POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Simulation of histological images by Generative Adversarial Networks

Politecnico di Torino

Supervisors

Prof. Massimo SALVI

Candidate

Alen SHAHINI

March 2023

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Prof. Massimo Salvi for his knowledge and the time he has dedicated to me in this thesis work.

The most special thanks go to my mother Klodjana and my brother Davide for supporting me all the days of my life. I thank my family, also my father Andi who is and will always be present.

Finally, I would like to thank my lifelong friends for all the moments we have experienced over the years. I feel I must thank those who have accompanied me along this beautiful journey: Federica and Valerio, you are special.

Abstract

Histological images are crucial for the diagnosis of many diseases. Cellular instance number, density, shape, morphology are key elements in the evaluation of histological images. Individual cellular instance segmentation is important for the extraction of such features. The study of the semantic content makes it possible to understand the interactions between cellular instances and the micro-environment. Deep learning (DL)-based techniques represent the state of the art in the automatic segmentation of individual cellular instances. This thesis work proposes the inversion of the traditional paradigm through artificial intelligence (AI) techniques capable of generating realistic histological images from a predefined ground truth, being able to control the semantic content of the image. This thesis work is part of the Image-to-Image (I2I) paired translation. The public CryoNuSeg dataset is used. It includes 30 paired haematoxylin and eosin (H&E)-stained histological images. Each image has three manual annotations. We propose an algorithm for merging the annotations into a single ground truth, used to train DL networks. The generation of histological images is performed via Generative Adversarial Networks (GANs).

Contents

Introduction

Histology is the branch of biology that studies plant and animal tissues. In medicine it plays an important role in pathological anatomy and the description of pathological phenomena, which is also essential for pre- and post-operative analysis in medical and surgical fields. Histopathology is a branch of pathological anatomy that studies tissue changes at the microscopic level through specific techniques.

Tissue staining is one of the most widely used techniques. They are divided, regardless of the mechanism of action of the staining agent, into histological staining (also called histomorphological) and histochemical staining. Histological staining is performed to make visible the different cell and tissue components (nucleus, cytoplasm, stroma, etc.), which are basically transparent and almost invisible under the microscope. Among histological stainings, the most common and basic one is haematoxylin and eosin (H&E) staining. On the other hand, histochemical staining is performed to identify the chemical nature and location of chemical constituents (molecules or reactive groups) in a tissue.

Digital pathology (DP) refers to the acquisition, management, sharing, visualisation, and interpretation of pathological information within a digital environment.

Whole-slide scanners are used to digitise histological slides. Digitisation does not require significant manual intervention and leads to an optimisation and standardisation of data storage, visualisation and transmission processes. The spread of DP is due to a significant increase in benefits for both pathologists and patients: decreased waiting times, improved accuracy of diagnosis, faster treatment and improved work efficiency of pathologists.

Tissue detection in WSI is a relevant application within digital pathology. The analysis and evaluation of tissues, their shape, number of nuclei, density, and morphology is crucial in the diagnosis of many diseases, including many types of cancer. The decrease in the number of pathologists, the increase in cancer-related diseases, and the presence of considerable inter-operator variability, greatly influenced by a different level of experience among pathologists, inevitably lead to an increased need for automated clinical decision support tools [1, 2, 3, 4]. Such tools facilitate pathologists' tasks such as instance nuclei detection.

Preceding the advent of deep learning (DL), approaches for detection were based on: watershed segmentation; morphological operations such as erosion, dilation, opening and closing; deformable models; thresholding and active contour techniques. These ones are not generally feasible to be applied to a large number of images due to variations in the morphology of the nuclei of different organs and tissues, variations in tissue colour and variability in image characteristics due to differences in the acquisition systems, protocols and environments used.

Machine learning (ML)-based approaches base their operation on the use of features extracted directly from images. The performance of such models is highly dependent on the images from the features extracted and selected for learning.

The automatic feature extraction from images, the possibility of learning information from a large number of images and the high performance have led to an important deployment of DL-based models in clinical and research settings.

Although manual operations, especially by experts in the field, represent the gold standard, in recent years DL-based artificial intelligence tools have proven to be an effective tool for the recognition of features that cannot normally be clearly identified, either by an experienced pathologist. Particularly for those with minor experience, automated ML/DL-based systems exhibit a large potential not only in reducing working time, but also in enhancing diagnostic accuracy.

This thesis work aims to invert the traditional paradigm by generating realistic histological images from a predefined ground truth, used to control the generated images semantic content, through generative models.

Chapter 1

Generative Adversarial Networks in Digital pathology

Within generative models, GANs are becoming increasingly popular, both in the field of image processing and signal processing. GANs [5] are artificial intelligence tools that base their operation on the competitive training between two neural networks, a generator and a discriminator. This architecture enables the neural network to learn to create new data that has the same distribution as those used in the training phase. The $G(z)$ generator maps random noise, $z \sim p_z(z)$, from the source domain into samples as similar as possible to the target domain data, $x \sim p_{data}(x)$, and the discriminator $D(x)$ aims to identify real data from generated data. During training, while the discriminator attempts to maximise the distance

between the actual data distribution and the generated data distribution, the generator tries to confound the discriminator by minimising this distance. GAN's ultimate goal is to reach equilibrium in a min-max problem:

$$\min_G \max_D \mathcal{L}(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (1.1)$$

In the imaging field, GANs have various applications: video prediction, text-based image generation, complex object generation, image detail enhancement, new product development phase. They were first introduced by Radford [6]. State-of-the-art GANs such as StyleGAN [7] and BigGAN [8] are capable of generating high-resolution images. Self-Attention GANs [9], Spectral Normalisation GANs [10] and BigGAN have generated high diversity images in datasets such as ImageNet [11].

It is essential to analyse the state-of-the-art of GANs applied to histological images. The study of the state of the art is conducted on PubMed, Google Scholar and Picopolito.

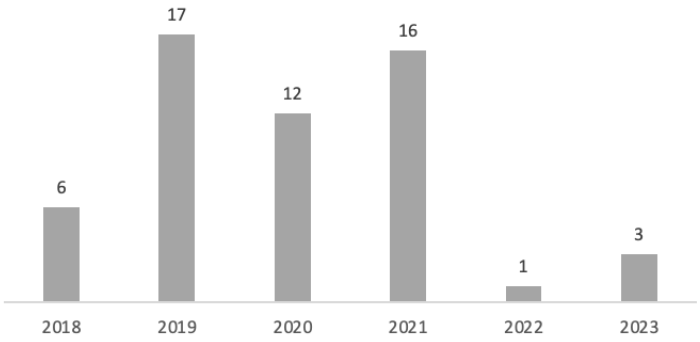

Figure 1.1: On PubMed, the most possible general search 'generative adversarial network* histolog* imag*' yielded 55 articles in all. The histogram shows the number of articles per year.

It can be deduced (Fig. 1.1) that GANs in the field of histological imaging started to be discussed around 2018. The topic is recent. The number of articles from 2018 to 2022 has remained stable. The articles include very different applications: detection, segmentation, data augmentation, normalisation of datasets. For a certainly not large number of articles, the applications and objectives are different. From these articles two reviews analyse the state of the art of GANs in digital pathology and imaging processing of histological images following two different classifications: according to the area of application [12] and according to the input and output of GANs [13].

The applications are categorised into two areas: image processing and image analysis. The latter is the most significant for the thesis aim. It includes image generation and image detail enhancement. Furthermore, the GAN architectures that find application in histological imaging can be divided into 3 classes: Z2I, I2I, I2L. Z2I (Latent-to-Image) refers to GANs that take noise as input and give an output image by passing through what is called latent space. I2I (Image-to-Image), similarly, takes an image as input and outputs another image. The I2L (Image-to-Label) takes images as input and outputs class labels. This thesis work fits within the I2I category, in which the most widely used architectures are Pix2Pix [14] and cycleGAN [15]. The Pix2Pix is a cGAN (conditional GAN) designed to work on paired data. The cycleGAN, on the other hand, works on unpaired data.

This thesis work aims to extract semantic content knowledge from a distribution of histological images used as target domain starting from a distribution of manual annotations used as source domain. The goal is to control the semantic content by modelling the latent space between a predefined ground truth and the histological

image. A direct correspondence between the target domain and the source domain is required. Image generation can be controlled through class labels. The ground truths used for training DL-based models present boundary information on individual cell instances and white areas of the corresponding histological images. The remaining tissues represent the last class.

As Pix2Pix works on paired data it fits perfectly with the objective of this thesis. The cycleGAN architecture is used for the implementation of a model that is always able to work on paired data, despite the fact that cycleGAN was born to work on unpaired data.

1.1 Pix2Pix

Pix2Pix [14] is a conditional GAN in which the source domain data, $x \sim p_{data}(x)$, supplied as input to the generator $G(x)$, is supplied as input to the discriminator together with the target domain data, $y \sim p_{data}(y)$ (Fig. 1.2). This structure is perfectly suited to paired domains in which there is a strong correspondence between the source and target domains.

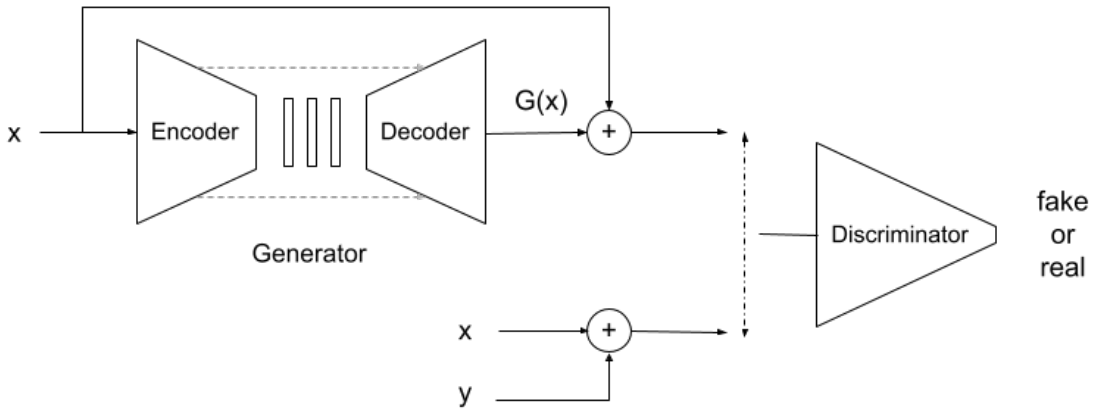

Figure 1.2: Pix2Pix network structure.

Adversarial loss is defined as follows

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,G(x)}[log(1 - D(x, G(x)))] \tag{1.2}$$

The L1 distance between the target data and the generated data

$$\mathcal{L}_{L1}(G) = E_{x,y}[||y - G(x)||_1] \tag{1.3}$$

is implemented. Using λ to control the relevance, the final objective of Pix2Pix

can be expressed as follows

$$\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{1.4}$$

1.2 CycleGAN

CycleGAN [15] is a particular GAN in which there are two generators, G and F, and two discriminators, D_Y and D_X. The two data domains, X and Y, are both source and target domains. The $G(x)$ function maps the data from the X domain to the data from the Y domain. The function $F(y)$ maps the data from domain Y into the data from domain X (Fig. 1.3). Adversarial loss is used for both mapping

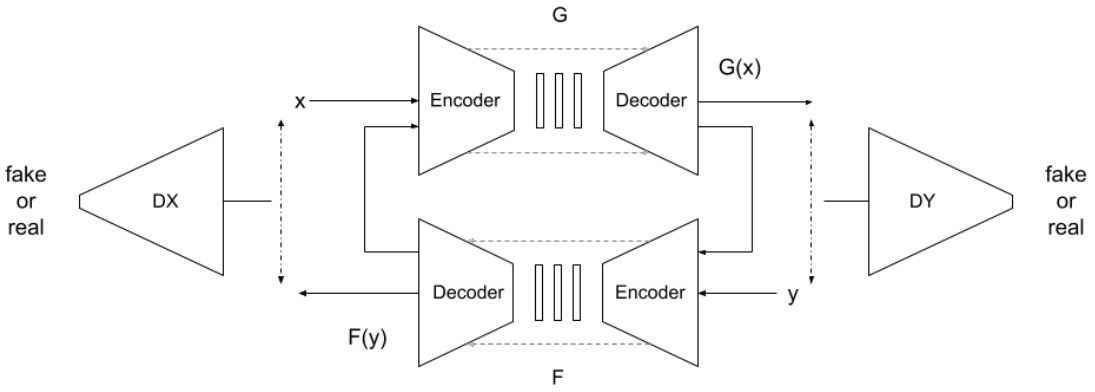

Figure 1.3: CycleGAN structure.

functions

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[logD_Y(y)] + E_{x \sim p_{data}(x)}[log(1 - D_Y(G(x)))] \quad (1.5)$$

$$\mathcal{L}_{GAN}(F, D_X, Y, X) = E_{x \sim p_{data}(x)}[logD_X(x)] + E_{y \sim p_{data}(y)}[log(1 - D_X(F(y)))] \quad (1.6)$$

An additional loss function is also included that takes into account the cycle process whereby $x \rightarrow G(x) \rightarrow F(G(x)) \rightarrow \tilde{x}$ and $y \rightarrow F(y) \rightarrow G(F(y)) \rightarrow \tilde{y}$. Cycle consistency loss is defined as follows

$$\mathcal{L}_{cyc}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] + E_{y \sim p_{data}(y)}[||G(F(y)) - y||_1] \quad (1.7)$$

Given the complete loss function

$$\mathcal{L}(G, F, D_Y, D_X) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \quad (1.8)$$

where λ controls the relevance between the two objectives, the final objective is

$$\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \quad (1.9)$$

Chapter 2

Dataset

The dataset must consist of histological images. The analysis of these is crucial in the diagnosis of many diseases, including almost all cancers. In fact, it leads to important information about individual cell instances. The shape, type, morphology of nuclei, number and density are some of the key information. Nuclei instance segmentation is necessary for the extraction of these features.

The development of DL techniques aimed at nuclei instance segmentation requires fully annotated datasets in order to train the model and evaluate its performance. Biomedical experts consider manual labelling of histological images to be the gold standard method for producing ground truth for nuclei instance segmentation, although they are affected by intra-operator and inter-operator variability.

Nuclei instance segmentation techniques can be distinguished according to the annotations they can lead to:

- detection: identification of cellular instances in terms of coordinates
- semantic segmentation: segmentation that generates binary foreground masks
- instance segmentation: identification of individual cell instances boundary coordinates

They can also be accompanied by the classification of cell instances.

2.1 Inclusion criteria

The search for the benchmark dataset involves the definition of inclusion criteria. The inclusion criteria must meet the thesis intent. The generation of images from ground truths and thus the reversal of the nuclei instance segmentation process requires histological images accompanied by manual individual cell instances annotations. These must be performed by at least three operators in order to start from a ground truth minimally affected by inter-operator variability and which presents a high correspondence with the paired histological images. Since the thesis concerns the development of a DL-based model, a large number of images is required.

In summary, the inclusion criteria are:

- histological images
- large number of images
- manual instance segmentation
- at least three annotations

The following table shows the results of the dataset search in literature. Publicly datasets whose criteria information could be found are presented.

Dataset	# Images	Tile size [pixels]	Magnification	# Organs	# Instances	# Annotators	# Annotations	Annotation type	Source
CoNSep [16]	41	1000x1000	40x	1	24319	2	1	instance + classification	UHCW
CPM-17 [17]	32	500x500, 600x600	40x, 20x	4	7570	n/a	1	instance	TCGA
CRCHisto [18]	100	500x500	20x	1	29756	1	1	detection + classification	UHCW
CryoNuSeg [19]	30	512x512	40x	10	7596	2	3	semantic + instance	TCGA
Janowczyk [20]	143	2000x2000	40x	1	12000	1	1	semantic	n/a
MoNuSAC [21]	209	81x113, 1422x2162	40x	4	31411	n/a	1	instance + classification	TCGA
MoNuSeg [22]	44	1000x1000	40x	9	31411	n/a	1	instance	TCGA

Table 2.1: Publicly available datasets with manual nuclei instance segmentation. CoNSep = Colorectal Nuclear Segmentation and Phenotypes; CPM: Computational Precision Medicine; CRCHisto: Colorectaladeno Carcinomas; MoNuSAC = Multi-Organ Nuclei Segmentation and Classification; MoNuSeg = Multi-Organ Nuclei Segmentation. For the CryoNuSeg dataset # Instances refers to the number of instances that annotator 1 segmented in its first manual mark-up cycle.

According to the search results and inclusion criteria, the CryoNuSeg dataset is selected.

2.2 CryoNuSeg

The development of a DL-based algorithm requires a high level of representativity. TCGA holds over 30000 WSI from over 50 human organs. The CryoNuSeg dataset, whose source is TCGA, includes WSIs from multiple centres, different sexes and a variety of diseases. It includes 30 histological patches, extracted from the most representative parts of the 30 WSIs of 10 human organs, 3 for each organ: adrenal gland, larynx, lymph node, mediastinum, pancreas, pleura, skin, testis, thymus, thyroid gland. The images are followed by three annotations made by two annotators. The same annotator makes two annotations 6 months apart.

The Dataset owes its name to:

- Cryo: Cryosectioned frozen tissues before H&E-staining
- Nu: Nuclei
- Seg: Segmentation

Individual cell instances annotations are provided by means of label masks. They are greyscale 16-bit images. A grey level is assigned to each cell instance. If the cell instances overlap, a grey level equal to the sum of the grey levels of the individual instances is assigned to the overlapping area.

To make the cell instances visible, they are converted in 8-bit format. If the number of cell instances is greater than 255, it starts again from 1. An example of images and their label masks is shown below (Fig. 2.1).

