Image-based characterization of flocculation processes through PLS inspired representation learning in convolutional neural networks

Baum, Andreas; Moiseyenko, Rayisa; Glanville, Simon; Martini Jørgensen, Thomas

Published in:
Journal of Chemometrics

Link to article, DOI:
10.1002/cem.3534

Publication date:
2024

Document Version
Version created as part of publication process; publisher's layout; not normally made publicly available

Received: 9 March 2023 | Revised: 27 December 2023 | Accepted: 29 December 2023

DOI: 10.1002/cem.3534

RESEARCH ARTICLE

Journal of CHEMOMETRICS WILEY

Image-based characterization of flocculation processes through PLS inspired representation learning in convolutional neural networks

Andreas Baum[1] | **Rayisa Moiseyenko**[1] | **Simon Glanville**[2] | **Thomas Martini Jørgensen**[1]

[1]Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark

[2]091 Downstream Optimization, Product and Process Development, Novonesis, Kalundborg, Denmark

Correspondence
Andreas Baum, Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark.
Email: andba@dtu.dk

Funding information
Danmarks Tekniske Universitet; Innovationsfonden, Grant/Award Number: 10513

Abstract

Monitoring of flocculation processes such as those used in downstream processing of a fermentation broth is essential for process control. One approach is to apply microscopic imaging combined with image analysis for characterizing the state of the process. In this work, we investigate and compare the use of supervised feedforward convolutional neural network (CNN) architectures to predict the process states from the image information and compare the results with the traditional alternative of characterizing flocs based on manually engineered image features guided by human expertise. From a well-defined image data set representing six process states, the objective is to establish end-to-end classification models which are accurate but at the same time learn meaningful latent variable space representations. Specifically, we evaluate three different CNN architectures with varying degrees of regularization and compare results with logistic regression models based on inputs from two different traditional feature engineering methods. By applying global average pooling as a structural regularizer to the CNN architecture, we significantly improve the generalization performance in comparison with the classification accuracies of the traditional feature engineered models. Furthermore, we show that by imposing a projection to latent structures (PLS) like regularization framework onto the CNN, it can also learn a latent variable representation that mimics the features selected by human expertise.

1 | INTRODUCTION

Successful process control during manufacturing of bio-products in biotech companies depends on identification of relationships between process parameters and critical quality attributes. Several sources of variation, such as slow process drifts, feedstock quality, and environmental disturbances, are challenges which need to be addressed. Consequently, continuous assessment and evaluation is essential to supervise and control the processes. In this paper, we specifically consider flocculation as it significantly contributes to process capacity, yield, and quality. One important

application relates to downstream processing of a fermentation broth. Coagulation/flocculation are essential pretreatment steps for isolating and subsequent purification of the desired product and are specifically used in the solid removal stage. Adequate control of the amounts of flocculant and coagulants added in these steps in combination with mixing calls for monitoring techniques able to characterize the quality of the flocs.

In the coagulation process, the negatively charged impurities are destabilized by adding a clarifying agent. Particles thus agglomerate, and by adding a polymer, which forms bridges between the particles, larger agglomerates are formed in the flocculation process. Precise process monitoring will provide the means for improved control giving higher yield, better quality, and a reduced consumption of water. In order to monitor the flocculation process, one can measure physico-chemical parameters like the sludge volume or the amount of suspended solids.[1] Such tests have to be carried out in the laboratory and are time-consuming, limiting the use for control. In the domain of waste water treatment, it has been suggested instead to apply image analysis to microscopic images of liquid samples in order to characterize the process state as in Khan et al[1] or da Motta et al.[2] This approach is based on the assumption that shape and texture information of flocs and filaments correlate with the physico-chemical parameters. Different process states would then be associated with different image characteristics, and one could expect to identify corresponding clusters in a latent variable (LV) space representing characteristic image features. Significant outliers from such identified clusters could be used to identify when the process is not running optimally and assist in developing automated control systems for flocculation quality.

Within the last two decades, literature has reported several studies on the use of digital image processing for monitoring of activated sludge systems for waste water treatment. Bright field and phase contrast microscopy are most often used to produce images from the collected samples,[1] but other imaging modalities such as epifluorescence[3] and confocal laser scanning microscopy[4] have also been applied. Most often, the images are acquired by using commercial microscopes and the associated software. Subsequently, noise and background removal are the initial image processing steps, followed by segmentation and characterization of the flocs.

In 1989, Li and Ganczarczyk[5] were successful in describing the spatial structure of particles appearing during water and wastewater treatment based on fractal theory. In addition, later studies indicated correlation between floc strength and size, and an empirical relationship has been developed.[6–9] For those reasons, the fractal dimension together with the floc size distribution have been the most common parameters to extract when applying image analysis to characterize the flocculation process.[10–12]

Da Motta et al[2] studied filamentous bacteria and found correlations between morphological features from the image analysis and the sludge volume index (SVI) as well as the settling velocity. The features were filament total length, the number of filaments, and floc size.

Amaral and Ferreira,[13] also using image processing, found a strong relationship between the total suspended solids (TSS) and the total aggregates area of biomass as well as a close correlation between the filamentous bacteria per suspended solids ratio and the SVI.

The approach by Amaral and Ferreira was extended in later studies studies[14,15] where they concluded that the image analysis methodology is a feasible method for the continuous monitoring of activated sludge systems and identification of disturbances. Partial least square (PLS) regression was used to extract linear combinations of 36 morphological parameters obtained from image analysis.

Jenné et al[16] also explored the characterization of sludge by calculating several image features, including size and shape information of flocs as well as the fractal dimension. Global characteristics were also calculated for filaments and fragments. Based on establishing a model to predict SVI from image features, the set of most relevant features turned out to be the total filament length, elongation of the flocs, and the fractal dimension.

Similarly Arelli et al[17] found a correlation between morphological image features and SVI, as well as between the fractal dimension parameter and SVI. Smoczynski et al[18] performed image analysis on scanning electron microscopy (SEM) images of wastewater sludge. They found that the analyzed sludge samples were made of self-similar aggregates-flocs with fractal characteristics, confirming the early study from 1989.[5]

In the domain of waste water treatment, Khan et al[19] considered images at lower objective magnification but with improved visibility due to the application of phase contrast. After image segmentation of the filamentous bacteria, image analysis parameters related to the morphology of the bacteria were identified and applied with the aim of modelling the SVI. Advanced image processing was used to extract characteristics such as average filament diameter, average filament length, and average filament curvature. They found that the total filament length—that is, summing the length of all the individual observed filaments—had the most predictive power concerning modelling of the SVI.

Recently, Molina et al[20] applied image analysis for characterization of flocs in order to predict sedimentation parameters in wastewater treatment. They used connected component analysis to identify clusters of individual flocs and obtained area and parameter of the individual flocs and found a correlation between the number of flocs per cluster and the sedimentation velocity. Leal et al[21] used quantitative image analysis to characterize the structure of the floccular and granular biomass by their equivalent diameters. Using chemometric methods, they found these features to be crucial for the prediction of the sludge settleability, density, and suspended solids.

In order to simplify the image analysis when characterizing flocculation and coagulation in waste water, Sivchenko et al[22,23] and Sivchenko[24] proposed to analyze the image texture as a whole instead of characterizing the shape of each particle/floc in the image. They used the gray level co-occurrence matrix (GLCM) to obtain a set of texture features, which in turn was used to derive entropy, contrast, variance, and homogeneity measures. Those were combined into a floc texture index and used to create a model for predicting the turbidity. In her PhD thesis, Sivchenko[24] found that flocs have distinct texture features, which correlate with the coagulation conditions.

Yu[25] used digital image analysis to characterize the flocculation and coagulation processes for industrial wastewater plants. He extracted various descriptive parameters regarding size and shape of the particles as well as the total volume and fractal dimension. An artificial neural network was then trained on these data to predict the measured suspended solid removal efficiencies.

The above illustrates quite well that traditional image analysis is based on extracting features being engineered from domain knowledge in combination with trial and error. The extracted features can then be tested in combination with statistical classifiers or machine learning models with the aim of establishing a relationship between the features and the process state. In this study, we conducted monitoring of a flocculation process at a Novozymes production facility located in Kalundborg, Denmark, using an automated microscope. As an alternative, we attempt to apply neural network architectures, which autonomously learn relevant image features while being trained to predict the process state. Hereby, we elucidate what kind of image features the networks pay attention to and compare these with the features extracted by means of human domain knowledge and traditional image analysis. This paper aims to evaluate the performance of three image analysis methods in detecting biomass and classifying among six potential process states (Figure 1). Our proposed methods are designed to identify appropriate LV space representations of the image data, which could describe the various states of the biomass throughout the six different process states. Additionally, these methods are intended to offer immediate and automated recommendations regarding the control status of a flocculation process based on acquired image(s) at a given process stage. Process engineers could utilize the LV representation along with the classification results to detect process deviations early on and initiate necessary process adjustments.

The paper is organized as follows. After a description of the experimental setup and the data material, we describe the methods for data augmentation, feature extraction and classification of the floc images. We first introduce two traditional image analysis methods for feature extraction, which, when being combined with logistic regression, served as classification models for prediction of process state. These two methods are based either on detection of total floc area

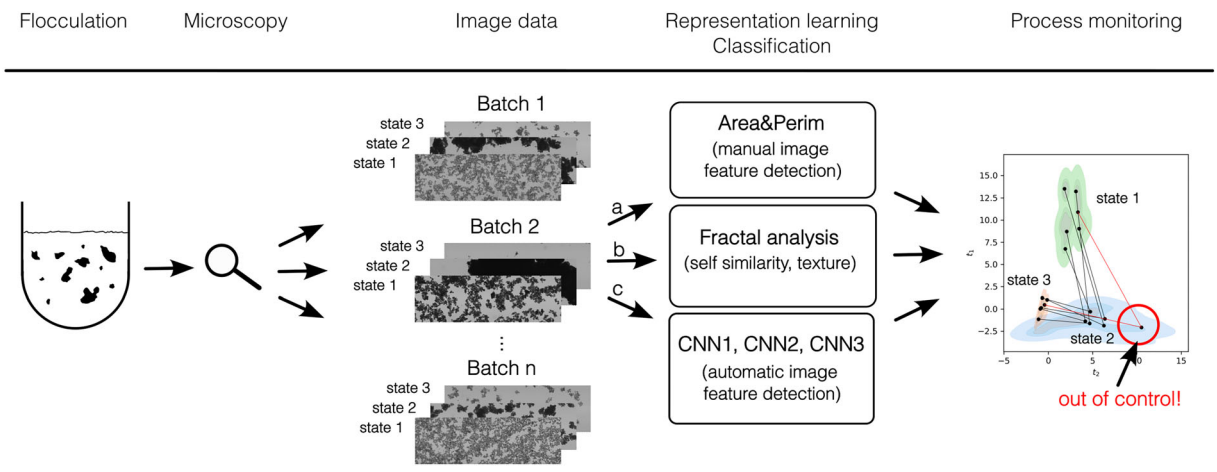

FIGURE 1 Our study focuses on monitoring a flocculation process using microscopic imaging. We compare two traditional image analysis methods, "Area&Perim" (A) and "Fractal" (B), with three CNN architectures, "CNN1", "CNN2", "CNN3" (C). For simplicity, this figure only illustrates three out of the six process states. CNN, convolutional neural network.

and perimeter or fractal analysis and will hereafter be referred to by the aliases "Area&Perim" and "Fractal", respectively. Secondly, we introduce three different convolutional neural network (CNN) architectures to be utilized for automatic feature extraction and classification. After presentation of the result of the three different methods, we compare them and discuss the benefits of using the CNN approach over traditional image feature extraction. The three CNN architecture will be referred to using the aliases "CNN1", "CNN2", and "CNN3".

2 | EXPERIMENTAL SETUP AND DATA

The production system used in this study is a continuous two-step separator system as part of a downstream process. In order to separate biomass, that is, remaining cells, from the produced protein, flocculation, and decantation, are performed sequentially yielding six major process states (see Figure 2). The detailed outline is as follows. The upstream fermentation broth, referred to as Feed 1, serves as input to the first separation step. Flocculation is initiated by adding various salts to the reaction broth in order to establish the correct charge dispersion for particle coagulation. This is followed by the addition of one or more polymers resulting in flocculation (Separator 1). Next, the flocculated biomass is removed from the broth using decantation resulting in Separator stream 1, which is passed on further downstream. The sedimented biomass, on the other hand, is rediluted resulting in Feed 2 which serves as input to a second separation step. Flocculation of Feed 2 is carried out as described above resulting in Separator 2. Finally, the flocculated biomass is then separated from the broth using decantation resulting in Separator stream 2 which is directed downstream for further processing. The sedimented biomass is rediluted and recycled in order to maximize yield.

From the production system shown in Figure 2, liquid samples were collected at each of the six processing steps, and digital microscopic images were recorded using an automated microscope (oCelloScope, BioSense Solutions, 1.3 μm resolution). A total of 118 images were obtained from 17 batches of the same product, resulting in a slightly unbalanced data set, that is, some batches were measured twice at given process states, while images for a number of batches were partly missing due to production and/or scheduling constraints.

