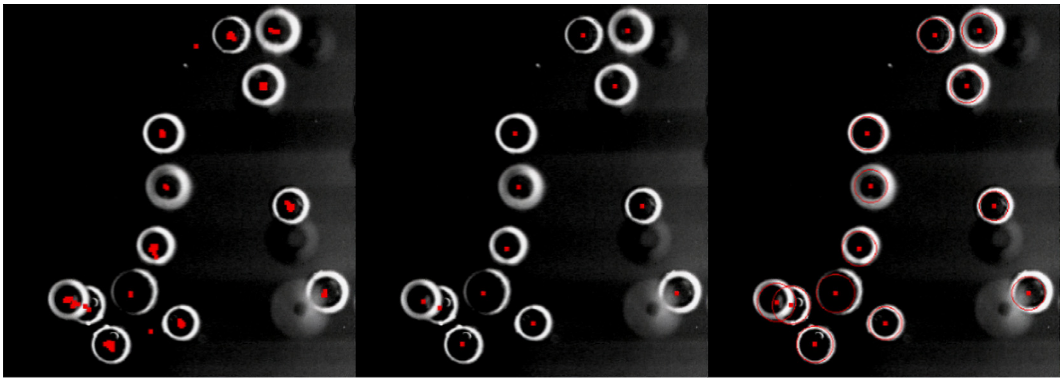

Fig. 8. Image processing algorithm. From left to right outputs for Algorithms 2–4 are presented.

$t_{clf} \approx 0.2$ and $t_{cen} \approx 0.23$, respectively. Such an approach, utilizing two confidence thresholds, allows minimizing the number of false positives in the bubble determination task.

It is important to note that the lower value s increases the total number of processed windows N. The processing time increases as $O(N)$ (see Section 6.2). In addition, experiments with both smaller and higher values of $s \in [5, 50]$ were carried out. We did not observe any significant impact on the bubble center determination precision. Considering the aforementioned factors, we chose $s = 15$ as the trade-off value, using the sliding window with size $a = 50$ pixels.

Furthermore, after processing the entire image, the obtained nearby centers were united, according to their probability weights (values of P defined above), as belonging to the single bubble. The centers of bubble $\mathbf{x}_1 = (x_{11}, y_{12})$ and $\mathbf{x}_2 = (x_{21}, y_{22})$ were considered nearby, if $\|\mathbf{x}_1 - \mathbf{x}_2\|_{L_2} \leq R$, where R is a parameter that depends on the bubbly jet gas content. On average, for the gas content values of $1 - 2\%$, the optimal value - which was revealed in the course of several experiments, is $20\,px$.

All steps above can be grouped into the following two algorithms (Algorithms 2 and 3):

When the obtained centers of the bubbles need to be refined:

At the last step of the algorithm, all image sections with bubbles recognized were cleaned up by the *autoencoder*. Thus, the cleaned bubbles were used for the determination of their approximate sizes (axis a, b) and orientation (angle θ), using the approach described in Section 4.2 (Algorithm 4):

The successive outcomes of all four algorithms are illustrated in Fig. 8.

Additionally, an emphasis should be put on the fact that the basic purpose of using the autoencoder in this study (Algorithm 4) is to simplify the task of determining the geometrical characteristics of the found bubble. Section 6.2 describes how this approach increases the quality metrics.

The described approach using NNs is not “tied” to operate with bubbles of some specific size, they can be different depending on the size of the sliding window. For example, for a window with a size of 80 x 80 pixels, bubbles with radii from 17 to 20 pixels to 29–33 pixels inside the window can be confidently recognized. These bubble sizes are set at the stage of synthetic image generation (see Section 3).

It is also important to note that in order to process the second time-frame image with bubbles (experimental data is a sequence of pairs of two nearby frames), information on the position of the bubbles from the previous frame was used as well as information on the maximum possible displacement of the bubble between the two frames. In this way, bubble centers were obtained as the results of the Algorithm 3 operation, for the first and second frames, correspondingly, were used to refine each other. This has resulted in reducing the number of errors in the construction of the velocity fields and also accelerated the processing of gathering experimental data as a whole.

5. Neural network architecture selection

In this study, we tested about 60 various configurations of CNNs and MLPs in each group to determine which architectures are the most appropriate for their specific tasks. The experimental framework was implemented in Python programming language, using the «TensorFlow» library for symbolic calculations (Abadi et al., 2016).

5.1. Final models

The selection of the final architecture was based on the evaluation of different precision and recall metrics via tests on synthetic data (Fig. 3). Validation of these finally chosen models was carried out on the real experimental data. The GPU was used for all the numerical calculations (see Section 6.2). During the experiments, all models were trained on 24,000 train examples, and the quality metrics were measured on 6000 test examples.

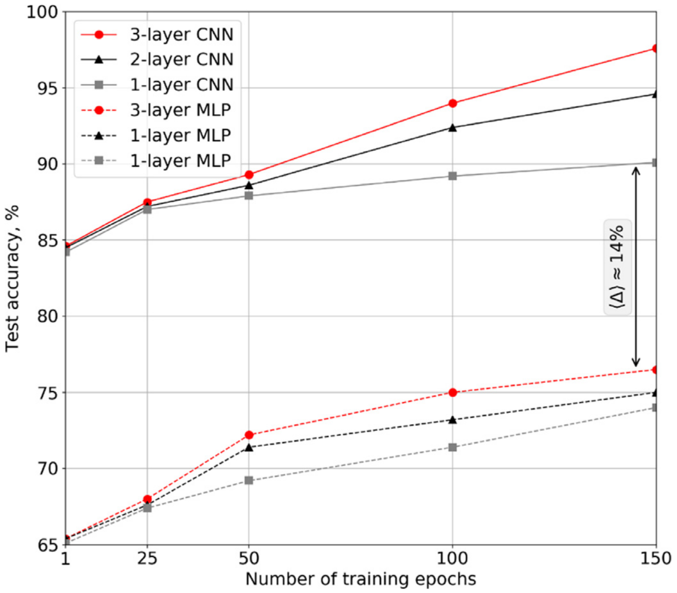

Fig. 9. Selection of the *classifier*. Obtained quality metrics are compared depending on the number of training epochs used. The accuracy difference between CNNs and MLPs is about 13–14% on average.

Table 2
Final results on the synthetic data.

NN	Best obtained results on the test set, %
Classifier	97.6
Centers finder*	94.5

* With precision of 3 pixels.

