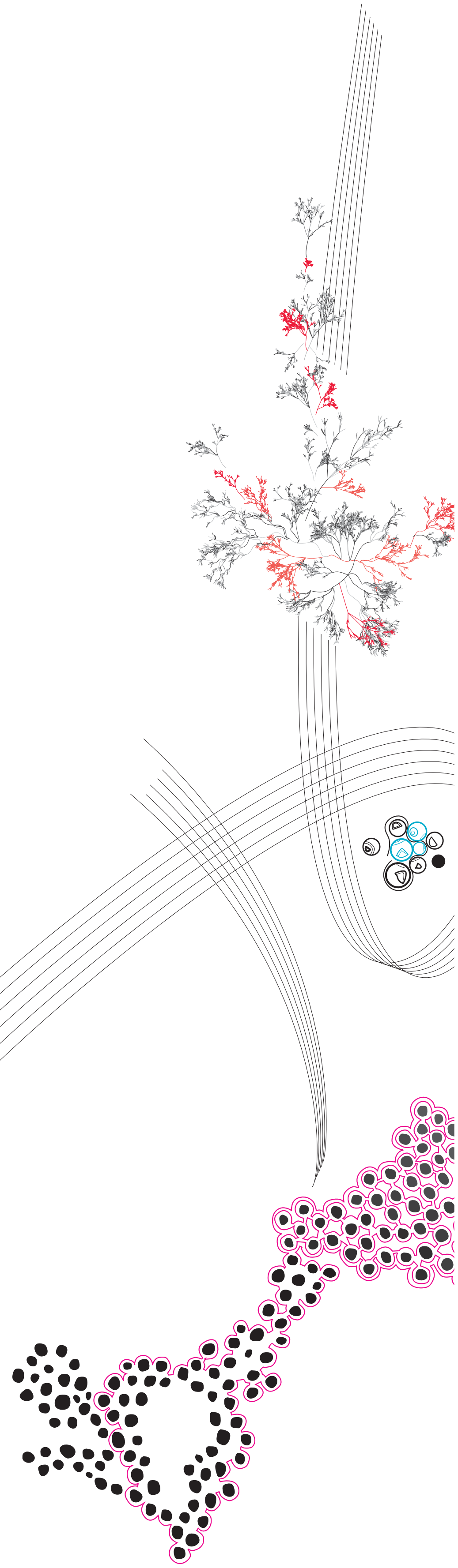
MSc Computer Science
Thesis



A Disentangled Representation Learning Approach for Deblurring Microscopy Images

Stan Ritsema

Graduation Committee
Dr. N. Strisciuglio
Prof. Dr. C. Brune
S. Wang, MSc.
S. Dummer, MSc.

August, 2023

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Acknowledgements

First, I would like to express my gratitude to the supervisor and committee chair of my Master thesis, dr. Nicola Strisciuglio, for overseeing the project and guiding me through the research process.

Besides, I would like to thank prof. dr. Christoph Brune for proofreading my thesis and providing interesting comments and angles to work on.

Moreover, I would especially like to express my appreciation to MSc. Shunxin Wang and MSc. Sven Dummer for their guidance and recommendations. The patience and creativity you put into assisting me during our weekly meetings has kept me interested and enthusiastic about the project.

Finally, I would like to thank my parents and my sister for keeping me motivated and always encouraging me during the past eight months.

Contents

Abstract

Microscopy images are vital for plant growth prediction. These images can be used by prediction models to forecast plant growth. The accuracy of such prediction models are best when the images are clear, e.g. there is no blur present in the images. However, microscope set-ups are often imperfect, which leads to microscopy images with a certain level of blur. Furthermore, some objects might not lie in the same focal plane, causing some objects to be blurry whilst others are in-focus. To remove blur from these images and improve performance of prediction models, we propose a deblurring approach based on Disentangled Representation Learning (DRL). Disentangled Representation Learning is a deep learning technique which attempts to disentangle in the latent space between multiple generative factors underlying a dataset. These generative factors each describe a separate part of the data, like size or colour. This work aims at deblurring microscopy images by disentangling image representations into two latent codes, one encoding blur and one encoding the identity of the image. This disentanglement between blur and identity is beneficial, since it allows direct altering of blur in an image without influencing the identity. The goal of our novel approach is to deblur microscopy images using this disentanglement between blur and identity in the latent space. We compare our approach with another DRL approach named Multi-Level VAE, which functions as a baseline. Furthermore, an ablation study is performed, which evaluates our contribution to earlier work on disentanglement learning. The dataset used for training and testing contains microscopy images of plant cells with different levels of blur.

Keywords: Disentangled Representation Learning, Manifold Learning, ML-VAE, Deblurring, Microscopy Images

Chapter 1

Introduction

In the study of plant growth, visualizing plants on a cellular level can provide a lot of information. During growth, plant cells go through four phases: G_1, S, G_2, and M [3, 16]. During G_1, the first gap phase, the cell prepares itself for the next phase. It functions as a checkpoint on whether the cell is ready to proceed with synthesis. In the synthesis phase (S), the DNA in the cell gets replicated. After the checkpoint in the second gap phase, the cell starts with its mitosis (M) in which the cell divides itself into two cells. Investigating these phases on a cellular level by observing division planes and division rates of the cells provides information about the growth and shape of the plant.

In order to investigate plants on a cellular level, a substantial zoom is needed. Over time, microscopes have been developed to make objects of diminutive size visible to the human eye. Microscopy plays an essential role in observing plant cells, human tissue, and microscopic creatures. Combining microscopes with photography provides the opportunity to create images of the zoomed-in view of a microscope. Therefore, it is possible to get detailed images of plant cells. These images can be used and analysed by a prediction model for plant growth. In order for these prediction models to be as accurate as possible, the images should be as clear as possible. However, microscope setups are often imperfect, leading to a level of blur in most microscopy images.

The field of Deep Learning has shown to be able to mitigate blur in images. Multiple approaches for deblurring images use some form of latent representations. However, these approaches do not use disentanglement learning. We introduce a deblurring method based on Disentangled Representation Learning (DRL). In DRL, an autoencoder is trained, which consists of an encoder and a decoder. Data points in a complex, high-dimensional space, are mapped upon a less complex latent space by the encoder. DRL assumes all data in the dataset is built upon a set of generative factors [20]. The goal of disentanglement learning is creating separation between these generative factors in the latent space. In our approach, we attempt to disentangle between two generative factors: blur and identity.

Whilst some deep learning based deblurring methods like Zhang et al. and Zhao et al. [23, 24] might reach a certain level of disentanglement between blur and identity in their latent representations, they do not fully exploit the possibilities of DRL. Disentangling blur from identity in the latent representation allows altering blur in an image whilst keeping the identity component of the image stable. Moreover, any modification of the blur latent code does not influence the reconstruction of the image's identity if the latent representation is truly disentangled. Therefore, it allows more direct targeting of blur when attempting to deblur an image. In our approach, reaching a state of disentanglement is the main training purpose instead of a by-product from other deep learning techniques, leading to the aforementioned ability to directly change blur in the image whilst preserving the identity.

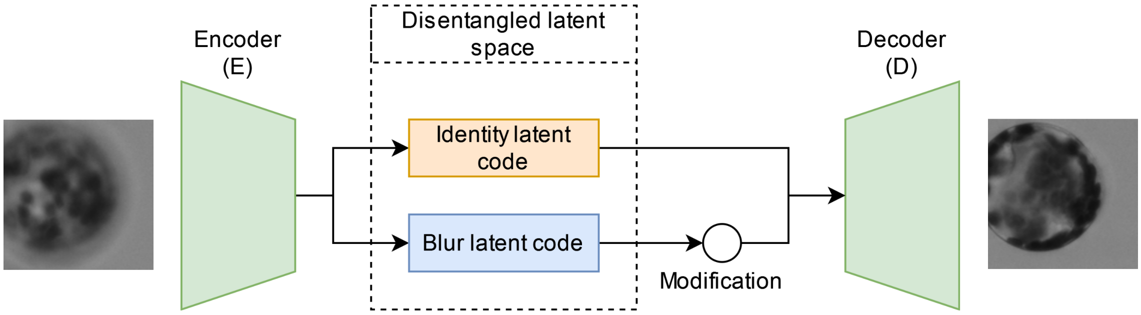


Figure 1.1: Main research idea. The encoder takes a blurry input, which gets encoded into two disentangled latent codes: one for identity and one for blur. The disentanglement allows directly modifying the blur latent code. Afterwards, decoding might lead to an in-focus image

As shown in Figure 1.1, the main idea behind our work is to disentangle between blur and identity in the latent space. First, the encoder E and decoder D are trained to be able to reconstruct an input image. Furthermore, a latent space with two latent disentangled latent codes, one for blur and one for identity, is enforced. This disentanglement allows direct modification of blur in an image without influencing the identity. This work aims to generate an in-focus image from a blurry image by modifying the blur latent code.

Our approach is based on a work on disentanglement using manifold learning [4]. Manifold learning assumes each generative factor is a submanifold. Together, these submanifolds form a manifold underlying the data. Manifold learning attempts to capture each submanifold into a separate latent code. When examining our case at hand, we assume the manifold underlying the images in the dataset can be factored into two submanifolds: blur and identity. Consequently, we disentangle between two latent codes in the latent space, one for blur and one for identity. By using the information at hand in the dataset, our novel approach expands on a work by Fumero et al. on manifold learning. This novel approach is compared to a baseline, for which a Multi-Level Variational Autoencoder (ML-VAE) is used [2]. In order to evaluate and compare these approaches, a dataset containing microscopy images of plant cells with different levels of blur is used.

The baseline comparison shows our novel approach outperforms ML-VAE in terms of reconstruction and deblurring. Besides the baseline comparison, an ablation study is performed to investigate our contribution to the manifold-based approach [4]. Our contribution proves to be vital in the ablation study.

Chapter 2

Background

2.1 Disentangled Representation Learning

DRL is a deep learning method that tries to capture high-dimensional data into a representation of lower, so-called latent dimensions. It is built upon the assumption that the high-dimensional data is generated by a set of generative factors. All of these generative factors are responsible for generating a distinct part of the high-dimensional data. DRL attempts to *disentangle* these generative factors in the latent space.

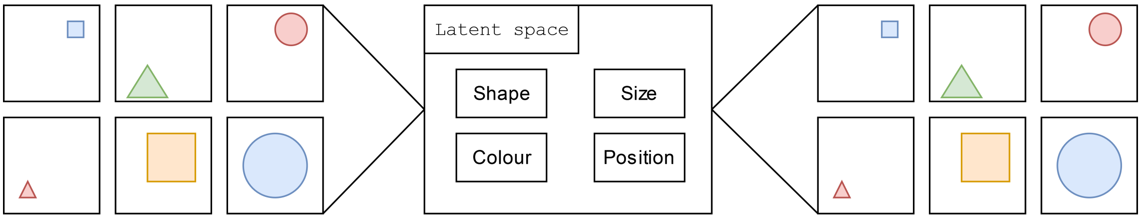


FIGURE 2.1: Example illustration of DRL (Based on Wang et al. [20]). Image dataset with four generative factors: shape, colour, position, and size. Images get encoded into a latent space, which disentangles between these generative factors. Latent codes get decoded back to images.

An example of DRL is presented in Figure 2.1. The high-dimensional image data can be captured by four simple dimensions: shape, size, colour, and position. These dimensions are the generative factors behind the dataset. The goal of DRL is to embed these generative factors appropriately in a latent space. More precisely, DRL splits the latent space into multiple latent codes. Each of these latent codes captures a different generative factor and should only encode information about this generative factor. In the relatively simple example of Figure 2.1, the data is mapped to four latent codes, each encoding a different generative factor. Together, these four latent codes should encode all information. If each latent codes captures only a single generative factor and all information is encoded in the latent codes, the representation is *truly* disentangled.

DRL often uses an encoder-decoder structure. The encoder maps a data point to multiple latent codes and the decoder reconstructs a data point from the different latent codes. The decoder is vital to ensure the latent codes can be combined and decoded to the original input. Once a disentangled representation has been established and the reconstruction quality is good enough, making a change in a certain latent code in the latent representation and reconstructing images can lead to the generation of new data. If the representation is truly disentangled, this allows modification of a single generative factor, without influencing other generative factors. For example, following Figure 2.1, one can adjust the size of the object, without influencing shape, colour, or position.

2.2 Autoencoder

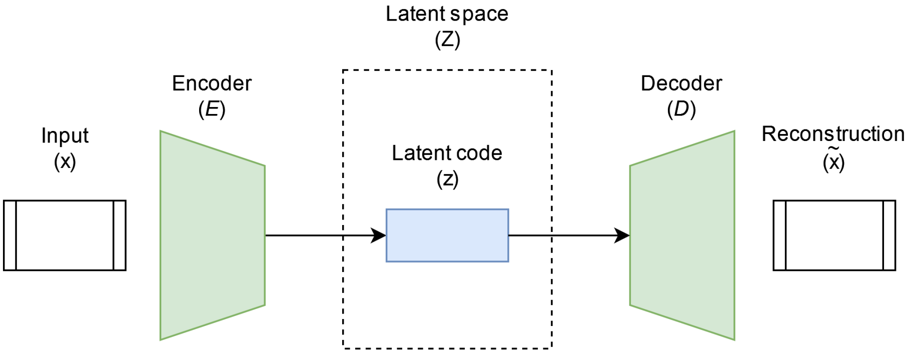


FIGURE 2.2: Example schema of an autoencoder architecture with an encoder E and a decoder D

$$z = E(x) \tag{2.1}$$

As shown in Figure 2.2, an autoencoder is a neural network which consists of two parts: an encoder E and a decoder D. The encoder takes a complex data point x from a dataset X as an input and compresses it to a less complex point in a so-called latent space Z, as shown in Eq. 2.1. This point in the latent space, z, is called a latent code.

$$\tilde{x} = D(z) \tag{2.2}$$

The decoder model does the inverse of the encoder model. It takes a latent code as an input and decodes it into a complex data point $\tilde{x}$, as shown in Eq. 2.2. The goal of the autoencoder is to reconstruct the encoder input by decoding the latent code, as displayed in Eq. 2.3.

$$D(E(x)) \approx x \tag{2.3}$$

This can be achieved during training by comparing the reconstruction of the decoder to the input and using it as the so-called reconstruction loss. Using extra losses alongside the general reconstruction loss, one can enforce certain desirable properties on the latent space, which might be beneficial when modifying the latent codes. For instance, one can attempt to create new complex data points by modifying a latent code and run it through the decoder. This can, for example, provide an option for targeted change in an image, e.g. changing the colour of an object from red to blue.