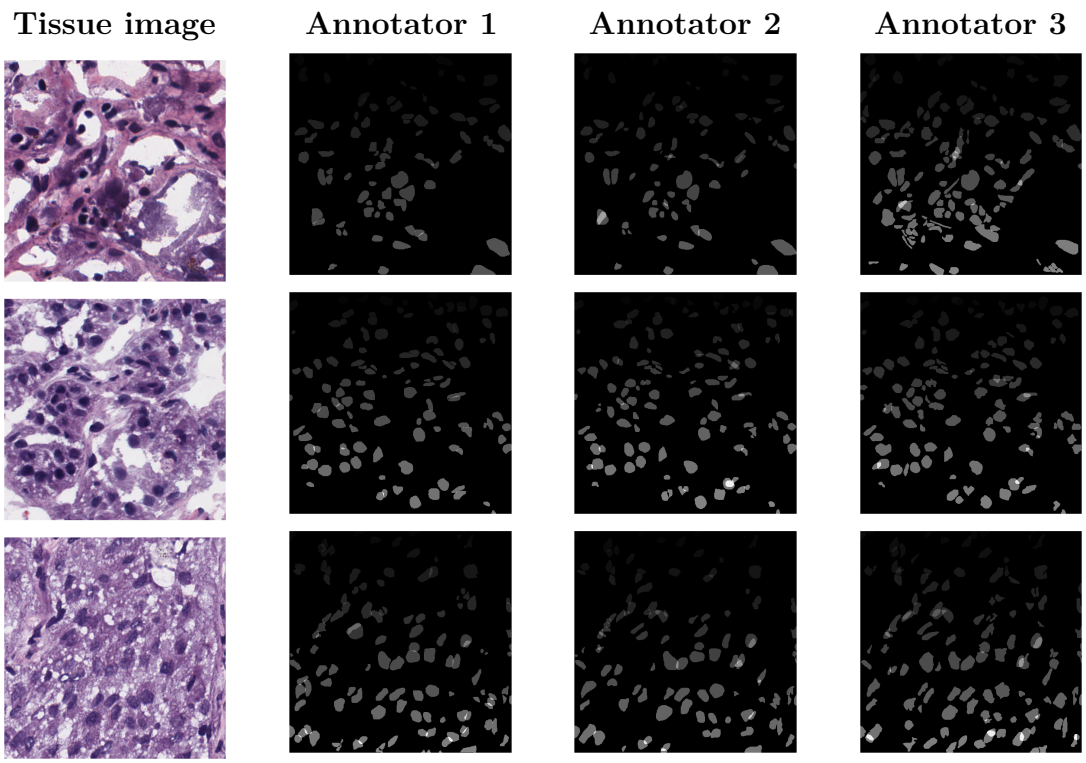

Figure 2.1: Human adrenal gland tissues and annotations.

Chapter 3

Construction of Training Set and Test Set

3.1 Data Pre-Processing

Histological images are affected by different types of variability, related to different factors. The histopathology process involves steps that require manual intervention, which together with device-related artefacts affect the final quality of the slide. All phases can affect the final tissue appearance in the WSI, including surgical removal, transport to the laboratory, fixation, staining, scanning and coverslipping. In a study, substantial accuracy losses dependent on the quality of H&E-staining, brightness and contrast are shown [23]. It was verified that DL-based algorithm performance is not significantly affected by the tissue fixation/inclusion procedure performed [19].

In multicentre studies, there are additional sources of variability. In addition to high-level characteristics, there are first-order characteristics such as brightness and contrast. In histological images, these characteristics are highly dependent on the system, protocol and acquisition environment. The use of different scanners results in a heterogeneous dataset. Adversarial learning shows to be a very efficient technique to learn scanner-independent features [24].

In order to observe how the generative models behave when faced with the aforementioned sources of variability and to understand whether they are able to handle them, the implemented pipeline does not include any staining-normalisation, style transfer or domain adaptation processes.

There are essentially two forms of heterogeneity present in manual annotations: intra-operator variability and inter-operator variability. It can be noticed that inter-operator variability has a higher impact on DL-based algorithm performance than intra-operator variability. [19]. The most important source of inter-operator variability is the annotators' experience. Differences in segmentation of the same cell instance can be found between different annotations. It is possible for a cell instance to be segmented by only one annotator. The differences between annotations are manifold.

For the intent of the Thesis, the different annotations must be merged in order to realise ground truths that are least affected by inter-operator variability and match the histological images as closely as possible. It is necessary that there is the highest correspondence between manual annotation and histological image. The merging process takes place at the individual cell instance level. In this regard, ground truths must present information at individual instance level. The following

describes pre-processing methods aimed at creating semantic ground truths from which realistic histological images can be generated.

3.1.1 Instance division

As they are provided within the CryoNuSeg Dataset, manual annotations must be processed to obtain information on individual cell instances. The masks in gray scale present a grey level for each instance. The crucial point is represented by overlapping cell instances. A grey level equal to the sum of the grey levels of the overlapping cell instances is assigned to the overlapping area. We refer to these masks as label masks.

The cell instance splitting operation is implemented in the first step of the pipeline in the file 'instance_division.py'. In it, the function 'counting_cells' is implemented, which works on a single label mask.

Given a label mask, all distinct objects are identified. An object is defined as either a single cell instance or a collection of cell instances. One object is analysed at a time. If the object has only one grey level, it coincides with a single cell instance. If the object has several grey levels, it consists of several cell instances. In this case, the division is performed on the assumption that the overlapping areas have grey levels obtained by a linear combination of the grey levels of the individual cell instances. The correct division is verified. The difference between the starting object and the sum of the individual cell instances must be a black mask. A negative result of the test corresponds to objects with adjacent cell instances. This situation is not considered by the starting hypothesis. A further cell division then takes place. Finally, the function returns the contour coordinates of the individual

cell instances (Fig. 3.1). The division of the cell instances is realised for all label masks of each annotator.

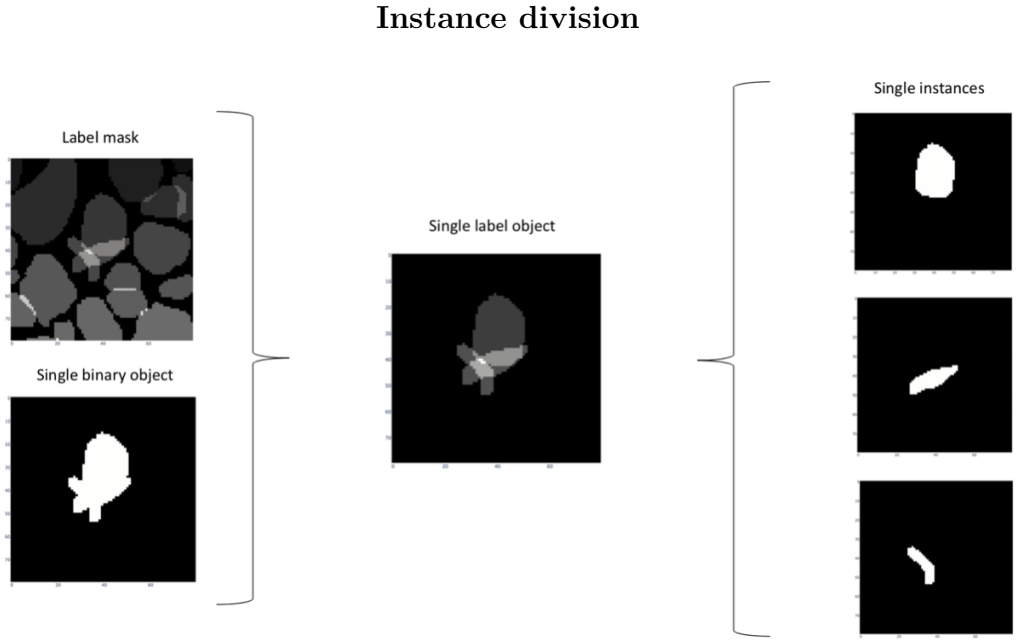

Figure 3.1: Cell instance splitting operation is implemented on a single label object that may consist of several instances.

3.1.2 STAPLE Algorithm

As soon as the coordinates of the individual cell instances are obtained, the different annotations are compared in order to obtain a single ground truth. The comparison and merging of the label masks are implemented in the second step of the pipeline in the file 'label_fusion.py', with the functions 'comparison' and 'run_staple' respectively. The functions are independent of the number of annotators.

Sets of instances of the different annotators are identified to be merged into

a single ground-truth. The comparison of individual cell instances is carried out between two operators at a time by fixing the reference annotator and changing the comparison annotator. Given N annotators, the first annotator is compared with $N-1$ annotators and is then no longer considered; the second with $N-2$ annotators and so on. $N(N-1)/2$ comparisons are carried out. A reference annotator instance is set, which is compared with all instances of the comparison annotator. The contours of the instances are filled in. The reference instance is regarded as true segmentation. The comparison instance is considered as predicted segmentation. If *recall* or *precision* is greater than or equal to 0.7, the instances are selected. The condition may be fulfilled between a reference instance and multiple instances of the comparison annotator.

The following checks are then carried out:

- If several instances of the reference annotator satisfy the condition with the same instances of the comparison annotators, they are placed in the same set of instances.
- Reference instances selected in previous comparisons are not selected if the condition was met with the same comparison instances.

As the corresponding sets of cellular instances between the different annotators are obtained, the STAPLE (Simultaneous Truth and Performance Level Estimation) algorithm is applied [25]. It is an algorithm that considers a set of segmentations of an image and at the same time performs a probabilistic estimation of the true segmentation and a performance measure of each segmentation against the true segmentation in terms of sensitivity and specificity. The algorithm is divided into two steps: expectation and maximisation. Given the segmentation decisions

and a previous estimate of the performance level of each segmentation generator, conditional probability density of the true hidden segmentation is obtained in the first step. In the absence of information on the performance level of each segmentation generator or true segmentation, the article recommends initialising sensitivity and specificity parameters to an identical value, the same for all evaluators. It is recommended to start with values very close to 1. A value of 0.99999 is set. Based on these parameters, the conditional probability density is estimated. The second step consists of a maximisation problem for the conditional probability density function. A convergence threshold of 0.01 and a maximum number of iterations of 1000 are set. As an output, the algorithm provides an image in which each pixel is assigned a covering weight between 0 and 1, representing the frequency of that pixel in the true segmentation. Given the small number of pixels in the cell instances, a threshold of 0.05 is set on the frequency to obtain cell instances with realistic contours. The algorithm is implemented in the file 'staple.py' and used in the function 'run_staple' at the level of the corresponding sets of instances between the different annotators. As an output, the function provides the contours of the created instances.

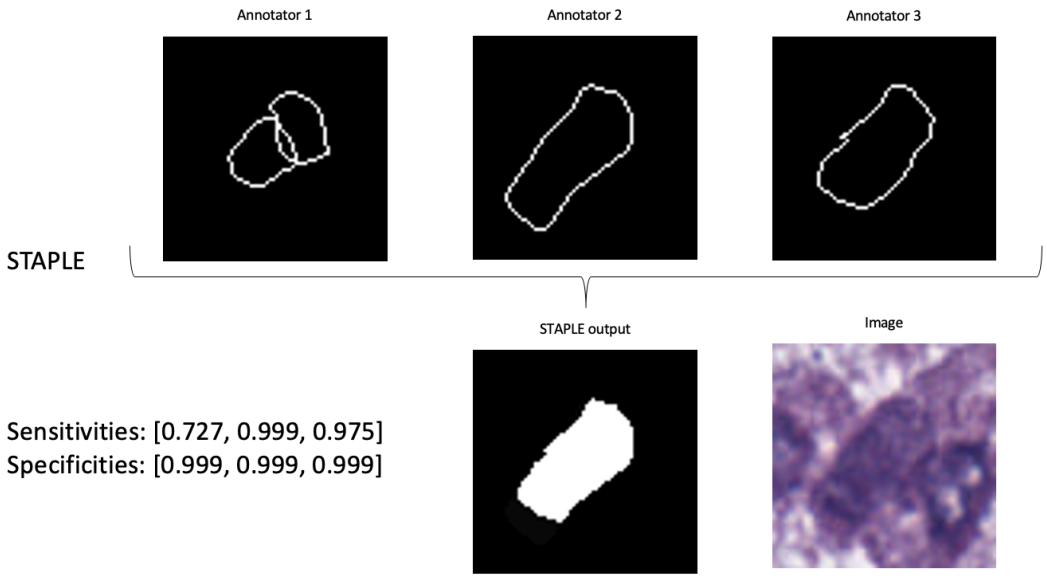

Figure 3.2: The sets of cell instances segmented by different annotators are merged together into a single instance, achieving the sensitivities and specificities shown in the figure.

3.1.3 Instance cleaning

Then checks are performed on the generated cell instances. They are implemented in the 'cleaning.py' file. It is unlikely that several instances are completely or almost completely overlapped. For each ground truth, objects consisting of overlapping instances are considered. For each object, the instances are sorted according to area in descending order and it is checked whether:

- given a reference instance and a comparison instance, the two instances are completely or almost completely overlapping. If they have $DSC \geq 0.9$, the instance with the smaller area is deleted (Fig. 3.3).

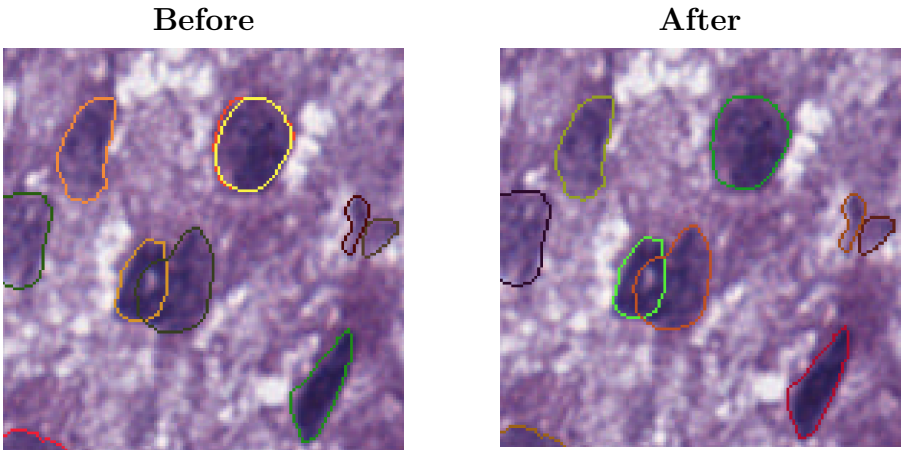

Figure 3.3: Instance boundaries before and after the first check.

- there are multiple overlapping instances of the same instance. Given a reference instance as true segmentation, it is compared with the instances of smaller area by calculating the recall. Considering the instances with $recall \geq 0.8$ as a single object, its DSC is calculated with the reference instance. If $DSC \geq 0.8$, cell division is preferred and the reference instance is deleted (Fig. 3.4).

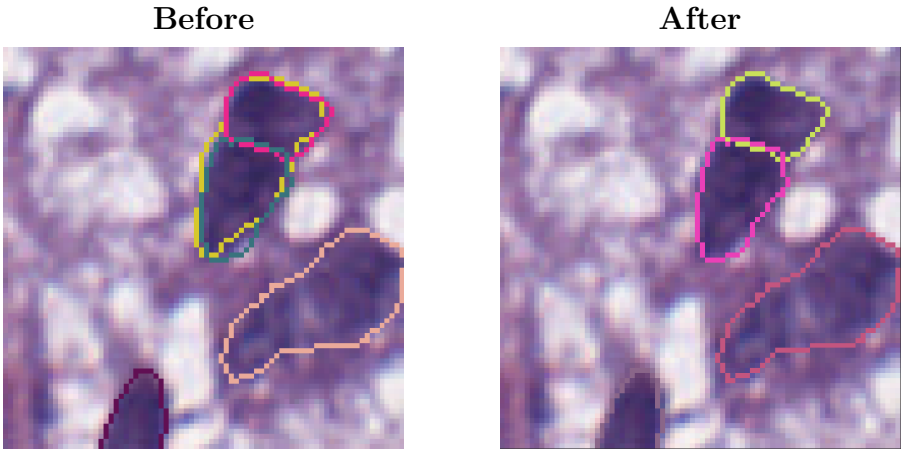

Figure 3.4: Instance boundaries before and after the second check.

- given a reference instance as true segmentation and a comparison instance as predicted segmentation, $recall \geq 0.8$ occurs. In this case, the comparison instance is deleted since it overlaps another instance by at least 80% of its area.

Organ	# Images	Ann 1	Ann 2	Ann3	Final GT
Adrenal gland	3	338	344	392	321
Larynx	3	622	641	765	639
Lymph node	3	1204	1308	1353	1214
Mediastinum	3	1274	1349	1354	1255
Pancreas	3	470	548	623	486
Pleura	3	497	515	638	462
Skin	3	457	436	507	422
Testis	3	752	793	687	691
Thymus	3	1521	1646	1483	1470
Thyroid gland	3	461	464	449	428
Total	30	7596	8044	8251	7388

Table 3.1: Number of cell instances per organ, among different annotators and following application of the algorithm to create a single ground truth.

3.2 White detection Algorithm

To give the semantic masks more information content by means of an additional class label, the white zones of the histological images are added. White zones

correspond to portions of tissue on which H&E staining has not been fixed. The detection of white areas is implemented in the file 'white_detection.py'. The function 'illuminant_estimation' identifies the triplet of whites corresponding to the RGB channels of the histological images. The average of this triplet is used to balance the whites. The balancing is implemented in the 'illuminant_correction' function. The images are then thresholded in order to identify white areas. To obtain white masks with smoother contours, small objects are removed and binary closing is implemented with a three-pixel disc as a structural element (Fig. 3.5.

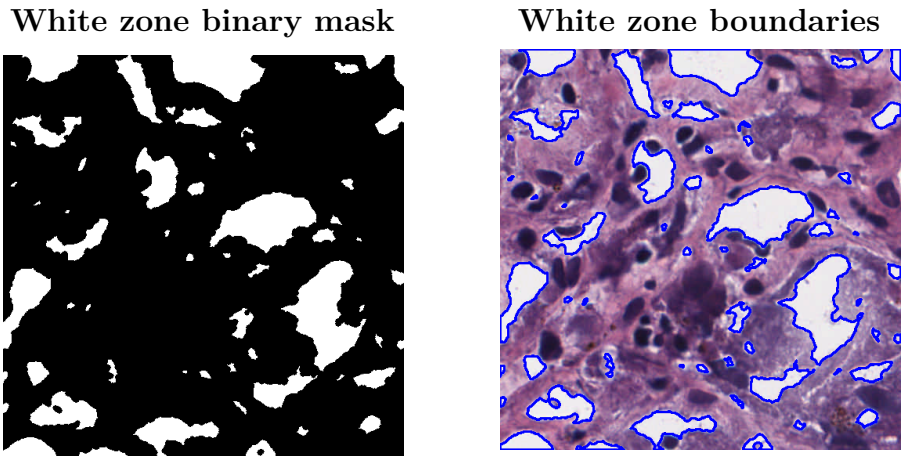

Figure 3.5: White boundaries detection.

