Master Thesis in Statistics and Machine Learning

Generative Adversarial Networks to enhance decision support in digital pathology

Alessia De Biase

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

June 2019

Supervisors
Anders Eklund, Nikolay Burlutskiy

Examiner
Fredrik Lindsten

You cannot save the world, but you can save yourself and the light that you bring. Because that is what the world needs.
More light!

Contents

Abstract

Histopathological evaluation and Gleason grading on Hematoxylin and Eosin (H&E) stained specimens is the clinical standard in grading prostate cancer. Recently, deep learning models have been trained to assist pathologists in detecting prostate cancer. However, these predictions could be improved further regarding variations in morphology, staining and differences across scanners. An approach to tackle such problems is to employ conditional GANs for style transfer. A total of 52 prostatectomies from 48 patients were scanned with two different scanners. Data was split into 40 images for training and 12 images for testing and all images were divided into overlapping 256x256 patches.

A segmentation model was trained using images from scanner A, and the model was tested on images from both scanner A and B. Next, GANs were trained to perform style transfer from scanner A to scanner B. The training was performed using unpaired training images and different types of Unsupervised Image to Image Translation GANs (CycleGAN and UNIT). Beside the common CycleGAN architecture, a modified version was also tested, adding Kullback Leibler (KL) divergence in the loss function. Then, the segmentation model was tested on the augmented images from scanner B.

The models were evaluated on 2,000 randomly selected patches of 256x256 pixels from 10 prostatectomies. The resulting predictions were evaluated both qualitatively and quantitatively. All proposed methods outperformed in AUC, in the best case the improvement was of 16%. However, only CycleGAN trained on a large dataset demonstrated to be capable to improve the segmentation tool performance, preserving tissue morphology and obtaining higher results in all the evaluation measurements. All the models were analyzed and, finally, the significance of the difference between the segmentation model performance on style transferred images and on untransferred images was assessed, using statistical tests.

Acknowledgments

I would like to thank ContextVision, for embracing my enthusiasm during my interview that day in early October last year, giving me the opportunity to work with them and the confidence to face such a challenging project. Thanks Arto, I'm so glad I went for the "interesting" plan either than the "safe" one!

Thanks to the Digital Pathology team to be so patience with my small knowledge in the medical field and for the nice moments we spent together. My special thanks to Giorgia, for being rough and sweet with me in the perfect amount, for opening my eyes in front of my biased judgments and for being my daily Italian pill, grazie bella.

My deep gratitude to my supervisors Anders Eklund and Nikolay Burlutskiy for support and guidance during the planning and the development of this research work, but also for the time they spent to work with me. Thanks Anders for always encouraging me to test new ideas, I learned to believe in them much more now. Thanks Nikolay for believing in this work maybe more than I did, for giving me opportunities I would have never imagined and for teaching me how to be autonomous and independent (especially after rebooting a computer).

Thanks to my opponent Simon Jönsson and my examiner Fredrik Lindsten for the useful comments provided at the revision meeting.

To my daily supporter Michele, the one who can be there always, even if not physically. Thanks for being my right arm since that first day of failure in the Algebra and Topology exam, thanks for showing me that there are still people with love and passion in what they study and not only robots who pass tests. Thanks for being there to complain together about the system, about life, about jobs and everything else; for being so patient and also a little harsh, just as much as I need.

To the best gift Sweden could ever give me, Beatrice! You really added light to the first darkest winter. Thank you for always sharing positive energy with me, for always seeing things from a complete different prospective, for opening my eyes when I only see dark and it's actually so bright. Thanks for making my extreme emotional ups and downs unique strengths and not weaknesses!

To the people who joined me in this two years adventure, to my classmates. To my faithful lab partner and best friend from day one, Alejandro. To all the sweet friends I met here, to the people I lived with, to everyone who made those two years unforgettable. A thousand times thank you!

To my little Italy, Le Borgate, thanks for being my safe, happy place. Thanks for being HOME!

My very great appreciation to my family, to my grandparents who express their love and approval asking about how cold is Sweden, to my sweet grandma who was my true example of who a warrior is, to my cousins who will always take me back in time, to Jajy who did not break our promise and never will.

Last but not least, to my mom, my dad and my sister, my guiding light, thanks for the possibility to choose, to leave, to travel, to design my future as I wish. Thanks for listening, supporting, encouraging, trusting. Thanks Consy for fighting battles always next to me! Thanks for all this, anything would have been possible without you!

Thanks Sweden, for our continuous love and hate relationship!

Nomenclature

AI	Artificial Intelligence
AUC	Area Under Curve
cGANs	Conditional Generative Adversarial Networks
CNN	Convolutional Neural Network
CycleGAN	Cycle-Consistent Generative Adversarial Network
DL	Deep Learning
DNN	Deep Neural Network
GANs	Generative Adversarial Networks
H&E	Hematoxylin and eosin
KL	Kullback Leibler
NN	Neural Network
SSIM	Structural Similarity Index
UNIT	Unsupervised Image to Image Translation
WSI	Whole Slide Image

Glossary

Prostatectomy: surgical removal of all or a part of the prostate gland.

Staining: artificial coloration of a substance to facilitate examination of tissues, microorganisms, or other cells under the microscope.

H&E: haematoxylin and eosin stain is one of the principal stains in histology, it makes use of a combination of two dyes – haematoxylin and eosin. Eosin is an acidic dye, staining structures red or pink. Haematoxylin can be considered as a basic dye, staining structures purplish blue.

Stain-Normalization: method which involves transforming an image I into another image J using a mapping function that matches the visual appearance of a given image to the target image.

RGB: additive color model in which red, green and blue light are added together in many ways in order to reproduce a broad array of colors.

RGBA: RGB color model supplemented with a 4th alpha channel indicating how opaque each pixel is.

YCbCr: additive color model defined by a mathematical coordinate transformation from an associated RGB color space. It is widely used in video and digital photography applications. Y is the luminance component, Cb and Cr are the blue-difference and red-difference chroma components.

Encoder: network that takes the input, and output a feature map/vector/tensor which hold the information, the features, that represent the input.

Decoder: network that takes the feature vector from the encoder, and gives the best closest match to the actual input or intended output.

Autoencoder: network that works as both encoder and decoder. It is trained to attempt to copy its input to its output.

1. Introduction

1.1. Background

Histopathology is the discipline of analyzing tissue samples on a cell level to determine the existence of an abnormal condition such as cancer. Traditionally, the tissue samples are analyzed under a microscope, after being stained with a procedure that makes the morphology of the sample visible. However, a shift to digitalization of microscopic evaluation has started in recent years. This means that the tissue samples are scanned with a high-resolution scanner and the analysis of the sample is performed at a workstation where images can be viewed, compared, enlarged, and eventually analyzed using digital applications.

Such digital applications could, for instance, detect and segment suspicious areas, count mitosis (cell division), grade cancer areas with respect to severity [9]. The most promising technology to create such decision support tools is Deep Learning (DL). ContextVision has started a new Digital Pathology business unit with the objective to design and sell tools for the analysis of various types of cancer in tissues. The first product is going to be used for prostate cancer, utilizing the most recent advances in DL and Artificial Intelligence (AI). This will help the pathologists to provide better and faster diagnoses.

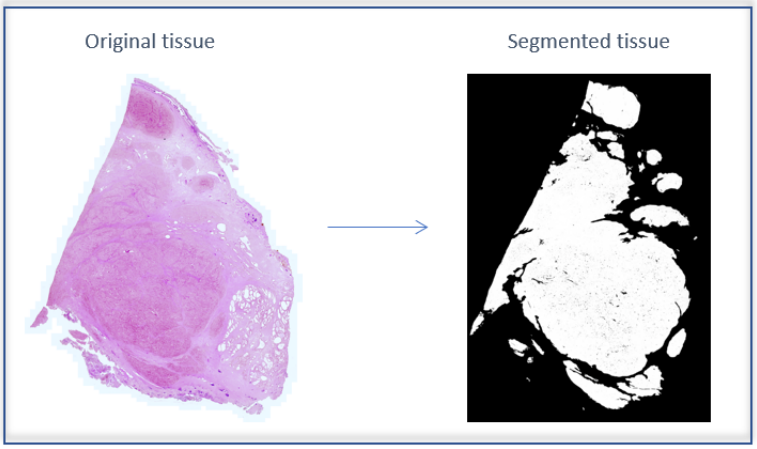

Figure 1.1.: Example of prostate tissue stained with H&E (on the left) and its cancer annotation (on the right).

Training a Deep Neural Network (DNN) requires a large amount of data which should summarize the huge variability existing in this field. Variability in histopathological data results from different experimental protocols across pathology labs, differences in slide preparation, different staining procedures, different scanners, etc. . These variations cause inconsistencies between pathologists, but they also affect the performance of the segmentation tool [21]. There are two approaches to overcome this problem, one focuses on training the segmentation tool on a big, diverse dataset and the other one operates on the test set, transferring it to the training set style (normalization of data). The first approach aims to increase variability in the data, the second one to decrease it.

1.2. Aim

The aim of this thesis work is to explore a new augmentation technique called Generative Adversarial Networks [11], to improve the performance of ContextVision's decision support tool. The goal is to augment test data using style transfer from the training set, such that the segmentation tool can become invariant to changes not strictly related to tissue morphology. In more details, this work aims to improve a segmentation model trained on images obtained with a specific scanner, while testing on images from a different one (see Figure 1.2). The objective of this thesis work can be summarized as following:

- Are Generative Adversarial Networks an effective approach, as preprocessing step, to reduce the impact that 'non-biological' variations on histopathology data has on the performance of a computer driven segmentation tool?
- Are all the Unsupervised Image to Image translation methods (CycleGAN and UNIT) able to significantly improve predictions of the segmentation tool in the same way?

These questions will be evaluated with ContextVision segmentation tool.

1.3. Related works

Increasing variability in histopathology data is a very challenging task, data augmentation techniques have been widely used to introduce differences in color, stain, etc., but capturing all variations that occur in real-world tissue staining is nearly impossible. A DL algorithm, which is able to detect cancer on tissues, needs to be tuned every time new variations are introduced, this is time-consuming and it is a bottleneck for a pathologist. For this reason, a strategy which aims to normalize images, to mimic the data that a network was trained on, was preferred in recent approaches. In more details, in recent works [23, 6, 21, 24], *Generative Adversarial Networks (GANs)* [11], especially *Conditional Generative Adversarial Networks*

(cGANs) [14], have been used as stain normalization methods for histopathological images, showing significant impact on performance of classification systems, enhancing their predictions. Common data augmentation techniques (i.e. flip, rotation, zooming, color augmentation, etc.) do not affect tissue morphology and, due to their linear nature, risk to ovesimplify data variability [23]. In a GAN setup, instead, a Generator network is responsible to learn a domain mapping from one style to another one (style transfer), generating synthetic images whom a Discriminator network learns to classify as fake or real. The learned mapping does not only change color or stain appearance, style transferred images may change significantly compared to the original in both content and structure [24].

Changes in morphology, on histopathology data, represent a big issue, for this reason related works tried to enforce the network to preserve tissues structure while learning. In [4], for example, this problem is addressed on a loss function level, using an edge-weighted L_2 regularization that encourages the Generator to preserve salient image edges of the ground truth input, multiplying both input image and generated image (using an element-wise multiplication) with the color gradient vector field of the input image. In [24] the photorealism and the structural similarity loss (SSIM) are introduced to keep the structural information unchanged. Photorealism loss uses Matting Laplacian transform (defined in [16]) to measure the structural differences and is calculated using all three RGB color channels of the images. SSIM has been used for assessing the image quality, to regulate structural changes instead of focusing on pixel to pixel transformations. Working with gray-scale images, instead of colored images, generally favors texture-based features, showing then large improvements in preserving tissue morphology. In [6], in fact, GANs are used to transfer a certain style after gray-normalization is performed first on input images.

Conditional Generative Adversarial Networks (cGANs) are used as both supervised, if paired data from two different scanners or institutes are available, or unsupervised methods, in case of unpaired data. One of the most common technique for style transfer between two image domains, under an unsupervised setting, is *Cycle-Consistent Adversarial Networks (CycleGANs)* [26]. *StainGAN,* in [21], for example, is a pure learning based approach that handles the problem of stain normalization as a style-transfer problem, using CycleGANs to transfer the H&E Stain Appearance between Hamamatsu to Aperio scanners. Even though the model was trained on unpaired images, paired images were available for evaluations of results showing significant improvement compared to the state-of-the-art methods on similarity metrics. Usually, in case of unpaired images, a classification network is used to assess the quality of style transfer methods [4, 6]. In *StainGAN*, for example, the Discriminator is asked to compute also the segmentation model task.

In this thesis work, the potential of unpaired Image to Image translation techniques, using GANs, is explored as a style transfer method for histopathology prostate images, scanned with two different scanners. Three different methods are proposed as solution: CycleGAN [26], a novel method obtained from a modified version of CycleGAN (a loss function is added to the main objective to reinforce the Generators

learning) and UNIT (UNsupervised Image to Image translation) [17]. Unpaired patches are used as training set, the test set obtained as output is then used as input for a segmentation tool implemented by ContextVision to obtain predictions of cancer areas. The evaluation of results in this thesis are in accordance with the requirements in ContextVision and are based on the data and the tools provided by the company.

1.4. Ethical considerations

The data used in this project consisted of human medical imaging data and corresponding meta-information. The data provider was responsible for handling the ethical, legal and privacy aspects relevant to the data. The images used for this work were anonymized, no information about patients was given beside the stained tissue digitalized image. According to General Data Protection Regulation (EU), european regulation on data protection and privacy, the type of data used in this work can be used for research purpose. In fact, it states the following:

"The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable. This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes." [1]

Doing research on human tissues may raise many ethical questions. When a person has tissues removed, as part of a treatment, he/she is asked for permission and consent to allow that tissue to be available for research studies. Being able to work with such personal and sensitive data allows to generate new knowledge, to support pathologists in their daily work and to speed up the process of cancer detection. Generating synthetics images, which are similar to real images, allows to reduce the number of tissues asked to laboratories to be scanned, increasing the number of quality data available for research. It is worth to notice that "real-world" data can not be totally substituted, there will always be a new variation which was never observed before and which is out of the dataset boundary. However, synthetic images yield considerable benefits for DL methods, which require huge amounts of data for reaching their full performance.

Style transfer and Image to Image translation can also be seen as a digital image manipulation technique, altering reality. For medical images, in particular, it can represent a big issue. Using Artificial Intelligence in healthcare is promising and powerful, however in this work it not used with the aim of replacing pathologists but as decision support.

[1] GENERAL DATA PROTECTION REGULATION (GDPR), recital 26.

