

INSTITUTE OF COMPUTER
SCIENCES
Master in Artificial Intelligence and Data
Science

Universitätsstr. 1 D–40225 Düsseldorf



Heinrich Heine
Universität
Düsseldorf

AI-based fluorescent labeling for cell line development

Hanna Pankova

Master thesis

Date of issue: 01. April 2022
Date of submission: 29. August 2022
Reviewers: Prof. Dr. Markus Kollmann
Dr. Wolfgang Halter

Erklärung

Hiermit versichere ich, dass ich diese Master thesis selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 29. August 2022

Hanna Pankova

Abstract

Cell line development is an expensive and time-consuming process, however that is the most modern approach for producing the proteins needed in various pharmaceuticals.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Notation	2
2	Domain knowledge	3
2.1	Biology	4
2.1.1	Cell line development process	4
2.1.1.1	Cell line development with CHO cells	4
2.1.1.2	CLD steps	5
2.1.2	Project specifications of cell line development for Merck KgaA	6
2.2	Deep learning and machine learning basics	6
2.2.1	Neural networks	6
2.2.2	Dimensionality reduction methods	7
2.2.2.1	UMAP	7
2.2.2.2	PCA	7
2.2.2.3	PacMAP	7
2.2.3	Clustering methods	7
2.2.3.1	DBSCAN	7
2.2.3.2	HDBSCAN	7
2.2.3.3	K-means	7
2.3	Imaging	7
2.3.1	Digital imaging	7
2.3.2	Microscopy imaging	7
2.3.2.1	Image acquisition peculiarity	7
2.3.2.2	Crops combination technique	8
3	Implementation and experiments	9
3.1	Model training	9
3.1.1	Neural network architecture	9
3.1.2	Loss functions	9
3.1.3	Available data	9

3.1.4	Training costs estimation	10
3.1.5	Augmentations	10
3.1.5.1	Special augmentations for rotation and scaling	10
3.1.6	Model setup	10
3.1.6.1	Weight Initialization	10
3.1.6.2	Regularization	10
3.1.6.3	Optimizers	11
3.2	Nuclei	12
3.2.1	Preprocessing	12
3.2.1.1	Thresholding algorithms	12
3.2.2	Training and predictions	12
3.2.2.1	Convergence	12
3.2.2.2	Predictions quality	15
3.2.3	Postprocessing for nuclei segmentation	15
3.2.4	Influence of scaling on predictions quality	16
3.3	Endoplasmic Reticulum	17
3.3.1	Preprocessing	17
3.3.2	Training and predictions	17
3.3.3	Combination of nuclei and actin predictions	18
3.3.4	Generalizability across phenotypes	18
3.4	Golgi	19
3.4.1	Preprocessing	19
3.4.1.1	Background removal algorithms	19
3.4.2	Training and predictions	20
3.4.3	Alternative ways to improve predictions	22
3.4.3.1	Asymmetrical losses	22
3.4.3.2	Use of gradient in loss	22
3.4.3.3	Noise reduction methods	22
3.5	GFP	23
3.5.1	Preprocessing	23
3.5.2	Predictions	23
3.5.3	Downstream metrics	24
3.5.4	Combination of GFP, nuclei and ER	24

3.6 Model evaluation	25
3.6.1 Metrics for downstream tasks	25
3.6.2 Influence of different loss functions on metrics for downstream tasks	25
4 Stability study	26
4.1 Stability study	26
4.1.1 Artificial corruptions	26
4.1.2 Real corruptions	27
4.1.2.1 Not fixed cells imaging as corrupted input	27
4.1.2.2 Real-world examples of corruptions	27
4.1.3 Influence of corruptions on metrics for downstream tasks	27
4.1.4 Improving predictions with additional corruption augmentations .	27
4.2 UNET embeddings study	28
4.2.1 Application of various dimentionality reduction methods	28
4.2.2 Autoencoder embeddings as an alternative	28
4.2.3 Clustering of PacMAP embeddings	31
4.2.3.1 Clustering on UNet embeddings	31
4.3 Drift detection	32
4.3.1 A need to detect drift	32
4.3.2 Maximum mean discrepancy for drift detection	32
4.3.3 Online version of MMD algorithm	32
5 Software Tools	34
5.1 Foundry. Palantir	34
5.2 AWS	34
5.3 Streamlit	34
6 Future research	35
7 Summary	36
List of Figures	37
List of Tables	38

1 Introduction

1.1 Motivation

Nowadays recombinant proteins are widely used in biomedical research and production of medicines that are used in the variety of therapeutic needs like vaccines and antibodies [TODO add references]. Therefore there is currently a great need for high-volume and high-quality recombinant protein production. That is why the optimization and improvement of cell line development (CLD) as a process in use for the production of recombinant proteins is extremely important.

Clone screening is a step of the CLD process in which cells are analyzed for further selection of the most stable and productive clones. Fluorescence microscopy provides data about the cell structure that enables better clone selection, however it is not only expensive and time-consuming, but also toxic for the cells. Automating fluorescence microscopy for clone selection via convolutional neural networks *in silico* significantly simplifies the existing procedure of clone selection, reducing phototoxicity, time and expenses needed for the analysis.

The goal of this thesis is to provide a proof of concept on whether an *in silico* approach to fluorescent labeling can substitute manual cell staining and provide all the needed information that would be used for further clone screening and selection. That is particularly why the research at hand is aimed towards the specific needs, pipelines and data used at Merck KgaA. In this research four UNet models (for four target proteins highlighting different cell organelles) were developed for automating fluorescence cell staining based on DIC microscopy imaging of CHO cells: nuclei, endoplasmic reticulum, [[green fluorescent protein]] and Golgi apparatus. Another important goal of this research that differentiates it from the similar studies like [TODO cite LaChance 2020 and cite Christiansen 2018] is to not only provide deep learning models for the fluorescence predictions but also study their reliability and be able to detect drift during image acquisition that can happen quite easily due to the sensitivity of the microscope settings as well as the cell phenotypes, scaling and fixation procedures.

This thesis is laid out as follows: Section 2 reviews the biological concepts needed to understand the application of this research, it also reviews machine and deep learning concepts used for data analysis; Section 3 provides an overview of the implementation and the results of the experimental *in silico* fluorescence predictions; Section 4 shows stability of the deep learning models developed in the previous section and provides valuable insights on the information from their embeddings; Section 5 details the practical tools used for the development at Merck KgaA and Section 6 explores possible future research questions that arose from the current analysis and provides concluding remarks and succinct recommendations.

1.2 Notation

$x^{(i)}$ The i-th input image (sample) from a dataset

X_{train} A set of training examples

$y^{(i)}$ The target image associated with the i-th input sample from a dataset

p_{data} Data generating distribution

2 Domain knowledge

The *in silico* fluorescence labeling approach has proven to be very promising as a substitute to the manual cell staining processes [TODO cite all the relevant references]. For example, the research of [TODO cite Christiansen 2018] did not only prove successful prediction of different cell stains with a variety of modalities and cell types, but it had also successfully determined cell viability. Nevertheless, the study is limited mainly to transmitted light (TL) z-stack imaging. This refers to the networks input being comprised of 3D images, which is not the case in this work. [cite Ounkomol 2018] too shows successful predictions of several organelles in bright-field TL 3D images using 3D convolutional neural networks. However, switching to 2D data did not yield adequate results for them. Other, newer studies like [cite Ugawa 2021] provide an application of label-free fluorescence predicting already at the sorting stage, when a high-throughput system sorts cells individually. However, only a single-pixel detector is used by this study, meaning that it captures a wave rather than an image. Nonetheless one can recover an image with heavy computations if needed [cite Sadao Ota 2018].

There are two very promising studies by [cite Cheng 2021] and [cite LaChance 2020]. Even though the former manages to reach a state-of-the art performance on label-free fluorescence reconstruction, it uses reflectance images from oblique dark-field illumination as the input, which is a more specific cell imaging approach. Still, this input provides higher structural contrast in comparison to any transmission technique [cite Boustany 2010]. The latter study uses an easier imaging technique (DIC imaging) as an input, which shows great results even with low-resolution data. Both of these studies provide results based not only on training metrics, but also on performance of the models for metrics used in the downstream tasks. This is very important in the label-free fluorescence labeling research and was not present in papers before LaChance. In the thesis at hand, many methods from the LaChance paper were used as both the data and the processes in the project pipeline of Merck KgaA align very well with the study conducted in that paper.

All of the studies mentioned above, as well as this work rely on the premise that the input imaging type (here DIC) contains enough information to predict the fluorescence signal from it. This is a reasonable assumption because DIC, as well as bright-field and phase contrast imaging, are very often used for determining cell morphology [TODO cite Kasprowicz 2017].

This chapter provides a brief overview of the biological background needed to understand the process of cell line development (CLD) and the role of fluorescent *in silico* labeling of DIC cell images within. It also covers the fundamentals of deep and machine learning techniques used here including clustering and dimensionality reduction approaches. At the end of the chapter, a brief summary of the microscopy image acquisition process used in the research is given.