3.3 Training Set and Test Set

For the construction of Training Set and Test Set, the contours of the individual cell instances are joined together with the white zones in a single ground truth. Each of these represents a class label which is assigned a different grey level on an 8-bit scale: 255 to the contours of the instances, 127 to the white areas and 0 to the background. In order to understand how the contour information of the individual

instances is captured by the generative models, two different ground truths are created: one with 2-pixel thick contours and one with 3-pixel thick contours.

In addition, to increase the numerosity of the dataset, the images of size 512x512 pixels are divided into patches of 256x256 pixels with a 50% overlap. The number of images increases from 30 to 270.

The Training Set consists of the images of 9 organs: adrenal gland, larynx, lymph node, mediastinum, pancreas, pleura, skin, testis, thymus (243 images). The Test Set consists of the images of the remaining organ: thyroid gland (27 images). This makes the Test Set independent of the Training Set.

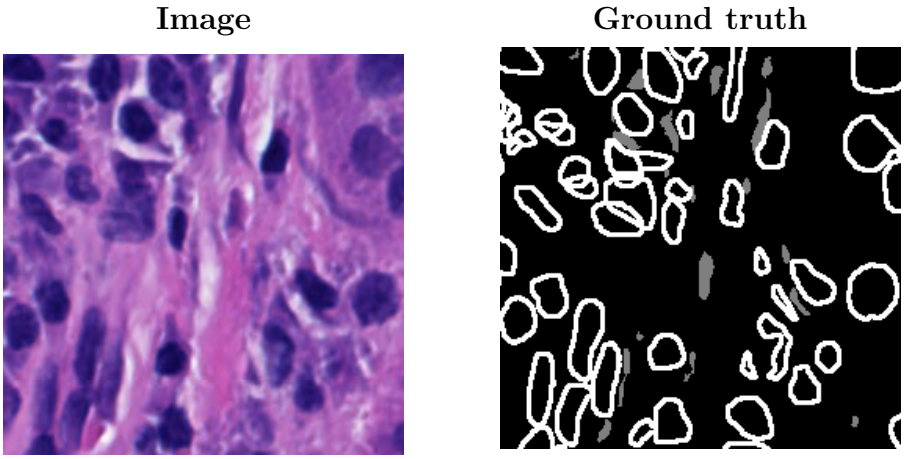

Figure 3.6: Histological image and relative ground truth.

Chapter 4

Methods

4.1 Architectures

The following GAN architectures are used for image generation: Pix2Pix [14] and cycleGAN [15]. All networks are built with PyTorch.

4.1.1 Pix2Pix

The Pix2Pix network consists of a U-Net [26], with 8 downsapmlings, as generator and a PatchGAN classifier as discriminator described in the original Pix2Pix article [14] with a number of convolutional layers set to 3. In both components, the number of filters in the first and the last convolutional layer can be controlled via the parameters ngf (number of generator filters) and ndg (number of discriminator filters).

Pix2Pix Generator U-Net
Conv2D Layer, 4x4, stride 2, pad 1, instance norm (In) input channels x image height x image width $\Longrightarrow$ ngf x 128 x 128
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 ngf x 128 x 128 $\Longrightarrow$ 2ngf x 64 x 64
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 2ngf x 64 x 64 $\Longrightarrow$ 4ngf x 32 x 32
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 4ngf x 32 x 32 $\Longrightarrow$ 8ngf x 16 x 16
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 8ngf x 16 x 16 $\Longrightarrow$ 8ngf x 8 x 8
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 8ngf x 8 x 8 $\Longrightarrow$ 8ngf x 4 x 4
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 8ngf x 4 x 4 $\Longrightarrow$ 8ngf x 2 x 2
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 8ngf x 2 x 2 $\Longrightarrow$ 8ngf x 1 x 1
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 1 x 1 $\Longrightarrow$ 8ngf x 2 x 2
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 2 x 2 $\Longrightarrow$ 8ngf x 4 x 4
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 4 x 4 $\Longrightarrow$ 8ngf x 8 x 8
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 8 x 8 $\Longrightarrow$ 8ngf x 16 x 16
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 16 x 16 $\Longrightarrow$ 4ngf x 32 x 32
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 4ngf x 32 x 32 $\Longrightarrow$ 2ngf x 64 x 64
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 2ngf x 64 x 64 $\Longrightarrow$ ngf x 128 x 128
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU ngf x 128 x 128 $\Longrightarrow$ output channels x image height x image width
Tanh

Table 4.1: Generator U-Net Architecture details of Pix2Pix model.

Pix2Pix Discriminator PatchGAN
Conv2D Layer, 4x4, stride 2, pad 1, leakyReLU 0.2 input channels x image height x image width $\Longrightarrow$ ndf x 128 x 128
Conv2D Layer, 4x4, stride 2, pad 1, instance norm (In), leakyReLU 0.2 ndf x 128 x 128 $\Longrightarrow$ 2ndf x 64 x 64
Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 2ndf x 64 x 64 $\Longrightarrow$ 4ndf x 32 x 32
Conv2D Layer, 4x4, stride 1, pad 1, In, leakyReLU 0.2 4ndf x 32 x 32 $\Longrightarrow$ 8ndf x 31 x 31
Conv2D Layer, 4x4, stride 1, pad 1 8ndf x 31 x 31 $\Longrightarrow$ 1 x 30 x 30

Table 4.2: Discriminator PatchGAN Architecture details of Pix2Pix model.

4.1.2 CycleGAN

The cycleGAN network has a Resnet, inspired by PathologyGAN [27], with 9 residual blocks at the bottleneck, as generator and a PatchGAN classifier, with Resnet layers, as discriminator [14, 27], with a number of convolutional layers set to 3. In both components, the number of filters in the first and the last convolutional layer can be controlled via the parameters ngf and ndg.

CycleGAN Generator Resnet
Resnet layer; Conv2D Layer, 3x3, stride 1, pad 1, instance norm (In), leakyReLU 0.2 input channels x image height x image width $\Longrightarrow$ ngf x 256 x 256
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 ngf x 256 x 256 $\Longrightarrow$ 2ngf x 128 x 128
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 2ngf x 128 x 128 $\Longrightarrow$ 4ngf x 64 x 64
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 4ngf x 64 x 64 $\Longrightarrow$ 8ngf x 32 x 32
Bottleneck: 9 Resnet layers
ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 8ngf x 32 x 32 $\Longrightarrow$ 4ngf x 64 x 64
Resnet layer; ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 4ngf x 64 x 64 $\Longrightarrow$ 2ngf x 128 x 128
Resnet layer; ConvTranspose2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 2ngf x 128 x 128 $\Longrightarrow$ ngf x 256 x 256
ConvTranspose2D Layer, 3x3, stride 1, pad 1 ngf x 256 x 256 $\Longrightarrow$ output channels x image height x image width
Tanh

Table 4.3: Generator Resnet Architecture details of cycleGAN model.

CycleGAN Discriminator PatchGAN
Resnet layer; Conv2D Layer, 3x3, stride 1, pad 1, instance norm (In), leakyReLU 0.2 input channels x image height x image width $\Longrightarrow$ ndf x 256 x 256
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 ndf x 256 x 256 $\Longrightarrow$ 2ndf x 128 x 128
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 2ndf x 128 x 128 $\Longrightarrow$ 4ndf x 64 x 64
Resnet layer; Conv2D Layer, 4x4, stride 2, pad 1, In, leakyReLU 0.2 4ndf x 64 x 64 $\Longrightarrow$ 8ndf x 32 x 32
Conv2D Layer, 3x3, stride 1, pad 0 8ndf x 32 x 32 $\Longrightarrow$ 1 x 30 x 30
Sigmoid

Table 4.4: Discriminator PatchGAN Architecture details of CycleGAN model.

4.2 Experiments

Taking the two architectures mentioned above as a reference, several models are trained, fixing and modifying certain characteristics appropriately. In all trained models, for both the generator and the discriminator, the Adam optimiser was used with $beta_1 = 0.5$ and $\beta_2 = 0.999$ and a learning rate of 0.0001. All models are trained for 200 epochs and are saved every 10 epochs in order to test them in the inference phase.

The experiments are aimed at analysing the performance of the models as the architecture, Pix2Pix or cycleGAN, the training dataset, 3 pixel or 2 pixel cell instance boundaries, the number of generator and discriminator filters and the loss function vary.

We first compare the two different architectures, using as a training dataset the one in which the instance boundaries have a thickness of 3 pixels. For both architectures, we used an LSGAN [28] and for the generator a number of filters equal to 32 and for the discriminator a number of filters equal to 128. For Pix2Pix we used as GAN loss the L1 distance (1.1) and $\lambda_{L1} = 100$; for cycleGAN as consistency loss (1.7) the L2 distance and $\lambda_{cyc} = 5$.

Furthermore, in order to have a consistent comparison between the two architectures, Pix2Pix is trained not only as a network for generating histological images, taking ground truth as source data and histological images as target data, but also as a segmentation network, taking histological images as source data and ground truth as target data.

The following experiment involves training the Pix2Pix architecture, with the

same characteristics as the previous experiment, with the dataset in which the cell instance boundaries are 2 pixels thick. This network is always trained in both directions.

Finally, the last experiment involves training different Pix2Pix networks by changing the number of generator and discriminator filters and the adversarial loss function (Table 4.5).

Dataset	Loss function	# Generator filters	# Discriminator filters
2 pixels cell instance boundaries	LSGAN [28]	9	9
		24	24
		32	32
		9	40
		24	96
		32	128
		64	16
	Vanilla [14]	9	9
		24	24
		32	32
		9	40
		24	96
		32	128
		64	16

Table 4.5: The table shows the characteristics of the Pix2Pix models trained in the last experiment, using the dataset in which instance boundary lines are 2 pixels thick. Two loss functions are used: LSGAN (least-square GAN) which uses the mean-squared error as loss function and vanilla which implements cross-entropy objective used in the original GAN paper. For both, models with the same number of filters per generator and discriminator are trained.

4.3 Evaluation metrics

4.3.1 Evaluation metrics mask to image

Because of the possibility to have a comparison image to use for assessing generated images quality, Image Quality Assessment (IQA) techniques can be divided into No-Reference (NR), Full-Reference (FR) and Distribution-Based (DB). Since the training uses paired data, NR assessment techniques are excluded.

Among the FR techniques, we will use Pearson Correlation Coefficient (PCC) to assess the linear correlation between the generated and real images. Given the real image X and the generated image Y

$$\mathrm{PCC}_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

where *cov* is the covariance of X and Y, σ_X is the standard deviation of X and σ_Y is the standard deviation of Y. Peak Signal-to-Noise Ratio (PSNR) is used as a distance measure between real and generated images.

$$\mathrm{PSNR} = 20 \cdot \log_{10}\left(\frac{\mathrm{MAX_I}}{\sqrt{\mathrm{MSE}}}\right) \tag{4.2}$$

where MSE is the mean squared error and, for 8-bit images $\mathrm{MAX_I} = 255$.

The aforementioned techniques are based on the pixel intensity of the images. In recent years, IQA techniques have been introduced to measure quantities closely related to the Human Vision System (HVS). These metrics emphasise the importance of HVS sensitivity to different visual signals, such as luminance, contrast, frequency content and the interaction amongst different signal components. They

include Structural Similarity Index Measure (SSIM) [29]. The great success of SSIM is due to the fact that HVS adapts to the structural information of images. SSIM measures three quantities that influence a person's perception of the quality of an image. The SSIM formula is based on three comparison measurements between the real image X and the generated image Y: luminance (l), contrast (c) and structure (s). The individual comparison functions are:

$$l(X,Y) = \frac{2\mu_X\mu_Y + c_1}{\mu_X^2 + \mu_Y^2 + c_1} \tag{4.3}$$

$$c(X,Y) = \frac{2\sigma_X\sigma_Y + c_2}{\sigma_X^2 + \sigma_Y^2 + c_2} \tag{4.4}$$

$$s(X,Y) = \frac{\sigma_{XY} + c_3}{\sigma_X\sigma_Y + c_3} \tag{4.5}$$

with μ_X the pixel sample mean of X, μ_Y the pixel sample mean of Y,σ_X^2 the variance of X, σ_Y^2 the variance of Y, σ_{XY} the covariance of X and Y, $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $c_3 = c_2/2$, L the dynamic range of the pixel-values, $k_1 = 0.01$ and $k_2 = 0.03$ by default. The final index is

$$\text{SSIM}(X,Y) = l(X,Y)^\alpha \cdot c(X,Y)^\beta \cdot s(X,Y)^\gamma \tag{4.6}$$

$$\text{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{4.7}$$

In addition to structural information, the HVS also includes an image based on its low-level features, such as edges and zero crossings. Feature Similarity Index Measure (FSIM) [30] is used to evaluate these features. The previously mentioned metrics perform measurements on the individual RGB channels of the coloured images and then average out the different scores. FSIM provides a variation for

coloured images, called FSIMc, which evaluates features based on the luminance channel Y and the chrominance channels, Q and I, of an image.

The luminance channel is used to extract phase congruency (PC) and gradient magnitude (GM) features for both the real and the generated image. PC is evaluated, under the assumption that visually discernible features correspond to points where Fourier waves at different frequencies have congruent phases. PC is used as the primary feature. To take contrast into account, GM is used as the secondary feature. Similarity maps are created for each of the two features. The similarity maps of the two images are used to extract similarity scores. These are integrated together with the similarity scores calculated through the chrominance channels to obtain the FSIMc score.

A widely used DB technique was employed to evaluate the performance of GANs: Fréchet Inception Distance (FID) [31, 32]. It quantifies the ability of GANs to reproduce the characteristics of the real data distribution. The distance between the real distribution and the distribution generated through Fréchet Distance is measured, not in pixel space, but in a space of HVS-relevant features extracted through a pre-trained ImageNet Inception Network. It is defined by article as follows

$$d_F((\mu_X, \Sigma_X), (\mu_Y, \Sigma_Y))^2 = \|\mu_X - \mu_Y\|_2^2 + \mathrm{tr}\left(\Sigma_X + \Sigma_Y - 2\left(\Sigma_X \cdot \Sigma_Y\right)^{\frac{1}{2}}\right) \quad (4.8)$$

with μ_X, μ_Y, Σ_X, Σ_Y, means and covariance matrices of real and generated distribution.

4.3.2 Evaluation metrics image to mask

Ground truths have 3 classes: cell instances, white zones and background. To evaluate the segmentation generated by the GANs, we used Dice Similarity Coefficient (DSC). Defining X as the actual segmentation and Y as the generated segmentation

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{4.9}$$

Chapter 5

Results

The results of the different trained models are shown according to the generated images that are histological images or segmentation masks.

5.1 Mask to image results

The loss values of the Pix2Pix network trained on the 3 pixels dataset (Fig. 5.1) show controlled improvement during the training phase. Losses calculated on the discriminator output rightly tend to zero values. Loss calculated between the original image and the image is the one that best exhibits the performance improvement during training. For cycleGAN (Fig. 5.2), the relativistic loss gradient was implemented [33]. The latter leads to unit loss for real images and zero loss for generated images. An improvement in performance can be seen through cycle consistency loss. Since the latter reaches zero values in a few steps and remains zero during training indicates that the cycleGAN network is training optimally in

both directions of generation.

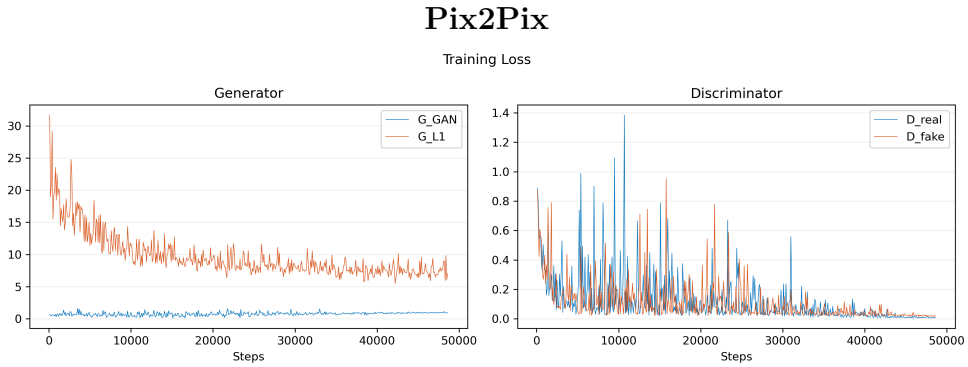

Figure 5.1: Loss values of generator and discriminator of Pix2Pix network during training, accomplished on the 3 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original image and the generated image, the discriminator loss for the real image, the discriminator loss for the generated image, respectively.

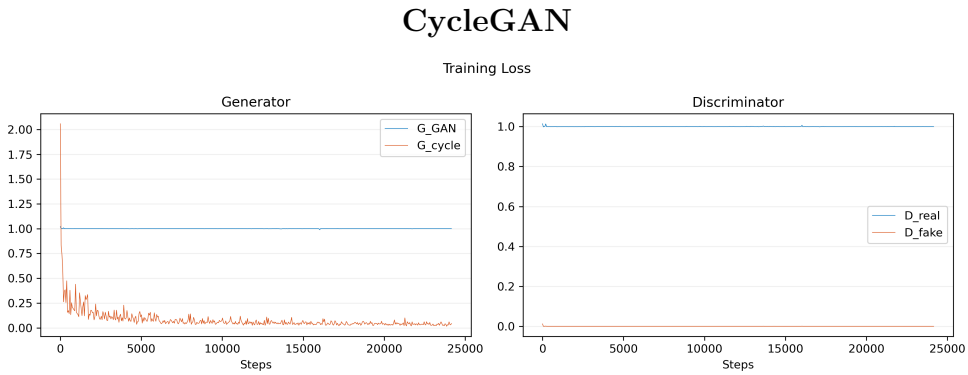

Figure 5.2: Loss values of generator and discriminator, aimed at creating histological images, of the cycleGAN network during training, accomplished on the 3 pixels dataset. G_GAN, G_cycle, D_real, D_fake indicate the generator loss, cycle consistency loss between the real image and the generated image, discriminator loss for the real image, discriminator loss for the generated image, respectively.