2.2.1 Variational Auto-Encoder

The Variational Auto-Encoder (VAE) is an extension of a normal autoencoder model introduced by Kingma et al. in 2014 [9]. Instead of encoding inputs as points in the latent space, it encodes inputs as a distribution in the latent space. In order to optimize the generative property of the auto-encoder, the latent space should obtain two important properties:

1) Points which are close together in the latent space should lead to similar generated outputs once fed to the decoder. This increases the interpretability of the latent space. A small change in the latent space should not lead to a massive difference in the generated output.

2) All points in the latent space should be meaningful, e.g. no points in the latent space should decode into meaningless outputs.

$$\mathrm{KL}(q_\phi(z|x)||p(z)) \tag{2.4}$$

Variational Auto-Encoders attempt to maximize these two properties by adding a regularization loss to the VAE objective. The regularization loss ensures the approximated posterior distribution is close to a predefined known distribution. Using this insurance, a point can be sampled from these predefined distributions and fed to the decoder, which results in a new data point. A typical regularization loss is the Kullback-Leibler divergence between the encoded distribution and a normal distribution. This is shown in Eq. 2.4, in which $q_\phi(z|x)$ is the inferred distribution of latent code z given data point x. Furthermore, $p(z)$ is the distribution of the latent code, which is generally fixed to a normal distribution.

$$\mathcal{L}_{ELBO}(x) = (x - \tilde{x})^2 - \mathbb{KL}(q_\phi(z|x)||p(z)) \tag{2.5}$$

Equation 2.5 shows the final loss function which combines the reconstruction loss, the Mean Squared Error in this example, with the Kullback-Leibler divergence. Together, these terms form the Evidence Lower Bound (ELBO).

Chapter 3

Related Work

3.1 Disentangled Representation Learning

3.1.1 Definition

In DRL, there is no generally accepted definition of a disentangled representation. Multiple definitions exist, which show coherency and overlap in their definition of disentanglement.

In 2013, Bengio et al. introduced an informal definition of a disentangled representation. [1]. They stated that a disentangled representation should divide between the generative factors of variation in the data. A resulting latent variable should be sensitive to variation in a single generative factor, whilst being invariant to variation in the other generative factors.

In 2018, Higgins et al. used group theory as the basis for disentanglement [5]. They first define a group action, which they assume can decompose into actions on subgroups. If each action of the subgroups acts on a subset of the latent dimensions and leaves the other subsets fixed, the group action is considered disentangled. Consequently, a representation is disentangled with respect to such a group action if the latent representation decomposes into independent subspaces. In other words, each of these subspaces is only affected by the actions of a single subgroup. This definition for disentanglement is commonly cited in other research as Higgins' definition

In 2016, Fumero et al. defined disentanglement using the manifold underlying the data [4]. They assume that each complex data point lies in the vicinity of a manifold of lower dimensionality. Furthermore, they assume this manifold can be decomposed as a product of submanifolds. In their definition, they consider a representation as being disentangled when a change in one of the latent codes corresponds to a change in exactly one submanifold.

3.1.2 Approaches

Many approaches have been proposed for the realization of DRL. Wang et al. separated these approaches in five categories: *Statistical*, *VAE Based*, *GAN Based*, *Hierarchical* and *Others* [20].

In 2017, Higgins et al. introduced β-VAE, an extension of normal VAEs presented by

Kingma et al. in 2014 [6]. The β parameter was added to regulate the trade-off between the reconstruction capabilities of the VAE and the level of disentanglement.

Building on top of β-VAE, Kim and Mnih introduced Factor-VAE in 2018 [8]. It attempts to minimize the trade-off between reconstruction quality and latent space regularization by stimulating the marginal distribution of representations to be factorial.

Kumar et al. proposed two approaches in 2018: DIP-VAE-I and DIP-VAE-II [10]. They attempt to minimize a distance measure between the inferred prior $q_\phi(z)$ and the disentangled generative prior $p(\mathbf{z})$. The two approaches differ in their disentangling regularizer. DIP-VAE-I only regularizes the covariance between the outputs of the encoder. DIP-VAE-II regularizes the covariance between the approximate posterior $q_\phi(z)$ and the expected posterior $p_\theta(z)$.

In 2017, Bouchacourt et al. introduced the multi-level variational autoencoder (ML-VAE) [2]. In their approach, disentanglement is created between multiple separate latent codes. An example of such different latent codes is the identity and style of a painting.

In 2016, Fumero et al. proposed a disentanglement learning approach based on the principles of manifold learning [4]. They assume all data points share an underlying manifold, which can be divided into submanifolds. Their approach creates disentanglement by learning and separating these submanifolds.

Our novel approach expands upon the work by Fumero et al. by using the information available in the dataset. In the work by Fumero et al., no prior knowledge about the generative factors behind the data is assumed. However, we know the generative factors we want to disentangle for our case. Furthermore, we know which latent code encodes which generative factor. This gives us an edge in enforcing the latent representation to be disentangled to our desire.

3.2 Deblurring

3.2.1 Kernel estimation

One approach at mitigating blur is solving the blind deconvolution problem [17, 15, 7, 13]. This line of work aims at estimating an unknown blur kernel and reconstructing the original picture using this kernel estimation. Computations in this line of work can be costly because of the need to invert the convolution. Furthermore, correctly estimating the underlying blurring kernel is difficult.

Deep learning can be applied to the field of blurring kernel estimation and solving the deconvolution problem [14, 22]. A work by Schuler et al. combined techniques of neural network learning and image deconvolution, resulting in a learning-based approach [18]. Furthermore, Yaar et al. applied deep learning to iteratively improve kernel estimation, whilst applying Super Resolution based on the estimated kernel [21].

3.2.2 Generative model based approaches

Another line of work uses generative model based approaches, which also show promising results in mitigating blur. DeblurGan is a deep learning approach using a conditional GAN proposed in 2018 by Kupyn et al. [11]. An improved version, DeblurGan-v2, was published by the same authors in 2019 [12]. However, these models were mostly trained and tested on images with motion blur, whereas the blur in microscopy images is often out-of-focus blur.

The subject of mitigating blur in microscopy images has also been investigated. In 2019, Zhao et al. published a work in which they use a Convolutional Neural Network (CNN) to mitigate blur. The model was trained and tested on a dataset containing microscopy images and showed promising results. Moreover, they stated solutions to the deconvolution problem are often costly and suboptimal when applied to real-world settings [24].

Furthermore, in 2022, Zhang et al. proposed a CycleGAN approach to mitigate blur [23]. It makes use of the cycle consistency of image transformations. E.g. transforming a winter landscape into a summer landscape and back into a winter landscape should result in the original image one has started the cycle with. Combining two GANs into this cyclic structure proved to be beneficial when mitigating blur.

None of the generative model based approaches use any disentanglement involving blur in their work. Our work investigates whether disentanglement between blur and identity in the latent space allows blur mitigation. This disentanglement might also prove beneficial when combined with these state-of-the-art approaches, since it allows altering blur in an image more directly.

Chapter 4

Method

To achieve our goal of deblurring microscopy images using DRL, we introduce our own novel approach based on manifold learning to disentangle between blur and identity in two separate latent codes. This novel approach is compared with a baseline: an ML-VAE model. This section describes the methodology behind our work, starting with the ML-VAE baseline. Subsequently, we discuss our novel approach by exploring the loss functions and the ideas behind each loss. Furthermore, we discuss the knowledge advantage over the original work by Fumero et al.

4.1 Baseline: ML-VAE

The baseline adopts the techniques of ML-VAE to disentangle between blur and identity [2]. It is used as a baseline to compare our novel manifold-based approach. Multi-level VAE disentangles by splitting the dataset into different groups. The groups are composed of data points which have one shared generative factor and the other generative factor different. This means all images of the same identity, e.g. of the same cell, but with different levels of blur form a group G, where $\mathbf{X}_G$ corresponds to the observations within G. Each observation X_i in $\mathbf{X}_G$ has a shared identity and a unique blur level. In our latent representation, this should be encoded into a shared identity latent code I_G and a unique blur latent code B_i. The shared identity latent code I_G is constructed from encoding all observations in that group and taking the product of their identity density functions, as displayed in Eq. 4.1.

$$q(I_G = i|\mathbf{X}_G = x_G; \phi_i) \sim \prod_{k \in G} q(I_G = i|X_k = x_k; \phi_i) \tag{4.1}$$

In the original work on ML-VAE, taking the product of normal distributions is defined as accumulating group evidence. The latent codes for identity in a group get combined into one identity latent code. This yields disentanglement, since the identity latent distribution is now the shared factor between all images in a group. Furthermore, they assume that with increasing the number of observations in a group, the variance of this grouped latent distribution decreases. The other latent code is the varying factor between all images in a group, which in our case is blur.

4.1.1 Extended VAE objective

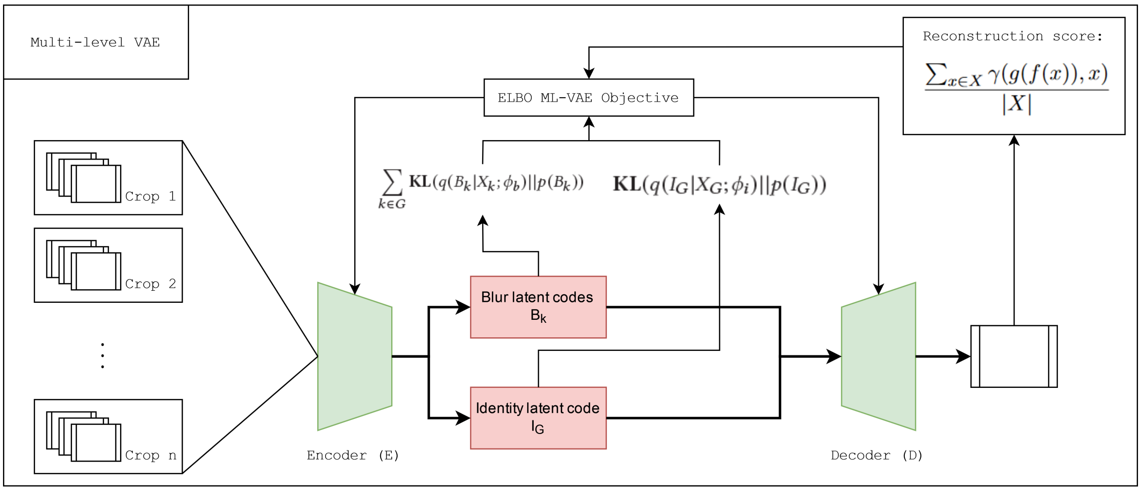

Multi-level VAE
Reconstruction score:
$\frac{\sum_{x \in X} \gamma(g(f(x)), x)}{|X|}$
ELBO ML-VAE Objective
$\sum_{k \in G} \mathbf{KL}(q(B_k|X_k;\phi_b)||p(B_k))$
$\mathbf{KL}(q(I_G|X_G;\phi_i)||p(I_G))$
Crop 1
Crop 2
Crop n
Blur latent codes B_k
Identity latent code I_G
Encoder (E)
Decoder (D)

FIGURE 4.1: Multi-Level VAE setup, extending the normal VAE objective. A group of images sharing their identity get encoded into blur latent codes and identity latent codes. The accumulating evidence technique is used to create a single identity latent code for the entire group. The Kullback-Leibler divergence terms for the blur and identity latent codes combined with the reconstruction score form the ML-VAE objective

During training, as presented in Figure 4.1, the ML-VAE objective consists of three terms: A reconstruction loss, which is further discussed in Section 4.3, and two regularization terms similar to the regularization term in an original VAE setup as discussed in Section 2.2.1.

$$\sum_{k \in G} \mathbb{KL}(q(B_k|X_k;\phi_b)||p(B_k)) \tag{4.2}$$

For the blur latent codes, the regularization term is the group average of the Kullback-Leibler divergence between the prior $p(B_k)$ and the variational approximation $q(B_k|X_k;\phi_b)$, as displayed in Eq. 4.2.

$$\mathbb{KL}(q(I_G|X_G;\phi_i)||p(I_G)) \tag{4.3}$$

Since there is only one latent code for the identity of the group, the regularization term for the identity latent code is just the KL divergence between the true posterior $p(I_G)$ and the variational approximation for the grouped identity code $q(I_G|X_G;\phi_i)$, as shown in Eq. 4.3. Together with the reconstruction loss, these terms form the ELBO (Evidence Lower Bound, Eq. 4.4) which is maximised during training.

$$\begin{aligned}\mathrm{ELBO}(G;\theta,\phi_b,\phi_i) = \frac{\sum_{x \in X} \gamma(g(f(x)), x)}{|X|} - \sum_{k \in G} \mathbb{KL}(q(B_k|X_k;\phi_b)||p(B_k)) \\ - \mathbb{KL}(q(I_G|X_G;\phi_i)||p(I_G))\end{aligned} \tag{4.4}$$

4.2 Novel manifold-based approach

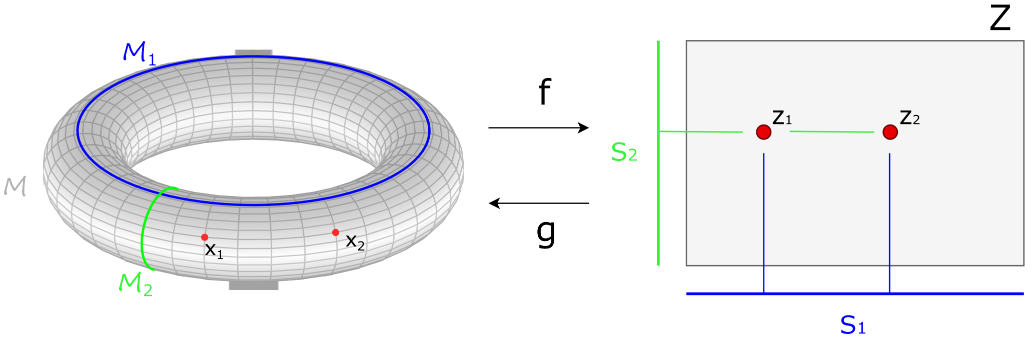


FIGURE 4.2: Example illustration of an underlying manifold. The manifold $\mathcal{M}$ underlies points that lie on a three-dimensional torus. This manifold is constructed by two submanifolds: $\mathcal{M}_1$ and $\mathcal{M}_2$. Function f maps points from the manifold M onto a two-dimensional plane Z. The plane Z can be factored into subspaces S_1 and S_2. These subspaces encode the two submanifolds $\mathcal{M}_1$ and $\mathcal{M}_2$. Image based on Fumero et al. [4]

The field of manifold learning assumes all complex data points in a dataset share a manifold. This low-level manifold $\mathcal{M}$ underlying the data consists of multiple submanifolds $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n$. Take the example presented in Figure 4.2, for which each data point is a point on a three-dimensional torus. The underlying data manifold is factored into two submanifolds, $\mathcal{M}_1$ and $\mathcal{M}_2$. A fundamental notion behind the idea of manifold learning is the independence of submanifolds. In our example, the position in the poloidal direction determined by $\mathcal{M}_2$ (green) does not influence the position in the toroidal direction determined by $\mathcal{M}_1$ (blue). Manifold learning is often put into practice, where the latent space is orchestrated to learn about the submanifolds underlying the data.