2.1 Biology

2.1.1 Cell line development process

2.1.1.1 Cell line development with CHO cells

Cell line development (CLD) is a process of generating single cell-derived clones that produce high and consistent levels of target therapeutic protein [TODO cite pharma.lonza.com/offerings/mammalian/cell-line-development]. Therapeutic proteins in this case are so-called recombinant proteins and they are spreadly used in the biomedical research, medicine production and for many different therapeutic needs like, for example, vaccines or monoclonal antibodies (mAbs) [TODO cite Ohtake 2013, Jefferis 2021, Funaro 1996]. A recombinant protein is defined by [cite Barbeau, J] as a modified or manipulated protein encoded by a recombinant DNA. Recombinant DNA in its turn consists of a plasmid, where the genes of a target protein of interest are cloned downstream of a promoter region. As soon as this plasmid will be transfected to a host cell (for example some mammalian cells that are able to produce the protein), the host wil start to express of this protein of interest. In nowadays industry and research there is a great need for production of recombinant proteins in high volumetric amounts and of a good quality [TODO cite Tihanyi 2020]. That is why the goal on many researches is in recombinant protein production is to create a their efficient expression and high-throughput systems to improve the CLD processes [cite Tihanyi 2020].

One of the most popular host cells used in CLD and in this work specifically are chinese hamster ovary (CHO) cells [cite Castan 2018]. Altough different cells can be used as hosts like bacterial, plant-based, yeast cells, the most popular ones remain mammalian [cite Beckman]. The reason behind this popularity hides in the fact that they can produce diverse correctly folded proteins and most importantly they have high productivity protein production rates. Productivity rate is measured in titre of produced protein, and CHO cells can reach 0.1 - 1 g/L in batch and 1-10 g/L in fed-batch cultures [cite Tihanyi 2020]. Mostly all of the mAbs are produced using CHO cells [cite Lalonde 2017]. Companies mostly use the same host cell line for their productions because already checked and qualified cells simplify the road to the clinic [cite Tihanyi 2020]. That is why current research has a wide applicability.

However there is a downside in using CHO as host cells - they are well-known by their instability. As a rapidly growing immortal cells CHO also genetically unstable and extremely heterogeous which ususally leads to their main problem - production instability. The problem of choosing the stable and high producing clones that simutaneously will be able to express protein qualitatively and quantitably over time is essentially the main goal of nowadays research. The big challenge in manufacturing here are the time and costs of production. Currently a lot of research attention is dedicated to the reduction of both as well as to developing techniques of high-throughput clone screening and characterization [cite Tihanyi 2020]. The latter one is of interest for this thesis. With the great amounts of data acquired over time and the development of the computational modelling and statistical analysis it is possible now to do the analysis *in silico*, meaning - computationally without intervening into the cells instead of the usual *in vitro* analysis.

2.1.1.2 CLD steps

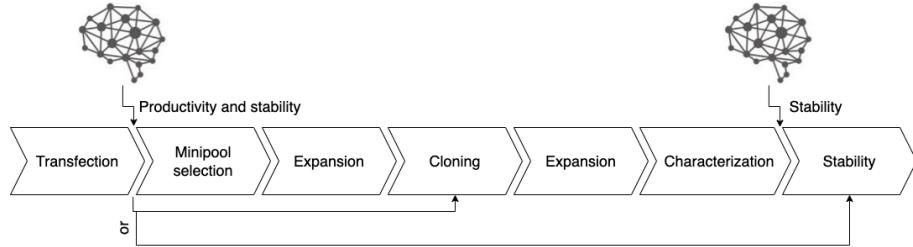


Figure 1: CLD process steps

First step of CLD is called transfection - the introduction of the gene of interest (GOI or a DNA vector or alternatively an expression vector) to CHO cells. And it has two main problems: first is that the transfection mostly results in a vector being inserted into a random site within a host cell genome and second, that it generally has low efficiency of integration [cite Tihanyi]. It is important to transfet a GOI into an optimal site of genome to secure a high protein expression over time during protein production, however practically transfection happens into a random location of genome. In cases when the gene was transfected into the inactive site of genome (and essentially the majority of genome is transcriptionally inactive) the cell will likely not be able to express the gene [cite Castan, Hong 2018].

The second step of the process is selection of cells minipoools that have successful and stable gene integrations for further expansion and cloning. The reason why not all of them are suitable is the following: during the transfection step only 80% of the cells will receive a vector of GOI [cite Castan]. Only a small percent of these cells actually integrate a vector into the genome and, as mentioned above, only a fraction of those are able to stably express the protein. [a better reference needed Shin 2020]. After the best mini pools are selected, they will be expanded.

The third step in CLD is to clone the cells. Chosen stable pools of cells are phenotypically and genetically diverse - meaning they have different growth rates, metabolic profile and etc. This is not ideal for industrial production - all the cells used for protein production should be derived from the same clone [cite [25] from Castan].

Once the cells are cloned, phenotypical and genetical heterogeneity is reduced, the next step is to characterize the cells for their expression of the GOI. One has to estimate clones productivity and stability. Such observations may take up to 90 days (usually the checks are made on the day 30, day 60 and day 90). If the clones remain stable after this time and are able to express enough of the protein, then they are suitable for further production. However this last step consumes a lot of time and maintaining costs for feeding and analysing the cells. Predicting productivity and stability of the cells on earlier stages would reduce this time significantly or even allow to escape this process completely.

2.1.2 Project specifications of cell line development for Merck KgaA

There are many different proteins that can be produced using such technologies, for example, vaccines, hormones, sugars and etc., however this research is dedicated to the production of monoclonal antibodies (mAbs).

CHOZN® Platform is a currently widely used product of Merk KgaA. CHOZN is a CHO mammalian cell expression system for the fast and easy selection and scale up of clones producing high levels of recombinant proteins [cite tech-bulletin]. The processes of developing the expression systems on this platform correspond to the general CLD process described in the previous subsection [put subsection number]. The scope of the project is to facilitate the prediction of stability of the expression system by inducing an *in silico* predictions of productivity and stability rates on early steps on the CLD process.

In order to choose single best cells for further cloning one asseses several parameters like cell size, granularity, cell surface protein expression and etc. This can be done with Fluorescent Activated Cell Sorting (FACS) technology that allows to sort signle cells. (<https://doi.org/10.1517/14712598.4.11.1821>). Unfortunately fluorescence labeling is expensive and may ruin the cell due to its phototoxicity (<https://doi.org/10.1371/journal.pone.0007497>).

2.2 Deep learning and machine learning basics

Introduction of the notaiton for the dataset, parameters, predictions.

2.2.1 Neural networks

Convolutional neural network, Autoencoder, embedding, optimizers, regularization, descriptions of how each layer works.

Convolutional neural networks (CNNs) capture nonlinear relationships over large areas of images, resulting in vastly improved performance for image recognition tasks as compared to classical machine learning methods

TODO cite Oukomol

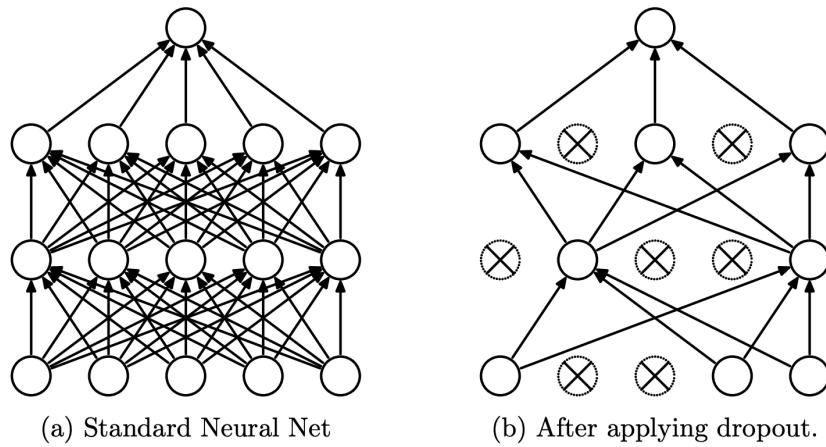


Figure 2: Dropout

2.2.2 Dimensionality reduction methods

2.2.2.1 UMAP

2.2.2.2 PCA

2.2.2.3 PacMAP

2.2.3 Clustering methods

2.2.3.1 DBSCAN

2.2.3.2 HDBSCAN

2.2.3.3 K-means

2.3 Imaging

2.3.1 Digital imaging

How image is stored in memory, which conventions there are (RGB, BGR (conventions are used in corruptions augmentations)).

