A supervised algorithm entirely trained on a synthetic dataset to segment granular suspension images

David Paulovics
CNRS UCA

Bruno Figliuzzi (bruno.figliuzzi@minesparis.psl.eu)
Mines Paris - PSL University

Théo Dumont
Mines Paris - PSL University

Frédéric Blanc
CNRS UCA

Research Article

Keywords: Convolutional neural network, Image segmentation, Morphological models, Local rheometry, Granular suspension

Posted Date: February 23rd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2598116/v1

Additional Declarations: No competing interests reported.

A supervised algorithm entirely trained on a synthetic dataset to segment granular suspension images

David Paulovics[1], Bruno Figliuzzi[2*†], Théo Dumont[2†] and Frédéric Blanc[1†]

[1]Institut de Physique de Nice, CNRS UCA, Parc Valrose, Nice, 06108 CEDEX 2, France.
[2]Center for Mathematical Morphology, Mines Paris - PSL University, 35, rue Saint Honoré, Fontainebleau, 77305, France.

*Corresponding author(s). E-mail(s): bruno.figliuzzi@minesparis.psl.eu;
Contributing authors: david.paulovics@univ-cotedazur.fr; dumont.theo@protonmail.com; frederic.blanc@univ-cotedazur.fr;
†DP, BF and TD developed and programmed the segmentation algorithms. FB carried out the rheology experiments. The authors participated equally in the writing of the article.

Abstract

We present in this article a novel methodology for segmenting experimental images of granular suspensions, which uses a convolutional neural network trained on synthetic images generated with a morphological model. In many image processing problems related to physical applications, the lack of annotated data prevents the use of state-of-the-art supervised algorithms. Our solution to overcome this issue is to alleviate the need for annotated images by using a generative morphological model to construct synthetic images subsequently used as training samples. When applied to actual images of a suspension, the convolutional neural network presents good generalization properties and surpasses the performances of traditional segmentation algorithms. This gain in accuracy is crucial to improve the estimation of the local concentration field in the suspension.

Keywords: Convolutional neural network · Image segmentation · Morphological models · Local rheometry · Granular suspension

1 Introduction

Image processing techniques are crucial to interpret the results of local rheometry experiments of non-Brownian suspensions. Two quantities are of particular importance in the characterization of the suspension properties: the viscosity and the concentration fields. Both are classically accessed by recording images of the suspension particles during the flow at regular time intervals. The principle of a recording device is schematized in Figure 1. A very thin flat laser sheet illuminates a transparent suspension and excites the fluorescence of a dye dissolved in the liquid. The fluorescent light is then collected by a camera perpendicular to the laser plane. Figure 2 shows an image obtained with the device, where the spherical particles of the suspension appear as black disks.

The measurement of the concentration field relies upon the detection and segmentation of the particles present in the image (D'Ambrosio et al., 2021; Snook et al., 2016). Many types of algorithms were developed to perform these particular

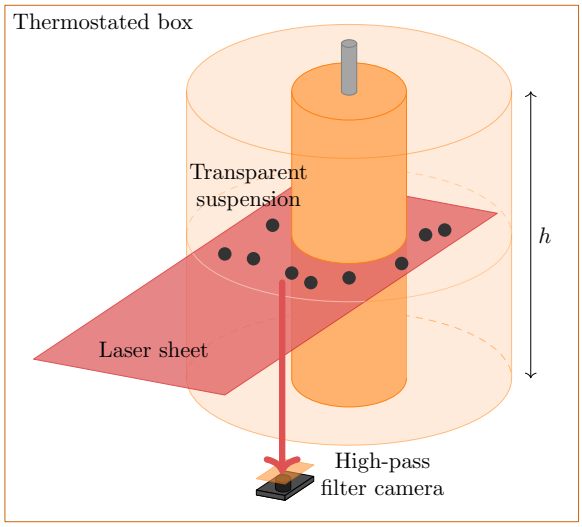


Fig. 1: Schematic view of a recording device (Blanc et al., 2013).

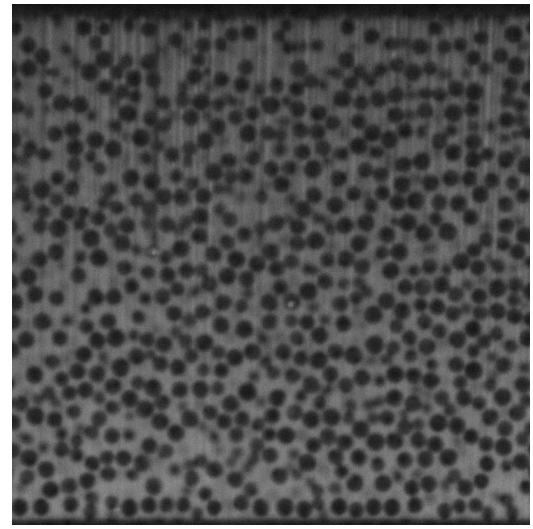


Fig. 2: Experimental image of a granular suspension from D'Ambrosio et al. (2021). The boundaries of the flow cell containing the suspension are visible at the top and at the bottom of the image.

tasks. To mention but a few, Crocker and Grier (1996) rely on centroid detection techniques of the intensity map, Brujić et al. (2003) use a deconvolution technique to infer 3D positions of spherical particles, Bierbaum et al. (2017) fit the image to the convolution of a physically accurate model of the sample and the point spread function of the optical set-up, whereas Blanc et al. (2013); D'Ambrosio et al. (2021); Dijksman et al. (2017); Dougherty (1992); Kimme et al. (1975); Snook et al. (2016) rely on morphological transformation tools. All these algorithms are subtle to implement due to the variability of the lighting conditions inherent to transparent suspensions, which makes it necessary to systematically readjust their parameters.

