# Technical information: Object detection details

This technical information summary supplements the main article with a more detailed elaboration on the developed and utilized automated image processing software. As the analysis of the images obtained from holographic measurements is described in sufficient detail in the main article and the references therein, this summary focuses on the techniques used for analysis of the obtained microscope images. With each of the particle mass experiments yielding around 20 image pairs, each containing up to several dozens of crystals and drops, measuring crystal size and drop diameter manually for every single object is unfeasible. To benefit from the large amount of image data generated through this setup, automated image analysis software was developed to automatically find and measure objects within the image. The methods used for this software are described in Sections S1 to S4, and an evaluation and comparison of accuracy follows in Section S5.

## S1 Global thresholding

Initially, a global grayscale threshold value was selected for binarizing the $2048 \times 2048$ pixel wide 8-bit grayscale images into object pixel components and background. As objects on the glass slide prevent light from the LED below from reaching the camera above through refraction, they usually have a darker shade in the image than the background. Therefore, pixels with grayscale values below the threshold are considered object pixels, while pixels with values higher than the threshold are considered part of the background (see Fig. 1).

The determination of a suitable global threshold has been performed using two different approaches. For the first one, the threshold $T_{mean}$ was determined relative to the distribution of grayscale values throughout the image following

$$T_{mean} = \overline{I} + 0.5\sigma_I, \tag{1}$$

where $\overline{I}$ and $\sigma_I$ are mean and standard deviation of the grayscale intensity in the image. Alternatively, an unsupervised method developed by Otsu (1979) has been used to automatically determine a suitable threshold for grayscale image segmentation from the image's grayscale histogram. Their algorithm calculates a threshold value that maximizes the separability of the two classes generated, which objectively optimizes the value selection for separating background from signal. This approach has been used for every fall streak image individually rather than determining a value that fits the varying circumstances in all different images, as lighting conditions can still vary moderately from image to image.

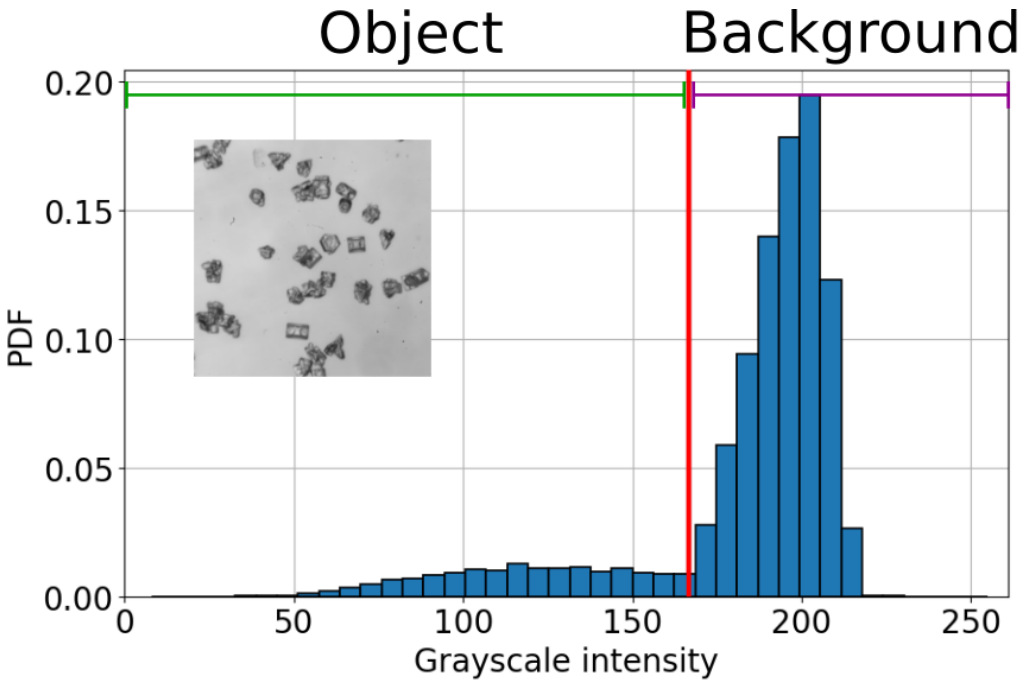


Figure 1: Histogram of grayscale values in the enclosed sample microscope image. The vertical red line shows the threshold value $T_{mean}$ determined for a sample image using Eq. (1).