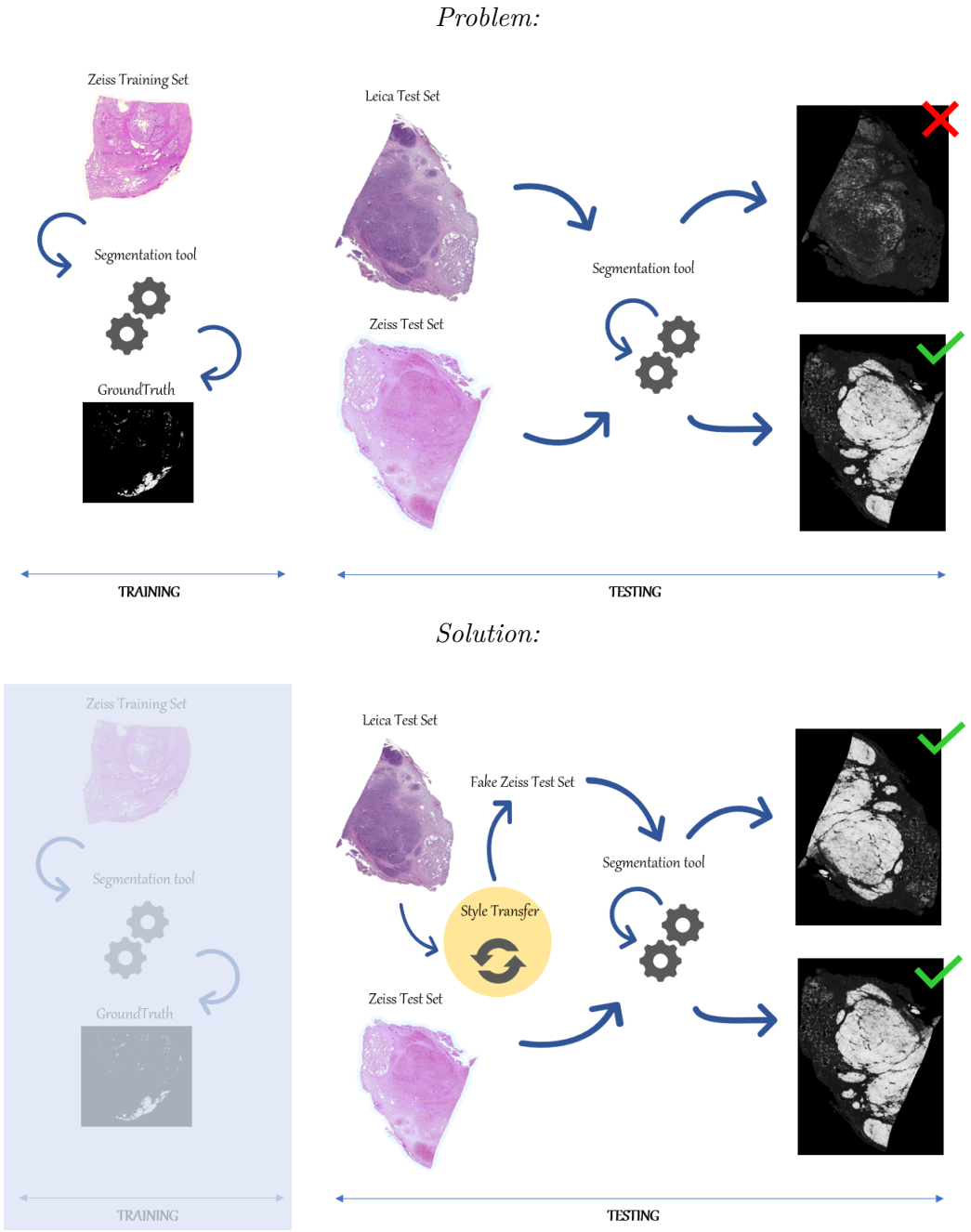

Figure 1.2.: Description of the problem addressed by this thesis project and the proposed solution. The segmentation tool used to predict cancer on tissues, in this work, is pre-trained on images scanned with Zeiss Axio Scan.Z1 scanner. The test set is made of images scanned with the same scanner used for training, but also images scanned with Leica Aperio AT2 scanner. The predictions obtained by testing on both sets show that the segmentation tool is sensitive to changes not related to tissues morphology. Visually comparing these predictions with the cancer annotation of this stained tissue (right image in Figure 1.1), in fact, predictions on Zeiss images results better in quality. The proposed solution aims to improve the performance of the segmentation tool when it is tested on images scanned with Leica scanner, transferring Zeiss style to them. The style transfer methods are trained on Zeiss and Leica training sets and then tested on Leica test set.

2. Data

Digital pathology data consist of tissue biopsy sample slides scanned as *Whole Slide Images* (WSIs). The scanning process aims to produce high quality images from conventional glass slides, to be able to analyze the tissue on a computer monitor instead of a microscope. The size of the scanned images is in the range from 50,000x50,000 to 100,000x100,000 pixels, making it impossible to work with in their maximum resolution, because of their size and the computer memory limitations. It is important for the pathologists to have the overview of the entire image, but it is also essential to access in finer details. For this reason, WSI are stored at multiple resolutions [1] to accommodate a streamlined method for loading them. Images are saved in a pyramid structure (see Figure 2.1): the WSI consists of multiple images at different magnification [2] where the pyramid provides distinct zoom levels. The base of the pyramid has the highest resolution while the top has the lowest one [9].

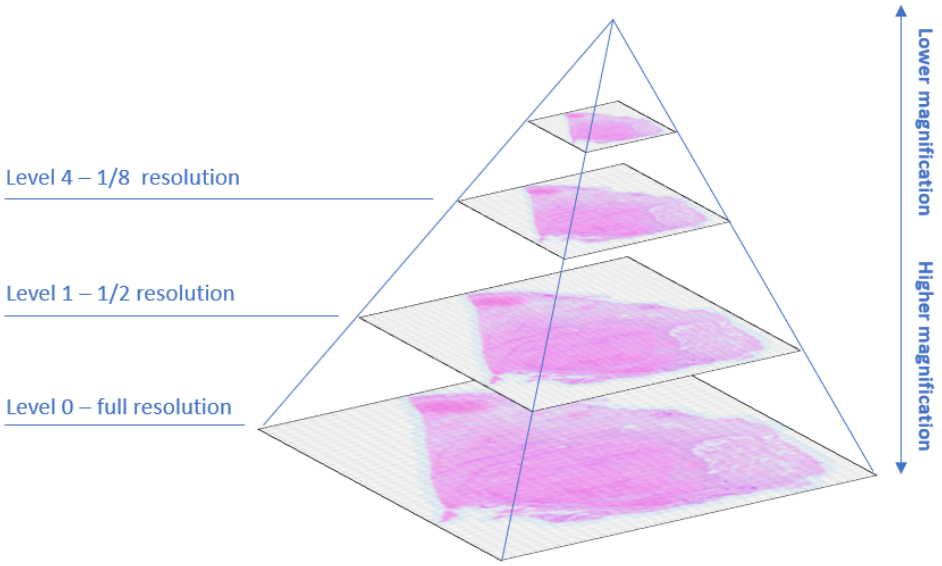

Figure 2.1.: Whole Slide Image pyramid structure

Given a slide, the pathologist identifies cancer areas and annotates those regions

[1] Resolution is the amount of information that can be seen in the image, the smallest distance below which two discrete objects will be seen as one.

[2] Magnification is how large the image is compared to real life.

generating a segmented image. In training and testing a segmentation model those annotations are the labels the predictions are compared with (see Figure 2.2), so they represent the ground truth images.

2.1. Data sources

The dataset used in this work is a collection of images of stained prostate tissues, from the medical technology company ContextVision, stained with Hematoxylin and Eosin (H&E) dyes. The physical size of the tissue samples is around 2x2 cm^2. In a style transfer problem between two styles A and B, two mappings from A to B and from B to A are learned, hence, for both training and testing, data from both domains are needed. The training set is composed of 40 slides scanned with a Zeiss Axio Scan.Z1 and 45 slides scanned with a Leica Aperio AT2. The test set is composed of 10 slides scanned with Zeiss Axio Scan.Z1 scanner and 9 slides scanned with Leica Aperio AT2 scanner. The same tissue sections were scanned by both scanners, some images were excluded due to quality and no registration [3] was performed. Hence, images are not aligned and not all paired, so they are treated as unpaired data. Slides were scanned at a resolution of 0.22 μm per pixel for Zeiss and 0.5 μm for Leica and then resized to 0.44 μm per pixel. Their dimension varies, both width and height values are between 30,000 pixels to 70,000 pixels.

In addition to WSIs, for each of the test set slides a ground truth image is also provided in a smaller resolution, obtained with the method described in [5] and approved by pathologists. This method is based on the idea according to which the presence of basal cells is an indicator for healthy glands, implying that their absence show potential cancerous areas. Compared to the Gleason grading clinical standard, this annotation technique resulted in more objective ground truth images, due to the fact that the presence of basal cells can be assessed by using immunohistochemical markers [5].

2.2. Data preprocessing

For consistency with the data used for training the segmentation tool by ContextVision, the level chosen in the WSI is level 1 is reasonable to detect prostate cancer. However, training neural networks on gigapixel resolution whole slide images is computationally expensive. For this reason from each slide 256x256 pixels patches are extracted, discarding cases where the background covered more than 40% of the total area.

To avoid boundary effects, overlapping (by 15%) patches are selected.

As a result, a large amount of patches is obtained for each set of slides (see Table 2.1).

[3] Image registration is the process of transforming different sets of data into one coordinate system.

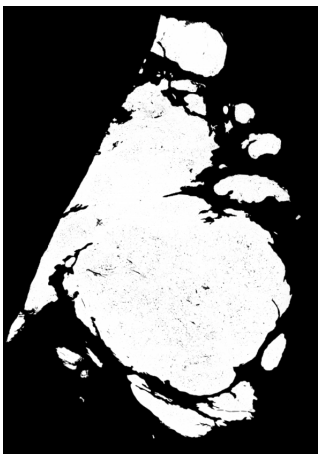

Figure 2.2.: Example of ground truth image: white pixels represent cancer while black pixels represent not cancer tissue.

	# training patches	*# testing patches*
Zeiss	202,268	59,226
Leica	262,349	67,040

Table 2.1.: Number of patches obtained after data preprocessing, each patch is 256x256 pixels.

The name of each patch follows the format *"SlideName_x_y.jpg"* , where x and y indicate the coordinates of the top left corner in the original slide at level 0, and *SlideName* indicates the name of the slide that patch comes from. When images are saved into arrays they can have different representations. One of them is as RGBA object: each pixel is a combination of four channels (red green and blue plus alpha indicating opacity) each of them represented by a number from 0 to 255. In this color scale white is obtained having 255 in each channel while black having 0 in each channel. A WSI usually has a white background but because of the differences across scanners, it can also happen to be darker than pure white, that is why the definition of background pixel in this work is stated as any pixel having all channels values of its RGBA representation above 235.

2.3. Data description

Because of the large amount of settings each scanner has, slides coming from two different scanners can look totally different in colors, brightness, contrast (see Figure 2.3).

Differences between Leica and Zeiss slides can be detected on a quality (see Figure 2.3) level but also numerically as will be shown later (see Table 2.2).

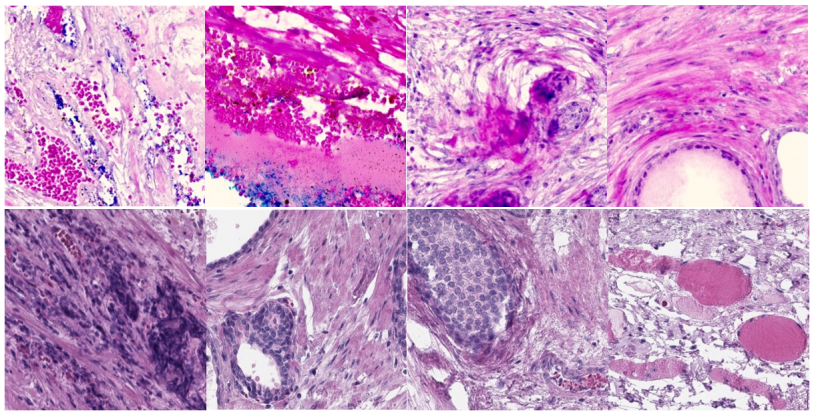

Figure 2.3.: Zeiss patches are showed in the first row while Leica's patches in the second one. The patches differ in style, including color and brightness. The main goal of this master thesis is to transfer the style of one scanner to the other one, using deep learning.

Visually Zeiss images look lighter and more fluorescent while Leica images are more opaque and less sharp in color differences. Also the color scales are quite different, hot and deep pink for Zeiss and more violet and lavender for Leica.

A different way of encoding an image is with YCbCr representation which also uses three components to describe a pixel. The first component describes a gray scale brightness called luminance (Y), the other two tell how much Blue (Cb) and Red (Cr) is needed to get a desired color. While in the RGBA model each color appears as a combination of red, green, and blue, YCbCr is more useful with digital images because of its luminance channel taking into account also the light intensity of the color (brightness). Given a pixel represented in RGB format, the YCbCr components can be obtained with the following equations [2]:

$$\begin{cases} Y = 16 + \frac{65.738R}{256} + \frac{129.057G}{256} + \frac{25.064B}{256} \\ Cb = 128 - \frac{37.945R}{256} - \frac{74.494G}{256} + \frac{112.439B}{256} \\ Cr = 128 + \frac{112.439R}{256} - \frac{94.154G}{256} - \frac{18.285B}{256} \end{cases}$$

A total of 2,000 random patches from Zeiss slides and 2,000 from Leica slides are used to detect differences for each color channel. First, each patch is split into the three channels, then pixels values of the same channel over all patches are merged for each of the two datasets and the histograms are calculated (see Figure 2.4). For both Cb and Cr the distributions are quite different across scanners, Zeiss histograms represent a higher variance compared to Leica in both cases. For the luminance channel, instead, the distributions seem to be very similar, just slightly shifted in mean value.

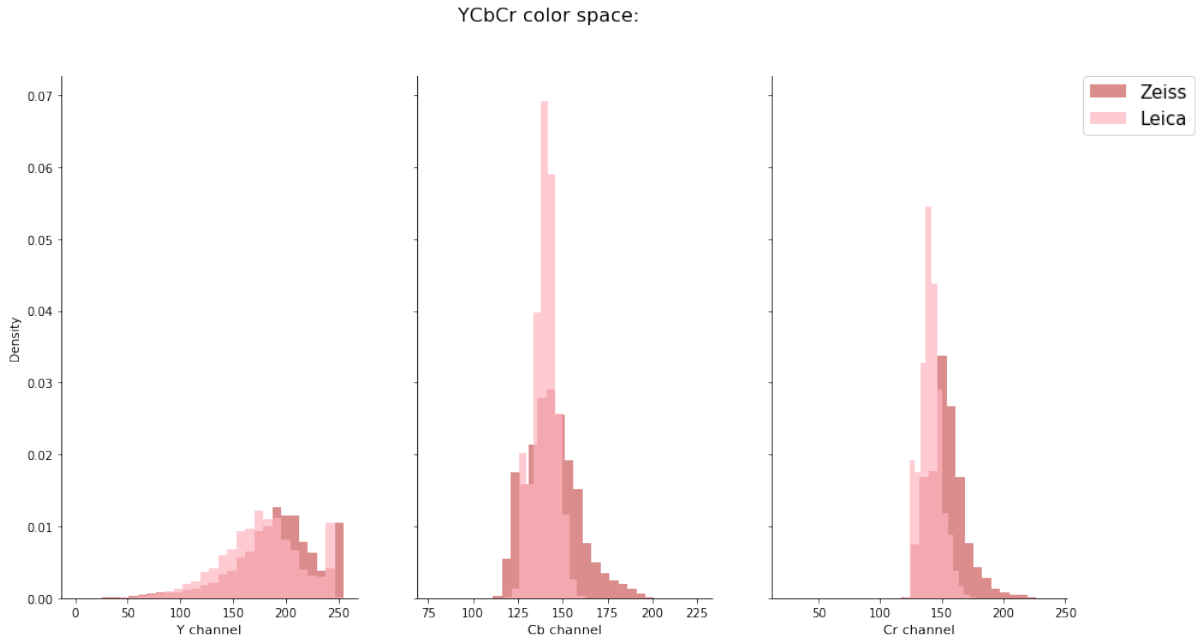

Figure 2.4.: Comparison between Y, Cb and Cr color histograms for Zeiss and Leica images

But how big is the difference? A good statistical way to measure how one probability distribution is different from a second one is through Kullback–Leibler divergence (D_{KL}), also called relative entropy, defined as:

$$D_{KL}(P \,||\, Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{2.1}$$

where, in this specific case, P refers to Zeiss histogram channels, Q to Leica histogram, X represents the range of possible pixel values each channel has. According to this definition, two distributions are identical if the KL divergence value is zero [13]. In Table 2.2 KL divergence is calculated per each channel between Zeiss images and Leica's. The result is totally coherent with the previous analysis from the color histograms, the main differences between Zeiss and Leica image domains are related to Cb and Cr channel where the value of KL divergence is higher than zero. Relative entropy does not measure distances between two image domains, while the amount of information lost per channel when Leica's images are used instead of Zeiss. From a deep learning point of view, high values of KL divergence (far from zero), per channel, could cause misleading results of predictions.