To overcome these difficulties and to be able to detect particles in a completely automated way, we developed an image processing algorithm based on a convolutional network (Chen et al., 2017). This approach brings an advantage over traditional image processing techniques by alleviating the need of updating the algorithm parameterization for each novel image. However, like all supervised learning algorithms, convolutional networks require a dataset of annotated experimental images to be trained, referred to as *training dataset* or *ground truth* in the following. To construct the images of the ground truth, it is necessary to identify manually the objects that have to be detected, here the positions and sizes of the particles. The annotation is a time consuming task, which renders the use of supervised algorithms difficult in a lot of problems related to physical applications. In addition, the annotation itself is error prone due to the inevitable loss of attention of the operator in charge of annotating the images. The main originality of our approach is that we entirely trained the network on a series of synthetic images generated with morphological models (Figliuzzi, 2019; Figliuzzi et al., 2021; Jeulin, 2021; Stoyan et al., 2013), commonly used to simulate micro-structures in materials engineering (Bortolussi et al., 2018; Figliuzzi et al., 2016), rather than on images of real experiments. With our approach of artificial synthesis of the ground truth, the positions and sizes of the particles are known by construction, which allows us to get rid of the difficulty of obtaining a reliable learning database[1].

The lack of annotated data has long been identified as a critical issue that prevents the use of state-of-the-art supervised algorithms in many image processing problems. In particular, annotating images obtained during physical experiments

[1]Code available in the following GitHub repository: https://github.com/bruno-figliuzzi/blob_net

is often expensive, which triggers interest in alternative methods where ground truth images are generated in a synthetic manner. The development of such methods is increasingly being studied in the literature (Jahanian et al., 2021; Nagy et al., 2022; Ravuri and Vinyals, 2019). In Besnier et al. (2020), a generative adversarial network is for instance used to generate a dataset of images similar to those of ImageNet. These generated images are then used to train a classification network. In Baradad et al. (2021), the authors investigate image generation models that produce images from simple random processes. These generated images are subsequently used as training data for a visual representation learner.

The outline of the article is as follows. In Section 2, we describe the neural network used to perform the segmentation task and the morphological model employed to generate synthetic training images. In Section 3, we evaluate the results of the algorithm on actual experimental images and compare them to classical segmentation algorithms. Conclusions are finally drawn in Section 4.

2 Segmentation algorithm

2.1 Network architecture

To perform the image segmentation, we rely upon the Context Aggregation Network (CAN) introduced in Yu and Koltun (2016). This network is entirely composed of convolutional layers, making it adaptable to any size of input image. Its main particularity is that it gradually aggregates contextual information without losing resolution through the use of dilated convolutions whose field of view increases exponentially over the successive network layers. This exponential growth grants global information aggregation with a very compact structure (Chen et al., 2017; Yu and Koltun, 2016). The two main reasons justifying the use of this network are therefore its adaptability to the size of the experimental images to be processed and its relatively low number of parameters, which makes it easier to train than more complex convolutional networks including for instance U-Net (Ronneberger et al., 2015).

In the CAN architecture, the input images travel through a set of layers $\{L^{(s)}\}_{1\leq s\leq \ell}$. We modify the output of the original network so that it is composed of an image with two channels corresponding to a segmentation mask M for the granular suspension particles and of an image C used to locate the centers of the particles, respectively.

We present in Tab. 1 the detailed architecture of the network. Each block $L^{(s)}$ for $s \in [\![2, \ell-2]\!]$ is made of a 3×3 *dilated convolution* with kernel $K^{(s)}$ and dilation parameter $r^{(s)} = 2^{s-1}$, followed by an *adaptive batch normalization* layer $\Psi^{(s)}$ (Chen et al., 2017) and a *leaky rectifier linear unit* (leaky ReLU) non-linear activation function Φ. The depth d of all hidden convolutional layers is kept fixed in the CAN architecture. We consider in particular architectures with $d = 24$ or 32, as discussed in Section 2.3. The output of an intermediate layer $L^{(s)} = (L_i^{(s)})_{1\leq i\leq d}$ at a spatial location represented by the 2-component coordinate vector $\mathbf{p}$ can be computed from the output of the previous layer $L^{(s-1)}$ as follows:

$$L_i^{(s)}[\mathbf{p}] = \Phi\Big(\Psi^{(s)}\Big(b_i^{(s)} + \sum_{j=1}^{d} L_j^{(s-1)} *_{r^{(s)}} K_{i,j}^{(s)}[\mathbf{p}]\Big)\Big)\,, \tag{1}$$

where $K_{i,j}^{(s)}$ and $b_i^{(s)}$ are the convolution kernels and the bias associated with the i-th channel of layer $L^{(s)}$. The dilated convolution operator $*_r$ is defined by

$$(L *_r K)[\mathbf{p}] = \sum_{\mathbf{u}+r\mathbf{v}=\mathbf{p}} L[\mathbf{u}]K[\mathbf{v}]\,, \tag{2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are 2-component coordinate vectors and $L[\mathbf{u}]$ and $K[\mathbf{v}]$ denote the values of tensor L and convolution kernel K at locations $\mathbf{u}$ and $\mathbf{v}$, respectively. To handle border effects, we use zero-padding. Hence, the spatial dimensions of the latent variables inside the network are identical to the spatial dimension of the input image. The *adaptive normalization function* $\Psi^{(s)}$ is defined by

$$\Psi^{(s)}(x) = \alpha^{(s)}x + \beta^{(s)}\,\mathrm{BN}(x)\,, \tag{3}$$

with BN the usual batch normalization (Ioffe and Szegedy, 2015):

$$\mathrm{BN}(x) = \gamma^{(s)}\frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} + \delta^{(s)}\,, \tag{4}$$

where the operations are componentwise. As such, $\Psi^{(s)}$ combines identity mapping and batch normalization. As for the weights and the biases of the convolution kernels, the parameters $\alpha^{(s)}$, $\beta^{(s)}$, $\gamma^{(s)}$ and $\delta^{(s)}$ are learned through gradient descent and backpropagation. They allow the model to choose between giving importance to the identity term or to the normalization term. Because it keeps the weights close to 0, batch normalization provides a way to regularize the network such that it generalizes more easily (Ioffe and Szegedy, 2015). This alleviates the need to use dropout or other regularization techniques to mitigate overfitting. Finally, the leaky ReLU activation function is defined by

$$\Phi(x) = \begin{cases} x & \text{if } x > 0 \\ 0.2x & \text{otherwise.} \end{cases} \tag{5}$$

As described in Tab. 1, the penultimate layer of the network is a classic convolution layer with a filter with size 3×3. The final layer is a 1×1 convolution used to perform dimension reduction. The neural network produces a segmentation mask M and an image C with bi-dimensional Gaussian functions placed at locations corresponding to the centers of the detected particles. We obtain a labeled image of the detected particles by applying a watershed algorithm (Figliuzzi et al., 2017; Vincent and Soille, 1991) to the segmentation mask M, previously thresholded at the value $1/2$, with the local maxima of C selected as markers.