## S2 Adaptive thresholding

As ice is transparent for visible light, only the crystals' edges create darker shades in the microscopic images, whereas flat surfaces are almost indistinguishable from the background without context information of the edge pixels around them. While global thresholding proved successful for many images, particularly those of crystals with a high degree of irregularity, the segmentation turned out notably worse for some more pristine crystal types. The edges of columns are straight and drawn-out, which often made them keep a lighter shade than the edges with more complex shapes that are common for irregular crystals. Dendritic crystals showed even less contrast to the surrounding background pixels, making them harder to detect. The contrast between ice crystals and background often became so narrow that it did not exceed the brightness variations within the images themselves by a sufficient margin to ensure proper segmentation. This difficulty prompted the addition of more binarization approaches to improve the detection and sizing of the crystals.

A natural improvement to the global thresholding approach was the utilization of a threshold that is dependent on the location of the investigated pixels in space. Through this, the effect that different lighting conditions in different areas of the image had on segmentation quality is decreased. A pixel was classified into object pixel or background pixel based on the following decision:

$$c(p) = \begin{cases} 0 & I(p) > T(p) \\ 1 & I(p) \leq T(p), \end{cases} \tag{2}$$

with $c(p)$ symbolizing the classification, $I(p)$ the pixel's grayscale intensity and $T(P)$ the local (adaptive) threshold determined for $p$. In this case, to determine the threshold value $T(x_0, y_0)$ for a pixel $p$, a 2D Gaussian window with a window size $k$ was created and a cross correlation with the $k \times k$ neighborhood of $p$ was calculated. The threshold value is then the weighted sum of this cross correlation, with the weights

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3}$$

where x and y are the distances of a pixel to $(x_0, y_0)$ in both horizontal directions (in pixel index coordinates) and $\sigma$ is the Gaussian window's standard deviation. The window size $k$ (in pixels) had to be chosen appropriately: the larger $k$ is, the closer the results converged with a global thresholding approach and the advantage of adaptive thresholding fades. However, if the window was chosen too small, the segmentation of crystals often remained incomplete, as the brighter regions in their center or at edges with low gradients had grayscale values higher than the locally limited threshold. Optimal results were observed for values of $600 < k < 1200$, depending on lighting conditions and crystal habit present in the image.

# S3 Canny edge detection

As edges of objects were very pronounced in the microscopic ice crystal images in this work due to the crystals' faces being transparent for light with visible wavelength, we looked for a segmentation technique that is based on topological context instead of just trying to separate pixel values into classes or clusters based on their own intensity alone. Edge detection algorithms attempt to find regions in images where the intensity has spatial discontinuities, for example in the form of sharp steps from lower to higher values. The use of spatial intensity gradients is a central tool for this analysis approach, as these steps appear as extrema in such fields. Canny (1986) proposed an approach to edge detection that supplements the use of gradients with other effective tools in a sequential combination. Similarly to the thresholding methods described above, a Gaussian filter is applied first to remove noise from the image. Afterwards, the intensity gradient $\mathbf{C}(p)$ and its direction $\mathbf{\Phi}(p)$ are determined for each pixel $p$ by applying Sobel operators $\mathbf{S}$ in each direction,

$$\mathbf{C}_x = \mathbf{S_x} * \mathbf{I} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathbf{I},$$

$$\mathbf{C}_y = \mathbf{S_y} * \mathbf{I} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I}, \tag{4}$$

$$\mathbf{C}(p) = \sqrt{\mathbf{C}_x^2 + \mathbf{C}_y^2}, \tag{5}$$

$$\mathbf{\Phi} = \tan^{-1}\left(\frac{\mathbf{C}_y}{\mathbf{C}_x}\right). \tag{6}$$

