ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Machine Learning for Computer Vision

GENERATIVE ADVERSARIAL NETWORKS FOR THREE-DIMENSIONAL MULTI-CHANNEL VIRTUAL FLUORESCENT STAINING

CANDIDATE
Lorenzo Venieri

SUPERVISOR
Prof. Samuele Salti

CO-SUPERVISOR
Andrea Masella

Academic year 2022-2023

Contents

Introduction

Histological and cytological staining is a key method of examining tissue and subcellular structures in clinical pathology and life-science research. This technique involves the use of chromatic dyes or fluorescent labels to visualize tissue and cellular structures, facilitating their microscopic assessment.

Traditional chemical staining is expensive, time-consuming, difficult to use on in vitro cultures, and allows only a limited number of structures to be stained and observed on the same specimen due to the limited spectrum of stains available and their spectral overlap. Moreover, the toxic chemical compounds used generate significant amounts of waste and can alter the analyzed section, preventing additional staining and further analysis.

Deep learning techniques have opened up new possibilities for staining methods, generating virtually stained images from label-free images. This breakthrough offers fast and cost-effective alternatives for visualizing tissue and cellular structures. It has the potential to make tissue and cellular examination more accessible, particularly in resource-limited settings, and to make it possible to perform in vivo imaging (Figure 1).

For this image-to-image translation task, the model is trained to predict the target cellular structures, represented by the fluorescent channels of the target images, starting from label-free (e.g. transmitted-light)

images of the sample. Models used for this task are usually based on a U-Net architecture, either trained to minimize a pixel-wise loss function that measures the dissimilarity between the prediction and the target or used as the generator in a conditional generative adversarial network (cGAN).

In this work, we present a novel Wasserstein GAN with Gradient Penalty (WGAN-GP) [2, 10] model that can take full advantage of three-dimensional and multi-channel information of our data.

0.1 OrganVision

This work is part of the OrganVision project: an EU-funded multinational project that brings together various members from academia and industry to collaborate on the development of new real-time 3D label-free imaging solutions for organoids, combining the fields of microscopy and artificial intelligence.
Organoids are tiny, three-dimensional tissue cultures derived from stem cells. They serve a crucial role in personalized medicine and the exploration of new drugs. Nonetheless, the development of a tool for real-time visualization of organoids has remained an unmet challenge. To tackle this issue, the OrganVision project, supported by the European Union, is dedicated to creating a label-free image processing technology that offers real-time, high-resolution imaging capabilities for organoid research.
This technology aims to unlock new possibilities in this field like elucidating the real-time functioning of individual cells within engineered heart muscles, shedding light on how various factors can impact their behavior.

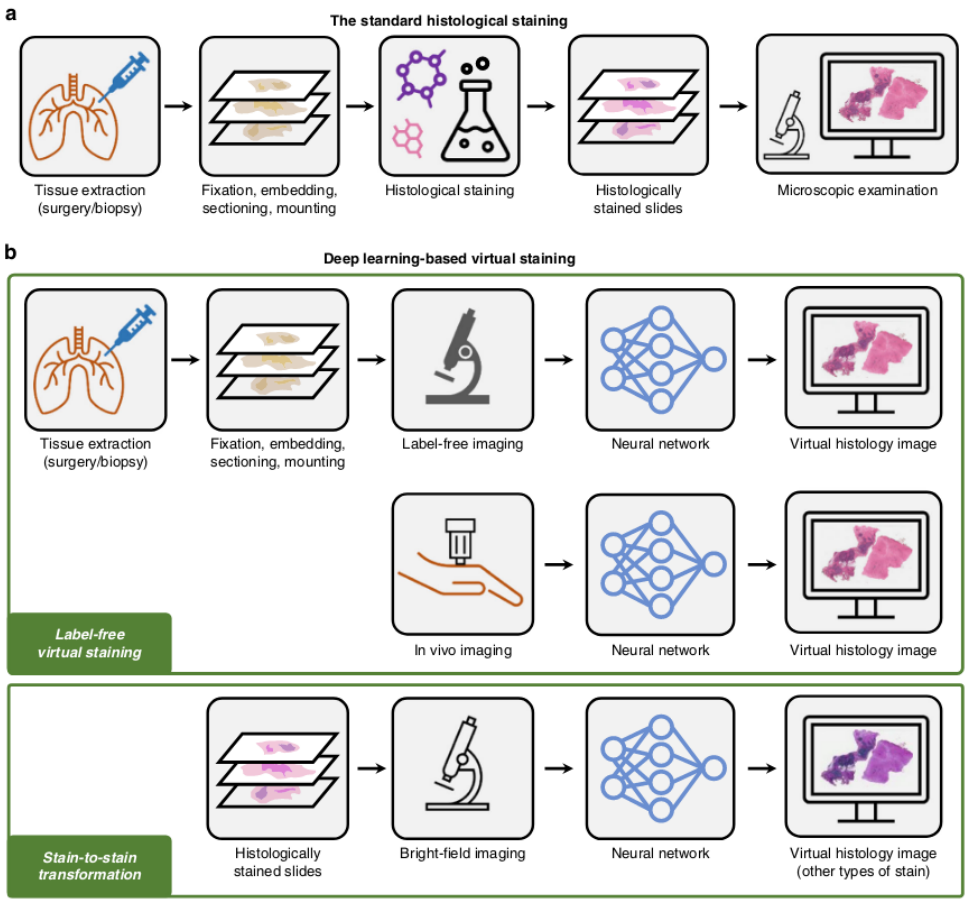

Figure 1: **Schematic of the standard histological staining and deep learning-based virtual staining.**
(**a**) Standard histological staining relies on laborious chemical-based tissue processing and labeling steps. (**b**) Pre-trained deep neural networks enable the virtual histological staining of label-free samples as well as the transformation from one stain type to another, without requiring any additional chemical staining procedures. [3]

The work presented in this thesis was carried out in collaboration with Datrix SpA, a company based in Milan that is part of the consortium of entities working on the OrganVision project.

CHAPTER 1

Background and related works

The term "virtual staining" is broadly used to refer to methods that digitally generate fluorescent stains using trained deep neural networks, including label-free staining, in-silico labeling, and stain-to-stain transformations (Figure 1).
In this work is developed a model for a specific task that in literature is referred to by different names, such as "in silico labeling"[6] or as "label-free prediction of fluorescence images/cell painting"[18, 7].
This task consists of digitally generating, from an unlabeled (unstained) source image of a cell colony or tissue, an image representing the structure that would be seen when looking at a specific fluorescent tag. This method differs from classical label-free staining, which generates a color image representing how the sample would look if chemically stained (Figure 1.1).[1] In silico labeling can be seen as a more powerful technique, as it potentially gives us the ability to separately observe the specific structures we are interested in, and not only the proxies of them given by the staining (Figure 1.2).

Deep learning enabled incredible possibilities for virtual staining, that were explored in the past years [3].

[1] See [21, 17, 20]

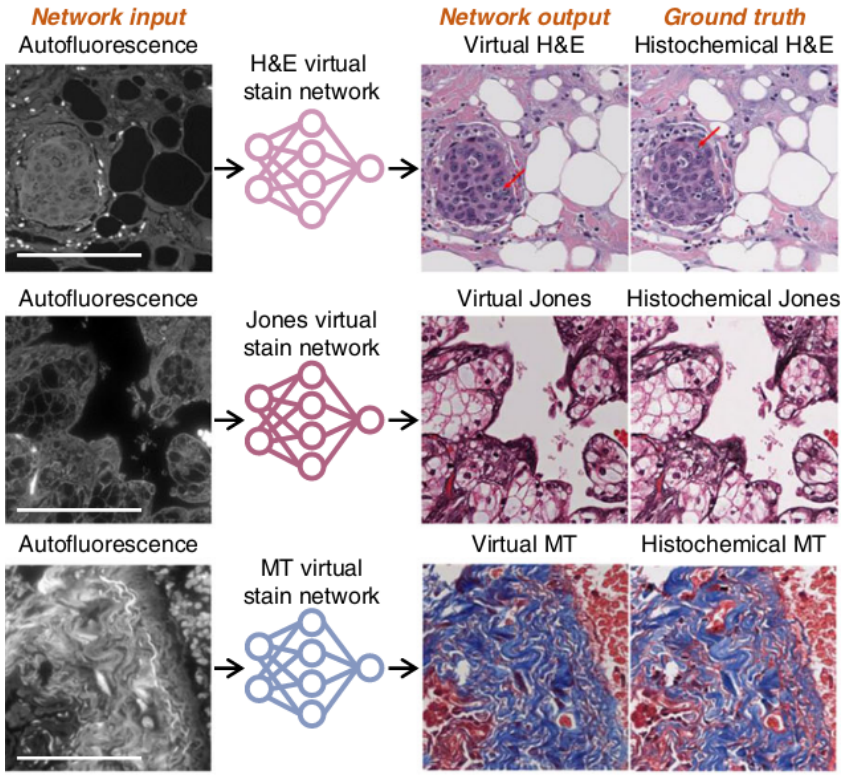

Figure 1.1: **Examples of label-free virtual staining from autofluorescence input images.** The network digitally generates a color image representing how the sample would look if chemically stained with a specific histochemical staining technique (Here are shown hematoxylin and eosin stain (H&E), Jones' Methenamine Silver stain (JMS), and Masson's trichrome stain(MT)). Scale bars represent 100 μm. [3]

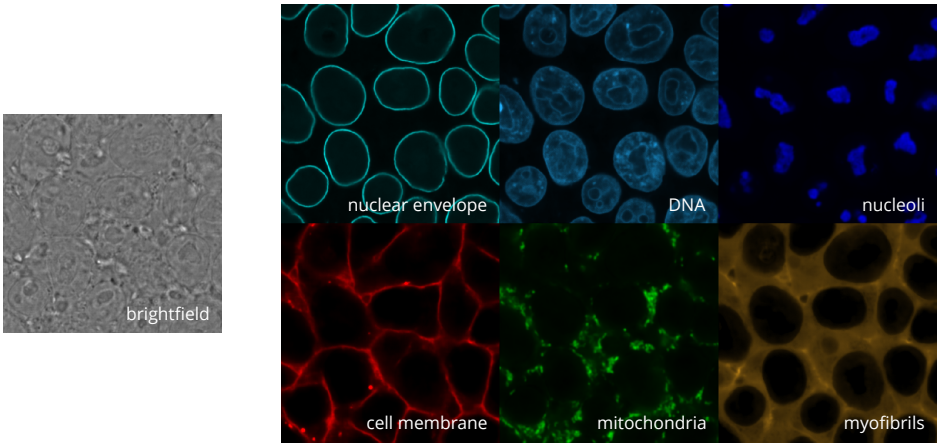

Figure 1.2: **Example of in-silico labeling from brightfield input images.** The network digitally generates the fluorescence channels corresponding to each cellular structure we want to observe.