2.3.2 Microscopy imaging

2.3.2.1 Image acquisition peculiarity

Which difficulties it may cause (validation loss is lower than train loss)

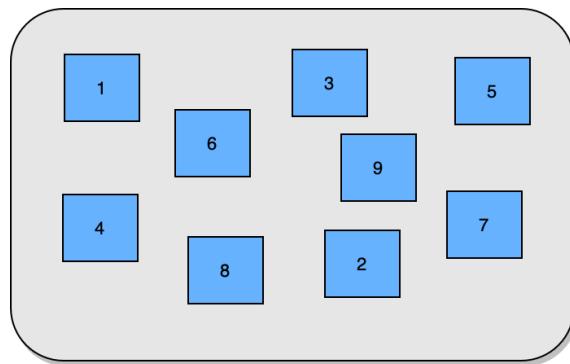


Figure 3: Way in which photos of the well-plate were taken

2.3.2.2 Crops combination technique

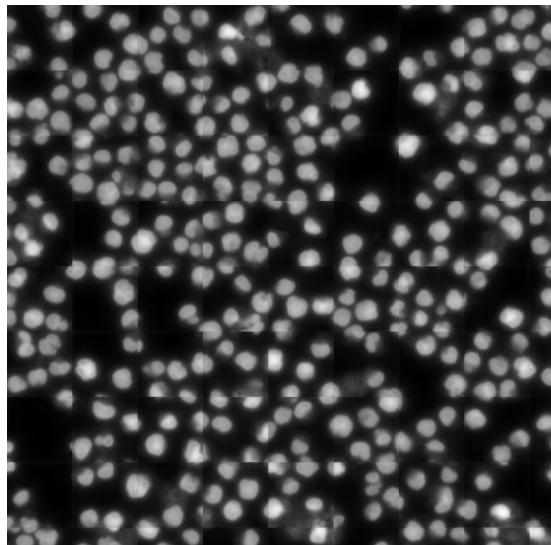


Figure 4: No overlap

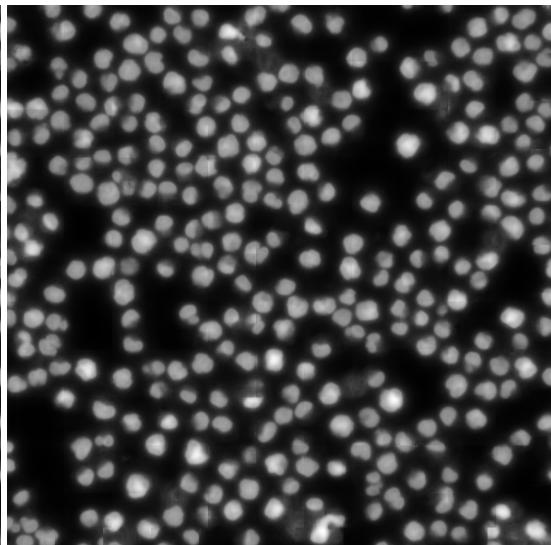


Figure 5: 30 pixels overlap

Improve this plot by showing the visible border explicitly, example of how it can influence a further segmentation perhaps?

3 Implementation and experiments

3.1 Model training

3.1.1 Neural network architecture

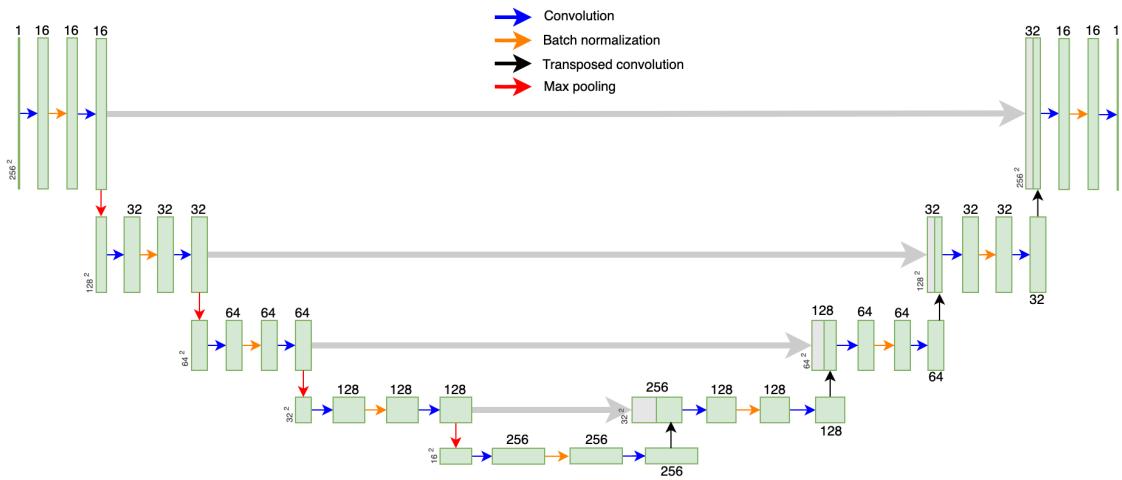


Figure 6: Unet

And information on the embeddings, output sizes, amount of parameters, etc.
take description from here file://Users/anniepank/Downloads/media-1.pdf

3.1.2 Loss functions

Which loss functions were used, Pearson correlation coefficient explained.

3.1.3 Available data

Description of the datasets and the amount of images in each category.

Table 1: Available data for each fo the organelles

	Total images	Training crops	Validation crops	Test crops
Nuclei	595	27,264	3,008	7,616
Actin	400	18,432	2,048	5120
Golgi	761	23,036	2,336	6,347
H19	400	27,264	3,008	7,61
Nucleolei	?	?	?	?

3.1.4 Training costs estimation

Table with the estimation of costs and times for AWS

3.1.5 Augmentations

Description of all augmentations used

3.1.5.1 Special augmentations for rotation and scaling

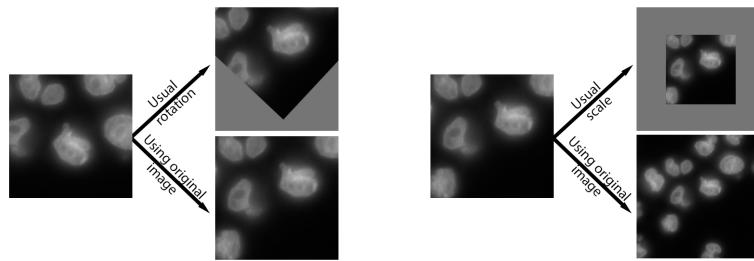


Figure 7: Using original image for rotation and scaling augmentations

3.1.6 Model setup

3.1.6.1 Weight Initialization

These plots represent MSE

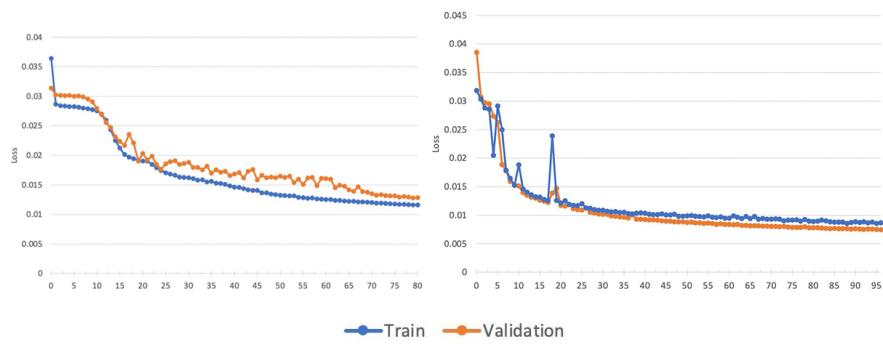


Figure 8: Nuclei training without and with custom weight initialization

3.1.6.2 Regularization

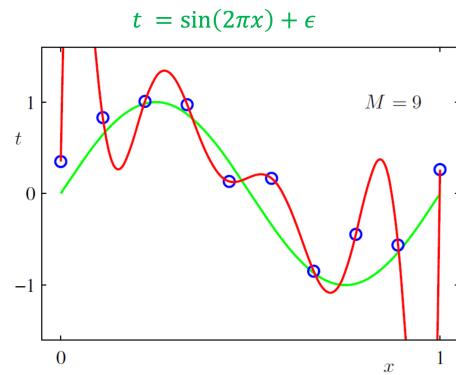


Figure 9: Overfitting

3.1.6.3 Optimizers

Comparison of different optimizers

3.2 Nuclei

3.2.1 Preprocessing

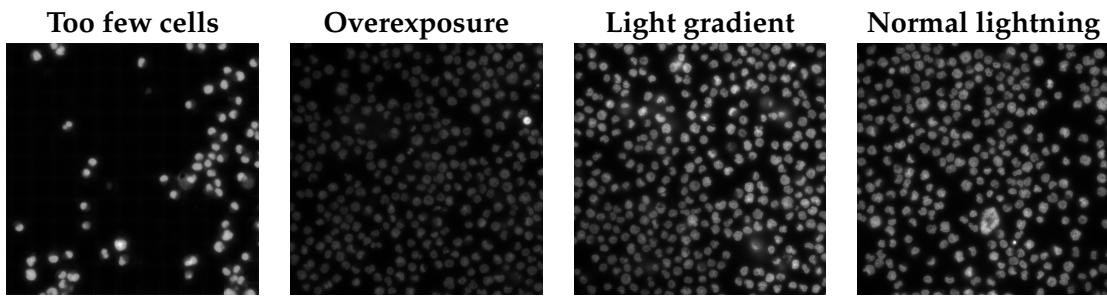


Figure 10: Different lightning conditions

3.2.1.1 Thresholding algorithms

Global and local thresholding

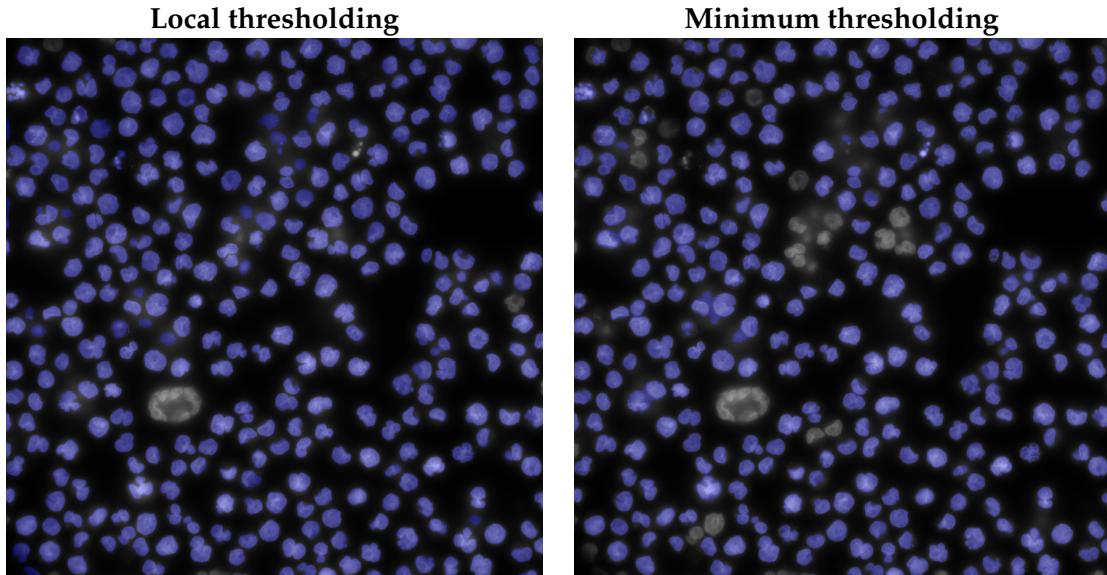


Figure 12: Local vs. Global thresholding (normal conditions)

3.2.2 Training and predictions

3.2.2.1 Convergence

Has the model converged or not. Will more data help?