Table 3
Comparison of optimization algorithms efficiency for CNN_8_16_75 classifier.

Algorithm	Epoch number *	Accuracy,%
Adam	56	96.7
AdaDelta	65	96.1
AdaGrad	85	95.9
SGD+Nesterov	102	95.5
SGD	128	94.1

* Epoch when the accuracy on the test data set exceeds 90%.

Table 4
Optimal training settings.

Architecture	Optimizer	Learning rate	Batch size
Classifier	Adam	0.002	256
Centers finder	Adam	0.001	64
Autoencoder	SGD + momentum	0.01 (0.9)	128

According to these architecture selection experiments (see example in Fig. 9), we found that the most appropriate architectures for the *classifier, centers finder and autoencoder* NNs are: *CNN_8_16_75*, *CNN_8_16_100* and *MLP_100_50_100*, respectively (see Section 2 for the architectures abbreviations). Table 1 shows the summary of their characteristics.

For the correct operation of the bubble parameter determination algorithm (see Section 4.2), it is important that the images of bubbles after cleaning by the autoencoder remain intact and stay in the same place. After several numerical tests on the real experimental images (Fig. 10), the MLP-based architecture of the autoencoder (see Table 1) was selected as the most robust to background noise.

Similarly, after conducting a series of numerical experiments, architectures for the other tasks have been chosen. For example, *CNN_8_16_75* (Fig. 1) is identified as the most effective for the classification task. Table 2 shows the best results for the trained *classifier* and *centers finder* models on the synthetic data.

5.2. Optimizer selection

Besides the architecture, it was also important to choose the most efficient optimization algorithm for each NN to be trained (Sutskever et al., 2013), to speed up the training and obtain better results. During the computational experiments for the loss function optimizers (Haykin, 1999), we tried the following algorithms: *Adam, AdaDelta, AdaGrad* and *SGD* and their various modifications (Goodfellow et al., 2016; Kingma and Ba, 2015). For training the *classifier*, a cross-entropy loss function was used. While for training the *autoencoder* and *centers finder* the NS Kullback–Leibler divergence and mean squared error loss functions were used, respectively.

All convolutional architectures were trained using the “ReLU” activation function for all non-linear transformations. MLPs and autoencoders were trained using the sigmoidal activation function.

For the architecture evaluation, we used the partition of the train data set into two parts - the train and the test set with a ratio of 4:1. Test samples do not occur in the training set.Table 3 below shows the optimization algorithms comparison done on a 1:1 train/test data set partitioning.

We also conducted various similar tests to identify the most effective optimizers for other NN types. The final set of the best hyperparameters is given in Table 4.

To prevent overfitting (Goodfellow et al., 2016), we used a variety of methods, such as batch normalization (Ioffe and Szegedy, 2015) and dropout (Srivastava et al., 2014), for all layers of the networks. Finally, owing to such regularization techniques, it became possible to obtain the absolute additive in the test accuracy of the synthetic images, at about 7 – 9% on average.

6. Approach verification on experimental data

6.1. Experimental conditions and the measurement system

We tested NNs on a free gas-saturated axisymmetric jet flow that was reproduced in a hydrodynamic rig in the Kutateladze Institute of Thermophysics SB RAS. Its description, along with the information on all instrumentation used to control flow parameters is available in Alekseenko et al., 2015) and Pervunin and Timoshevskiy (2016). The temperature of the operating liquid (distilled water) was kept constant at 30 ± 0.1 °C. The bubbly upward jet was formed by a round converging nozzle of 100 mm height with an inner diameter at its outlet D_N = 20 mm. The contraction ratio of the nozzle (i.e., the ratio of its inlet and outlet cross-section ar-

Table 1
Characteristics of the final NN architectures .

Task	Architecture configuration	Input size	Output size	Parameters (x10^3)
Classifier	CNN_8_16_75	2500	1	~175
Centers Finder	CNN_8_16_100	2500	625	~270
Autoencoder	MLP_100_50_100	2500	2500	~584

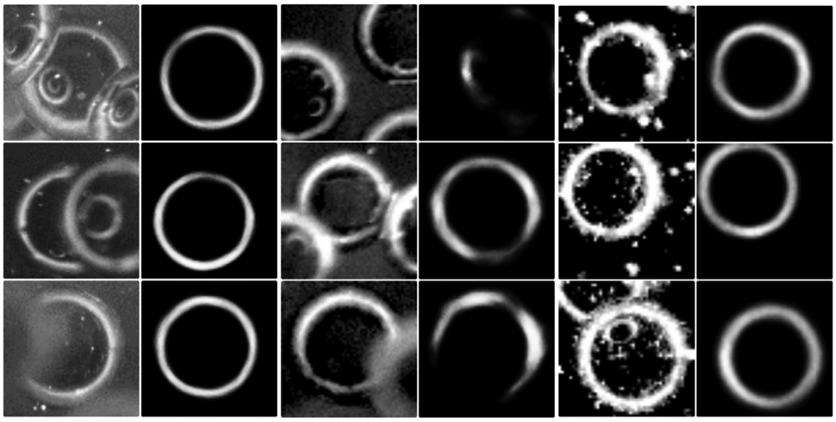

Fig. 10. Example of the experimental images processed by the autoencoder.