The gradient direction is then discretized by rounding to one of four values, resulting in either a horizontal, vertical or one of the diagonal directions. A non-maximum suppression is applied to the gradient image $\mathbf{C}$ to more precisely emphasize the edges that become visible in the gradient image. For this step, the neighborhood of each pixel $p$ along the gradient direction $\mathbf{\Phi}(p)$ is compared to

the pixel itself in the gradient image $\mathbf{C}$. If the gradient is at a local maximum at $p$, the gradient value is retained. If one of the neighbor pixels' gradient values is larger than $\mathbf{C}(p)$, the pixel $p$ is suppressed and $\mathbf{C}(p)$ thus set to zero. This results in an image $\mathbf{C}_{reduced}$ containing thin lines representing the locally strongest gradients. In the final step of the edge detection algorithm, all edge pixels in $\mathbf{C}_{reduced}$ are subjected to a double thresholding using two values $t_{min}$ and $t_{max}$. All pixels with gradient values lower than $t_{min}$ are discarded and set to zero. All values greater than $t_{max}$ are classified as definitive edge pixels. Pixels with values $t_{min} < \mathbf{C}(p) < t_{max}$ are classified as edge pixels if they are directly connected with a definitive edge pixel (or a previously classified connected edge pixel), and discarded otherwise. To obtain a segmented image from the contours created by the Canny edge detection algorithm in the application to ice crystal images, a suitable filling method had to be used. Alternatively, the segmentation step could be skipped and the ice crystal's size and shape could be derived directly from the contour information. As the Canny algorithm detected all the inner structures of individual crystals as their own contour in the images at hand, the actual crystal outlines had to be filtered out, which was non-trivial especially for images containing many crystals.

# S4 k-means clustering

Another approach to object detection was a segmentation based on the application of clustering algorithms. Clustering is an unsupervised machine learning type where image pixels are separated into an arbitrary number of different classes, usually while trying to maximize both inner-class similarity and the contrast between different clusters. A wide variety of methods exists, including different implementations such as mean-shift clustering, expected maximization clustering, or k-means clustering (e.g. Dhanachandra et al., 2015). For the segmentation of the ice crystal microscope images at hand, the k-means clustering method has been selected as it is a versatile and efficient method that can easily be adapted for different problems. The k-means approach is centroid-based, which means each of the clusters has a data point that is considered its center, and all other data points are evaluated based on their similarity to the centroids which has to be quantified by an appropriate metric. The cluster centroids have been automatically initialized by the algorithm in this application and were shifted if required, for example, if the observed resulting segmentation was inaccurate. The implementation used here is based on the `scikit-learn` toolbox, a machine learning toolbox written in Python (see Pedregosa et al., 2011).

Applying k-means clustering to the grayscale images themselves, without any additional channel, is equivalent to a thresholding approach, as the calculation of a pixel's similarity to the centroids is only based on the grayscale intensity. The possibility of including more dimensions or channels to the clustering allows taking the spatial context of the pixels into account, such as a grayscale gradient or an image obtained from applying a convolution filter. That way, a $2048 \times 2048 \times n$ representation of the image could be constructed and used as the foundation of an automatic segmentation of the initial grayscale image. For this segmentation, a secondary image channel $\mathbf{D}(x, y)$ has been calculated using a convolution of the image with a Sobel operator similar to the approach taken for Canny edge detection. The value of this secondary channel at each pixel was thus calculated from

$$\mathbf{D}_x = \mathbf{S}_\mathbf{x} * \mathbf{I} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathbf{I},$$