Our novel approach uses techniques from the field of manifold learning to disentangle between blur and identity. To implement our approach, we use an encoder-decoder model. The latent space is divided in two latent codes: One latent code encoding the blur, and one latent code encoding the identity of the image. After encoding, the latent codes are concatenated and fed to the decoder. Multiple losses are used in an attempt to create the disentanglement between blur and identity and their latent codes. This section displays the train of thought behind these losses.

$$\mathcal{L} = \mathcal{L}_{rec} + \beta \cdot (\mathcal{L}_{cons} + \mathcal{L}_{cont} + \mathcal{L}_{cross}) \tag{4.5}$$

The final loss function is displayed in Eq. 4.5. The reconstruction loss is discussed in Section 4.3. The other losses are discussed in Sections 4.2.1, 4.2.2, and 4.2.3. The β parameter is discussed in Section 4.2.4.

4.2.1 Consistency Loss

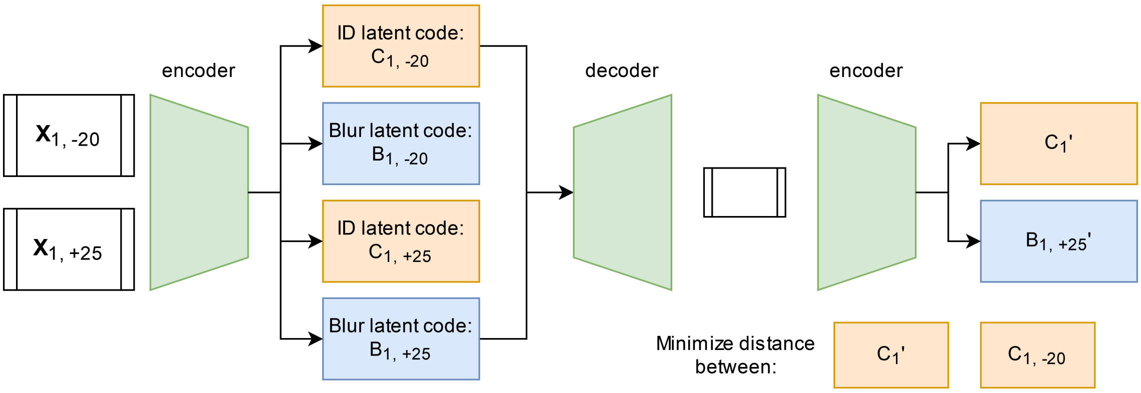


FIGURE 4.3: Consistency loss, same identity, varying z-level. Two images with equal identities get encoded. A change in the blur latent code should not influence the identity latent code. The blur latent code gets changed, combined with original identity latent code $C_{1,-20}$ and decoded. The resulting reconstruction gets encoded again, after which the identity latent code should not have changed. Therefore, the distance between the original identity latent code $C_{1,-20}$ and the new identity latent code C_1' is minimised.

The consistency loss is introduced to ensure the principle of true disentanglement. In a truly disentangled representation, a change in one latent dimension has no influence on the other latent dimensions. Applying it to the problem at hand: if the blur latent code is adapted, the identity of the image should not be influenced. Moreover, if the identity latent code is adapted, the blur in the image should not be influenced. To achieve this, the consistency loss is calculated in three steps, as shown in Figure 4.3. First, we encode two images with one varying factor. Then, the latent code encoding the varying factor is changed. We combine this new latent code for the varying factor with the latent code for the stable factor and decode the combination. Afterwards, the resulting decoding is encoded again into two latent codes. The latent code for the stable factor should not have been influenced by the change. Therefore, the consistency loss measures the latent distance between the original stable latent code and the stable latent code of the re-encoded decoding output. Figure 4.3 only displays the consistency loss with the identity latent code as the stable latent code. In practice, this case is combined with another case in which the blur latent code is the stable latent code and the identity is varied. In this second case, the distance between the original blur latent code and the blur latent code of the re-encoded decoding output is measured.

$$\mathcal{L}_{cons} = \delta(E(\hat{x})_i, z_{1,i}) \tag{4.6}$$

Equation 4.6 shows the consistency loss. Let X_1 and X_2 denote the input images with one varying factor. Z denotes the latent representation for the input images, e.g. $Z_1 = E(X_1)$ and $Z_2 = E(X_2)$, for which E is the encoder. $z_{1,i}$ denotes the latent code which is similar, whereas $z_{1,j}$ denotes the latent code encoding the varying factor between the image pair. $\hat{x}$ is the decoding of $z_{1,i}$ and $z_{2,j}$, e.g. $\hat{x} = D(z_{1,i}, z_{2,j})$. The resulting loss is the Euclidean distance normalized by vector length, δ, between the latent code of the equal factor $E(\hat{x})_i$ and the original latent code of the equal factor $z_{1,i}$.

4.2.2 Contrastive Loss

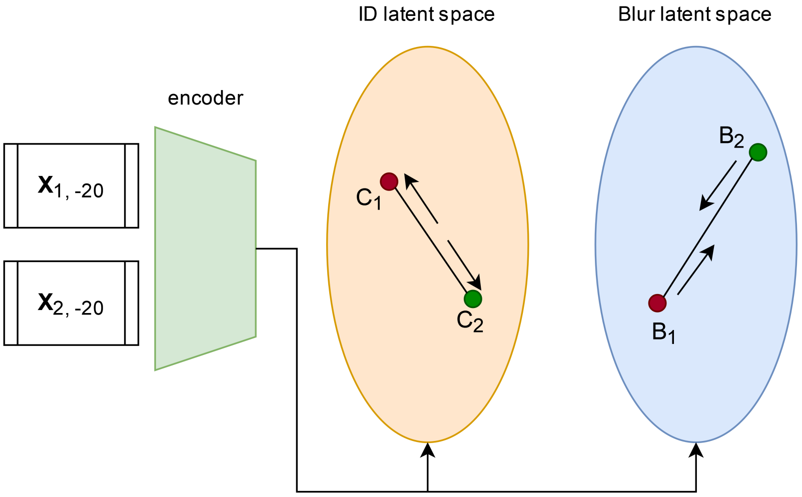


FIGURE 4.4: Contrastive loss: different identity, same z-level. Two images with varying identity but equal blur level get encoded. The contrastive loss makes sure the latent codes in the identity latent space are pushed away from each other, since they are dissimilar. The latent codes in the blur latent space are pushed together, since they should be equal.

The contrastive loss is based on the premise that images that share the same generative factor have equal latent codes representing this factor. On the contrary, images that do not share a generative factor have latent codes representing that factor which are different. These two premises of the contrastive loss can be incorporated in the loss function for the case at hand in the following way.

Firstly, we encode a pair of images with different identities and an equal blur level (Figure 4.4). Since both blur latent codes of the two images should encode the same information, we attempt to push them towards each other by minimising the latent distance between the two latent codes. Since the identities are unequal, the identity latent codes for the two images are pushed away from each other by maximising the latent distance between them.

Secondly, we encode a pair of images with equal identities and a different blur level. For this pair, the identity latent codes for the two images are forced together, whilst the blur latent codes are pushed away from each other.

Equation 4.7 shows the contrastive loss in mathematical notation. We denote the latent code encoding the equal factor as z_e. $\beta = 1$ if the corresponding latent code encodes the equal factor, e.g. $z = z_e$. If not, $\beta = 0$. The first term of the contrastive loss is used to pull equal latent codes together. The second term of the loss is used to push dissimilar latent codes away. We introduce a max distance parameter m which makes sure pushing away latent codes from each other is not unbounded, just like in Fumero et al. For our experiment, m = 0.6. To measure the latent distance, we use the Euclidean distance normalized by vector length δ.

$$\mathcal{L}_{dist} = (1-\beta)\cdot\delta + \beta\cdot max(m-\delta, 0) \tag{4.7}$$

4.2.3 Cross Reconstruction Loss

The cross reconstruction loss is a novel element to the overall training objective, which is not in the work by Fumero et al. It is introduced to ensure a change in one of the latent codes results in expected behaviour in terms of reconstruction. Such a latent code change should not influence the other factor in terms of the resulting reconstruction. For example, a change in the blur latent code should not influence the location of the cell, which is part of the identity. Besides not influencing the other generative factor, a latent code change should result in a reconstruction that reflects the change in the latent code correctly. For example, changing the blur latent code to a different Δ z-level should reconstruct the input image at the new Δ z-level.

In order to calculate the cross reconstruction loss, we use a three-fold data traversal. For example, we encode the pair of images $X_{1,-20}$ and $X_{1,+25}$ with the same identity (C_1) but different Δ z-levels (-20 and +25). This encoding results in identity latent codes $C_{1,1}$, $C_{1,2}$ and blur latent codes $B_{-20,1}$, $B_{+25,2}$. Since their identity is equal, we can swap out their blur latent codes and expect the image reconstructions produced by the decoder D to also be swapped when compared to the input images. $D(C_{1,1}, B_{+25,2})$ should decode to $X_{1,+25}$ and $D(C_{1,2}, B_{-20,1})$ to $X_{1,-20}$. The eventual loss for this example can be calculated as listed in Eq. 4.8, in which γ is the reconstruction loss explained in Section 4.3:

$$\mathcal{L}_{cross} = \gamma(D(X_{1,1}, B_{+25}), X_{1,+25}) + \gamma(D(X_{1,2}, B_{-20}), X_{1,-20}) \tag{4.8}$$

Just like $\mathcal{L}_{cons}$ and $\mathcal{L}_{dist}$, the same loss is also calculated for a pair of images with equal Δ z-level but different identities, as displayed in Figure 4.5.

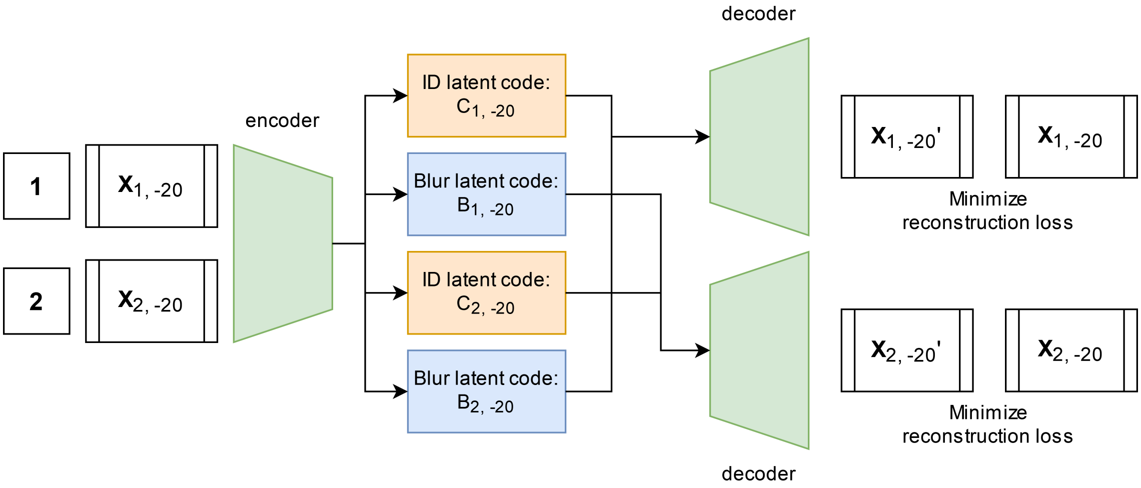

decoder
encoder
1
$X_{1,-20}$
2
$X_{2,-20}$
ID latent code: $C_{1,-20}$
Blur latent code: $B_{1,-20}$
ID latent code: $C_{2,-20}$
Blur latent code: $B_{2,-20}$
$X_{1,-20}'$
$X_{1,-20}$
Minimize reconstruction loss
$X_{2,-20}'$
$X_{2,-20}$
Minimize reconstruction loss
decoder

FIGURE 4.5: Cross recon loss procedure. Two images with different identities but equal blur level get encoded. Their blur latent codes get swapped during decoding. Since the blur latent codes are equal, this swap should not influence the reconstruction. Therefore, the reconstructions are compared to the original input

4.2.4 Delaying losses

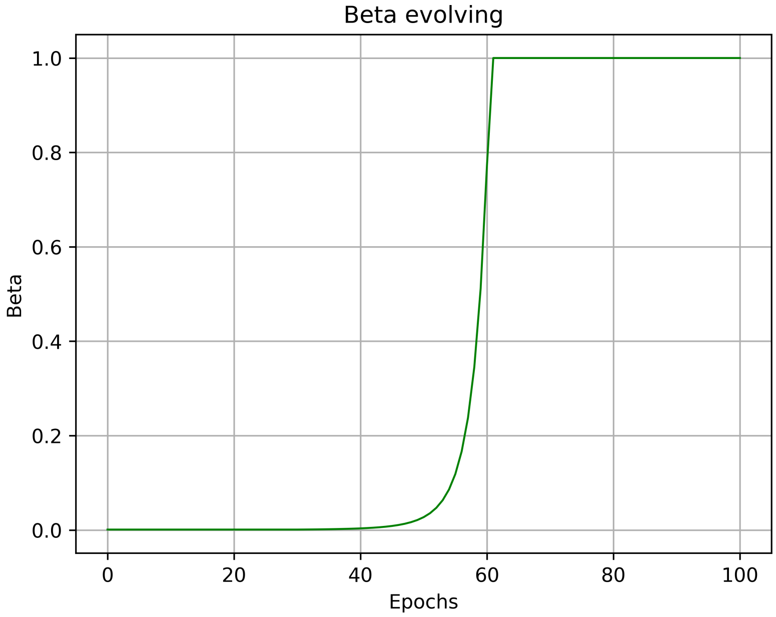


FIGURE 4.6: Delaying losses using β parameter over time. β parameter exponentially increases after 30 epochs are used

Another technique that is used in Fumero et al. is delaying the distance-based losses. At the start of the training procedure, the latent spaces for blur and identity are unstructured. The relative distance between points in the latent space does not yet indicate whether these points are similar or not. Therefore, calculating a distance within these latent spaces is meaningless. This makes the distance-based losses meaningless to some extent. We can *delay* the distance-based losses by adding a parameter β, which increases exponentially from 0 to 1 after approximately 30 epochs, as shown in Figure 4.6.