The comparison between Pix2Pix and cycleGAN shows, in terms of PCC (Fig. 5.4), SSIM (Fig. 5.6) and FSIM (Fig. 5.7), comparable results. PSNR (Fig. 5.5) and FID (Fig. 5.8) prove to be the most informative metrics as both show substantial differences between the two architectures. For Pix2Pix, as opposed to cycleGAN, these metrics show less data dispersion, by epoch, and a more uniform trend going to a plateau. This difference can be seen directly from the generated images. Although visually both architectures succeed in exhibiting the semantic content of the ground truths from which the histological images are generated, cycleGAN generates images with obvious artifacts, in larger quantities than Pix2Pix. The images contain unrealistic patterns, white areas and black areas, which the network fails to define properly or at least realistically the white zones and the instances from the ground truth (Fig. 5.3).

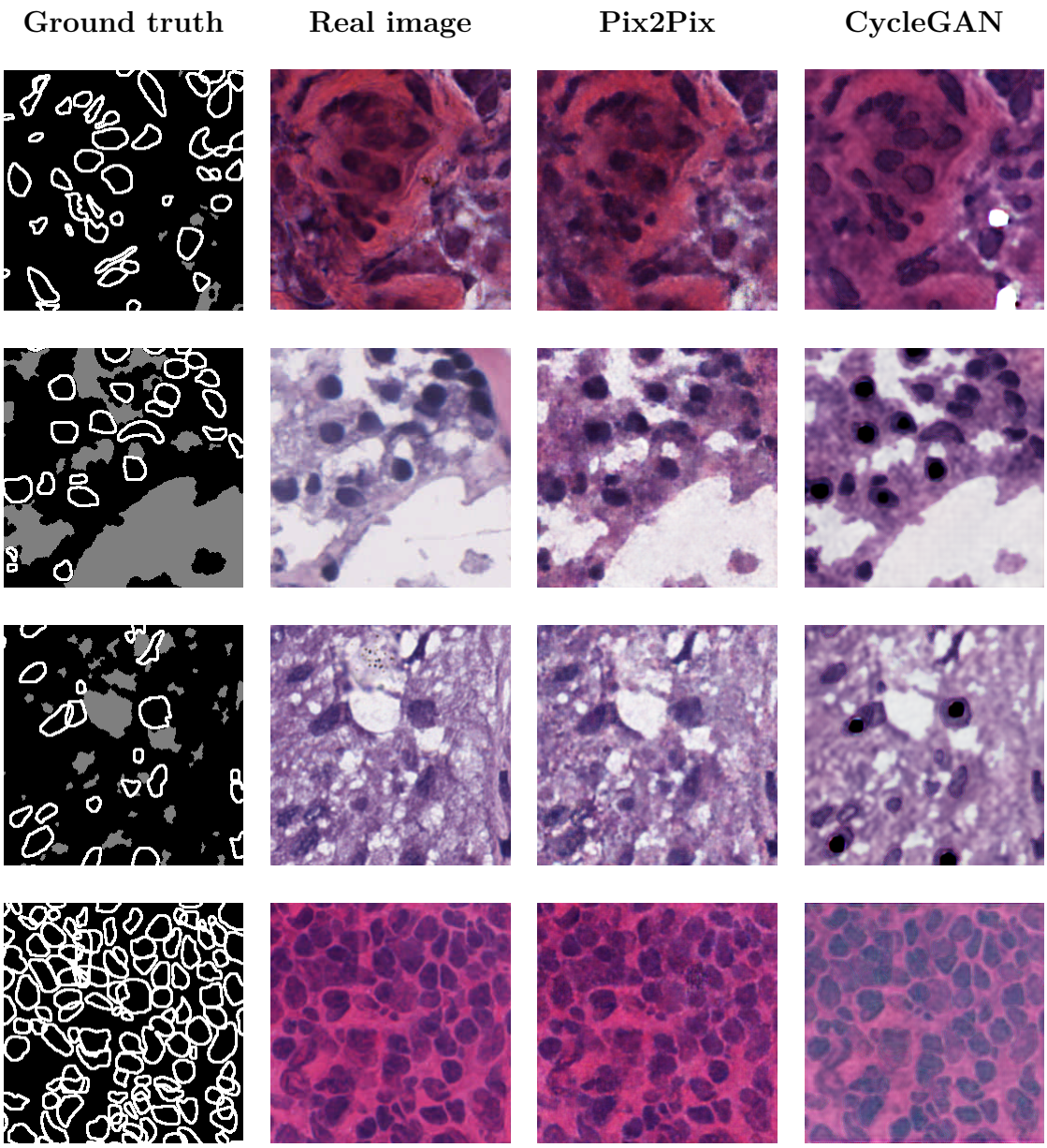

Figure 5.3: Comparison between real images and Pix2Pix and cycleGAN generated images at 200th epoch.

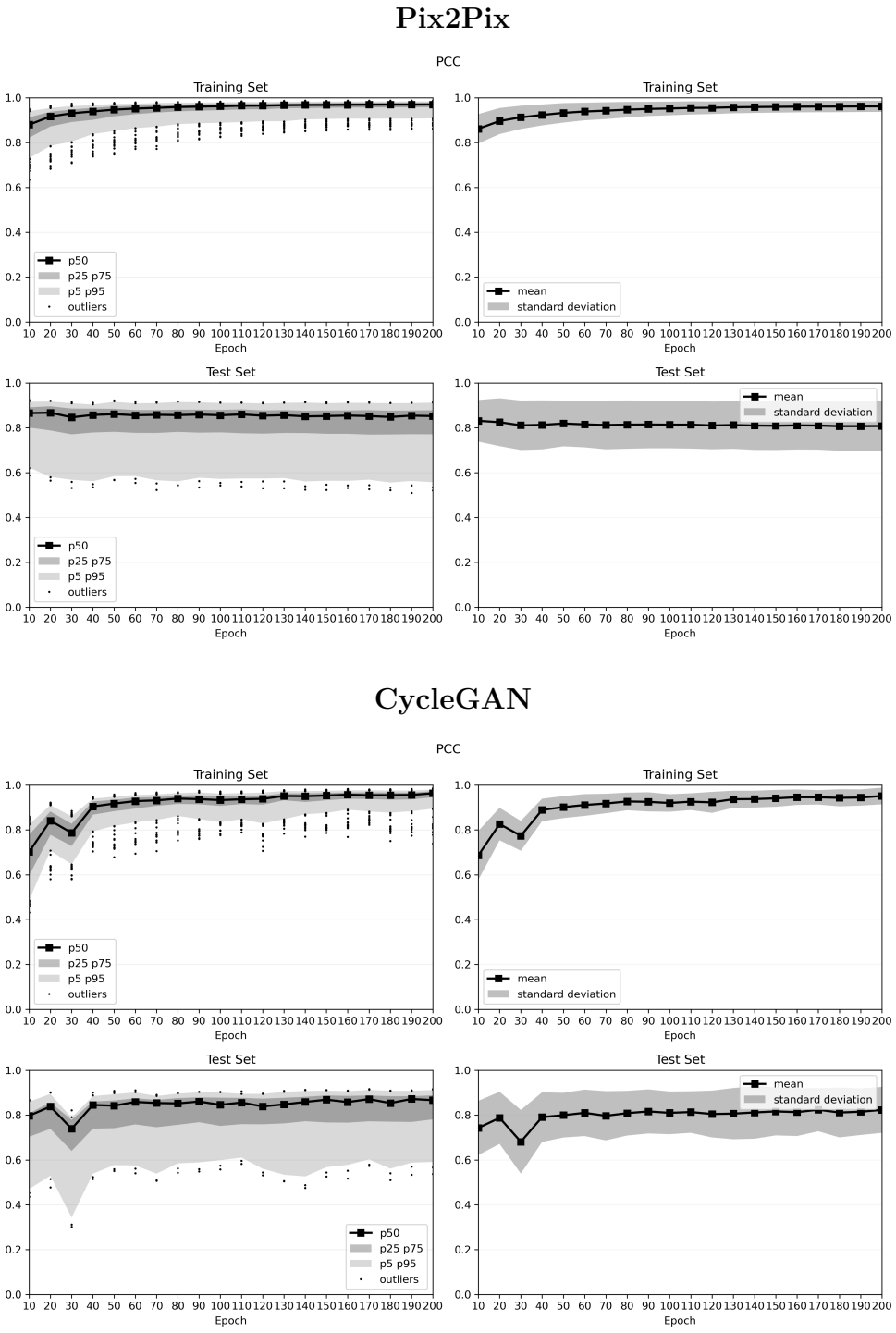

Figure 5.4: PCC measurements by epoch of Pix2Pix and cycleGAN networks trained with the 3-pixels dataset.

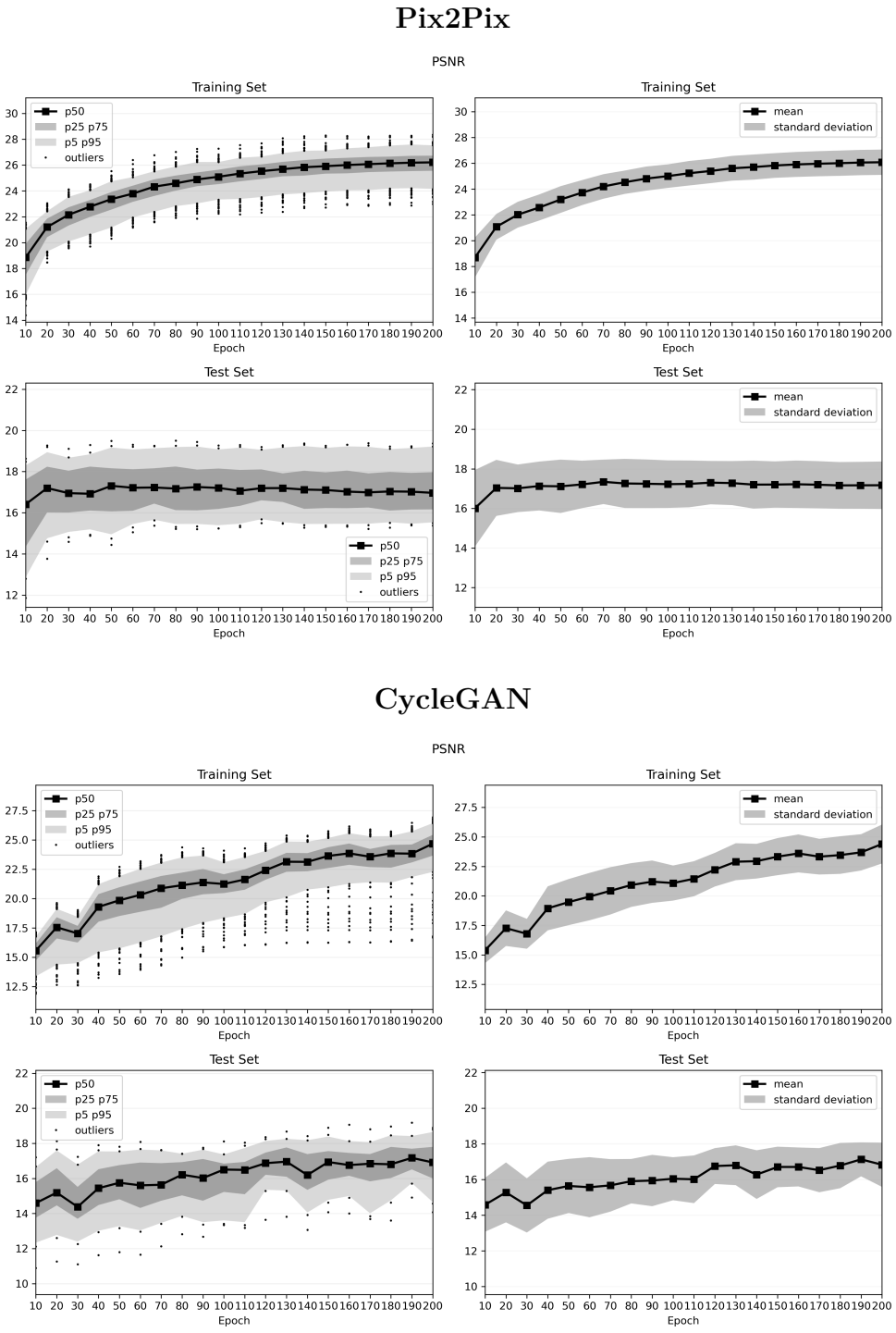

Figure 5.5: PSNR measurements by epoch of Pix2Pix and cycleGAN networks trained with the 3-pixels dataset.

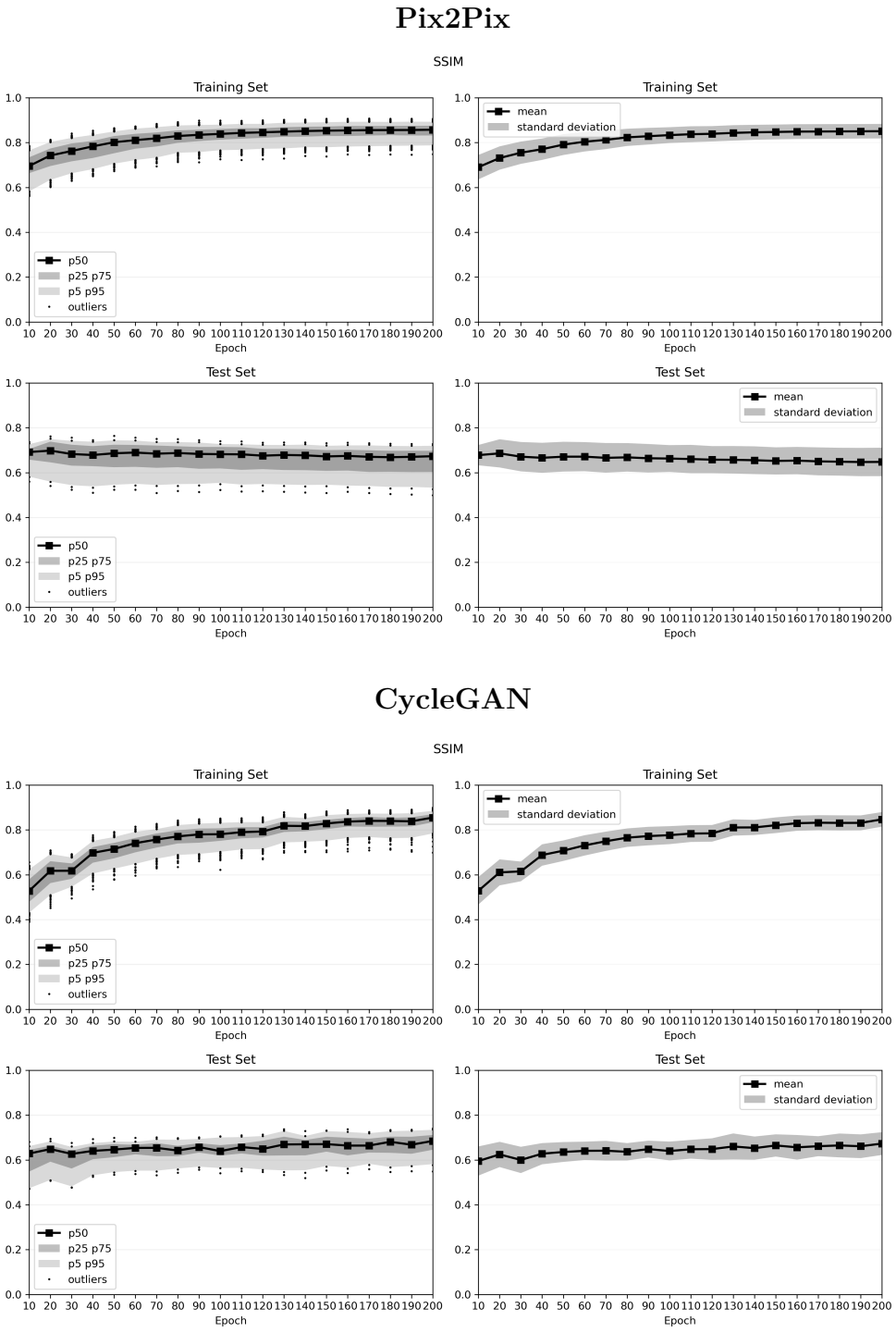

Figure 5.6: SSIM measurements by epoch of Pix2Pix and cycleGAN networks trained with the 3-pixels dataset.