Christiansen et al.[6] in 2018 presented the first work utilizing a deep learning approach to in silico labeling.
They developed a multi-scale modular model predicting eight fluorescence channels from transmitted-light unlabeled images. Their model takes as input z-stacks of thirteen 250 × 250 pixels slices of the same sample, collected at equidistant intervals along the z-axis. Interesting to note is that the z-dimension is treated as a feature dimension, and they apply 2D convolutions. The output consists of nine tensors: eight corresponding to fluorescence channels and one autoencoding tensor that is used to facilitate debugging of certain training pathologies. For each fluorescence channel, the network outputs a discrete probability distribution (over 256 intensity values) for each pixel. To construct images they take the median of the predicted distribution for each pixel. To make predictions, the network is applied in a sliding-window manner. To infer the complete image, the input images are divided into 250 × 250 patches with a stride of 8. These patches are then fed to the network as inputs, generating outputs of size 8 × 8, which are subsequently stitched together to form the final image.
The model's losses are calculated as the cross-entropy errors between the predicted distributions and the true discretized pixel intensity.
The network predicts the fluorescent label for all the eight different structures simultaneously, but no well in the data has more than three fluorescent labels. To deal with this limitation, they use a masked loss that ensures that at most three fluorescence heads would be updated for any given training example. This mask indicates for each training sample whether a particular fluorescent label is provided or not. The total loss for the fluorescence channel is the weighted average of the gated losses. This loss is then combined with the loss computed for the autoencoding task.

At inference time, the model takes as input the thirteen slices of the transmitted-light source image and is able to simultaneously predict the fluorescent labels of all the eight different structures.

The study also explored using different numbers of z-depths in their network, finding that performance improved with more input z-slices, but the benefit from each additional image diminished as more were added.

In their work, the 3D information is solely used to improve the model's accuracy when predicting the fluorescence channels corresponding to the central slice of the z-stack. However, in theory, their methodology could potentially be extended to predict entire 3D fluorescent images from unlabeled z-stack data.

In this study, the authors chose a straightforward approach by embedding the z-planes into the feature dimensions of the network. This method proves effective when dealing with a limited range of potential z values. Nevertheless, its practicality diminishes when faced with a large number of possible z values. In such situations, the study recommends the adoption of 3D convolutions within the network architecture as a more suitable alternative to 2D convolutions.

The authors recognize the main limitation of their approach in the lack of global coherence in the images. The employed network relies on an approximate loss function that operates at the pixel level, which only partially represents and enforces similarity between prediction and ground truth images. This approach has implications for the network's predictive capabilities. Specifically, it causes the network to independently predict individual pixels, resulting in a lack of global coherence in the final output images. This lack of coherence is most noticeable in the representation of long, thin structures, resulting in discontinuous or averaged predictions.

To address this issue, the study suggests the potential use of established

machine learning techniques such as adversarial models, to enhance the network's ability to produce coherent and accurate images, particularly when dealing with intricate and fine structures in the data.

In the same year, Ounkomol et al. [18], present a study where is investigated a model for fluorescent label prediction that takes as input three-dimensional transmitted-light (TL) live cell images. Their model uses a U-Net [22] architecture trained to minimize the mean squared error (MSE) between generated and ground truth fluorescent channels. Contrary to Christiansen et al., their model does not perform simultaneous prediction of multiple fluorescent channels. Instead, they train a different model for each structure. To evaluate the performances they use the Pearson correlation coefficient.
In their work, they also establish the possibility of using this approach for digital fluorescent labeling of time lapses by applying the model trained solely on static images to a single TL 3D time series.
They compared the results of the 3D model versus a model that takes as input only 2D slices, measuring better performances with the 3D model. This establishes that 3D patterns are valuable for predicting subcellular organization.

In 2019, Rivenson et al. [21] publish the first GAN architecture for virtual staining. Their network predicts, starting from label-free unstained auto-fluorescence images, the corresponding bright-field images as if they were chemically stained (Hematoxylin & Eosin staining).
To train their GAN model, the loss of the generator, which follows the design of a U-Net, combines the pixel-wise mean squared error between output and target images, total variation of the output image (to encourage less blurring), and the adversarial loss computed by the

discriminator. The discriminator does not take as an input the source unstained auto-fluorescence image, so it only penalizes fake-looking images, but not images that look realistic but that are not realistic fluorescent labeling of the source unstained image. Isola et al.[12] found out that for image-to-image transformation tasks, passing as input to the discriminator also the source image, usually produces better results.

In 2022, Cross-Zamirski et al. [7] develop two models for the prediction of five fluorescent channels, taking as input three 2D brightfield slices from different focal z-planes.
The first model is a U-Net trained to minimize L1 loss (MAE), that takes as input 2D 256 × 256 pixels patches of the original images. This model is trained for 15 000 steps with a batch size of 10 (around 30 h of training). After this, the same trained U-Net model is used as the generator of a conditional Wasserstein GAN, and is now trained to minimize a combination between L1 loss and the adversarial loss computed by the critic model:

$$\mathcal{L}_G(G, D) = \lambda_1 \mathcal{L}_{L^1} + \lambda_e \mathcal{L}_{ADV}$$

where λ_1 is a weighting parameter for the L1 objective and $\lambda_e = 1/epoch$ is an adaptive weighting parameter to prevent the unbounded adversarial loss $\mathcal{L}_{ADV}$ overwhelming the L1 component.
The outputs of the model have the same shape as the input, and the full images are reconstructed by stitching together the 256×256 patches with a stride of 128, computing the median value of the pixels in the overlapping portions (two overlapping patches along the edges and four in the central portion of the image). This approach often produces artifacts along the lines where the patches are stitched together.

The predicted and target images were evaluated both at the image level and at the morphological feature level. The metrics used for the image-level evaluation are: mean absolute error (MAE), mean squared error (MSE), structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), and the Pearson correlation coefficient (PCC). These metrics show slightly better performances for the WGAN model.

1.1 Original Contributions

The work presented in this thesis tries to combine all the insights and promising features of these past works.

Our model takes as input 3D images and outputs 3D images like the model of Ounkomol et al. [18]. To leverage the correlation and additional information present in the other channels, our model is trained to predict six fluorescent channels simultaneously. Each sample in our training data, like in the dataset of Christiansen et al. [6], only has three fluorescent labels. So to train the model we apply the same idea of a masked loss used in their work.

To deal with coherence and lack of finer details, we use a conditional GAN approach. In particular, we used a WGAN-GP model, like Cross-Zamirski et al. [7], that was found to perform very well thanks to its stability in training.

Moreover, the model designed here has the only requirement for the input images to have at least 8 pixels on one of the axis, and deals with the three dimensions in the same way. Thanks to this feature our model is almost completely agnostic to the size of the input images, taking away the need to stitch together predicted patches from multiple steps of inference.

To the best of our knowledge, this is the first work developing a single GAN model simultaneously predicting different fluorescent channels, that uses the three-dimensional information present in the data, produces 3D images for each fluorescent channel, and takes away the need to stitch together predicted patches from multiple steps of inference.

1.2 U-Net

The U-Net is a popular deep learning architecture initially presented in the work of Ronneberger, Fischer, and Brox [22] for the task of image segmentation in biomedical images.
It comprises two essential components: a contracting path and an expansive path. (Figure 1.3)
In the contracting path, there are encoder layers responsible for capturing contextual information and diminishing the spatial resolution of the input. On the other hand, the expansive path is composed of decoder layers that decode the previously encoded data and incorporate information from the contracting path through skip connections.
The contracting path in U-Net is responsible for identifying the relevant features in the input image. The encoder layers perform convolutional operations, reducing the spatial resolution of the feature maps and increasing their depth. This enables the capture of progressively abstract representations of the input data.
Conversely, the expansive path is dedicated to decoding the previously encoded information and locating the features while maintaining the spatial resolution of the input. The decoder layers within the expansive path increase the size of the feature maps through upsampling and applying convolutional operations. The crucial role of the skip connections from the contracting path is to preserve the spatial information

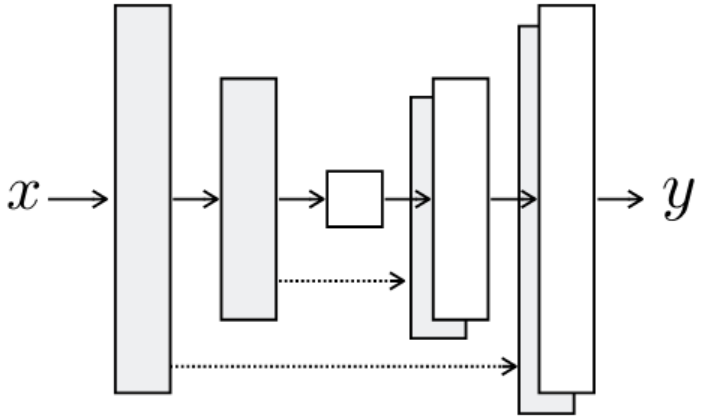

Figure 1.3: **Visual scheme of U-Net architecture.**
Each box is a feature map. The input is encoded by the contracting path, through convolutional downsampling operations. Once it has reached the bottleneck, upsampling operations in the expansive path decode the information in the previous layer, and the output is concatenated with the same level feature map from the contracting path. [12]

that was initially lost during the contracting phase, enhancing the decoder layers' precision in locating the features.
This is exactly what we want for an image-to-image translation task as the structure in the input is roughly aligned with the structure in the output.