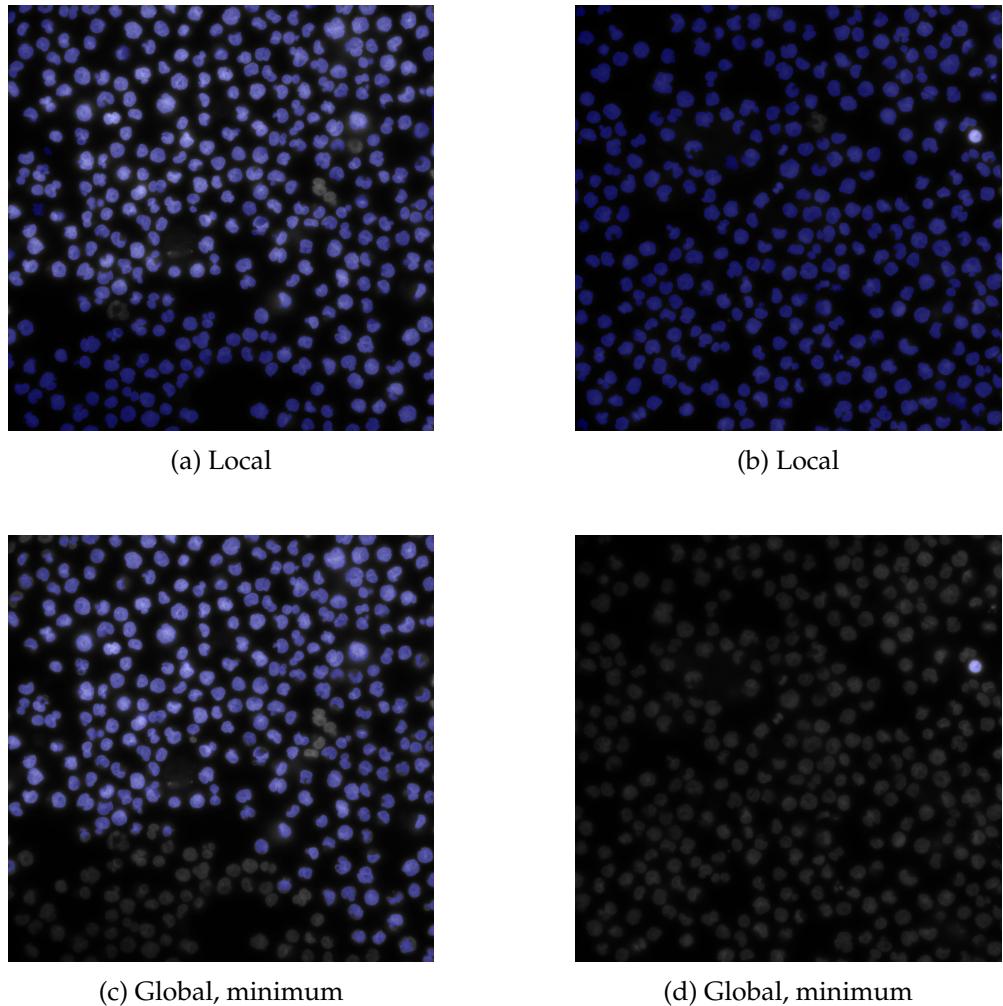


Figure 11: Local vs. Global thresholding

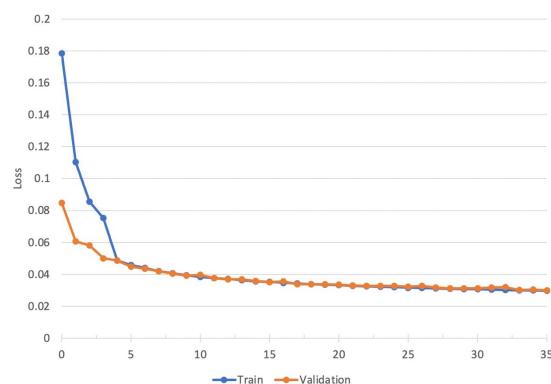


Figure 13: Having more data makes training more stable

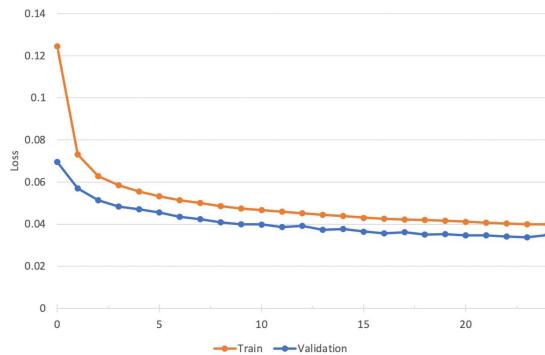


Figure 14: With regularization and augmentations PCC

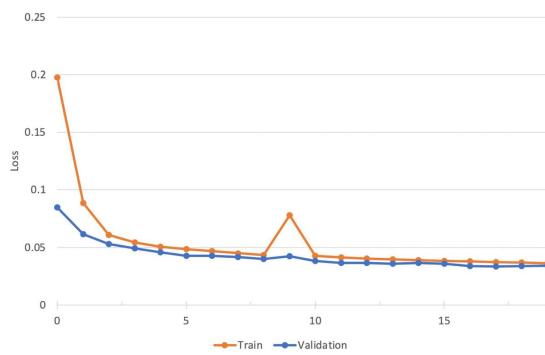


Figure 15: No regularization but augmentations

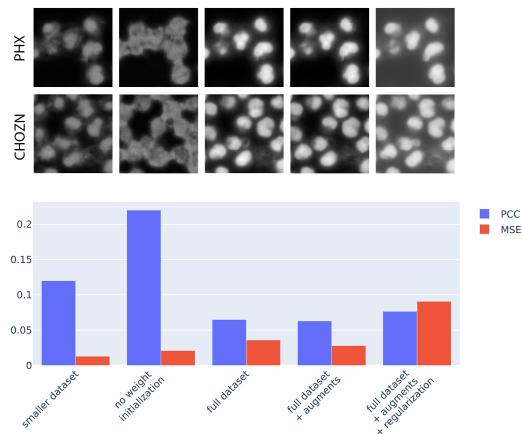


Figure 16: Difefrent models predictions and scores comparison

3.2.2.2 Predictions quality

Blurry, boundaries, not enough of details and possible improvements

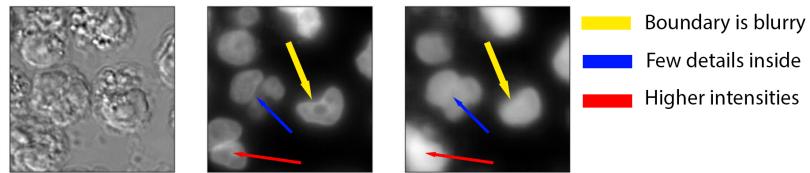


Figure 17: Problems in predictions

3.2.3 Postprocessing for nuclei segmentation

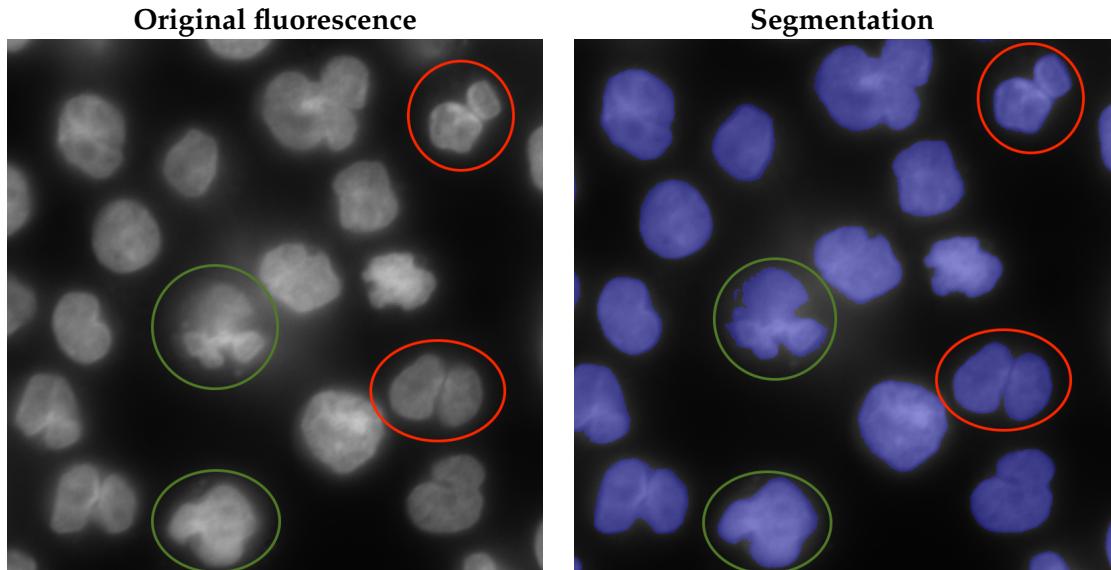


Figure 18: Closely located cells

Overall algorithm

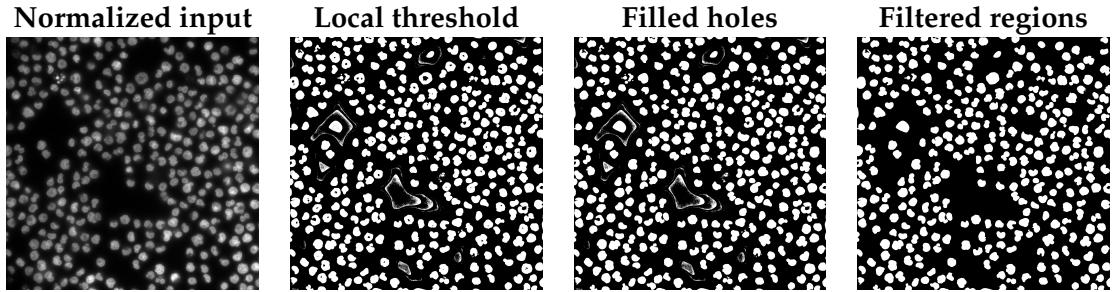


Figure 19: Fluorescence segmentation

3.2.4 Influence of scaling on predictions quality

Examples of predictions quality with different scales.

Table 2: Pearson correlation coefficients for downstream tasks for different scaling factors

	1.3 scale	0.7 scale	Train (1.0 scale + augments) Predict (1.3 scale)	Train (1.3 scale) Predict (1.0 scale)	Train (1.3 scale) Predict (0.7 scale)
Number of nuclei	0.987	0.995	0.975	0.971	?
Total intensity	0.902	.88	0.861	0.856	?
Mean intensity	0.922	0.906	0.88	0.872	?
Area	0.991	0.992	0.961	0.952	?

3.3 Endoplasmic Reticulum

3.3.1 Preprocessing

Algorithm 1 Fluorescence segmentation

1. Normalize image
 2. Apply global *threshold_mean* to receive initial mask.
