A Novel Semi-Supervised Approach for Red Blood Cell Shape Analysis in Flow

Andrea Caputo[1,2][0009−0003−7788−8829]✉, Lars Kaestner[2,3][0000−0001−6796−9535], and Stephan Quint[1][0000−0003−4412−7559]

[1] Cysmic GmbH, Saarbrücken, Germany
andrea.caputo@cysmic.de
[2] Dynamics of fluids, Saarland University, Saarbrücken, Germany
anca00013@uni-saarland.de
[3] Theoretical Medicine and Biosciences, Saarland University, Homburg, Germany

Abstract. During microcapillary flow, red blood cells (RBCs) show a broad variability in shapes. Although widely studied from a physics point of view, there is no globally accepted reference for classes of RBCs in this condition. Previously, to classify them, different supervised learning (SL) models have been developed. Conversely, we proposed a novel approach, based on the semi-supervised learning (SSL) paradigm, to mostly rely on RBCs' intrinsic properties while minimizing any dependence on the labels. We developed a solid model, leveraging Cellpose-SAM (CPSAM) combined with two neural networks (NNs) for contour extraction and a variational autoencoder (VAE) as clusterer. To remain consistent with the SSL paradigm, we tailored our VAE taking inspiration from the Variational Deep Embedding (VaDE) approach. With the proposed method, we obtained 3 different groups of cells that express significant differences among them, achieving an accuracy of 94% on the labeled validation set.

Keywords: Erythrocytes · Microfluidic flow · Semi-supervised learning

1 Introduction and Background

The deformability of human RBCs is fundamental for them to carry out their functions, as during their entire lifespan they are exposed to flows inside the microvascular networks, where vessel diameters are similar to their size. During flow they express a broad variability in shapes due to the complex interaction between the RBC itself and the channel, depending on both intrinsic cell properties and external conditions [11]. Studying RBCs during flow is an essential way to diagnose several hematologic diseases, and although several studies have been carried out both experimentally and numerically [10], there is no general artificial intelligence (AI) tool to analyse them. In this study, we focus on the shape analysis of RBCs in microfluidic flow, and we aim to develop an SSL approach in order to minimize the influence of any human bias arising from cell labeling, a time-consuming and error-prone process. Nonetheless, several studies have been carried out in this sense, with a classical supervised learning (SL) approach. In

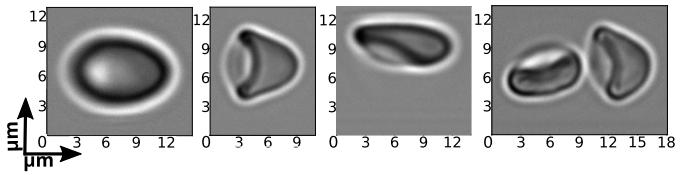

Fig. 1. Different cases of RBC morphologies during microcapillary flow - axes in μm. From left to right: *ball*, *croissant*, *slipper* and a *cell cluster*.

[4], the authors developed a *tolerant* SL model to classify cells in the two most known and accepted RBC categories: *slipper-shaped* cells and *croissant-shaped* ones [9]. This constitutes an important step forward for studying shapes according to intrinsic RBC properties, but the human bias is still present. Since we are solely interested in RBCs' intrinsic properties, our model primarily relies on cell shapes, limiting the use of labels. Therefore, we developed a mixture of SL models (CPSAM-NNs) to extract the cell contour, and a SSL model (VaDE) to cluster them, accounting for a single residual supervised loss contribution. As this is a novel approach, in our study we considered, for the training phase of this clustering head, a dataset containing healthy-shaped RBCs (Fig. 1).

2 Material and Methods

2.1 Model Architecture

Overall Pipeline
We can decompose our model into three different parts: 1) **Cellpose-SAM**: a fine-tuned version of this foundation model [8], which is devoted to extracting two pieces of information pixel-wise: the probability of belonging to a cell and the flow towards the cell center; 2) **Filter-Ranker Neural Networks**: the aim of these two NNs, which operate jointly, is to recognize, among different contours elaborated from CPSAM outputs, the right one; 3) **VaDE**: this is the head of the model, and its role is to correctly cluster RBCs, in the form of feature vectors of their contours, into meaningful groups. The general inference workflow is shown in Fig. 2. As our clustering method represents a novelty in the field of RBCs in flow, we will focus our attention on an in-depth description of it, while providing a more concise description of the aforementioned extraction method.