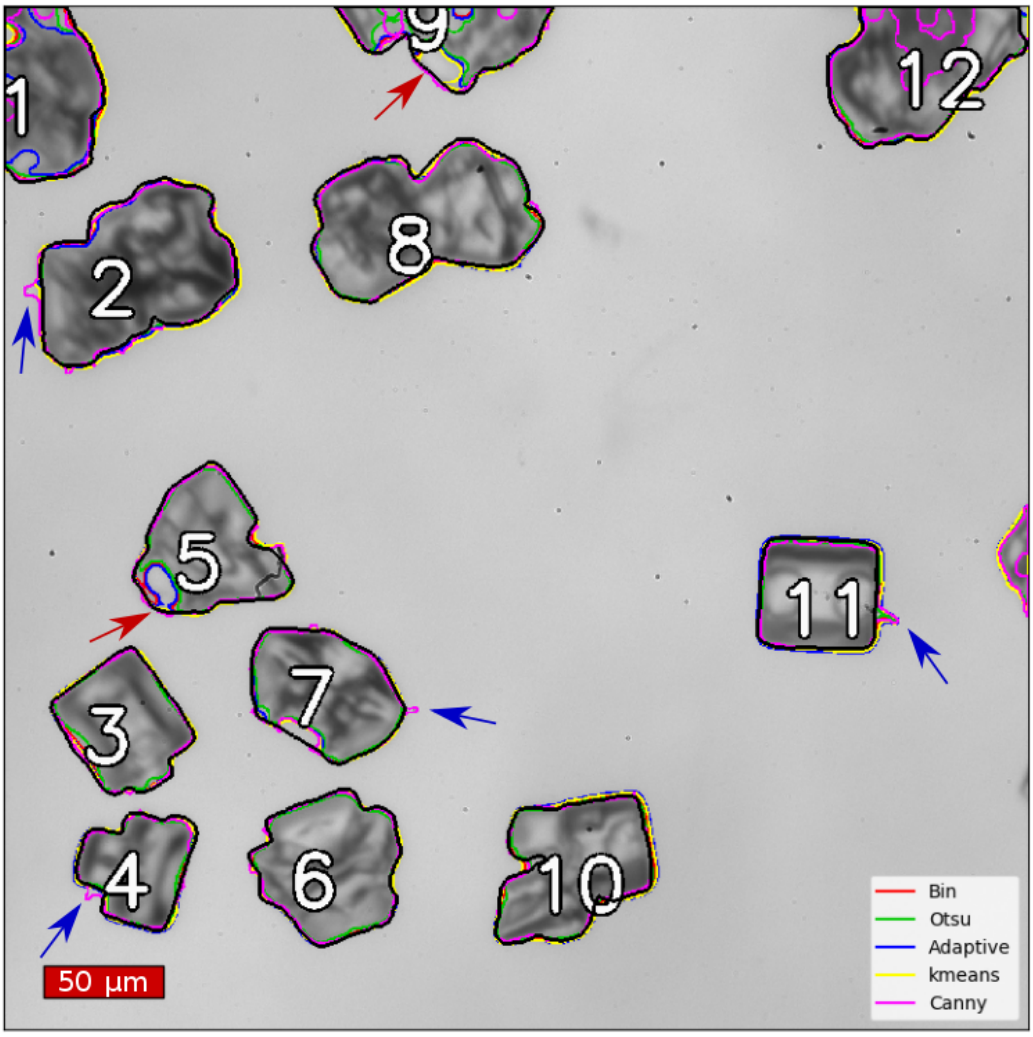


Figure 2: Example image for crystal sizing using different segmentation methods with optimized parameters. Crystal contours marked by operator in black, different thresholds in colors. Blue arrows show examples of oversizing through noise contour merging through dilation, red arrows show examples of area underestimation through incomplete patches.

$$\mathbf{D}_y = \mathbf{S_y} * \mathbf{I} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I},$$

$$\mathbf{D}(x,y) = \sqrt{\mathbf{C}_x^2 + \mathbf{C}_y^2}, \tag{7}$$

with $\mathbf{I}$ representing the grayscale intensity of the initial image. This image filter combines a spatial grayscale gradient of the initial image $\mathbf{I}$ with a Gaussian smoothing, thus emphasizing the impact of

a pixel's spatial context on its cluster membership instead of relying solely on the pixels' grayscale intensities.

## S5 Segmentation method comparison and evaluation

To allow for a quantitative comparison and thus an assessment of performance of the algorithms relative to each other, several images were labeled by an operator tracing the edges of crystals by hand. While this approach was subjective to a certain degree and can contain errors from inaccuracy introduced by the person tracing the contours, it was considered as the "ground truth" in this attempt to evaluate contour quality. Fig. 2 shows contours detected in the image by the different methods described above in direct comparison, with the ground truth contour displayed in black. For most crystals, the colored contours are almost indistinguishable from each other and from the ground truth. However, in several cases brighter parts of the crystals have been missed by the detection algorithms. For the crystals labeled with number 5 (center left of the image) and 9 (central region of the image's top edge), only the Canny edge detection method was able to trace the contour around the brighter regions (red arrows). The other methods drew the crystals' contour in a way that left holes in their shapes. Those missed regions can lead to an underestimation of particle size from the smallest enclosing circle $D_{sec}$ depending on the severity and location of the holes, and always leads to an underestimation of the area equivalent diameter $D_{ae}$.