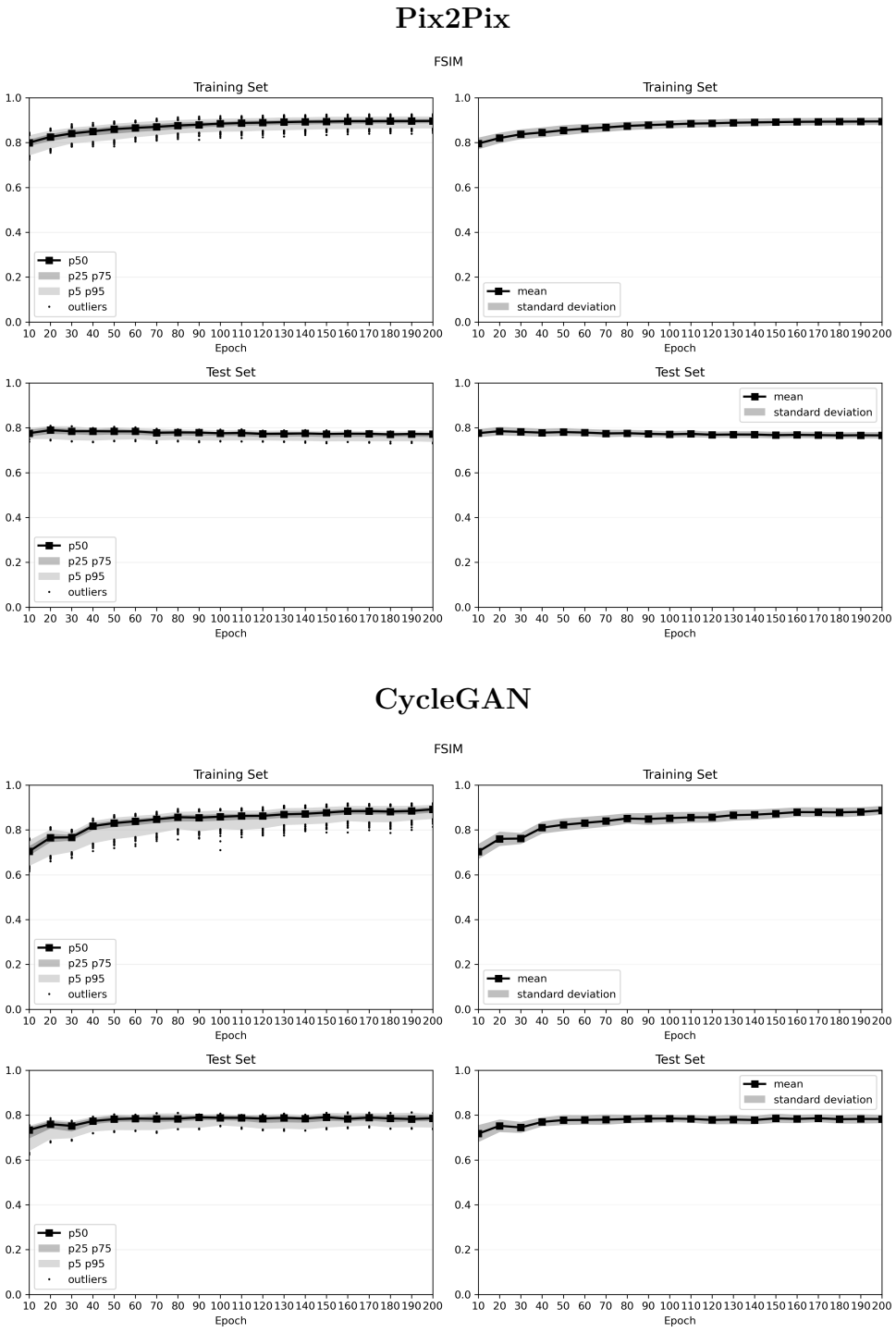

Figure 5.7: FSIM measurements by epoch of Pix2Pix and cycleGAN networks trained with the 3-pixels dataset.

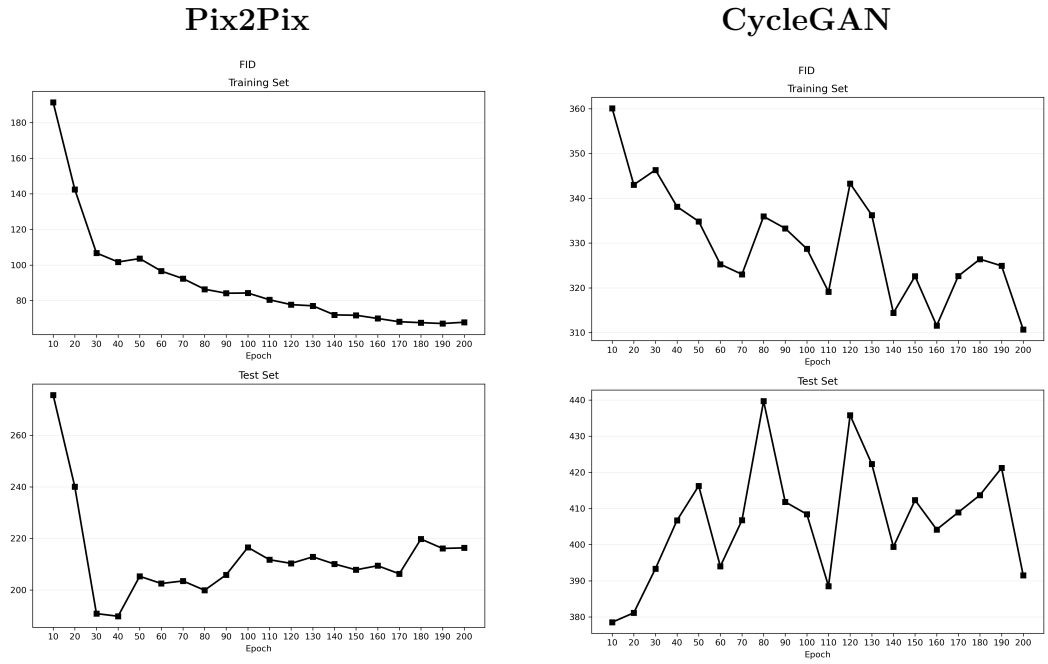

Figure 5.8: FID measurements by epoch of Pix2Pix and cycleGAN networks trained with the 3-pixels dataset.

From the Training Set to the Test Set there is an increase in data dispersion by epoch. Although the Test Set images represent the semantic content and present, in terms of PCC, SSIM and FSIM, acceptable results, overfitting on the Training Set is evident. For the Pix2Pix, it is clearly visible from the 50th epoch through the FID (Fig. 5.8). Visually, the generated images express the semantic content contained in the ground truths. Although Pix2Pix goes into overfitting, it can be seen from the images and metrics (Table. 5.2), as far as the Test Set is concerned, that the quality of the images generated by cycleGAN in the last epochs (Fig. 5.10) is achieved by Pix2Pix in the first epochs (Fig. 5.9), presenting also fewer artifacts.

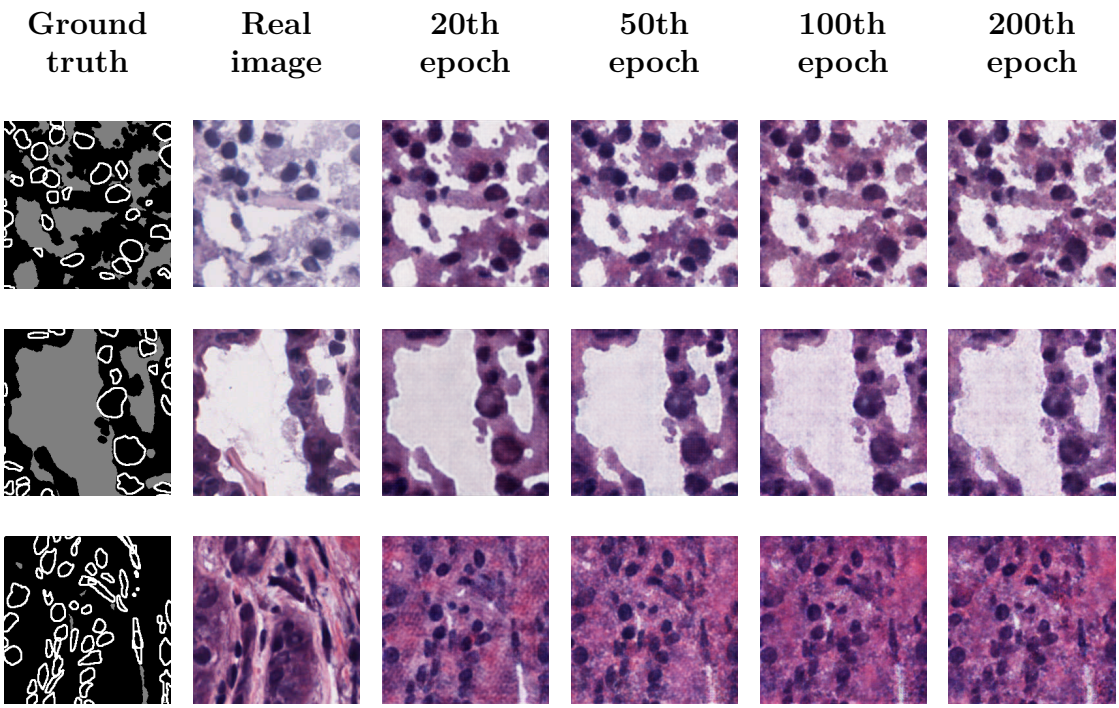

Figure 5.9: Test Set Pix2Pix generated images among epochs.

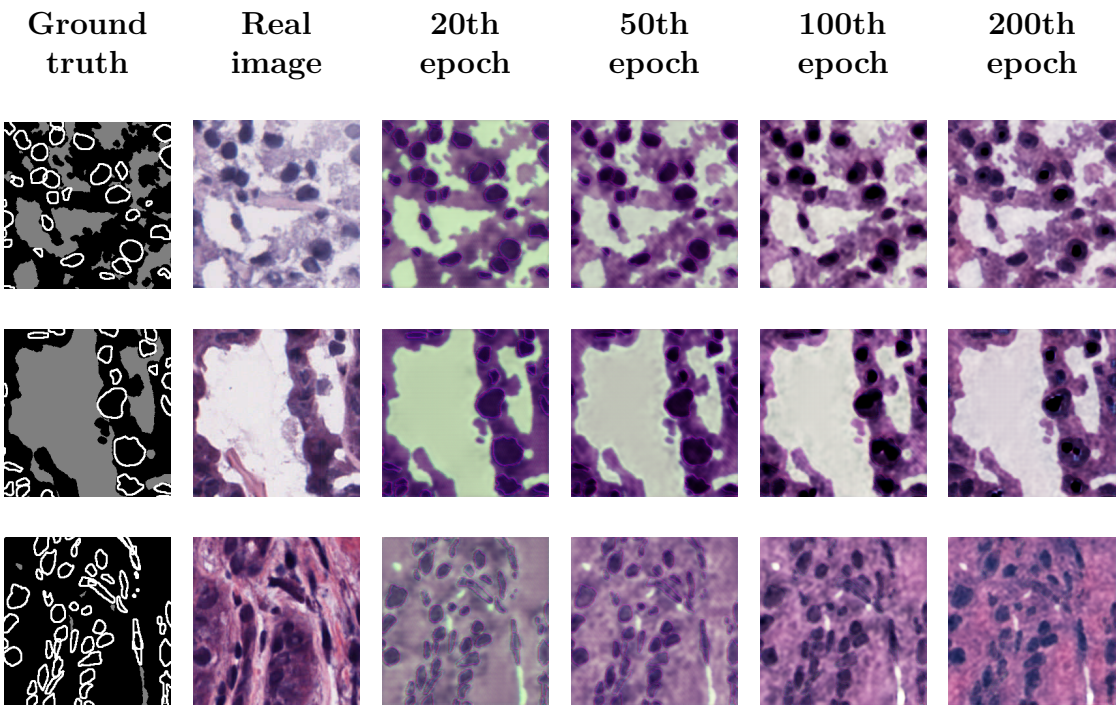

Figure 5.10: Test Set cycleGAN generated images among epochs.

Training Set

Dataset	GAN	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
3 pixels cell instance boundaries	Pix2Pix	20	89.61 ± 5.8	21.08 ± 0.99	73.11 ± 5.2	82.03 ± 2.34	142.47
		50	93.25 ± 4.34	23.19 ± 1.03	79.08 ± 4.63	85.48 ± 2.27	103.65
		100	95.19 ± 3.03	25 ± 0.91	83.33 ± 3.53	88.11 ± 1.77	84.29
		200	96.15 ± 2.48	26.08 ± 0.98	85.08 ± 3.23	89.44 ± 1.61	67.86
	CycleGAN	20	82.6 ± 7.25	17.26 ± 1.51	61.06 ± 5.79	76.02 ± 3.23	343.02
		50	90.16 ± 4.89	19.47 ± 1.95	70.73 ± 4.57	82.28 ± 2.78	334.84
		100	92 ± 3.9	21.09 ± 1.49	77.65 ± 4.01	85.26 ± 2.63	328.72
		200	95.13 ± 3.71	24.4 ± 1.65	84.69 ± 3.26	88.71 ± 1.96	310.71

Table 5.1: Except for FID which is calculated on the whole Training Set, metrics are expressed as mean ± standard deviation. In the Training Set, by comparing the two architectures, Pix2Pix and cycleGAN, it appears that in terms of PCC, SSIM and FSIM the results are comparable. The PSNR values of cycleGAN at the 200th epoch are reached by Pix2Pix at the 50th epoch. In addition, the FID indicates a better quality of the images generated by the Pix2Pix.

Test Set

Dataset	GAN	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
3 pixels cell instance boundaries	Pix2Pix	20	82.47 ± 10.66	17.03 ± 1.41	68.58 ± 6.28	78.48 ± 1.9	240.05
		50	81.9 ± 10.17	17.11 ± 1.35	67.11 ± 6.64	78.08 ± 1.76	205.3
		100	81.38 ± 10.52	17.22 ± 1.2	66.28 ± 5.98	77.08 ± 1.39	216.52
		200	80.8 ± 10.98	17.17 ± 1.2	64.77 ± 6.33	76.61 ± 1.46	216.33
	CycleGAN	20	78.77 ± 11.61	15.28 ± 1.68	62.45 ± 5.62	75.21 ± 2.81	381.13
		50	79.96 ± 9.95	15.64 ± 1.52	63.55 ± 4.43	77.78 ± 2.24	416.26
		100	81 ± 9.52	16.04 ± 1.21	63.99 ± 4.2	78.46 ± 1.56	408.41
		200	82.31 ± 10.23	16.83 ± 1.24	67.3 ± 5.07	78.25 ± 1.88	391.49

Table 5.2: Apart from FID which is calculated on the whole Test Set, metrics are expressed as mean ± standard deviation. In the Test Set, they exhibit a clear increase in data dispersion. Pix2Pix shows overfitting. The results between the two architectures are comparable, except for FID: Pix2Pix achieves much better values than cycleGAN.

The same Pix2Pix network shows better results when it is trained with the dataset in which cell instances boundaries are 2 pixels thick, both on the Training Set (Table 5.3) and the Test Set (Table 5.4). PCC exhibits higher dispersion on the Test Set. FID evidences better performance on the Test Set at the 50th epoch, followed by deterioration at later epochs (Fig 5.11).

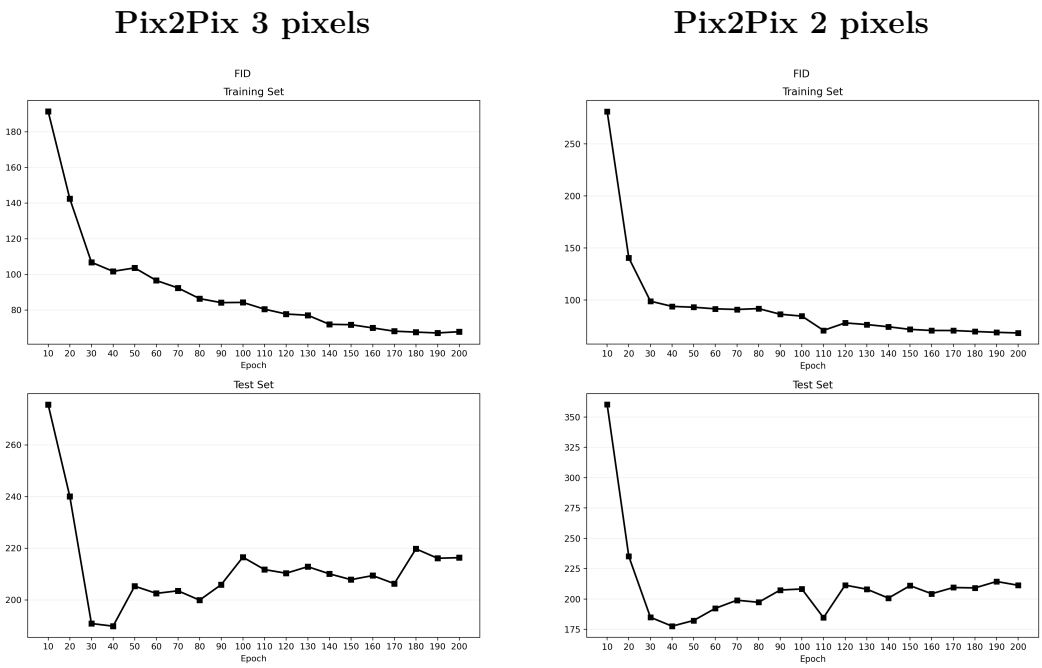

Figure 5.11: FID measurements by epoch of Pix2Pix networks trained with the 3 pixels and 2 pixels dataset.

Training Set

GAN	Dataset	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
Pix2Pix	3 pixels	20	89.61 ± 5.8	21.08 ± 0.99	73.11 ± 5.2	82.03 ± 2.34	142.47
		50	93.25 ± 4.34	23.19 ± 1.03	79.08 ± 4.63	85.48 ± 2.27	103.65
		100	95.19 ± 3.03	25 ± 0.91	83.33 ± 3.53	88.11 ± 1.77	84.29
		200	96.15 ± 2.48	26.08 ± 0.98	85.08 ± 3.23	89.44 ± 1.61	67.86
	2 pixels	20	88.72 ± 6.74	21.06 ± 0.99	72.25 ± 5.4	81.96 ± 2.49	140.65
		50	93.65 ± 4.08	23.43 ± 1.08	80.21 ± 4.49	85.9 ± 2.32	93.17
		100	95.32 ± 3.01	25.13 ± 0.92	83.55 ± 3.58	88.17 ± 1.93	84.66
		200	96.27 ± 2.48	26.23 ± 0.95	85.51 ± 3.17	89.64 ± 1.69	68.46

Table 5.3: Except for FID which is calculated on the whole Training Set, metrics are expressed as mean ± standard deviation. Training on the 2 pixels dataset shows slightly better results in terms of PCC, PSNR, SSIM and FSIM. FID exhibits better performance in the early epochs.