3 | METHODS

In the following, tensors will be denoted using calligraphic letters, matrices upper-case bold, vectors lower-case bold, scalars lower-case italic, and tensor as well as matrix dimensions using upper-case italic letters. Tensor elements, for example, a tensor element at position i, j, k, will be denoted using $\mathcal{X}(i,j,k)$.

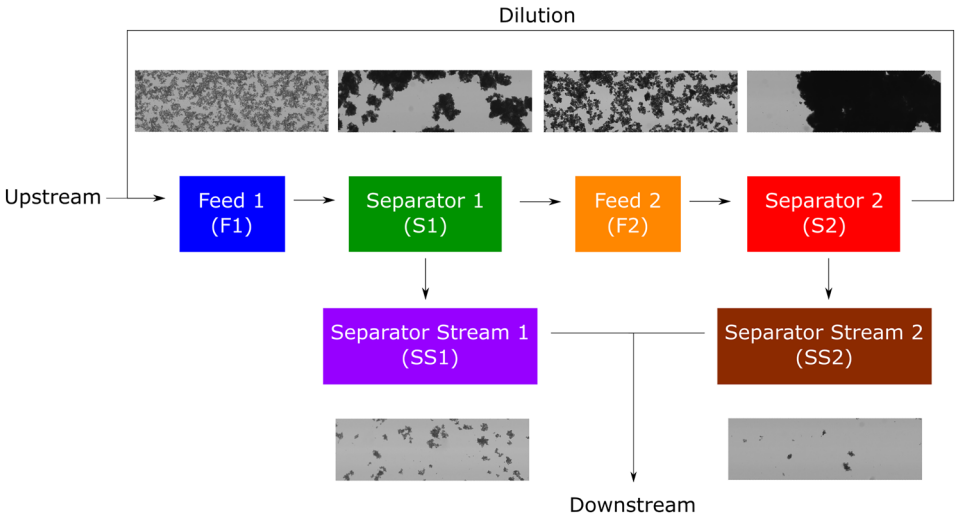

FIGURE 2 The six sequential process states and representative microscopic images. It is noteworthy that Separator 2 contains denser flocculated biomass in comparison with Separator 1. Separator stream 1 contains more chunks of remaining biomass compared with Separator stream 2

3.1 | Prediction models based on traditional image analysis

From the studies reported in the literature, we know that various size measures of flocs as well as the self-similar texture feature of flocs have been found to be good predictors for the thermodynamic state of the flocculation process. We therefore have also established logistic regression models based on such measures. In addition, we utilize the engineered features as reference values for our CNN analysis. Similar to the work reported by Sivchenko et al[22,23] and Sivchenko[24], we aim at characterizing the flocculation by focusing on the overall structure of the textual flocculation patterns instead of isolating and calculating features for individual flocs.

3.1.1 | Model based on floc area and perimeter (Area&Perim)

The first model based on traditional image analysis is based on performing multinomial logistic regression models based on estimates of area and perimeter of the overall image texture of a given image. The area is an estimate of the biomass in each sample image, and by perimeter, we mean the accumulated sum of perimeters around the individual floc structures as given by a connected component analysis. First, we separated the biomass from the background making use of so-called morphological reconstruction, Vincent[26], where the marker image used in this process is obtained by erosion using a disk (5 pixels wide) as structural element. The resulting image was then thresholded using Otsu's method,[27] followed by a filter that removes tiny isolated foreground islands. Specifically, objects each with an area less than 0.1% of the total image area were removed. The total area covered by flocs was then estimated by counting the remaining foreground pixels (gaps within the flocs do not contribute to the area calculation). With regard to the procedure for estimating the accumulated number of perimeters of the flocculation structures, we performed contour detection using the algorithm described in Suzuki and Be.[28] We include contours to the second level in order to include both outer and inner boundaries in the perimeter calculation. Finally, the area and perimeter estimates from our training set are used as inputs to a multinomial logistic regression model using the Limited-memory- Broyden-Fletcher-Goldfarb-Shanno (LBFGS) solver in sklearn[29] to handle the multinomial loss. Please note that we utilize the detected areas and perimeters values as reference for the interpretation of CNN1, CNN2, and CNN3 results.

3.1.2 | Characterize floc texture using fractal analysis (Fractal)

The second reference model is based on the fact that several studies (see, e.g., Smoczyński et al.[18]) have shown that the flocs typically exhibit a self-similar structure and therefore can be characterized by its fractal dimension. The fractal dimension of an image is typically estimated using the box-counting method,[30] which calculates the so-called Minkowski or box-counting dimension. The box counting method operates by overlaying an image with grids of varying scales, thereby covering the image with a scale-dependent number of boxes. For each scale of the grid, one counts the number of grid boxes n_{boxes} which include any foreground pixels of the image. In case the image is characterized by a self-similar structure, one can then estimate the fractal dimension D from the following formula describing the expected dependency between the scale or box length L and the number of boxes n_{boxes}

$$ln[n_{boxes}] = D \times ln[L] + constant \tag{1}$$

More generally, one can use the box counting numbers as image features representing the scaling behavior of the binary textual pattern. This especially becomes meaningful in case that the curves are not well-approximated by the above formula. Following this approach, we have chosen to perform Principal Component Analysis (PCA) on the set of box counting numbers and then use the scores of the first two PCA components as input features to a multinomial logistic regression model. With reference to Equation (1), we select the box lengths of our grids such that they are equidistant on a log scale. In addition, we have experimented with using both the raw number of counts as well as the log number of counts as input values to the PCA.

3.2 | Convolutional neural networks

Three different convolutional neural network architectures, denoted as CNN1, CNN2, and CNN3, were developed and evaluated during this study. An overview over the three architectures is given in Figure 3. All three architectures share the common part of convolutional layers 1 and 2, denoted as Conv1 and Conv2. Subsequently, the networks differ as described in subsections below. The convolutional layers allow the network to operate as a set of convolutional filter kernels extracting image features from the images. During training, one attempts to adapt the weights of these kernels as to identify features that can discriminate the different flocculation stages.

3.2.1 | First convolutional layer (Conv1)

Gray-scale images were used as inputs to the CNN; hence, the input to the first convolutional layer is a matrix $\mathbf{X}$ of size $H \times W$ where H denotes the height of the image and W the width. The two-dimensional convolution denoted as Conv1 can be understood as a concatenation of U convolution kernels of size $K \times K$ represented by the tensor $\mathcal{K}_1$ and applied to the input matrix $\mathbf{X}$, resulting in an output tensor $\mathcal{Y}$ of size $H' \times W' \times U$ with $H' = H - K + 1$ and $W' = W - K + 1$. The convolution operation is shown element-wise in Equation (2).

$$\begin{aligned} \mathcal{Y}(h',w',u) &= \sum_{i=1}^{K}\sum_{j=1}^{K} \mathcal{K}_1(i,j,u)\mathbf{X}(h_i,w_j) \\ h_i = h' - 1 + i \quad w_j &= w' - 1 + j \quad h' \in \{1,2 \ldots H'\} \quad w' \in \{1,2 \ldots W'\} \quad u \in \{1,2 \ldots U\} \end{aligned} \tag{2}$$

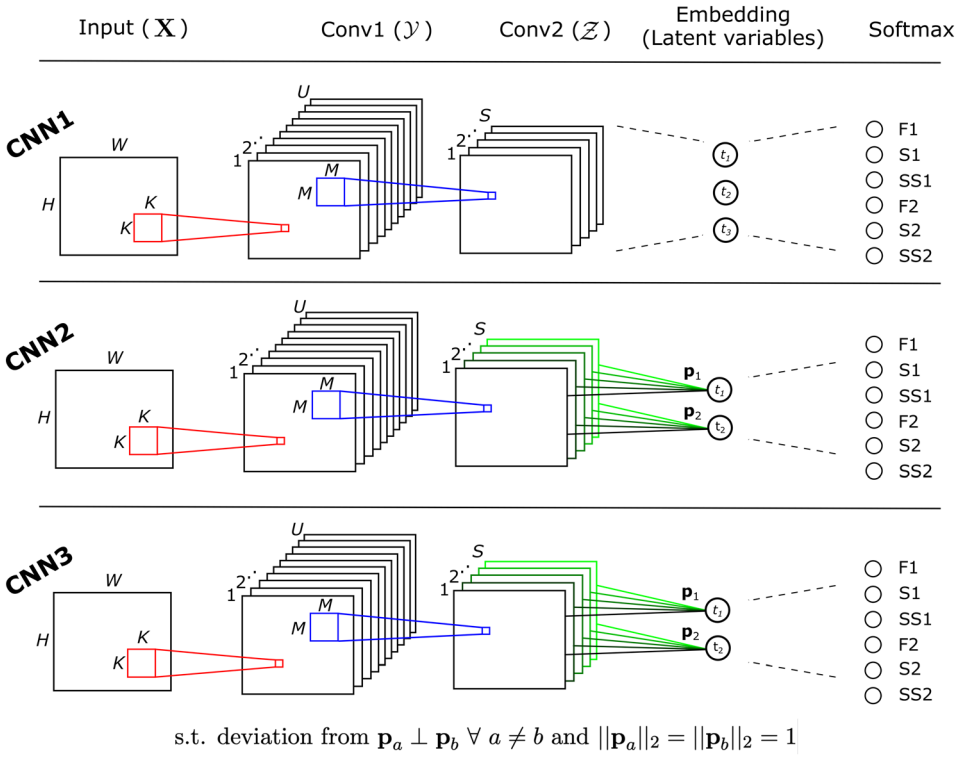

FIGURE 3 Network architectures of CNN1, CNN2, and CNN3 (from top to bottom). Dashed lines indicate fully connected layers. Shades of green are used to illustrate global average pooling. CNN2 and CNN3 differ due to the additional regularisation term involving $\mathbf{P}$. The number of neurons in the embedding layer indicates the number of latent variables R and t_r denotes scores in the latent variable representation. CNN, convolutional neural network.

The number of convolution kernels, U, also represents the number of applied filters, that is, the number of stacked rectangles for Conv1 as illustrated in Figure 3. We highlight that Equation (2) corresponds to a stride of one and that no padding was used (which can generally be assumed during this study).

3.2.2 | Second convolutional layer (Conv2)

The second convolutional layer (Conv2) conceptually performs a three-dimensional (3D) convolution, because it most importantly has the role of combining features across the U filter outputs (channels) from the first layer. However, the 3D convolution kernel size in the last dimension is equal to the number of output channels, and therefore, the result of the 3D convolution is still a 2D feature map for each 3D kernel. For this reason, the operation can mathematically also be seen as a weighted sum of U 2D-convolutions. With a kernel size M and S output channels, the Conv2 layer can be represented by a tensor $\mathcal{K}_2$ of size $M \times M \times U \times S$ applied to the tensor $\mathcal{Y}$. This results in a tensor $\mathcal{Z}$ of size $H'' = H' - M + 1$ and $W'' = W' - M + 1$. The convolution operation is described in Equation (3) and illustrated in Figure 3.

$$\begin{aligned} \mathcal{Z}(h', w', s) &= \sum_{u=1}^{U} \sum_{i=1}^{M} \sum_{j=1}^{M} \mathcal{K}_2(i, j, u, s)\, \mathcal{Y}(h_i, w_j, u) \\ h_i = h' - 1 + i \quad w_j &= w' - 1 + j \quad h' \in \{1, 2 \ldots H''\} \quad w' \in \{1, 2 \ldots W''\} \quad s \in \{1, 2 \ldots S\} \end{aligned} \tag{3}$$

3.2.3 | Fully connected embedding (CNN1)

We consider three possible ways of connecting the Conv2 layer to the embedding layer. First, we consider the case of full connectivity as illustrated for CNN1 in Figure 3A. Here, the feature maps contained in $\mathcal{Z}$ are vectorized to obtain a one-dimensional representation as shown in Equation (4). This vector is then fully connected to each of the R nodes, acting as LVs, in the embedding layer using linear activations; see Equation (5).

$$\mathcal{Z}^{H'' \times W'' \times S} \rightarrow \mathbf{z}^{1 \times H'' W'' S} \tag{4}$$

$$\mathbf{t} = \mathbf{z}\mathbf{P} + \boldsymbol{\beta}_0 \tag{5}$$

Here, S denotes the number of feature maps of dimensionality H'' times W'' generated by Conv2. The matrix $\mathbf{P} \in \mathbb{R}^{H'' W'' S \times R}$ contains the loading weights, which, when applied to $\mathbf{z}$, yield LV scores $\mathbf{t} \in \mathbb{R}^{1 \times R}$ in the embedding layer. $\boldsymbol{\beta}_0 \in \mathbb{R}^{1 \times R}$ denotes learned bias values for these scores. With the full connectivity between the Conv2 and the embedding layers, we have $R(H''W''S + 1)$ parameters to estimate (including one bias for each node in the embedding layer). The R output values of the embedding layer–acting as LVs scores–serve as a dimension-reduced representation of the input image. In other words, we let t_r denote the score of a single image for the r-th node (or LV) in the embedding layer as illustrated in the "Embedding" column of Figure 3.