1.3 Conditional GAN

U-Nets, like other Neural Networks, are trained to minimize a loss function, which serves as an assessment of the quality of the outputs. Despite the automated nature of the learning process, a significant level of manual work must be dedicated to formulating effective loss functions: we still need to tell the model what metric we wish to minimize. In our image-translation task, we would need a metric that represents how much the generated image is similar to the target image. Designing a loss function that effectively encapsulates our human perception of image similarity poses significant challenges.

The naive approach is to use pixel-wise metrics such as the Euclidean distance between predicted and ground truth pixels. Even if we had perfectly registered images, such that we don't have to deal with translational or rotational offsets, when a model optimizes for this metric we stumble into Goodhart's law: "When a measure becomes a target, it ceases to be a good measure."

It's known that models trained to minimize a pixel-wise measure will produce blurry results, simply because this is the most effective way of minimizing that loss in the presence of noise or when the problem is underconstrained. This happens because by predicting the mean of the distribution, the model minimizes the mean pixel-wise error. [29, 19]

"Coming up with loss functions that force the CNN to do what we really want – e.g., output sharp, realistic images – is an open problem and generally requires expert knowledge."[12]

What we would like is to define a high-level goal like "output images indistinguishable from the ground truth" and then automatically learn a loss function that pushes the model towards this goal. Luckily, this is exactly what a Generative Adversarial Network (GAN) does. GANs train by optimizing a loss function that distinguishes between real and fake output images, while simultaneously training a generative model to minimize this loss. Since blurry images look clearly fake, the model is incentivized to produce sharper contours.

In this work, we don't simply need to produce a realistic image, but a realistic image representing the structure we are interested in, given a specific transmitted-light source image. For this reason, we use a Conditional GAN (cGAN) that learns a conditional generative model of data: a mapping from a source image x to an output image y:

$$G : x \mapsto y$$

The objective of a conditional GAN can be expressed as

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathbb{E}_{x,y}\left[\log D(x, y)\right] + \mathbb{E}_{x}\left[\log(1 - D(x, G(x))\right] \quad (1.1)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. G* = $\arg\min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D)$.

In other works [12, 19] it has been established that, for image-to-image translation tasks, it is beneficial to add to the adversarial loss a more traditional loss, like the L1 distance. In this way, the generator's task is not only to fool the discriminator but also to produce images that are close to the ground truth in an L1 sense.

$$\mathcal{L}_{L^1}(G) = \mathbb{E}_{x,y}\left[\|y - G(x)\|_1\right] \quad (1.2)$$

$$\mathcal{L}(G, D) = \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{L^1}(G) \quad (1.3)$$

So the final objective is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{L^1}(G) \quad (1.4)$$

Training Generative Adversarial Networks is widely recognized as a challenging and unstable process. GANs are notorious for their sensitivity to hyperparameters, data quality, and network architectures, making them prone to issues like mode collapse, training divergence, and other instabilities. The theoretical reasons for this problems were investigated by Arjovsky and Bottou in [1].
Training the discriminator is usually far easier than training the generator. Moreover, the traditional cost functions of GANs (Kullback-Leibler

divergence [2] or Jensen-Shannon divergence[3]) provide gradients close to zero when the distributions of the real and generated images are too distant. These two issues combined have a big responsibility in the instability of the training process of GANs: when the generator performance is poor, the gradient provided by the discriminator vanishes, hampering the learning process and impeding the generator's improvement.

More precisely, the authors state that given two distributions, if their supports are disjoint or lie in low dimensional manifolds, there is always a perfect discriminator between them, and the gradient for the GAN generator's objective function will be zero almost everywhere.

The gradient vanishes when the discriminator becomes optimal (D is close to D^*):

$$\lim_{\|D-D^*\|\to 0} \nabla_\theta \mathbb{E}_{z\sim p(z)} \left[\log\left(1 - D\left(g_\theta(z)\right)\right)\right] = 0 \tag{1.5}$$

To deal with this problem, an alternative cost function was proposed, so that the gradient step for the generator becomes:

$$\nabla_{\theta_g} \log D\left(G\left(x\right)\right) \tag{1.6}$$

But Arjovsky and Bottou [1] proves that, while this gradient doesn't necessarily suffer from vanishing gradients, it causes massively unstable updates under the presence of a noisy approximation to the optimal

[2] $$KL\left(\mathbb{P}_r \| \mathbb{P}_g\right) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x)$$
where both $\mathbb{P}_r$ and $\mathbb{P}_g$ are assumed to be absolutely continuous distributions $\in$ Prob(χ), and therefore admit densities, with respect to a same measure μ defined on X.

[3] $$JS\left(\mathbb{P}_r, \mathbb{P}_g\right) = KL\left(\mathbb{P}_r \| \mathbb{P}_m\right) + KL\left(\mathbb{P}_g \| \mathbb{P}_m\right)$$
where $\mathbb{P}_m$ is the mixture $(\mathbb{P}_r + \mathbb{P}_g)/2$. Note that, contrary to the KL-divergence, this divergence is symmetric and always defined.

discriminator: the updates to the model follow a centered Cauchy distribution that has zero mean and infinite variance.

Another proposed solution was to add continuous noise to the inputs of the discriminator to smoothen the data distribution of the probability mass. When we add noise ϵ to two distributions $\mathbb{P}_r$ and $\mathbb{P}_g$ that have their supports on manifolds that are close, but not overlapping, the noise terms will make the noisy distributions $\mathbb{P}_{r+\epsilon}$ and $\mathbb{P}_{g+\epsilon}$ almost overlap. As a result, the Jensen-Shannon divergence between them becomes small.
However, when we consider the noiseless variants, $\mathbb{P}_r$ and $\mathbb{P}_g$, the JSD between them remains at its maximum, regardless of how close the underlying manifolds are. In other words, the JSD between noiseless distributions is always at its maximum, indicating high dissimilarity, even if the underlying manifolds are close.
While it might be tempting to use the JSD of the noisy variants ($\mathbb{P}_{r+\epsilon}$ and $\mathbb{P}_{g+\epsilon}$) as a measure of similarity between the original distributions ($\mathbb{P}_r$ and $\mathbb{P}_g$), this approach is not without challenges, because it depends on the level of noise introduced (the value of ϵ). Therefore, it's not an intrinsic or absolute measure of the similarity between $\mathbb{P}_r$ and $\mathbb{P}_g$ because it can vary based on the amount of noise added.

Finally, Arjovsky and Bottou introduce the idea of using a different metric as a cost function that has a smoother gradient everywhere, in an attempt to stabilize the training process: the Wasserstein distance. We will see the details of the GAN model trained using this cost function in the next section.

1.3.1 Wasserstein GAN

The distance ρ used to measure how close the model distribution $\mathbb{P}_\theta$ and the real distribution $\mathbb{P}_r$ are has a big impact on the convergence of the training. This is because a sequence of distributions $(\mathbb{P}_t)_{t\in\mathbb{N}}$ converges if and only if there is a $\mathbb{P}_\infty$ such that $\rho(\mathbb{P}_t, \mathbb{P}_\infty) \to 0$ when $t \to \infty$. But this notion depends on how the distance ρ is defined.
Of course, we want our model distribution to be a continuous mapping from parameters θ to distributions $\mathbb{P}_\theta$: if $\theta_t \to \theta$ than $\mathbb{P}_{\theta_t} \to \mathbb{P}_\theta$.
Now, the weaker the distance we defined between distributions, the easier it is for the distributions to converge, and hence the easier it is to have a continuous mapping from parameters space to distributions space.
Arjovsky, Chintala, and Bottou [2] proved that the Wasserstein-1 distance (also known as Earth-Mover distance) (eq. 1.7) is weaker than the Jensen-Shannon distance, and makes the mapping continuous under weaker assumptions.
Wasserstein distance:

$$W\left(\mathbb{P}_r, \mathbb{P}_g\right) = \inf_{\gamma\in\Pi\left(\mathbb{P}_r,\mathbb{P}_g\right)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{1.7}$$

where $\Pi\left(\mathbb{P}_r, \mathbb{P}_g\right)$ is the set of all joint distributions $\gamma(x,y)$ whose marginals are respectively P_r and P_g. Intuitively, $\gamma(x,y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions P_r into the distribution P_g. The Wasserstein distance then is the "cost" of the optimal transport plan.

However, finding the infimum in eq. 1.7 is intractable. But it is

possible to use the Kantorovich-Rubinstein duality to prove that

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)] \tag{1.8}$$

where the supremum is over all the 1-Lipschitz functions $f : X \to \mathbb{R}$:

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|$$

Regarding how to find such f function, we can train a neural network to approximate it. This neural network indeed is very similar to a classic discriminator, with the only difference being that it outputs a scalar score instead of a probability (it doesn't have a final sigmoid activation). This output represents how real the images passed as input to the model are, that is, how close the generator model distribution and the real distribution are. To emphasize the difference with respect to the classical discriminator of a GAN, the authors call this model *critic*. To enforce the 1-Lipschitz constraint, the simplest idea is to confine the weights of this neural network to a compact space. To do this, it is enough to clip the weights of the model to be in a fixed box after each gradient update.

But this method has many downsides. If the clipping parameter is too large, the time needed for any weights to reach their limit could be too large too, making it unfeasible to reach optimality during the training of the critic. On the other hand, a too small clipping parameter can lead to vanishing gradients, especially if the model has many layers. Tuning this hyperparameter is very difficult and time and resource intensive. Nonetheless, this method was enough to already surpass the classic cGAN performance.