Contouring method The CPSAM [8] is the latest model from the Cellpose family to be released, and leverages, as a backbone, the Segment Anything Model (SAM) [6]. Thanks to its robustness, CPSAM is considered a foundation model in cell segmentation, able to perform well also in a zero-shot scenario. Unfortunately, our case study is different from any case on which CPSAM has been trained. For this reason, we implemented a fine-tuning pipeline for CPSAM, to let the model adapt to our particular domain, where noise and focus instability

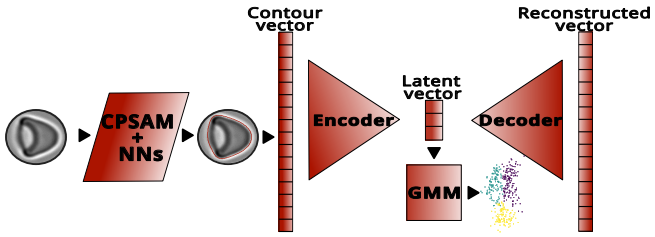

Fig. 2. Diagram of the model architecture. The inference proceeds as follows: 1) The input image is processed by the fine-tuned CPSAM and different contours are found - 2) Two NNs choose the best contour and a 22-dimensional contour vector is built - 3) The vector is fed into the VaDE-style encoder to be projected into the latent space (3D) - 4) The latent vector is passed to the GMM to assign the corresponding cluster.

make this task harder. The dataset we used for this fine-tuning process is composed of 259 different RBCs, taken under different focus conditions, along with the corresponding masks. Thanks to this process, our CPSAM version performs as well as the original version on the test set used in [8]. Due to the computational cost of a full inference of CPSAM, we consider, for our task, only its encoder, discarding the segmentation head. Thus, we take advantage of the intermediate output from CPSAM (i.e., the probability and the flow maps). We developed our segmentation head in two parallel processes, each using one of the CPSAM outputs. From the probability map, we implemented a GPU-powered k-means [7] to group pixels according to their probability; different contours are then considered as borders between different groups. From the flow map, we evaluate the divergence; thanks to the unique information it contains, we implemented a watershed algorithm [1] on it, obtaining a number of masks equal to the actual number of objects in the images. Once these two steps are performed, we discard any multiple-cell case and compute, along each contour, a set of features on the real cell images. To choose the right contour cell-wise, we use two simple NNs that work sequentially. The first one (i.e., *filter network*) is trained with a classical binary cross-entropy loss, while the second (i.e., *ranker network*) is trained to minimize a margin ranking loss. The former network has the role of discarding any contour that is unlikely to be the right one and passing to the *ranker network* only valid contours. The *ranker* then produces a proper ranking across the contours for each cell, and the best-ranked contour is selected. More details on their architectures are provided in Section S1 of the Supplementary Material.

VaDE To efficiently cluster cells into groups using an SSL approach, we designed, as the head of our model, a VAE architecture inspired by the VaDE approach [3]. The key strength of the standard VaDE lies in keeping the clustering totally unsupervised, while grouping instances only based on their intrinsic features. It leverages the representation power of VAEs [5], and manipulates the latent space to better adhere to the case considered. This manipulation

is achieved through the introduction of another prior in place of the standard Gaussian of VAEs, defined by a tailored Gaussian Mixture Model (GMM) [2]. To show the general training workflow of a VaDE algorithm, we can divide it into two distinct phases:

1. The model is trained with a traditional loss composed of:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + D_{\text{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big) \tag{1}$$

   where the first term is the usual reconstruction error (implemented as mean square error (MSE)) and the second is the Kullback-Leibler (KL) regularization term considering a normal distribution $p(z) = \mathcal{N}(0, I)$. From now on, we will refer to this step as the *pre-training* step.
2. The raw latent space from *pre-training* is then refined with a further step, called *fine-tuning* step, with a modified version of Equation 1. In this second phase, the prior consists of multiple Gaussians evaluated with a GMM: $p(z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z; \mu_k, \Sigma_k)$