3.2.4 | Partially connected embedding (CNN2)

Here, we essentially reduce each feature map of size H'' times W'' to a single value by constraining all weights from each feature map to each latent node to be identical (equivalent to using the average value of all pixels in each feature map). This implies that we only have one connection between each feature map and a node in the embedded layer. This structural regularization exploits the spatial invariance that characterizes the classification task and helps avoid overfitting. A similar approach was first suggested in Lin et al,[31] and the technique is generally denoted as global average pooling. We let t_r denote the output score of a single image for the r-th LV in the embedding layer as illustrated in the "Embedding" column of Figure 3B. Hereby, t_r is calculated as described in Equation (9).

$$\mathcal{Z}^{H''\times W''\times S}\rightarrow \mathbf{Z}^{H''W''\times S} \tag{6}$$

$$\mathbf{V}=\mathbf{ZP}+\boldsymbol{\beta}_0 \tag{7}$$

$$\mathbf{P}=[\mathbf{p}_1|\mathbf{p}_2|\ldots|\mathbf{p}_R] \tag{8}$$

$$t_r=\frac{\sum_{i=1}^{H''W''}v_{i,r}}{H''W''}\qquad r\in\{1\ldots R\} \tag{9}$$

Here, $\mathbf{V}\in\mathbb{R}^{H''W''\times R}$ are activation maps based on weighted feature maps with the loading weights given by $\mathbf{P}\in\mathbb{R}^{S\times R}$ (elaborated further in Section 3.3). The bias terms $\boldsymbol{\beta}_0\in\mathbb{R}^{1\times R}$ and $\mathbf{t}\in\mathbb{R}^{1\times R}$ are of similar dimension as described for CNN1 (see Section 3.2.3). Please note that the bias vector $\boldsymbol{\beta}_0$ in Equation (7) is added to all rows of $\mathbf{ZP}$.

3.2.5 | Partially connected embedding with regularization (CNN3)

Here, we again apply the global average pooling regularization as described in Section 3.2.4, but, in addition, we attempt to impose orthonormality on the column vectors of $\mathbf{P}$ (Equation (10)) by adding an extra regularization term as described in Section 3.2.7. The LV scores for a single image, $\mathbf{t}\in\mathbb{R}^{1\times R}$, are calculated similar as described for CNN2 (see Equation (9)).

$$\mathbf{p}_a\perp\mathbf{p}_b\ \forall\ a\neq b\quad\text{and}\quad ||\mathbf{p}_a||_2=||\mathbf{p}_b||_2=1 \tag{10}$$

3.2.6 | Softmax layer

The class predictions of CNN1, CNN2, and CNN3 are obtained by application of the softmax function to the result of the fully connected layer[32] as illustrated in Figure 3 using dashed lines.

3.2.7 | Loss function

The loss $\mathcal{L}$ for a batch of N images is calculated using average cross entropy as described in Equations (11)–(13). Hereby, $\hat{p}_{c,n}$ represents the softmax output, that is, the estimated probability of an image n to belong to class c and $p_{c,n}$ holds the one-hot encoded class belongings (ground truth). Please note that the loss function for CNN3 contains an extra term as shown in Equation (13) in order to penalize deviations from orthonormality. All CNNs were trained using backpropagation with Adam optimization.[33]

$$l_n=-\sum_{c=1}^{C}p_{c,n}ln\hat{p}_{c,n} \tag{11}$$

$$\mathcal{L}_{CNN1,CNN2}=\frac{\sum_{n=1}^{N}l_n}{N} \tag{12}$$

$$\mathcal{L}_{CNN3}=\frac{\sum_{n=1}^{N}l_n}{N}+||\mathbf{P}^T\mathbf{P}-\mathbf{I}||_F \tag{13}$$

3.2.8 | Activation functions and regularization

The rectifier (ReLU) activation function is applied to Conv1 and Conv2 layers. This applies to all three network architectures, namely, CNN1, CNN2, and CNN3. In addition, dropout with a probability of 0.3 was applied to the activation results of Conv1 and Conv2 to avoid overfitting.

3.3 | Activation mapping

To visualize the impact of S feature maps on the positioning of scores in the LV representation, activation maps were calculated. These maps represent weighted feature maps based on the network's parameters and were generated for each LV. By using R activation maps for a single image, we can identify which image features are responsible for a particular position in the R dimensional LV score space. This approach can help with interpreting the learned representation. Further details about the activation maps for CNN1, CNN2, and CNN3 are outlined in the following sections.

3.3.1 | CNN1

Let $\mathbf{Z}_{h'',w''} \in \mathbb{R}^{N\times S}$ represent the values of the S feature maps at pixel h'', w'' for N training instances/images and $\mathbf{P}_{h'',w''} \in \mathbb{R}^{S\times R}$ be the weights for the corresponding R LVs at pixel h'', w'', then activations for the N images $\mathbf{V}_{h'',w''} \in \mathbb{R}^{N\times R}$ at pixel h'', w'' can be calculated as described in Equation (14). This is a direct consequence of Equation (5).

$$\mathbf{V}_{h'',w''} = \mathbf{Z}_{h'',w''}\mathbf{P}_{h'',w''} + \frac{\boldsymbol{\beta}_0}{H''W''} \tag{14}$$

Summation in Equation (14) is performed row-wise as previously described for Equation (7).

3.3.2 | CNN2 and CNN3

For CNN2 and CNN3, Equation (14) simplifies yielding Equation (15). This is a direct consequence of Equations (7)–(9).

$$\mathbf{V}_{h'',w''} = \mathbf{Z}_{h'',w''}\mathbf{P} + \boldsymbol{\beta}_0 \tag{15}$$

Notice that the number of weights in $\mathbf{P}$ have been reduced due to application of global average pooling. In other words, for a single LV, all pixels within a given feature map are weighted similarly by application of only one weight (see Section 3.2.4). In contrast, CNN1 weights the pixels of the feature maps individually due to the fully connected layer.

3.4 | Rotation of LV space

With the model obtained after training, it is of interest to investigate the LV space, that is, to evaluate if meaningful patterns can be extracted from the learned representation. If this is the case, it is also of interest to link such patterns to distinct and interpretable activation patterns as calculated in Section 3.3. It is overall unlikely that activation patterns of the LVs resemble clearly distinguishable image features, such as area and perimeter. However, the question arises if one can rotate the LV space $\mathbf{T}$ in order to increase interpretability of the obtained activation patterns in $\mathbf{V}$, assuming that rotation of the score space and the loading weights $\mathbf{P}$ will preserve the feature maps for a given set of N images. The rotation of LV space representations is known in the domain of chemometrics and is, for example, applied when performing multivariate curve resolution.[34,35] The two theorems below formally introduce the rotation of the embedded LV space representations of the CNNs. The corresponding proofs can be found in Appendices A.1 and A2.

Theorem 1. The LV space representation is rotational ambiguous. Rotated activation maps $\tilde{\mathbf{V}}$ can be obtained via multiplication with an invertible rotation matrix $\boldsymbol{\theta}$.

Theorem 2. Rotation of LV scores $\mathbf{T}$ is equivalent to rotation of activation maps $\mathbf{V}$.

3.4.1 | Rotation matrices

In this study, the following rotation matrices $\boldsymbol{\theta}$ were applied, where $\rho \in [0,2\pi]$ is the rotation angle.

$$\boldsymbol{\theta}_{CNN2/CNN3} = \begin{bmatrix} cos(\rho) & -sin(\rho) \\ sin(\rho) & cos(\rho) \end{bmatrix} \tag{16}$$

$$\boldsymbol{\theta}_{CNN1} = \begin{bmatrix} cos(\rho) & -sin(\rho) & 0 \\ sin(\rho) & cos(\rho) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Equation (16) shows a rotation matrix $\boldsymbol{\theta} \in \mathbb{R}^{2\times 2}$ corresponding to rotation of scores around the center of a two-dimensional coordinate system. Equation (17) shows a rotation matrix $\boldsymbol{\theta} \in \mathbb{R}^{3\times 3}$ which results in scores invariant with respect to the third dimension, thereby not having an effect on the third LV projections/scores. In this study, the rotation angle ρ was chosen, such that $\tilde{t}_1$ scores indicated maximal correlation with reference perimeters and $\tilde{t}_2$ scores with reference areas (see Equation (18)). By doing so, we expect to obtain best possible aligned rotations of CNN1, CNN2, and CNN3, therefore allowing for direct comparison of the learned representations. Please note that reference areas and perimeters were obtained through the Area&Perim method.

$$\rho = \text{argmax}[corr(\tilde{t}_1, d) + corr(\tilde{t}_2, A)] \tag{18}$$

3.5 | Data augmentation

In order for the neural network models to learn a meaningful feature representation that can be used to characterize the diversity of the floc images, it is important that the training set is sufficiently representative of the distribution to be learned. In the case of images, a common way to extend the coverage is by slight modification of the existing training

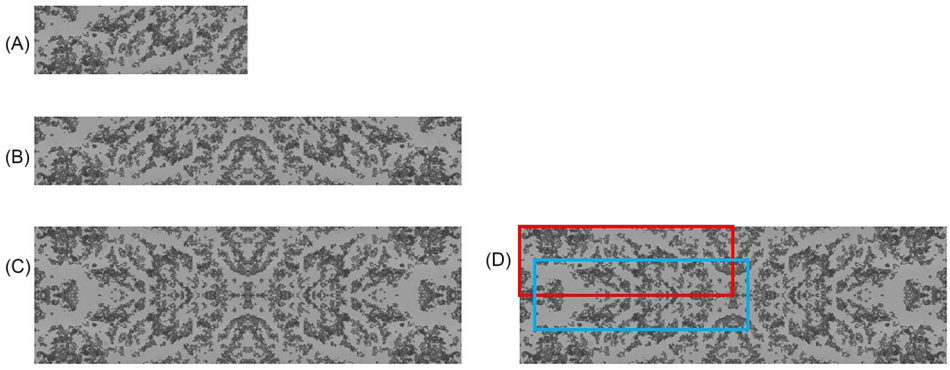

FIGURE 4 Data augmentation by horizontal and vertical shifts: an original image (A) is extended by horizontal flipping (B) followed by vertical flipping (C). Within the extended image, we generate new images by vertical and horizontal shifts (D)

examples, e.g. by scale, shift and/or rotation of the images.[36] In our case, the use of shifts and rotations is meaningful as we could as well have recorded the images at different orientations and positions of the the microscope relative to the sample. We have specifically extended the training set by using combination of horizontal and vertical shifts as illustrated in Figure 4.

3.6 | Model selection

The data set was split into a train and a test set in order to evaluate model selection for Area&Perim, Fractal, CNN1, CNN2, and CNN3. Hereby, data were separated batch-wise, meaning that all images from one batch would end up either in the train or the test set. When training (deep) neural networks, one is typically interested in having a third data set, namely, the validation set, that is, to tune model parameters, such as learning rate, dropout rate, and convolutional layer parameters. However, due to small sample size, this was not feasible in this study. Because the main scope of our experimental design was to investigate whether the CNNs could learn a meaningful feature representation–as opposed to concentrating on achieving an optimal classification performance–the number of convolutional layers and other hyperparameters were selected from manually trying different numbers (a rough grid search) and inspecting the performance on the training set. In the course of training the CNNs, there was a simultaneous assessment of the activation maps and the representations that were learned. Particular attention was given to the visual examination of rotated activation maps to ascertain if the LVs provide clear correlations with area and perimeter. Due to the lack of a specific validation set, the insights gained from this process were instrumental in establishing the stopping criteria for the training procedures, specifically in deciding which epoch to select for the final model, ensuring that the results produced the most meaningful representation and activation maps. We want to underline that the test set was used to compare the different methods, that is, the Area&Perim, Fractal as well as CNN1, CNN2, and CNN3.

The train set contained 100 images which correspond to samples measured using the microscope. In addition, the train set contained 4100 images obtained through data augmentation (see Section 3.5). Because our data augmentation does not assure true variability under real conditions, no image augmentation was performed for the test set, resulting in 18 pure test images. It is noteworthy that the image distribution across the six different process states was the same for the train and the test set in order to facilitate optimal generalization of the trained models.

3.7 | Software

Python version 3.6.9 was used throughout this study. Manually extracted image features, such as area and perimeter, were extracted using OpenCV version 4.1.1. CNN modeling was performed using Pytorch version 1.9.1+cu102 utilizing a single GeForce GTX 1080ti GPU with 11 GB RAM. Scikit Learn version 0.21.3 was utilized for PCA and logistic regression analysis.

4 | RESULTS

4.1 | Analysis using area and perimeter estimates (Area&Perim)

Floc areas and perimeters were determined as described in Section 3.1.1. It should be emphasized that extracted areas and perimeters were calculated for entire images, therefore representing snapshots of the thermodynamic states of the samples taken. This approach stands in contrast to estimating area and perimeter features at the individual floc level, which would provide a distribution of areas and perimeters for a given image.[9]

A scatter plot of the detected areas versus perimeters is shown in Figure 5A, and selected images are visualized in Figure 5C, where perimeters of detected flocs are indicated by cyan blue and yellow color. Specifically, contours of the first level are shown in cyan blue, and contours of the second level are shown in yellow.