1.3.2 WGAN-GP

Gulrajani et al. developed an improvement over gradient clipping to enforce the Lipschitz constraint: gradient penalty [10]. They use the fact that a differentiable function f is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere to prove that points interpolated between the real and generated data should have a gradient norm of 1 for f. This fact makes it possible to define and compute a term to be added to the loss function that penalizes the model when the gradient norm moves away from its target value of 1:

$$\mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1\right)^2\right] \tag{1.9}$$

where $\hat{x} = t\tilde{x} + (1-t)x$ with $0 \leq t \leq 1$.
The final loss function for the critic model is:

$$L = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1\right)^2\right] \tag{1.10}$$

Important to note is that, with this approach, batch normalization must be avoided for the critic, because the correlations it creates between samples in the same batch can lower the effectiveness of the gradient penalty.
At the cost of a bit more computation, experiments show that WGAN with Gradient Penalty (WGAN-GP) produces better quality images than WGAN with gradient clipping.

CHAPTER 2

Dataset and preprocessing

2.1 Dataset

The dataset used for this work is the hiPSC Single-Cell Image Dataset introduced in [26]. This dataset comprises 3D, high-resolution images of over 200,000 live cells from 25 isogenic human induced pluripotent stem cell (hiPSC) lines from the Allen Cell Collection [1].
HiPSCs are growed in tightly packed, epithelial-like monolayer colonies, on Matrigel-coated glass plates that are compatible with high-resolution imaging while preserving their normal pluripotent state.
Each line contains one fluorescently tagged protein, created via endogenous CRISPR/Cas9 gene editing, representing a key cellular structure or organelle.
The cells were imaged in 3D using spinning-disk confocal microscopes. To reference the locations of fluorescent protein (FP)-tagged cellular structures relative to the cell boundary and the nucleus or mitotic chromosomes, for each cell are included fluorescent cell-membrane and DNA dyes. Cells were imaged live and in 3D, as a z-stack of two-dimensional images, at high resolution (120x magnification, 1.25 NA), generating 18 186 fields of view (FOVs) in four acquisition channels,

[1] https://www.allencell.org/cell-catalog.html

representing the FOV-specific protein, the cell membrane, the DNA, and the transmitted-light channel.
Both the FOV images and the single-cell dataset are available as downloadable files as Quilt packages [2] and through interactive online visual-analysis tools [3].

2.1.1 Cell lines and culturing

Each cell line created through gene editing originated from the parental WTC-11 hiPS cell line and featured an endogenously tagged fluorescent protein associated with a specific cellular structure. The cell lines were generated using CRISPR–Cas9-mediated genome editing techniques. Fifteen additional cell lines from the Allen Cell Collection were also produced using the same methods.

All these cell lines were maintained on an automated cell-culture system developed on a Hamilton Microlab STAR Liquid Handling System by the Hamilton Company. They were cultured in a Cytomat 24 (Thermo Fisher Scientific) incubator at 37 °C and 5% CO_2 in mTeSR1 medium with or without phenol red (STEMCELL Technologies), supplemented with 1% penicillin–streptomycin (Thermo Fisher Scientific). For imaging purposes, the cells were plated on Matrigel-coated glass-bottom, black-skirt, 96-well plates with 1.5 optical grade cover glass (Cellvis). [26]

2.1.2 Microscopy and image acquisition

Imaging was conducted using three identical ZEISS spinning-disk confocal microscopes equipped with either 10×/0.45 NA Plan-Apochromat or 100×/1.25 W C-Apochromat Korr UV Vis IR objectives (Zeiss) and

[2] https://open.quiltdata.com/b/allencell/packages/aics/hipsc_single_cell_image_dataset
[3] https://cfe.allencell.org/

ZEN 2.3 software (blue edition; ZEISS). These spinning-disk confocal microscopes featured a 1.2× tube lens adapter, resulting in final magnifications of 12× or 120×, respectively. In our work, we used only the images with 120x magnification.
Standard laser lines were employed with the following laser powers when using 10× objectives: 405 nm at 0.28 mW, 488 nm at 2.3 mW, 561 nm at 2.4 mW, and 640 nm at 2.4 mW. Emission from specific fluorophores was collected using Band Pass (BP) filter sets (Chroma): 450/50 nm for DNA dye, 525/50 nm for mEGFP tag, 600/50 nm for mTagRFP-T tag, and 706/95 nm for cell-membrane dye detection. Image acquisition had an exposure time of 200 ms. Cells were imaged in a phenol red-free mTeSR1 medium on microscope stages equipped with a humidified environmental chamber, maintaining cells at 37 °C with 5% O2 during imaging. Transmitted light (bright-field) images were acquired using either a white LED light source with a broad emission spectrum or a red LED light source with a peak emission of 740 nm and a narrow range, along with a BP filter 706/95 nm for bright-field light collection. Fast z-acquisition was facilitated by using a Prior NanoScan Z 100 mm piezo z stage (ZEISS).

For image acquisition, after selecting the field of view (FOV) position from the well overview acquisition, cells' DNA was initially stained for 20 minutes with NucBlue Live (Thermo Fisher Scientific). Subsequently, the cell membrane was stained with CellMask Deep Red (CMDR, Thermo Fisher Scientific) while NucBlue Live was still present, and the cells were washed once before imaging, within a maximum time frame of 2.5 hours. Three-dimensional FOVs at 120× magnification were acquired at the pre-selected positions. Four channels were captured at each z-step (interwoven channels) in the following sequence: bright field, mEGFP or mTagRFP-T, CMDR, and NucBlue Live. [26]

2.2 Preprocessing

Portions of each FOV suffer from defocus aberration and noise. This is especially relevant for the intended target channels and thus the training of our model. The dataset was therefore processed to mitigate these problems.

2.2.1 Selection of in-focus slices: PLLS

In order to automatically select the in-focus z-planes in the z-stack, we compute the power spectrum log-log slope (PLLS) [5].
This metric evaluates the slope of the power spectral density of the pixel intensities on a log-log scale. The power spectrum shows the strength of the spatial frequency variations as a function of frequency. It is always negative and usually decreases in value as blur increases and high-frequency image components are lost (more negative values indicate a steeper slope, which means that the image is composed mostly of low spatial frequencies).
In our case, PLLS works well to spot the on-focus region, but behaves the opposite as expected regarding its values: PLLS is consistently more negative for on-focus images. We believe this behavior to be due to the presence of noise. Experiments with denoising suggested that in our images the high-frequencies don't represent structure but rather noise, which is equally present in on-focus and out-of-focus images. The difference resides in medium frequencies, that in our images encode the structure we want to visualize. The higher power of medium frequencies in on-focus images makes the power spectrum slope steeper, and hence more negative (see Fig. 2.1).

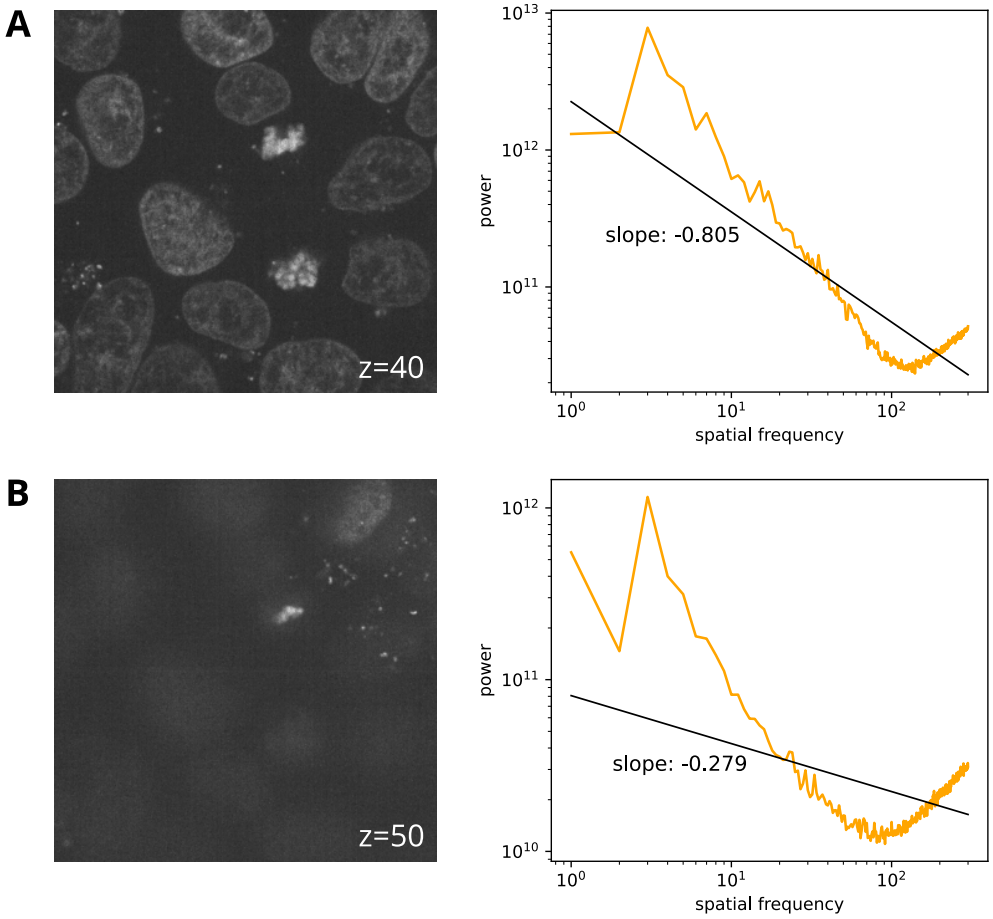

Figure 2.1: Power spectrum plots for an in-focus slice (**A**) and a defocused slice (**B**) of the same sample (DNA channel).