While the two training steps are present in the same way in our model, the most relevant difference lies in the composite loss we defined for our task, which is very challenging due to the input considered. From each cell, indeed, we extract a 22-dimensional vector containing relevant features about its contour; a description of these features is provided in Section S2 of the Supplementary Material. We chose this feature set to ensure consistency across different image conditions (e.g., a different objective that results in a different magnification of the cell). This choice ensures a generalizable model that only classifies cells based on size- and orientation-invariant properties. Starting from this input, we defined a fully connected architecture for our model (Table 1), with a mirrored structure between encoder and decoder. We chose a 3-dimensional latent space for 2 reasons: firstly, because we observed that the input features contain strong correlations among them, allowing the useful information to be efficiently reduced to 3 dimensions; secondly, it lets us keep track, in an immediate visual way, of each dimension during the training. Despite the first motivation remaining the most important one, checking the dimensions constitutes an effective way to keep track of cluster evolution, since our main focus was to establish a meaningful and general latent space into which cells can be projected. As already mentioned, beyond the model structure, the main difference lies in the complex loss we defined, to improve cluster quality and make the network separate different cell instances more effectively. Our loss is composed of 5 different components, defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \beta(t) \cdot \mathcal{L}_{\text{KL}}^{\text{VaDE}} + \lambda(t)\mathcal{L}_{\text{sil}} + \gamma(t)\mathcal{L}_{\text{fisher}} + \psi(t)\mathcal{L}_{\text{triplet}} \tag{2}$$

Each component of $\mathcal{L}_{\text{total}}$ is thus defined as follows, where r_{ik} is the GMM responsibility of sample i w.r.t. cluster k, μ_i and σ_i^2 are the encoder posterior mean and variance for the i^{th} sample, c_k is k^{th} cluster centroid, B is the training

batch size:

$$\mathcal{L}_{\text{rec}} = \frac{1}{B}\sum_{i=1}^{B}\left[\frac{1}{2}\cdot\frac{1}{d}\sum_{n=1}^{d}(x_{i,n}-\hat{x}_{i,n})^2 + \frac{1}{2}\left(1-\frac{x_i^\top \hat{x}_i}{\|x_i\|_2\|\hat{x}_i\|_2}\right)\right] \tag{3}$$

$$\mathcal{L}_{\text{KL}}^{\text{VaDE}} = \frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} r_{ik}\, D_{\text{KL}}\big[\mathcal{N}(\mu_i,\sigma_i^2)\,\|\,\mathcal{N}(c_k,\sigma_k^2)\big] \tag{4}$$

$$\mathcal{L}_{\text{sil}} = \frac{1}{2}\left(1-\frac{1}{B}\sum_{i=1}^{B}\frac{s_i^{out}-s_i^{in}}{\max(s_i^{in},s_i^{out})}\right) \tag{5}$$

with

$$s_i^{in} = \sum_{k=1}^{K} r_{ik}\|\mu_i - c_k\|^2 \;,\; s_i^{out} = \frac{\sum_{k=1}^{K}(1-r_{ik})\|\mu_i-c_k\|^2}{\sum_{k=1}^{K}(1-r_{ik})}$$

$$\mathcal{L}_{\text{fisher}} = \frac{\sigma^2_{\text{intra}}}{\sigma^2_{\text{inter}}+\varepsilon} \tag{6}$$

with

$$\sigma^2_{\text{intra}} = \frac{\sum_{i=1}^{B}\sum_{k=1}^{K} r_{ik}\|\mu_i-c_k\|^2}{\sum_{i=1}^{B}\sum_{k=1}^{K} r_{ik}} \;,\; \sigma^2_{\text{inter}} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{B} r_{ik}\|c_k-\bar{\mu}\|^2}{\sum_{k=1}^{K}\sum_{i=1}^{B} r_{ik}}$$

$$\mathcal{L}_{\text{triplet}} = \frac{1}{N_h}\sum_{i=1}^{N_h}\max\left(0, d(a_i,p_i) - d(a_i,n_i) + m\right) \tag{7}$$

with

$$d(x,y) = \|x-y\|_2^2 \;,\; m = 0.3\cdot\frac{1}{K(K-1)}\sum_{k\neq j}\|c_k-c_j\|_2^2$$