By inspecting the seven selected images it appears meaningful that images with large and dense flocs (3) possess less perimeter than images with abundant, loose and evenly distributed biomass (1). On the other hand, it is noteworthy that these large flocs (3) are characterized by a large area relatively seen to the detected total perimeter. Consequently, images with many patches of small and loosely distributed biomass (1) yield low areas relative to the detected

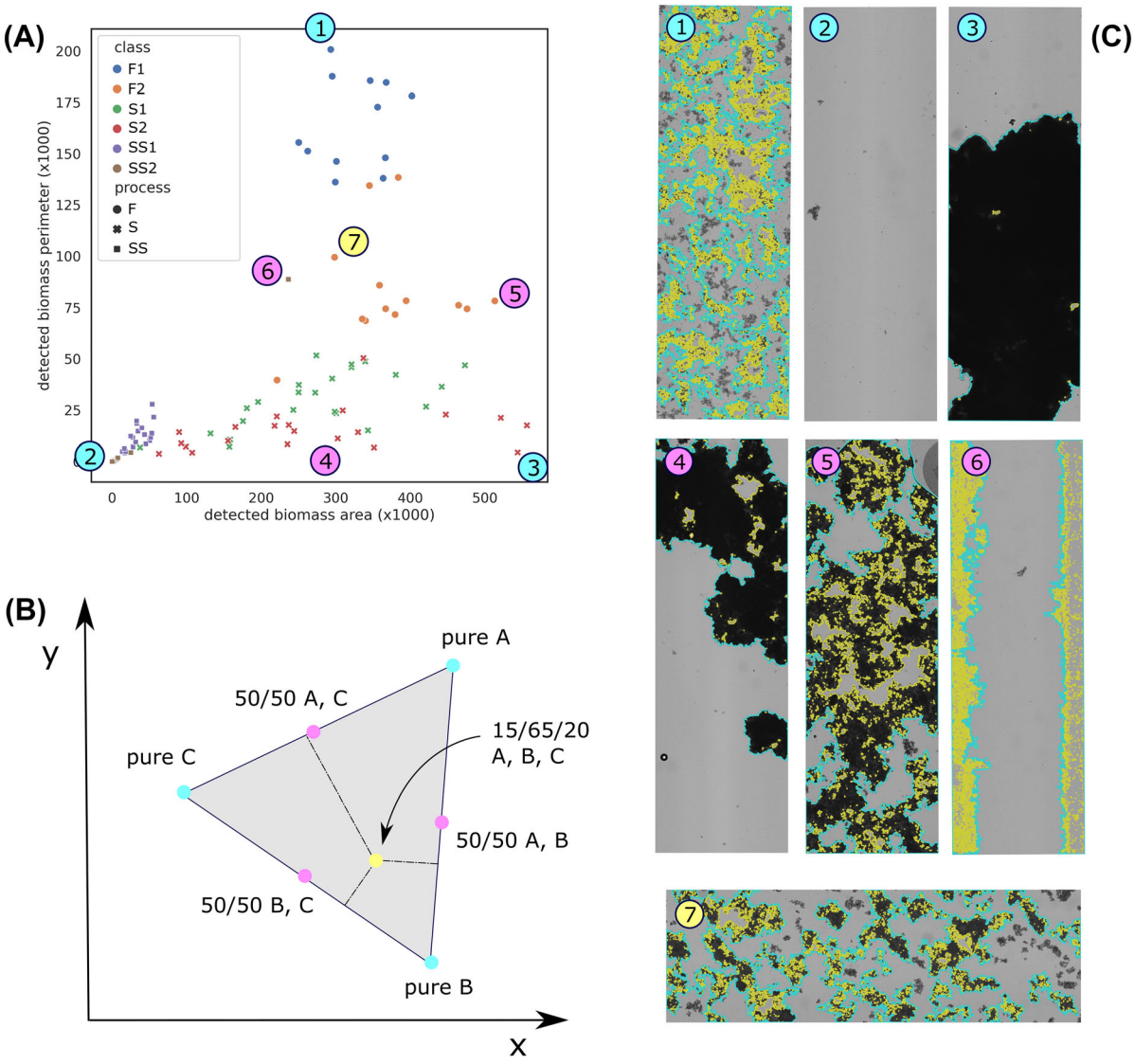

FIGURE 5 (A) Biomass area versus perimeter for all training set images (in pixels). (B) A schematic illustration of a ternary system, where pure states—described by the corners of the triangle—are characterized by three distinct sets of coordinates in x and y. (C) Some representative images indicating archetypical characteristics are shown (1–3), alongside with images which represent mixtures of these extreme states (4–7). In these images, cyan blue color represents outer perimeters, while yellow color indicates inner perimeters of the detected biomass. Note that the contour detection algorithm failed for image 6 as biomass perimeter was detected although pure background was present.

perimeter. However, the area and perimeter detection algorithm failed when being applied to images, where (almost) no biomass or particles were present (6). In this case, image processing by Otsu thresholding failed due to low dynamic range of the image.

Overall, we can state that Feed, Separator and Separator stream images appear well separated in the area versus perimeter scatter plot (Figure 5A). However, Separator 1 and 2 data points overlap significantly. Furthermore, the following observations can be made. Images from the Separator stream stages of the process, especially Separator stream 2, indicate close to zero perimeters and areas, which is meaningful because there should not be any biomass left in these images as the biomass should have been entirely removed at this step. It is also noteworthy that Separator stream 1 contains more remaining biomass, indicated by slightly higher detected perimeters and areas. Please note that there is one obvious outlier from the Separator stream 2 group indicating relatively high perimeter and area. This sample refers to image (6) in Figure 5C and contained almost only gray background. As mentioned previously, the Otsu thresholding method failed in this case.

Images from the Feed stages contain the largest perimeter values. This is especially pronounced for Feed 1 as these samples contained unflocculated biomass, while a lot of this biomass has already been removed before obtaining Feed 2 images; hence, less perimeter is observed for Feed 2.

Separator images indicate rather low perimeters as flocs are dense and compact. However, Separator flocs spread through the entire range of possible area counts. This can be explained by sampling uncertainty, that is, the uncertainty of capturing the entire floc within the microscope acquisition window. Additionally, flocs can vary in compression, meaning that some of these flocs are darker and therefore denser than others. This will also affect the overall area detected within a Separator image. It is noteworthy that this textural difference cannot be detected by the Area&Perim method.

In Figure 5A, all observations to a very good approximation appear confined to a triangular region. This triangular configuration is a well-documented phenomenon in physical chemistry and is commonly recognized as a ternary system.[37] Explained briefly, in a ternary system, where the relative compositions of three constituents are considered, the sum of their fractions always equals 100%, a characteristic known as closure. This attribute implies that when two relative concentrations are known within a sample, the third can be deduced accordingly. Figure 5B provides an illustration of a ternary system, where pure states (or constituents) are denoted by blue points at the vertices, 50/50 mixtures of connected states are represented by pink points, and an example of a 15/65/20 mixture is indicated by a yellow point. Consequently, all conceivable mixtures of the three states must fall within the confines of the ternary system, delineated by the gray area in Figure 5B. Note that a ternary system can be defined within the euclidean space, for example, defined by two axes (x,y). In this paper, area and perimeter as well as LV scores may serve for (x,y).

It is evident that all captured images can be effectively modeled within a ternary system by utilizing the identified floc areas and perimeters. In other words, the three corners of the triangle symbolize three distinct pure thermodynamic states recorded during the flocculation process (as observed in cyan blue points in Figure 5A). Specifically, image (2) corresponds to the pure background (Separator stream), image (3) represents the image with substantial and area-intensive flocs (Separator), and image (1) characterizes the image with evenly distributed biomass (Feed). Consequently, all other acquired images can be understood as combinations of these three fundamental characteristics. The pink observations (images 4 and 5) signify mixtures of two of these three states, while the yellow observation (7) approximates an image that closely aligns with a 33/33/33 mixture. Noticeable, acquired images cannot be characterized by areas and perimeters above certain values due to well defined acquisition settings defined by image size and resolution. This ensures that all possible images from the same downstream process remain confined to the boundaries of the ternary system.

Based on areas and perimeters as inputs to a logistic regression model (see Section 3.1.1), we obtained classification performances as shown in Table 1. It appears that the Area&Perim method classified 73% of the training set images correctly, whereas only 61% of the test set images were classified correctly. This result confirms the visual impression of major class overlaps as it can be seen in the area versus perimeter scatter plot depicted in Figure 5A. The overlap is in particular visible for Separator 1 and Separator 2, as well as for Separator stream 1 and Separator stream 2 images.

4.2 | Fractal analysis

As an alternative to the Area&Perim method, fractal analysis was applied to the flocculation process images as described in Section 3.1.2. The results are visualized in Figure 6A, where the box count has been plotted against the box size. In order to support interpretation of the data, standardized fractal counts have been plotted in Figure 6B (mean centering and scaling to unit variance). Noticeably, when observing the standardized plots, Feed counts indicate a

TABLE 1 Classification accuracies for the three image analysis methodologies.

	Area&Perim	Fractal	CNN1	CNN2	CNN3
Training set	0.73	0.73	0.94	0.87	0.84
Test set	0.61	0.67	0.67	0.79	0.83

Note: Please note that the three CNNs were trained on augmented data as described in Section 3.5; however, accuracies reported in this table were calculated solely based on original images.

Abbreviation: CNN, convolutional neural network.

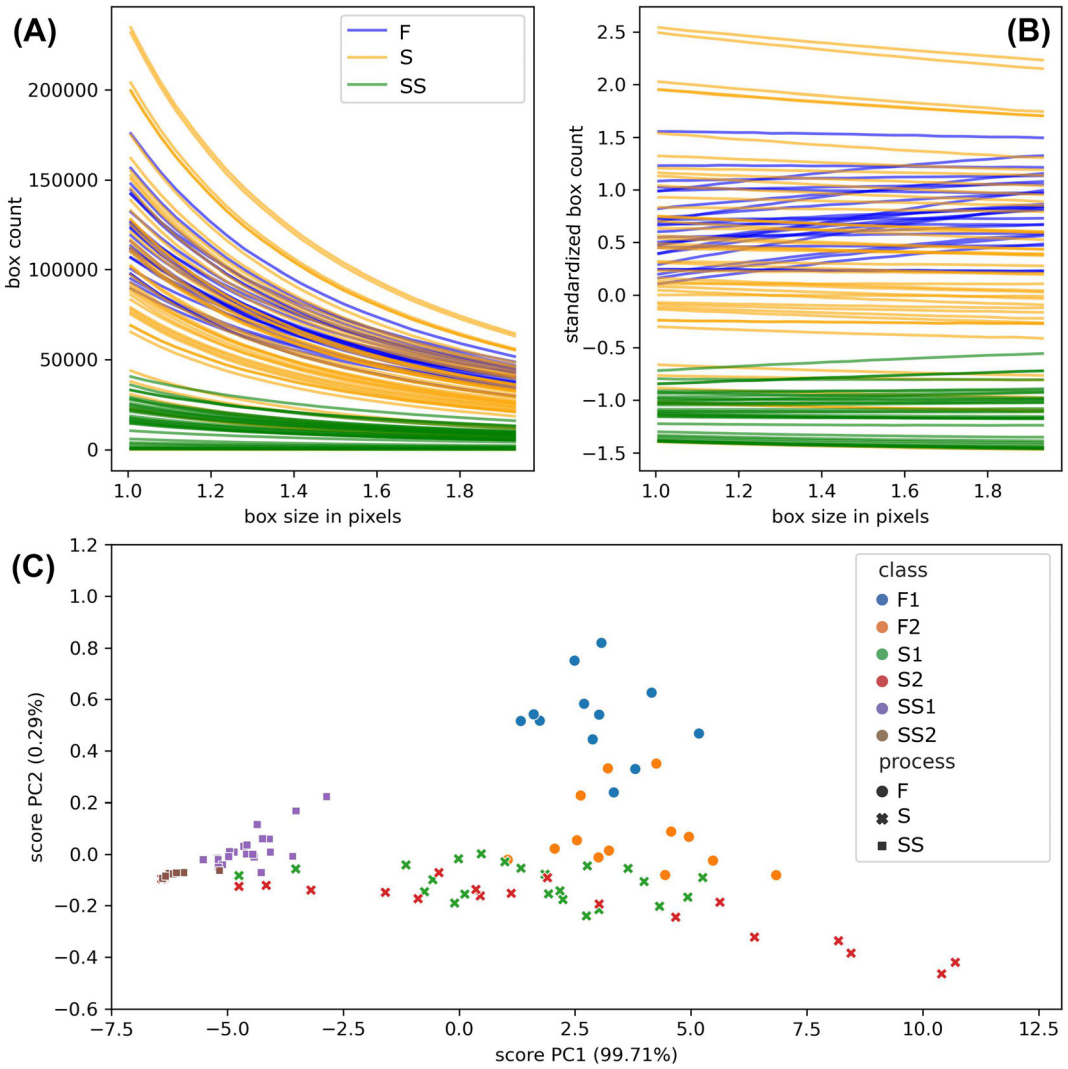

FIGURE 6 (A) Fractal spectra extracted for all images of the training data set. (B) Standardized fractal spectra. (C) PCA score plot of the standardized fractal spectra. Note that the F, S, and SS classes have been combined in A and B for illustration purposes. Legend in A is also valid for B.

positive slope, while Separator images indicate the opposite. Separator counts appear dispersed over the entire plotting range. In order to recognize hidden patterns in the data, PCA was applied to the standardized counts. Two components explained 99.99% of the variation in the counts. Although the second component only explains 0.3% of the variation, we observe that it plays a crucial role in distinguishing the Feed images from the Separator images. The respective scores obtained by the PCA decomposition are displayed in Figure 6C. One can clearly observe that the scores are distributed in a triangular shape, again, describing a ternary system (see Figure 5B for comparison). Again, the three major process states appear well separated in the LV space, while visual discrimination within process states, for example, between Separator 1 and Separator 2 scores remains difficult.