4.2.5 Knowledge advantage over normal manifold projection

In Fumero et al. no prior knowledge about which latent code encodes which generative factor is assumed [4]. This touches upon an important distinction between Fumero et al. and this work. In the case at hand, much more information about the input data is given, which can be used to our advantage when attempting to disentangle in the latent space. First, we know which latent code encodes which generative factor, because we simply assign each of the generative factors we want to disentangle between to their own latent code. This gives us the possibility to calculate the losses in Fumero et al. more directly.

For example, for the consistency loss, they encode two images with one different generative factor. They assume of the existence of some oracle O that tells exactly which latent code captures the difference between the two images. In Fumero et al., the oracle is estimated by encoding this pair of images. Thereafter, they measure the distance between the pair for each latent code and select the latent code which has the highest distance to calculate the consistency loss. This method is not flawless, since we do not know for sure whether the correct latent code is chosen. In our setup, we know exactly which latent code captures which generative factor. Because of this knowledge advantage, we are completely sure which latent code captures the change introduced in calculating the consistency loss.

Furthermore, the knowledge advantage together with the nature of the dataset enables the usage of the cross reconstruction loss. We can make a change in the blur latent space and know what the output of the reconstruction should look like. Furthermore, there is a ground truth for almost all of these changes in the blur latent space, since we have images at many different Δ z-levels. The added value of this cross reconstruction loss is examined in Section 5.3.

4.3 Reconstruction Loss

For the reconstruction loss in both the ML-VAE approach as our novel manifold-based approach, we use a pre-trained VGG-16 model. The VGG-16 model was introduced by Simonyan and Zisserman in 2015 [19]. Originally, this model was trained for image recognition, e.g. localising objects within images and classifying images. However, the model can also be used to calculate an image similarity score between two images. During training, we experienced a vast improvement in terms of reconstruction quality when using the VGG-16 model compared to using the L1-norm or L2-norm as the reconstruction loss. The loss function can be expressed as Eq. 4.9, in which γ is the VGG-16 model score, E is the operation by the encoder and D is the operation by the decoder.

$$\mathcal{L}_{recon} = \frac{\sum_{x \in X} \gamma(D(E(x)), x)}{|X|} \tag{4.9}$$

4.4 Evaluation metrics

4.4.1 Disentanglement evaluation

To investigate the level of disentanglement between blur and identity reached by the models, we perform a quantitative evaluation using the ML-VAE score.

ML-VAE score

The quantitative method used to evaluate the level of disentanglement achieved by a model is the ML-VAE score. It is a technique presented in the work on ML-VAE [2]. The concept revolves around the level of information about the grouped latent code (identity latent code) in the observation-specific latent code (blur latent code). One should not be able to identify an image's identity based on its blur latent code.

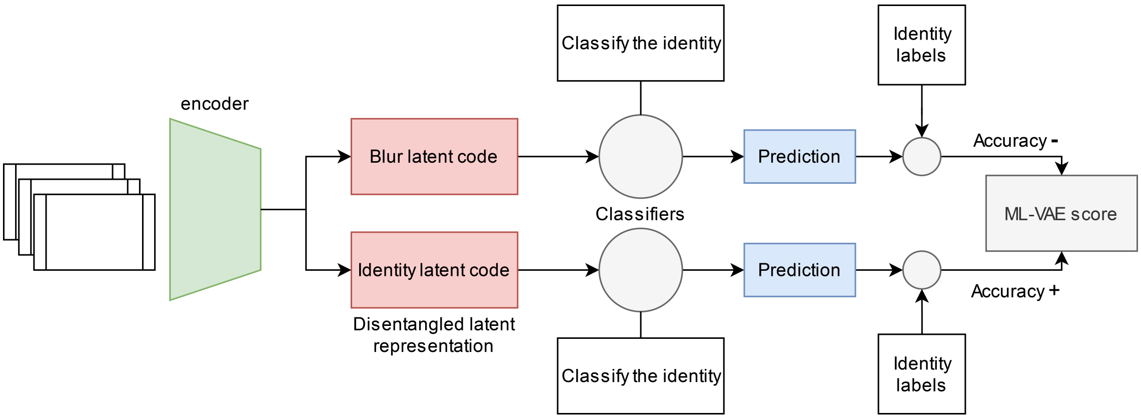


FIGURE 4.7: ML-VAE score calculation procedure

The calculation of the ML-VAE score is displayed in Figure 4.7. A set of images $\boldsymbol{X}$ with identities $\boldsymbol{I}$ get encoded by the encoder. Two classifiers are trained for 300 epochs, one for the identity latent codes $\boldsymbol{C}$ and one for the blur latent codes $\boldsymbol{B}$. The classifiers take the latent codes as an input and attempt to predict the identity of the images. Logically, the prediction based on the identity latent codes should be as accurate as possible, whereas the prediction based on the blur latent codes should be as inaccurate as possible. The accuracies for the predictions combined lead to the ML-VAE score, for which the identity predictions are a positive influence and the blur predictions are a negative influence. This leads to the formula displayed in Eq. 4.10.

$$\text{ML-VAE}_{score} = \frac{\sum_k^{|\boldsymbol{I}|} P_{id}(C_k) = I_k}{|\boldsymbol{I}|} - \frac{\sum_j^{|\boldsymbol{I}|} P_{blur}(B_j) = I_j}{|\boldsymbol{I}|} \tag{4.10}$$

4.4.2 Reconstruction evaluation

To measure the reconstruction capability, the model takes an input image, encodes and decodes it. The resulting reconstruction is compared to the original input image in terms of their Peak Signal-To-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), which measure image similarity. Furthermore, one can visually observe the quality of the reconstructions.

4.4.3 Deblurring evaluation

To evaluate the deblurring capabilities of the different approaches, we use the following deblurring strategy.

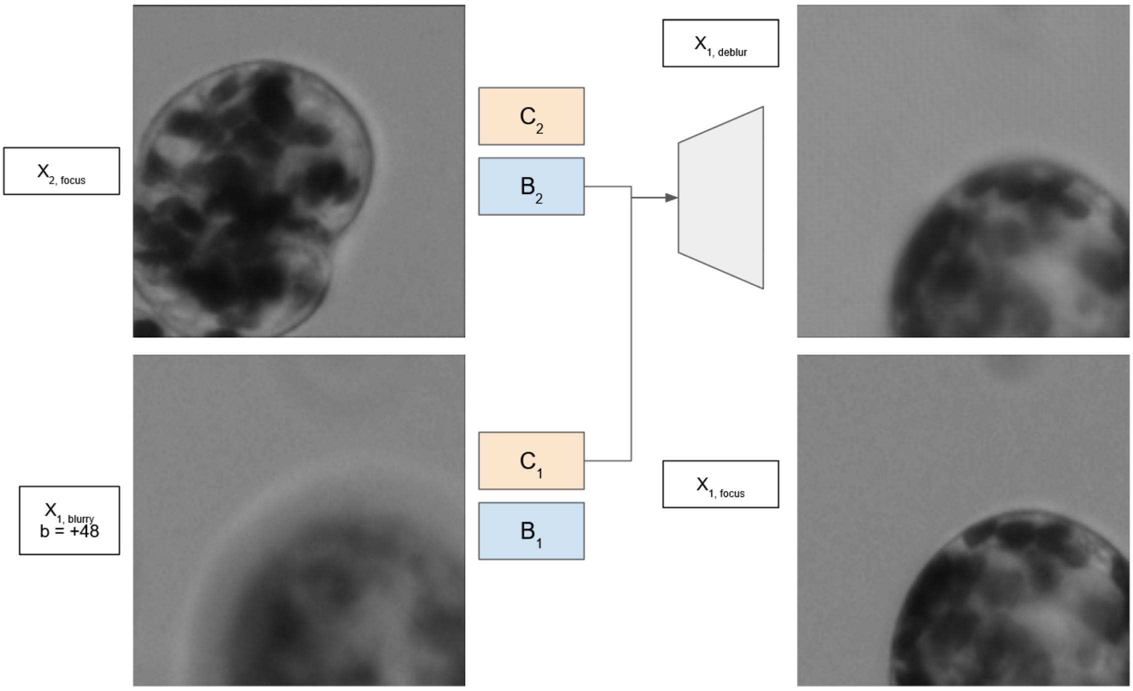


FIGURE 4.8: Deblurring procedure

First, a blurry image $X_{1-blurry}$ is sampled from the dataset. This is the image we attempt to deblur. Secondly, we sample a random in-focus image $X_{2-focus}$ with a different identity.

These images are encoded into B_1, C_1 and, B_2, C_2 respectively. In an attempt to deblur, we combine the identity latent code of the blurry image with the blur latent code of the in-focus image. This combination gets decoded. In order to evaluate the performance, we compare the reconstruction $(X_{1-deblur})$ with the target in-focus image $X_{1-focus}$. The reconstruction is compared to the in-focus image using the PSNR and SSIM, similar to the reconstruction evaluation.

Chapter 5

Experimental Results

5.1 Experiment setup

5.1.1 Dataset

The training and testing dataset consists of bright-field microscopy images of dividing protoplast plant cells. These images were taken daily under the same circumstances in terms of lighting, room temperature, zoom-level (20x) and microscope used. However, the cells depicted on the image vary each day. Each day, 100 images are taken with varying z-levels. For each day, one of the images is marked as the in-focus image. Furthermore, a projection image is generated which combines all z-levels into one where the most cells are in-focus.

Data preprocessing

The original images contain multiple cells and are of size 2048×1536. As a first preprocessing step, the images are downsized to a size of 1024×768. In order to have a feasible input dimension for the neural network, the images are cropped to 48 images of 128×128. For each crop, we have images at z-levels ranging approximately from 50 z-levels below the in-focus plane and 50 z-levels above the in-focus plane. The preprocessing results in the following folder structure:

```
Data
└─ Brightfield
    ├─ 23-08-2022-location-17
    │   ├─ crop-1
    │   │   └─
    │   │       ├─ -40.tif
    │   │       ├─ -39.tif
    │   │       ├─ -38.tif
    │   │       └─ ...
    │   ├─ crop2
    │   │   └─
    │   │       ├─ -40.tif
    │   │       ├─ -39.tif
    │   │       ├─ -38.tif
    │   │       └─ ...
    │   └─ ...
    └─ ...
```

5.1.2 Model Architecture

The model architecture for both approaches (Sections 4.1 and 4.2) is largely kept equal. In both cases, a convolutional neural network existing of an encoder and decoder is used.

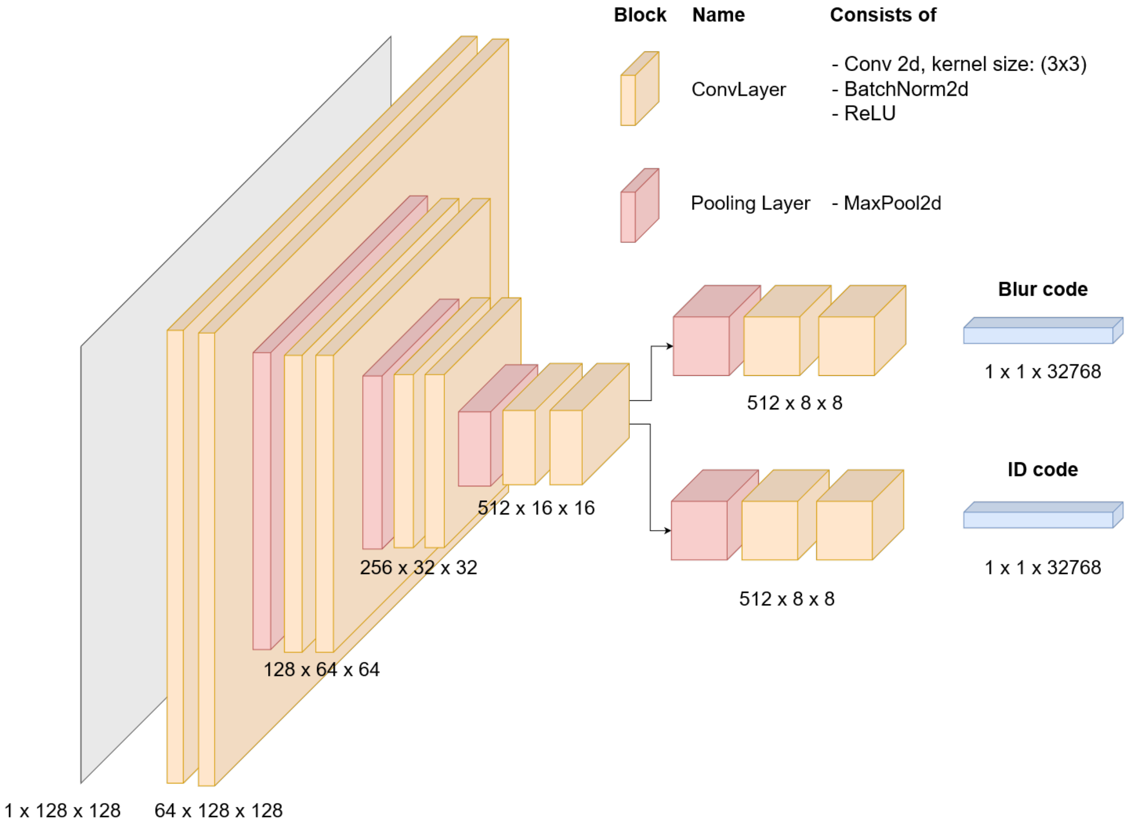


FIGURE 5.1: Encoder architecture

Figure 5.1 shows the architecture used for the encoder. For our purpose, we use convolutional layers and pooling layers.

The convolutional layers use a kernel size of 3×3, with a stride of 1. After the convolutional layer, batch normalization is applied to speed up the training process. Moreover, a ReLu activation function is used. Besides these convolutional layers, pooling layers using max pooling are used to reduce dimensions. Combining a pooling layer and two convolutional layers, a convolutional *block* is formed.