A different source of error that caused oversizing can be seen for crystals $2, 4$ and particularly crystal number 11 (blue arrows in Fig. 2). Even though a Gaussian blur filter was applied before object detection was started, small dirt spots in the microscope optics or on the glass slide are visible as dark spots in the image, and were thus also detected by the binarization. The small spots themselves were then filtered by a criterion disregarding all components that span less than a certain area $A_{min}$, but a step in the processing procedure which involves a dilation of the contours by a few pixels may have led to them being combined with one of the larger components that were later considered crystals. This created artifacts such as the small thorn protruding from the right side of crystal number 11, which led to an oversizing of the crystal. Crystal 7 exhibits both error sources.

In Table 1, the total numbers of incorrectly labeled pixels in Fig. 2 are listed and compared between the different methods. The values show pixels falsely considered object or background pixels as a fraction of the total number of object pixels in the entire image. In this example, the k-means approach yields the best results with the lowest sum of errors, followed by binary thresholding with a threshold value of $T = \bar{I} - 0.5\sigma_I$, with the mean intensity $\bar{I}$ over the whole image and the intensity standard deviation $\sigma_I$. The intensity value determined as a threshold this way was higher than the threshold determined by Otsu's method ($T_{mean} = 168$ vs. $T_{otsu} = 156$) and thus constituted a less strict criterion, as pixels darker than the threshold are considered object pixels. This is consistent with the observation from Table 1 that the binary method misses fewer object pixels while simultaneously falsely labeling more pixels as part of an object.

Taking a pixelwise approach when evaluating segmentation methods quantifies the impact of mislabeled pixels on determining the area equivalent diameter $D_{ae}$. On the other hand, the impact of missed pixels in the regions marked by red arrows in Fig. 2 on the diameter of the smallest enclosing circle $D_{sec}$ is often very minor. By directly comparing the crystal sizes $D_{sec}$ and $D_{ae}$ determined from different methods in Fig. 2 to the size of the objects in the labeled image, the effect of the segmentation errors on crystal sizing can be evaluated in a more direct way. Fig. 3 shows the size error for each individual crystal in the image when compared to the operator-drawn contours for