Classification using logistic regression with the PCA scores from Figure 6C as input variables resulted in 73% and 67% correctly classified images for the training and test set, respectively. The test set performance is comparable with the Area&Perim. It is noteworthy that the fractal analysis did not fail when being applied to images containing pure background (see image (6) in Figure 5C). A thorough interpretation of the loading vectors of the PCA in relation to Equation (1) is given in Appendix A3.

4.3 | CNN

4.3.1 | Network parameters

A general overview over the three CNN architectures is provided in Figure 3. As described in Section 3.2, the three networks have a common part, that is, the two convolutional layers, namely, Conv1 and Conv2, while they differ with respect to how information is forwarded into the embedded LV space. The number of output channels for Conv1 and Conv2 were chosen to be $T = 10$ and $S = 5$, respectively, and the kernel size was $K = 5$ for both layers. For CNN1, three LVs, $R = 3$, were used, while only two LVs, $R = 2$, indicated better performance for CNN2 and CNN3.

4.3.2 | CNN1

The embedding layer of the first convolutional neural network (CNN1) is fully connected with the second convolutional layer, meaning that each pixel z_i from the five output channels of Conv2 is associated with a loading weight parameter $p_{i,r}$ (see Equation (5)). The entire network comprises a total of 2.3 million parameters. The training progress in terms of classification accuracy evaluated over 30 epochs is depicted in Figure 7 for training and test set, respectively. Although training accuracy approached close to 100% at around eight epochs, the large gap between training and test set performance indicates significant overfitting. The model classification accuracy estimated on the test set was on par with results from fractal analysis as it can be seen in Table 1.

The LV space representation of all training set observations, which includes augmented images, underwent a rotation as outlined in Section 3.4. The outcome, demonstrating the highest correlation between $\tilde{t}_1$ and $\tilde{t}_2$ scores with reference perimeters and areas, can be observed in the top-left subplot of Figure 7. While a triangular representation of the process images can be identified, the ternary system does not appear as clear as reported for the Area&Perim and Fractal Analysis method (see Figures 5 and 6). Moreover, the triangle remains positioned somewhat tilted, suggesting that a clear affiliation of the LVs with area and perimeter is difficult. To investigate which characteristics are represented by

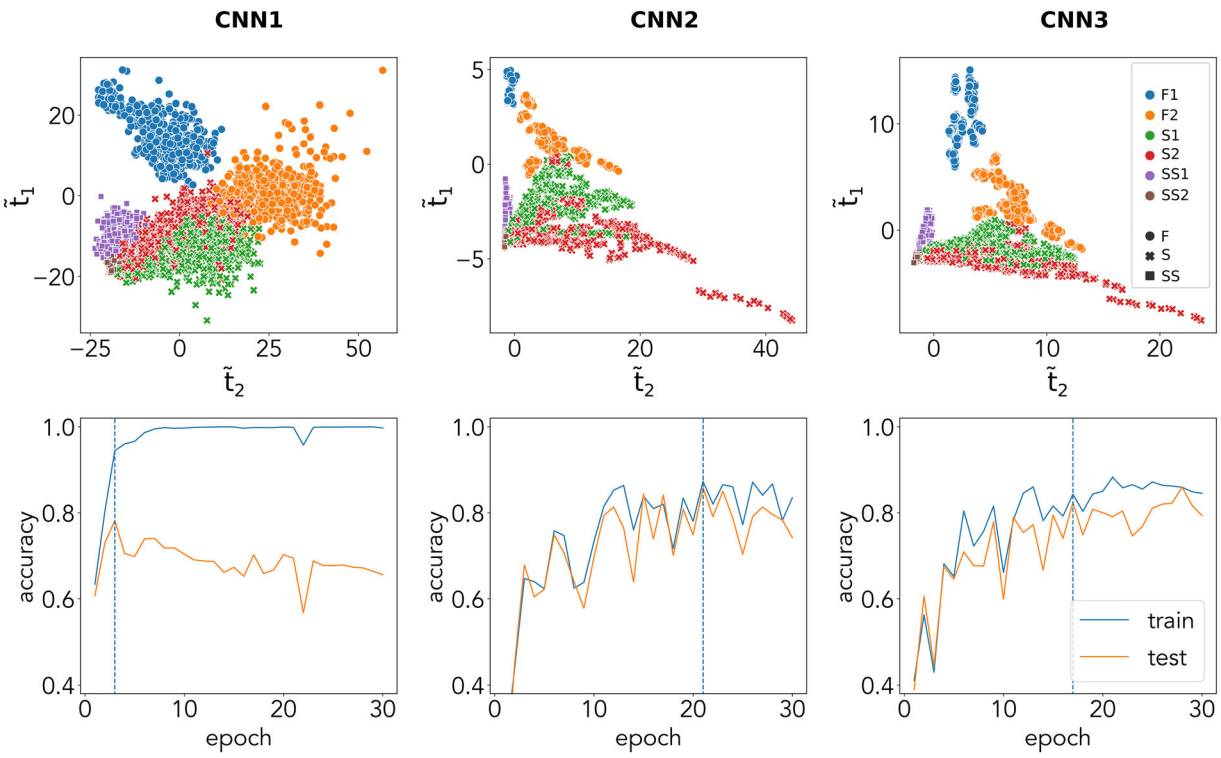

FIGURE 7 Rotated latent variable space representations of augmented training data (top) and learning progress (bottom) for CNN1, CNN2, and CNN3 (from left to right). The number of epochs required to obtain the representations and reported classification performances are indicated by dashed lines. Legends are valid across all three columns of the figure. CNN, convolutional neural network.

LVs 1, 2, and 3, the rotated activation maps $\tilde{\mathbf{V}}$ have been calculated as described in Section 3.3.1 and plotted for six images as shown in Figure 8. Only activation maps for LV1 and LV2 are shown as the visualization of the third LV did not add valuable information. When examining Figure 8, it appears to be challenging to discern specific patterns solely associated with either area or perimeter. Given the large number of parameters of the fully connected layer, it is no surprise that we obtain somewhat noisy activation maps. In order to provide a quantitative evaluation of the interpretability of the LVs, correlation coefficients of rotated $\tilde{t}_1$ and $\tilde{t}_2$ scores with reference perimeters and areas are provided in Table 2. Although correlations of $\tilde{t}_1$ and $\tilde{t}_2$ scores with perimeters and areas are, respectively, high, it is noteworthy that significant cross-correlation is present, that is, it turns out that $\tilde{t}_1$ scores are correlated, both, to reference perimeters and areas as indicated by correlation coefficients of $corr(\tilde{t}_1,d) = 0.85$ and $corr(\tilde{t}_1,A) = 0.48$. A similar effect can be observed for $\tilde{t}_2$ scores, that is, correlations are $corr(\tilde{t}_2,d) = 0.37$ and $corr(\tilde{t}_2,A) = 0.79$, respectively. These numbers support the conclusion that area and perimeter related features mix across the two activation maps, corresponding to activations in two LVs.

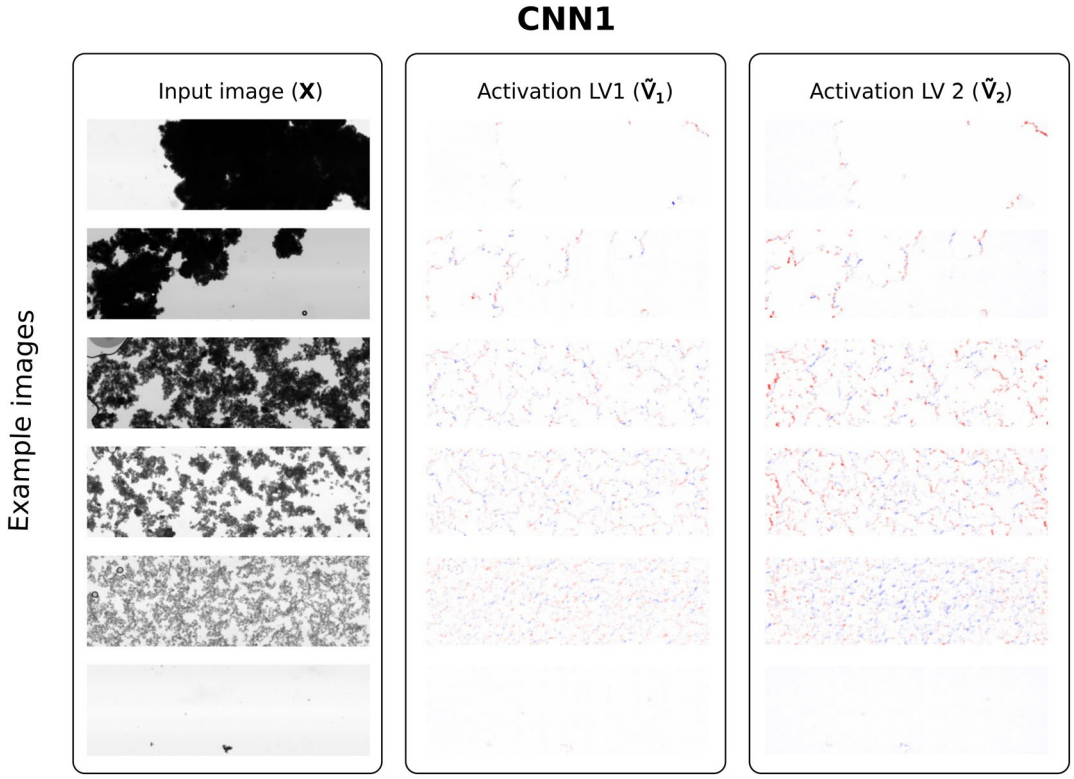

FIGURE 8 Six input images (left column) and corresponding rotated activation maps $\tilde{\mathbf{V}}$ for LV1 (middle column) and LV2 (right column). Red denotes positive and blue color negative activations. Interpretation of the activations patterns appears to be difficult for CNN1. CNN, convolutional neural network.

TABLE 2 Correlations between rotated latent variable scores and floc areas (A) and perimeters (d) as extracted by the Area&Perim method.

	CNN1	CNN2	CNN3
$corr(\tilde{t}_1,d)$	0.85	0.92	0.90
$corr(\tilde{t}_2,A)$	0.79	0.71	0.87
$corr(\tilde{t}_1,A)$	0.48	0.41	0.23
$corr(\tilde{t}_2,d)$	0.37	−0.07	0.22

Note: Noticeably, CNN2 indicates the highest correlation of $\tilde{t}_1$ scores with reference perimeters, but CNN3 indicates a better overall performance when also looking at the correlation coefficient between $\tilde{t}_2$ scores and reference areas.
Abbreviation: CNN, convolutional neural network.

4.3.3 | CNN2

Whereas CNN1 had full connectivity from the output of the Conv2-layer to the embedding layer, the CNN2 architecture makes use of global average pooling as outlined in Section 3.2.4, so that we only have one loading weight parameter to learn between each feature map of the Conv2-layer and each node in the embedding layer. The training progress in terms of classification accuracy evaluated over 30 epochs is depicted in Figure 7 for both training and test set. The training set performance did not improve significantly after the 21st epoch, where it achieved approximately 87% accuracy in correctly classifying images, while the test set performance stood at 79%. The gap between training and test set performances is much smaller when compared with the CNN1 model indicating less overfitting and therefore a more robust model. The superior classification performance is due to the application of global average pooling, which reduces the number of network parameters from 2.3 million in CNN1 to 1545 in CNN2. Classification performance is also superior when compared with Area&Perim and Fractal analysis logistic regression results (see Table 1).

The axes of the LV space were again rotated so that the resulting $\tilde{t}_1$ and $\tilde{t}_2$ scores indicated maximal correlation with reference perimeters and areas, respectively (see Section 3.4). Overall, the learned representation appears somewhat similar to the ones obtained through the Area&Perim and Fractal analysis as shown in Figures 5 and 6. Compared with the CNN1 case, a triangular shape is now more apparent. It is noteworthy that some Separator 2 images, that is, the red data points with high $\tilde{t}_2$ scores around 30–40 are located far off its cluster.

CNN2 activations were computed for six images and can be seen in Figure 9. It clearly appears that perimeters/edges are activated as indicated by red color in LV1. On the other hand, biomass area is detected positively as indicated by red color in LV2. Although this seems overall promising, negative activations corresponding to background or densely flocculated biomass area are also apparent in LV1. This indicates that CNN2 is not entirely able to separate the detection of perimeter and area features between LV1 and LV2. This is supported by the fact that correlation coefficients of $\tilde{t}_1$ scores with reference perimeters and areas indicate cross-correlation, that is, $corr(\tilde{t}_1, d) = 0.92$ and $corr(\tilde{t}_1, A) = 0.41$, respectively (see Table 2).