Analysing each component separately, we used a variant of the usual $\mathcal{L}_{\text{rec}}$, considering both magnitude (MSE) and direction (cosine) differences (Equation 3) as we wanted to preserve the feature structure as much as possible, expecting a high correlation among them. We then considered the VaDE loss (Equation 4) with a minor change as we did not consider the KL-entropy term from [3] to enhance training stability. In addition, we also adopted 2 cluster-based losses (Equation 5, 6) to obtain well-separated cell groups. The last contribution (Equation 7) is a variant of the classical triplet loss [12] and is meant to address the most challenging cell cases in order to assign them to the cluster with the highest similarity. This last term is the only supervised component of our entire VaDE model, as it uses real labels from the dataset. Indeed, our particular version of the triplet loss considers GMM responsibilities as confidence scores (instead of ground-truth labels) and, to address the shortage of training data, falls back on the GMM centroids if no hard true label is found in the specific batch. Furthermore, the margin m shown in Equation 7 is evaluated according to the clusters' distribution in the latent space after the *pre-training*. Despite reducing the adaptability to the evolving latent space, this ensures a more stable training. To control the contribution of each described loss during training, we additionally defined time-dependent hyperparameters (Equation 2).

2.2 Dataset

We considered cell images obtained from experiments with particular microfluidic conditions: (1) straight channels without any constrictions; (2) micro-capillary size comparable with cell sizes; (3) flow velocities in the physiological range. All these images were acquired in-house with Erysense, through private experiments. We used, for the training and evaluation of our VaDE clustering model, 2 datasets restricted to healthy donors, from which 3 cell groups were hypothesized (Fig. 1) and subsequently confirmed via unsupervised selection (Section 3.1). The first dataset is a labeled set of 1449 cells - referred to as *labeled*, evenly divided among the 3 classes. Since it aggregates cells from different donors, whose identity has not been retained during acquisition, it is not possible to guarantee donor-wise independence during train/validation splits. The second dataset - referred to as *test*, composed of 3119 cells, was used solely for visual inspection, and it is a totally raw dataset, without any labels. We used this dataset to assess our model's reproducibility and robustness under substantially different focus and noise conditions compared to the *labeled* one. Therefore, it does not constitute an independent quantitative test set.

3 Results

3.1 VaDE training

To train our VaDE, we used the *labeled* dataset, dividing it into the training set (75% of images) and the validation set (25% of images). Training hyperparameters for both training stages are summarized in Table 2. In both phases, we used a cyclic β-annealing to ensure the model can iteratively explore different reconstructions while mitigating posterior collapse through latent space regularization; to also avoid dimension-wise collapse to the prior (and thus meaningless dimensions) we introduced, in this *pre-training*, the so-called $free_{bits}$ parameter, which requires each dimension to contribute at least this value to the $\mathcal{L}_{\mathrm{KL}}$. In *fine-tuning*, we designed a linear ramp schedule for both cluster-based losses (λ, γ) and the triplet loss (ψ), introducing them gradually to let the model adapt to the new GMM prior. First, λ and γ were introduced together, to refine the raw cluster boundaries, followed by ψ, to address challenging cases. Particularly, our hard-case triplet loss defines anchors, positive, and negative examples (a_i, p_i and n_i in Equation 7) according to the GMM, but considering real labels. Specifically, anchors are selected by low GMM confidence to assign them to a cluster ($p < 0.75$), while p_i and n_i are instances with $p > 0.9$. The GMM was fitted on the *pre-training* latent space, with *diagonal* covariance (for consistency with the mean-field KL approximation (Equation 4)). To select the number of sub-populations $K = 3$ with an unsupervised approach, we considered the minimization of the Bayesian Information Criterion (BIC) [13].

Table 1. VaDE architecture. Arrow: linear layer; BN: batch normalization.

Stage	VaDE
Encoder 1	22→16, BN, ReLU
Encoder 2	16→8, BN, ReLU
Latent	$\mu, \log\sigma^2$: 8→3
Sampling	$z \sim \mathcal{N}(\mu, \sigma^2)$
Decoder 1	3→8, BN, ReLU
Decoder 2	8→16, BN, ReLU
Output	16→22

Table 2. Training hyperparameters for *pre-training* and *fine-tuning*.