 3. Zero out pixels outside the mask
 4. Apply local thresholding.
 5. Apply *fill_holes* transformation.
 6. Morphological opening from OpenCV and Gaussian blur.
 7. Run *findContours* from OpenCV in order to obtain separate regions and filter out too small regions.
-

Segmentation steps are also illustrated in Figure

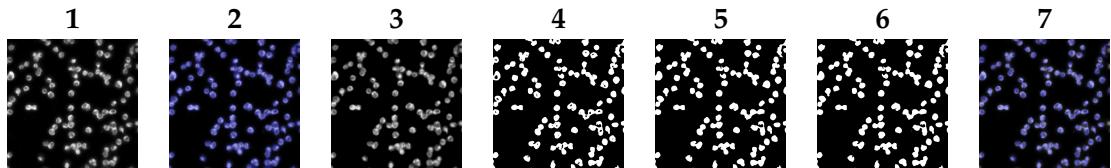


Figure 20: ER prediction

3.3.2 Training and predictions

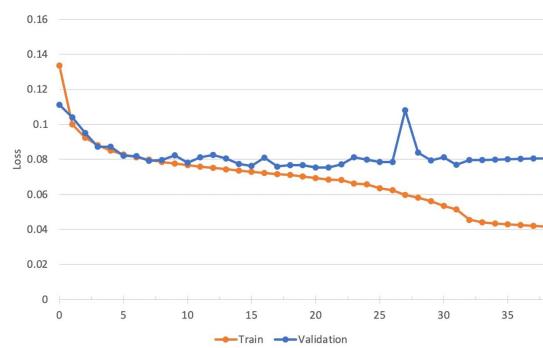


Figure 21: Overfit

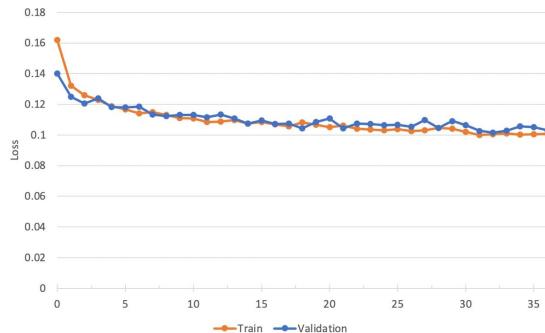


Figure 22: No overfit with augmentations

3.3.3 Combination of nuclei and actin predictions

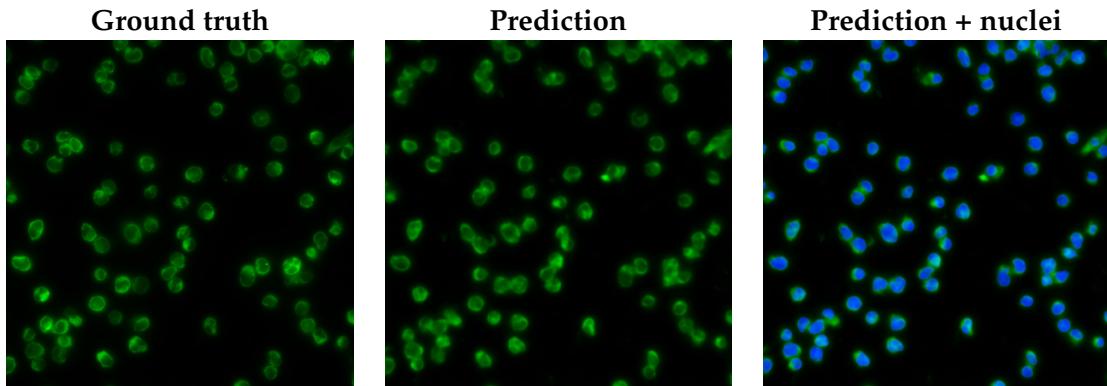


Figure 23: ER prediction

3.3.4 Generalizability across phenotypes

TODO train the model on one phenotype and predict on the other, compare predictions (visually?) postprocessing with metrics then? Add metrics

3.4 Golgi

3.4.1 Preprocessing

Enhancement

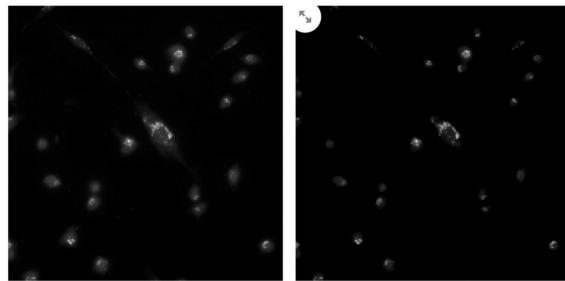


Figure 24: Golgi enhancement

3.4.1.1 Background removal algorithms

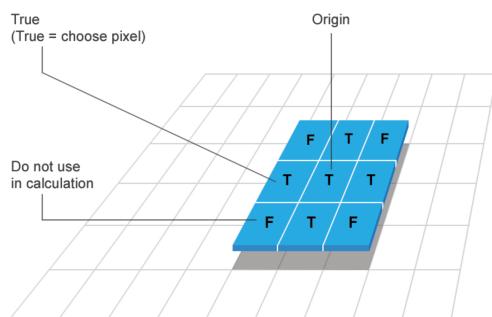


Figure 25: Structuring Element

Rolling ball algorithms

Rolling ball still leaves some noise

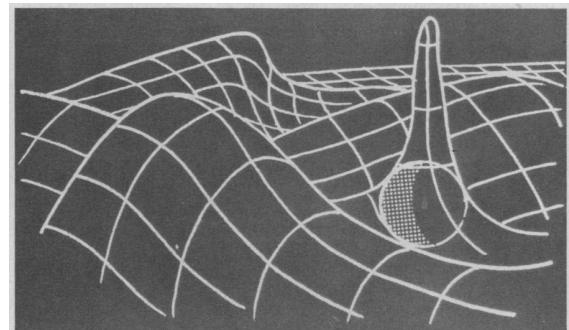


Figure 26: Rolling Ball

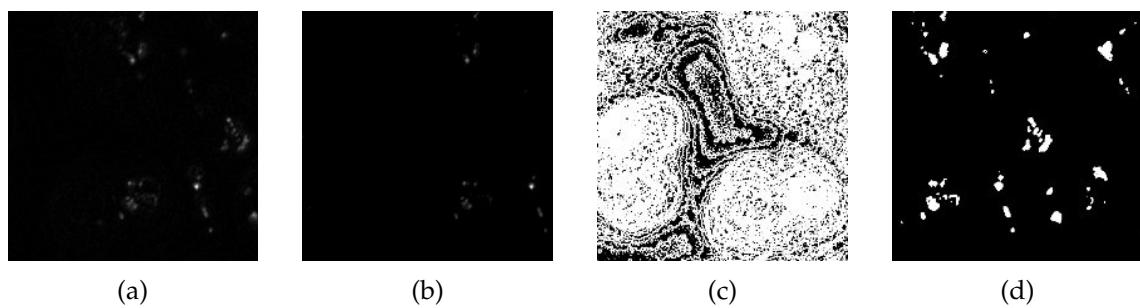


Figure 27: (a) Vanilla pre-processing with automatic background removal algorithm only; (b) Additional clipping of lower intensities after vanilla pre-processing; (c) masked or subfigure (a); (d) mask of subfigure (b)