The input (1×128×128 dimensional) goes through three of these blocks, after which the encoder splits into an identity part and a blur part. These separate parts both consist of another convolution block. This results in two representations with a dimension of 512×8×8. These are flattened to two latent codes of 32768 dimensions, one for identity and one for blur.

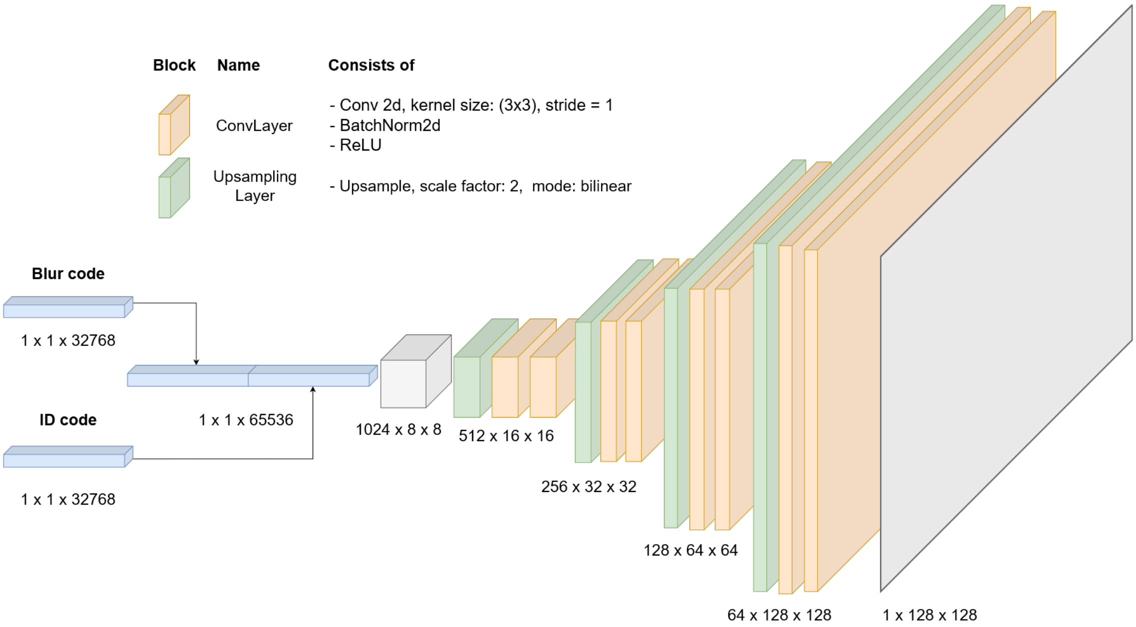


FIGURE 5.2: Decoder architecture

The decoder takes two 32768-dimensional latent codes, one for blur and one for identity. These latent codes get concatenated and reshaped into a 1024×8×8 shape. Four blocks follow, each consisting of a bilinear upsampling layer with scale = 2 and two convolutional layers. The convolutional layers are the same as in the encoder architecture. The final convolutional layer in the final block reconstructs the original 1×128×128 input.

5.1.3 Training setup

The models are trained on the JupyterLab instance of the University of Twente. For both models, a Jupyter notebook is created. The models are trained on PyTorch v1.13.1 using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate for the optimizer is 0.001, but is decreased during training using a learning rate scheduler. The learning rate is multiplied by 0.98 every five epochs. Each model is trained for 400 epochs in total. The dataset is split into a training set and a test set, with 150 and 50 identities respectively. For each identity, there are approximately 100 images at different z-levels, resulting in 15,000 images in the training set and 5,000 images in the test set.

5.2 Baseline comparison: ML-VAE and novel approach

The first experiment is a comparison between the baseline (ML-VAE) and our novel manifold learning based disentanglement approach. For the manifold learning based approach, we investigate two versions: A standard approach in which both the blur and identity latent code are 32678-dimensional and an approach in which the blur latent code is one-dimensional. The reasoning behind the one-dimensional latent code can be found in Section 5.2.1.

5.2.1 One-dimensional blur latent code reasoning

In the dataset at hand, the different levels of blur are described by the difference in z-level from the in-focus plane. This Δ z-level is one-dimensional. Therefore, it would make sense to encode the level of blur in an image into a one-dimensional latent code, e.g. just a single number. This one-dimensional blur latent code would provide two desirable properties.

First, the level of information about the identity of an image present in the blur latent code is minimal. Because of the single latent dimension, it simply cannot hold information on the identity of the image. Therefore, the disentanglement between blur and identity might be better.

Second, the one-dimensionality of the latent code ensures linearity in the latent space. Since the latent code is one-dimensional, a correct linearity when graphing latent codes by their Δ z-level is ensured if the principle behind the contrastive loss is implemented correctly. This might prove to be beneficial during linear interpolation and deblurring.

5.2.2 Reconstruction capabilities

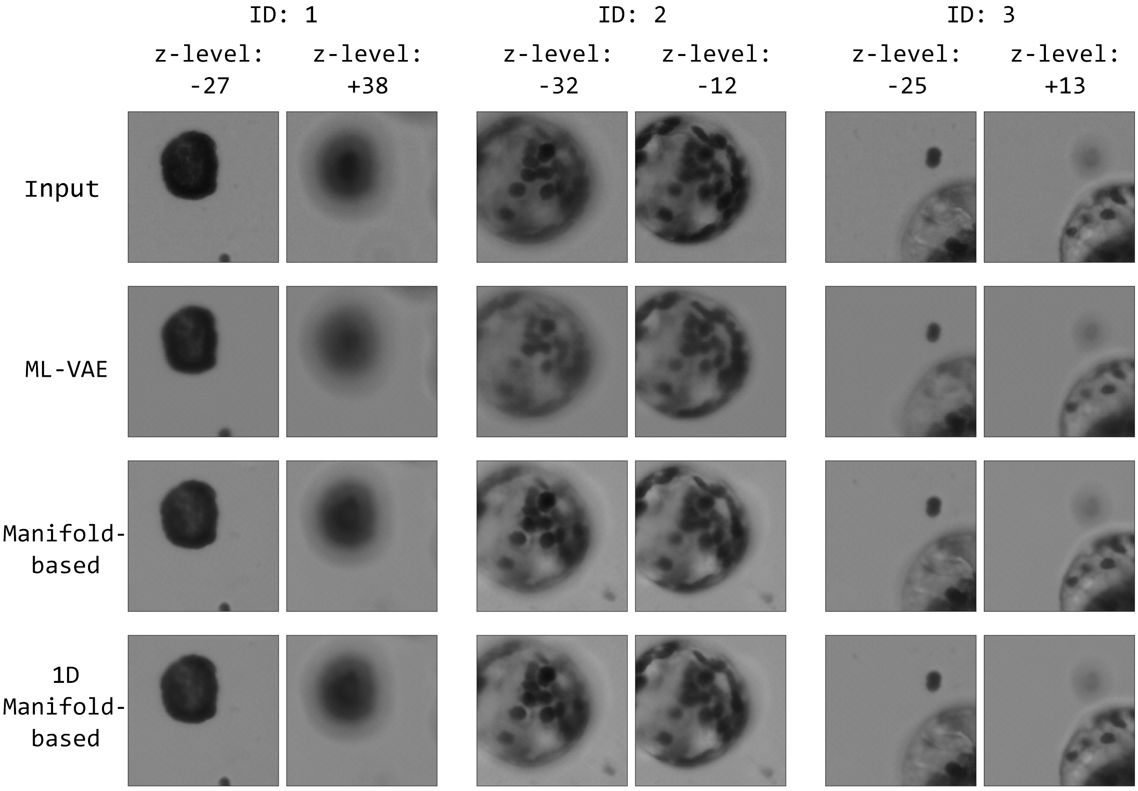


FIGURE 5.3: Reconstructions ML-VAE, manifold-based model. Top row: input, second row: ML-VAE, third row: manifold-based, bottom row: 1D manifold-based. Two images of equal identity but different blur levels in each column. Images from the training data

Figure 5.3 shows the reconstruction capabilities of the ML-VAE model, manifold-based model and 1-dimensional blur latent code manifold-based model. For ML-VAE, the overall reconstructions shown in the second row are fine, but not spectacular. The third and fourth images shows that some detail is lost in the reconstructions if the input image is a complex cell with lots of detail. Furthermore, for the detailed images, the reconstructions are a bit less outspoken. The black colour gets a more greyish tone.

The reconstructions generated by the manifold-based approach displayed in the third row are good. The colours are more vibrant than the ML-VAE approach, the grey-out effect is not as prevalent. Furthermore, the complex images in the third and fourth column show a higher level of detail than the ML-VAE approach. Another interesting facet to notice for the images in the third and fourth column is the noise that occurs in the bottom-right corner.

The reconstructions for the 1-dimensional blur latent code model shown in the bottom row are good. The quality is similar to the normal manifold-based model's reconstruction quality, and the level of detail in the cell is comparable. This shows the model can still reconstruct complex images, even though the blur latent code is trimmed down to a single dimension.

Model	Test set		Training set	
	SSIM	**PSNR**	**SSIM**	**PSNR**
ML-VAE	0.9101	31.923	0.9233	32.279
Manifold-based	0.9455	37.927	0.9462	38.455
1D manifold-based	0.9422	36.560	0.9444	37.437

Table 5.1: Quantitative evaluation of the reconstruction capabilities of the different approaches

Table 5.1 shows the average SSIM and PSNR score as a measure of reconstruction performance for both the test and training set. We can observe that the ML-VAE model is outperformed by the manifold-based approaches in terms of reconstruction. Furthermore, the 1D manifold-based model performs slightly worse than the normal manifold based model. This can be explained by the massive size difference of the blur latent code between the normal manifold-based approach and the 1D manifold-based approach. Because of this size difference, the overall level of information in the blur latent code might be higher. Therefore, the reconstructions might be slightly better. However, the 1D manifold-based model might disentangle better.

5.2.3 Disentanglement evaluation

One of the goals for this research is the true disentanglement between blur and identity in the latent space. In order to evaluate the model's disentanglement, we evaluate the level of disentanglement between the latent spaces quantitatively, as described in Section 5.2.3. Furthermore, we graph multiple latent codes in the latent spaces to visually investigate the structure of the latent space.

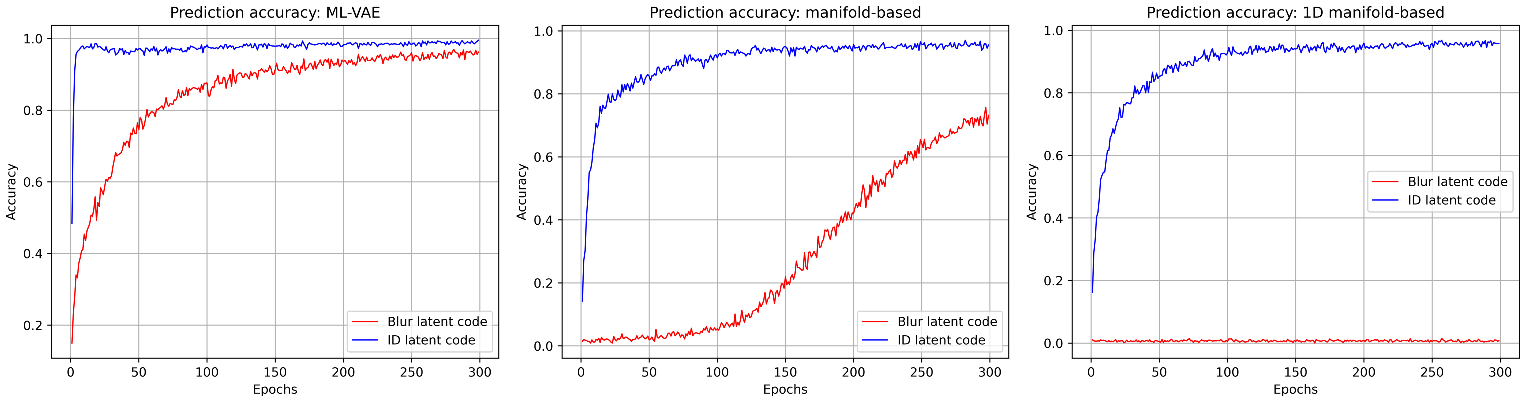
Model	Blur latent code accuracy	Identity latent code accuracy	ML-VAE score
ML-VAE	0.962	0.995	0.033
Manifold-based	0.732	0.955	0.263
1D manifold-based	0.007	0.958	0.951

Table 5.2: ML-VAE score for three approaches: ML-VAE, manifold-based and 1D manifold-based

Table 5.2 shows the accuracy of predicting the identity of an image based on its latent codes. For the ML-VAE model, both the identity latent code as the blur latent code are good predictors for the identity. The blur latent code still holds a lot of information about the identity of the image, which is not good. Therefore, the ML-VAE score for the ML-VAE approach is relatively low.

The manifold-based approach shows some improvement, seeing as the blur latent code holds less information about the identity of the image than in the ML-VAE approach. Resulting, the ML-VAE score is higher for the manifold-based approach. However, the blur latent code still holds information about the identity, as the blur predictor for the manifold-based approach still reaches an accuracy of 0.732 after 300 epochs.

The one-dimensional manifold-based approach proves to be the best, as the ML-VAE score is the highest. The accuracy for the blur latent code predictor is extremely low. This makes sense, because the single dimension cannot encode something as complex as identity.