Test Set

GAN	Dataset	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
Pix2Pix	3 pixels	20	82.47 ± 10.66	17.03 ± 1.41	68.58 ± 6.28	78.48 ± 1.9	240.05
		50	81.9 ± 10.17	17.11 ± 1.35	67.11 ± 6.64	78.08 ± 1.76	205.3
		100	81.38 ± 10.52	17.22 ± 1.2	66.28 ± 5.98	77.08 ± 1.39	216.52
		200	80.8 ± 10.98	17.17 ± 1.2	64.77 ± 6.33	76.61 ± 1.46	216.33
	2 pixels	20	81.36 ± 11.05	17.08 ± 1.37	67.79 ± 5.94	78.02 ± 1.91	235.17
		50	81.38 ± 11.3	16.99 ± 1.5	67.42 ± 6.28	78.2 ± 1.8	182.25
		100	81.1 ± 11.06	17.09 ± 1.28	65.81 ± 6.03	77.08 ± 1.6	208.21
		200	80.81 ± 11.3	16.89 ± 1.36	64.77 ± 6.65	76.75 ± 1.5	211.29

Table 5.4: Apart from FID which is calculated on the whole Test Set, metrics are expressed as mean ± standard deviation. Test Set performance is comparable between the two datasets, except that FID shows better values on the 2 pixels dataset at the 50th epoch.

Visually, the instances show higher definition, at the same epoch, in the images generated through the 2 pixels dataset. The network succeeds in expressing the semantic content, encapsulated by the ground truths, in both datasets. Where the original image has areas where no cellular instances or white areas are present, the network is not successful in expressing the semantic content of those areas in the generated images (Fig. 5.12).

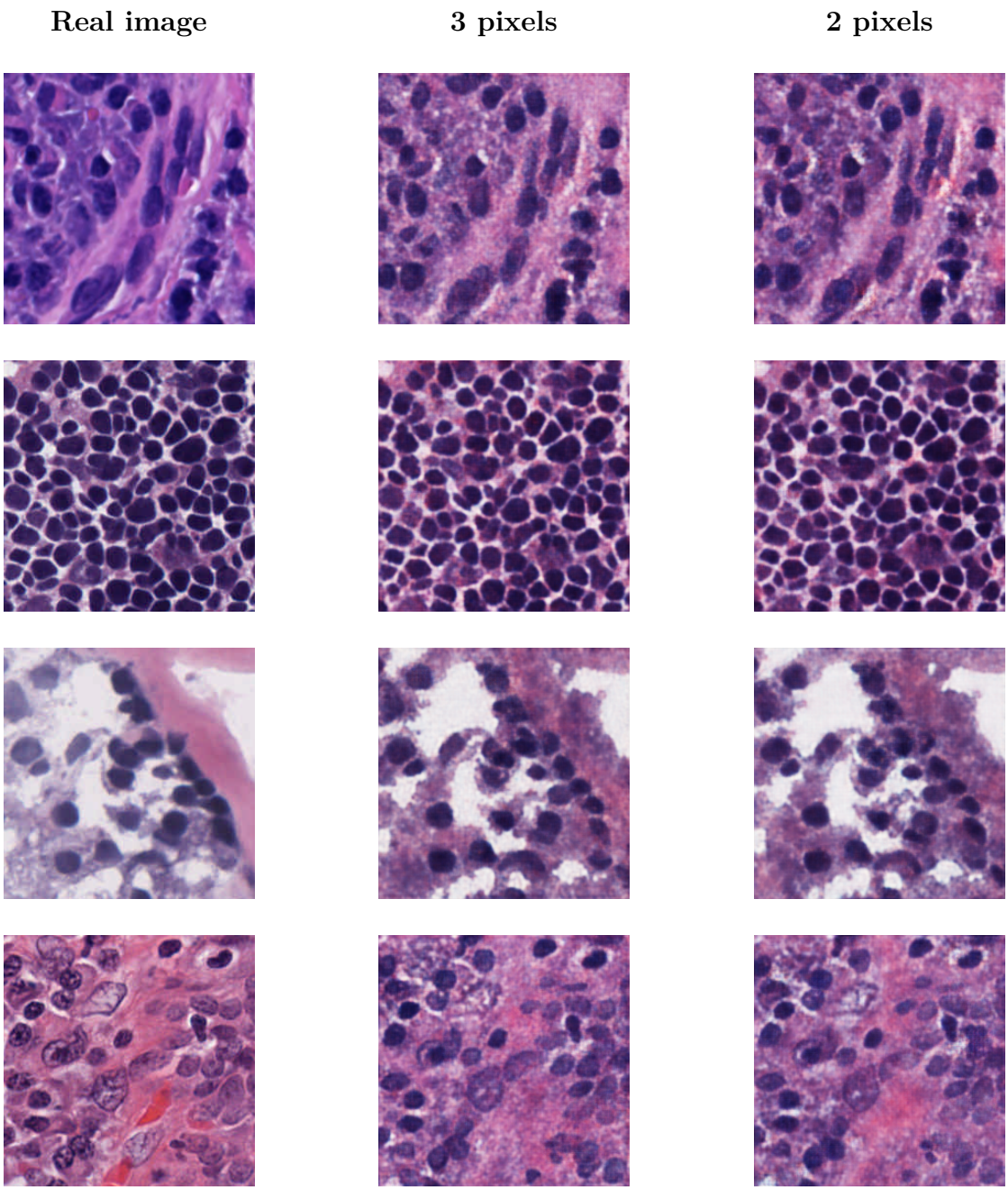

Figure 5.12: Comparison between real images and Pix2Pix generated images, trained on 2 pixels and 3 pixels datasets, at 50th epoch.

Pix2Pix networks trained with the 2 pixel dataset, with different loss functions show comparable results on both Training Set (Table 5.5) and Test Set (Table 5.6). For balanced models, that is, with the same number of filters per generator and discriminator, better performance is noticed on the Training Set as the number of filters increases. This improvement is not observed in the Test Set. Unbalanced models exhibit better performance as the number of filters increases and for more number of generator filters than the number of discriminator filters. The best networks are those with a number of generator filters equal to 64 and a number of discriminator filters equal to 16.

Training Set

Epoch	Loss	ngf	ndf	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
50	LSGAN	9	9	86.57 ± 6.68	20.18 ± 0.89	69.58 ± 5.45	80.59 ± 2.18	127.26
		24	24	91.91 ± 5.13	22.55 ± 1.02	77.37 ± 5.13	84.2 ± 2.32	92.04
		32	32	93.4 ± 4.32	23.44 ± 1.07	79.95 ± 4.57	85.76 ± 2.35	89.12
		9	40	86.62 ± 6.9	20.14 ± 0.89	70.05 ± 5.38	80.63 ± 2.45	153.05
		24	96	91.65 ± 4.84	22.12 ± 0.92	75.97 ± 4.91	83.92 ± 2.32	106.61
		32	128	93.65 ± 4.08	23.43 ± 1.08	80.21 ± 4.49	85.9 ± 2.32	93.17
		64	16	95.86 ± 2.46	25.35 ± 1.02	85.3 ± 3.57	88.76 ± 2.13	63.84
	Vanilla	9	9	85.4 ± 7.57	19.8 ± 0.85	67.8 ± 5.52	79.83 ± 2.45	129.01
		24	24	91.43 ± 4.7	22.24 ± 0.94	75.96 ± 4.61	83.7 ± 2.19	95.16
		32	32	93.4 ± 3.93	23.33 ± 1.02	79.59 ± 4.27	85.79 ± 2.13	92.07
		9	40	83.86 ± 7.89	19.37 ± 0.79	65.61 ± 5.54	78.84 ± 2.32	145.02
		24	96	90.76 ± 5.79	21.82 ± 0.98	74.88 ± 4.95	83.17 ± 2.42	105.88
		32	128	92.48 ± 4.86	22.86 ± 0.97	77.69 ± 4.58	84.98 ± 2.12	87.37
		64	16	95.92 ± 2.56	25.61 ± 1.01	85.7 ± 3.39	89.19 ± 2.04	64.20

Table 5.5: Training Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions and different number of filters for the generator and the discriminator. ngf and ndf mean the number of filters per generator and discriminator.

				Test Set				
Epoch	**Loss**	**ngf**	**ndf**	**PCC (%)**	**PSNR**	**SSIM (%)**	**FSIM (%)**	**FID**
50	LSGAN	9	9	80.46 ± 12.29	17.2 ± 1.29	67.7 ± 6.68	78.03 ± 1.87	229.75
		24	24	81.36 ± 11.49	17.46 ± 1.29	69.25 ± 6.84	78.28 ± 1.99	182.77
		32	32	81.03 ± 11.9	17.27 ± 1.27	68.29 ± 6.33	78.21 ± 1.82	180.09
		9	40	80.71 ± 11.2	16.66 ± 1.47	67.59 ± 6.03	77.76 ± 1.72	247.37
		24	96	80.98 ± 11.42	16.93 ± 1.34	66.48 ± 6.62	77.87 ± 1.88	196.38
		32	128	81.38 ± 11.3	16.99 ± 1.5	67.42 ± 6.28	78.2 ± 1.8	182.25
		64	16	82.54 ± 10.83	17.51 ± 1.29	69.37 ± 6.7	78.91 ± 1.94	162.39
	Vanilla	9	9	80.27 ± 10.34	16.12 ± 1.47	64.92 ± 5.61	76.48 ± 2.01	231.12
		24	24	80.58 ± 11.05	17.02 ± 1.26	66.53 ± 6.23	77.25 ± 1.66	185.83
		32	32	81.37 ± 10.72	16.82 ± 1.39	66.45 ± 5.56	77.3 ± 1.46	201.45
		9	40	79.42 ± 12.51	16.22 ± 1.65	64.22 ± 6.74	76.04 ± 1.9	231.58
		24	96	79.8 ± 11.17	16.46 ± 1.4	64.46 ± 5.61	76.01 ± 1.75	200.04
		32	128	80.45 ± 11.13	16.59 ± 1.48	64.4 ± 5.18	76.49 ± 1.29	202.98
		64	16	81.51 ± 11.54	17.37 ± 1.34	68.21 ± 6.48	78.13 ± 1.9	169.57

Table 5.6: Test Set performance of Pix2Pix models trained on 2 pixels dataset, with different loss functions and different number of filters for generator and discriminator. ngf and ndf mean the number of filters per generator and discriminator.

From the loss values in the training phase, it can be seen that LSGAN (Fig. 5.13) and vanilla (Fig. 5.14) show similar trends. Overall, the vanilla Pix2Pix and LSGAN Pix2Pix, with 64 generator filters and 16 discriminator filters, exhibit comparable results, both on the Training Set (Table 5.7) and the Test Set (Table 5.8). LSGAN turns out to be slightly better.

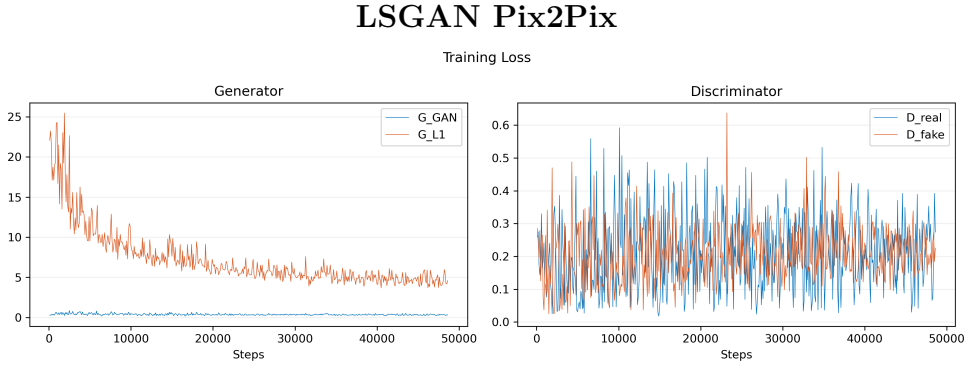

Figure 5.13: Loss values of generator and discriminator of LSGAN Pix2Pix network during training, accomplished on the 2 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original image and the generated image, the discriminator loss for the real image, the discriminator loss for the generated image, respectively.

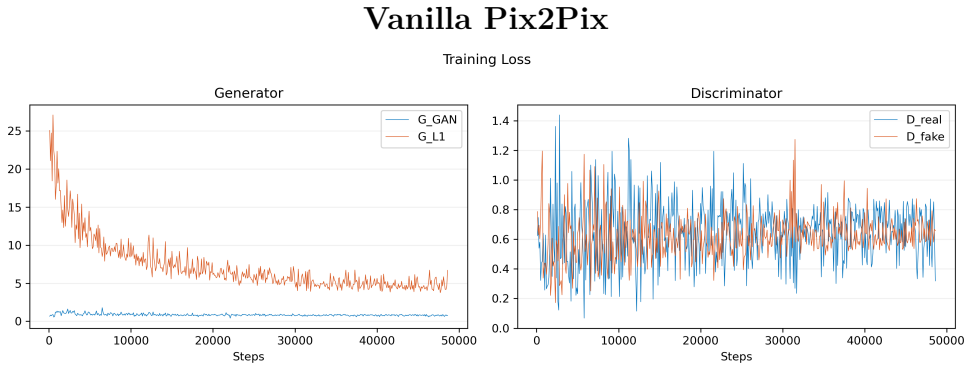

Figure 5.14: Loss values of generator and discriminator of vanilla Pix2Pix network during training, accomplished on the 2 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original image and the generated image, the discriminator loss for the real image, the discriminator loss for the generated image, respectively.

Training Set

Loss	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
LSGAN	20	92.22 ± 4.67	22.5 ± 1	78.05 ± 4.95	84.14 ± 2.52	119.11
	50	95.86 ± 2.46	25.35 ± 1.02	85.3 ± 3.57	88.76 ± 2.13	63.84
	100	97.57 ± 1.57	27.8 ± 0.99	90.15 ± 2.36	92.21 ± 1.47	55.68
	200	98.46 ± 1.06	30.2 ± 1.11	92.78 ± 1.77	94.38 ± 1.09	40.88
Vanilla	20	92.3 ± 4.73	22.47 ± 1.06	77.45 ± 4.92	84.58 ± 2.49	83.67
	50	95.92 ± 2.56	25.6 ± 1.01	85.7 ± 3.39	89.19 ± 2.04	64.2
	100	97.58 ± 1.58	28.04 ± 1	90.17 ± 2.3	92.41 ± 1.38	51.88
	200	98.45 ± 1.08	30.19 ± 1.19	92.72 ± 1.86	94.39 ± 1.12	39.65

Table 5.7: Training Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions and 64 filters for generator and 16 for discriminator.

Test Set

Loss	Epoch	PCC (%)	PSNR	SSIM (%)	FSIM (%)	FID
LSGAN	20	82.45 ± 11.44	17.67 ± 1.24	70.62 ± 7.24	78.69 ± 2.13	241.54
	50	82.54 ± 10.83	17.52 ± 1.29	69.37 ± 6.7	78.91 ± 1.94	162.39
	100	82.88 ± 10.17	17.51 ± 1.24	68.93 ± 6.67	78.79 ± 1.99	174.85
	200	82.36 ± 10.6	17.64 ± 1.24	67.97 ± 6.76	78.32 ± 1.96	185.6
Vanilla	20	81.37 ± 11.59	17.15 ± 1.36	67.71 ± 6.54	78.51 ± 1.86	176.11
	50	81.51 ± 11.54	17.37 ± 1.34	68.21 ± 6.48	78.13 ± 1.9	169.57
	100	81.9 ± 10.97	17.4 ± 1.33	67.45 ± 6.06	77.95 ± 1.81	189.92
	200	81.94 ± 11.08	17.35 ± 1.43	67.47 ± 6.26	78.05 ± 1.74	196.78

Table 5.8: Test Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions and 64 filters for generator and 16 for discriminator.

The LSGAN Pix2Pix, trained on 2 pixel dataset with 64 generator filters and 16 discriminator filters, is able to generate better images among all trained networks (Fig. 5.16). It can be seen that the network achieves good semantic content and image quality as early as the 50th epoch. Performance worsening in later epochs is visually demonstrated by the network's tendency to separate larger area cell instances into smaller ones (Fig. 5.15). Furthermore in some Test Set images, although the network succeeds in expressing the semantic content of the ground truths, it is unable to account for the staining of the original histological images.

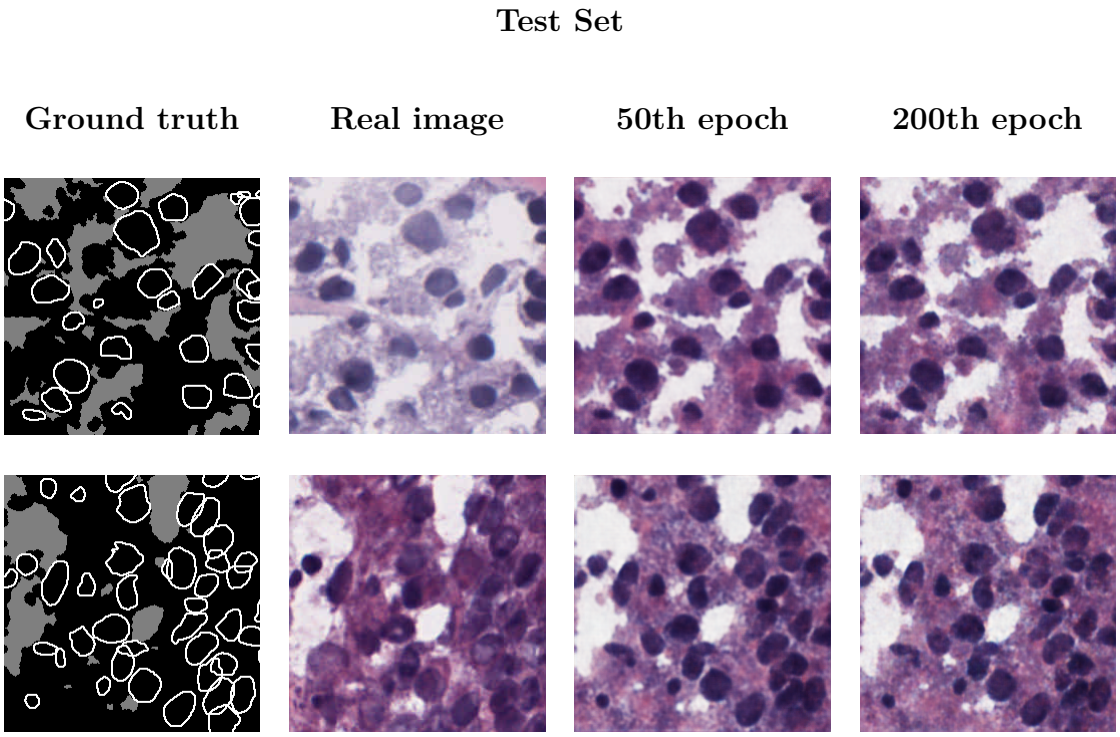

Figure 5.15: Comparison between Test Set real images and generated images, with LSGAN Pix2Pix (64 generator filters and 16 discriminator filters), at 50th and 200th epochs.