2.2 Generation of synthetic training images

As discussed in the introduction, a major difficulty associated with the use of convolutional networks is that these architectures require a significant amount of annotated data to be trained. The manual annotation is a time-consuming task, especially when the number of particles to annotate in each image is in the order of thousands.

To train the neural network, we rely on a dataset of synthetic images constructed with a morphological model. The use of synthetic images enables us to obtain training images along with a ground truth without the need of annotating a subset of the experimental images manually. The difficulty associated with the use of synthetic images is that these images must be highly similar to the experimental images to ensure good generalization properties of the trained neural network architecture.

Our approach consists in generating gray level images encoded on 8 bits through the use of random morphological models. The image generation proceeds in several subsequent steps:

- **Step 1.** We start by specifying the dimension $w\times h$ of the synthetic image (4000×1000 pixels) and we build a mask specifying the location of the wall of the flow cell at the image borders. We assign distinct gray levels to the mask and to the interior to obtain an intensity image denoted $\bar{I}$.
- **Step 2.** A characteristic feature of the experimental images is that they exhibit quasi-periodic stripes patterns. To simulate these patterns, we perturb the intensity at each pixel location $[x, y]$ in the image according to the relationship:

$$\hat{I}_1[x, y] = \bar{I} + \sum_{i=1}^{2} A_i \cos(2\pi f_i \phi(x, y)) \,. \tag{6}$$

  In this equation, the amplitudes A_1 and A_2 and the frequencies f_1 and f_2 are specified randomly for each generated image from uniform distributions on the intervals 7–10 and 0.025–0.5, respectively. The quantities $\phi(x, y)$ defined at each location are independent random variables drawn from a normal distribution with mean x and standard deviation σ. They are used instead of the coordinate x in order to add randomness to the geometry of the patterns. Note that all spatial distances are indicated in pixels in our presentation, but can be adapted at the adequate size to reproduce the experimental images.
- **Step 3.** We use a Boolean model of disks to simulate a mask for the particles. The Boolean model is a grain model obtained by implanting independent random primary grains G' on the germs $\{x_k\}$ of a Poisson point process with intensity θ. The intensity of the Boolean model corresponds to the average number of implanted points per unit area. The resulting set $\mathcal{G}$ is

$$\mathcal{G} = \bigcup_{x_k \in \mathcal{P}} G'_{x_k} \,, \tag{7}$$

Layer $L^{(s)}$ for $s =$		1	2	3	4	5	6	7
Input channels		3	24	24	24	24	24	24
Output channels		24	24	24	24	24	24	2
Conv.	kernel size	3×3	3×3	3×3	3×3	3×3	3×3	1×1
	dilation $r^{(s)}$	1	2	4	8	16	1	1
	padding	1	2	4	8	16	1	0
Adaptive BN		✓	✓	✓	✓	✓	✓	✓
Leaky ReLU		✓	✓	✓	✓	✓	✓	—
Number of parameters		722	5258	5258	5258	5258	5258	50

Table 1: Architecture of the Context Aggregation Network (CAN). The total number of trainable parameters for this architecture is 27162.

where G'_{x_k} denotes the translated of the primary grain G' at point x_k.
In general, the grains of a Boolean model can overlap. To avoid this, we add the grains of the Boolean model sequentially. When a grain intersects a grain which is already present, we simply remove it from the simulation. The primary grains that we use to construct the model are random disks whose radii are drawn according to a normal distribution with mean $\bar{R}$ and standard deviation σ_R specified for each image. In practice, to ensure the obtaining of a variety of geometrical settings, $\bar{R}$ is drawn from an uniform distribution on the interval 9–15 pixels and σ_R from an uniform distribution on 1–2 pixels for each generated image. A gray level is finally selected independently for each particle according to an uniform law on the interval 20–35. The gray level background is set equal to 255. This results in the obtaining of a particle image $\hat{P}$. The synthetic image is updated by taking the minimum value between the background image $\hat{I}_1$ and the particles image $\hat{P}$:

$$\hat{I}_2[x, y] = \min\{\hat{I}_1[x, y], \hat{P}[x, y]\} . \tag{8}$$

The particle image $\hat{P}$ is used to generate a binary mask image $\hat{M}$ indicating the presence of the particles in the generated image. In addition, we create an image $\hat{C}$ recording the centers $(x_i, y_i)_{1 \leq i \leq N}$ of the N implanted suspension particles by setting:

$$\hat{C}[x, y] = \sum_{i=1}^{N} \frac{1}{2\pi s^2} \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2s^2}\right), \tag{9}$$

where $s = 5$ pixels. In this image, each particle is identified by a normalized bi-dimensional Gaussian function. $\hat{M}$ and $\hat{C}$ constitute the ground truth images associated with the synthetic image.

- **Step 4.** To complete the image generation, we add blur to the synthetic image by convolving it with a Gaussian kernel G with standard deviation set equal to 3 pixels, as well as white noise. The synthetic image is therefore described by:

$$\hat{I}[x, y] = \max\{0, (\hat{I}_2 * G)[x, y] + W[x, y]\} , \tag{10}$$

where the quantities $\{W[x, y]\}_{1 \leq x \leq w,\, 1 \leq y \leq h}$ are independent centered Gaussian random variables with standard deviation 8.

We display in Fig. 3 a synthetic image of the suspension constructed with the aforementioned procedure. We remark that synthetic images are visually very close to the suspension images obtained in the experiments.