2.2.2 Denoising: Noise2Void

The other problem we have in our target images is the presence of noise. When we train a model with the generative adversarial framework, noise in the ground truth images becomes a problem because it gives the discriminator an easy way to distinguish between real and generated images, making the discriminator's job even easier than it already is, with bad consequences for training stability.

To denoise our target images we use the Noise2Void model [13], a deep learning based technique that exploits the assumption that signal has a predictable structure, while noise doesn't, to make it possible to train the denoising images directly on the body of data to be denoised. So

there's no need for clean targets.

To do this, it employs a blind-spot network that operates by masking the central pixel of the image patch. (Figure 2.2). During training, the network tries to reconstruct the input image patch but excludes information from the center pixel of the patch. This creates a "blind spot" that forces the network to rely on the surrounding context to predict the clean value of the masked pixel.

By training to accurately predict the missing value at the blind spot, the network essentially learns to distinguish between the underlying image structure (signal) and the random noise.

Once trained, the network leverages the knowledge gained from predicting missing values to effectively remove noise and reveal the clean image content.

Noise2Void made it possible to denoise our dataset with results comparable to classical methods like block-matching and 3D filtering (BM3D) in a fraction of the time and without the need to estimate the amount of noise beforehand.

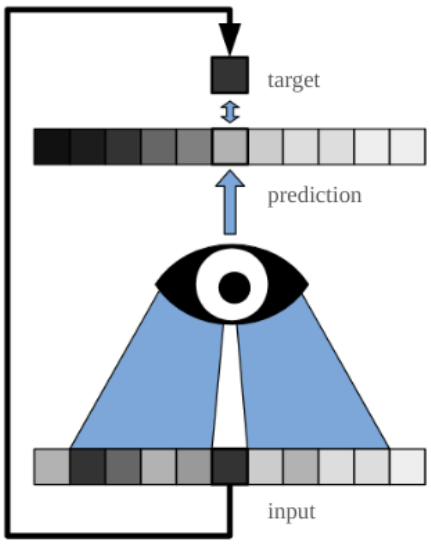

Figure 2.2: **Blind-spot network:** The receptive field of each pixel excludes the pixel itself, preventing the model from learning the identity.

CHAPTER 3

Methodology

3.1 Architecture

Our model is a WGAN-GP [2, 7], where the generator is a U-Net [22] and the critic follows the design of a PatchGAN discriminator (Figure 3.3) [12].

3.1.1 Generator

Our generator is a U-Net of depth 3. The downsampling path is composed of convolutional blocks with 64, 128, and 256 filters respectively. Each block is composed by two identical sub-blocks made of a 3D convolution with a kernel size of 3, stride 1, and padding 1, followed by instance normalization and Leaky ReLU activations. Between blocks, a 3D convolution with kernel size 2 and stride 2 performs the downsampling halving the spatial dimensions.

The upsampling path is symmetric to the downsampling, with transposed convolutions. The skip connections from the downsampling path are concatenated with the corresponding feature maps during the upsampling path to facilitate information flow across different scales. The final layer employs a 3D convolution with kernel size 3 and padding

1 to generate the 6 fluorescent channels as the output.

In total, our generator has 23.1 M trainable parameters.

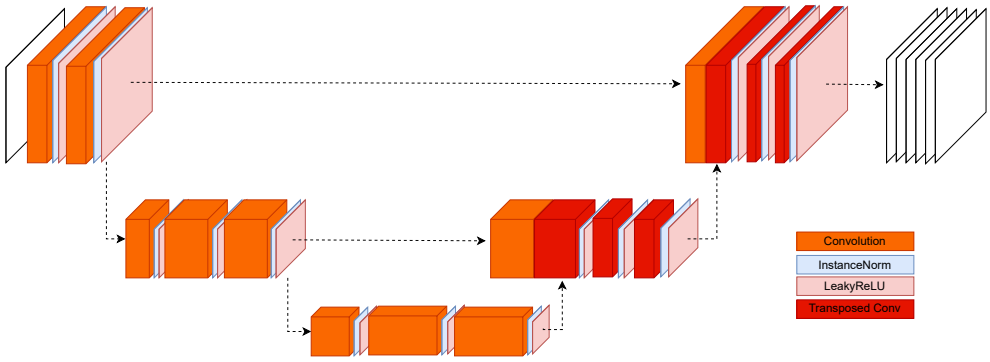

Figure 3.1: **Generator architecture diagram**
Downsampling path composed of convolutional blocks with 64, 128, and 256 filters respectively. Each block is made of two identical sub-blocks (3D convolution with a kernel size of 3, stride 1, and padding 1, followed by instance normalization and Leaky ReLU activations). Between blocks, a 3D convolution with kernel size 2 and stride 2 performs the downsampling halving the spatial dimensions. The upsampling path is symmetric to the downsampling, with transposed convolutions. The skip connections from the downsampling path are concatenated with the corresponding feature maps during the upsampling path. The final layer employs a 3D convolution with kernel size 3 and padding 1 to generate the 6 fluorescent channels as the output.

3.1.2 Critic

The critic is composed of five convolutional blocks. The first one is followed just by a Leaky ReLu activation, while the others perform instance normalization before. The first three blocks perform convolutions with a kernel size of 4, stride of 2, and padding of 1, while the last two preserve the spatial dimensions using a kernel size of 3, stride of 1, and padding of 1. The final layer has a receptive field of size 54, so the critic model can only penalize unrealistic structure at this scale (Figure 3.3). In total, our critic has 6.2 M trainable parameters.

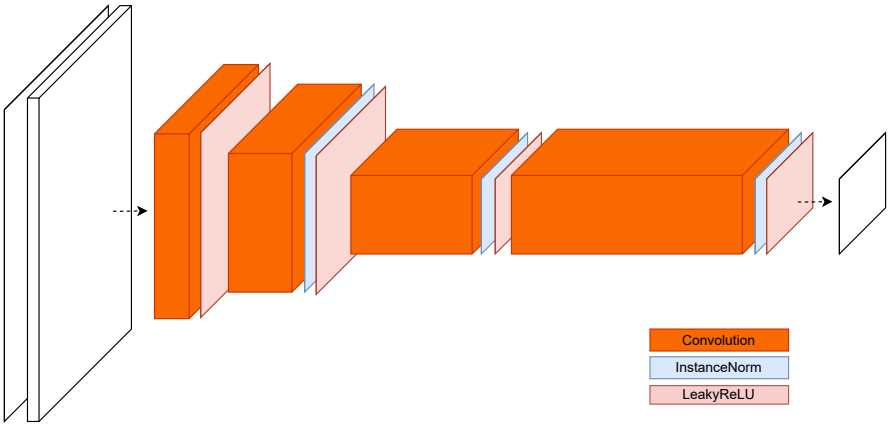

Figure 3.2: **Discriminator architecture diagram**
Five convolutional blocks in succession: the first one is followed just by a Leaky ReLu activation, while the others perform instance normalization before. The first three blocks perform convolutions with a kernel size of 4, stride of 2, and padding of 1, while the last two preserve the spatial dimensions using a kernel size of 3, stride of 1, and padding of 1.

3.2 Training Setup

3.2.1 Loss function

The loss function of our generator model is a weighted sum between L1 distance and the adversarial loss calculated by a PatchGAN discriminator, which encourages local realism.

Each ground truth sample presents just three populated channels (DNA, cell membrane, and specific structure) while the others are just populated by NaN values. To deal with this, we mask the image produced by the generator so that it has only zeros in the channels corresponding to the void channels of the target image. These channels in the target image are processed so that the NaN values become zeros. In this way, we can compute the pixel-wise component of our loss on the whole image. Note that the model now would have a perfect score on the void channels: they are populated with zeros both in the generated image

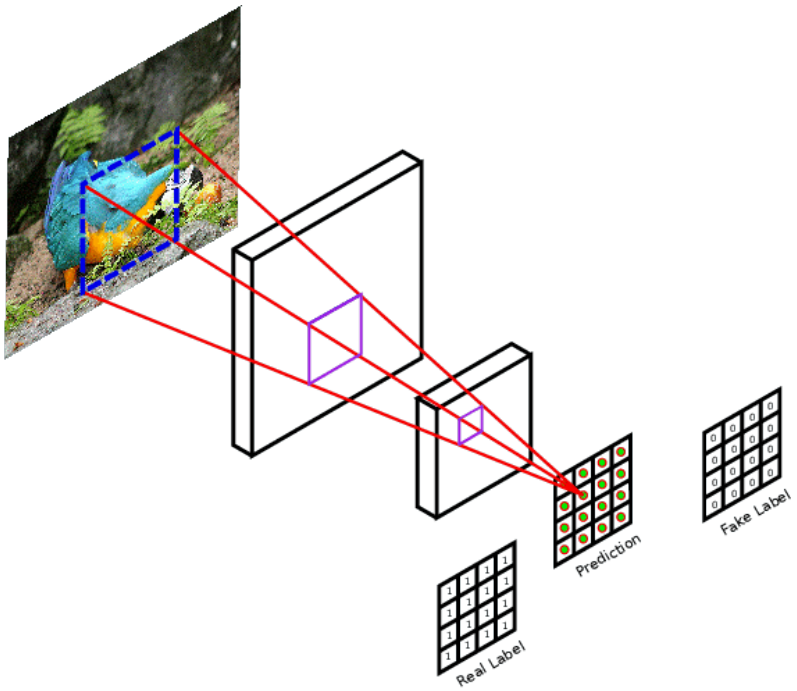

Figure 3.3: **PatchGAN discriminator.** Each value of the output matrix represents the probability of whether the corresponding image patch is real or artificially generated. [8].
Since we are using a Wasserstein GAN architecture, our PatchGAN critic differs in the fact that each value of the output matrix is a real number representing the "realness score" that the critic assigns to that specific image patch, and not a probability.

and in the target image. This is why we compute our loss without reduction, we mask it using the same mask we used before, and then we compute the mean.