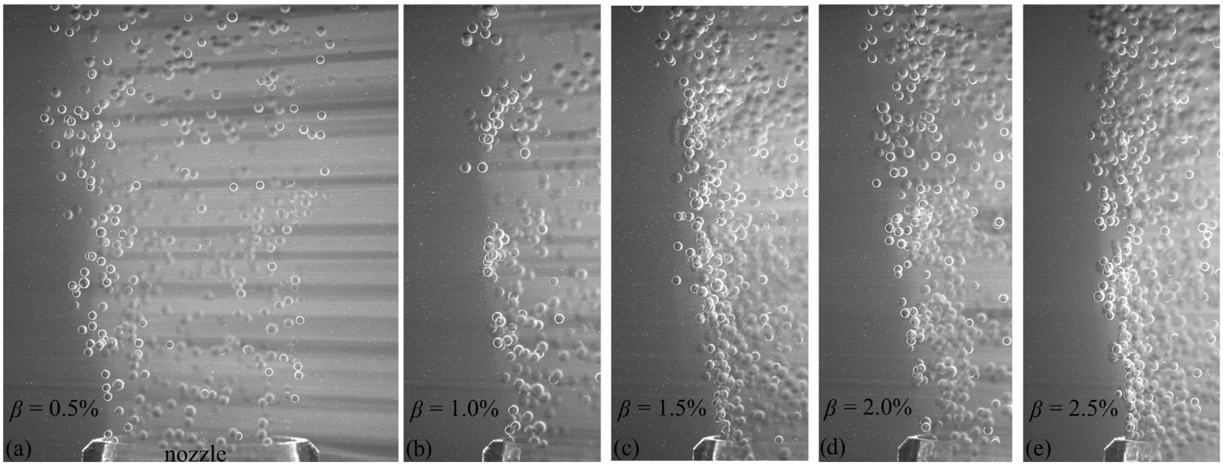

Fig. 11. Examples of original PFBI/PIV images of the free axisymmetric bubbly jet for different air volume fractions β_{exp} = (a) 0.5%, (b) 1.0%, (c) 1.5%, (d) 2% and (e) 2.5%. For the flow regime corresponding to mark (a), we show the whole field of the flow under investigation for clarity, whereas, for the flow conditions with marks (b–e), merely half-images are provided as, hereinafter, only the left-hand sides of these flows are analyzed in detail because of the jet symmetry.

eas) was 9. In order to saturate the flow with air bubbles of close-to-monodisperse size distribution, an air-water mixer of a special construction was utilized (Timkin and Gorelik, 2009).

The length of a stainless-steel pipe, with a 20 mm inner diameter between the mixer outlet and nozzle inlet, was 0.57 m. Different air-water flow-rate ratios (or air volume fractions) $\beta_{exp} = Q_A/(Q_W + Q_A)$, where Q_A and Q_W are volume flow rates of the supplied air and distilled water (measurement error within 2%), were achieved by changing Q_A using a fine-tuning needle valve. The experiments were carried out for β_{exp} = 0, 0.5, 1, 1.5, 2 and 2.5%. The mean diameter of air bubbles D_B was estimated to be approximately 1 mm for all β_{exp}. The Reynolds number based on D_N and the superficial air-water mixture velocity $V_0 = 4 \cdot (Q_W + Q_A)/(\pi \cdot D_N^2) = 0.5$ m/s was equal to $Re = 12,500$ for the one-phase flow (Q_W = 9.32 l/min, Q_A = 0 std. l/min) and 13,000 when β_{exp} = 2.5% (Q_W = 9.32 l/min, Q_A = 0.46 std. l/min). To validate the bubble identification procedure (Section 4), we considered only an initial region of the free jet limited to a distance of the double nozzle diameter ($2D_N$) downstream from the nozzle edge (Fig. 11a).

In order to implement the planar fluorescence for bubbles imaging (PFBI) approach (Akhmetbekov et al., 2010; Dulin et al., 2012) together with a PIV technique, Rhodamine 6G was used as a fluorescent dyeand fluorescent PMMA seeding particles filled with Rhodamine B (fraction 1–20 μm) of Microparticles GmbH production were premixed into the operating liquid. The concentration of Rhodamine 6G was quite low (about 20 μg/l), so the water properties (especially its viscosity and surface tension) could be considered unchanged. A PIV-system with a high spatial resolution consisting of a double-pulse Nd:YAG Quantel EVG00200 laser (wavelength 532 nm, repetition rate 15 Hz, pulse duration 10 ns, pulse energy 200 mJ), an IMPERX IGV-B4820M CCD-camera (digit capacity 16 bits, resolution 4904 × 3280 pixels, acquisition rate 4 Hz) equipped with a Sigma DG Macro 105 mm 1:2.8 lens and a low-pass optical filter (transmission edge at 570 nm) and a POLIS pulse/delay generator for external synchronization were employed to illuminate and register bubbles and tracer particles suspended in the flow simultaneously in the same image.

A laser light sheet was formed by an optical head (divergence angle 7°, focal distance from 0.3 to 1.5 m) containing a cylindrical lens with antireflection coating for the 532 nm wavelength. The thickness of the laser sheet was approximately 1 mm in the region of interest. The distance between the camera matrix and the measurement section that intersected the jet axis of symmetry was 437 mm, and the one from the laser optical head to the jet axis of symmetry was about 410 mm. The optical magnification of the measurement system equaled 0.48. The actual size of registered images was 3960 × 3080 pixels, but the processed image domain was subsequently reduced to 2640 × 3080 pixels, which corresponded to a 40.56 × 47.32 mm area in space (e.g., Fig. 11a), to shorten processing time, discarding those image parts where bubbles were absent. The measurements were performed at a sampling rate of 2 Hz. For each flow regime on β_{exp} (apart from β_{exp} = 0%), 10000 PFBI/PIV image pairs were gathered continu-

Table 5
Average processing time of the experimental image, sec.

Algorithm	$\beta_{exp} = 0.5\%$	$\beta_{exp} = 2.5\%$
NNs	5.2	6.3
Correlation method	38	24

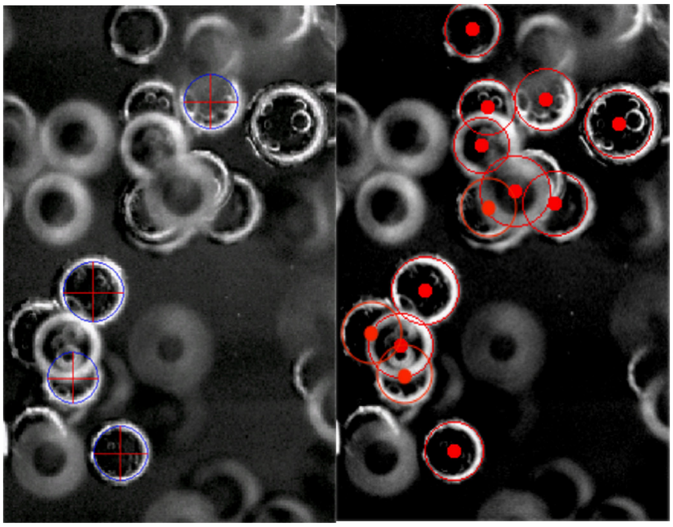

Fig. 12. Illustration of processing a part of the experimental image with the original gas content $\beta_{exp} = 2.5\%$. Left: correlation method. Right: NNs-based approach.

ously during 5000 s. For $\beta_{exp} = 0\%$, half of the series (i.e., 5000 image pairs) was regarded to be sufficient and the total acquisition time was, thereby, reduced to 2500 s.