3.4.2 Training and predictions

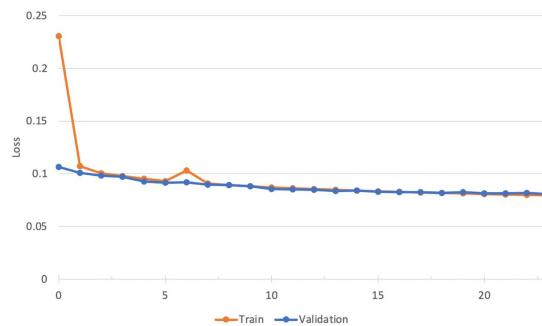


Figure 28: Straightforward training doesn't work

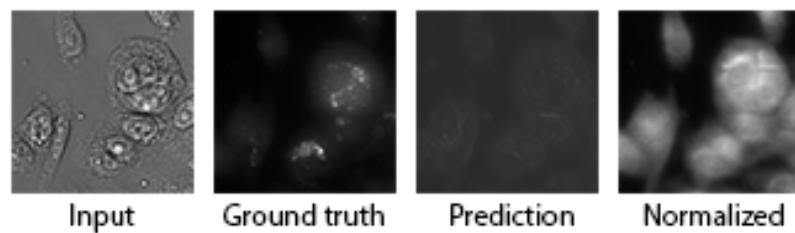


Figure 29: Training on original data

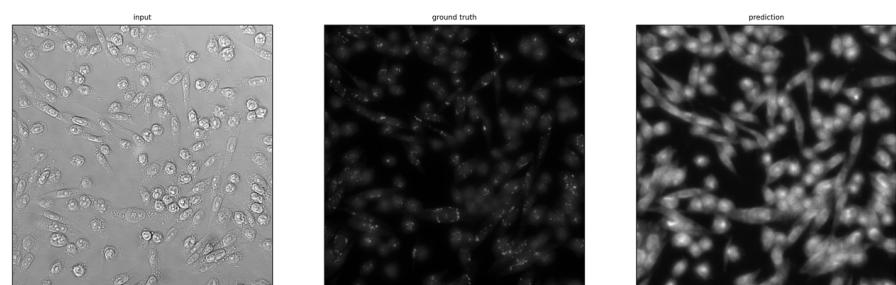


Figure 30: Full size predictions

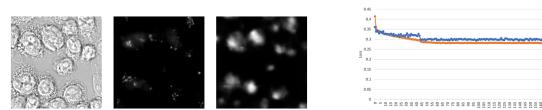


Figure 31: Training on the enhanced data

3.4.3 Alternative ways to improve predictions

3.4.3.1 Asymmetrical losses

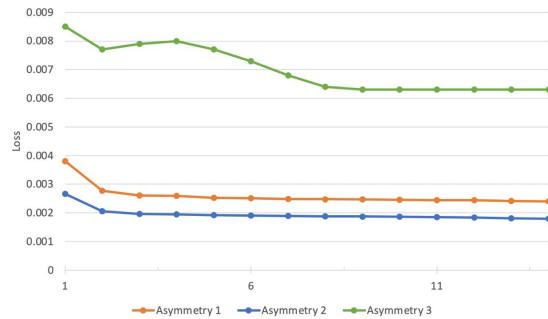


Figure 32: Asymmetrical training

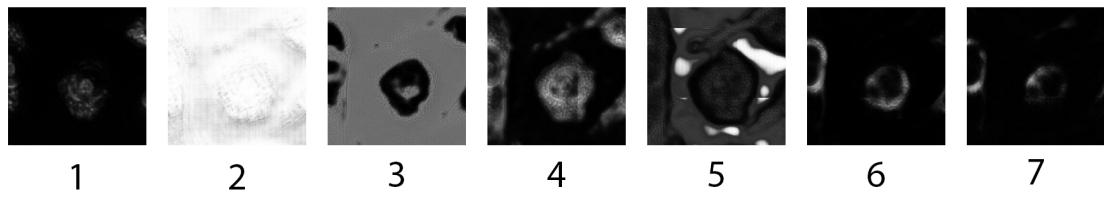


Figure 33: Asymmetrical training predictions

3.4.3.2 Use of gradient in loss

3.4.3.3 Noise reduction methods

3.5 GFP

3.5.1 Preprocessing

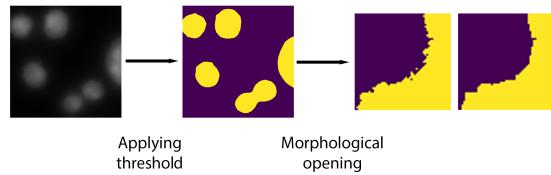


Figure 34: Converting GFP to a binary mask

3.5.2 Predictions

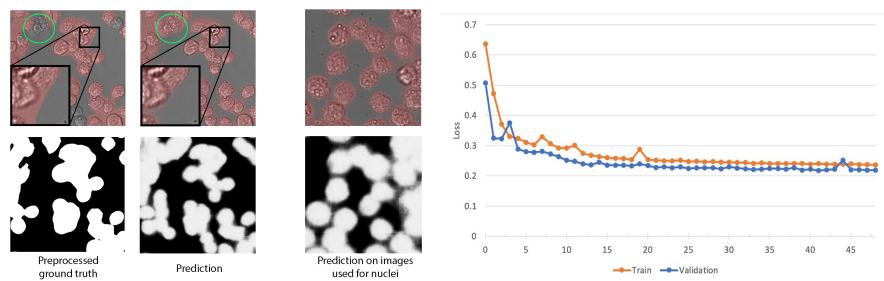


Figure 35: Training with BCE loss

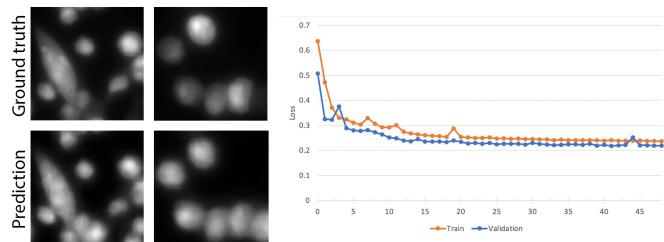


Figure 36: Training with Pearson correlation loss

Table 3: Correlation coefficients for downstream tasks

Binary training	Pearson	Spearman
Number of ER	0.67	0.64
Area	0.82	0.75
Continuos training	Pearson	Spearman
Number of ER	0.57	0.55
Area	0.26	0.64

3.5.3 Downstream metrics

TODO move to separate chapter?

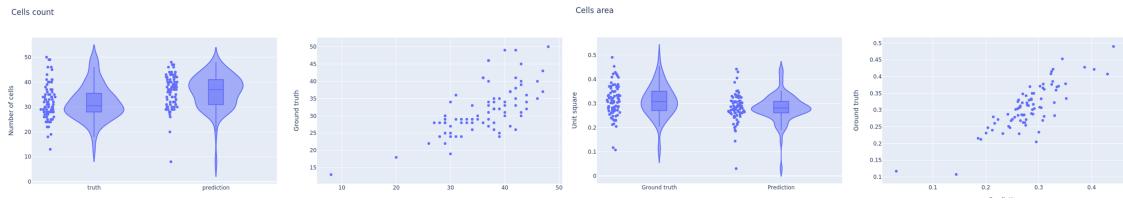


Figure 37: Downstream metrics

3.5.4 Combination of GFP, nuclei and ER

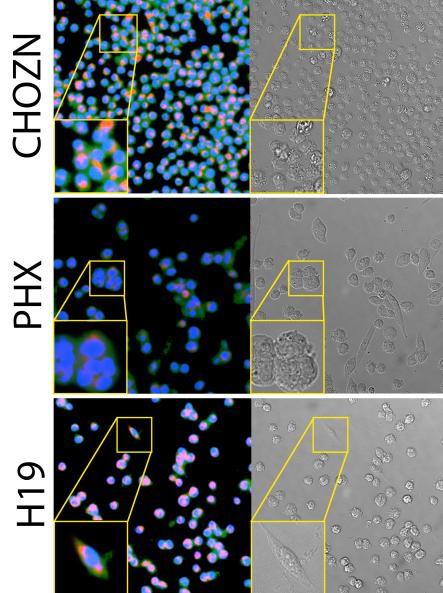


Figure 38: GFP, Nuclei and ER combined

3.6 Model evaluation

3.6.1 Metrics for downstream tasks

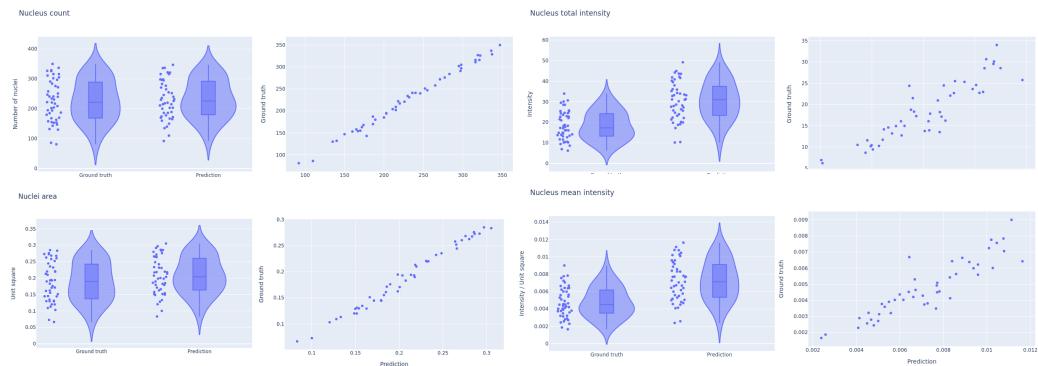


Figure 39: Metrics for downstream tasks on nuclei

Table 4: Correlation coefficients for downstream tasks on nuclei

	Pearson	Spearman
Number of nuclei	0.995	0.994
Total intensity	0.902	.911
Mean intensity	0.907	0.904
Area	0.992	0.990

3.6.2 Influence of different loss functions on metrics for downstream tasks

4 Stability study

4.1 Stability study

4.1.1 Artificial corruptions

Description of artificial corruptions.