2.3 Training of the neural network

To train the neural network architecture, we generate a training set and a validation set containing respectively 2240 and 360 synthetic images along with their corresponding ground truth images. We use the Euclidean distance between the output of the network and the ground truth images as loss function to train the algorithm, therefore formulating the segmentation as a regression problem. We rely on data augmentation techniques (Shorten and Khoshgoftaar, 2019) to improve the robustness of the network : the network is fed with random crops of the training

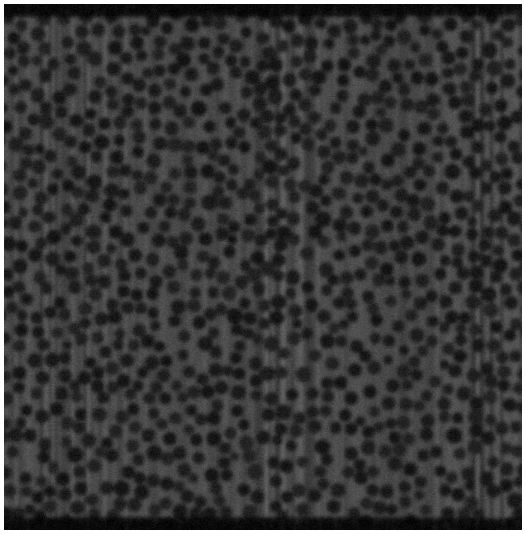


Fig. 3: Synthetic image of the suspension constructed with our procedure.

images with randomly distorted gray level histogram. To train the neural network, we use the Adam optimizer with a learning rate initially set to 0.1 and a batch size of 8, and we divide the learning rate by a factor of 2 every 50 epochs. We fix the maximal number of epochs to 400, and retain the weights of the epoch that lead to the minimal error on the validation set. The training process is schematized in Fig. 4. In this figure, we present the training and validation loss with respect to the training epochs. We also present the output images produced by the neural network architecture at different epochs. The local maxima of the output C of the network, which provides a map marking the locations of the centers of the detected particles by a bi-dimensional Gaussian function, are superimposed on the segmentation mask M produced by the network.

During training, we use the validation set to select hyper-parameters related to the neural network architecture. In particular, we train several versions of the network architecture with distinct numbers ℓ of layers and with distinct depths d (24 and 36) for the inner layers. Based upon the validation loss, we noticed that increasing the number of layers from $\ell = 6$ to $\ell = 7$ leads to a significant increase in performance. However, the validation losses obtained with $\ell = 7$ and $\ell = 8$ were almost similar, so we decided to set the value of ℓ equal to 7. Similarly, the performance gain associated with an increased depth for the inner layers being negligible, we fixed the depth d to 24 in the final architecture.

3 Results and discussion

3.1 Evaluation dataset

To allow for a quantitative evaluation of the results and to study the generalization of the algorithm trained on synthesized images to actual experimental images, we manually annotated 5 images obtained experimentally by labeling all the suspension particles by a disk. Each particle becomes therefore characterized by the center and the radius of the disk. Although 5 represents a relatively small amount of images, these images are very large and therefore contain a significant number of particles, as shown in Tab. 2. Hence, the detection results are tested against a large number of particles, which ensures their statistical validity. In addition, we specifically selected in the experimental dataset a poor quality image where a significant illumination gradient occurs. This image, displayed in Fig. 5, is also particularly blurry. On average, each image contains 1037 particles. The time required for properly annotating an image manually is therefore on the order of one to two hours. Again, this illustrates the fact that annotating manually a whole set of images for training a convolutional network architecture is out of reach in many applications.

3.2 Overview of classical segmentation algorithms

We compare the results of the convolutional network architecture trained on synthetic data to results obtained with traditional algorithms, including Otsu thresholding, adaptive thresholding and K-means segmentation. We provide a brief description of these algorithms below.

K-means

The K-means segmentation algorithm (Bishop, 2006) performs the segmentation of the suspension images by applying a K-means algorithm with $K = 2$ classes to the gray levels of the image. The darkest cluster identified by the K-means algorithm is retained as a mask for the suspension