6.2. Performance comparison of neural networks and the correlation method

For all numerical calculations, the following hardware was used:

- NNs. CPU: AMD Opteron Processor 6168 1.9 GHz; GPU: Nvidia GeForce GTX Titan
- Correlation algorithm. CPU: Intel Core i7-3970X 3.5 GHz

The experiments have shown that the overall increase in efficacy is about 6 to 8 times (Table 5). It is worth noting that, in the single-threaded mode on a single CPU core, the correlation method spends, on average, 30–35 s per image, whereas all three NN joint execution time does not exceed 5 to 6 s while utilizing a GPU.

An example of the joint operation of all three final NN architectures is shown in Fig. 12. Fig. 13 illustrates the average number of the bubbles found in the entire frame by both methods. As the experiment has shown, the correlation algorithm finds, on average, half the bubbles within the entire possible range of bubble sizes.

Moreover, it was determined that on average, the left part of the area of the jet contains 1.2–1.3 times more bubbles than the right one. This stems from the fact that the laser light was scattered during the movement and flowed, losing intensity, making the conditions for identifying bubbles less suitable. Therefore, for further statistics collection, only the information from the left half of the jet was used. For all the data presented below, the image of the jet is mirrored relative to the central vertical axis.

During the calculations, data on the bubble size distribution obtained by both methods for different gas contents were also obtained (Fig. 14).

It was discovered that the average value of the bubble diameter was about 1.01 and 1.12 mm for the NNs and the correlation algorithm, respectively. Both these values are close enough to the 1 mm that was originally specified during the experiment. Therefore, the calculated diameter fluctuated within ± 5–6% for all five possible gas contents used during the experiments.

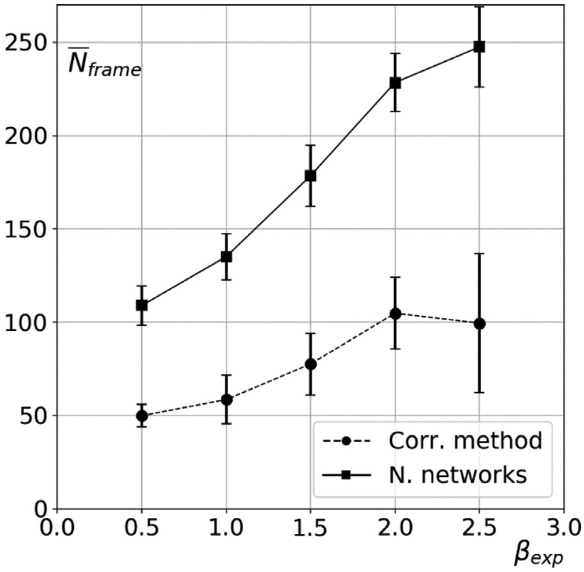

Fig. 13. Dependence of the average number of bubbles in one frame $\underline{N}_{frame}$ for NNs and the correlation method for different values of the experimental gas content. Errors are estimated depending on the approximate number of misclassified/unrecognized bubbles.

Additionally, various gas content profiles and distributions were calculated, using the following approach. Let us assume, that $\Delta V_{gas}[i, j]$ is the bubble volume inside corresponding to the (i, j)th pixel with a fixed area $S = 1\text{px}^2$. H is the laser sheet depth (Fig. 15). Then, the local gas content for every pixel $\alpha[i, j]$, local gas content averaged along the i-th axis $\langle\alpha\rangle_I$, and the average gas content in the three-dimensional region for the I-section $\overline{\langle\alpha\rangle}_I$ can be averaged for N given images and estimated as follows:

$$\begin{aligned} &\alpha[i, j] = \overline{\frac{\Delta V_{gas}[i,j]}{H \cdot S}}_N \; ; \;\; \alpha_i = \sum_X \overline{\frac{\Delta V_{gas}[i,j]}{H \cdot S}}_N \cdot \frac{1}{\max x_i}; \\ &\overline{\alpha}_I = \sum_{x \in I} \sum_{y \in I} \overline{\frac{\Delta V_{gas}[x_i, y_i]}{H \cdot S}}_N \cdot \frac{1}{\max\limits_{x \in I} x \cdot \max\limits_{y \in I} y}; \;\; \beta_{calc} = \; \overline{\alpha}_X \end{aligned} \tag{4}$$

where, β_{calc} is the estimation of the real β_{exp} value, given by averaging$\langle\alpha\rangle$ inside the cylindrical region X above the nozzle. The height and diameter of this cylinder were chosen to be equal to the diameter of the nozzle D_N. In this approach, we assume that all bubbles which participate in the gas content are independent. Thus, it is required to correspondingly process the pixels which were considered more than once (i.e., overlapping volumes). To do so, during the gas content calculation, each pixel accumulated the local gas content, so that ultimately this value was truncated by the laser sheet depth H.

Some examples of the calculations using (4) are shown in Fig. 16.

As it can be seen from these gas content distributions, bubbles in the jet have a feature of forming the clusters on the edges of the nozzle, which later spread upstream. This figure also illustrates the gas content profiles along three different sections. The average gas content $\overline{\langle\alpha_\iota\rangle}_Z$ values are preserved similarly, from section to section (with a certain degree of accuracy ±6%). This confirms the fulfillment of the mass conservation law along the jet.

The gas content was also averaged over the radial direction of the flow. This value $\overline{\langle\alpha\rangle}_r$ is shown in Fig. 17.