	Y Channel	*Cb Channel*	*Cr Channel*
KL(Zeiss,Leica)	0.61	1.41	1.13

Table 2.2.: Kullback-Leibler divergence between Zeiss and Leica Y, Cb and Cr color histograms.

3. Methods

The methods used to achieve the thesis objective belong to the family of GANs, Generative Adversarial Networks [11], a class of machine learning systems with deep neural network architectures. In the first part of this chapter, a definition of how neural networks are applied to image data is given, followed by an overview on what GANs are and a more detailed description of the Image to Image translation techniques used in this work. In the final part, the evaluation methods are described and the assessments of data quality will be given.

3.1. Image to Image translation

An image-to-image translation problem consists of mapping an image from one domain to a corresponding image from another domain (e.g. converting a summer image into a winter image). From a probabilistic point of view, the key point is to learn two data generating distributions to be able to perform image translation across the two domains.

3.1.1. Convolutional Neural Networks

Neural Networks are particularly powerful for analysis of images, especially for their ability to automatically extract useful features from unstructured data. Images are arrays of numbers representing each pixel, so training a neural network on such data would not take into account the spatial structure of the image but it would consider all pixels independently. For this reason a special architecture resulted a better option for image analysis: Convolutional Neural Networks (CNN). They are made of an input layer, an output layer and several hidden layers (hence the name "deep" networks), some of which are convolutional. Unlike Neural Networks, the layers of a Convolutional Network have neurons arranged in 3 dimensions: width, height, depth (see Figure 3.1).

Convolution is one of the main building blocks of a CNN and it is performed on the input image using a *filter* (the convolutional matrix) in a *Feature Extraction* step. The filter, with a given size, slides over the input, performing element-wise multiplication and the resulting sum goes into the feature map. The amount, by which the filter slides, is referred to as the *stride* (see Figure 3.2).

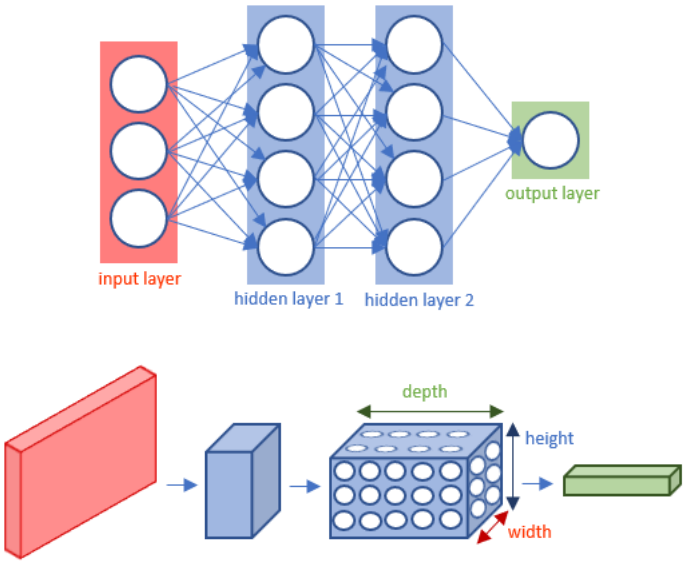

Figure 3.1.: Neural Network (on the top) vs Convolutional Neural Network (on the bottom) Architecture. A CNN arranges its neurons in three dimensions (width, height, depth). The input layer in a CNN holds the image, so its width and height would be the dimensions of the image and the depth would represent Red, Green and Blue channels (dimension of 3). A 3D input volume is transformed into a 3D output volume of neuron activactions.

Feature maps are built performing many convolutions on the input matrix, and then all the feature maps are put together as final output of the convolution layer. For a given layer in a CNN the weights are shared. After each convolution layer a pooling layer aims to reduce dimensionality and also the number of parameters and computations in the network. The last step in a CNN is *Classification,* here fully connected layers use the features obtained in the last pooling operation to perform prediction or classification (see example Figure 3.3). Training a CNN translates into updating the filters weights during backpropagation for all layers [10].

Another reason to use a CNN is that the number of parameters to learn is greatly reduced compared to NN. An image of 256x256 pixels can be seen as a vector of 65 536 values, and a single layer with 100 nodes in a fully connected network would require learning 65 536 * 100 weights. For a CNN using 100 filters of size 3 x 3, it is only necessary to learn 900 weights + 100 bias terms.

3.1.2. Generative Adversarial Networks

Machine Learning algorithms take data as input and then they perform a task which is among classification, regression or clustering. There are two different kinds of approaches used to face those tasks: a *Generative approach* and a *Discriminative*

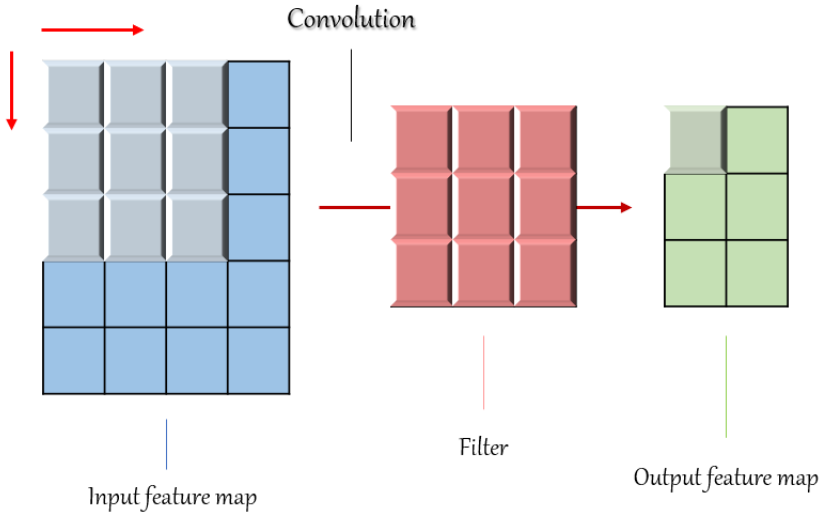

Figure 3.2.: Example of how convolution is performed in a CNN. A filter of 3x3 slides over an input feature map of dimensions 5x4, with stride equal to 1, generating an output feature map of 3x2.

approach.

Discriminative models map features into labels, trying to understand where data belongs to from its main characteristics. Generative models are their opposite, they try to predict features given a label. In more probabilistic terms, while Discriminative models learn the boundary between classes, Generative models model the distribution of individual classes.

For example, in a classification problem, let x be the input (observable variable) and y be the label (target variable), a generative classifier learns a model of the joint probability $p(x, y)$ and makes its prediction using Bayes rules to calculate the conditional probability $p(y \mid x)$ and then picking the most likely label y; a discriminant classifier models the posterior $p(y \mid x)$ or learns a direct map from x to the class labels [19].

Generative Adversarial Network (GAN) is a deep neural network architecture made of two (convolutional) networks "competing" with each other (hence the name "adversarial") and trained simultaneously: a generative model (*Generator*) and a discriminative model (*Discriminator*). The Generator G aims to capture the data distribution while the Discriminator D estimates the probability that a sample came from the training data or from G (see Figure 3.4) [11]. D is trained such that it learns how to assign the correct label to both training data and samples from G , while G is trained such that it creates images whom D can not distinguish from the real ones.

In more theoretical terms, let p_g be the generator distribution, x (real images) the data and $p_z(z)$ a prior distribution on input noise variables z, then $G(z; \theta_g)$ is the

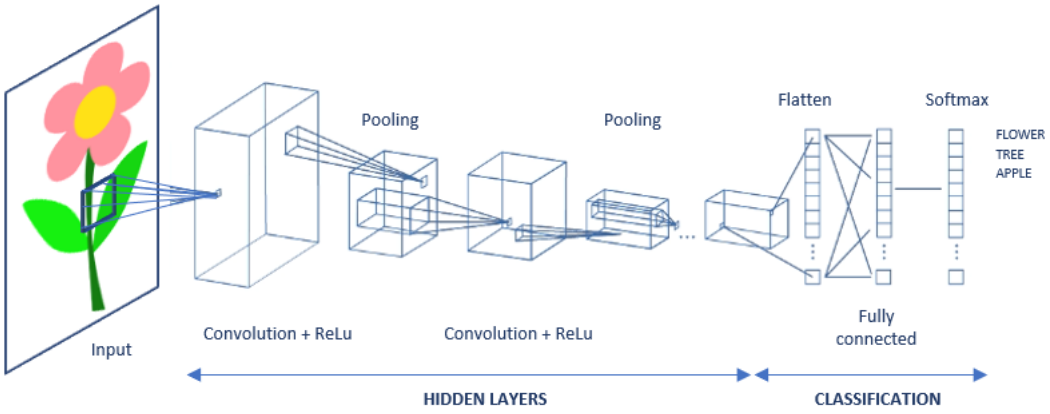

Figure 3.3.: Example of Convolutional Neural Network architecture

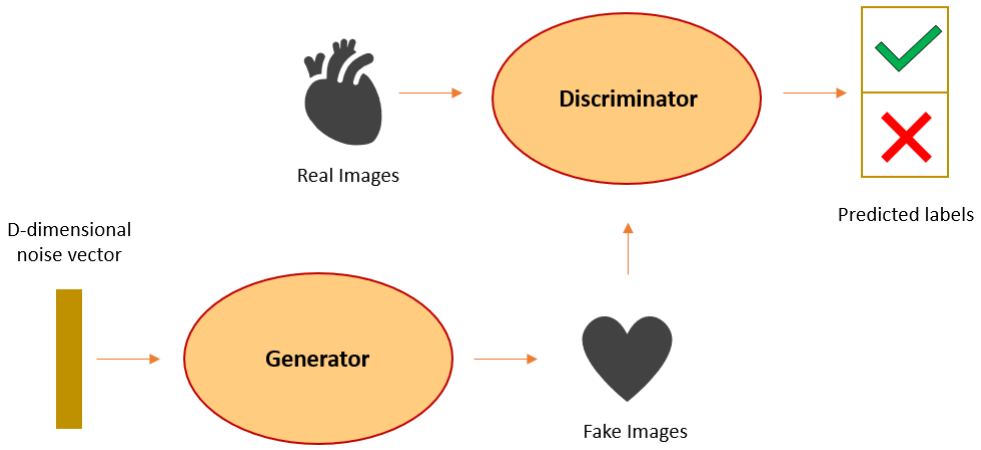

Figure 3.4.: Generative Adversarial Network (GAN) architecture.

differential mapping function to the data space of the fake images, $D(x;\theta_d)$ is the mapping function to the data space of the predicted labels (see Figure 3.4), while θ_g and θ_d are G and D's respective hyperparameters. G and D are trained to learn p_g over the data so the goal turns into solving the following optimization problem:

$$\min_G \max_D L_{GAN}(G, D, Z, X) = \min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.1}$$

Where $D(x)$ represents the probability that x came from the data rather than from p_g [11].

L_{GAN} in equation 3.1 is the *adversarial loss,* so called because G aims to minimize this objective against an adversary D which tries to maximize it. The approach to

solve the min-max problem is iterative and numerical (see Algorithm 3.1). It was proved that the algorithm converges to a global optimum for $p_g = p_{data}$ [11].

Algorithm 3.1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter [11] .

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

end for

- Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

end for
The gradient-based updates can use any standard gradient-based learning rule.

GANs have been successful in generating images within many vision, graphic and medical imaging problems [25, 15]. One example is image to image translation with its wide number of applications such as style transfer. In those kinds of applications, GANs are used in a conditional setting. This means that generating an output image is done conditioning on an input image (cGANs) [14]. This problem can be faced in a supervised or unsupervised way, depending on the type of data available: paired training data or unpaired training data. To summarize the difference between the approaches, paired data are such that $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ while unpaired data are such that $\{x^{(i)}\}_{i=1}^{N}$ with $x^{(i)} \in X$ and $\{y^{(j)}\}_{j=1}^{M}$ with $y^{(j)} \in Y$.

In this work, due to the lack of paired images, an unsupervised approach is therefore needed. Unpaired Image-to-Image Translation methods using GANs, even if they share the same goal, mainly differ in objective functions and/or architecture according to the task they are solving. The leading idea is to map images from one domain X to another domain Y, and vice versa, using two generator and discriminator pairs ((G_x , D_x) and (G_y , D_y)) instead of one, as shown in Figure 3.5.

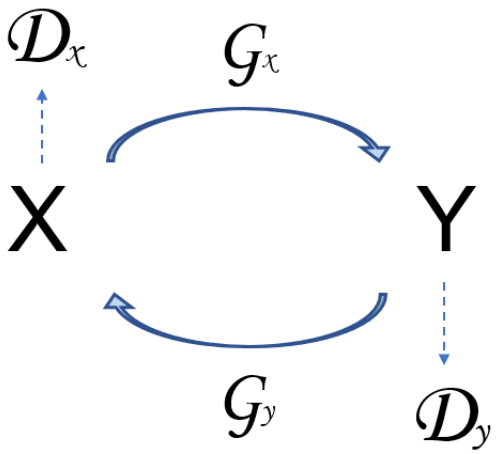

Figure 3.5.: Image to Image Translation. The model has two image domains X and Y, two mapping functions G_x: X $\rightarrow$ Y and G_y: Y $\rightarrow$ X and two adversarial discriminators D_X and D_Y. A conditional GAN thereby contains four CNNs (two generators and two discriminators), while a normal GAN contains two CNNs (a generator and a discriminator).

3.1.3. CycleGAN

CycleGAN [26] is one of the most popular GANs for image to image translation. What CycleGAN model focuses on is trying to preserve similar structure between generated images and the target domain. The objective function is made of two terms: *adversarial losses* (typical of GANs) which match the distributions of generated images to the data distribution in the target domain, and *cycle consistency loss* (hence the name CycleGAN) which prevents the learned mappings G_x and G_y from contradicting each other [26] .

Assuring the cycle-consistency means "following" the entire cycle from $x \in X$ to $G_x(x)$ and back to $G_y(G_x(x)) = \hat{x} \approx x$ (*forward cycle consistency*) and from $y \in Y$ to $G_y(y)$ and back to $G_x(G_y(y)) = \hat{y} \approx y$ (*backward cycle consistency*) for each image x and y in domains X and Y respectively, to induce the learned distribution to match the target one.

The cycle consistency loss is defined as:

$$L_{cyc}(G_x, G_y) = \mathbb{E}_{x \sim p_{data}(x)}[\|G_y(G_x(x)) - x\|_{L_1}] + \mathbb{E}_{y \sim p_{data}(y)}[\|G_x(G_y(y)) - y\|_{L_1}]. \quad (3.2)$$

The full objective function the network optimizes is:

$$\begin{aligned} &L_{CycleGAN}(G_x, G_y, D_X, D_Y) = \\ &\quad = L_{GAN}(G_x, D_Y, X, Y) + L_{GAN}(G_y, D_X, Y, X) + \lambda_{cyc} L_{cyc}(G_x, G_y) \end{aligned} \quad (3.3)$$

where $L_{GAN}(G_x, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G_x(x)))]$, $L_{GAN}(G_y, D_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - D_X(G_y(y)))]$ are the adversarial losses and λ_{cyc} controls the relative importance of the cycle consistency loss [26]. The optimal mapping functions are such that:

$$G_x^*, G_y^* = \arg \min_{G_x, G_y} \max_{D_X, D_Y} L_{CycleGAN}(G_x, G_y, D_X, D_Y). \tag{3.4}$$

3.1.3.1. Implementation

The network architecture for CycleGANs used in this work is inspired by [26] which showed impressive results in many applications of style transfer where paired data where not available. Two generator networks and two discriminator networks are needed for the implementation.