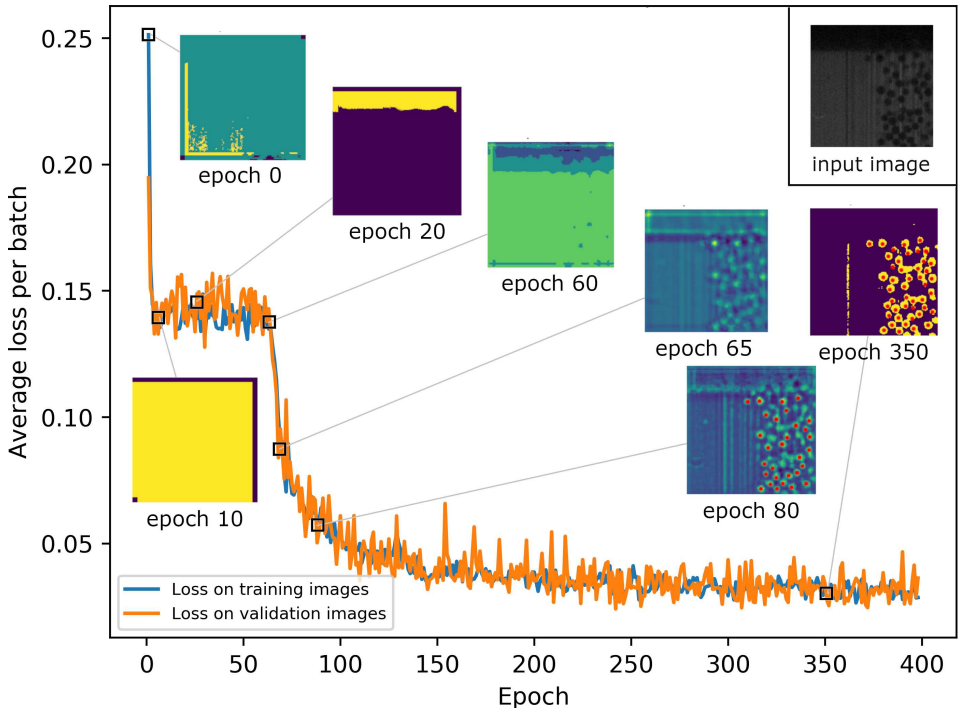


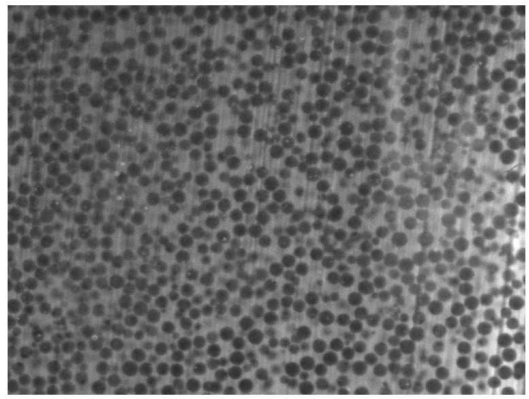
Fig. 4: Schematic view of the training process. Output images produced by the neural network architecture are displayed for different epochs. In these images, the local maxima of the output C of the network, which provides a map marking the locations of the centers of the detected particles by a bi-dimensional Gaussian function, are superimposed on the segmentation mask produced by the network.

		Context Aggregation Network				**K-means**			
Image	Particles	Recall	Precision	D [px]	IoU	Recall	Precision	D [px]	IoU
#1	1339	0.949	0.978	0.68	0.784	0.827	0.954	1.25	0.626
#2	1329	0.944	0.977	0.67	0.771	0.839	0.963	1.46	0.581
#3	964	0.926	0.983	0.72	0.752	0.817	0.962	2.05	0.487
#4	1051	0.947	0.996	0.55	0.869	0.808	0.948	1.75	0.666
#5	487	0.961	0.998	0.5	0.883	0.879	0.949	1.49	0.671
Average	1034	0.945	0.986	0.62	0.812	0.834	0.955	1.6	0.606

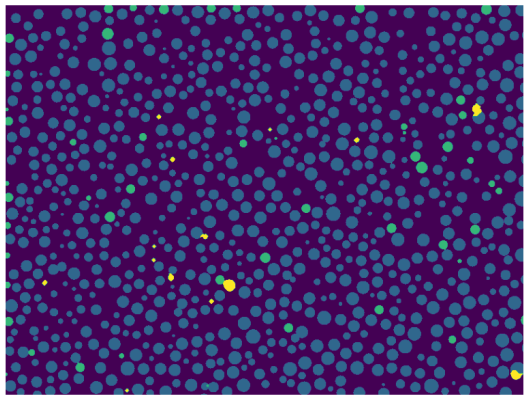
		Otsu thresholding				**Adaptive thresholding**			
Image	Particles	Recall	Precision	D [px]	IoU	Recall	Precision	D [px]	IoU
#1	1339	0.819	0.953	1.24	0.639	0.845	0.956	1.2	0.654
#2	1329	0.833	0.967	1.46	0.591	0.875	0.965	1.25	0.612
#3	964	0.812	0.958	1.92	0.5	0.898	0.964	1.3	0.585
#4	1051	0.808	0.948	1.75	0.666	0.866	0.921	1.78	0.662
#5	487	0.830	0.967	1.43	0.707	0.899	0.946	1.52	0.697
Average	1034	0.821	0.959	1.56	0.621	0.877	0.95	1.2	0.66

Table 2: Segmentation metrics for the CAN, K-means, Otsu and adaptive thresholding algorithms.

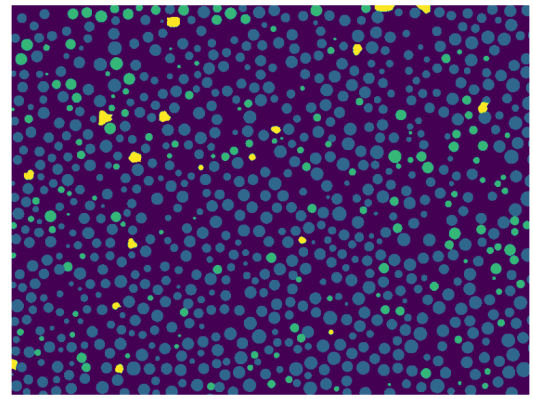
particles. A morphological opening with size 3 pixels is applied to the mask to remove the spurious small particles identified by the clustering procedure. Next, to identify the suspension particles