We should note that the precision of the gas content estimation depends on the number of analyzed images. Fig. 16 (right) illus-

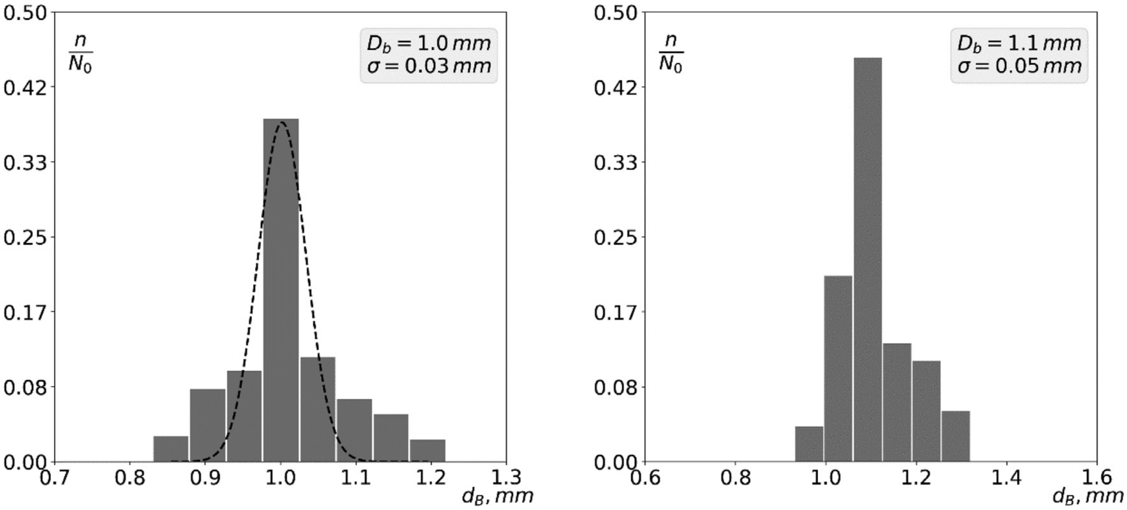

Fig. 14. Distribution of the bubble diameter obtained by processing 500 pairs of the experimental images with $\beta_{exp} = 0.5\%$ by the NNs (left) and the correlation algorithm (right).

trates the comparison of the profiles of $\overline{\langle\alpha\rangle}_r$ for $\beta_{exp} = 2\%$, calculated for 500 and 1250 pairs of images, respectively. Such behavior of NNs indicate that the analyzed images number increase significantly, contributing to the edge effects leveling off that is associated with the PFBI approach peculiarity.

Obtained two-dimensional gas content maps $\bar{\alpha}$ were also used to calculate the average gas content values β_{calc} in the cylindrical region above the nozzle, using (4). These calculations also take into account the bubble size determination inaccuracy (see Section 7). For both the correlation method and NNs, the same depth of the averaging H was used (Fig. 15). This parameter might be assumed as the ratio coefficient that considers the constant portion of those bubbles which were not included in the gas content calculations. The results of these calculations are shown in Fig. 18. According to the obtained data, for $H \approx 3.2$mm, both methods provide the most accurate estimations of the average gas content, β_{calc}. This averaging depth is in good agreement with the fact that there may be recognized overlapping bubbles located at different depths relative to the laser sheet plane (see Fig. 14).

The obtained results are in good agreement with the estimates of the average number of bubbles recognized in the image (see Fig. 13). NNs provide a slight overvalue for most gas content values. This is due to the presence of some number of false positives, that is, some areas of the image were mistakenly regarded as bubbles. Nevertheless, for gas content β_{exp} of two or more percent, the effect of overvaluing is leveled due to too many bubbles overlapping each other, making themselves difficult to distinguish. Moreover, the final calculations presented in Fig. 18 clearly indicate a systematic underestimation of the gas content obtained by the correlation method, in fact, for all experiments that were carried out.

7. Approach constraints

The final algorithm of bubble pattern recognition chosen in this research (see Section 4) does not pretend to be the most accurate approach, because it can be definitely refined. Furthermore, the bubble center recognition precision can be improved by generating more varied and realistic data, for example, having added

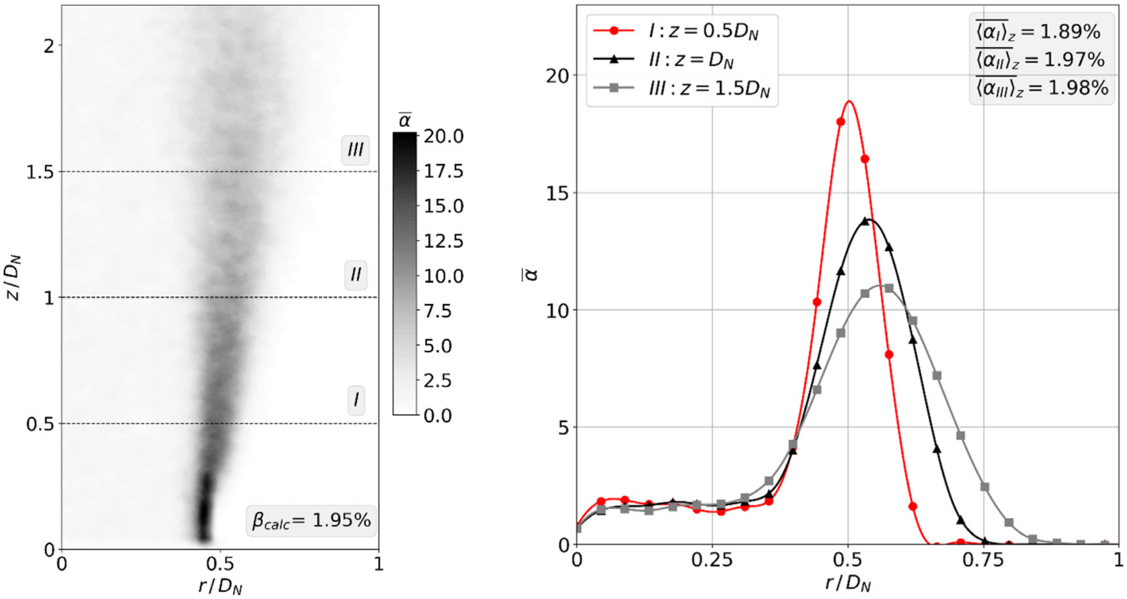

Fig. 16. Two-dimensional map of local relative gas content $\bar{\alpha}$ (left) and profiles of gas content for different sections *I*, *II* and *III* (right). NNs, $\beta_{exp} = 2\%$.