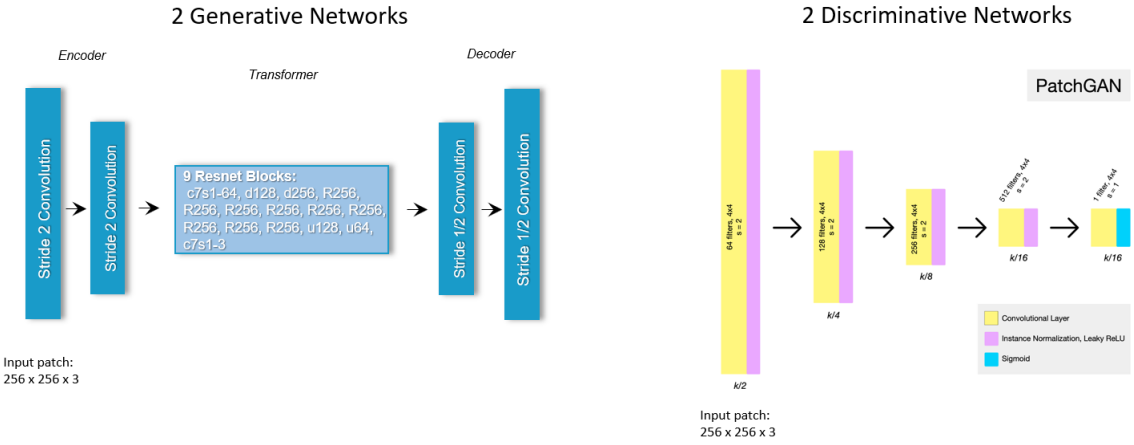

Figure 3.6.: CycleGAN architecture, consisting of two generators and two discriminators, which are trained together.

The two Generative Networks are made of three blocks: an *encoder* which extracts features from an image, a *transformer*[1] which creates the vector of features of the output image and a *decoder* which generates the output image from a feature vector (see Figure 3.6). The two Discriminative Networks are simply binary classifiers with four convolutional layers which work on image at a scale of patches, hence the name *PatchGAN* [14]. To solve the optimization problem, ADAM optimizer was used [26].

More information about the architecture and the implementation is in Appendix A. A working implementation was used in this master thesis, since the goal is to evaluate how GANs can improve segmentation.

[1]More architecture details can be found at https://github.com//Adi-iitd/AI-Art

3.1.4. CycleGAN + Kullback-Leibler divergence

Kullback-Leibler divergence (KL) is a non-symmetric measure of the difference between two probability distributions $p(x)$ and $q(x)$ over the same discrete random variable x [13]. $D_{KL}(p(x), q(x))$ is defined as: $D_{KL}(p(x)\|q(x)) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}$ and measures the amount of information lost when $q(x)$ is used to approximate $p(x)$. In this definition $p(x)$ represents the "true" distribution of data while $q(x)$ is the approximation of $p(x)$.

The *cycle consistency loss* in 3.2 is calculated using the Manhattan distance between the two image domains, this means that a pixel to pixel comparison is computed. L_1 is a very powerful metric for similarity, but because it summarizes into the summation of the pixel-wise intensity differences, small deformation may result in large distances.

In CycleGANs, KL divergence can be added to equation 3.3, representing the loss of information encountered when the normalized gray-scale histogram of the image generated by the mapping function is used to approximate the target domain. The gray-scale histogram of an image refers to a histogram of the pixel intensity values where the possible values go from 0 (representing black) to 255 (representing white). Normalizing this histogram consists of transforming the distribution of intensities into a discrete distribution of probabilities.

For example, consider a digital image of dimensions 256x256 pixels in gray-scale, let n be a vector of length 255 representing the frequencies of each pixel intensity, r_k the number of pixels with intensity equal to n_k, then for $k \in [0, 255]$, $p(r_k) = \frac{n_k}{256*256}$ is the discrete distribution of probabilities of the gray-scale image. The number of bins can also be lower than the number of possible intensity values, in this case pixels with intensity values in a certain range are summed up together.

One limitation of KL divergence is encountered when one event e is possible for the target distribution ($p(e) > 0$) but it is impossible for the approximated distribution ($q(e) = 0$), in this case $D_{KL}(p(e), q(e)) = \inf$. One easy way to overcome this problem is computing KL Divergence by Smoothing the discrete distributions of probability so that there are not zero values of probabilities [12].

The KL divergence Loss is then defined as:

$$L_{KL}(G_x, G_y) = \underset{x \sim p_{data}(x)}{D_{KL}}(p(x), p(G_y(G_x(x)))) + \underset{y \sim p_{data}(y)}{D_{KL}}(p(y), p(G_x(G_y(y)))) \quad (3.5)$$

and the full objective function the network optimizes:

$$L_{CycleGAN+KL}(G_x, G_y, D_X, D_Y) = L_{CycleGAN}(G_x, G_y, D_X, D_Y) + \lambda_{KL} L_{KL}(G_x, G_y)$$

(3.6)

Where λ_{KL} controls the relative importance of the KL loss.

The optimal mapping functions are:

$$G_x^*, G_y^* = \arg \min_{G_x, G_y} \max_{D_X, D_Y} L_{CycleGAN+KL}(G_x, G_y, D_X, D_Y). \tag{3.7}$$

3.1.5. UNIT

Another popular architecture for unsupervised Image to Image translation is UNIT [17], which stands for UNsupervised Image-to-image Translation. As CycleGAN, UNIT also aims to map images from one domain to another using only unpaired data, but with the difference of trying to overcome this difficulty introducing a shared-latent space assumption. UNIT goal is to learn the joint distribution $p(x, y)$, where X and Y are two image domains, given their marginal distributions $p(x)$ and $p(y)$. The assumption is that for each pair of images (x, y), where $x \in X$ and $y \in Y$, there exists a "code" $z \in Z$ such that both images can be recovered from this space [17].

In more details, beside the Generators there exists two other functions called Encoders E_X and E_Y, such that $z = E_x(x) = E_y(y)$ and $x = G_x(z)$, $y = G_y(z)$. The mapping functions to be learnt by the model are $F_{x \to y} = G_y(E_x(x))$ and $F_{y \to x} = G_x(E_y(y))$, shown in Figure 3.7, where the arrow "$\to$" indicates the direction of the mapping function which will perform image to image translation (i.e. $x \to y$ shows that an image $x \in X$ is translated into an image in Y domain).

The architecture of this model is based on GANs, with the difference that also variational autoencoders (*VAEs*) are used for each of the two encoder-generator pairs. Autoencoders are a type of NN which try to learn the representation of the data by compressing it into a compact representation, and uncompressing the representation such that the ouput matches the input data. Variational autoencoders are based on the autoencoders structure with the following assumptions:

1. The data is generated by a directed graphical model $p(x|z)$;
2. The encoder aims to learn an approximation $q_\phi(z|x)$ to the posterior distribution $p_\theta(x|z)$ where ϕ is the parameter of the encoder and θ the one of the decoder;
3. The prior over the latent variables is a multivariate Gaussian $p_\theta(z) \sim N(0, I)$.

The objective functions of the *VAEs* for UNIT are defined as:

$$L_{VAE_x}(E_x, G_x) = \lambda_1 D_{KL}(q_x(z_x|x), p_\theta(z)) - \lambda_2 \mathbb{E}_{z_x \sim q_x(z_x|x)}[\log p_{G_x}(x|z_x)] \tag{3.8}$$

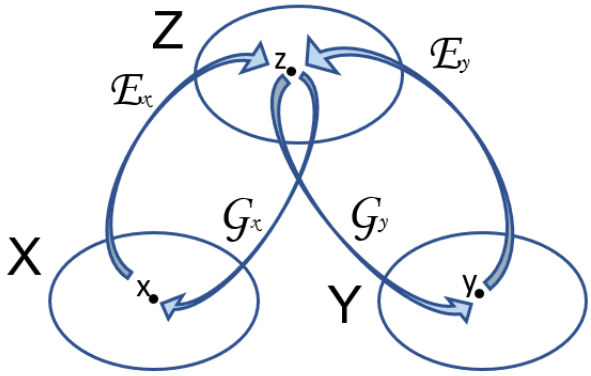

Figure 3.7.: UNIT: X and Y are the two image domains, Z is the latent space which contains latent representations which pairs of corresponding images from X and Y can be mapped to.

$$L_{VAE_y}(E_y, G_y) = \lambda_1 D_{KL}(q_y(z_y|y), p_\theta(z)) - \lambda_2 \mathbb{E}_{z_y \sim q_y(z_y|x)}[\log p_{G_y}(y|z_y)] \quad (3.9)$$

where $q_x(z_x|x) \sim N(z_x|E_{\mu,1}(x), I)$, $q_y(z_y|y) \sim N(z_y|E_{\mu,1}(y), I)$ with $E_{\mu,1}(x)$ and $E_{\mu,1}(y)$ being the mean vectors output from the encoders and I the identity matrix. D_{KL} is KULLBACK-LEIBLER DIVERGENCE, while p_{G_x} and p_{G_y} are modeled using Laplacian distributions [17]. VAE loss aims to adapt the latent space according to the image domains, minimizing the distance between probability distributions. KL divergence term is a measure of how the distribution of domain specific code, z_x or z_y, diverges from the prior distribution $p_\theta(z)$. In the second term, minimizing the negative log-likelihood term is equivalent to minimize the absolute distance between the image and the reconstructed image [17].

The GAN objective functions in this case are formulated as follows:

$$L_{GAN_x}(E_y, G_x, D_x) = \lambda_0 \mathbb{E}_{x \sim p_{data}(x)}[\log D_X(x)] + \lambda_0 \mathbb{E}_{y \sim q_y(z_y|y)}[\log(1 - D_X(G_x(z_y)))] \quad (3.10)$$

$$L_{GAN_y}(E_x, G_y, D_y) = \lambda_0 \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \lambda_0 \mathbb{E}_{x \sim q_x(z_x|x)}[\log(1 - D_Y(G_y(z_x)))] \quad (3.11)$$

Besides these objective functions, also the *cycle-consistency constraint* is used to make sure that there is consistency in passing from one distribution to the other one. It is defined as:

$$L_{CC_x}(E_x, G_x, E_y, G_y) = \lambda_3 D_{KL}(q_x(z_x|x), p_\theta(z)) + \lambda_3 D_{KL}(q_y(z_y|x^{x\to y}), p_\theta(z)) + \\ -\lambda_4 \mathbb{E}_{z_y \sim q_y(z_y|x^{x\to y})}[\log p_{G_x}(x|z_y)] \tag{3.12}$$

$$L_{CC_y}(E_y, G_y, E_x, G_x) = \lambda_3 D_{KL}(q_y(z_y|y), p_\theta(z)) + \lambda_3 D_{KL}(q_x(z_x|y^{y\to x}), p_\theta(z)) + \\ -\lambda_4 \mathbb{E}_{z_x \sim q_x(z_x|y^{y\to x})}[\log p_{G_y}(y|z_x)] \tag{3.13}$$

where $x^{x\to y} = G_y(z_x \sim q_x(z_x|x))$ and $y^{y\to x} = G_x(z_y \sim q_y(z_y|y))$, with $x^{x\to y} \in Y$, $y^{y\to x} \in X$ indicating the translated images.

The *cycle-consistency constraint* has the same purpose of the one from CycleGAN, with some adjustments to the framework.

The learning problem to be solved in UNIT can be summarized into the following:

$$E_x^*, E_y^*, G_x^*, G_y^* = \arg \min_{E_x, E_y, G_x, G_y} \max_{D_X, D_Y} L_{VAE_x}(E_x, G_x) + L_{VAE_y}(E_y, G_y) + \\ + L_{GAN_x}(E_y, G_x, D_X) + L_{GAN_y}(E_x, G_y, D_Y) + \\ + L_{CC_x}(E_x, G_x, E_y, G_y) + L_{CC_y}(E_y, G_y, E_x, G_x). \tag{3.14}$$

Where $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyperparameters controlling the impact of the many objective terms in final objective function.

3.1.5.1. Implementation

The network architecture for UNIT used in this work is inspired by [17]. The network architecture is made of a total of six subnetworks: two generator networks, two discriminator networks and two encoders networks. The two encoder networks consist of three convolutional layers and four basic residual blocks; the Generative Networks are made of three generator residual blocks and three deconvolutional layers as decoder; the two Discriminator Networks consist of six convolutional layers (see Figure 3.8). To solve the optimization problem, the ADAM optimizer was used [17].

More information about the architecture and the implementation are in Appendix A.

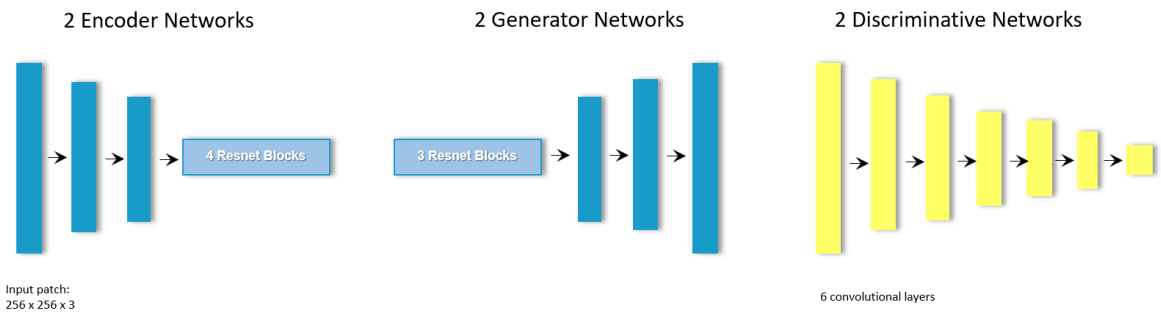

Figure 3.8.: UNIT architecture, consisting of two encoders, two generators and two discriminators.

3.2. Evaluation methods

As this thesis objective is to enhance performance of the company tool, the evaluation of the results does not focus mainly on the quality of the images generated by GANs, but on the predictions obtained using them as input in the segmentation model. Gray-scale prediction patches are first converted into binary images and then evaluated against corresponding patches of the binary ground truth images (Figure 2.2).

Thresholding, in image processing, is the simplest method for image segmentation and the method adopted to convert a gray-scale image into a binary image. Each pixel is replaced according to the following rule:

$$I_{i,j} = \begin{cases} 0 & I_{i,j} < T \\ 1 & I_{i,j} \geq T \end{cases} \tag{3.15}$$

where $T \in [0, 255]$ is the threshold and $(i, j) \in [0, height]$x$[0, width]$ where $height$ x $width$ being the image (I) size. As a result, the problem can be performed as a binary classification.

3.2.1. Measures of performance for classification methods

Binary classification problems involve classifying data into two classes (i.e. Positive and Negative) which represent the possible outcome of an algorithm. There are plenty of methods to measure performance in classification, both numerical and graphical. First, the calculation of a metric called CONFUSION MATRIX is required. It compares predicted classes with true classes (see Table 3.1) showing how many examples are correctly classified, *True Positive (TP)* and *True Negative* (TN), and

how many are misclassified, *False Positive (FP)* and *False Negative (FN)*. A *False Negative* is also called *Type II error*, while a *False Positive* is also called *Type I error.* A confusion matrix is the base for the calculation of all other performance measures.