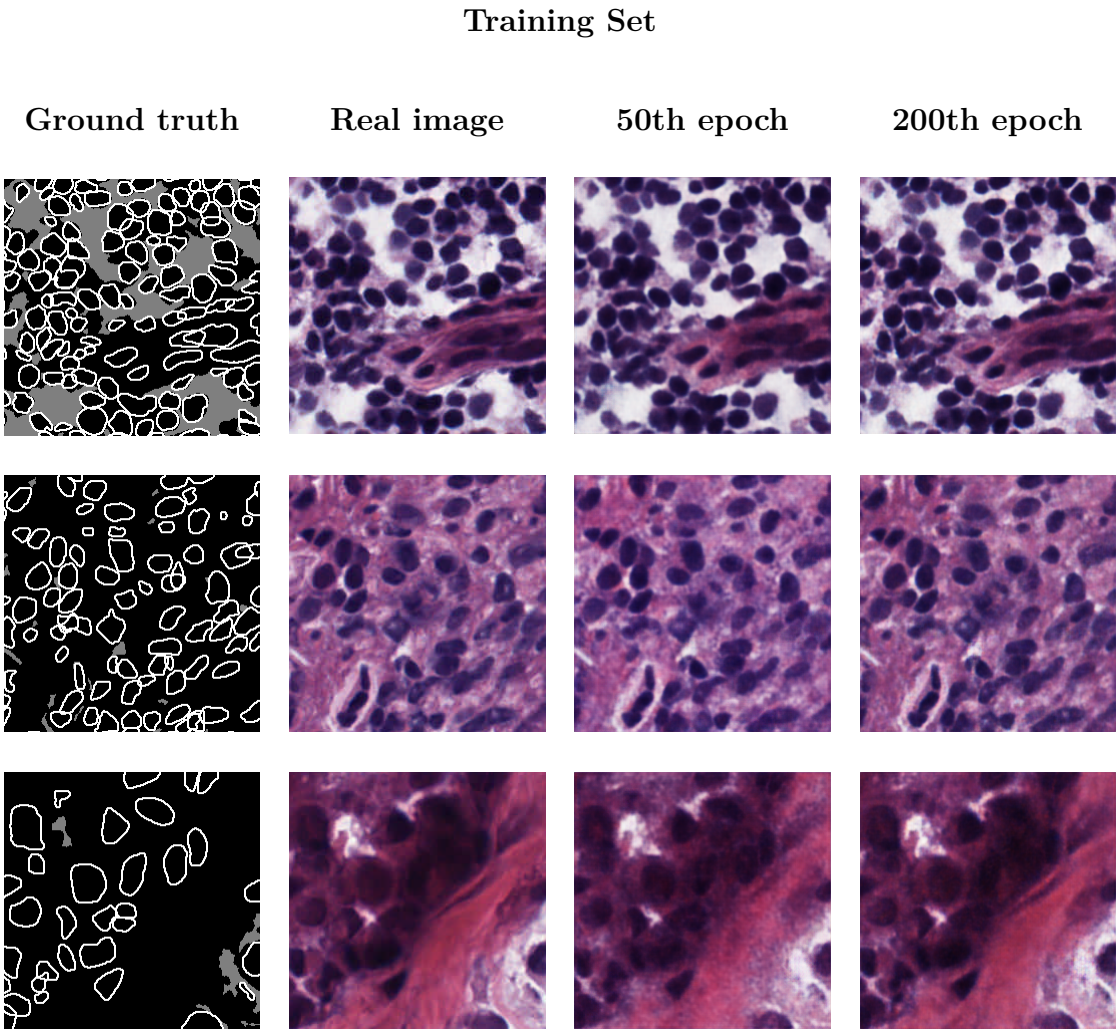

Figure 5.16: Comparison between Training Set real images and generated images, with LSGAN Pix2Pix (64 generator filters and 16 discriminator filters), at 50th and 200th epochs.

5.2 Image to mask results

Because cycleGAN operates two-way image generation, the same Pix2Pix models were also trained in automatic mask generation. The same logical thread of the experiments for histological image generation is followed.

Loss values of the Pix2Pix network trained on the 3 pixels dataset for mask generation (Fig. 5.17), show predictable trends, comparable to the reverse generation direction. Because cycleGAN is trained simultaneously in both generation directions, the trends are similar (Fig. 5.18) to the other generation direction.

Figure 5.17: Loss values of generator and discriminator of Pix2Pix network during training, accomplished on the 3 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original mask and the generated mask, the discriminator loss for the real mask, the discriminator loss for the generated mask, respectively.

To compare the two architectures, Pix2Pix and cycleGAN, in automatic mask generation, DSC is used as metric. The latter is evaluated with respect to the two classes of interest: cell instance boundaries and white zones.

On the Training Set (Table 5.9), while cycleGAN is able to identify the contours of instances better than Pix2Pix, the performance on identifying white zones is

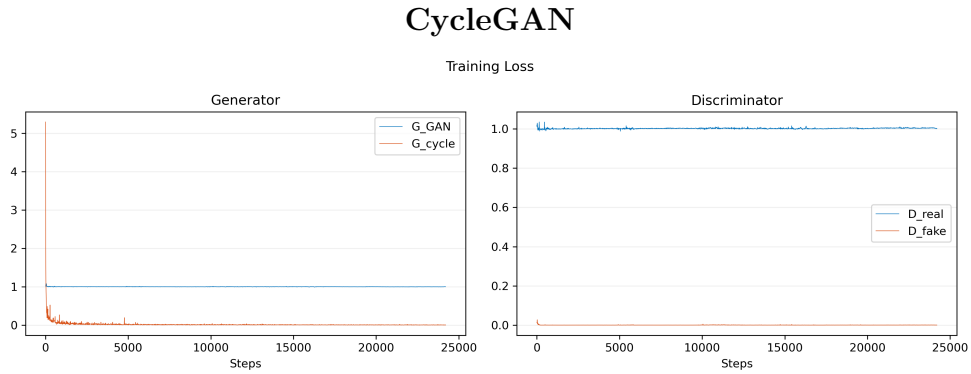

Figure 5.18: Loss values of generator and discriminator, aimed at creating histological images, of the cycleGAN network during training, accomplished on the 3 pixels dataset. G_GAN, G_cycle, D_real, D_fake indicate the generator loss, cycle consistency loss between the real mask and the generated mask, discriminator loss for the real mask, discriminator loss for the generated mask, respectively.

worse than Pix2Pix. However, it is important to note that the performance drops on the Test Set (Table 5.10), particularly in boundary detection. Furthermore, the high dispersion in DSC whites values precludes a comparison between the two architectures.

Training Set

Dataset	GAN	Epoch	DSC boundaries (%)	DSC whites (%)
3 pixels cell instance boundaries	Pix2Pix	20	74.35 ± 4.48	67.32 ± 20.41
		50	85.17 ± 2.12	77.39 ± 16.85
		100	91.92 ± 1.3	83.97 ± 12.68
		200	95.31 ± 1.03	87.92 ± 9.67
	CycleGAN	20	71.6 ± 4.41	43.34 ± 22.91
		50	86.81 ± 1.72	52.24 ± 24.99
		100	94.06 ± 1.19	67.77 ± 24.35
		200	97.19 ± 0.71	78.05 ± 21.53

Table 5.9: Training Set performance of the two architectures, trained on the 3 pixels dataset, in automatic mask generation.

		Test Set		
Dataset	**GAN**	**Epoch**	**DSC boundaries (%)**	**DSC whites (%)**
3 pixels cell instance boundaries	Pix2Pix	20	56.33 ± 5.24	74.87 ± 22.6
		50	56.48 ± 4.97	80.53 ± 18.1
		100	58.46 ± 5.3	81.92 ± 17.62
		200	58.74 ± 5.77	82.74 ± 16.97
	CycleGAN	20	52.4 ± 6.16	54.28 ± 24.82
		50	56.85 ± 6.33	45.47 ± 25.03
		100	58.26 ± 6.99	53.79 ± 27.52
		200	57.62 ± 8.02	55.3 ± 27.54

Table 5.10: Test Set performance of the two architectures, trained on the 3 pixels dataset, in automatic mask generation.

The same Pix2Pix network is also trained in automatic mask generation by taking as reference the dataset in which the cell instance boundaries are 2 pixels thick. Decreasing the thickness of the instance contours lowers the probability that the network can correctly identify the boundaries. This is demonstrated by a decrease in performance from training on the 3 pixels dataset to training on the 3 pixels dataset (Tables 5.11, 5.12). Since we did not change the white areas between the two datasets, the results are comparable.

Training Set

GAN	Dataset	Epoch	DSC boundaries (%)	DSC whites (%)
Pix2Pix	3 pixels	20	74.35 ± 4.48	67.32 ± 20.41
		50	85.17 ± 2.12	77.39 ± 16.85
		100	91.92 ± 1.3	83.97 ± 12.68
		200	95.31 ± 1.03	87.92 ± 9.67
	2 pixels	20	60.94 ± 6.48	62.95 ± 24.53
		50	74.14 ± 4.58	74.67 ± 21.94
		100	85 ± 3.24	81.16 ± 19.97
		200	90.53 ± 2.51	84.88 ± 18.96

Table 5.11: Training Set performance of the LSGAN Pix2Pix network trained on two different datasets. The network has 128 generator filters and 32 discriminator filters.

Test Set

GAN	Dataset	Epoch	DSC boundaries (%)	DSC whites (%)
Pix2Pix	3 pixels	20	56.33 ± 5.24	74.87 ± 22.6
		50	56.48 ± 4.97	80.53 ± 18.1
		100	58.46 ± 5.3	81.92 ± 17.62
		200	58.74 ± 5.77	82.74 ± 16.97
	2 pixels	20	17.84 ± 9.46	74.93 ± 23.31
		50	42.09 ± 5.69	78.91 ± 18.48
		100	44.69 ± 5.14	81.42 ± 18.07
		200	44.48 ± 5.43	82.16 ± 17.65

Table 5.12: Test Set performance of the LSGAN Pix2Pix network trained on two different datasets. The network has 128 generator filters and 32 discriminator filters.

In order to compare Pix2Pix networks trained on the 2 pixels dataset with different loss functions and different numbers of filters per generator and discriminator, the 50th epoch is chosen as the reference. It is noted for balanced models, same number of filters per generator and discriminator, that higher number of filters results in increased performance in boundary detection. For the unbalanced models, the performance improvement is driven by an increase in the number of filters and a greater number of filters for the generator than the number of filters for the discriminator, especially for the LSGAN Pix2Pix.

Training Set

Epoch	Loss	ngf	ndf	DSC boundaries (%)	DSC whites (%)
50	LSGAN	9	9	0 ± 0	76.02 ± 21.41
		24	24	64.33 ± 5.67	72.26 ± 22.75
		32	32	71.54 ± 5.02	74.34 ± 22.32
		9	40	0 ± 0	76.45 ± 21.01
		24	96	63.25 ± 6.29	71.84 ± 23
		32	128	74.14 ± 4.58	74.67 ± 21.94
		64	16	85.65 ± 2.79	81.75 ± 19.96
	Vanilla	9	9	13.99 ± 4.91	52.09 ± 24.41
		24	24	60.76 ± 5.28	70.09 ± 22.71
		32	32	65.15 ± 5.14	73.44 ± 21.93
		9	40	38.78 ± 7.05	58.68 ± 25.03
		24	96	63.62 ± 4.74	73.44 ± 21.45
		32	128	66.9 ± 4.25	75.5 ± 20.94
		64	16	80.45 ± 3.25	80.45 ± 20.13

Table 5.13: Training Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions and different number of filters for the generator and the discriminator. ngf and ndf mean the number of filters per generator and discriminator.

White zone detection performance does not drop by moving from the Training Set (Table 5.13) to the Test Set (Table 5.14), however, the data show high scatter. This may be due to the number of pixels, belonging to the white areas, which changes significantly from image to image. There is a noticeable drop in performance on the Test Set with regard to contour identification. Overall, the best models turn out to be the LSGAN Pix2Pix and vanilla Pix2Pix with 64 filters for the generator and 64 filters for the discriminator.

				Test Set	
Epoch	**Loss**	**ngf**	**ndf**	**DSC boundaries (%)**	**DSC whites (%)**
50	LSGAN	9	9	0 ± 0	82.2 ± 16.9
		24	24	39.09 ± 5.77	77.94 ± 18.82
		32	32	42 ± 4.86	80.05 ± 18.09
		9	40	0 ± 0	81.56 ± 18.82
		24	96	35.67 ± 6.7	78.3 ± 18.75
		32	128	42.09 ± 5.69	78.91 ± 18.48
		64	16	46.53 ± 5.48	82.02 ± 18.44
	Vanilla	9	9	9.74 ± 4.63	67.64 ± 22.72
		24	24	46.51 ± 5.18	79.24 ± 18.99
		32	32	47.42 ± 5.19	80.22 ± 17.58
		9	40	28.69 ± 7.68	73.85 ± 23.34
		24	96	47.46 ± 5.09	79.7 ± 20.04
		32	128	48.58 ± 5.8	80.59 ± 18.94
		64	16	49.87 ± 6.18	82.28 ± 17.32

Table 5.14: Test Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions and different number of filters for the generator and the discriminator. ngf and ndf mean the number of filters per generator and discriminator.

There are no evident differences in loss trends between LSGAN Pix2Pix and vanilla Pix2Pix (Fig. 5.19, 5.20). LSGAN

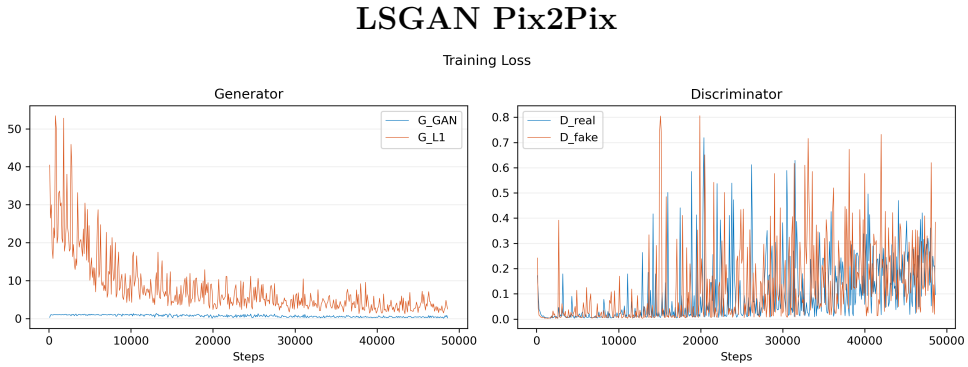

Figure 5.19: Loss values of generator and discriminator of LSGAN Pix2Pix network during training, accomplished on the 2 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original image and the generated image, the discriminator loss for the real image, the discriminator loss for the generated image, respectively.

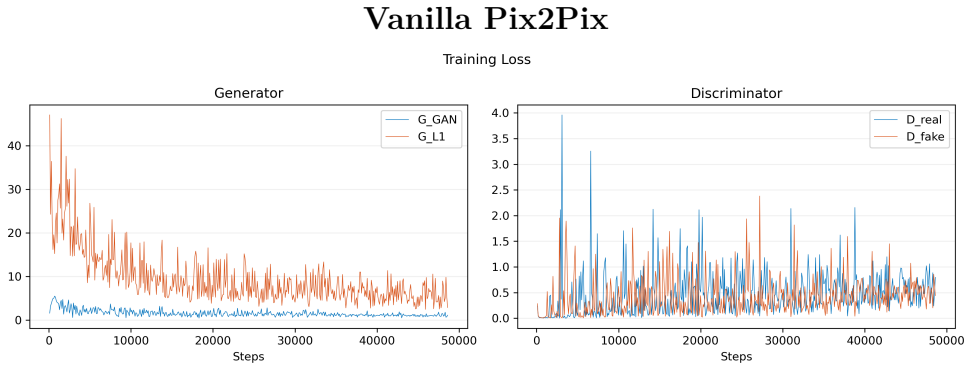

Figure 5.20: Loss values of generator and discriminator of vanilla Pix2Pix network during training, accomplished on the 2 pixels dataset. G_GAN, G_L1, D_real, D_fake indicate the generator loss, the loss between the original image and the generated image, the discriminator loss for the real image, the discriminator loss for the generated image, respectively.

Regarding boundary identification, a comparison of the two Pix2Pix networks with 64 generator and 16 discriminator filters and different loss functions shows that

the vanilla Pix2Pix achieves marginally better results on the Test Set, although there is a noticeable drop in performance compared to the Training Set. White zone identification achieves comparable results between the two networks. In addition, there is no noticeable decrease in white area identification performance, although the data shows high dispersion in the Training Set as well.

Training Set

Loss	Epoch	DSC boundaries (%)	DSC whites (%)
LSGAN	20	66.13 ± 4.83	71.06 ± 23.34
	50	85.65 ± 2.79	81.75 ± 19.96
	100	91.27 ± 2.02	86.52 ± 18.75
	200	94.73 ± 1.56	90.09 ± 18.26
Vanilla	20	68.49 ± 4.14	71.46 ± 22.92
	50	80.45 ± 3.25	80.45 ± 20.13
	100	86.49 ± 2.76	85.64 ± 18.55
	200	90.08 ± 2.54	89.18 ± 18.19

Table 5.15: Training Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions, 64 filters for generator and 16 for discriminator.