Regarding the adversarial component of the loss, we feed the critic model with the masked generated image, so that it has the same amount of void channels as a real sample, and the critic can't use the number of non-zero channels as additional info to distinguish between fake and real samples.

3.2.2 Hyperparameters

Activation functions Early experiments were conducted both with ReLU and with LeakyReLU. LeakyReLU was found to be the most promising.
This is in accordance to what general empirical knowledge tells about the choice between these two similar activation functions: LeakyReLU usually performs better for regression tasks, with deep architectures, and when data has a lot of noise or outliers.
Once LeakyReLUs were selected as the model's activation function, we also performed hyperparameter optimization on their negative slope. Experiments have shown an improved performance, with sharper contours in the generated images, when we used higher non-linearity (smaller negative slope). The final value selected was 0.05.

Batch size Experiments showed better performances with bigger batch sizes. Because of memory constraints given by the size of the files we are processing, we selected a batch size of 16 z-stacks.

Optimizer RMSprop was found to lead to more stable training of the WGAN. The training was performed with a constant learning rate of 0.00005, both for the generator model and for the critic model.

3.2.3 Training loop

Here is the pseudocode for the training loop:

Algorithm 1 WGAN-GP.
Hyperparameters: $\lambda_{\text{adv}} = 0.05$, $\lambda_{\text{gp}} = 10$, $n_{\text{critic}} = 5$, $lr = 0.00005$, $m = 16$

Require: The adversarial loss weight λ_{adv}, the gradient penalty coefficient λ_{gp}, the number of critic iterations per generator iteration n_{critic}, the batch size m, the RMSProp learning rate lr.
Require: Initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
  for $t = 1, \dots, n_{\text{critic}}$ **do**
    for $i = 1, \dots, m$ **do**
      Sample real data: a pair $(x, y) \sim \mathbb{P}_r$, where x is the source image and y represents the fluorescent channels
      $\tilde{y} \leftarrow G_\theta(x)$
      $\tilde{y} \leftarrow \text{mask}(\tilde{y})$
      Sample a random number $\epsilon \sim U[0, 1]$
      $\hat{y} \leftarrow \epsilon x + (1 - \epsilon)\tilde{y}$
      $L^i \leftarrow D_w((x, \tilde{y})) - D_w((x, y)) + \lambda \left(\left\| \nabla_{\hat{y}} D_w(\hat{y}) \right\|_2 - 1 \right)^2$
    end for
    $w \leftarrow \text{RMSProp}\left(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, lr \right)$
  end for
  $\theta \leftarrow \text{RMSProp}\left(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \left(\|y - \tilde{y}\|_1 - \lambda_{\text{adv}} D_w\left((x, \tilde{y})\right) \right), \theta, lr \right)$
end while

CHAPTER 4

Evaluation

Evaluating the quality of synthesized images is an open and difficult problem [24]. Using pixel-wise metrics for evaluation presents the same issues as using them as losses for training (e.g. rewarding blurry results). Moreover, the absolute value of image metrics is greatly influenced by the characteristics of the data, so these metrics can be appropriately used only for the purpose of comparing models on the same data.

When dealing with cellular data, a significant drawback is that they treat every pixel in the image equally, even though pixels representing cellular structures are undoubtedly more critical than background (empty) pixels. Additionally, some image channels, like the nucleoli channel, exhibit sparser content compared to other channels, resulting in a difference in the number of pixels that matter versus background pixels.

Classical evaluation frameworks for GANs make use of inception score (IS) or Fréchet inception distance (FID), but the fact that these methods don't use a specialized encoder trained on our dataset makes them unreliable for the evaluation of our model because the pre-trained models learned features are ineffective on specific domains that are far from the ImageNet dataset [16].

4.1 Perceptual classification metric

To better evaluate our model we designed a perceptual classification metric following the intuition that if the generated images are realistic, a classifier trained on real images will have good accuracy when classifying the synthesized images.
We trained a ResNet classifier to recognize which structure it is looking at: given a 3D patch of one of the channels of a ground truth image, the model outputs to which channel it belongs.
When we compare the performance of this classifier on the images generated by different models, we have a measure of how much the structure present in the output images is realistic and preserves the features that make it distinguishable from other cellular structures.

4.2 Results

The U-Net trained with L^1 loss optimizes image metrics like structural similarity index measure (SSIM), MSE, mean absolute error (MAE), and Pearson correlation coefficient (PCC) better than the WGAN model (Table 4.1), but this doesn't correspond to better and more realistic output images. This is in contrast to what was found in [7], where the WGAN model also showed better performance for image metrics. Their WGAN model was not trained from scratch but starting from a U-Net generator pretrained to minimize L^1 loss. We also experimented with this training setup, finding intermediate results between the U-Net and our WGAN, as expected, both for image metrics and for visual human analysis.

Table 4.1: Image metrics for each channel for the two models, computed on the test set. F_1clf is the F_1 score of the classifier on the images produced by the two models. The values corresponding to the best-performing model for each channel metric are highlighted in bold.

Channel	Model	SSIM	MSE	MAE	PCC	F_1 clf
DNA	U-Net	**0.75 ± 0.08**	**0.4 ± 0.1**	0.42 ± 0.06	**0.79 ± 0.07**	0.9750
	WGAN	0.73 ± 0.09	0.43 ± 0.14	**0.39 ± 0.07**	0.76 ± 0.08	**0.9796**
Cell mem.	U-Net	**0.69 ± 0.09**	0.6 ± 0.1	**0.43 ± 0.06**	**0.65 ± 0.08**	0.8908
	WGAN	0.64 ± 0.09	0.6 ± 0.1	0.46 ± 0.06	0.61 ± 0.08	**0.9889**
Mito.	U-Net	**0.79 ± 0.05**	**0.35 ± 0.06**	**0.29 ± 0.03**	**0.81 ± 0.04**	0.9086
	WGAN	0.77 ± 0.05	0.39 ± 0.08	0.31 ± 0.04	0.79 ± 0.05	**0.9528**
N. env.	U-Net	**0.85 ± 0.05**	**0.19 ± 0.04**	**0.24 ± 0.02**	**0.89 ± 0.02**	0.7707
	WGAN	0.82 ± 0.06	0.25 ± 0.05	0.27 ± 0.03	0.86 ± 0.03	**0.9410**
Myofib.	U-Net	**0.72 ± 0.09**	0.5 ± 0.2	**0.43 ± 0.07**	0.7 ± 0.1	0.9855
	WGAN	0.7 ± 0.1	0.5 ± 0.2	0.47 ± 0.09	0.7 ± 0.1	**0.9919**
Nucleoli	U-Net	**0.93 ± 0.03**	**0.14 ± 0.05**	**0.15 ± 0.02**	**0.92 ± 0.03**	0.9500
	WGAN	0.91 ± 0.04	0.19 ± 0.06	0.17 ± 0.03	0.90 ± 0.04	**0.9789**

The perceptual classification metric we designed is in better accordance with the human eye evaluation of the produced images, giving us a better proxy for what we want to evaluate than classical metrics. In fact, our WGAN model, which produces more realistic images, with sharper edges and finer details, produces images that are recognized better by the classifier (Fig. 4.1).

A visual analysis of the model's predictions shows how the predicted structures have sharp edges and coherence, even for long and thin structures (Figure 4.2). In the DNA channel, it is often possible to see regions inside the structures with a different intensity, corresponding to the nucleoli. These structures are often less visible in ground truth images because of noise and the low precision of the chromatic dyes, making the digital staining approach even more effective and precise than classical chemical staining in locating these structures (Figure 4.3). This perk may be attributable to the simultaneous training on

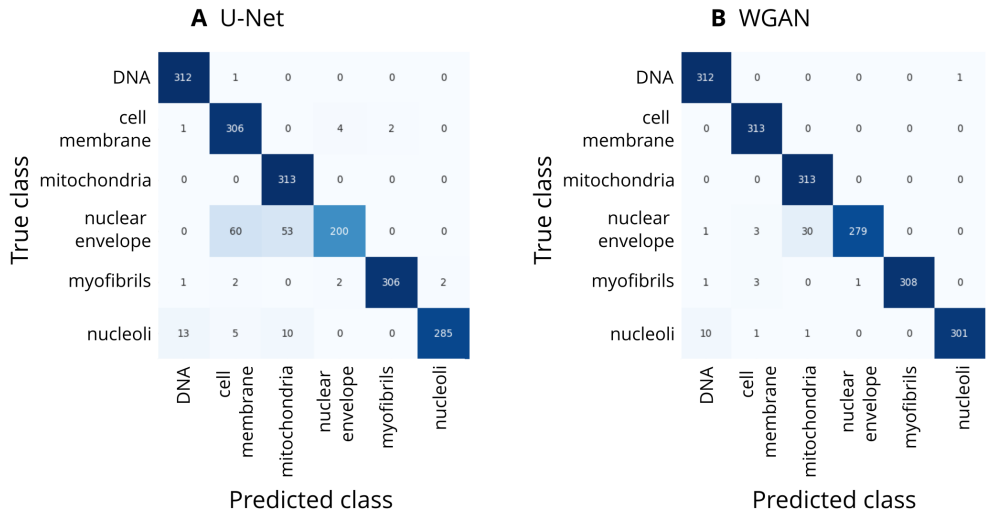

Figure 4.1: Confusion matrices of the classifier on images generated by (**A**) the U-Net trained with fixed L^1 loss and by (**B**) the WGAN-GP (right). The outputs produced by the WGAN are recognized much better by the classifier.