		True classes	
		Positive	Negative
Predicted classes	Positive	*True Positive*	*False Positive*
	Negative	*False Negative*	*True Negative*

Table 3.1.: Confusion Matrix for a binary classifier.

Accuracy is one of the most used measures for classification performance, defined as the ratio between correctly classified samples and the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.16}$$

However, accuracy does not take into account how data is spread between TP and TN, resulting in not very accurate estimations for samples where classes are not balanced.

Two metrics, which are less sensitive to this problem, are Sensitivity (also called True Positive Rate or Recall), representing the ratio between the positive correctly classified samples and the total number of positive samples, and Specificity (also called True Negative Rate), the ratio of correctly classified negative samples and total number of negative samples:

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{3.17}$$

$$Specificity = \frac{TN}{FP + TN} = \frac{TN}{N} \tag{3.18}$$

Thus, Specificity represents the proportion of correctly classified negative samples, while Sensitivity is the proportion of correctly classified positive samples.

A measure which reflects the positive predictive value is Precision, defined as the proportion of correctly classified positive samples to the total number of positive predicted samples:

$$Precision = \frac{TP}{FP + TP} \tag{3.19}$$

F1 SCORE is the harmonic mean of PRECISION and RECALL, interpreted as their weighted average, and is ranged between 0 and 1. High values of F1 SCORE indicate high classification performance.

$$F1 = \frac{2 * (Recall * Precision)}{Recall + Precision} \tag{3.20}$$

F1 score takes into account both false positive and false negative, that is why it is widely used instead of accuracy to evaluate how good a model is, especially with uneven class distribution [1].

One common way to evaluate decision making systems or machine learning systems is the ROC CURVE (RECEIVER OPERATING CHARACTERISTICS CURVE). The ROC curve offers a graphical illustration of the trade-off between a test sensitivity and specificity and depicts TP rate (on the y-axis) against FP rate (on the x-axis), for each threshold value. Each threshold generates only one point in the ROC curve. The lower left corner of the curve, (0,0), represents a classifier where there is no positive classification and all negative samples are correctly classified; the lower right corner, (1,0), represents a classifier where all positive samples are correctly classified and the negative samples are misclassified. The perfect classifier is represented by that point in the ROC space where all positive and negative samples are correctly classified in the upper left corner (0,1), that is why this point is called *Ideal operating point* [1].

Comparing different classifiers using their ROC curves can be performed calculating the AREA UNDER THE CURVE (AUC) metric, which is a value bounded between 0 and 1, where 1 represents the optimum value. Given two classifiers A and B, for example, A is said to achieve better performance than B if $AUC_A > AUC_B$.

Another curve, used to compared different classifiers, is PRECISION-RECALL CURVE (PR CURVE). As ROC, PR CURVE is also calculated across different threshold values. In this case, the relationship between precision (on the y-axis) and recall (on the x-axis) is showed instead. Given two different classifiers, the one with better classification performance generates a curve which is the closest to the upper right corner. A drawback of PR curve is that it completely ignores the performance of correctly handling negative examples (TN) [1].

3.2.2. Image Similarity Measures

Image similarity is the measure of how similar two images are, in this context it helps to measure how similar predictions and corresponding ground truth patches are.

Black and white images can be seen as matrices where each location is represented by a pixel containing 0 (black) or 1 (white). The most traditional and simple method to measure distances between two images I and J, both with size $M * N$, is calculating the Mean Square Error (MSE). MSE is a pixel-based metrics, calculating the mean square error between each pixels for the two images I and J:

$$MSE(I,J) = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}|I(i,j) - J(i,j)|^2. \tag{3.21}$$

According to this measure, the higher the similarity, the lower the MSE is. One of the biggest disadvantages of MSE is being poorly correlated with human perception of visual system [22]. For example, given three images I , J and K, with $MSE(I,J) = MSE(I,K)$ does not always imply that I and K are similar.

To overcome this problem and to extract structural information from images, in other words to extract the inter-dependencies that are present in spatially close area, a more qualitative metric is used, the Structural similarity Metric:

$$SSIM(I,J) = \frac{(2\mu_I\mu_J + c_1)(2\sigma_{IJ} + c_2)}{(\mu_I^2 + \mu_J^2 + c_1)(\sigma_I^2 + \sigma_J^2 + c_2)} \tag{3.22}$$

with:

- $\mu_I = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)$;
- $\mu_J = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} J(i,j)$;
- $\sigma_I^2 = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - \mu_I)^2$;
- $\sigma_J^2 = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(J(i,j) - \mu_J)^2$;
- $\sigma_{IJ} = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - \mu_I)(J(i,j) - \mu_J)$;
- $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ two variables to avoid instability in cases when $\mu_I^2 + \mu_J^2$ is too close to zero;
- L is dynamic range of pixel value $(2^{\#bits\,per\,pixel} - 1)$ [22];
- k_1 and k_2 are two small constants.

Lower and upper bounds for SSIM are -1 and 1 and value 1 is reachable in case of identical images with perfect structural similarity [22].

While MSE estimates absolute errors, SSIM is a perception-based model which perceives changes in structural similarity.

To calculate similarities between segmented images two other measures are of interest: Pixel Accuracy and Mean Intersection Over Union:

$$Pixel\,Accuracy = \frac{\#correctly\,classified\,pixels}{M*N} \tag{3.23}$$

$$Mean\ IoU = \frac{1}{\#\,classes} \sum_{k=1}^{\#classes} \frac{TP_k}{TP_k + FN_k + FP_k} \tag{3.24}$$

Mean Intersection Over Union (Mean IoU or Mean IU) metric quantifies the percentage of overlapping between the target image and the prediction separately, and then averages over all classes to provide a global score for the semantic segmentation [2] prediction. Pixel Accuracy, instead, only reports the percentage of correctly classified pixels with no distinction of classes. Both measures are between 0 and 1, with 1 representing the highest similarity value.

3.2.3. Statistical tests to compare two paired samples

For each patch a new one is generated by GANs and, from both, segmentation tool predictions are obtained as paired images. Enhancing (or reducing) the performance of a segmentation tool can be seen as obtaining generated images predictions significantly more (or less) similar to ground truth patches, compared to how the original images predictions are.

Two paired groups of samples can be calculated measuring similarities between *ground truth* patches and *original images predictions* (s_1), and *ground truth* patches and *generated images predictions* (s_2) (see Figure 4.1). Data values from s_1 and s_2 are not independent, because both are obtained comparing images with the same set of ground truth patches. To establish if s_1 and s_2 are significantly different, the difference between their mean value is tested with two different statistical approaches: parametric or nonparametric [20]. One of the main differences is that, while in the first case several assumptions about the parameters of the population distribution, from which the sample is drawn, need to be made, in the second case fewer assumptions are necessary.

3.2.3.1. Parametric Test: Paired t-test

Student t-test is a statistical test which is used to compare the mean value of two groups of samples. The question this test aims to answer is: *are the means of the two sets significantly different from each other?*

More in particular, paired t-test is used to compare the means between two related groups of samples s_1 and s_2 (i.e to compare values of blood pressure before and after a treatment), when data values are in a pairing [20].

[2] Semantic segmentation describes the process of associating each pixel of an image with a class label.

Let $\mathbf{d} = s_1 - s_2$ and m_d be the sample mean of $\mathbf{d}$, s_d its sample standard deviation and n its size. Assuming that $d_1, \ldots, d_n$ constitute a sample from a normal population $\mathcal{N}(\mu_d, \sigma_d^2)$ having unknown mean μ_d and unknown standard deviation σ_d. Saying that there is no difference between the two paired groups translates into the statement that $\mu_d = 0$, so the hypothesis to test is:

$$\begin{aligned} H_0 &: \mu_d = 0 \\ H_1 &: \mu_d \neq 0 \end{aligned} \tag{3.25}$$

The null hypothesis H_0 can be rejected when the estimator of the population mean μ_d, represented by the sample mean m_d, is far from 0. Estimating the unknown standard deviation with the sample standard deviation and setting a significance level α, H_0 is rejected, in favor of the alternative hypothesis, if the *p-value* of the **t statistic value** $t = \frac{m_d}{s_d} * \sqrt{n}$, with **degrees freedom** $df = n - 1$, is less than half the chosen significance level α (two-sided test). The *p-value* represents the risk indicated by the t-test table for the calculated *t value [20].*

A paired t-test needs to satisfy the following assumptions:

1. The data are continuous;
2. The differences for the matched-pairs follow a normal probability distribution;
3. The sample of pairs is a random sample from its population.

It has been proved that paired t-test is robust to violation of the normality assumption of the differences in the samples, when some conditions hold, such as sample size is 25 or more per group [8]. In case of large samples, then, the normality assumption does not need to be tested.

3.2.3.2. Nonparametric Test: One-sample Permutation Test

Hypotheses tests can also be used in situations where the underlying distribution of the data is not required to have any particular form. Because the validity of these tests does not rest on the assumption of any particular parametric form (such as normality) for the underlying distribution, these tests are called nonparametric [20]. PERMUTATION TESTS are a class of nonparametric tests which test the hypothesis for which relabeling the observed data is justified by exchangeability of the observed random variables .

In case of paired data, $\mathbf{d} = s_1 - s_2$ is calculated. The permutation test is based on the idea that under the null hypothesis, d_i, with $i \in [1, n]$, is symmetric around the mean value m_d. Under H_0, d_i is equally likely to be lower or higher in value than m_d. Let $Z_i = +1$ or -1 with probability $\frac{1}{2}$ for each observation d_i. Calculating the mean on the 2^n possibilities of sign vector Z, or on a fixed number of sign flips, multiplied

by the observations $\mathbf{d}$ ($\mathbf{d}*\mathbf{Z_j}$ with $j \leq 2^n$), allows to generate a *conditional empirical null distribution* of the test statistic [18]. The hypothesis to be tested can be then translated into:

$$\begin{aligned} H_0 &: \mu_d = m_d \\ H_1 &: \mu_d \neq m_d \end{aligned} \tag{3.26}$$

where the null hypothesis H_0 can be rejected if the p-value of the test is less than the chosen significance level α. The p value is defined as the percentage of test statistic values x such that $|x - m_d| \geq 0$ in the *conditional empirical null distribution*.

For example, let $X = \{0.5, 0.4, 0.3\}$ and $Y = \{0.2, 0.8, 0.1\}$ be the values of SSIM between three ground truth images and three patches before (X) and after (Y) style transfer. Let $D = X - Y = \{0.3, -0.4, 0.2\}$ and $m_d = \frac{1}{3}(0.3 - 0.4 + 0.2) = 0.03$. Choosing 4 as fixed number of sign flips, four possible outcomes of sign vector Z are shown in Table 3.2. The test p-value is 1, because all four trials give as results a value which is lower or equal to $-m_d$ and higher or equal to m_d . For a significance level $\alpha = 0.05$ the null hypothesis can not be rejected, resulting in no significant improvement, in SSIM, after style transfer.

	D	$D * Z_1$	$D * Z_2$	$D * Z_3$	$D * Z_4$
	0.3	0.3	-0.3	0.3	-0.3
	-0.4	0.4	-0.4	0.4	0.4
	0.2	-0.2	-0.2	0.2	-0.2
mean value	0.03	0.17	-0.3	0.3	-0.03

Table 3.2.: Example of sign flipping nonparametric test. D represents the difference between two paired samples X and Y. From the second to the fifth column, sign flipping is performed four times. The last row, *mean value*, indicates the value of the test statistic calculated for each column.

The p-value gives the probability of observing the test results under the null hypothesis. The lower the p-value, the lower the probability of obtaining a result like the one that was observed if the null hypothesis was true. In both parametric and nonparametric tests, if the alternative hypothesis is an inequality, the test only checks the significance of the difference between distributions (two sided test); a disequality, instead, assesses which sample is significantly better than the other one.

4. Results

In this section, results from some of the most promising style transfer experiments are reported. In the first part the evaluation procedure is described step by step, while in the second part a more detailed report of results is given.

4.1. Evaluation Procedure

The evaluation procedure mainly consists of six different steps:

1. **Style transfer evaluation**: style transfer methods, described in sec. 3.1, are first trained and then tested on LEICA test patches, generating, as result, FAKE ZEISS patches. Initially, a qualitative evaluation is given, comparing the images before and after style transfer. Then, Y, Cb and Cr color channels histograms of the generated and of the real ZEISS images are compared both visually and numerically (KL divergence).
2. **Prediction**: the patches generated applying style transfer to the test set (FAKE ZEISS) with different models and the test patches themselves (LEICA) are all given as input to the segmentation tool for prediction. The tool is pre-trained by ContextVision on WSI ZEISS images from the train set described in chapter 2;
3. **Ground truth patches extraction**: for each of the patches in the test set, a ground truth patch is extracted from the corresponding ground truth image. Each patch is resized (from approximately 20x20 pixels to 256x256 pixels) such that it can be compared to the predictions obtained in the previous step;
4. **Overall evaluation**: ROC, precision vs recall, F1 vs threshold curves and AUC are calculated for both FAKE ZEISS SETS and LEICA TEST SET across different thresholds on the predicted patches. For each model the best threshold is chosen according to the best F1 score value. Confusion matrices and all metrics related to them are reported, to compare models between each other and each model with the baseline represented by predictions on LEICA TEST SET;
5. **Local evaluation**: for each model, and for the baseline test set (LEICA), each patch is compared with its ground truth patch, obtained in step 2, through image similarity measures, generating vectors of measures with the same length of the test set. For each of these vectors some summary statistics are reported;

6. **Statistical analysis for comparison**: step 1-4 help to point out how the models different from each other and from the baseline set (LEICA). To evaluate the significance of these differences, parametric and nonparametric hypothesis testing is used on the paired data vectors obtained in step 4.

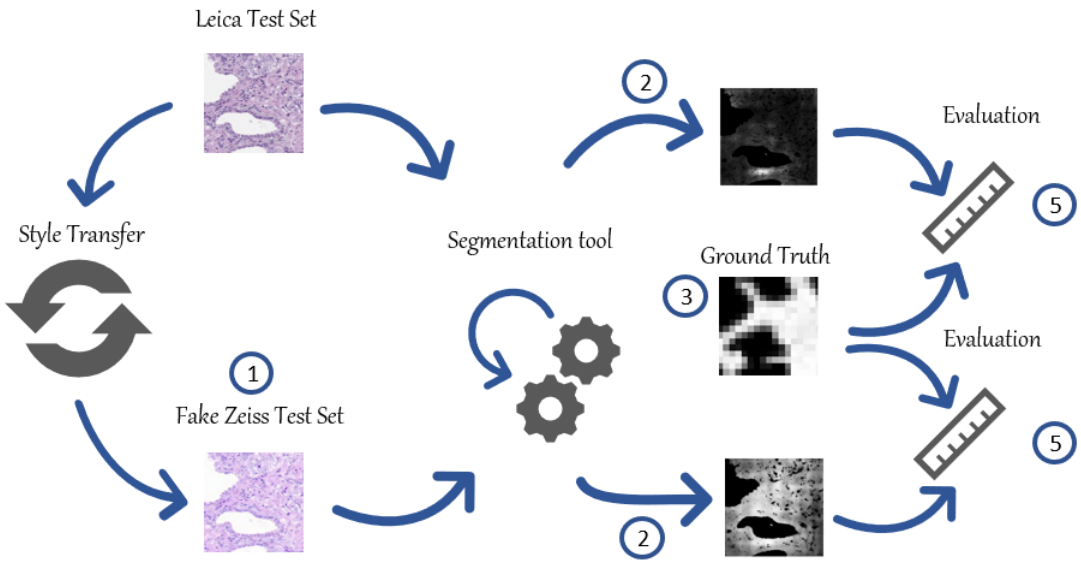

Figure 4.1.: Evaluation procedure shown at a patch level. First, Leica test set (baseline set) is used as input of a style transfer model generating Fake Zeiss test set. In step 1, the generated patches are qualitative evaluated. In step 2 predictions from the segmentation tool are obtained for both Leica test set and Fake Zeiss test set. In step 3 ground truth patches are extracted from ground truth images. Predictions on before and after style transfer patches are compared with ground truth images in an evaluation step, patch by patch, in step 5. Repeating steps 1, 2, 3 and 5 for all patches in Leica Test Set, an overall evaluation can be computed (step 4) and statistical analysis (step 5) can be used to assess improvement in performance of the segmentation tool.