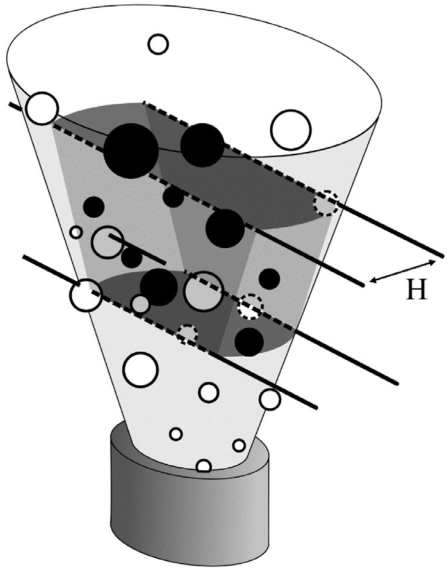

Fig. 15. Illustration of the averaging scheme (4). Bubbles located outside of the laser sheet are not taken into account in the gas content estimation and shown in white, whereas the ones which are inside are in black. Bubbles which cross the laser sheet borders are marked by the dotted boundary.

elliptical arcs as the separate training examples. It is worth emphasizing that all steps of Algorithms 1–4 were chosen after having performed a lot of evaluation experiments. Besides, different other approaches were considered as the alternative options for some of the steps. For instance, to determine bubble sizes, the nonlinear least squares method and its modifications were probed as well as the Hough transform or even CNN.

It is important to note that the key limitation at this moment in the general developed approach is the relative complexity of setting the range of variation of bubble sizes in the images. In order to change this configuration, it is necessary to regenerate the training set with the new parameters and the sizes of the input windows that the speed of the NN operations depend on. However, perhaps the existing approach can be modified to solve this problem, in the sense that the window scanning of the original image (see Algorithm 1) can be automatically scaled, depending on the size of the bubble getting into it. This will eliminate the need to change the training set due to the variation in the sizes of the bubbles.

It is also worth noting that in the future, it is necessary to improve the methodology for calculating the gas content to take into account the contribution of coalescence of the bubbles more accurately. It is required to properly tackle the variety of the bubble curvature participating in coalescence and their relative depth location.

8. Error analysis

In this section we discuss the errors associated with the algorithm of the bubble recognition. Accuracy of the calculated volume of gas fraction depends upon the accuracy of the obtained values of the bubble sizes and the coordinates of their centers. In the present paper the discretization step for the mentioned quantities equals 1.0 pixel. Actually this step can be set arbitrarily small (fraction of the pixel) for the bubbles size. However, the smaller the step, the longer the operation time of the size determination algorithm is. It should be noted that, when counting the volume of gas fraction, the bubbles are considered as spheres, although it is not always the case. Referring to the local gas volume for a spherical bubble from (4), it is calculated according to the following formula: $\Delta V_{gas}[i,j] = 2S\sqrt{r_B^2 - (i-x_0)^2 - (j-y_0)^2}$, where r_B, x_0, y_0 are known with absolute inaccuracy of 1 pixel. It can be shown that the approximate absolute accuracy for $\Delta V_{gas}[i,j]$ is a sum of the absolute accuracies for r_B, x_0 and y_0. The region with minimal relative accuracy is located at the center of the bubble, having approximately 3/40 or 7.5% for our case. The worst relative accuracy for the local gas volume ΔV_{gas} will be at the rim of a bubble, and it can reach 100% or more for the transient or instantaneous gas content distribution. The above reasoning presumes that one has found all the bubbles at the right places, which is unlikely, so one needs to consider the detection accuracy separately while dealing with the average gas content. It is known that during the calculation of the mean quantities, an uncertainty decreases $\sqrt{N}$ times. This means that the gas content relative accuracy at the rim of the bubbles can be decreased to the reasonable values of several percentages, using 10,000 images for the calculation of the mean quantities.

In this study, we assume that the number of coalesced bubbles is very small and, therefore, all overlapping bubbles that are involved in the gas content calculation are separated. This, of course, negatively affects the overall precision of the gas content calculation because some overlapping bubbles are actually involved in the coalescence, although this effect is regarded as neglectable com-

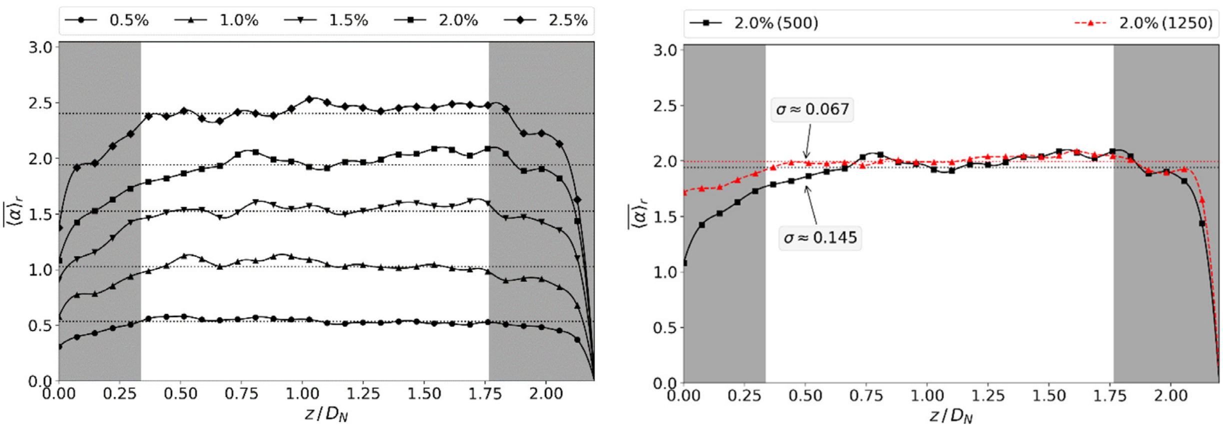

Fig. 17. *Left:* Distributions $\langle\alpha\rangle_r$ for different β_{exp}. Regions shown in gray (from $z/D_N = 0$ to 0.35 and from $z/D_N = 1.8$ to 2.25) were not included in the calculation of β_{calc} due to the edge effects. The dotted lines correspond to the gas distribution values $\overline{\langle\alpha\rangle}_r$ averaged over the region shown in white. *Right:* Distributions $\overline{\langle\alpha\rangle}_r$ for $\beta_{exp} = 2\%$, depending on the number of processed image pairs. RMS σ values for the averaged $\overline{\langle\alpha\rangle}_r$ distributions are shown to illustrate the error decrease with the increase of the image number.