(a) ML-VAE (b) Manifold-based (c) 1D manifold-based

FIGURE 5.4: ML-VAE score predictors' accuracy over time. Left: ML-VAE. Middle: manifold-based. Right: 1D manifold-based

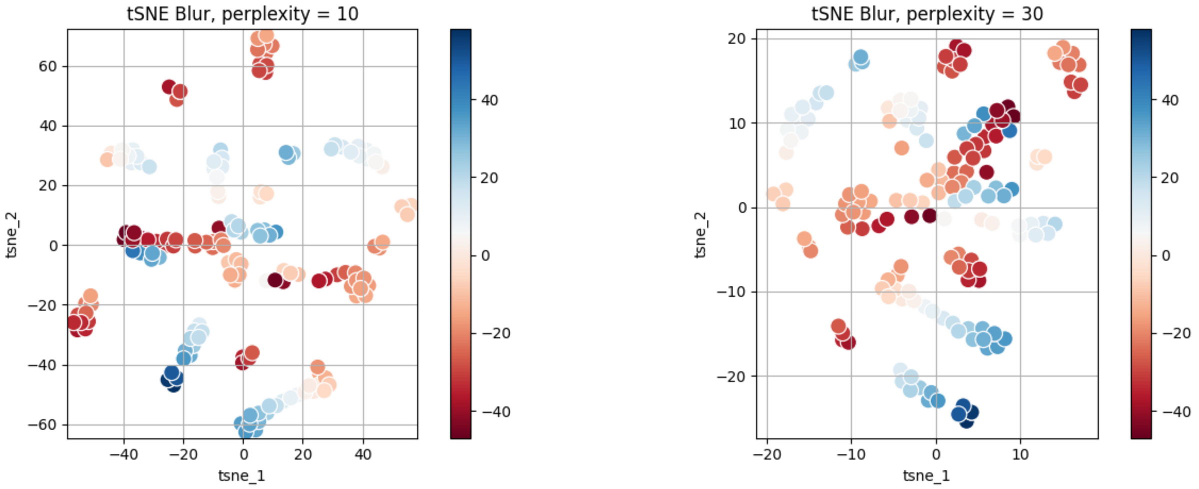
Figure 5.4 shows how the accuracy of the predictors evolve over time during training. We can observe that the blur latent code predictor of the ML-VAE approach (left) can predict identity quite well after a small amount of epochs. On the contrary, the blur latent code predictor for the manifold-based approach (middle) takes more time to be able to predict the identity of an image. The one-dimensional manifold-based approach (right) is not able to classify an image based on the blur latent code at all, which is good. All three approaches show good results in terms of classification based on the identity latent code.

Latent space visualization

In order to qualitatively evaluate whether the latent space shows disentanglement, we create t-distributed stochastic neighbour embedding (tSNE) plots. tSNE is a technique similar to dimensionality reduction, which allows graphing data with three or more dimensions into a 2-dimensional plot. We create tSNE plots for the blur and identity latent codes. In order to generate these tSNE plots, we sample 30 images for 8 different identities, resulting in 240 images. These images are encoded by the encoder, which results in 240 blur latent codes and 240 identity latent codes. The tSNE algorithm is first applied to the identity latent codes. The resulting output is plotted and coloured by their identity. Afterwards, the tSNE algorithm is applied to the blur latent codes. The output is plotted and coloured by their Δ z-level.

For the tSNE algorithms, we use principal component analysis as its initialization method. Moreover, two plots per latent space are generated with different perplexities: 10 and 30. It is important to denote that patterns in tSNE can be a bit deceiving, e.g. a linear pattern in the tSNE plot does not necessarily indicate linearity in the latent space.

Blur Latent Space In order to investigate the structure of the two latent space, we created two tSNE plots with perplexities 10 and 30. These images are coloured by their Δ z-level, since this is vital for how we want the latent space to be structured.


tSNE Blur, perplexity = 10

tSNE Blur, perplexity = 30

FIGURE 5.5: ML-VAE tSNE blur latent space, coloured by Δ z-level

The structure of the blur latent space for the ML-VAE approach displayed by Figure 5.5 is quite unorganized. There is some group forming to be observed in both plots, in which images with a similar Δ z-level are grouped together in terms of their blur latent code, but the effect is minimal. Furthermore, we observe some linear patterns in which the gradient is nicely linearly structured, but this effect is minimal. Important to note that a linear pattern in a tSNE plot does not necessarily indicate linearity in the latent space.

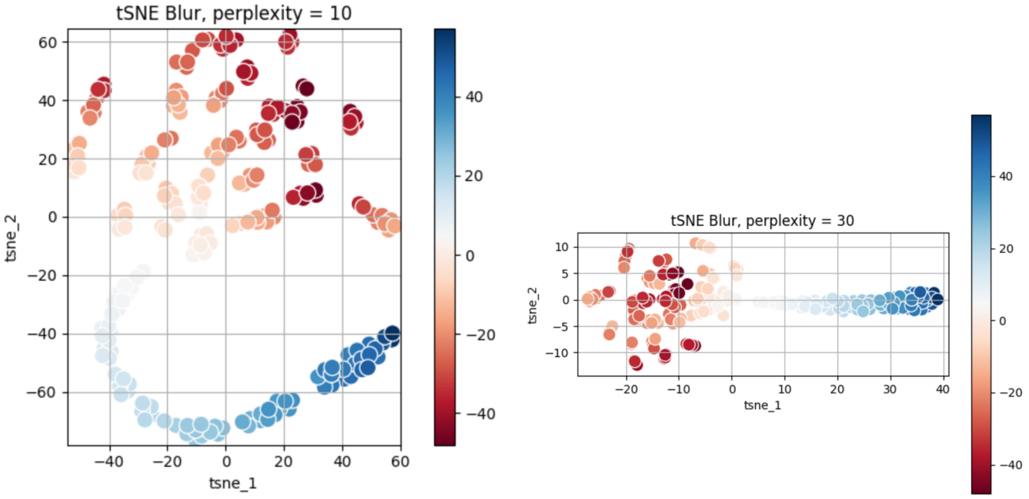


FIGURE 5.6: Manifold-based approach tSNE blur latent space, coloured by Δ z-level

The structure of the blur latent space for the manifold-based approach displayed by Figure 5.6 shows a better structure than the ML-VAE approach. We can see that blur latent codes for images with a high Δ z-level get grouped together nicely. Furthermore, there is a clear gradient when traversing from the high Δ z-level to the zero point. The blur latent codes for images with a negative Δ z-level are a bit less organized, but still show some nice grouping properties

Graphing the blur latent space is different for the one-dimensional blur latent code model. Since the blur latent code is one-dimensional, there is no need to use the tSNE algorithm. We can simply plot the latent code values on both the x- and y-axis.

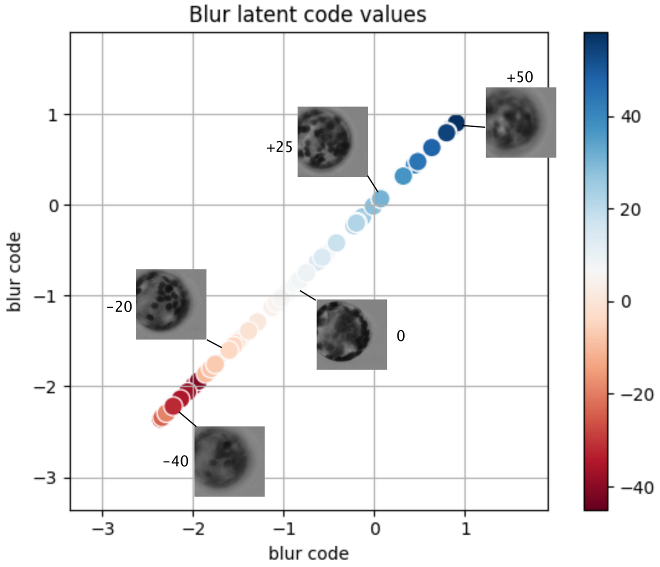
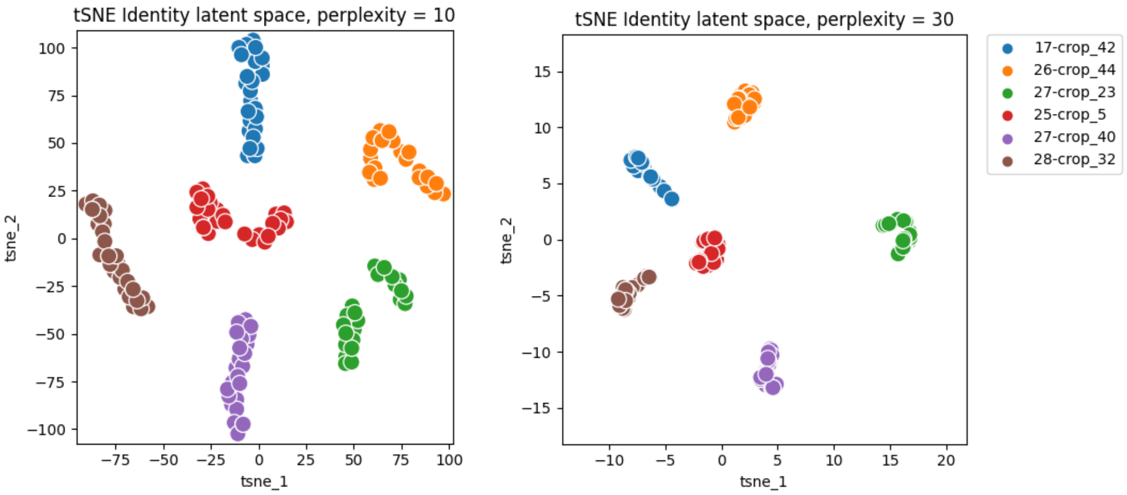


FIGURE 5.7: one-dimensional blur latent code model, blur latent space, coloured by Δ z-level

We can observe a gradient in terms of the Δ z-level in the blur latent space plot (Figure 5.7).

This is a desirable pattern for the latent space, since latent codes for similar blur levels are close together in the latent space. Furthermore, the structure of the latent space is linear. There is no need for any complex enforcement on the latent space to be linear.

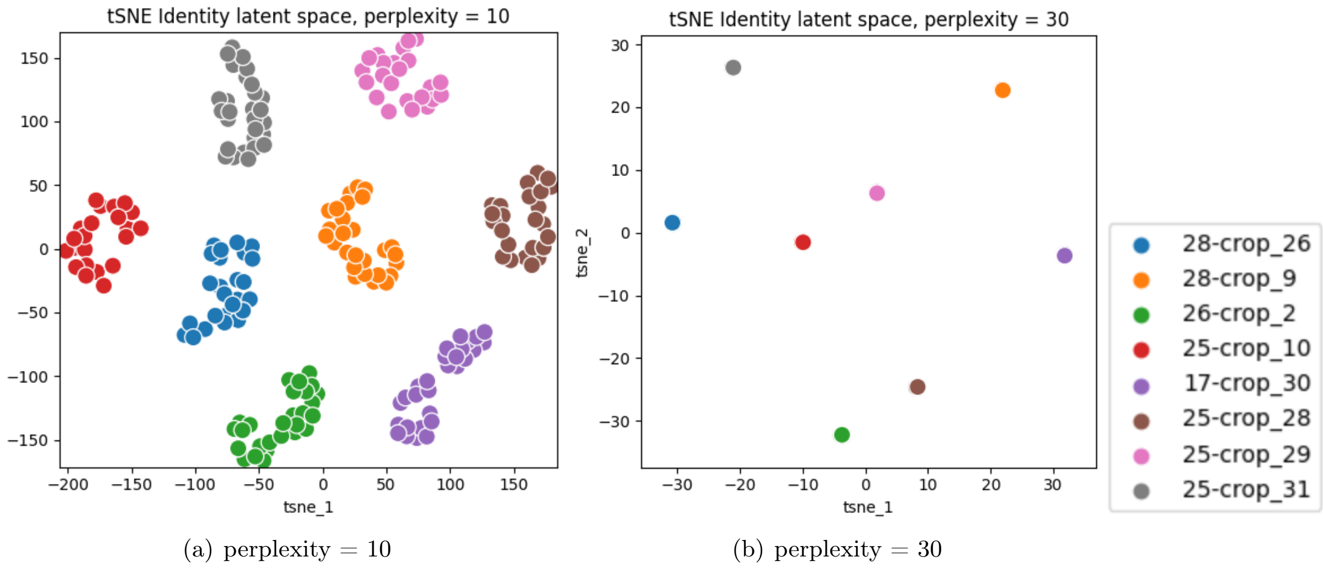


(a) perplexity = 10

(b) perplexity = 30

FIGURE 5.8: ML-VAE tSNE identity latent space, coloured by identity

Identity latent space The tSNE plots of the identity latent space for the ML-VAE model (Figure 5.8) display groupings based on the image's identity. This shows one can distinguish between identities based on the identity latent code and latent codes of images with equal identities tend to be similar. However, they do not yet encode to one point. This indicates there is some variance in the identity latent codes, which is caused by the presence of blur information in the identity latent code. This shows the disentanglement is only partially working.

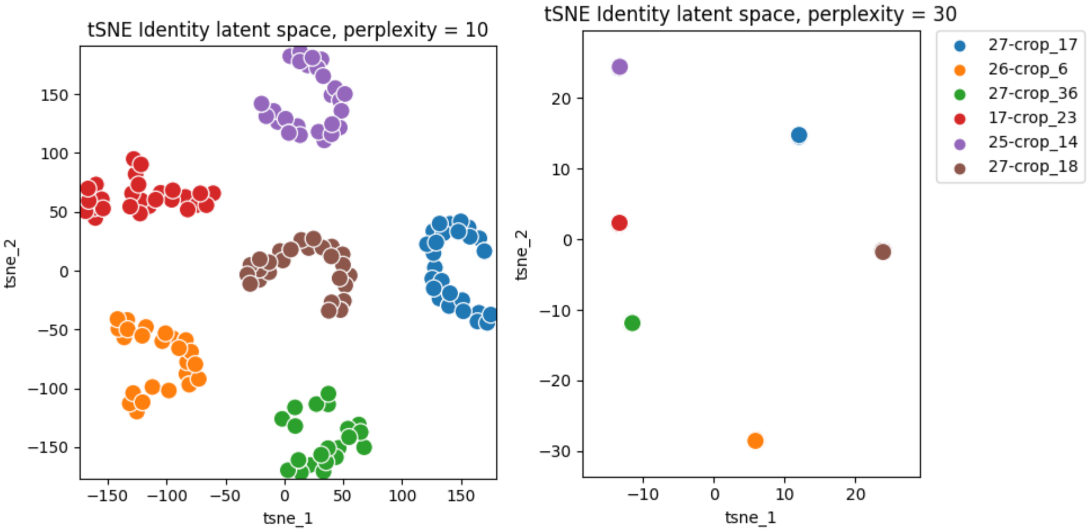


(a) perplexity = 10

(b) perplexity = 30

FIGURE 5.9: Manifold-based approach tSNE identity latent space, coloured by identity

The tSNE plot with perplexity = 10 of the identity latent space for the manifold-based approach (Figure 5.9) shows a similar structure. An interesting difference can be observed in the perplexity = 30 plot. The 30 identity latent codes are grouped together better than in the ML-VAE approach plot, indicating that the identity latent codes are more similar in the manifold-based approach than in the ML-VAE approach.