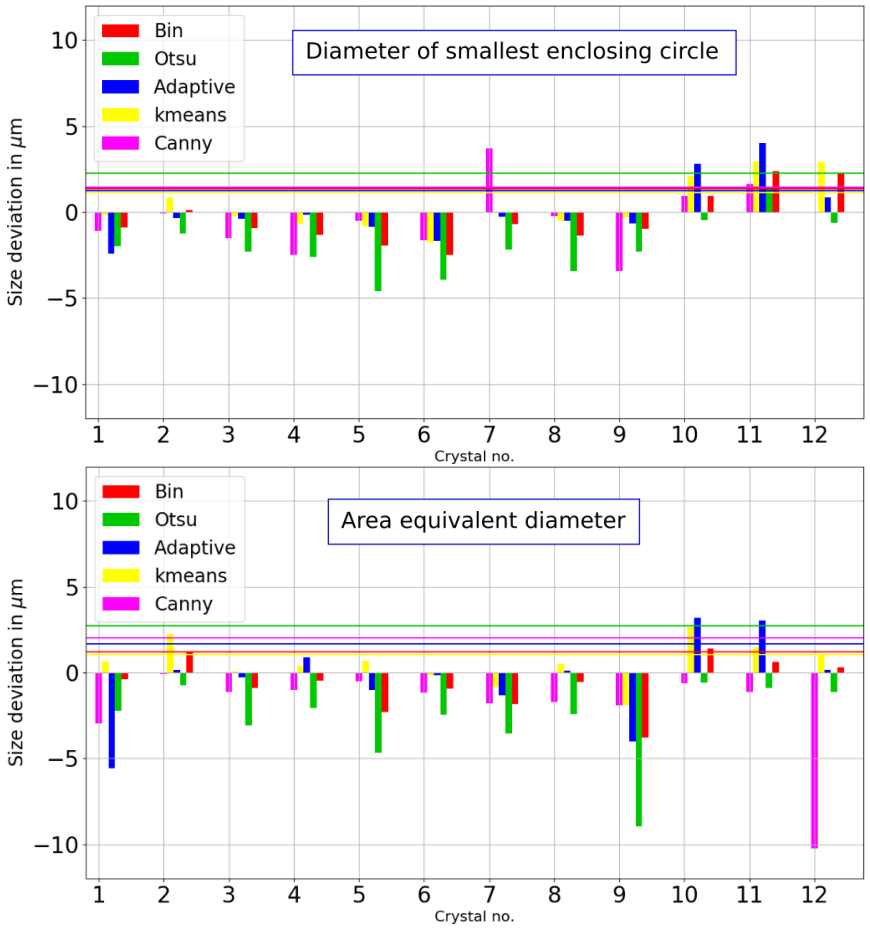


Figure 3: Sizing errors in $D_{sec}$ and $D_{ae}$ introduced by the detected contours shown in Fig. 2. Mean absolute error as horizontal lines.

| **Binarization method** | **False positive pixels** | **False negative pixels** | **Sum of errors** |
|---|---|---|---|
| Global Thr. - Mean | 2.6% | 4.2% | 6.8% |
| Global Thr. - Otsu | 0.9% | 8.8% | 9.7% |
| Adaptive Threshold | 4.0% | 4.6% | 8.6% |
| k-means Clustering | 4.5% | 2.1% | 6.6% |
| Canny Edge Detection | 1.6% | 7.3% | 8.9% |

Table 1: Error in pixel coverage detected by different binarization algorithms over the entire image (Fig. 2, total pixel number $2048 \times 2048 = 4194304$, total number of object pixels in label image 306796). "False positive" refers to pixels predicted as object pixels that are not part of the label objects, "False negative" refers to label object pixels missed by the automated binarization method.

| **Binarization method** | **Mean sizing error** $\Delta D_{sec}$ | | **Mean sizing error** $\Delta D_{ae}$ | |
|---|---|---|---|---|
| | absolute | relative | absolute | relative |
| Global Threshold - Mean | 1.4 µm | 3.4% | 1.2 µm | 2.9% |
| Global Threshold - Otsu | 2.3 µm | 3.6% | 2.7 µm | 5.0% |
| Adaptive Threshold | 1.2 µm | 3.6% | 1.7 µm | 4.0% |
| k-means clustering | 1.1 µm | 3.4% | 1.1 µm | 2.9% |
| Canny Edge Detection | 1.5 µm | 4.3% | 2.0 µm | 3.3% |

Table 2: Mean error of ice crystal sizing relative to operator-labeled images in Fig. 2 for different binarization methods.

each binarization method. The effects of the errors can be seen in the positive or negative size deviations of the respective crystals. Table 2 lists the mean sizing errors over all crystals. The adaptive thresholding method performs better than global thresholding for determining $D_{sec}$, but worse for $D_{ae}$. This implies that a local threshold is more capable of correctly finding object edges for the example at hand, but inconsistently labels the inner object regions, which agrees well with the larger error in pixel count. Overall, the smallest error in both sizing and area estimation is generated by the machine learning approach utilizing k-means clustering. Its capability of including context information around the objects in their segmentation provides a clear advantage compared to the thresholding approaches. However, as the cluster centroids are initialized from randomized locations the algorithm does not always converge to a similar solution. Therefore, manual adjustment of the accepted cluster indices or a clustering repetition using a different initial centroid seed is occasionally necessary to receive the optimal binarization result.

# References


Canny, J. (1986). "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. ISSN: 01628828. DOI: 10.1109/TPAMI.1986.4767851.

Dhanachandra, N., K. Manglem, and Y. J. Chanu (2015). "Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm". In: *Procedia Computer Science* 54, pp. 764–771. ISSN: 18770509. DOI: 10.1016/j.procs.2015.06.090.

Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transaction on Systems, Man and Cybernetics* 20.1, pp. 62–66. ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076.

Pedregosa, F., V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher, M. Perrot, and É. Duchesnay (2011). *Scikit-learn: Machine Learning in Python*. Tech. rep. Parietal, INRIA Saclay, pp. 2825–2830.