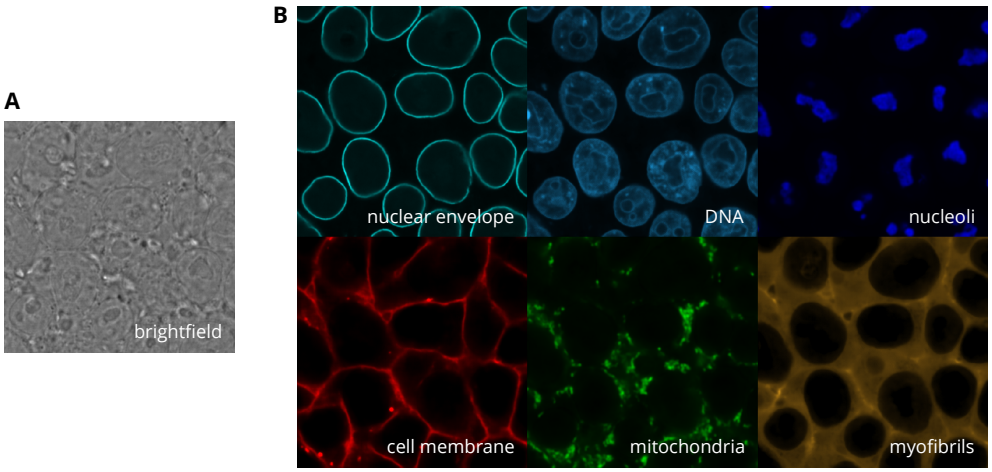

Figure 4.2: Input image (**A**) and WGAN model prediction (**B**) from a sample not seen in training. Each structure is shown through the central z-plane (512 × 512 pixels) taken from the 3D cube (32 × 512 × 512 pixels).

many channels: the network has access to information about the position of nucleoli from the nucleoli channel and can integrate its DNA channel prediction using this information.

In general, the multi-channel structure of the model ensures inter-structure coherence. The nuclear envelope is always perfectly aligned to the edge

of the DNA (nucleus) channel, mitochondria and myofibrils are always located in the space between the nucleus and the cell membrane, etc. In Figure 4.4 we can see some inference results of the U-Net model, trained with the same architecture and training setup as our final WGAN model, whose inference results are shown in Figure 4.3, but without the adversarial component of the loss. The differences are small but clearly visible. The DNA channel produced by the U-Net model has a more uniform and less realistic brightness. The cell membrane is less smooth and with less sharp edges. The biggest difference is visible in the myofibrils channel, where the WGAN model produces images with sharper contours for the filaments that follow the cell membranes, keeping them more visible than the rest of the structure that correctly remains at a lower pixel intensity. Overall, the images produced by the WGAN model seem more realistic and have sharper edges, as expected considering the theoretical advantages of the conditional GAN approach over the classical U-Net trained to minimize a fixed pixel-wise loss function.

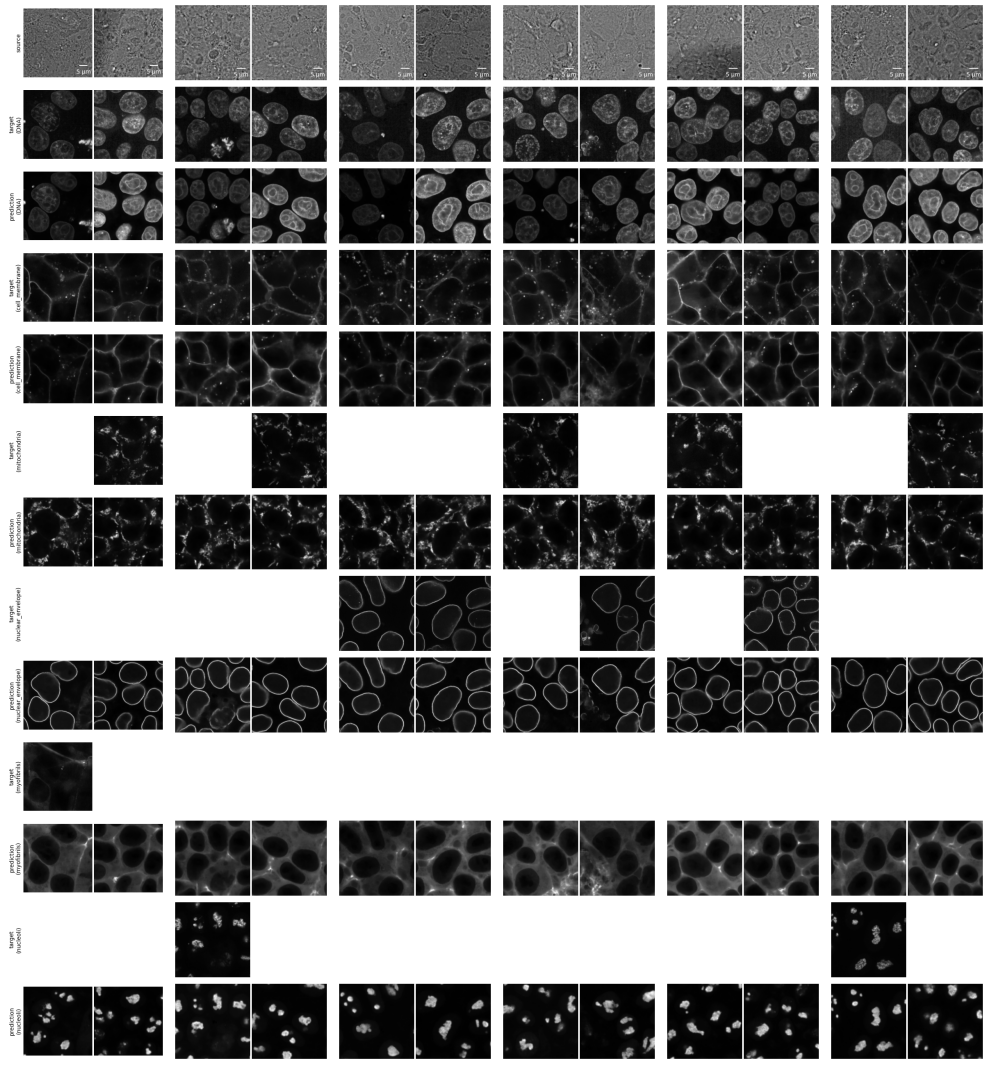

Figure 4.3: **Some inference results of the WGAN model on the validation set.** Each column represents a different sample. Rows are alternating between ground truth images and generated predictions. Each column has only three target channels. Predictions are shown also for channels where the ground truth is not available for comparison. The model produces 3D images, here is shown the central z-slice. Each 2D slice is 384 × 384 pixels.

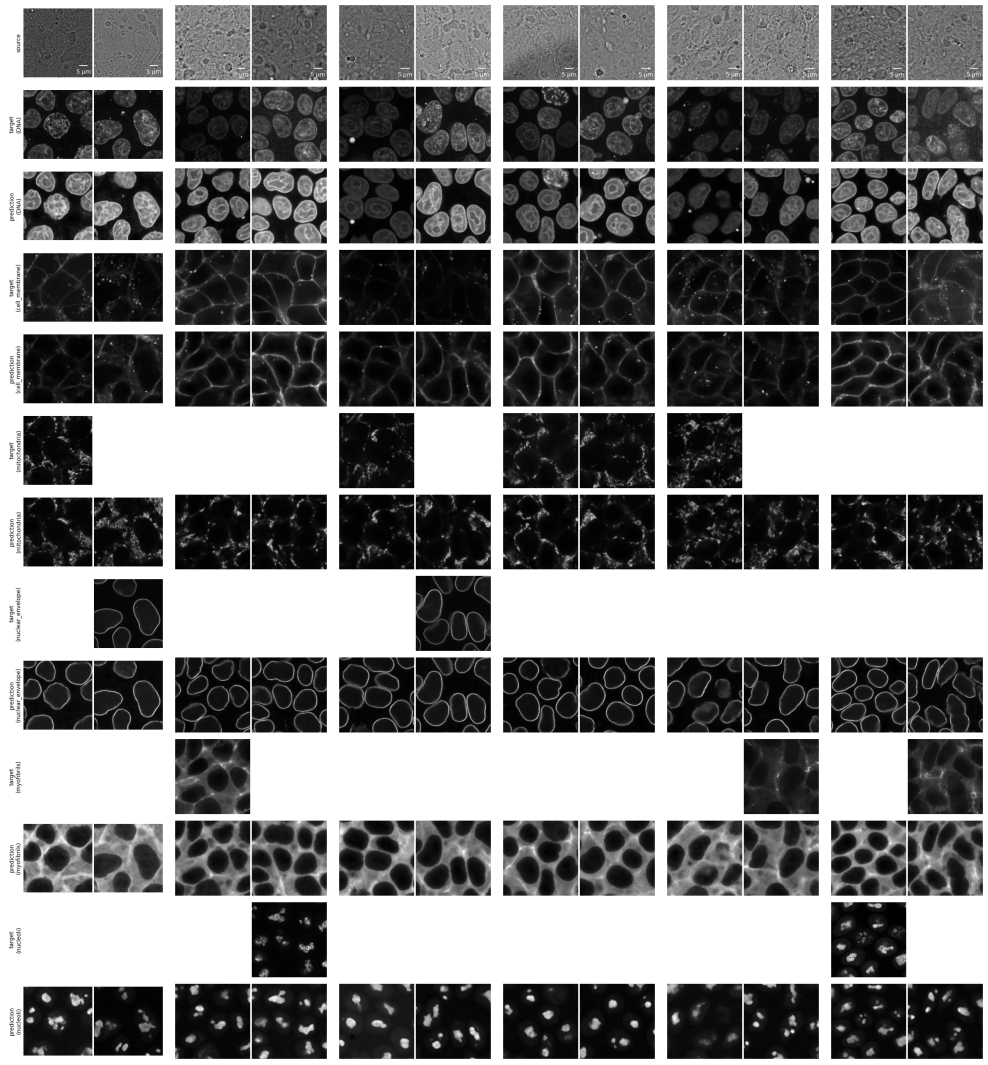

Figure 4.4: **Some inference results of the U-Net model on the validation set.** Each column represents a different sample. Rows are alternating between ground truth images and generated predictions. Each column has only three target channels. Predictions are shown also for channels where the ground truth is not available for comparison. The model produces 3D images, here is shown the central z-slice. Each 2D slice is 384 × 384 pixels.