Table 5: Hyperparameterization for different artificial corruption severities

Corruption \ Severity	-5	-4	-3	-2	-1	0	1	2	3	4	5
Defocus blur (radius)	-	-	-	-	-	0	0.5	1.0	1.5	2	3
Contrast (gain)	3.5	3.0	2.5	2.0	1.5	1	0.9	0.8	0.7	0.5	0.3
Brightness (bias)	-150	-135	-120	-90	-50	0	50	90	120	135	150

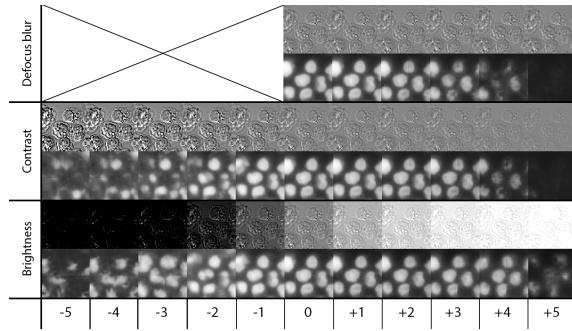


Figure 40: Influence of artificial corruptions on the predictions

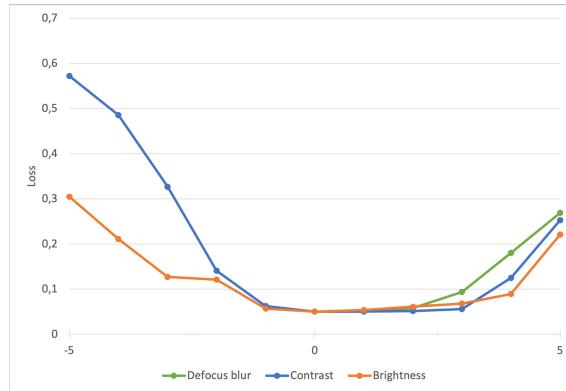


Figure 41: Change of Pearson correlation loss for artificial corruptions

4.1.2 Real corruptions

4.1.2.1 Not fixed cells imaging as corrupted input

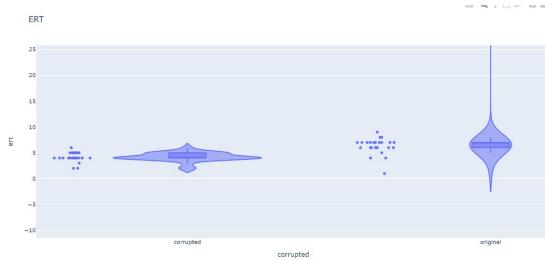


Figure 42: Online drift detection of not fixated cells

Scores of 0.91 however the threshold is 6, not corrupted data (fixed cells) mostly ert of 7 whereas corrupted data (not fixed cells) have an ert of 4. The threshold is therefore 6.

4.1.2.2 Real-world examples of corruptions

4.1.3 Influence of corruptions on metrics for downstream tasks

Calculate how metrics worsen when the evaluation stays the same, but the input is corrupted.

4.1.4 Improving predictions with additional corruption augmentations

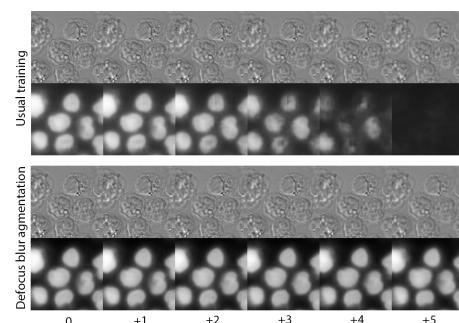


Figure 43: Using corruptions as augmentations improves predictions

4.2 UNET embeddings study

4.2.1 Application of various dimentionality reduction methods

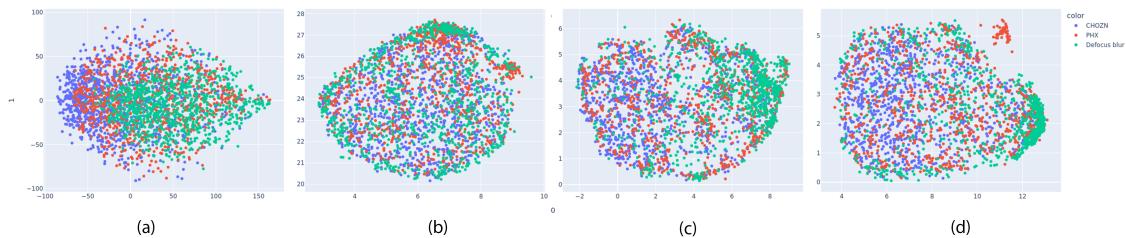


Figure 44: (a) PCA, (b) UMAP, (c) combination of PCA and UMAP with 10 and (d) 50 components

4.2.2 Autoencoder embeddings as an alternative

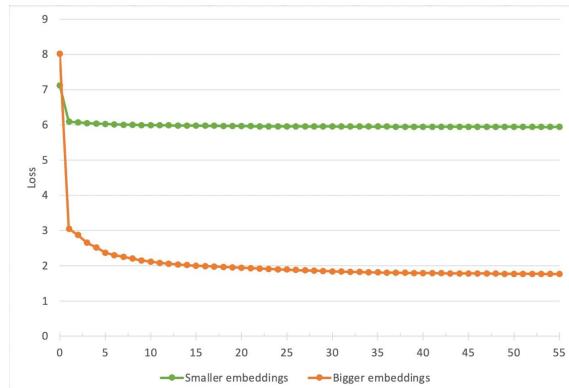


Figure 45: Autoencoders training convergence

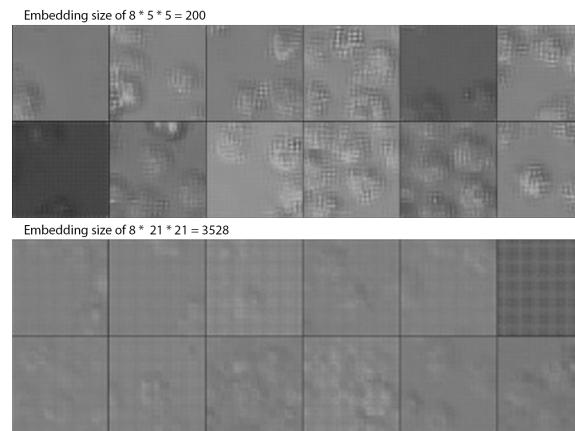


Figure 46: Samples drawn from the trained autoencoder

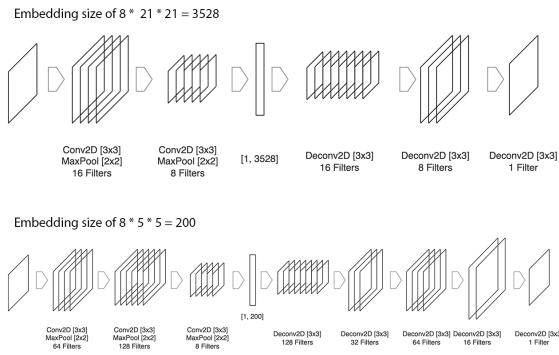


Figure 47: Architectures of two autoencoders

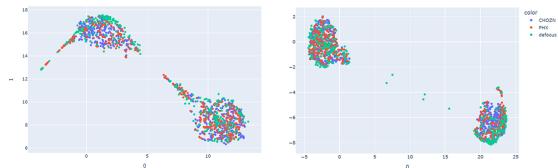


Figure 48: Autoencoder embeddings after applying PCA with 10 components and UMAP afterwards. Earlier epoch VS later epoch