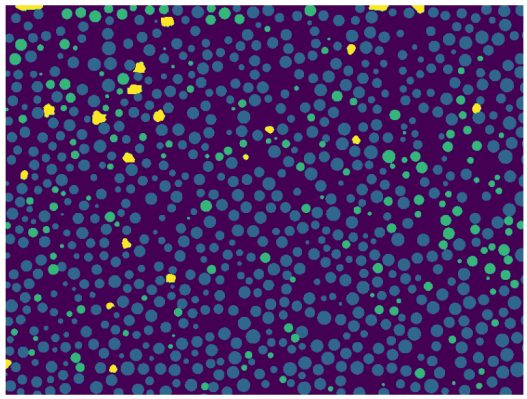
(a) Original image



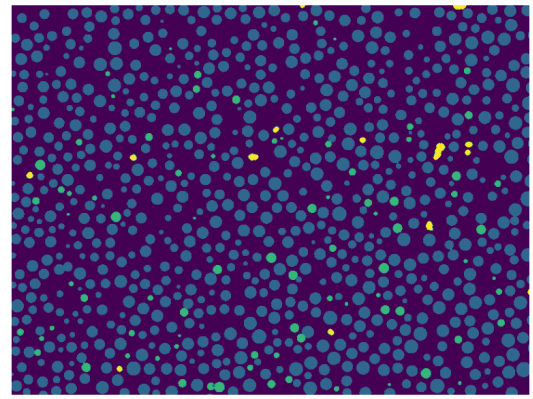
(b) CAN segmentation



(c) K-means segmentation

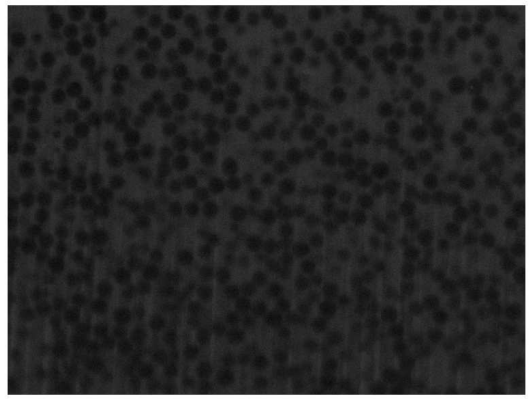


(d) Otsu thresholding

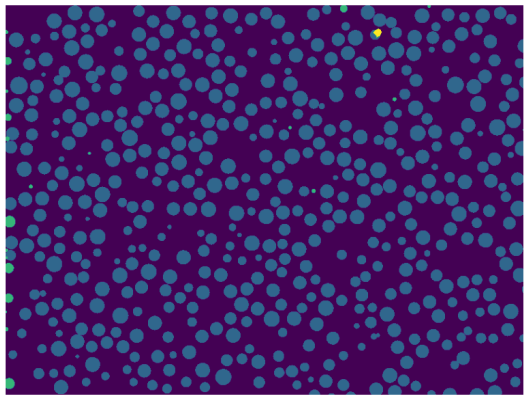


(e) Adaptive thresholding

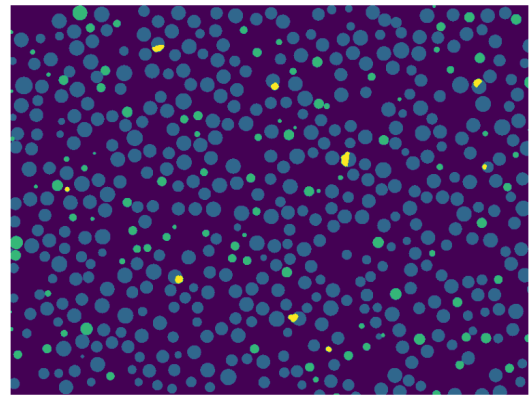
Fig. 5: Segmentation results obtained for the image #3 (a) with the CAN network (b), K-means (c), Otsu thresholding (d) and adaptive thresholding (e). Correct detections (`tp`) are displayed in blue, false positives (`fp`) in yellow and false negatives (`fn`) in green.



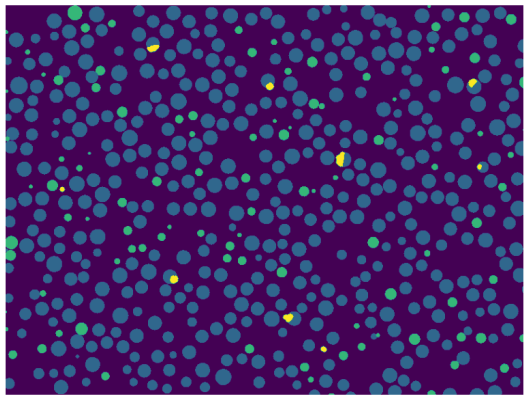
(a) Original image



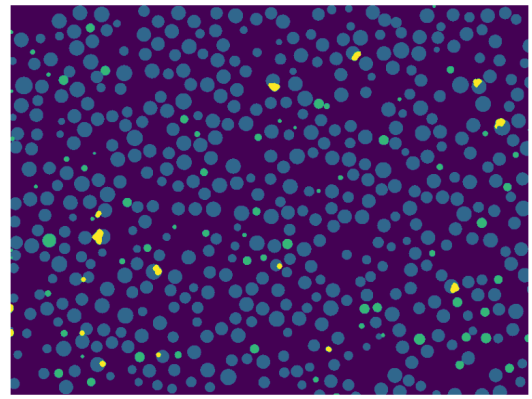
(b) CAN segmentation



(c) K-means segmentation



(d) Otsu thresholding



(e) Adaptive thresholding

Fig. 6: Segmentation results obtained for the image #4 (a) with the CAN network (b), K-means (c), Otsu thresholding (d) and adaptive thresholding (e). Correct detections (`tp`) are displayed in blue, false positives (`fp`) in yellow and false negatives (`fn`) in green.

individually, we compute a distance function on the mask and apply a watershed algorithm selecting as initial markers the maxima of the distance function.

Otsu thresholding

Otsu's algorithm (Otsu, 1979) is a classical thresholding method in image processing, which exploits the gray levels histogram of the image to automatically propose a threshold value. The algorithm exhaustively searches for the threshold that minimizes the intra-class variance of the two clusters of gray levels identified after the thresholding. As for the K-means clustering approach, the interest of the Otsu thresholding algorithm is that it allows to automatically adapt the value of the threshold selected for the segmentation to the gray levels of the image. The threshold value remains nevertheless selected from the global gray level histogram of the image. After the thresholding, we dispose of a binary mask indicating the presence or the absence of particles. To identify the suspension particles individually, we proceed as for the K-means approach by computing a distance function on the mask and applying a watershed algorithm using the maxima of the distance function as markers.

Adaptive thresholding

Adaptive thresholding (Gonzalez and Woods, 2007) consists in thresholding each pixel with a threshold value adapted to the neighborhood of the pixel considered. In practice, for each pixel, the algorithm calculates the average of the gray levels in a fixed neighborhood and subtracts from the result a threshold value specified beforehand. The main advantage of this approach is that the threshold is not specified for the entire image but is adapted locally. This allows to increase the robustness of the segmentation with respect to changes of illumination. Adaptive thresholding allows us to obtain a binary mask indicating the presence or not of a particle. Again, for labeling each individual particle, we compute a distance function on the mask and apply a watershed transform using the maxima of the distance as markers.