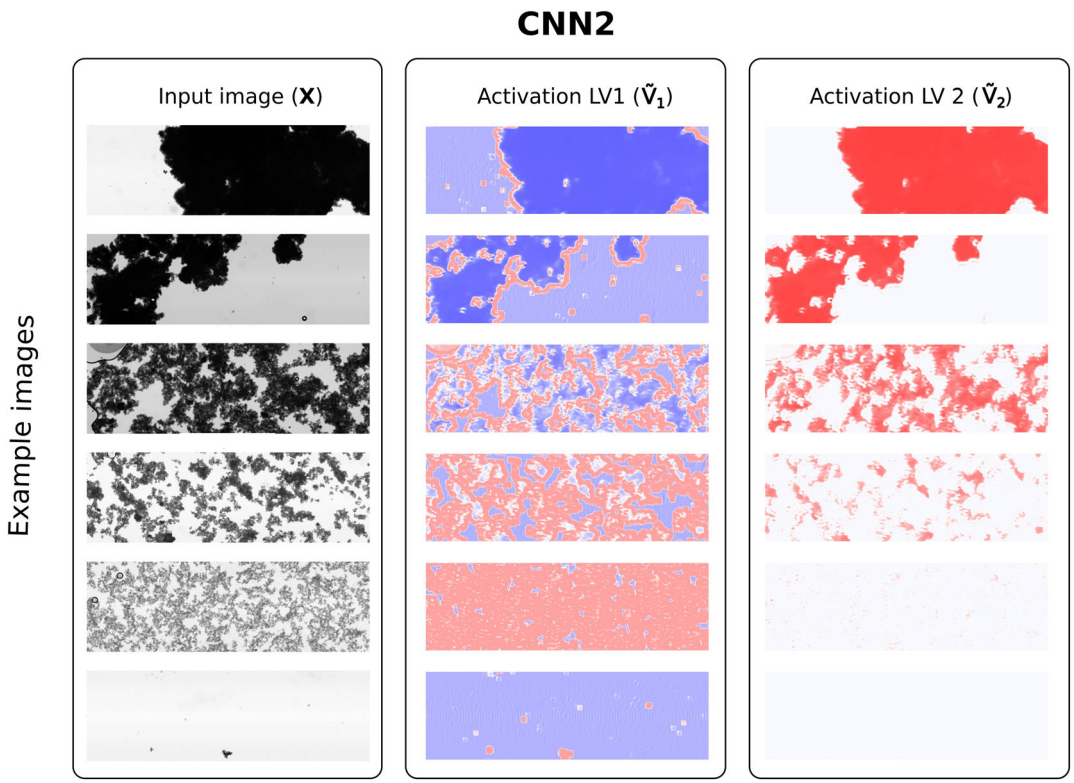

FIGURE 9 Six input images (left column) and corresponding rotated activation maps $\tilde{\mathbf{V}}$ for LV1 (middle column) and LV2 (right column). Red denotes positive and blue color negative activations. CNN2 is able to detect floc perimeters and areas, but is also paying attention to the image background. CNN, convolutional neural network.

Noticeably, CNN2 learned to pay attention to texture related image features instead of solely detecting biomass areas and perimeters. This can be seen in the last two images of Figure 9. It appears that more areas are activated in LV1 as biomass texture becomes more loose. On the other hand, this has the consequence that in LV2 less area is detected for loose biomass as activations here are close to zero (white color). Hence, we can conclude that LV2 focuses on densely flocculated biomass, while LV1 pays attention to biomass edges and loosely textured biomass.

Because the overall sums of the activations in LV1 and LV2 determine the score position in the embedded LV space (see Equation (9)), we can conclude that images with high amount of biomass edges/perimeter and loose texture will result in high $\tilde{t}_1$ scores, while images with large and dense flocs will lead to high $\tilde{t}_2$ scores.

4.3.4 | CNN3

For CNN3, we added extra regularization penalizing deviations from orthonormal loading vectors $\mathbf{P}$, thereby affecting the learned representation (see Section 3.2.5). As a result of this, only CNN3 featured approximately orthonormal loading vectors as further elaborated in Appendix A4. The training progress in terms of classification accuracy evaluated over 30 epochs is depicted in Figure 7 for the training and test set. The training and test curve tend to separate after 17 epochs indicating the risk of overfitting for the next epochs. Furthermore, inspection of the learned representation suggested a meaningful result at this stage. We therefore used the model based on 17 epochs giving classification accuracies of 84% and 83% for training and test set, respectively. The gap between training and test set performance is similar with respect to the CNN2 model as shown in Figure 7.

Once again, the two-dimensional embedding ($R = 2$) as shown in Figure 7 was rotated such that $\tilde{t}_1$ and $\tilde{t}_2$ scores indicated maximal correlation with reference perimeters and areas, respectively. Similar to CNN2, the learned representation of CNN3 resembles the ternary system well. In fact, the triangular shape is more apparent for CNN3 compared with CNN1 and CNN2. Again, some Separator 2 data points are positioned far off with very high $\tilde{t}_2$ and very low $\tilde{t}_1$ scores. Looking at the raw images for these data points, it turns out that very large biomass flocs covering almost the entire acquisition window were detected. If flocs become to large to be captured within the acquisition window of the microscope, CNN3 could not detect any perimeters, therefore leading to lower $\tilde{t}_1$ scores.

The rotated activation maps of CNN3 offer very good and intuitive interpretability; that is, obtained activation patterns for LV1 and LV2 correspond to perimeter and area characteristics, respectively, as it can be seen in Figure 10. It is clear that LV1 and LV2 activation patterns clearly distinguish between detected biomass edges/loose texture and (densely) flocculated biomass area. Although slight negative activations of biomass area (light blue color) are still visible in LV1 activation maps, the effect is overall very minor in comparison with the strong positive activations (red color) of the biomass edges/perimeters. Contrasting CNN2, it is also visible that the two latent dimensions in CNN3 are better at separating edges versus biomass areas, as it can clearly be observed for the last image in Figure 10, where even very small flocs of biomass are contoured by LV1 and vice-versa detected as biomass area in LV2. The good separation of perimeter and area features is also supported by correlation coefficients which indicate low cross-correlation as it can be seen in Table 2. Hereby, correlation coefficients of $\tilde{t}_1$ and $\tilde{t}_2$ scores with reference biomass perimeters and areas are $corr(\tilde{t}_1, d) = 0.90$ and $corr(\tilde{t}_2, A) = 0.87$, respectively, marking highest overall correlation with the reference values, while at the same time indicating least cross-correlation due to low values of $corr(\tilde{t}_1, A)$ and $corr(\tilde{t}_2, d)$. This separation is likely to be due to the encouraged orthonormality. Overall, CNN3 exhibits superior classification performance when compared with the other models. However, the superior classification performance of CNN3 compared with CNN2 remains insignificant (see Table 1). Calculation of the mean test set accuracy ± 2 times the standard deviation across the 10 last epochs yields [0.694, 0.877] and [0.752, 0.853] for CNN2 and CNN3, respectively. The large variance of the test set accuracies during the last 10 epochs can be explained by the small sample size of the test set, that is, only 18 images were available; hence, misclassification of a single image has a major impact on the performance measure. Nonetheless, we demonstrate better performance, including better generalization ability of CNN3, due to more meaningful activation mapping and superior interpretability of the projections in the LV space.

5 | DISCUSSION

As discussed above, we have been considering and comparing the use of neural network classifiers to the traditional machine learning approach of combining engineered features with classification models. Although specifically dealing

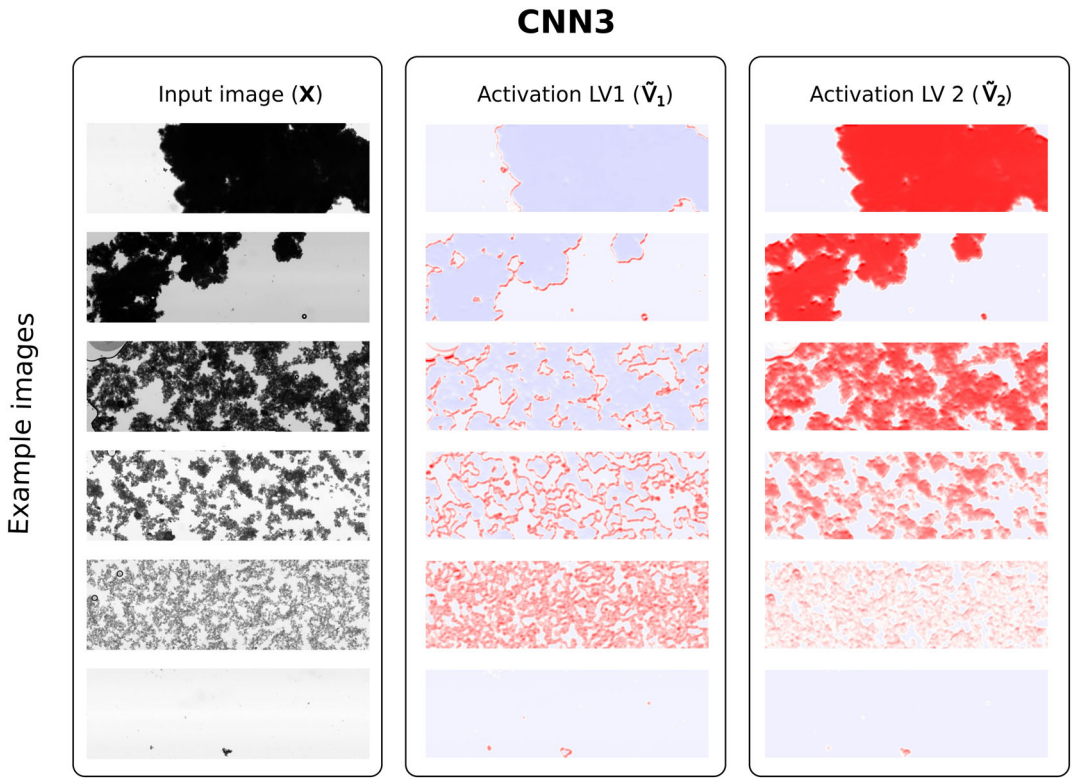

FIGURE 10 Six input images (left column) and corresponding rotated activation maps $\tilde{\mathbf{V}}$ for LV1 (middle column) and LV2 (right column). It is appears clear that CNN3 detects edges/perimeters in LV1 and densely flocculated biomass in LV2. CNN3 offers best interpretability of the activation maps. CNN, convolutional neural network; LV, latent variable.

with classifying images of flocs from the downstream of a fermentation process, the more general focus is to explore the value of representation learning in the embedding layer of a neural network. Applying an end-to-end (E2E) learning approach only focuses on the overall target such as classification performance and does not automatically pay specific attention to the quality of the internal representation being learned, although the architecture inherently includes a bottleneck/embedding layer. This is illustrated by our CNN1 model, which does not succeed in obtaining a representation, where the inherent features learned correspond to what domain knowledge tells us is of interest. In addition the generalization performance is poor due to many parameters in the network relative to the number of training examples available. With CNN2, we limit the amount of parameters by using global average pooling, which significantly improves the generalization performance. We can also observe that the activation maps corresponding to the individual LVs (nodes) of the embedding layer now are correlated with area and perimeter features, which we expect are relevant features. With CNN3, we modify our loss function to encourage orthonormality on the loading weight vectors in order to achieve maximal independence between the activation patterns of the LVs. Conceptually, this resembles a (projection to latent structures) PLS strategy, since we thereby seek to obtain loading vectors $\mathbf{P}$ and corresponding scores $\mathbf{T}$ that carry maximal information about the output classes ("Y") while being mutually orthogonal.

Furthermore, we showed that the learned LV space representations within CNNs are rotational ambiguous (see Section 3.4). The property of rotational ambiguity is well-known for matrix decomposition methods[34,35,38] and is derived and applied to CNN1, CNN2, and CNN3 representations in this paper. Hereby, loading weights $\mathbf{P}$, activation maps $\mathbf{V}$, and LV scores $\mathbf{T}$ can be rotated, such that activation patterns in $\tilde{\mathbf{V}}$ make most intuitive sense. To perform this rotation based on an objective approach, we find the optimal rotation angle that maximizes the sum of the correlation coefficients between scores and the reference perimeters and areas. The results reported in Figure 7 were obtained using rotation based on this objective. In order to illustrate the impact of the rotation angle, Figure 11 showcases rotated activation maps $\tilde{\mathbf{V}}$, rotated scores $\tilde{\mathbf{T}}$ as well as rotated loading vectors $\tilde{\mathbf{P}}$ for two rotation angles (CNN3) applied to a single image denoted as "original". The left side of the figure shows the result of the original obtained model, while the right side depicts results when applying a rotation angle of 212°, resulting in activation maps which are easier to comprehend. Specifically, the activation map corresponding to LV1 contains negative activations (blue) for floc area,