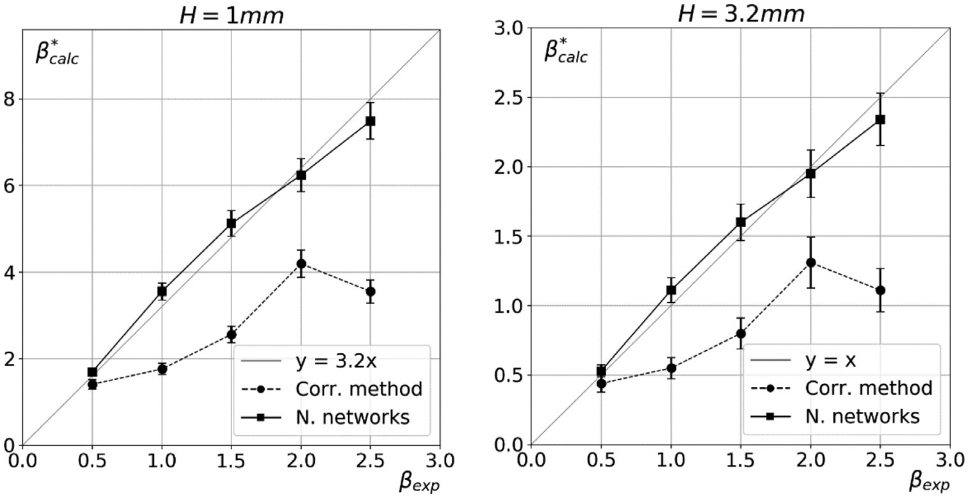

Fig. 18. Comparison of the methods: Values of the calculated average gas content over the nozzle. β^*_{calc} and β_{calc} - calculated average gas content values with different depths of averaging. β_{exp} is the real gas content in the pipe.

pared to other errors given by the inaccuracy of the bubble position and size determination. Thus, in some cases, the gas content can be overestimated, but the probability of such a situation is quite low. This is due to the overall homogeneity of the studied bubbly jets. Specifically, bubbles are typically strongly deformed before their coalescence and neighboring bubbles, at the moment whereupon they coalesce, are not highlighted brightly enough in PFBI images to be recognized by the algorithms.

Another important factor that has a great impact on the gas content estimation uncertainty is the bubble recognition error. A human-evaluated NN-based approach showed that bubble recognition accuracy on the real data drops from 95% to 86% for $\beta_{exp} = 0.5\%$ and $\beta_{exp} = 2.5\%$, respectively. In total, all discussed factors together lead to the $10 - 20\%$ error in the estimation of the actual gas content value β_{calc}, calculated near the nozzle. These errors are depicted in Figs. 13 and 18.

9. Conclusions

This study demonstrates that modern computational techniques, drawn from machine learning, can be successfully applied to the analysis of physical data. The key result of this research is a ready-to-use software package that utilizes a newly developed neural network (NN)-based multi-step algorithm for bubble patterns recognition. Several numerical experiments allowed us to determine optimal NN architectures for performing specific tasks. For instance, it was found that convolutional neural networks (CNNs) are better suited for the problem of classifying bubbles and determining their geometrical centers, while multilayer perceptron (MLP) functions are better for the task of removing noise from images.

The proposed approach was thoroughly tested on real experimental data to evaluate local gas content in a turbulent bubbly jet, with volume air fractions varied from 0% to 2.5%. The new algorithm is shown to allow detecting bubbles with a 20% higher precision for a noticeably shorter period (6–8 times less) compared to the correlation-based algorithm (Akhmetbekov et al., 2010). The processing time of an 8-megapixel image using the developed algorithm is only several seconds on a GPU-equipped standard office PC. Furthermore, the new bubble-detection algorithm is proved to be robust to deviations of the bubble shape from circularity.

Additionally, in this study, we developed an automatic method for the generation of realistic synthetic data based on typical PFBI/PIV images, which was used to train all NNs. This procedure made the data processing pipeline adaptable to various experimental conditions, with a much wider range of possible volume gas fractions. According to the performed tests, the new algorithm detects two times as many bubbles as the bubble mask correlation algorithm in the same images for the considered range of volume gas fractions from 0% to 2.5%.

The errors introduced by various components of the developed algorithm were also analyzed. The estimation uncertainty of integral volume gas fractions in the near-to-nozzle region and close to the edge of the measurement area is considered to be equal to 10–20%. In order to enhance the volume gas fraction estimation, more sophisticated algorithms to identify highly overlapping and partially illuminated bubble patterns should be employed. Finally, this study opens new opportunities for further improving bubble detection methods using more advanced architectures of NNs. This will facilitate an analysis of complex gas-liquid two-phase flows, making it possible to find ways to control momentum and mass transfer.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the research reported in this paper.

CRediT authorship contribution statement

Igor Poletaev: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing, Visualization, Formal analysis. **Mikhail P. Tokarev:** Methodology, Validation, Writing - review & editing, Formal analysis. **Konstantin S. Pervunin:** Methodology, Validation, Writing - review & editing, Resources, Funding acquisition, Investigation, Formal analysis.

Acknowledgments

The novel approach based on NNs for bubble recognition has been developed with funding from the Russian Science Foundation (Project No. 19-79-30075). The bubbly jet experiment was carried out using PFBI technique under the state contract with IT SB RAS (AAAA-A19-119052190039-8). The authors also appreciate the valuable assistance of PhD student Mikhail Nichik in performing the experiment.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. USENIX Association, pp. 265–283.

Akhmetbekov, Y.K., Alekseenko, S.V., Dulin, V.M., Markovich, D.M., Pervunin, K.S., 2010. Planar fluorescence for round bubble imaging and its application for the study of an axisymmetric two-phase jet. Experiments in Fluids 48 (4), 615–629. doi:10.1007/s00348-009-0797-0.

Alekseenko, S.V., Dulin, V.M., Markovich, D.M., Pervunin, K.S., 2015. Experimental investigation of turbulence modification in bubbly axisymmetric jets. Journal of Engineering Thermophysics 24 (2), 101–112 doi:10.1134/S1810232815020010.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L.V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., Zhu, Z., 2016. Deep speech 2: end-to-end speech recognition in English and Mandarin, In: Proceedings of the 33rd International Conference on Machine Learning. In: Proceedings of Machine Learning Research, 48. JMLR Inc., pp. 173–182. doi:10.1145/1143844.1143891 JMLR W&CP.

Baldi, P., 2012. Autoencoders, unsupervised learning, and deep architectures. Proceedings of ICML Workshop on Unsupervised and Transfer Learning. Proceedings of Machine Learning Research, 27. JMLR Inc., pp. 37–50. doi:10.1561/2200000006.

Bongiovanni, C., Chevaillier, J.P., Fabre, J., 1997. Sizing of bubbles by incoherent imaging: defocus bias. Experiments in Fluids 23 (3), 209–216. doi:10.1007/s003480050104.