(a) perplexity = 10

(b) perplexity = 30

FIGURE 5.10: one-dimensional blur latent code manifold-based approach tSNE identity latent space, coloured by identity

Figure 5.10 shows the tSNE plots of the identity latent space for the one-dimensional blur latent code model. These plots are similar to the normal manifold-based model in Figure 5.9. Furthermore, they show that identity latent codes for images with an equal identity are grouped together in the identity latent space. This means the model is able to extract the identity from the image and encode it in the identity latent code.

5.2.4 Deblurring evaluation

The main goal of this research is to deblur blurry microscopy images. In order to evaluate the deblurring capabilities, we use the deblurring strategy as discussed in Section 4.4.3.

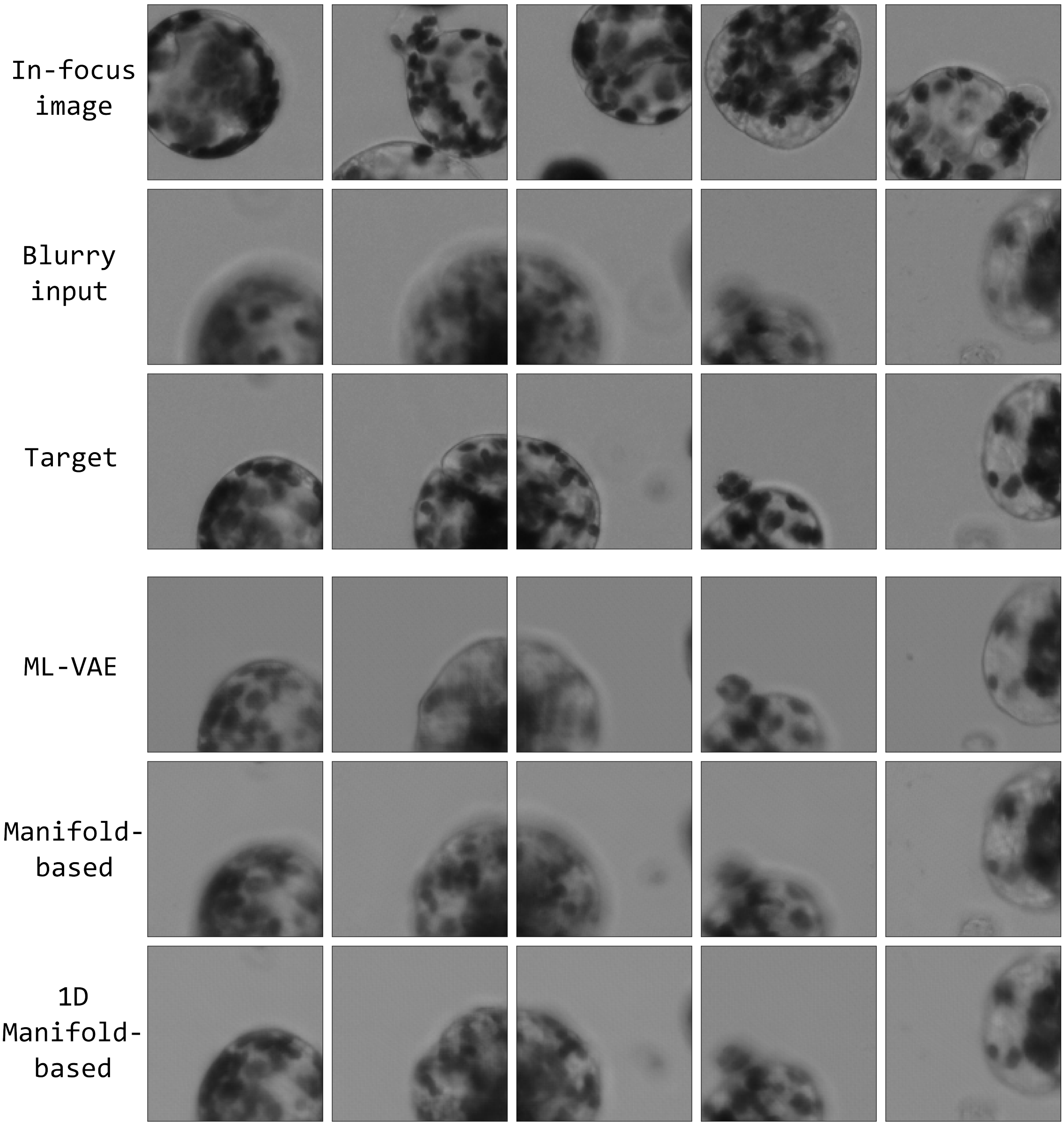


FIGURE 5.11: Deblurring results of the ML-VAE model, manifold-based model and 1D manifold-based model. Top row: in-focus image from which the blur code is used. Second row: blurry image from which the identity latent code is used. Third row: Target in-focus image. Fourth row: ML-VAE deblurring result. Fifth row: Manifold-based deblurring result. Bottom row: 1D manifold-based deblurring result.

Figure 5.11 shows the deblurring results for the ML-VAE model, manifold-based model and 1D manifold-based model. When following the deblurring procedure using the ML-VAE model, we see some interesting results displayed in the fourth row. The decoding result of the first column shows a good improvement in terms of the outline of the cell. The edge of the cell becomes very sharp. The components inside the cell are more detailed than the blurred image, but are not yet as detailed as the target in-focus image. The phenomenon of not being able to reconstruct the level of detail of the in-focus image can be observed best in the second and third column. These reconstructions again show the sharp edges. However, the inside of the cell remains vague and undetailed.

The fifth row shows the results of the deblurring procedure when using the manifold-based model. The main differences with the ML-VAE model are edge sharpness and cellular detail. The edge sharpness of the deblurring attempts is a bit less good than the ML-VAE model. A part of the vague glow around the edge remains present in the reconstructions of the manifold-based approach. However, the level of detail is better when comparing it to the deblurring attempts of the ML-VAE model.

The deblurring results for the 1D manifold-based model in the bottom row show an improvement over both the results for ML-VAE and the normal manifold-based model. The edge of the cells in the first three columns are as sharp as the ML-VAE reconstructions. Furthermore, the level of detail in the cell is better than the deblurring attempts from the normal manifold-based model. The deblurring attempt in the first column shows a level of detail which is almost similar to the target in-focus image.

Model	Test set		Training set	
	SSIM	**PSNR**	**SSIM**	**PSNR**
ML-VAE	0.8629	28.816	0.8927	29.801
Manifold-based	0.8925	31.062	0.9095	32.150
1D manifold-based	0.9179	33.020	0.9315	34.492

TABLE 5.3: Quantitative evaluation of the deblurring capabilities of the different approaches following the deblurring procedure shown in Figure 4.8

Table 5.3 shows a quantitative analysis of the deblurring performance of the three approaches. Similar to the quantitative reconstruction analysis (Section 5.2.2), the ML-VAE model gets outperformed by the manifold-based models. Moreover, the 1D manifold-based model outperforms the normal manifold-based model. A possible cause for this observation might be the improved level of disentanglement and structure in the latent space for the 1D manifold-based model we observed in Section 5.2.3.

5.2.5 Linear interpolation

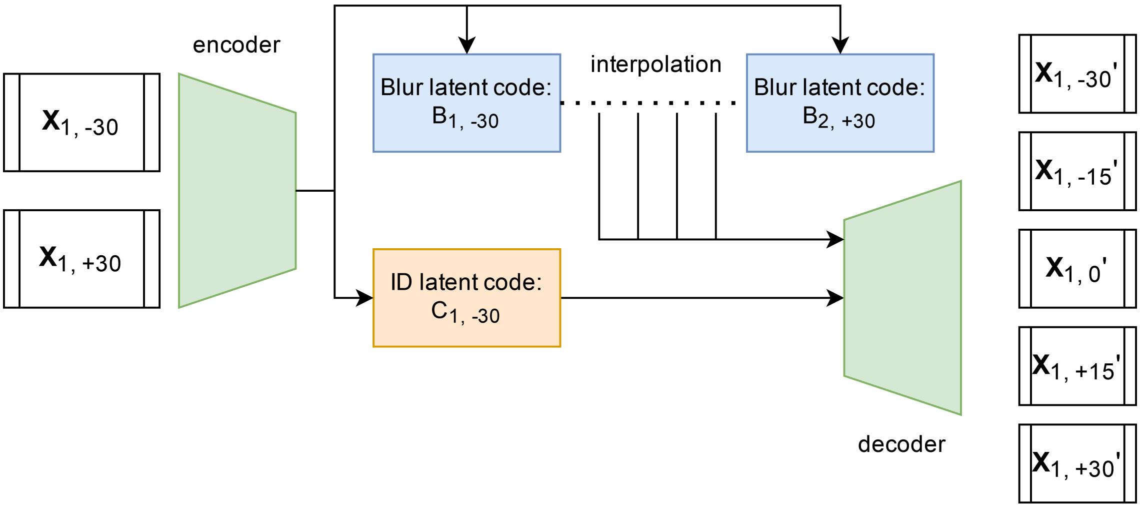


Figure 5.12: Linear interpolation procedure

Once training is finished and the disentanglement between blur and identity is fulfilled, we can visually examine its effectiveness by interpolating between two images with equal identities but different Δ z-levels. As shown in Figure 5.12 we sample a pair of images. One with Δ z-level -30 and one with Δ z-level +30. X_1 can be encoded into identity code C and blur code B_1. X_2 can be encoded into identity code C and blur code B_2. We assume the identity codes are equal, since the identity is shared between the images.

$$B_{inter} = (1 - ratio) \cdot B_1 + ratio \cdot B_2 \tag{5.1}$$

We can linearly interpolate between the two blur latent codes using Eq. 5.1, in an attempt to find the in-focus point somewhere in the middle. The ratios are 6 steps between 0 and 1: $[0, 0.16, 0.33, \cdots, 1]$. We expect to find the in-focus images at ratio $= 0.5$.

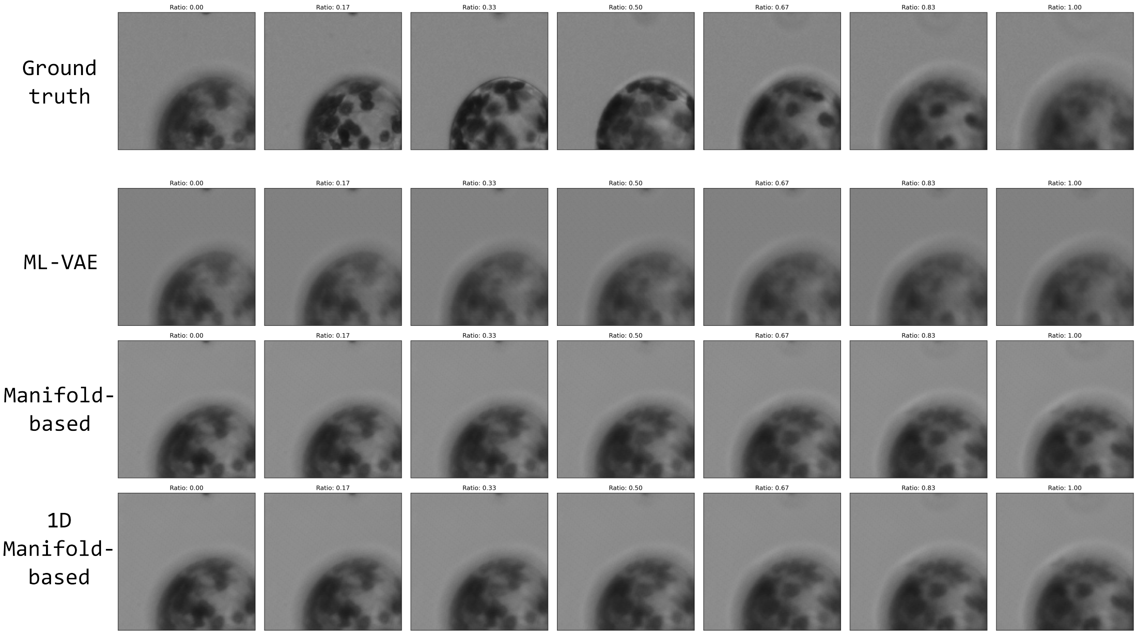


FIGURE 5.13: Linear interpolation on the blur latent space for the ML-VAE model, the manifold-based approach and the 1D manifold-based approach. Top row: ground truth, images at decreasing focal lengths. Second row: interpolation results for ML-VAE model. Third row: interpolation results for manifold-based model. Bottom row: Interpolation results for 1D manifold-based model

As shown in Figure 5.13, a linear interpolation on the blur latent space of the ML-VAE model does not lead to an in-focus image somewhere along the interpolation. E.g. we are unable to locate the sweet spot in the blur latent space using a linear interpolation. This corresponds quite well with the results of Section 5.2.3, which showed that the blur latent space for the ML-VAE approach is relatively unorganized. A final thing we can observe is that such an interpolation in the blur latent space does in fact only influence the level of blur in the image and not the identity. For example, in the first row, we can observe the image slowly becoming blurrier as we move to a blur latent with a high Δ z-level. This shows that the blur latent space mostly captures the blur of the image and not the identity.

The third row of Figure 5.13 displays the linear interpolations of the manifold-based approach. This shows a more promising interpolation when compared to the ML-VAE model interpolations. We can observe new detail arising in the image and other details disappearing during the interpolation, which is desired behaviour when interpolating on the blur latent space. In the middle of the cell, a new part of the cell appears.

The reconstructions for the 1-dimensional blur latent code model shown in the bottom row of Image 5.13 are good. The quality is similar to the normal manifold-based model's reconstruction quality and the level of detail in the cell is comparable. This shows the model can still reconstruct complex images, even though the blur latent code is trimmed down to a single dimension.

5.3 Ablation study: novel approach

The second experiment is an ablation study for our novel manifold-based approach. As described in Section 4.2, multiple losses from the Fumero et al. paper are used in the manifold-based approach. Furthermore, we add the cross reconstruction loss because of the knowledge advantage present with the given dataset, as mentioned in Section 4.2.5. In order to investigate whether all the different losses are improving disentanglement and deblurring, we perform an ablation study. This is done by retraining the manifold model. With each run, one of the losses is turned off and does not contribute to the overall training loss. Afterwards, we compare the model performance qualitatively by visually inspecting the reconstructions and deblurring attempts. The training run with all losses active functions as a baseline. Furthermore, we quantitatively measure the performance, in similar fashion to the comparison experiment described in Section 5.2.