4.2. Simulations Results

Both CycleGAN and UNIT have complex architectures, made of four and six networks respectively, resulting in a large number of hyperparameters. Related works [23, 24, 21] demonstrated the success of the default CycleGAN architecture and settings [26] in histopathology. Because of the long training times those models require and of the limited time available, this work aims to discover how the amount of training data, number of epochs, different loss functions and different GANs influence the performance of the Image to Image translation problem and therefore of the segmentation model.

Different style transfer models are trained on datasets made of randomly selected images from the training patches obtained in chapter 2. Two different datasets are used for training: Dataset 1, made of 40,000 patches, and Dataset 2, made of 4,000 patches. Each of the sets contains half Zeiss and half Leica training patches, in order to "learn" both ways mapping functions. While CycleGAN can be trained on both large and small amount of data, UNIT requires much more computer memory to process large datasets, for this reason, in the experiments, it is only trained on Dataset 2.

After training, all models are tested on one test set of 2,000 images, randomly selected from Leica testing patches in Table 2.1, and are assessed on five sets of 2,000 images, obtained in the same way.

4.2.1. Dataset 1

The following table shows the methods used on Dataset 1 with the number of epochs and the loss functions details:

Methods	# epochs	loss functions
CycleGAN	90	adversarial + cycle consistency
CycleGAN + KL	70	adversarial + cycle consistency + KL divergence

Table 4.1.: Summary of methods used on Dataset 1: models, number of epochs and loss functions.

In generating new patches, no matching images in the target domain are available for comparison so it is hard to judge the performance of the model. Visually it is important to ensure that the patch texture is preserved also in the synthetic image and that the style looks similar to the target one (in Figure 2.3 an idea of the different styles is given).

In Figure 4.2, a few examples of the results obtained applying the above mentioned methods are showed. Texture details are preserved after style transfer and tissue is distinguished from the white background correctly. Both models maintain these characteristics, CycleGAN + KL sharpens the difference between darker and lighter areas compared to CycleGAN.

To have a numerical idea of how similar the generated images and Zeiss images styles are for both models, a comparison between Y, Cb and Cr color histograms is shown in Figure 4.3 and KL divergence (Table 4.2) is calculated per channel, to compare with results from chapter 2. The two channels with the greatest improvement, after style transfer, are Cb and Cr. On the overall patches, CycleGAN+KL seems to learn chrominance better than simple CycleGAN does, especially Cr (confirming

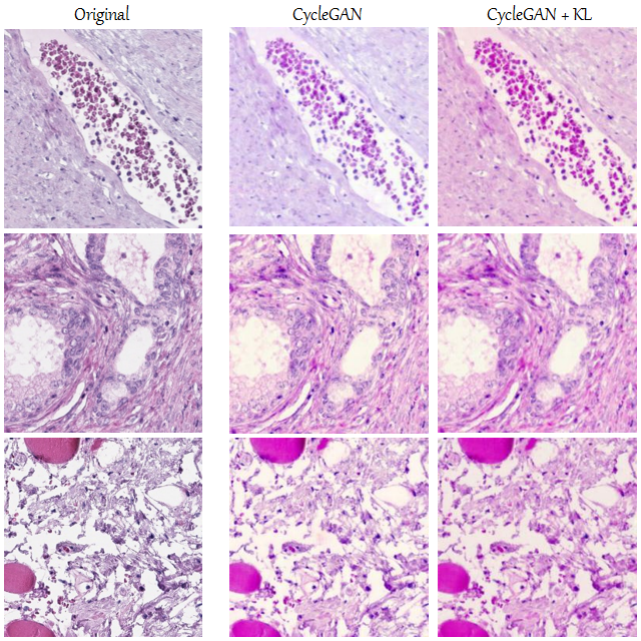

Figure 4.2.: Before and after Style Transfer. The first column on the left contains three Leica original patches from the test set, while the other two columns on the right contain the style transferred images with CycleGAN and CycleGAN+KL respectively.

the sharpness in Figure 4.2). Looking at the gray scale version of the original images, CycleGAN is more capable to represent the target domain (see Y Channel in Table 4.2).

		Y Channel	*Cb Channel*	*Cr Channel*
KL(Zeiss,Fake Zeiss)	CycleGAN	0.11	0.02	0.10
	CycleGAN+KL	0.26	0.03	0.05

Table 4.2.: Kullback-Leibner divergence between Zeiss and Fake Zeiss Y, Cb and Cr color histograms for all methods trained on Dataset 1.

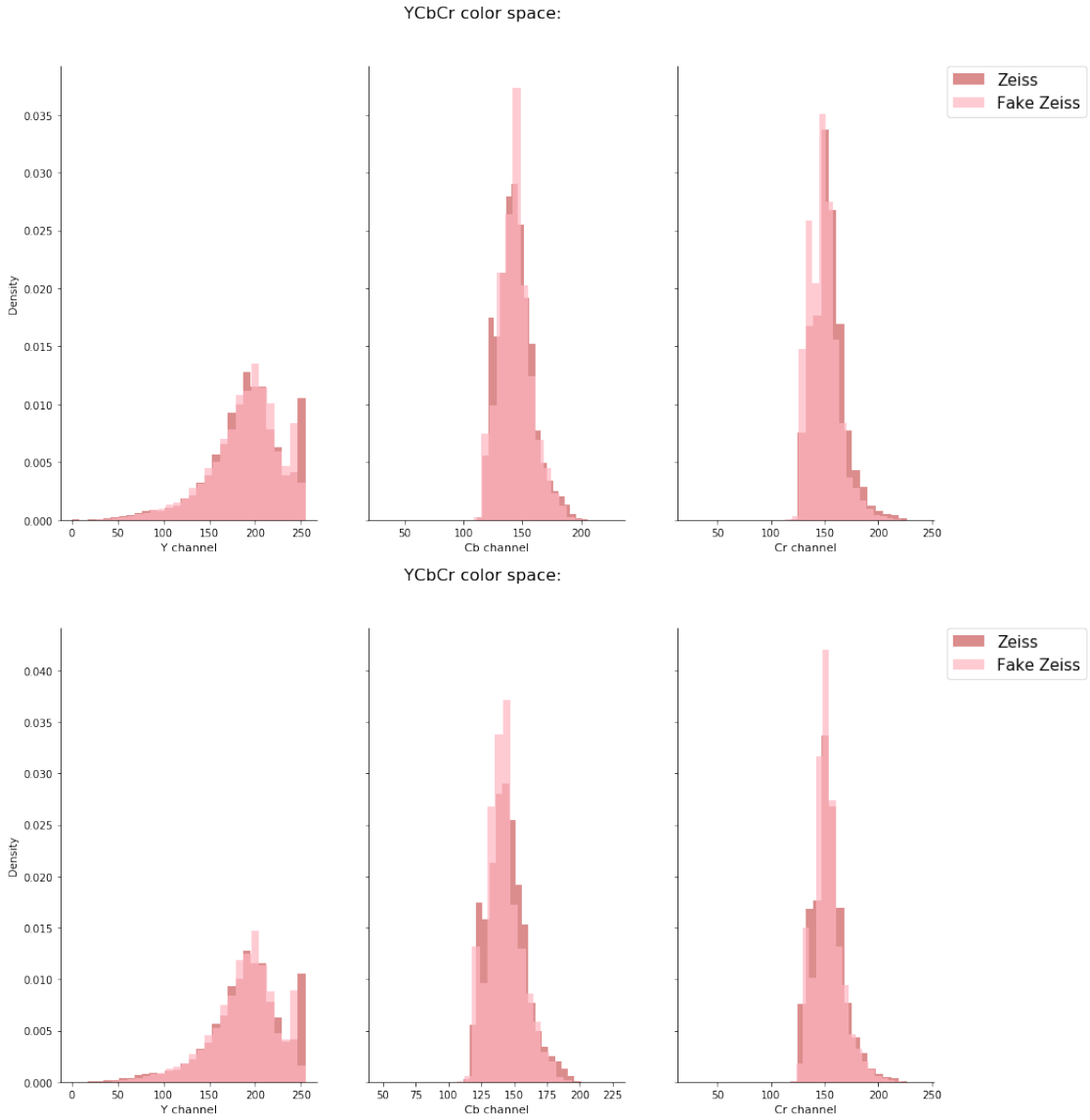

Figure 4.3.: Comparison between Y, Cb and Cr color histograms for Zeiss and Fake Zeiss images: CycleGAN on the top and CycleGAN+KL on the bottom.

4.2.1.1. Measurements of performance of the classifier on different methods

Tumor segmentation predictions are calculated for LEICA TEST SET and for the two FAKE ZEISS SETS obtained by applying the style transfer methods described above. Before choosing the best threshold, to transform gray scale images into binary images, the methods are visually compared across thresholds ($0 \leq t \leq 255$) using ROC curve (Figure 4.4), PR curve, plotting F1 values against thresholds (Figure 4.5) and calculating AUC (Table 4.3).

According to ROC curve and AUC, the performance of the tumor classifier network

increases on stain normalized images with CycleGAN compared to the original Leica images. The ROC curve for the untransferred images is always below the one for CycleGAN and the one for CycleGAN+KL transferred images, and this translates into a lower value of AUC as shown in Table 4.3. CycleGAN outperforms CycleGAN+KL by 3.6% in AUC and increases the original performance by 16% in average AUC.

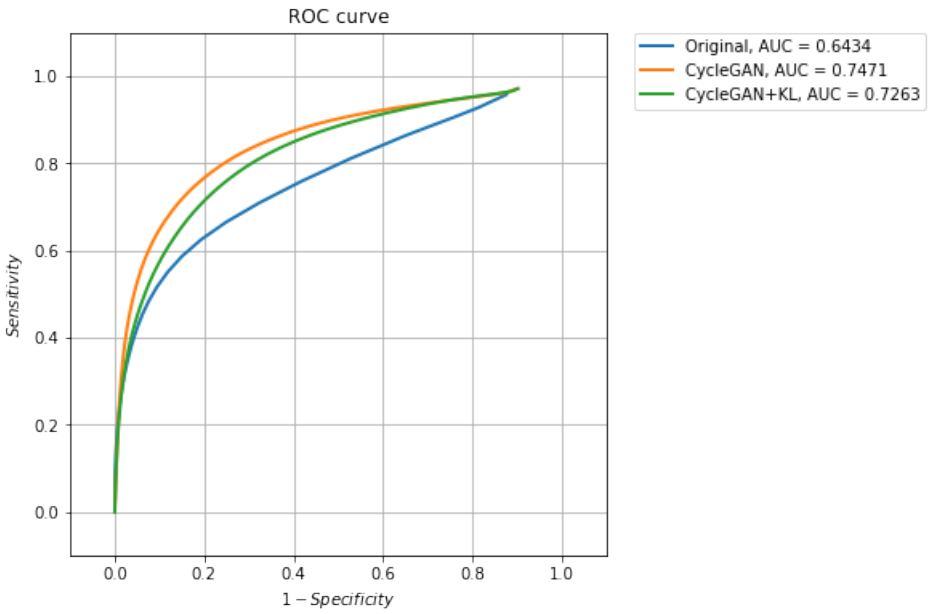

Figure 4.4.: ROC curves generated over predictions calculated on CycleGAN and CycleGAN+KL transferred patches (FAKE ZEISS) and Original untransferred patches (LEICA). The highest value of AUC is obtained with CycleGAN.

METHODS	AUC
ORIGINAL	0.644±0.013
CYCLEGAN	0.747±0.007
CYCLEGAN + KL	0.721±0.008

Table 4.3.: AUC mean values and standard deviation calculated over 6 sets of test data consisting of 2000 patches each, for models trained on Dataset 1.

In Figure 4.5, on the left side, F1 score is plotted against threshold values, while precision and recall are plotted on the right side. For the original images the best value of the F1 score is obtained for threshold $t_{original} = 14$, while for the CycleGAN generated images it is obtained for threshold $t_{CycleGAN} = 119$ and the CycleGAN+KL

generated images for $t_{CycleGAN+KL} = 102$. Knowing that 0 refers to black pixel values and 255 to white pixel values, predictions from original images result to be darker than the ones from generated images. PR curve for the untransferred images outperforms the one for transferred images only for low threshold values, as shown in Figure 4.5, at the same time CycleGAN and CycleGAN+KL seem to have more or less the same behavior, with slightly higher values of precision and recall for the first model, compared to the second.

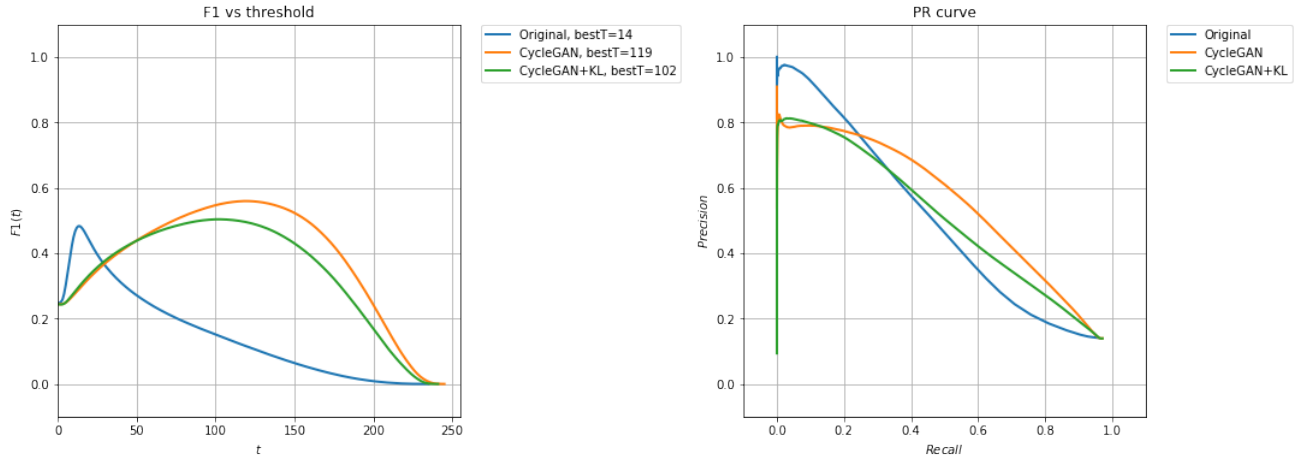

Figure 4.5.: On the left F1 score plotted against different thresholds values, on the right PR curve calculated on results obtained training on Dataset 1.

After finding the best thresholds according to F1 score, thresholding is performed for all patches and confusion matrices are calculated:

Methods	TP	FP	TN	FN
Original	5.78%	5.44%	81.82%	6.96%
CycleGAN	6.86%	5.68%	82.33%	5.14%
CycleGAN + KL	6.37%	6.84%	81.06%	5.73%

Table 4.4.: Confusion matrices for transferred and untransferred patches predictions results in percentage on the overall number of pixels, for models trained on Dataset 1.