Bröder, D., Sommerfeld, M., 2003. Combined PIV/PTV-measurements for the analysis of bubble interactions and coalescence in a turbulent flow. The Canadian Journal of Chemical Engineering 81 (3–4), 756–763. doi:10.1002/cjce.5450810356.

Caupin, F., Herbert, E., 2006. Cavitation in water: a review. Comptes Rendus Physique 7 (9–10), 1000–1017. doi:10.1016/j.crhy.2006.10.015.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (Eds.), Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1. IEEE Computer Society, pp. 886–893. doi:10.1109/CVPR.2005.177.

Davies, E.R., 1989. Finding ellipses using the generalised Hough transform. Pattern Recognition Letters 9 (2), 87–96. doi:10.1016/0167-8655(89)90041-X.

Dulin, V.M., Markovich, D.M., Pervunin, K.S., 2012. The optical principles of PFBI approach. AIP Conference Proceedings 1428 (1), 217–224. doi:10.1063/1.3694709.

Fu, Y., Liu, Y., 2019. BubGAN: bubble generative adversarial networks for synthesizing realistic bubbly flow images. Chemical Engineering Science 204, 35–47. doi:10.1016/j.ces.2019.04.004.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 580–587. doi:10.1109/CVPR.2014.81.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. In: Deep Learning. In: Adaptive Computation and Machine Learning. MIT Press.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, 27. Curran Associates Inc., pp. 2672–2680.

Haykin, S., 1999. Neural Networks-A Comprehensive Foundation. In: Horton, M., Bauer, H.-U., Der, R. (Eds.), Neural networks: a comprehensive foundation, 2nd. Prentice Hall Inc doi:10.1017/S0269888998214044.

Ho, C.-T., Chen, L.-H., 1995. A fast ellipse/circle detector using geometric symmetry. Pattern Recognition 28 (1), 117–124. doi:10.1016/0031-3203(94)00077-Y.

Huang, H., Yu, P.S., Wang, C., 2018. An introduction to image synthesis with generative adversarial nets. arXiv:1803.04469v2 [cs.CV].

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167v3 [cs.LG].

Ilonen, J., Juránek, R., Eerola, T., Lensu, L., Dubská, M., Zemčík, P., Kälviäinen, H., 2018. Comparison of bubble detectors and size distribution estimators. Pattern Recognition Letters 101, 60–66. doi:10.1016/j.patrec.2017.11.014.

Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., Dean, J., 2017. Google's multilingual neural machine translation system: enabling zero-shot translation. Transactions of the Association for Computational Linguistics 5, 339–351. doi:10.1162/tacl_a_00065.

Karpathy, A., Fei-Fei, L., 2017. Deep visual-semantic alignments for generating image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4), 664–676. doi:10.1109/TPAMI.2016.2598339.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, 25. Curran Associates Inc., pp. 1097–1105.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In: Touretzky, D.S. (Ed.), Advances in Neural Information Processing Systems, 2. Morgan Kaufmann Publishers, pp. 396–404.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86 (11), 2278–2324 doi:10.1109/5.726791.

LeCun, Y., Huang, F.J., Bottou, L., 2004. Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2. IEEE Computer Society, pp. 97–104 II-97–104.

LeCun, Y., Kavukcuoglu, K., Farabet, C., 2010. Convolutional networks and applications in vision. In: ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems. IEEE Publishing, pp. 253–256. doi:10.1109/ISCAS.2010.5537907.

Lindken, R., Merzkirch, W., 2002. A novel PIV technique for measurements in multiphase flows and its application to two-phase bubbly flows. Experiments in Fluids 33 (6), 814–825. doi:10.1007/s00348-002-0500-1.

Liu, Z.Q., Qi, F.S., Li, B.K., Cheung, S.C.P., 2016. Modeling of bubble behaviors and size distribution in a slab continuous casting mold. Int J Multiphas Flow 79, 190–201. doi:10.1016/j.ijmultiphaseflow.2015.07.009.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60, 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Ohmi, K., 2008. SOM-Based particle matching algorithm for 3D particle tracking velocimetry. Applied Mathematics and Computation 205 (2), 890–898. doi:10.1016/j.amc.2008.05.101.

Orbach, J., 1962. Principles of neurodynamics. Perceptrons and the theory of brain mechanisms.. Archives of General Psychiatry 7 (3), 218–219. doi:10.1001/archpsyc.1962.01720030064010.

Pervunin, K.S., Timoshevskiy, M.V., 2016. Bubbly free and impinging jets: experimental study by means of PIV and PFBI. Journal of Physics: Conference Series 754, 032016. doi:10.1088/1742-6596/754/3/032016.

Sapkota, A., Ohmi, K., 2009. A neural network based algorithm for particle pairing problem of PIV measurements. IEICE Transactions on Information and Systems E92–D (2), 319–326. doi:10.1587/transinf.E92.D.319.

Sapkota, A., Ohmi, K., 2008. Neural network approach to stereoscopic correspondence of three-dimensional particle tracking velocimetry. IEEJ Transactions on Electrical and Electronic Engineering 3 (6), 612–619. doi:10.1002/tee.20322.

Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4), 640–651. doi:10.1109/TPAMI.2016.2572683.

Smith, L.I., 2002. A tutorial on principal components analysis (Computer Science Technical Report No. OUCS-2002-12). Otago University Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15 (1), 1929–1958 doi:10.1214/12-AOS1000.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In: Dasgupta, S., McAllester, D. (Eds.), Proceedings of the 30th International Conference on Machine Learning, 28. JMLR Inc., pp. 1139–1147, Jmlr W&Cp. doi:10.1109/ICASSP.2013.6639346.

Szegedy, C., Toshev, A., Erhan, D., 2013. Deep neural networks for object detection. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, 26. Curran Associates Inc., pp. 2553–2561.

Timkin, L.S., Gorelik, R.S., 2009. The problems of the upward monodispersed microbubble flow in a vertical tube. Journal of Engineering Thermophysics 18 (1), 69–71. doi:10.1134/S1810232809010093.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. Association for Computing Machinery, pp. 1096–1103.

Xie, Y., Ji, Q., 2002. A new efficient ellipse detection method. Object recognition supported by user interaction for service robots, 2. IEEE Publishing, pp. 957–960.

Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. arXiv:1412.6980v9 [cs.LG].