3.3 Detection metrics

A prerequisite to quantitatively evaluate the detection results is to establish a correspondence between the particles present in the ground truth and the particles detected by the algorithm, referred to as the detections in this paragraph. Let us denote by $(c_1, \dots c_P)$ the spatial coordinates of the centers of the P particles $(\mathcal{P}_1, \dots \mathcal{P}_P)$ present in the ground truth and by $(R_1, \dots, R_P)$ their respective radii, directly available from the annotations. Similarly, let us denote by $(\hat{c}_1, \dots \hat{c}_Q)$ the centers of the Q detected particles $(\mathcal{D}_1, \dots \mathcal{D}_Q)$ and by $(\hat{R}_1, \dots, \hat{R}_Q)$ their equivalent radii. We can define a dissimilarity measure $C_{p,q}$ for each pair of particle and detection $(\mathcal{P}_p, \mathcal{D}_q)$ by considering the Euclidean distance separating the center of the particle $\mathcal{P}_p$ from the center of the detection $\mathcal{D}_q$. When $\|c_p - \hat{c}_q\|_2 > R_p + \hat{R}_q$, the particle and the detection do not overlap and we fix the value of $C_{p,q}$ to a large arbitrary value. Using the dissimilarity matrix C, we can then associate each detection to a single particle in the ground truth in a way that minimizes the global dissimilarity measure

$$J(A) = \sum_{\substack{1 \leq p \leq P \\ 1 \leq q \leq Q}} C_{p,q} A_{p,q} \,, \tag{11}$$

where A is a $P \times Q$ matrix whose coefficients satisfy

$$A_{p,q} = \begin{cases} 1 & \text{if } \mathcal{P}_p \text{ is affected to } \mathcal{D}_q \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

When the number of detections is higher than the number of particles in the ground truth, we make sure that each detection is associated to at most one particle and that each particle in the ground truth is associated to exactly one detection by requiring A to satisfy the constraints

$$\begin{cases} \text{for all } q \in [\![1, Q]\!], & \sum_{p=1}^{P} A_{p,q} \leq 1 \\ \text{for all } p \in [\![1, P]\!], & \sum_{q=1}^{Q} A_{p,q} = 1 \,. \end{cases} \tag{13}$$

Similarly, when the number P of particles in the ground truth is higher than the number Q of detections, we make sure that each particle is associated to at most one detection and that each detection in the ground truth is associated to

exactly one particle by requiring A to satisfy the constraints

$$\begin{cases} \text{for all } p \in [\![1, P]\!], & \sum_{q=1}^{Q} A_{p,q} \leq 1 \\ \text{for all } q \in [\![1, Q]\!], & \sum_{p=1}^{P} A_{p,q} = 1\,. \end{cases} \tag{14}$$

Equation (11) along with the constraints is a well-known problem in discrete optimization, referred to as linear sum assignment problem, for which efficient algorithms are readily available (Crouse, 2016). Importantly, in the solution of (11), it remains possible for a detection to be associated with a particle in the ground truth that it does not intersect. In this case, we discard the association between $\mathcal{D}_q$ and $\mathcal{P}_p$:

$$A_{p,q} = 0 \quad \text{if } \mathcal{P}_p \cap \mathcal{D}_q = \emptyset\,. \tag{15}$$

Once the association is established between the particles and the ground truth, we can determine the number `fp` (for false positive) of false detections by counting the number of detections that are not associated with any particle. Similarly, we can determine the number `fn` (false negative) of undetected particles in the ground truth by counting the number of particles in the ground truth left unassociated. The number `tp` (true positive) of correct detections corresponds to the number of associations established between the particles of the ground truth and the detections. The ability of the algorithm to properly detect the suspension particles is described in terms of precision and recall, defined by

$$\text{Recall} = \frac{\texttt{tp}}{\texttt{tp} + \texttt{fn}}, \quad \text{Precision} = \frac{\texttt{tp}}{\texttt{tp} + \texttt{fp}}\,. \tag{16}$$

In practice, the recall characterizes the proportion of particles that are detected, while the precision characterizes the accuracy of the detections. For correct detections, we can compute different metrics that characterize the quality of the segmentation including the distance $D = \|c_p - \hat{c}_q\|_2$ between the center of the detection and the actual center of the particle as annotated in the ground truth or the intersection over union (IoU) of the particle and the detection, defined by

$$\text{IoU}(\mathcal{P}_p, \mathcal{D}_q) = \frac{\mathcal{P}_p \cap \mathcal{D}_q}{\mathcal{P}_p \cup \mathcal{D}_q}\,. \tag{17}$$

For physical applications, obtaining a segmentation with good IoU metrics is of particular importance when stereological measurements are conducted to infer the volume fraction of the suspension from the bidimensional segmented images.

3.4 Discussion

The results for the CAN neural network, the K-mean segmentation and the Otsu and adaptive thresholding algorithms for the segmentation metrics (precision P, recall R and average IoU between the particle and the detection mask) are reported in Fig. 7 and Tab. 2, where we also report the average distance D (in pixels) between the centers of the particles and the centroid of their corresponding detection masks.

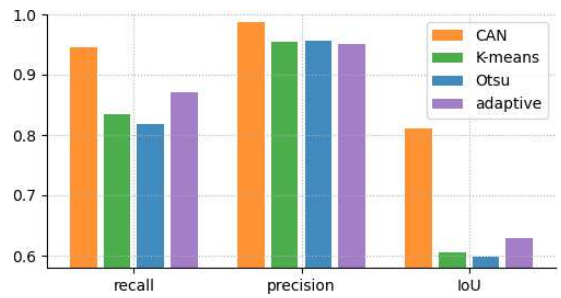


Fig. 7: Average over the 5 images of Tab. 2 of the segmentation metrics (recall, precision, IoU) for the CAN, K-means, Otsu and adaptive thresholding algorithms.

Overall, we can note that the quality of the segmentation obtained with the convolutional neural network significantly surpasses the performances of traditional approaches for all considered metrics. Among the images of the test dataset, we can notice a slightly lower performance of the algorithm for image #3: this is explained by the fact that this image is of lower quality than the other images, which makes its segmentation more difficult. It is interesting to note that all proposed approaches yield relatively similar results in terms of precision, which is systematically greater than 0.95. It is essentially on the recall metric that the performance of the convolutional network stands out. There is also a very significant improvement in the localization of the center of the particles, which falls below one pixel on average for the particles of the test images with the convolutional

network architecture. Finally, the IoU metric is significantly higher than the one obtained by conventional algorithms. Among the classic algorithms, it should finally be noted that adaptive thresholding leads to better detection results than the K-means approach or Otsu thresholding. This highlights the importance of adapting the threshold value locally, rather than setting it based upon global image information.