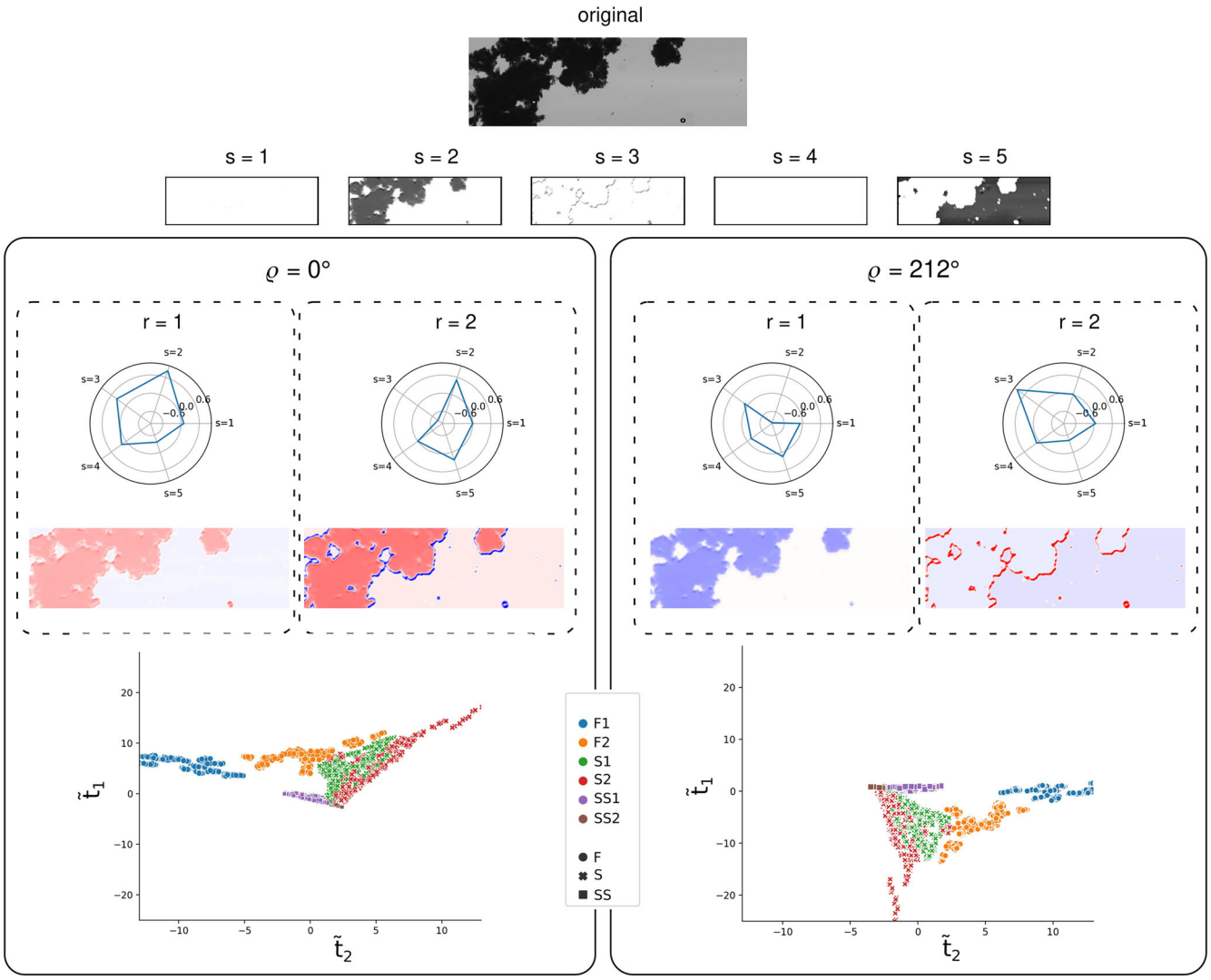

FIGURE 11 From top to bottom: example input image, corresponding feature maps ($\mathbf{Z}$), rotated loading weights $\tilde{\mathbf{P}}$, rotated activation maps $\tilde{\mathbf{V}}$ and rotated scores $\tilde{\mathbf{T}}$. Left side: learned representation without rotation; right side: learned representation after rotation offering better interpretability. All plots refer to outputs generated by CNN3. CNN, convolutional neural network.

resulting in more negative scores $\tilde{t}_2$ when large and dense flocs are present. On the other hand, the activation map for LV2 indicates positive activations (red) resembling edges, therefore yielding high $\tilde{t}_1$ scores when many small flocs and loosely distributed biomass appears in the image. Here, the rotation angle of 212° was arbitrarily chosen to illustrate the effect when rotating the representation to a meaningful position. While potentially many of these orientations could be interpretable, all the reported results for CNN1, CNN2, and CNN3 (see Figure 7) were obtained by selecting rotation angles indicating maximal positive correlations of the LV scores with reference perimeters and areas.

With regard to the addressed problem, that is, the characterization of flocculation processes, we wanted to demonstrate how the learned LV representation can be utilized to monitor industrial processes. Given satisfactory classification results and superior interpretability of the activation patterns of CNN3, we can assume that the learned representation reflects process variability across the six flocculation steps in a satisfactory manner. Figure 12 shows trajectories for six production batches across the flocculation process states Feed 1, Separator 1, and Separator stream 1 on the left subplot and Feed 2, Separator 2, and Separator stream 2 on the right-hand side subplot. One can observe that five of the six batches progressed similarly trough the six process states (black lines), while one batch, depicted using red lines, behaved abnormal in the Separator 1 and Separator 2 phase. Due to the fact that high $\tilde{t}_1$ scores represent large abundance of biomass edges or loose texture and high $\tilde{t}_2$ scores large amounts of flocculated biomass area, we can conclude that the batch was abnormal in the sense that it contained less flocculated biomass in the Separator 1 stage. On

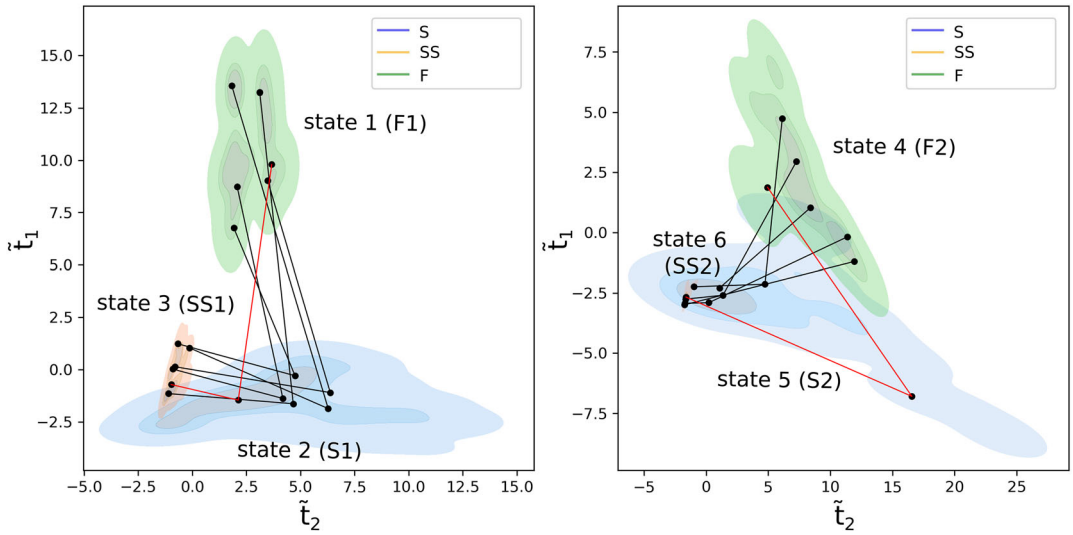

FIGURE 12 Kernel density plot of the rotated scores $\tilde{t}_1$ and $\tilde{t}_2$ visualized for the first three flocculation process states (left) and the last three steps (right). Five batches depicting ordinary progression throughout the process are visualized in black, while a single batch illustrating outlying behavior is shown in red. The results are obtained using CNN3. CNN, convolutional neural network.

the other hand, the same batch indicated abnormally high amounts of densely flocculated biomass in the Separator 2 phase as $\tilde{t}_2$ scores are very high and $\tilde{t}_1$ scores low. One reason might be that the batch in question failed to flocculate well in the first flocculation step (assessed by images taken at Separator 1), while at the same time leading to more flocculated material in the second flocculation step of the process (assessed by images taken at Separator 2). Given the example in Figure 12, it suggests that the learned representation can be utilized to track flocculation process performance. Procedures could potentially be automated, employing further test statistics, which can be used to generate alarms when flocculation process performance appears to be out of control.

6 | CONCLUSION

In this study, we investigated the use of CNN as an alternative approach to traditional domain-knowledge guided image processing for analyzing microscopic images of a flocculation process. Our goal was to compare the capabilities of these two approaches in distinguishing between six different processing steps. In order to deal with the challenge that a CNN approach may sacrifice interpretability of detected image features, we introduced a bottleneck layer and specific regularization techniques to ensure independence of information in the embedding layer (LV space). In addition, our study demonstrates that the learned LV space representation exhibits rotational ambiguity, enabling superior interpretability of the detected features, such as the anticipated area and perimeter characteristics, upon rotation. Our final CNN model achieved better classification performance compared with the two traditional image analysis methods, despite being trained on a limited dataset of only 118 images. This was made possible by incorporating global average pooling, which significantly reduces the number of weights in the neural network. Our findings suggest potential application of this methodology for monitoring of industrial flocculation processes, thereby contributing to improved efficiency and productivity. Furthermore, the presented results may also have implications for various fields that rely on image analysis techniques.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/cem.3534.

ORCID

Andreas Baum https://orcid.org/0000-0003-1552-0220

REFERENCES

1. Khan MB, Lee XY, Nisar H, Ng CA, Yeap KH, Malik AS. Digital image processing and analysis for activated sludge wastewater treatment. *Adv Exp Med Biol*. 2015;823:227-248. doi:10.1007/978-3-319-10984-8_13
2. da Motta M, Pons MN, Roche N. Study of filamentous bacteria by image analysis and relation with settleability. *Water Sci Technol J Int Assoc Water Pollut Res*. 2002;46:363-369.
3. Mesquita DP, Dias O, Elias RAV, Amaral AL, Ferreira EC. Dilution and magnification effects on image analysis applications in activated sludge characterization. *Microsc Microanal*. 2010;16:561-568. doi:10.1017/S1431927610093785
4. Chu CP, Lee DJ, Tay JH. Bilevel thresholding of floc images. *J Colloid Interface Sci*. 2004;273:483-489. doi:10.1016/j.jcis.2004.01.002
5. Li DH, Ganczarczyk J. Fractal geometry of particle aggregates generated in water and wastewater treatment processes. *Environ Sci Technol*. 1989;23:1385-1389. doi:10.1021/es00069a009
6. François RJ. Strength of aluminium hydroxide flocs. *Water Res*. 1987;21:1023-1030. doi:10.1016/0043-1354(87)90023-6
7. Leentvaar J, Rebhun M. Strength of ferric hydroxide flocs. *Water Res*. 1983;17:895-902. doi:10.1016/0043-1354(83)90163-X
8. Yeung AKC, Pelton R. Micromechanics: a new approach to studying the strength and breakup of flocs. *J Colloid Interface Sci*. 1996;184:579-585. doi:10.1006/jcis.1996.0654
9. Jarvis P, Jefferson B, Gregory J, Parsons SA. A review of floc strength and breakage. *Water Res*. 2005;39:3121-3137. doi:10.1016/j.watres.2005.05.022
10. Chakraborti RK, Gardner KH, Atkinson JF, Benschoten JEV. Changes in fractal dimension during aggregation. *Water Res*. 2003;37:873-883. doi:10.1016/S0043-1354(02)00379-2
11. Li T, Zhu Z, Wang D, Yao C, Tang H. Characterization of floc size, strength, and structure under various coagulation mechanisms. *Powder Technol*. 2006;168:104-110. doi:10.1016/j.powtec.2006.07.003
12. Xu Y, Chen T, Cui F, Shi W. Effect of reused alum-humic-flocs on coagulation performance and floc characteristics formed by aluminum salt coagulants in humic-acid water. *Chem Eng J*. 2016;287:225-232. doi:10.1016/j.cej.2015.11.017
13. Amaral AL, Ferreira EC. Activated sludge monitoring of a wastewater treatment plant using image analysis and partial least squares regression. *Anal Chim Acta*. 2005;544:246-253. doi:10.1016/j.aca.2004.12.061
14. Mesquita DP, Dias O, Amaral AL, Ferreira EC. Monitoring of activated sludge settling ability through image analysis: validation on full-scale wastewater treatment plants. *Bioprocess Biosyst Eng*. 2009;32:361-367. doi:10.1007/s00449-008-0255-z
15. Mesquita DP, Dias O, Dias AMA, Amaral AL, Ferreira EC. Correlation between sludge settling ability and image analysis information using partial least squares. *Anal Chim Acta*. 2009;642:94-101. doi:10.1016/j.aca.2009.03.023
16. Jenné R, Banadda EN, Gins G, et al. Use of image analysis for sludge characterisation: studying the relation between floc shape and sludge settleability. *Water Sci Technol*. 2006;54:167-174. doi:10.2166/wst.2006.384
17. Arelli A, Luccarini L, Madoni P. Application of image analysis in activated sludge to evaluate correlations between settleability and features of flocs and filamentous species. *Water Sci Technol*. 2009;59:2029-2036. doi:10.2166/wst.2009.119
18. Smoczyński L, Ratnaweera H, Kosobucka M, Smoczyński M. Image analysis of sludge aggregates. *Sep Purif Technol*. 2014;122:412-420. doi:10.1016/j.seppur.2013.09.030
19. Khan MB, Nisar H, Ng CA. Image processing and analysis of phase-contrast microscopic images of activated sludge to monitor the wastewater treatment plants. *IEEE Access*. 2018;6:1778-1791. doi:10.1109/ACCESS.2017.2780249
20. Molina MA, Pérez CAA, Leiva CA. Characterization of filamentous flocs to predict sedimentation parameters using image analysis. *J Sensors*. 2020;2020:1-8. doi:10.1155/2020/5248509
21. Amaral AL, Mesquita DP, Ferreira EC. Automatic identification of activated sludge disturbances and assessment of operational parameters. *Chemosphere*. 2013;91:705-710. doi:10.1016/j.chemosphere.2012.12.066
22. Sivchenko N, Kvaal K, Ratnaweera H. Evaluation of image texture recognition techniques in application to wastewater coagulation. *Cogent Eng*. 2016;3:1206679. doi:10.1080/23311916.2016.1206679
23. Sivchenko N, Ratnaweera H, Kvaal K. Approbation of the texture analysis imaging technique in the wastewater treatment plant. *Cogent Eng*. 2017;4:1373416. doi:10.1080/23311916.2017.1373416
24. Sivchenko N. *Image Analysis in Coagulation Process Control*: Norwegian University of Life Sciences; 2018. 978-82-575-1484-6.
25. Yu R-F. On-line evaluating the SS removals for chemical coagulation using digital image analysis and artificial neural networks. *Int J Environ Sci Technol*. 2014;11:1817-1826. doi:10.1007/s13762-014-0657-1.
26. Vincent L. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Trans Image Proc*. 1993;2:176-201. doi:10.1109/83.217222
27. Otsu N. Threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern*. 1979;SMC-9:62-66. doi:10.1109/tsmc.1979.4310076
28. Suzuki S, Be KA. Topological structural analysis of digitized binary images by border following. *Comput Vision Graph Image Proc*. 1985;30:32-46. doi:10.1016/0734-189X(85)90016-7
29. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *J Mach Learn Res*. 2011;12(Oct):2825-2830.
30. Falconer K. Fractal and applications mathematical foundations geometry; 2003.
31. Lin M, Chen Q, Yan S. Network in network. arXiv 2013, arXiv:1312.4400. Available online: https://arxiv.org/abs/1312.4400; 2013.
32. Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y. What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th international conference on computer vision IEEE; 2009:2146-2153.
33. Kingma DP, Ba J. Adam: a method for stochastic optimization; 2014.