Parameters	Pre-tr.	Fine-t.
Optimizer	Adam	Adam
l_{rate}	1e-3	1e-3
w_{decay}	4e-4	5e-4
Epochs	500	400
Patience	35	90
β cycle [ep]	33	80
β_{max}	0.05	0.45
$free_{bits}$	0.05	–
λ, γ ramp [ep]	–	60→100
$\lambda_{max}, \gamma_{max}$	–	$5, 0.5$
ψ ramp [ep]	–	100→140
ψ_{max}	–	0.1

3.2 Clustering performance

To demonstrate the improvement achieved by our model compared with a classical VAE, we present classification results on the validation set as confusion matrices and visual representation of the evolution of dimensions before and after *fine-tuning*. As we can notice from the first confusion matrix (Fig. 3a), the model after the *pre-training* already leads to a good accuracy across all the classes, showing a particularly high number of misclassifications among *croissant* and *ball* cells. As this can be expected, we can nevertheless see from the dimensional comparison (Fig. 4a-b) that during this first phase the model successfully reduces dimensions to the 3D latent space, producing a highly meaningful latent space, in which we can already notice a distinction between different groups of cells. The large overlap and the concentration in just a portion of space are a consequence of the simplistic Gaussian prior. This is exactly where the *fine-tuning* acts, with the GMM-tailored prior; it is indeed clearly visible that there is a significant improvement in both the confusion matrix (Fig. 3b) and the dimensional comparison (Fig. 4c-d). Observing the evolution of the latent space, it is clear that this second stage refines the latent space to achieve a better cluster separation, strongly driven by both the new prior and the cluster-based losses. Thanks to the triplet loss, we observe a substantial improvement in the overall accuracy (Table 4), resulting in a reduced overlap between *croissant* and *ball* predictions. It is clear that this emphasis on cluster separation inevitably increases the reconstruction error (and thus the total loss, as it is the main component). However, this is justified by the model's improved ability to distinguish among the broad range of cell shapes, as shown in Fig. 4e-f, where a similar latent organization is reproduced on the *test* set. Although the clustering quality metrics show a slight degradation on the *test*, with the silhouette score decreasing from 0.91 to 0.88 and the Davies—Bouldin index increasing from 0.12 to 0.21, the changes remain limited and indicate good generalization of the latent structure.

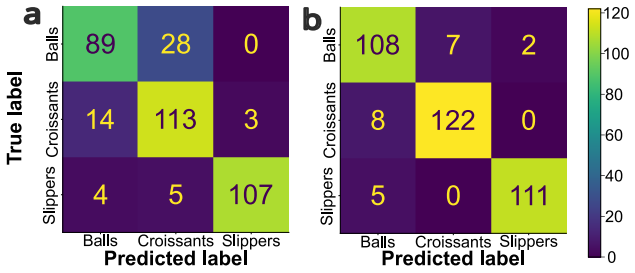

Fig. 3. Comparison of confusion matrices. a) After *pre-training*; b) After *fine-tuning*.

Furthermore, the model consistently projects *test* instances into the same latent regions corresponding to the training clusters, preserving the cell characteristics within each cluster.

4 Discussion and Conclusion

We proposed a novel approach to classify RBCs in flow, keeping the model training as free as possible from biases due to human labeling. As our model is largely unsupervised, except for the triplet loss contribution, it achieves notably high accuracy in grouping cells based only on their shape. Examining the classification metrics for the final model (Table 3), we observe strong performance, with high F1-scores for all classes. Taking into account only the 2 currently accepted classes (i.e., *slippers* and *croissants*), the results improve further, with no misclassification between them (Fig. 3b). To contextualize these results, we compared our model against a simple GMM applied to the scaled 22D input vectors (with K fixed to 3 for a fair comparison). This approach reaches an overall accuracy of 65%, struggling especially to distinguish between *slippers* and *croissants* - as expected, given the need to re-organize the feature space into a more suitable latent representation. We also compared our model to an external supervised baseline: the model developed in [4]. It achieves, as our model does, no misclas-

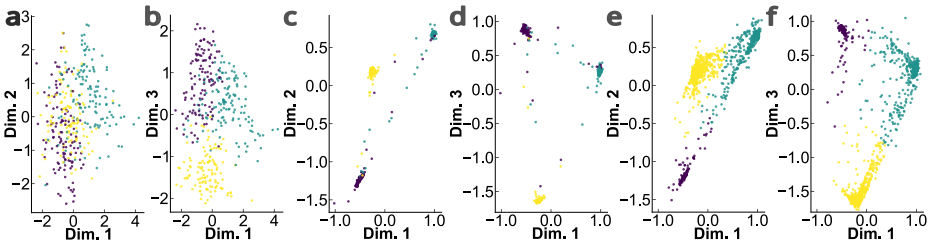

Fig. 4. Latent space dimensions during the two-stage training procedure. a-b) *Labeled* dataset after *pre-training* (ground-truth labels); c-d) *Labeled* dataset after *fine-tuning* (ground-truth labels); e-f) *Test* dataset after *fine-tuning* (model predictions).