5.3.1 Disabling Losses

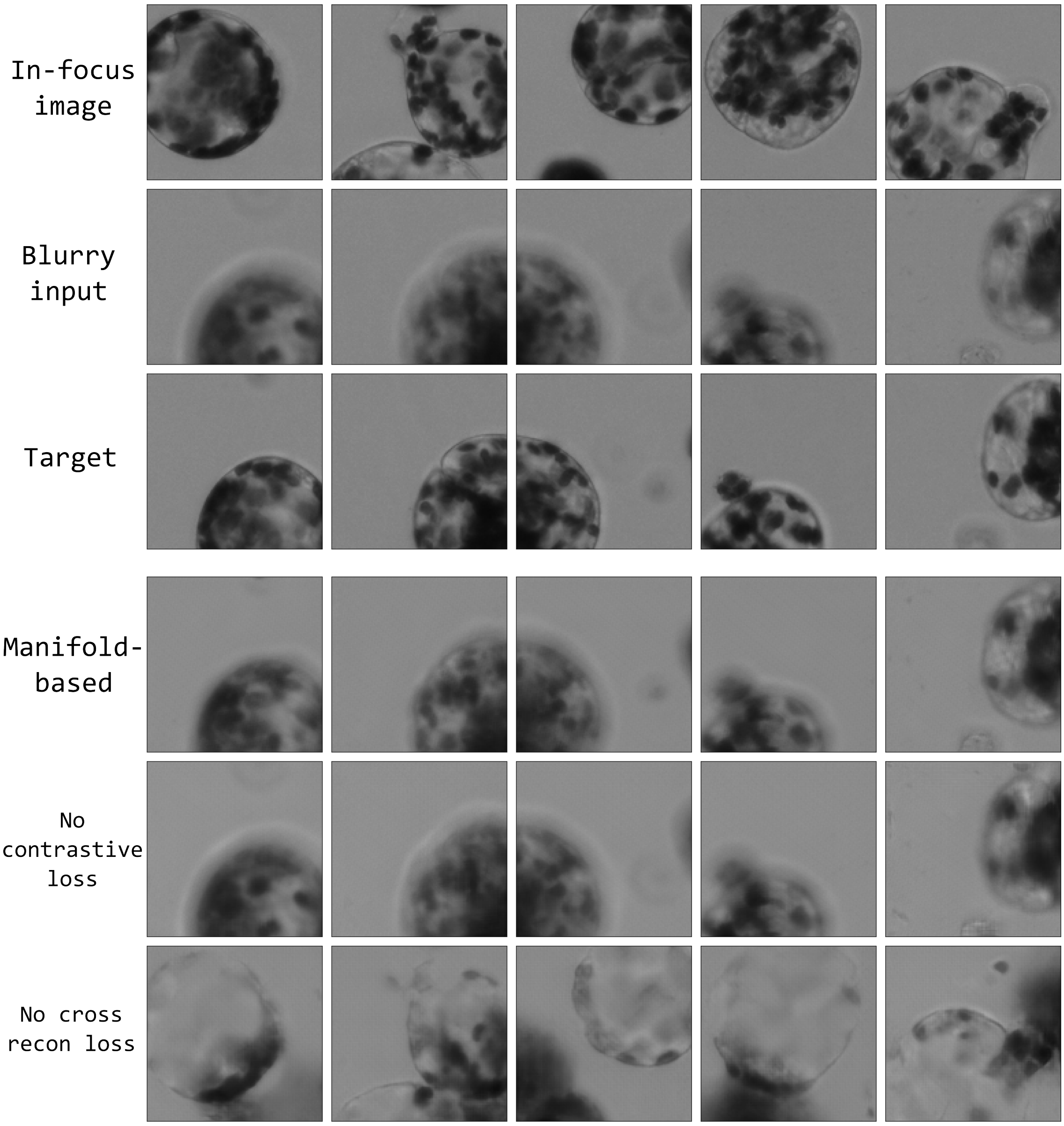


FIGURE 5.14: Deblurring results of the manifold model with different losses switched off. Top row: in-focus image from which the blur code is used. Second row: blurry image from which the identity latent code is used. Third row: Target in-focus image. Fourth row: manifold-based deblurring result. Fifth row: manifold-based without $\mathcal{L}_{cont}$ deblurring result. Bottom row: manifold-based without $\mathcal{L}_{cross}$ deblurring result.

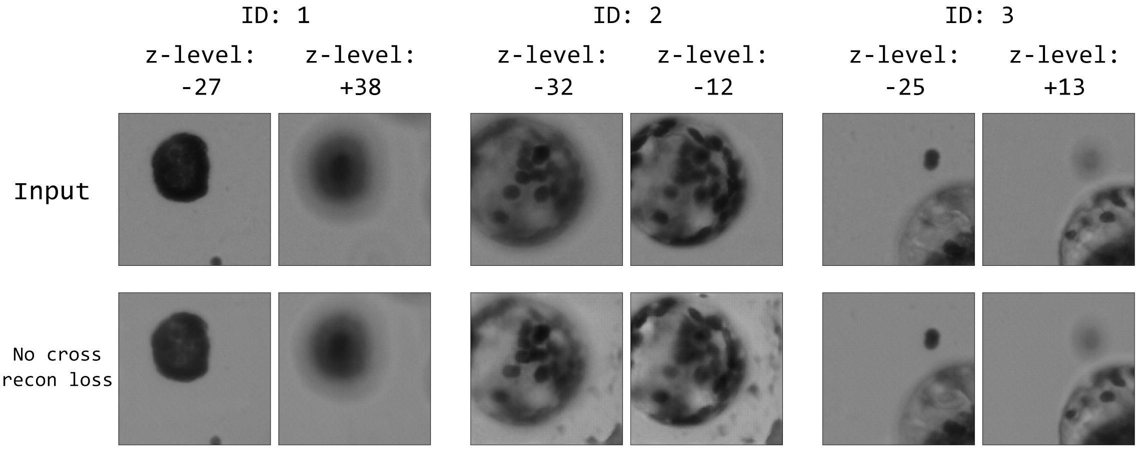


FIGURE 5.15: Reconstructions manifold-based model without cross reconstruction loss

Model	Test set				Training set			
	Reconstruction		Deblurring		Reconstruction		Deblurring	
	SSIM	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**
Manifold-based	0.9455	37.927	0.8925	31.062	0.9462	38.455	0.9095	32.150
No Cross Recon Loss	0.9439	38.098	0.7724	24.736	0.9452	38.323	0.8123	25.406
No Contrastive Loss	0.9487	38.914	0.8722	29.642	0.9486	38.239	0.8935	30.563

TABLE 5.4: Performance of models in ablation study in terms of reconstruction and deblurring

Disabling Contrastive Loss

After disabling the contrastive loss ($\mathcal{L}_{cont}$), the resulting model shows a slight increase in reconstruction capability, as displayed in Table 5.4. However, the model shows a drop in deblurring performance for both the training and test set. When investigating the deblurring attempts in the sixth row of Figure 5.14, we observe that the identity of the image is the same as the blurry image, which is good. However, the blur in the image changes a lot less than the normal manifold-based approach. This shows the contrastive loss is vital for the deblurring performance.

Disabling cross reconstruction loss

When disabling the cross reconstruction loss ($\mathcal{L}_{cross}$), we can observe some interesting things. As shown in Figure 5.15, the reconstructions remain on the same level as the manifold-based model with cross reconstruction loss, although some noise can be observed in the third and fourth column. The model without cross reconstruction loss even shows a slight increase in terms of quantitatively evaluating the reconstruction capability, as presented in Table 5.4. However, we observe a massive drop in the deblurring performance of the model. This shows the cross reconstruction loss is vital for the deblurring performance of the model. The visualizations of the deblurring attempts presented in Figure 5.14 strengthen our observation. The deblurring attempts show some hybrid form between the in-focus image and the blurry image, which shows the blur latent code still encodes part of the identity.

Chapter 6

Conclusions and Future Work

In-focus microscopy images are fundamental for the performance of computational models predicting plant growth. However, microscope setups are often imperfect, leading to a certain level of blur in the resulting images. In order to mitigate such blur, deep learning models can be used. By investigating a novel approach using disentangled representation learning, this work has shown that disentangling between blur and identity in two separate latent codes is not only possible, but allows improving image quality. This disentanglement between blur and identity allows modification of the blur in an image without disturbing the identity of the image.

Furthermore, by qualitatively and quantitatively assessing our approach, this work has shown that our novel manifold-based approach outperforms an ML-VAE baseline in terms of disentanglement, reconstruction, deblurring and structure of the latent space. Moreover, our approach proves to be able to capture the level of blur in a 1-dimensional latent code, which ensures linearity in the latent space. This linearity is backed up by a visualization of the latent space. Disentangled representation learning proved to be beneficial for deblurring, because it allows directly changing blur in an image without influencing the identity. This feature of disentanglement could prove vital to improve other state-of-the-art deblurring methods using latent representations.

Future research in the direction of blur mitigation using disentangled representation learning should focus on strengthening the work. This means comparing the method proposed in this work to other state-of-the-art deblurring methods. Furthermore, our method should be tested on other datasets with different sorts of blur, to see to what extent the disentanglement generalizes. Moreover, the independent adaptability provided by disentangled representation learning might allow improvement in state-of-the-art deblurring approaches using latent codes, which is worth investigating.

Bibliography

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, August 2013. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2013.50.

[2] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations, May 2017. arXiv:1705.08841 [cs, stat]. URL: http://arxiv.org/abs/1705.08841, doi:10.48550/arXiv.1705.08841.

[3] Dennis Francis. The plant cell cycle 15 years on. *New Phytologist*, 174(2):261–278, 2007. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.2007.02038.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.2007.02038.x, doi:10.1111/j.1469-8137.2007.02038.x.

[4] Marco Fumero, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Learning disentangled representations via product manifold projection, October 2021. arXiv:2103.01638 [cs]. URL: http://arxiv.org/abs/2103.01638, doi:10.48550/arXiv.2103.01638.

[5] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations, December 2018. arXiv:1812.02230 [cs, stat]. URL: http://arxiv.org/abs/1812.02230, doi:10.48550/arXiv.1812.02230.

[6] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. July 2022. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[7] Kwang Eun Jang, Hee Won Yang, and Jong Chul Ye. Single channel 2-D and 3-D blind image deconvolution for circularly symmetric fir blurs. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1313–1316, November 2009. ISSN: 2381-8549. doi:10.1109/ICIP.2009.5413582.

[8] Hyunjik Kim and Andriy Mnih. Disentangling by Factorising. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2649–2658. PMLR, July 2018. ISSN: 2640-3498. URL: https://proceedings.mlr.press/v80/kim18b.html.

[9] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022. arXiv:1312.6114 [cs, stat]. URL: http://arxiv.org/abs/1312.6114, doi:10.48550/arXiv.1312.6114.

[10] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, December 2018. arXiv:1711.00848 [cs, stat]. URL: http://arxiv.org/abs/1711.00848, doi:10.48550/arXiv.1711.00848.

[11] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks, April 2018. arXiv:1711.07064 [cs]. URL: http://arxiv.org/abs/1711.07064, doi:10.48550/arXiv.1711.07064.

[12] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better, August 2019. arXiv:1908.03826 [cs]. URL: http://arxiv.org/abs/1908.03826, doi:10.48550/arXiv.1908.03826.

[13] Anat Levin, Yair Weiss, Fredo Durand, and William T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, June 2009. ISSN: 1063-6919. doi:10.1109/CVPR.2009.5206815.

[14] Cewen Liu, Mengyao Sun, Nanxun Dai, Wei Wu, Yanwen Wei, Mingjie Guo, and Haohuan Fu. Deep learning-based point-spread function deconvolution for migration image deblurring. *Geophysics*, 87(4):S249–S265, June 2022. doi:10.1190/geo2020-0904.1.

[15] S. Ramya and T. Mercy Christial. Restoration of blurred images using Blind Deconvolution Algorithm. In *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pages 496–499, March 2011. doi:10.1109/ICETECT.2011.5760166.

[16] Carolyn G. Rasmussen and Marschal Bellinger. An overview of plant division-plane orientation. *New Phytologist*, 219(2):505–512, 2018. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/nph.15183. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/nph.15183, doi:10.1111/nph.15183.

[17] Dongwei Ren, Wangmeng Zuo, David Zhang, Jun Xu, and Lei Zhang. Partial Deconvolution With Inaccurate Blur Kernel. *IEEE Transactions on Image Processing*, 27(1):511–524, January 2018. Conference Name: IEEE Transactions on Image Processing. doi:10.1109/TIP.2017.2764261.

[18] Christian J. Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to Deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, July 2016. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. doi:10.1109/TPAMI.2015.2481418.

[19] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015. arXiv:1409.1556 [cs]. URL: http://arxiv.org/abs/1409.1556, doi:10.48550/arXiv.1409.1556.

[20] Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled Representation Learning, November 2022. arXiv:2211.11695 [cs]. URL: http://arxiv.org/abs/2211.11695, doi:10.48550/arXiv.2211.11695.

[21] Asfand Yaar, Hasan F. Ates, and Bahadir K. Gunturk. Deep Learning-Based Blind Image Super-Resolution using Iterative Networks. In *2021 International Conference*

on Visual Communications and Image Processing (VCIP), pages 01–05, December 2021. ISSN: 2642-9357. doi:10.1109/VCIP53242.2021.9675367.

[22] Ruomei Yan and Ling Shao. Blind Image Blur Estimation via Deep Learning. *IEEE Transactions on Image Processing*, 25(4):1910–1921, April 2016. Conference Name: IEEE Transactions on Image Processing. doi:10.1109/TIP.2016.2535273.

[23] Chi Zhang, Hao Jiang, Weihuang Liu, Junyi Li, Shiming Tang, Mario Juhas, and Yang Zhang. Correction of out-of-focus microscopic images by deep learning. *Computational and Structural Biotechnology Journal*, 20:1957–1966, January 2022. URL: https://www.sciencedirect.com/science/article/pii/S2001037022001192, doi:10.1016/j.csbj.2022.04.003.

[24] Huangxuan Zhao, Ziwen Ke, Ningbo Chen, Songjian Wang, Ke Li, Lidai Wang, Xiaojing Gong, Wei Zheng, Liang Song, Zhicheng Liu, Dong Liang, and Chengbo Liu. A new deep learning method for image deblurring in optical microscopic systems. *Journal of Biophotonics*, 13(3):e201960147, 2020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jbio.201960147. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jbio.201960147, doi:10.1002/jbio.201960147.