34. Abdollahi H, Tauler R. Uniqueness and rotation ambiguities in multivariate curve resolution methods. *Chemom Intell Lab Syst*. 2011; 108(2):100-111.
35. Vosough M, Mason C, Tauler R, Jalali-Heravi M, Maeder M. On rotational ambiguity in model-free analyses of multivariate data. *J Chemom: J Chemom Soc*. 2006;20(6-7):302-310.
36. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J Big Data*. 2019;6:60. doi:10.1186/s40537-019-0197-0
37. Porter DA, Easterling KE, Sherif MY. *Phase Transformations in Metals and Alloys*: CRC Press; 2021. doi:10.1201/9781003011804
38. Paatero P, Hopke PK, Song X-H, Ramadan Z. Understanding and controlling rotations in factor analytic models. *Chemom Intell Lab Syst*. 2002;60(1-2):253-264.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Baum A, Moiseyenko R, Glanville S, Martini Jørgensen T. Image-based characterization of flocculation processes through PLS inspired representation learning in convolutional neural networks. *Journal of Chemometrics*. 2024;e3534. doi:10.1002/cem.3534

APPENDIX A

A1 | Proof of theorem 1

Proof. Let $\mathbf{V}_{h'',w''} \in \mathbb{R}^{N\times R}$ be the values of the activation maps at pixel h'', w'' for N images/instances across R LVs, $\mathbf{Z}_{h'',w''} \in \mathbb{R}^{N\times S}$ be the values of the S feature maps at pixel h'', w'' for N images/instances and $\mathbf{P}_{h'',w''} \in \mathbb{R}^{S\times R}$ be the weight matrix at pixel h'', w'' (please note that $\mathbf{P}_{h'',w''}$ becomes $\mathbf{P}$ for CNN2 and CNN3 as described in Section 3.3.2 and used below). We obtain activation maps as described in Equations (14) and (15) and reformulate such that

$$\mathbf{V}^*_{h'',w''} = \mathbf{V}_{h'',w''} - \boldsymbol{\beta}_0 = \mathbf{Z}_{h'',w''}\mathbf{P} \tag{A1}$$

$$\mathbf{V}^*_{h'',w''}\mathbf{P}^\dagger\mathbf{P} = \mathbf{Z}_{h'',w''}\mathbf{P} \tag{A2}$$

$$\mathbf{Z}_{h'',w''} = \mathbf{V}^*_{h'',w''}\mathbf{P}^\dagger \tag{A3}$$

$\mathbf{P}^\dagger$ is the pseudo inverse of the weight matrix $\mathbf{P}$. If we re-substitute $\mathbf{V}^*_{h'',w''}$ we get

$$\mathbf{Z}_{h'',w''} = \mathbf{V}_{h'',w''}\mathbf{P}^\dagger - \boldsymbol{\beta}_0\mathbf{P}^\dagger \tag{A4}$$

We insert an invertible matrix $\boldsymbol{\theta} \in \mathbb{R}^{R\times R}$ with desirable rotation properties in order to transform the LV space representation as follows.

$$\mathbf{Z}_{h'',w''} = \mathbf{V}_{h'',w''}\boldsymbol{\theta}\boldsymbol{\theta}^{-1}\mathbf{P}^\dagger - \boldsymbol{\beta}_0\boldsymbol{\theta}\boldsymbol{\theta}^{-1}\mathbf{P}^\dagger \tag{A5}$$

We obtain rotated terms as shown in Equations (A6)–(A8).

$$\tilde{\mathbf{V}}_{h'',w''} = \mathbf{V}_{h'',w''}\boldsymbol{\theta} \tag{A6}$$

$$\tilde{\mathbf{P}}^{\dagger} = \boldsymbol{\theta}^{-1}\mathbf{P}^{\dagger} \tag{A7}$$

$$\tilde{\boldsymbol{\beta}}_0 = \boldsymbol{\beta}_0\boldsymbol{\theta} \tag{A8}$$

We can further show that

$$\tilde{\mathbf{V}}_{h'',w''} = \mathbf{Z}_{h'',w''}\mathbf{P}\boldsymbol{\theta} + \tilde{\boldsymbol{\beta}}_0 = \mathbf{Z}_{h'',w''}\tilde{\mathbf{P}} + \tilde{\boldsymbol{\beta}}_0 \tag{A9}$$

by rewriting Equation (A5) with rotated terms and re-arranging as shown below.

$$\mathbf{Z}_{h'',w''} = \tilde{\mathbf{V}}_{h'',w''}\tilde{\mathbf{P}}^{\dagger} - \tilde{\boldsymbol{\beta}}_0\tilde{\mathbf{P}}^{\dagger} \tag{A10}$$

$$\mathbf{Z}_{h'',w''} = (\tilde{\mathbf{V}}_{h'',w''} - \tilde{\boldsymbol{\beta}}_0)\tilde{\mathbf{P}}^{\dagger} \tag{A11}$$

$$\mathbf{Z}_{h'',w''}\tilde{\mathbf{P}} = (\tilde{\mathbf{V}}_{h'',w''} - \tilde{\boldsymbol{\beta}}_0)\tilde{\mathbf{P}}^{\dagger}\tilde{\mathbf{P}} \tag{A12}$$

$$\mathbf{Z}_{h'',w''}\tilde{\mathbf{P}} = \tilde{\mathbf{V}}_{h'',w''} - \tilde{\boldsymbol{\beta}}_0 \tag{A13}$$

$$\tilde{\mathbf{V}}_{h'',w''} = \mathbf{Z}_{h'',w''}\tilde{\mathbf{P}} + \tilde{\boldsymbol{\beta}}_0 \tag{A14}$$

Using Equation (A14) rotated activation maps can be computed. ■

A2 | Proof of theorem 2

Proof. Let $\mathbf{V} \in \mathbb{R}^{H''W'' \times R}$ be the activation maps across R LVs for a given input image $\mathbf{X}$ and $\boldsymbol{\theta} \in \mathbb{R}^{R \times R}$ be the rotation matrix. Following Equation (A6), we can define rotated activation maps $\tilde{\mathbf{V}} \in \mathbb{R}^{H''W' \times R}$ for a single image such that

$$\tilde{\mathbf{V}} = \mathbf{V}\boldsymbol{\theta} = \begin{bmatrix} \mathbf{v}_1\boldsymbol{\theta} \\ \mathbf{v}_2\boldsymbol{\theta} \\ \vdots \\ \mathbf{v}_{H''W''}\boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{v}}_1 \\ \tilde{\mathbf{v}}_2 \\ \vdots \\ \tilde{\mathbf{v}}_{H''W''} \end{bmatrix} \tag{A15}$$

Given Equation (9), it follows that

$$\tilde{\mathbf{t}} = \frac{\mathbf{v}_1\boldsymbol{\theta} + \mathbf{v}_2\boldsymbol{\theta} + \ldots + \mathbf{v}_{H''W''}\boldsymbol{\theta}}{H''W''} = \frac{\mathbf{v}_1 + \mathbf{v}_2 + \ldots + \mathbf{v}_{H''W''}}{H''W''}\boldsymbol{\theta} = \mathbf{t}\boldsymbol{\theta} \tag{A16}$$

Hereby, the rotated scores $\tilde{\mathbf{t}}$ can be calculated as the average pixel values in columns of $\tilde{\mathbf{V}}$. Alternatively, rotated scores $\tilde{\mathbf{t}}$ can be obtained directly via multiplication of the scores $\mathbf{t}$ with the rotation matrix $\boldsymbol{\theta}$. ■

A3 | Fractal analysis: Additional results

In Table A3, loading vectors of the two first principal components (PC) are depicted for three different standardization schemes. The loading vectors describe the weights that the PCs put on the box counts at each scale, starting from the smallest box size to the largest one. We observe that the first PC basically calculates an average box count for all scales corresponding to a kind of density estimator of the foreground pixels (area-like). The second PC can be seen as a filtered average of the overall slope of the box counts as a function of scale. If we perform a log transformation of the box counts before the PCA (with and without standardisation, i.e., centering and scaling), we obtain the loading vectors as shown in Table A3-b. Again, we observe the first PC extracting an average box count over the box scale, while the second PC estimates the slope of the curves/lines. In the light of equation 1 this slope acts as an estimate of the fractal dimension. Compared with the Area&Perim method, the perimeter measure is thus replaced by a measure of the self similarity of the floc structures.

TABLE A3 Loading vectors for the first two principal components when using (a) counts and standardisation, (b) log counts and standardisation, and (c) log counts and no standardisation.

(a) Loadings		(b) Loadings		(c) Loadings	
PCA 1	PCA 2	PCA 1	PCA 2	PCA 1	PCA 2
0.223	−0.36	−0.224	−0.36	−0.232	−0.37
0.223	−0.32	−0.224	−0.32	−0.231	−0.32
0.223	−0.29	−0.224	−0.29	−0.23	−0.29
0.224	−0.25	−0.224	−0.24	−0.23	−0.24
0.224	−0.22	−0.224	−0.21	-0.229	−0.21
0.224	−0.18	−0.224	−0.17	−0.227	−0.16
0.224	−0.14	−0.224	−0.14	−0.227	−0.13
0.224	−0.10	−0.224	−0.10	−0.226	−0.09
0.224	−0.06	−0.224	−0.04	−0.225	−0.04
0.224	−0.02	−0.224	−0.02	−0.224	−0.02
0.224	0.02	−0.224	0.02	−0.223	0.03
0.224	0.06	−0.224	0.06	−0.222	0.07
0.224	0.10	−0.224	0.09	-0.222	0.10
0.224	0.13	−0.224	0.12	−0.22	0.13
0.224	0.18	−0.224	0.15	−0.22	0.16
0.224	0.21	−0.224	0.22	−0.218	0.23
0.224	0.25	−0.224	0.24	−0.217	0.25
0.223	0.29	−0.224	0.30	−0.216	0.30
0.223	0.33	−0.224	0.34	−0.215	0.33
0.223	0.37	−0.224	0.38	−0.214	0.38

A4 | Orthonormality diagnostics

We inspect to which extend CNN1, CNN2, and CNN3 yield orthonormal loading weight vectors. The following results have been calculated at epochs 3, 21, and 17 for CNN1, CNN2, and CNN3, respectively.

A5 | CNN1

$$\mathbf{P}^T\mathbf{P} = \begin{bmatrix} 1.9436 & 0.2154 & 0.2314 \\ 0.2154 & 1.9213 & -0.5327 \\ 0.2314 & -0.5327 & 2.4314 \end{bmatrix} \tag{A17}$$

$$\mathbf{P}^T\mathbf{P} = \begin{bmatrix} 2.6727 & -1.4118 \\ -1.4118 & 13.3944 \end{bmatrix} \tag{A18}$$

$$\mathbf{P}^T\mathbf{P} = \begin{bmatrix} 0.9928 & 0.0207 \\ 0.0207 & 1.0112 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A19}$$

Hereby, a perfect orthonormal projection shall yield the result $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Looking at the respective inner products for CNN1, CNN2, and CNN3 above, this condition is approximately fulfilled for CNN3 only (see Equation (A19)) and can be explained by the added regularization term penalizing deviations from orthonormality (see Equation (13)).