Table 3. Per-cluster classification report for final model on validation set.

Metric	Balls	Croissants	Slippers
Precision	0.89	0.95	0.98
Recall	0.92	0.94	0.96
F1-score	0.91	0.94	0.97

Table 4. Overall accuracy on validation set obtained by progressively adding the proposed loss components.

Metric	Accuracy
VAE loss	0.85
VaDE loss	0.88
VaDE + cluster losses	0.90
Full loss	0.94

sification between *slippers* and *croissants*, but shows more errors between *others* and *croissants*. This results in an overall accuracy of 87%, while our model increases this value by 7%. Despite not sharing the exact same classes (*others* in [4] shares several characteristics with our *balls*), this comparison suggests that our model better handles challenging cases. The clear improvement in terms of overall accuracy over the classical VAE and VaDE approaches (Table 4) justifies the introduction of such a complex loss, despite the difficulty in balancing its components. This is the direction of our future research; we believe, especially for the triplet loss, that the small size of the datasets limits its full potential. We aim to further develop this SSL approach and introduce more features, to identify additional classes characterized by distinct cell properties.

Disclosure of Interests. Andrea Caputo is employed by Cysmic, the company that provides the Erysense, the device which generates the images, where Stephan Quint serves as Chief Executive Officer and Lars Kaestner is a shareholder.

References

1. Beucher, S., Meyer, F.: The Morphological Approach to Segmentation: The Watershed Transformation, vol. Vol. 34, p. 433–481 (01 1993). https://doi.org/10.1201/9781482277234-12
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: An unsupervised and generative approach to clustering (2017), https://arxiv.org/abs/1611.05148
4. Kihm, A., Kaestner, L., Wagner, C., Quint, S.: Classification of red blood cell shapes in flow using outlier tolerant machine learning. PLOS Computational Biology **14**(6), 1–15 (06 2018), https://doi.org/10.1371/journal.pcbi.1006278
5. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2022), https://arxiv.org/abs/1312.6114
6. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023), https://arxiv.org/abs/2304.02643
7. MacQueen, J.: Some methods for classification and analysis of multivariate observations (1967), https://api.semanticscholar.org/CorpusID:6278891
8. Pachitariu, M., Rariden, M., Stringer, C.: Cellpose-sam: superhuman generalization for cellular segmentation. bioRxiv (2025), https://www.biorxiv.org/content/early/2025/05/01/2025.04.28.651001

9. Quint, S., Christ, A.F., Guckenberger, A., Himbert, S., Kaestner, L., Gekle, S., Wagner, C.: 3d tomography of cells in micro-channels. Applied Physics Letters **111**(10), 103701 (09 2017), https://doi.org/10.1063/1.4986392
10. Recktenwald, S.M., Graessel, K., Maurer, F.M., John, T., Gekle, S., Wagner, C.: Red blood cell shape transitions and dynamics in time-dependent capillary flows. Biophysical Journal **121**(1), 23–36 (2022), https://www.sciencedirect.com/science/article/pii/S000634952103900X
11. Recktenwald, S.M., Lopes, M.G.M., Peter, S., Hof, S., Simionato, G., Peikert, K., Hermann, A., Danek, A., van Bentum, K., Eichler, H., Wagner, C., Quint, S., Kaestner, L.: Erysense, a lab-on-a-chip-based point-of-care device to evaluate red blood cell flow properties with multiple clinical applications. Frontiers in Physiology **Volume 13 - 2022** (2022), https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2022.884690
12. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). p. 815–823. IEEE (2015), http://dx.doi.org/10.1109/CVPR.2015.7298682
13. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics **6**(2), 461–464 (1978)