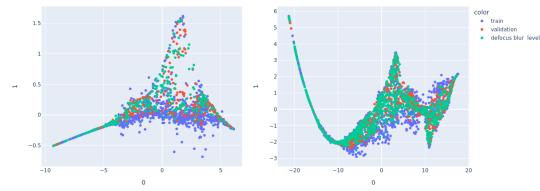


Figure 49: PacMAP does not provide information on the corruption

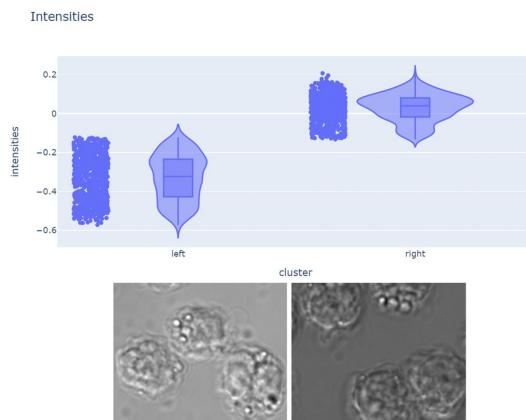


Figure 50: What do two UMAP clusters represent

4.2.3 Clustering of PacMAP embeddings

4.2.3.1 Clustering on UNet embeddings

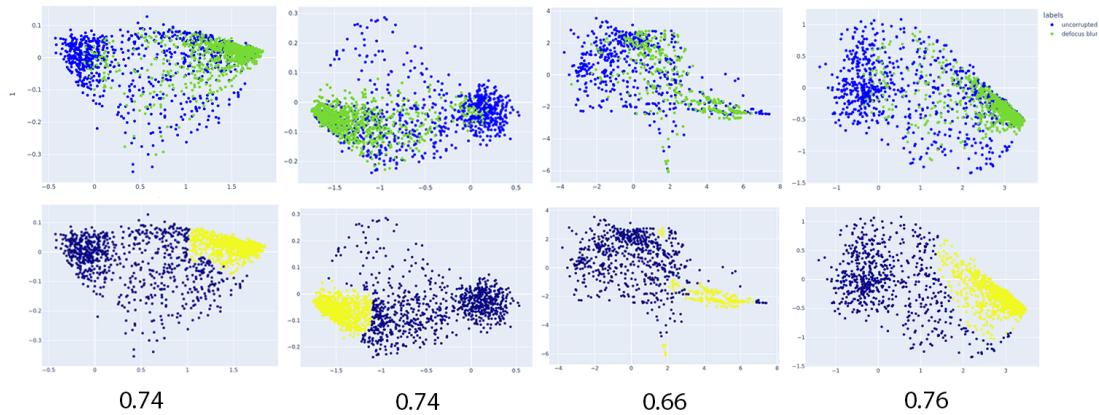


Figure 51: Clustering of UNet embeddings after PacMAP

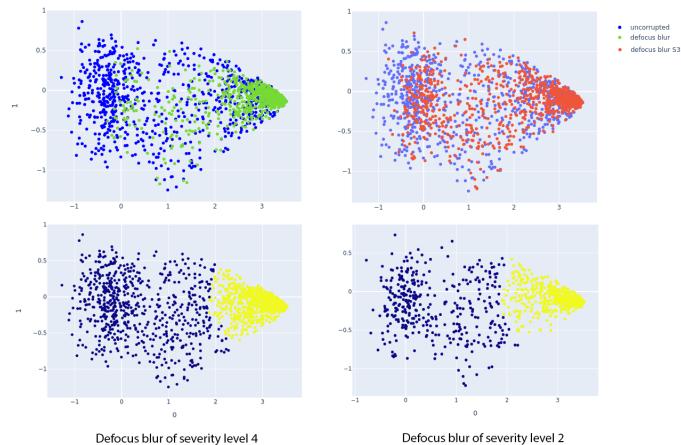


Figure 52: Clustering of UNet embeddings after PacMAP for different severities levels

TABLE with F1-score: 0.76 VS 0.64

4.3 Drift detection

4.3.1 A need to detect drift

4.3.2 Maximum mean discrepancy for drift detection

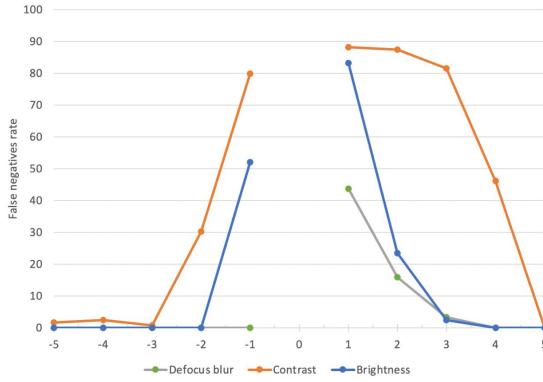


Figure 53: False negatives rate for drift detection on artificial corruptions

4.3.3 Online version of MMD algorithm

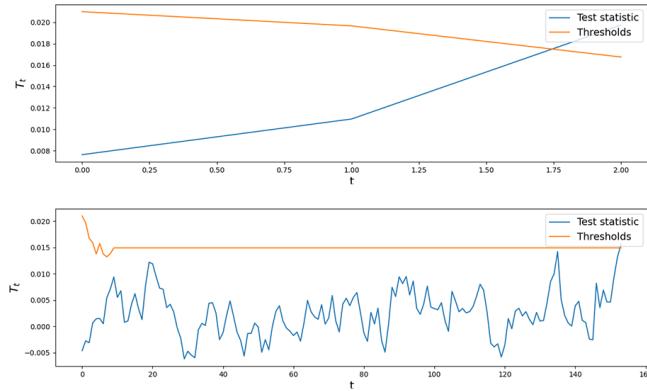


Figure 54: Expected runtime (ERT) for corrupted and in-distribution data

Table 6: Test window size influence on separability

W	2	5	10	15	20
Auc-Roc	0.85	0.92	0.98	0.90	0.88

Table 7: ERT influence on separability

W	32	64	128	256
Auc-Roc	0.90	0.95	0.98	0.98

Table 8: Severity of corruptions on separability

W	Level 2	Level 3	Level 4
Auc-Roc	0.84	0.92	0.98



Figure 55: AUC ROC scores for various defocus corruptions severities

5 Software Tools

5.1 Foundry. Palantir

5.2 AWS

5.3 Streamlit

6 Future research

" One limitation of our current work is that it is based on fixed cells that does not allow longitudinal imaging. This can be overcome by using fluorescent reporter cell lines or live cell dyes to provide the fluorescence ground truth (10) and enable dynamic observation. Another limitation of the DL framework we used here is that it cannot be generalized to different types of cells. Techniques based on transfer learning (<https://doi.org/10.1038/s41551-019-0362-y>, <https://downloads.spj.sciencemag.org/bmef/2020/9647163.pdf>) and domain adaptation (38) will be investigated in our future work to overcome this limitation." TODO rephrase <https://www.science.org/doi/10.1126/sciadv.abe0431>

7 Summary

List of Figures

1	CLD process steps	5
2	Dropout	7
3	Way in which photos of the well-plate were taken	8
4	No overlap	8
5	30 pixels overlap	8
6	Unet	9
7	Using original image for rotation and scaling augmentations	10
8	Nuclei training without and with custom weight initialization	10
9	Overfitting	11
10	Different lightning conditions	12
12	Local vs. Global thresholding (normal conditions)	12
11	Local vs. Global thresholding	13
13	Having more data makes training more stable	13
14	With regularization and augmentations PCC	14
15	No regularization but augmentations	14
16	Difefrent models predictions and scores comparison	14
17	Problems in predictions	15
18	Closely located cells	15
19	Fluorescence segmentation	16
20	ER prediction	17
21	Overfit	17
22	No overfit with augmentations	18
23	ER prediction	18
24	Golgi enhancement	19
25	Structuring Element	19
26	Rolling Ball	20
27	(a) Vanilla pre-processing with automatic background removal algorithm only; (b) Additional clipping of lower intensities after vanilla pre-processing; (c) masked or subfigure (a); (d) mask of subfigure (b)	20
28	Straightforward training doesn't work	20
29	Training on original data	21
30	Full size predictions	21

31	Training on the enhanced data	21
32	Asymmetrical training	22
33	Asymmetrical training predictions	22
34	Converting GFP to a binary mask	23
35	Training with BCE loss	23
36	Training with Pearson correlation loss	23
37	Downstream metrics	24
38	GFP, Nuclei and ER combined	24
39	Metrics for downstream tasks on nuclei	25
40	Influence of artificial corruptions on the predictions	26
41	Change of Pearson correlation loss for artificial corruptions	26
42	Online drift detection of not fixated cells	27
43	Using corruptions as augmentations improves predictions	27
44	(a) PCA, (b) UMAP, (c) combination of PCA and UMAP with 10 and (d) 50 components	28
45	Autoencoders training convergence	28
46	Samples drawn from the trained autoencoder	29
47	Architectures of two autoencoders	29
48	Autoencoder embeddings after applying PCA with 10 components and UMAP afterwards. Earlier epoch VS later epoch	29
49	PacMAP does not provide information on the corruption	30
50	What do two UMAP clusters represent	30
51	Clustering of UNet embeddings after PacMAP	31
52	Clustering of UNet embeddings after PacMAP for different severities levels	31
53	False negatives rate for drift detection on artificial corruptions	32
54	Expected runtime (ERT) for corrupted and in-distribution data	32
55	AUC ROC scores for various defocus corruptions severities	33

List of Tables

1	Available data for each fo the organelles	9
2	Pearson correlation coefficients for downstream tasks for different scaling factors	16

3	Correlation coefficients for downstream tasks	24
4	Correlation coefficients for downstream tasks on nuclei	25
5	Hyperparameterization for different artificial corruption severities	26
6	Test window size influence on separability	32
7	ERT influence on separability	33
8	Severity of corruptions on separability	33