		Test Set	
Loss	**Epoch**	**DSC boundaries (%)**	**DSC whites (%)**
LSGAN	20	43.97 ± 5.34	78.54 ± 20.85
	50	46.53 ± 5.48	82.02 ± 18.44
	100	48.24 ± 5.79	83.04 ± 17.88
	200	48.79 ± 5.49	83.52 ± 17.34
Vanilla	20	49 ± 5.62	78.98 ± 19.46
	50	49.87 ± 6.18	82.28 ± 17.32
	100	49.45 ± 5.52	82.84 ± 17.97
	200	50.21 ± 5.66	84.1 ± 16.92

Table 5.16: Test Set performance of Pix2Pix models trained on the 2 pixels dataset, with different loss functions, 64 filters for generator and 16 for discriminator.

Examples of automatic masks of the Test Set and Training Set are presented below. As confirmed by the metrics, the network is able to segment white areas, performing even on the Test Set (Fig. 5.21). On the other hand, it can be seen that this does not happen for the cellular instance boundaries. When there are multiple overlapping instances, the network does not achieve realistic segmentation. Moreover, many times the circular crown closure that should characterize the cell instance boundary is missed, especially on the Test Set but also in the Training Set (Fig. 5.22). Moreover, the network is unable to detect cellular instances when they, in the original image, do not present adequate contrast with their surroundings, even on the Training Set.

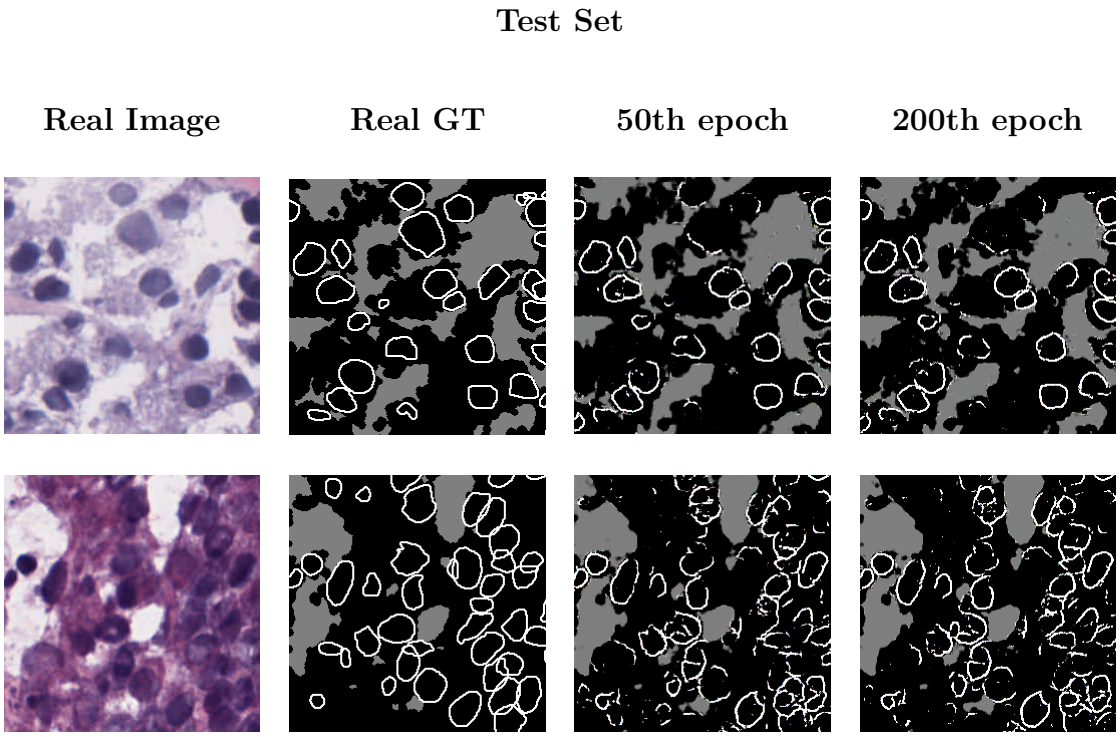

Figure 5.21: Comparison between Test Set real masks and generated soft masks, with LSGAN Pix2Pix (64 generator filters and 16 discriminator filters), at 50th and 200th epochs.

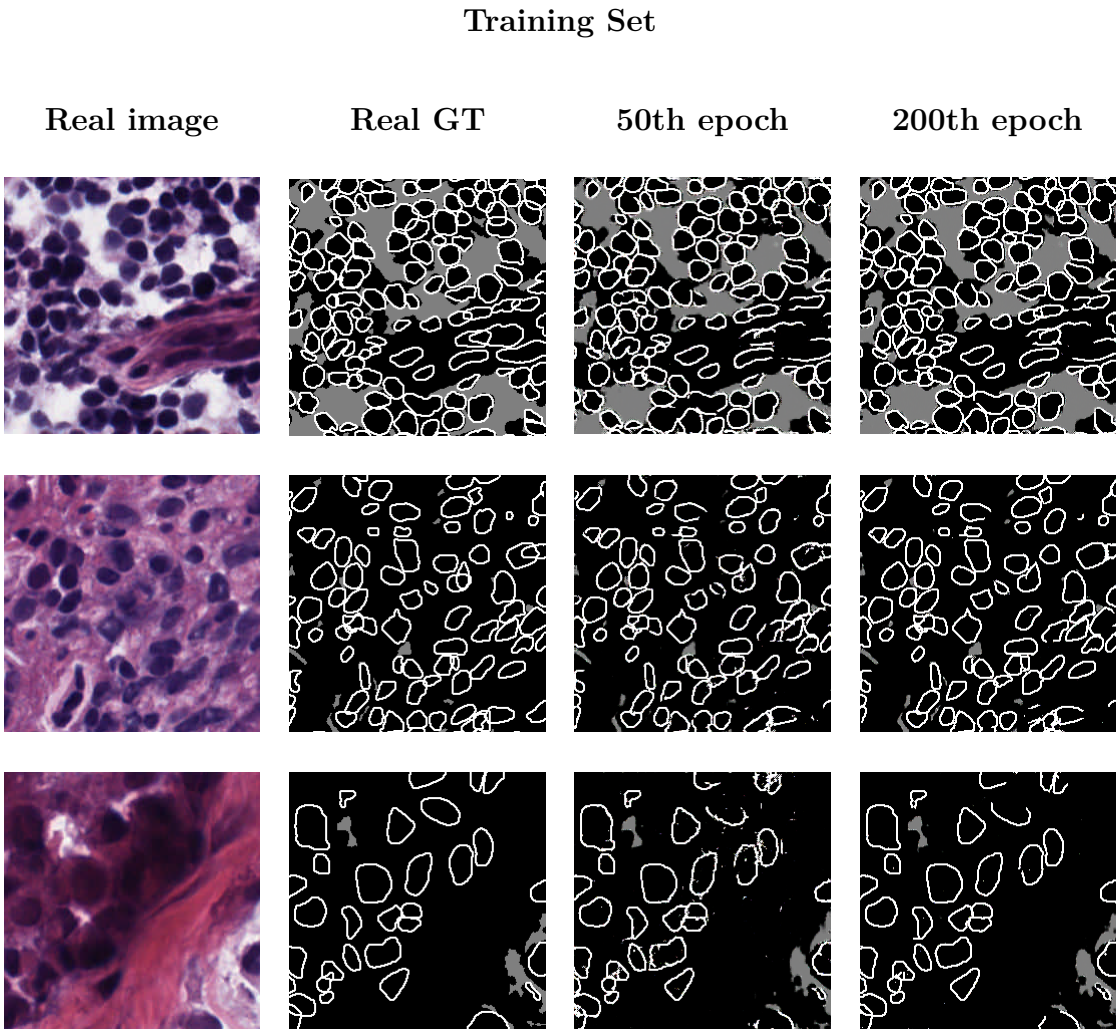

Figure 5.22: Comparison between Training Set real masks and generated soft masks, with LSGAN Pix2Pix (64 generator filters and 16 discriminator filters), at 50th and 200th epochs.

Chapter 6

Conclusions

In this thesis work, we proposed a new application of deep learning in digital pathology. A comparison is presented between two GAN models, Pix2Pix and cycleGAN, aimed at generating realistic histological images through predefined ground truths. The aim is to control the semantic content of the generated images. The models are trained on paired data. In order to achieve a better match between ground truth and histological images, manual annotations of three different operators are combined. With the aim of controlling the semantic content of the generated histological images, information on the contours of the individual cell instances and the white areas of the corresponding histological images are inserted. The quality of the generated images is assessed by means of the following metrics: Pearson Correlation Coefficient (PCC), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Feature Similarity Index Measure (FSIM) and Fréchet Inception Distance (FID). In recent years, numerous efforts have been made to create metrics that can objectively judge the image quality. The metrics used

for model validation have led to satisfactory results. However, it would be useful to submit the generated images to the assessment of experienced pathologists.

Although Pix2Pix leads to images with fewer artefacts and better results than cycleGAN, both models are able to express the semantic content contained by ground truth. With the same training parameters and characteristics Pix2Pix is able to generate visually better images and achieve appreciable results in fewer training epochs than cycleGAN. Both models underperform on the Test Set. This is certainly due to the use of images from an organ that has never been seen in the training phase and the presence of different forms of heterogeneity, chief among them, the use of different scanners and acquisition protocols. Visually, an appreciable correspondence is achieved between the real and generated images at the level of the cell instances and at the level of the white zones. The models are able to recreate the same number of instances, the same shape and boundary morphology. The models are able to distinguish the cell instances, just by defining their boundaries.

Even though the models succeed in achieving the desired semantic content, they are not able to adequately reconstruct the background areas of the real images. In order to improve the proposed model, this aspect suggests that, in the future, the information content within the ground truth can be increased by means of additional class labels for the detection of other tissue components, so as to better control the semantic content and generate images that are as realistic as possible. The models are many times unable to recreate the staining of the real image, also due to the presence of different tissues from multiple sources within the dataset. In order to allow the generative models to focus on other aspects, future work

could be aimed at understanding if standardising the images from a staining point of view, through the downstream use of stain normalisation algorithms or the implementation within the proposed models of style transfer techniques, can lead to better performance.

The images used for training are limited in size (256x256 pixels at 40x magnification). There is nothing to prevent the proposed model from working on images with higher resolution and a larger field of view. It should also be considered that the performance by varying the overlap percentage of patches was not evaluated. Future work could consider these analyses.

The use of the same models, previously trained for image generation, as segmentation networks led to acceptable results for the detection of white areas but not for the detection of individual cell boundaries.

Histological image synthesis through ground truths with specific characteristics leads to several advantages. By checking semantic ground truths, there is the possibility of inserting new images within datasets to increase the generalizability of a ML/DL-based model or to evaluate the performance of such models. Furthermore, the possibility of creating new images can be useful for educational purposes.

This thesis work seeks to offer insights and stimuli in order to continue creating increasingly detailed and informative ground truths to control the semantic content of histological images.

Acronyms

AI artificial intelligence

DL Deep Learning

DSC Dice Similarity Coefficient

DP Digital Pathology

FID Fréchet Inception Distance

FSIM Feature Similarity Index Measure

GAN Generative Adversarial Network

GT Ground Truth

H&E Hematoxylin and Eosin

HVS Human Visual System

IQA Image Quality Assessment

I2I Image-to-Image

ML Machine Learning

M2I Mask-2-Image

PCC Pearson Correlation Coefficient

SSIM Structural Similarity Index Measure

STAPLE Simultaneous Truth and Performance Level Estimation

TCGA The Cancer Genome Atlas

UHCW University Hospitals Coventry and Warwickshire

WSI Whole Slide Image

Z2I Latent-to-Image

Bibliography

[1] Hitoshi Tsuda et al. «Evaluation of the Interobserver Agreement in the Number of Mitotic Figures Breast Carcinoma as Simulation of Quality Monitoring in the Japan National Surgical Adjuvant Study of Breast Cancer (NSAS-BC) Protocol». In: *Japanese journal of cancer research* 91.4 (2000), pp. 451–457 (cit. on p. 7).

[2] Josefin Persson, Ulrica Wilderäng, Thomas Jiborn, Peter N Wiklund, Jan-Erik Damber, Jonas Hugosson, Gunnar Steineck, Eva Haglind, and Anders Bjartell. «Interobserver variability in the pathological assessment of radical prostatectomy specimens: findings of the Laparoscopic Prostatectomy Robot Open (LAPPRO) study». In: *Scandinavian journal of urology* 48.2 (2014), pp. 160–167 (cit. on p. 7).

[3] David M Metter, Terence J Colgan, Stanley T Leung, Charles F Timmons, and Jason Y Park. «Trends in the US and Canadian pathologist workforces from 2007 to 2017». In: *JAMA network open* 2.5 (2019), e194337–e194337 (cit. on p. 7).

[4] Aldis H Petriceks and Darren Salmi. «Trends in pathology graduate medical education programs and positions, 2001 to 2017». In: *Academic pathology* 5 (2018), p. 2374289518765457 (cit. on p. 7).

[5] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. «Generative adversarial networks (2014)». In: *arXiv preprint arXiv:1406.2661* 1406 (2014) (cit. on p. 9).

[6] Alec Radford, Luke Metz, and Soumith Chintala. «Unsupervised representation learning with deep convolutional generative adversarial networks». In: *arXiv preprint arXiv:1511.06434* (2015) (cit. on p. 10).

[7] Tero Karras, Samuli Laine, and Timo Aila. «A style-based generator architecture for generative adversarial networks». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410 (cit. on p. 10).

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. «Large scale GAN training for high fidelity natural image synthesis». In: *arXiv preprint arXiv:1809.11096* (2018) (cit. on p. 10).

[9] H Zhang, I Goodfellow, D Metaxas, et al. «Odena. Self-attention generative adversarial network». In: *Proc. Int. Conf. Mach. Learn.* 2019, pp. 7354–7363 (cit. on p. 10).

[10] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. «Spectral normalization for generative adversarial networks». In: *arXiv preprint arXiv:1802.05957* (2018) (cit. on p. 10).

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. «Imagenet: A large-scale hierarchical image database». In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on p. 10).

[12] Laya Jose, Sidong Liu, Carlo Russo, Annemarie Nadort, and Antonio Di Ieva. «Generative adversarial networks in digital pathology and histopathological image processing: A review». In: *Journal of Pathology Informatics* 12.1 (2021), p. 43 (cit. on p. 11).

[13] Maximilian E Tschuchnig, Gertie J Oostingh, and Michael Gadermayr. «Generative adversarial networks in digital pathology: a survey on trends and future potential». In: *Patterns* 1.6 (2020), p. 100089 (cit. on p. 11).

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. «Image-to-image translation with conditional adversarial networks». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134 (cit. on pp. 11, 13, 33, 36, 39).

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. «Unpaired image-to-image translation using cycle-consistent adversarial networks». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232 (cit. on pp. 11, 15, 33).

[16] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. «Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images». In: *Medical Image Analysis* 58 (2019), p. 101563 (cit. on p. 19).

[17] Quoc Dang Vu et al. «Methods for segmentation and classification of digital microscopy tissue images». In: *Frontiers in bioengineering and biotechnology* (2019), p. 53 (cit. on p. 19).

[18] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. «Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images». In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1196–1206 (cit. on p. 19).

[19] Amirreza Mahbod, Gerald Schaefer, Benjamin Bancher, Christine Löw, Georg Dorffner, Rupert Ecker, and Isabella Ellinger. «CryoNuSeg: A dataset for nuclei instance segmentation of cryosectioned H&E-stained histological images». In: *Computers in biology and medicine* 132 (2021), p. 104349 (cit. on pp. 19, 22, 23).

[20] Andrew Janowczyk and Anant Madabhushi. «Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases». In: *Journal of pathology informatics* 7.1 (2016), p. 29 (cit. on p. 19).

[21] Ruchika Verma, Neeraj Kumar, Abhijeet Patil, Nikhil Cherian Kurian, Swapnil Rane, and Amit Sethi. «Multi-organ nuclei segmentation and classification challenge 2020». In: *IEEE transactions on medical imaging* 39.1380-1391 (2020), p. 8 (cit. on p. 19).

[22] Neeraj Kumar et al. «A multi-organ nucleus segmentation challenge». In: *IEEE transactions on medical imaging* 39.5 (2019), pp. 1380–1391 (cit. on p. 19).

[23] Birgid Schömig-Markiefka et al. «Quality control stress test for deep learning-based diagnostic model in digital pathology». In: *Modern Pathology* 34.12 (2021), pp. 2098–2108 (cit. on p. 22).

[24] Amjad Khan et al. «Impact of scanner variability on lymph node segmentation in computational pathology». In: *Journal of pathology informatics* 13 (2022), p. 100127 (cit. on p. 23).

[25] Simon K Warfield, Kelly H Zou, and William M Wells. «Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation». In: *IEEE transactions on medical imaging* 23.7 (2004), pp. 903–921 (cit. on p. 26).

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. «U-net: Convolutional networks for biomedical image segmentation». In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241 (cit. on p. 33).

[27] Adalberto Claudio Quiros, Roderick Murray-Smith, and Ke Yuan. «PathologyGAN: Learning deep representations of cancer tissue». In: *arXiv preprint arXiv:1907.02644* (2019) (cit. on p. 36).

[28] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. «Least squares generative adversarial networks». In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2794–2802 (cit. on pp. 38, 39).

[29] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. «Multiscale structural similarity for image quality assessment». In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003.* Vol. 2. Ieee. 2003, pp. 1398–1402 (cit. on p. 41).

[30] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. «FSIM: A feature similarity index for image quality assessment». In: *IEEE transactions on Image Processing* 20.8 (2011), pp. 2378–2386 (cit. on p. 41).

[31] DC Dowson and BV666017 Landau. «The Fréchet distance between multivariate normal distributions». In: *Journal of multivariate analysis* 12.3 (1982), pp. 450–455 (cit. on p. 42).

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. «Gans trained by a two time-scale update rule converge to a local nash equilibrium». In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 42).

[33] Deepak Baby and Sarah Verhulst. «Sergan: Speech enhancement using relativistic generative adversarial networks with gradient penalty». In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2019, pp. 106–110 (cit. on p. 44).