CycleGAN increases the number of true positive and true negative but it also results into a bigger number of false positive and lower number of false negative. These observations prove the higher value of accuracy, precision and recall as shown in Table 4.5. CycleGAN+KL fails in detecting true negative compared to the original untransferred images and this affects the accuracy by 1 percentage point, resulting in a higher number of false positive. This model still outperforms the original untransferred images in terms of true positive but not as much as CycleGAN does.

Methods	Accuracy	Precision	Recall	F1
Original	0.874±0.004	0.47±0.02	0.47±0.03	0.47±0.01
CycleGAN	0.895±0.002	0.53±0.01	0.57±0.01	0.55±0.01
CycleGAN + KL	0.875±0.003	0.46±0.02	0.51±0.02	0.48±0.02

Table 4.5.: Performance of segmentation tool on images before (Original) and after (CycleGAN, CycleGAN+KL) style transfer.

4.2.1.2. Image Similarity Measures on different methods

After evaluating the models considering all the patches as an overall image, a more local analysis is done. SSIM, pixel accuracy, Mean Intersection Over Union and MSE are calculated between each prediction and the corresponding ground truth patch for the untransferred set and the transferred set with CycleGAN and CycleGAN+KL respectively (see Figure 4.6). In Table 4.6 the mean and standard deviation values are reported for all image similarity measures showing an overall improvement of the performance of the segmentation tool on transferred images. CycleGAN shows improvement for all measures, beside that, standard deviation values are somewhat high and this questions if the improvement can be considered statistically significant or not. CycleGAN+KL, instead, only outperforms the original untransferred images in SSIM, while results in lower performance for all the measures calculated on a pixel to pixel comparison.

Methods	SSIM	Pixel accuracy	Mean IU	MSE
Original	0.81±0.24	0.88±0.21	0.85±0.24	8060.93±13623.30
CycleGAN	0.85±0.24	0.89±0.19	0.86±0.24	7030.84±12732.60
CycleGAN + KL	0.83±0.24	0.87±0.21	0.84±0.24	8172.35±13487.79

Table 4.6.: Image similarity measures between predicted and ground truth patches for Leica original images and images transferred with CycleGAN and CycleGAN+KL.

4.2.2. Dataset 2

To be able to run UNIT, a lower number of training data is used and on the same set CycleGAN and CycleGAN+KL are also trained.

The following table shows the methods used on this set, with the number of epochs and the loss functions details:

Methods	# epochs	loss functions
CycleGAN	70	adversarial + cycle consistency
CycleGAN + KL	70	adversarial + cycle consistency + KL divergence
UNIT	90	adversarial + cycle consistency + VAE

Table 4.7.: Summary of methods used on dataset 1: models, number of epochs and loss functions.

In Figure 4.7 some examples of the results obtained applying the three methods in Table 4.7 are shown. Using a smaller amount of training data, influences the background for CycleGAN and CycleGAN+KL, without modifying the texture of the image. It applies changes more on color level. UNIT, instead, does not always show preservation of the structure but modifies the image also reducing the background information in some areas.

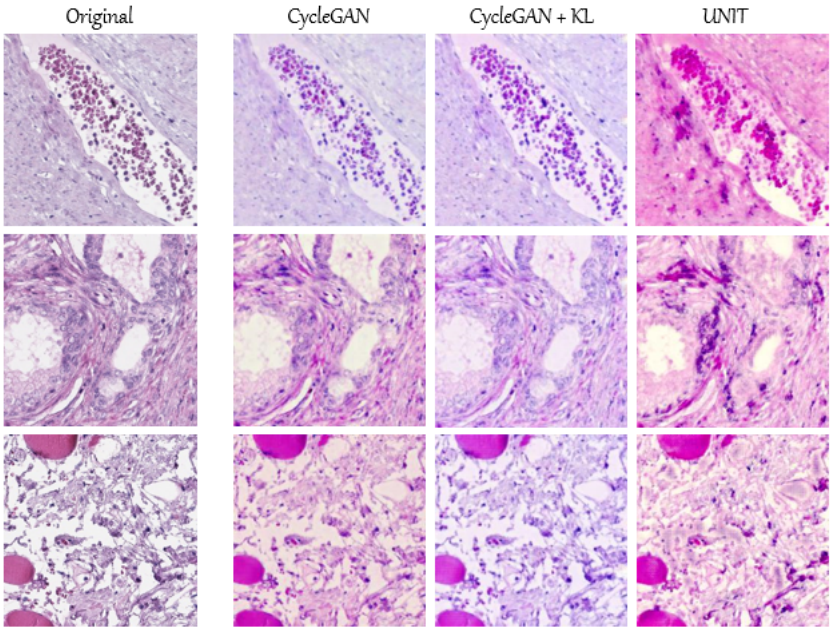

Figure 4.7.: Before and after Style Transfer. The first column on the left contains three Leica original patches from the test set while the other three columns on the right contain the style transferred images with CycleGAN, CycleGAN+KL and UNIT.

A comparison between Y, Cb and Cr color histograms is shown in Figure 4.8 and KL divergence (Table 4.8) is calculated per channel to compare with results from chapter 2 and with results obtained training on Dataset 1. All style transfer methods showed improvement in Y, Cb and Cr channels, however, training on a smaller dataset, negatively affects the results (as shown in Figure 4.7), especially on Cr

channel. UNIT results are the closest to real Zeiss style in terms of colors, but they lose morphology information.

		Y Channel	*Cb Channel*	*Cr Channel*
KL(Zeiss,Fake Zeiss)	CycleGAN	0.35	0.06	0.32
	CycleGAN+KL	0.22	0.12	0.37
	UNIT	0.09	0.07	0.09

Table 4.8.: Kullback-Leibner divergence between Zeiss and Fake Zeiss Y, Cb and Cr color histograms for all methods trained on Dataset 2.

4.2.2.1. Measurements of performance of the classifier on different methods

Tumor segmentation predictions are calculated for leica test set and for the three sets obtained applying style transfer methods described above. ROC curve (Figure 4.9), PR curve , F1 values against thresholds (Figure 4.10) and AUC values (Table 4.9) are then calculated.

According to ROC curve and AUC, the performance of the tumor classifier network increases regardless the method, but it reaches the best result with CycleGAN+KL. The ROC curve for the untransferred images is always below the others, resulting in a lower value of AUC, as shown in Table 4.9. Despite that, the values of AUC are lower than the ones obtained in Table 4.3, but CycleGAN+KL shows high performance, similar to CycleGAN trained on Dataset 1. UNIT and CycleGAN trained on Dataset 2 show very similar performance in the ROC curve, in average, though, the first method results in a 3.7% of improvement in AUC compared to the first one. The model with the highest AUC value, trained on this dataset, increases the original performance by 15% in average.

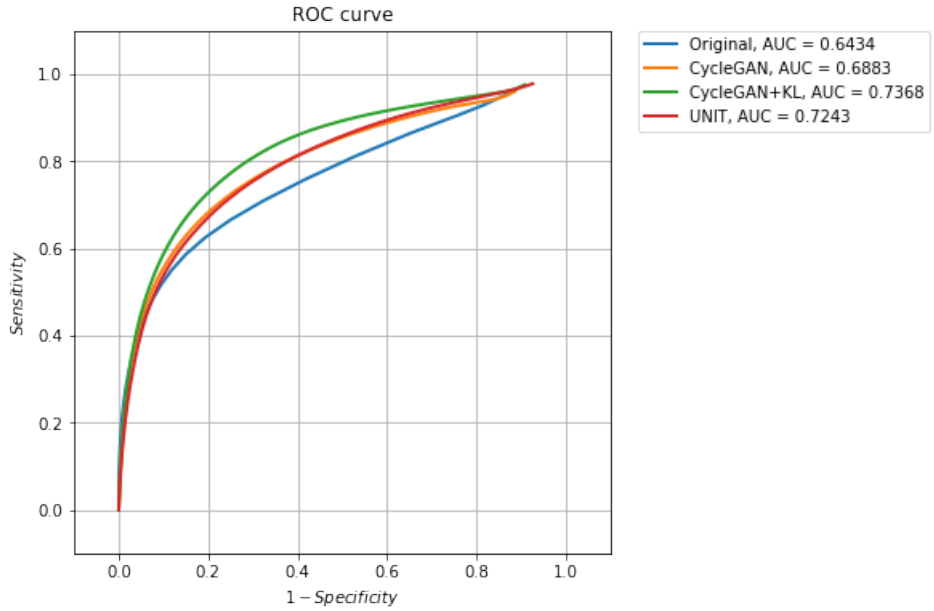

Figure 4.9.: ROC curves generated over predictions calculated on CycleGAN, CycleGAN+KL, UNIT transferred patches (FAKE ZEISS) and Original untransferred patches (LEICA). The best performance according to AUC is obtained with CycleGAN+KL.

METHODS	AUC
ORIGINAL	0.644±0.013
CYCLEGAN	0.699±0.006
CYCLEGAN + KL	0.740±0.008
UNIT	0.725±0.012

Table 4.9.: AUC mean values and standard deviation calculated over 6 sets of test data consisting of 2000 patches each.

In Figure 4.10, in the left plot F1 scores are compared for each method at different threshold values. For the original images the best value of the F1 score is obtained for threshold $t_{original} = 14$, for the CycleGAN+KL generated images it is obtained for threshold $t_{CycleGAN+KL} = 143$, for the UNIT generated images it is obtained for threshold $t_{UNIT} = 125$ and for the CycleGAN generated images it is obtained for threshold $t_{CycleGAN} = 123$. CycleGAN and UNIT show similar results in terms of F1 scores vs threshold, being above CycleGAN+KL curve before $t = 100$ and then below it. PR curves show that precision and recall values are affected by the lower number of training data used for the models to learn, in fact, the curves have lower values of F1 compared to the ones in Figure 4.5.

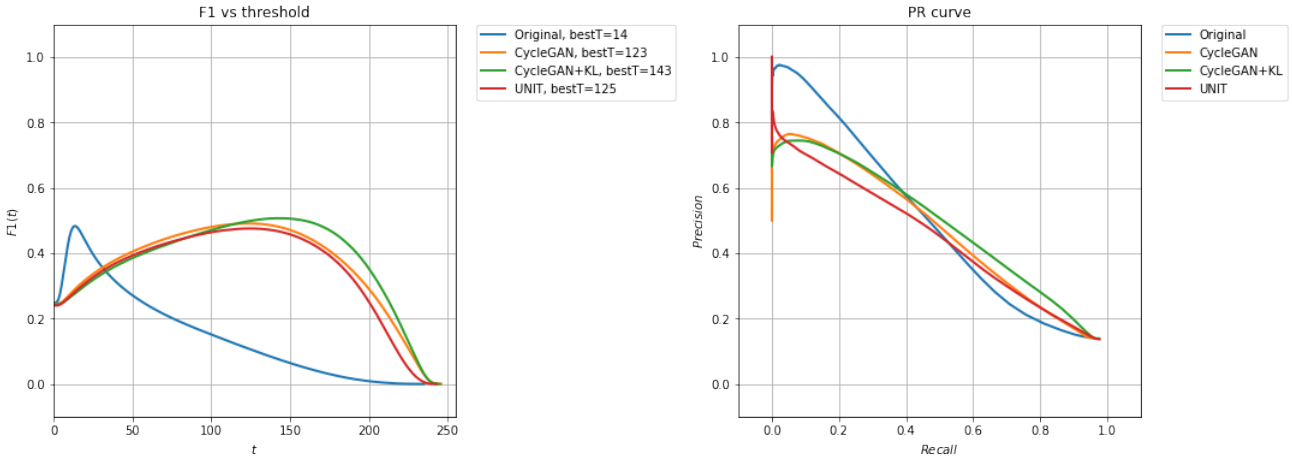

Figure 4.10.: On the left F1 score plotted against different thresholds values, on the right the PR curves.

After finding the best thresholds according to F1 score, thresholding is performed for all patches and confusion matrices are calculated:

Methods	TP	FP	TN	FN
Original	5.78%	5.44%	81.82%	6.96%
CycleGAN	6.06%	6.66%	81.37%	5.92%
CycleGAN + KL	6.49%	7.29%	80.87%	5.35%
UNIT	6.07%	7.50%	80.54%	5.90%

Table 4.10.: Confusion matrices for transferred and untransferred patches predictions results in percentage on the overall number of pixels, for models trained on Dataset 2.

All methods increase the number of true positive and decrease the number of false negative, but they have worst performance in detecting true negative and they also negatively affect the number of false positive. For this reason, even if the graphical methods from the previous analysis showed the success of style transfer methods, precision does not show such an improvement for transferred images compared to untransferred images, as reported in Table 4.11.

Methods	Accuracy	Precision	Recall	F1
Original	0.874±0.004	0.47±0.02	0.47±0.03	0.47±0.01
CycleGAN	0.878±0.007	0.46±0.02	0.52±0.01	0.48±0.01
CycleGAN + KL	0.882±0.006	0.47±0.02	0.55±0.01	0.51±0.01
UNIT	0.872±0.002	0.43±0.01	0.51±0.02	0.47±0.01

Table 4.11.: Performance of segmentation tool on different stain normalization methods on Dataset 2.

4.2.2.2. Image Similarity Measures on different methods

SSIM, pixel accuracy, Mean Intersection Over Union and MSE are here calculated between each prediction and the corresponding ground truth patch for the untransferred set and the transferred set with CycleGAN, CycleGAN+KL and UNIT. In Table 4.12 the applied methods do not show any improvement in performance of the segmentation tool for both pixel to pixel and structural measurements. CycleGAN+KL method results in same pixel accuracy, Mean Intersection Over Union and MSE values of Table 4.6. CycleGAN, instead, shows to be more sensitive to the number of training data, having a significant decrease in performance for all measurements.

Methods	SSIM	Pixel Accuracy	Mean IU	MSE
Original	0.81±0.24	0.88±0.21	0.85±0.24	8060.93±13623.30
CycleGAN	0.80±0.28	0.87±0.20	0.84±0.24	8174.79±13206.01
CycleGAN + KL	0.81±0.27	0.87±0.21	0.84±0.25	8218.20±13572.94
UNIT	0.81±0.27	0.87±0.23	0.83±0.26	8708.02±14648.69

Table 4.12.: Image similarity measures between predicted and ground truth patches for Leica original images and images transferred with CycleGAN, CycleGAN+KL and UNIT.

4.2.3. Statistical testing to assess significance of differences in Image Similarity Measures distributions

The ground truth patches, the predictions are compared with, have a lower resolution. For this reason the image similarity measures are not 100% accurate. Beside that, this does not influence the objective of the thesis work because what needs to be significant is the improvement of the performance of the segmentation tool on style transferred images, instead of original images.

At this point, the Image Similarity Measures are calculated between LEICA predictions and ground truth patches and between FAKE ZEISS predictions and ground truth patches resulting in two vectors of paired data for each of the methods described above. The significance of the difference between them is now investigated for the models which resulted in higher values of one among the measures calculated in the previous subsections.

4.2.3.1. Paired t-test:

Because of the size of the test set, testing the normality of the difference between samples is not needed. Anyways, a visual inspection of the distribution of the differences between paired data for each measure is showed in Figure 4.11.