As an illustration, segmentation examples are displayed in Figs. 5 and 6 for the compared algorithms. In the figures presenting the segmentation results, the particles from the ground truth that were correctly detected are displayed in blue, and false negatives are displayed in green. To represent the false positives, we superimposed them in the ground truth image in yellow color. We can note that both false positives and negatives often occur for particles for which the manual segmentation is relatively ambiguous or even inaccurate in some cases. This is for instance the case in Fig. 8, where the segmentation algorithm yields a false detection that clearly results from an annotation mistake. This illustrates once again the fact that the reliability of manual annotations remains questionable. In practice, it can be considered that the neural network architecture allows to obtain a segmentation of quality similar to that of a manual segmentation.

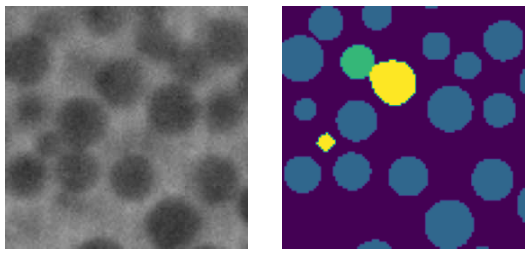


Fig. 8: Example of annotation mistake. The segmentation algorithm produces a detection classified as a false positive (in yellow) due to the fact that the corresponding particle was mistakenly forgotten during the annotation process.

4 Conclusion

In this article, we present a novel methodology for segmenting experimental images of a suspension using a convolutional neural network trained on synthetic images generated with a morphological model. When applied to actual images of the suspension, the convolutional neural network presents good generalization properties and surpasses the performances of traditional segmentation algorithms. To make the best use of images gathered during physical experiments, and in particular of the rheology experiments in which we are interested in this article, it is crucial to be able to rely on efficient image processing tools. Supervised learning algorithms such as convolutional neural networks constitute the current state of the art in image segmentation. Our study underlines the interest of relying on morphological models to generate reliable training samples in situations where annotated images are not available in order to be able to use these supervised approaches.

Declarations

Ethical Approval

Not applicable.

Competing interests

I declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Authors contribution

D.P., B.F. and T.D. developed and programmed the segmentation algorithms. F.B. carried out the rheology experiments. The authors participated equally in the writing of the article.

Funding

This work was supported by GDR EGRIN CNRS 3485 and by the French government, through the UCAJEDI Investments in the Future project managed by the National Research Agency (ANR) under reference number ANR-15-IDEX-01. The authors are grateful to the OPAL infrastructure and the Université Côte d'Azur's Center for High-Performance Computing for providing resources and support.

Availability of data and materials

The code and the data used in this article are available in the GitHub repository https://github.com/bruno-figliuzzi/blob_net and upon request from the corresponding author.

References

Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34, 2021.

Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020.

Matthew Bierbaum, Brian D Leahy, Alexander A Alemi, Itai Cohen, and James P Sethna. Light microscopy at maximal precision. *Physical Review X*, 7(4):041007, 2017.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Frédéric Blanc, Elisabeth Lemaire, Alain Meunier, and François Peters. Microstructure in sheared non-brownian concentrated suspensions. *Journal of rheology*, 57(1):273–292, 2013.

Vincent Bortolussi, Bruno Figliuzzi, François Willot, Matthieu Faessel, and Michel Jeandin. Morphological modeling of cold spray coatings. *Image Analysis and Stereology*, 37(2):145–158, 2018.

Jasna Brujić, Sam F Edwards, Dmitri V Grinev, Ian Hopkinson, Djordje Brujić, and Hernán A Makse. 3d bulk measurements of the force distribution in a compressed emulsion system. *Faraday discussions*, 123:207–220, 2003.

Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017.

John C Crocker and David G Grier. Methods of digital video microscopy for colloidal studies. *Journal of colloid and interface science*, 179(1): 298–310, 1996.

David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

Enzo D'Ambrosio, Frédéric Blanc, and Elisabeth Lemaire. Viscous resuspension of non-brownian particles: determination of the concentration profiles and particle normal stresses. *Journal of Fluid Mechanics*, 911, 2021.

Joshua A Dijksman, Nicolas Brodu, and Robert P Behringer. Refractive index matched scanning and detection of soft particles. *Review of Scientific Instruments*, 88(5):051807, 2017.

Edward Dougherty. *Mathematical morphology in image processing*. CRC press, 1992.

Bruno Figliuzzi. Eikonal-based models of random tessellations. *Image Analysis & Stereology*, 38 (1):15–23, 2019.

Bruno Figliuzzi, Dominique Jeulin, Matthieu Faessel, François Willot, Masataka Koishi, and Naoya Kowatari. Modelling the microstructure and the viscoelastic behaviour of carbon black filled rubber materials from 3d simulations. *Technische Mechanik-European Journal of Engineering Mechanics*, 36(1-2):32–56, 2016.

Bruno Figliuzzi, Kaiwen Chang, and Matthieu Faessel. Hierarchical segmentation based upon multi-resolution approximations and the watershed transform. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 185–195. Springer, 2017.

Bruno Figliuzzi, Antoine Montaux-Lambert, François Willot, Grégoire Naudin, Pierre Dupuis, Bernard Querleux, and Etienne Huguet. A bayesian approach to morphological models characterization. *Image Analysis & Stereology*, 40(3):171–180, 2021.

Rafael C Gonzalez and Richard E Woods. *Digital Image Processing (3rd Edition)*. Pearson, 2007.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

Dominique Jeulin. *Morphological models of random structures*. Springer, 2021.

C Kimme, D Ballard, and J Sklansky. Finding circles by an array of accumulators. *Communications of the ACM*, 1975.

Manon Nagy, Luca Istrate, Matei Simtinică, Sébastien Travadel, and Philippe Blanc. Automatic detection of marine litter: A general framework to leverage synthetic data. *Remote Sensing*, 14(23):6102, 2022.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Braden Snook, Jason E Butler, and Élisabeth Guazzelli. Dynamics of shear-induced migration of spherical particles in oscillatory pipe flow. *Journal of Fluid Mechanics*, 786:128–153, 2016.

Dietrich Stoyan, Wilfrid S Kendall, Sung Nok Chiu, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13 (06):583–598, 1991.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, May 2016.