Conclusion

Our study introduces a Wasserstein GAN model for three-dimensional virtual fluorescent staining. The model, inspired by prior works, enhances its predictive capabilities by training on 3D images and adopting a masked loss strategy for handling multiple fluorescent labels simultaneously. The use of a conditional WGAN-GP model ensures stability in training and addresses coherence and finer details, while the model's architecture allows it to be agnostic to the size of input images, eliminating the need for patch stitching.

4.3 Limitations and future work

Despite the promising contributions of our study, we should acknowledge some limitations:

- **Training data constraints:** Histological characteristics can vary significantly among different tissues, introducing challenges in capturing the diverse structures and textures. Moreover, variations in lighting, staining protocols, imaging equipment, and techniques must be considered. The model may struggle to generalize well to tissues and conditions not represented in the training data, potentially leading to unreliable predictions. It was observed that models that are initially trained using one cell type, such as hiPSC, exhibit reduced performance when dealing with

inputs featuring significantly distinct cellular morphologies. Additionally, model accuracy decreases when attempting to predict fluorescence images from input transmitted light (TL) stacks acquired with a shorter inter-slice interval compared to the interval used in the training data. [18] Our model is trained on cell cultures and not tissue samples. While there are many reasons to expect it to perform similarly well if trained on tissue samples, this has to be checked.

- **Limitations common to every digital staining approach:** Fluorescent channels carry a substantially greater amount of information compared to a brightfield z-stack. Consequently, the prediction of complete fluorescent channels from brightfield images represents a significantly more complex and demanding task.
  For some structures, such as sarcomeres, the association between transmitted light (TL) and fluorescence images is inherently weaker, leading to worse prediction performances.
  Moreover, it's important to note that there might not exist a direct quantitative correlation between the predicted intensity of a tagged structure and the actual protein levels.

- **Masked loss strategy sensitivity to sparsity:** The effectiveness of the masked loss strategy is influenced by the distribution of fluorescent labels in the training dataset. If certain labels are sparse or imbalanced, the model may prioritize more prevalent labels during training, potentially leading to suboptimal performance for rare or less frequent structures.

- **Clinical validation:** Although our model demonstrates promising results in generating 3D images of fluorescent channels, its clinical applicability and accuracy in real-world scenarios need further validation through rigorous testing on diverse clinical

datasets.

To address the limitations regarding training data, future work should be aimed at the creation of more diverse and comprehensive histological datasets. This includes incorporating a broader range of tissue types and imaging conditions. Diversifying the training data will enhance the model's generalizability to a wider array of tissues and clinical scenarios.
Our evaluation method could be improved by pairing our classification perceptual metric with a deep perceptual metric that compares the deep activation of the classifier when looking at real or generated fluorescence channels [30]. It would be interesting to experiment adding a term in the loss function during training representing this deep perceptual loss, like in [9].

The potential impact of this work extends to making subcellular and tissue examination more accessible and facilitating in vivo imaging, marking a significant contribution to the field of deep learning-based histological staining methods.

Acknowledgments

This study was supported by the European Union-funded project OrganVision under Grant agreement No. 964800.

Bibliography

[1] M. Arjovsky and L. Bottou. *Towards Principled Methods for Training Generative Adversarial Networks*. Jan. 2017. DOI: 10.48550/arXiv.1701.04862. arXiv: 1701.04862 [cs, stat] (cit. on pp. 17–19).

[2] M. Arjovsky, S. Chintala, and L. Bottou. *Wasserstein GAN*. Dec. 2017. DOI: 10.48550/arXiv.1701.07875. arXiv: 1701.07875 [cs, stat] (cit. on pp. 5, 20, 29).

[3] B. Bai et al. "Deep Learning-Enabled Virtual Histological Staining of Biological Samples". In: *Light: Science & Applications* 12.1 (Mar. 2023), p. 57. ISSN: 2047-7538. DOI: 10.1038/s41377-023-01104-7 (cit. on pp. 6–8).

[4] Y. Benny et al. "Evaluation Metrics for Conditional Image Generation". In: *International Journal of Computer Vision* 129.5 (May 2021), pp. 1712–1731. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-020-01424-w. arXiv: 2004.12361 [cs, eess].

[5] M.-A. Bray et al. "Workflow and Metrics for Image Quality Control in Large-Scale High-Content Screens". In: *Journal of Biomolecular Screening* 17.2 (Feb. 2012), pp. 266–274. ISSN: 1087-0571. DOI: 10.1177/1087057111420292 (cit. on p. 26).

[6] E. M. Christiansen et al. "In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images". In: *Cell* 173.3 (Apr. 2018), 792–803.e19. ISSN: 1097-4172. DOI: 10.1016/j.cell.2018.03.040 (cit. on pp. 7, 9, 13).

[7] J. O. Cross-Zamirski et al. "Label-Free Prediction of Cell Painting from Brightfield Images". In: *Scientific Reports* 12.1 (June 2022), p. 10001. ISSN: 2045-2322. DOI: 10.1038/s41598-022-12914-x (cit. on pp. 7, 12, 13, 29, 36).

[8] U. Demir and G. Unal. "Patch-Based Image Inpainting with Generative Adversarial Networks". In: (Mar. 2018) (cit. on p. 32).

[9] A. Dosovitskiy and T. Brox. *Generating Images with Perceptual Similarity Metrics Based on Deep Networks*. Feb. 2016. arXiv: 1602.02644 [cs] (cit. on p. 44).

[10] I. Gulrajani et al. *Improved Training of Wasserstein GANs*. Dec. 2017. arXiv: 1704.00028 [cs, stat] (cit. on pp. 5, 22).

[11] J. Hui. *GAN — Wasserstein GAN & WGAN-GP*. May 2021.

[12] P. Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. Nov. 2018. DOI: 10.48550/arXiv.1611.07004. arXiv: 1611.07004 [cs] (cit. on pp. 12, 15–17, 29).

[13] A. Krull, T.-O. Buchholz, and F. Jug. *Noise2Void - Learning Denoising from Single Noisy Images*. Apr. 2019. DOI: 10.48550/arXiv.1811.10980. arXiv: 1811.10980 [cs] (cit. on p. 27).

[14] A. B. L. Larsen et al. *Autoencoding beyond Pixels Using a Learned Similarity Metric*. Feb. 2016. DOI: 10.48550/arXiv.1512.09300. arXiv: 1512.09300 [cs, stat].

[15] C. Ledig et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. May 2017. DOI: 10.48550/arXiv.1609.04802. arXiv: 1609.04802 [cs, stat].

[16] S. Liu et al. *An Improved Evaluation Framework for Generative Adversarial Networks*. July 2018. DOI: 10.48550/arXiv.1803.07474. arXiv: 1803.07474 [cs] (cit. on p. 35).

[17] X. Meng, X. Li, and X. Wang. "A Computationally Virtual Histological Staining Method to Ovarian Cancer Tissue by Deep Generative Adversarial Networks". In: *Computational and Mathematical Methods in Medicine* 2021 (July 2021), e4244157. ISSN: 1748-670X. DOI: 10.1155/2021/4244157 (cit. on p. 7).

[18] C. Ounkomol et al. "Label-Free Prediction of Three-Dimensional Fluorescence Images from Transmitted-Light Microscopy". In: *Nature Methods* 15.11 (Nov. 2018), pp. 917–920. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0111-2 (cit. on pp. 7, 11, 13, 43).

[19] D. Pathak et al. *Context Encoders: Feature Learning by Inpainting*. Nov. 2016. DOI: 10.48550/arXiv.1604.07379. arXiv: 1604.07379 [cs] (cit. on pp. 16, 17).

[20] A. Picon et al. "Autofluorescence Image Reconstruction and Virtual Staining for In-Vivo Optical Biopsying". In: *IEEE Access* PP (Feb. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3060926 (cit. on p. 7).

[21] Y. Rivenson et al. "Virtual Histological Staining of Unlabelled Tissue-Autofluorescence Images via Deep Learning". In: *Nature Biomedical Engineering* 3 (June 2019). DOI: 10.1038/s41551-019-0362-y (cit. on pp. 7, 11).

[22] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. May 2015. DOI: 10.48550/arXiv.1505.04597. arXiv: 1505.04597 [cs] (cit. on pp. 11, 14, 29).

[23] U. Sara, M. Akter, and M. S. Uddin. "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study". In: *Journal of Computer and Communications* 7.3 (Mar. 2019), pp. 8–18. DOI: 10.4236/jcc.2019.73002.

[24] L. Theis, A. van den Oord, and M. Bethge. *A Note on the Evaluation of Generative Models*. Apr. 2016. arXiv: 1511.01844 [cs, stat] (cit. on p. 35).

[25] D. Ulyanov, A. Vedaldi, and V. Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization*. Nov. 2017. DOI: 10.48550/arXiv.1607.08022. arXiv: 1607.08022 [cs].

[26] M. P. Viana et al. "Integrated Intracellular Organization and Its Variations in Human iPS Cells". In: *Nature* 613.7943 (Jan. 2023), pp. 345–354. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05563-7 (cit. on pp. 23–25).

[27] M. P. Viana et al. *Robust Integrated Intracellular Organization of the Human iPS Cell: Where, How Much, and How Variable*. Preprint. Cell Biology, Dec. 2020. DOI: 10.1101/2020.12.08.415562.

[28] S. Xiang and H. Li. *On the Effects of Batch and Weight Normalization in Generative Adversarial Networks*. Dec. 2017. DOI: 10.48550/arXiv.1704.03971. arXiv: 1704.03971 [cs, stat].

[29] R. Zhang, P. Isola, and A. A. Efros. *Colorful Image Colorization*. Oct. 2016. DOI: 10.48550/arXiv.1603.08511. arXiv: 1603.08511 [cs] (cit. on p. 16).

[30] R. Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. https://arxiv.org/abs/1801.03924v2. Jan. 2018 (cit. on p. 44).