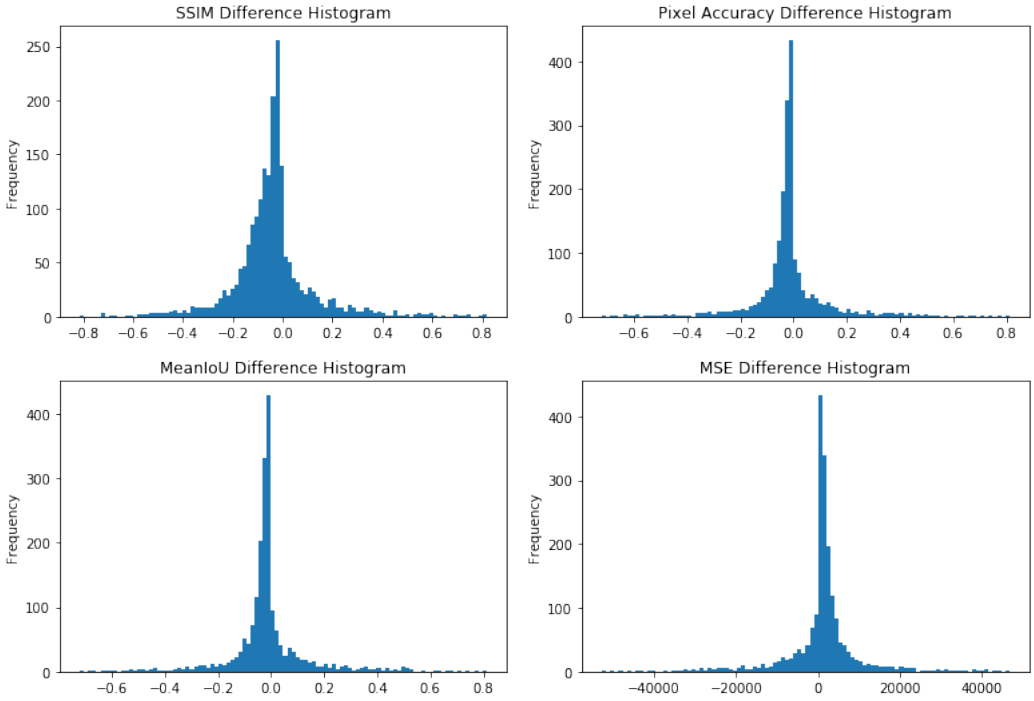

Figure 4.11.: To test the assumption of normality, a variety of methods are available, but the simplest is to inspect the data visually using a tool like a histogram. All graphs show an approximate bell-shape which is typical of real-world data to not be perfectly normal.

In Table 4.6, predictions obtained on CycleGAN generated images showed to be more similar to the ground truth, compared to original LEICA images predictions, according all similarity measures. In fact, SSIM, Pixel Accuracy and Mean IoU for CycleGAN are above and MSE is below the respective measures calculated for original images.

The following table shows the probabilities of observing the test results under the null hypothesis where the null hypothesis states that there is no difference between the image similarity measures calculated between original images prediction and ground truth images, and style transferred images prediction and ground truth images.

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	$1.39 * 10^{-28}$	$9.245 * 10^{-7}$	0.001	$9.245 * 10^{-7}$
test statistic	−11.26	−4.92	−3.28	4.92

Table 4.13.: Paired t-test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN method from sec. 4.2.1) predictions and ground truth images.

Because of the low of p-values, there is less than 5% chance of obtaining a result like the one that was observed if the null hypothesis was true so the null hypothesis can be rejected.

But what about CycleGAN+KL trained on Dataset 1? Can this method be considered significantly different for some of the measurements? After assessing the normality of the differences, a paired t-test is performed using transferred images predictions from CycleGAN+KL method trained on Dataset 1 (Table 4.14), showing that the null hypothesis can not be rejected for pixel accuracy, mean IU and MSE because the p-value is too high, while can be rejected for SSIM, as supposed by looking at Table 4.6. This means that the segmentation tool does not change performance with the aid of style transfer on most all the metrics but SSIM.

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	$2.887 * 10^{-09}$	0.645	0.277	0.645
test statistic	−5.964	0.461	1.087	−0.461

Table 4.14.: Paired t-test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN+KL method from sec. 4.2.1) prediction and ground truth images.

Because of the high values of AUC obtained by CycleGAN+KL trained on Dataset 2, a t-test is performed also for this method prediction results (Table 4.15). The null hypothesis is rejected for SSIM and mean IU but can not be rejected for pixel accuracy and MSE. Both those measures have low p-values, but, as shown in Table 4.12 and tested with a one side test, mean IU does not support the hypothesis of improvement using style transfer.

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	0.032	0.485	0.015	0.485
test statistic	−2.141	0.698	2.420	−0.698

Table 4.15.: Paired t-test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN+KL method from sec. 4.2.2) prediction and ground truth images.

4.2.3.2. One-sample Permutation Test

Under the assumption that the data have a symmetric distribution (see Figure 4.11), the hypothesis to test in this nonparametric method is that the sample of standardized differences is distributed symmetrically about 0, against the alternative that the sample of standardized differences comes from a population with mean different from 0.

The number of signs flips for the permutation test is set equal to 10,000. At each step the sign of random elements in the sample are flipped and the overall mean value is calculated. The results obtained from this test are reported in the following table:

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	0.00009	0.00009	0.002	0.00009
test statistic	−0.04	−0.02	−0.01	1030.09

Table 4.16.: Sign flipping test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN method from sec. 4.2.1) prediction and ground truth images.

Also in this case the null hypothesis is rejected in favor of the alternative hypothesis, because of the low p-values obtained.

For CycleGAN+KL trained on Dataset 1 and CycleGAN+KL trained on Dataset 2, results of nonparametric tests are reported in Table 4.17 and Table 4.18, confirming the previous reasoning, done for the parametric tests.

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	0.00009	0.656	0.267	0.640
test statistic	−0.024	0.002	0.004	−111.421

Table 4.17.: Sign flipping test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN+KL method from sec. 4.2.1) prediction and ground truth images.

	SSIM	Pixel Accuracy	Mean IU	MSE
p-value	0.030	0.494	0.015	0.478
test statistic	−0.008	0.002	0.008	−157.271

Table 4.18.: Sign flipping test results for SSIM, pixel accuracy, mean IU and MSE measures calculated between original images prediction and ground truth images, and style transferred images (CycleGAN+KL method from sec. 4.2.2) prediction and ground truth images.

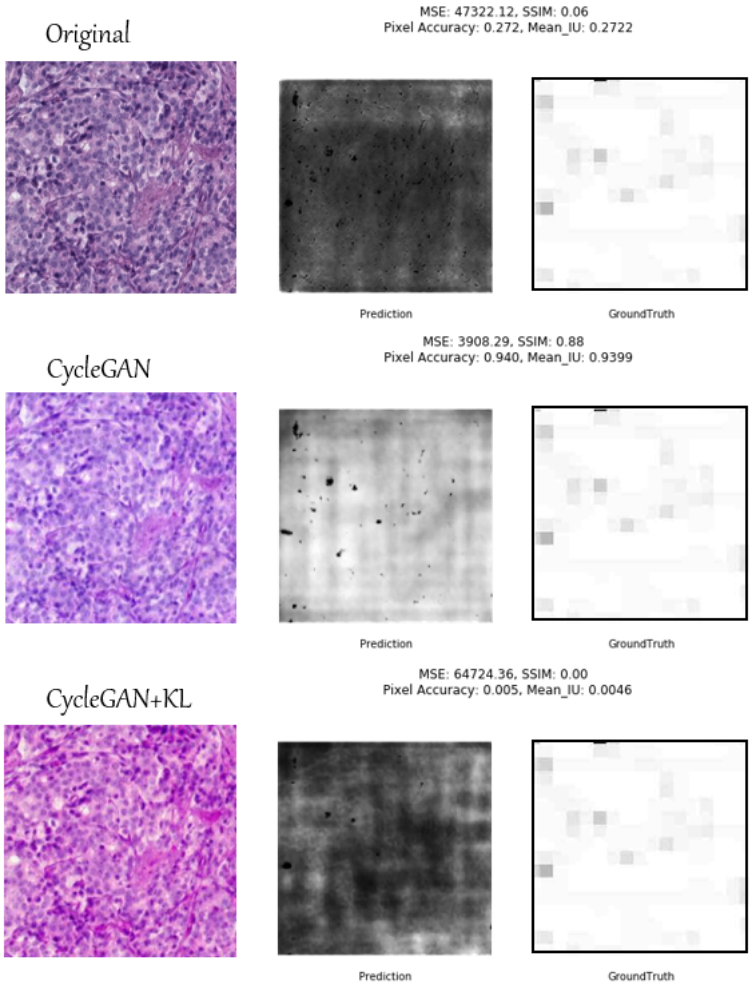

Figure 4.6.: In the first row the original Leica image is showed, with its prediction generated by ContextVision segmentation tool and the ground truth image the prediction needs to be compared with (on the right). The second and the third row are respectively the outputs of CycleGAN and CycleGAN+KL models when the original Leica image was used as input, with their prediction and the same ground truth image. All measures are reported for the three cases, they are calculated after transforming both prediction and ground truth into binary images according to the best threshold. CycleGAN, in this case, shows a great improvement of performance of the segmentation tool during prediction, recognizing the patch as cancer area.

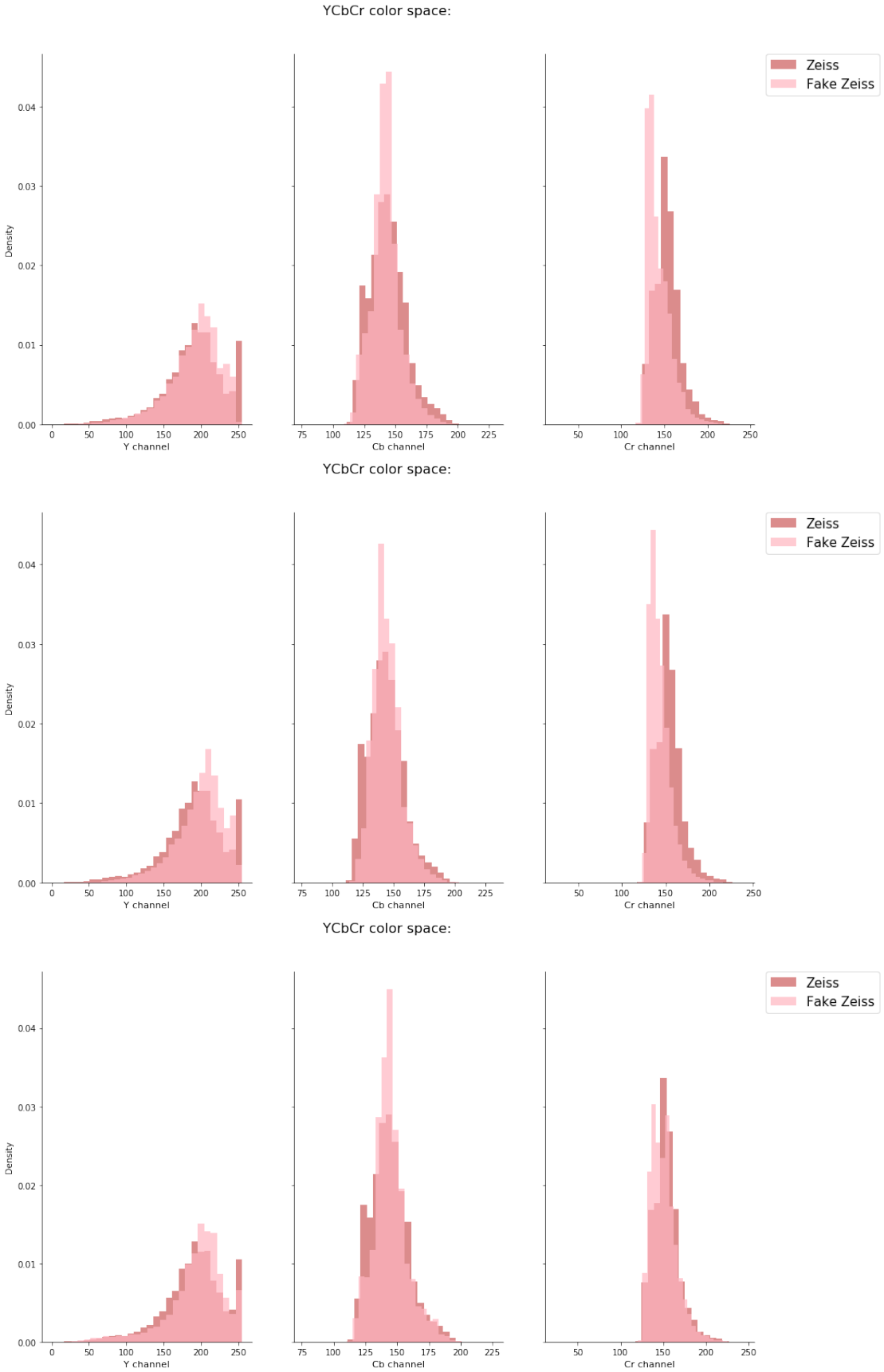

Figure 4.8.: Comparison between Y, Cb and Cr color histograms for Zeiss and Fake Zeiss images: first CycleGAN, second CycleGAN+KL and third UNIT.

5. Discussion

In this thesis work, unlike other relative works [21], the segmentation model is not included in the stain transfer network architecture but they are considered as separate. For this reason, first the methods and then the results and their evaluations are discussed.

5.1. Methods

Using GANs [11] in Digital Pathology is a quite challenging task. Pathologists analyzing tissues using very accurate microscopes are pretentious about image quality, so they increase the expectations of synthetic images. Hence, the Image to Image translation problem in this context needs to take into account tissue morphology more than in other style transfer problems (e.g. transferring Picasso painting style to a picture). The choice of using patches instead of images with original dimensions is both due to the networks memory consumption and to the preservation of details. The patch size is chosen to be 256x256 pixels because it showed to be successful in the literature [21, 24], but it would be interesting to test how increasing it affects the stain transfer problem. Training both CycleGAN and UNIT networks showed interesting behaviors.

The first trials on CycleGAN failed in recognizing the difference between background information and tissue, the output of the testing set, in fact, resulted in images with reversed darker tissues and white background. One possible explanation was having too few patches with background in the training data, such that the network was not able to understand this difference, but the problem persisted when increasing this number so the motivation does not stand. Another possible explanation is that CycleGAN is trained on patches of 256x256 pixels, while the original images can be 50,000x50,000 pixels, and the field of view can therefore be too small for the network to recognize background pixels. The output resulted to be very sensitive to the initialization of parameters and its randomness, so one solution used during simulations was running the model, checking after few epochs the saved samples and restarting the training in case the phenomenon appeared. The choice of the number of epochs also resulted from several trials and analysis of loss functions behaviors and as trade-off with the number of training patches. Without changing the number of epochs, training CycleGAN on a small (sec. 4.2.1) or on a large (sec. 4.2.2) dataset affects the results mainly on a color level (see Figure 4.2 and Figure 4.7), softening

and enhancing the color contrast respectively. CycleGAN shows to have the power of keeping the shape of the object and not producing distorted images. This is probably the main reason why the method fits the objective well.

The CycleGAN+KL model was first introduced as a solution to the background-tissue reversing problem. During learning the mapping functions, the distance between the gray scale histograms of the generated and target image is asked to be minimized. Adding a KL divergence loss did not succeed in achieving this goal, but showed to sharp color contrast more than simple CycleGAN did (sec. 4.2.1, sec. 4.2.2).

Beside CycleGAN, a more complex unsupervised Image to Image translation method was used: UNIT. Instead of four as in CycleGAN, six networks are trained simultaneously in this model, increasing the running time and the complexity. Because of computer memory limitations, it was only able to train on a dataset of 4,000 patches. To compare CycleGAN and CycleGAN+KL with UNIT, they were both also trained on a smaller dataset. Comparing models trained on such different number of data would have been unfair. While for CycleGAN the background-tissue reversing problem is consistent in all tested patches given the model (either all patches were reversed or not), for UNIT it is not. Stained transferred images show that the model affects the texture of the tissue beside the color appearance and even if it recognizes some background areas, it is not consistent in the identification.

From a pathologist point of view, CycleGAN and CycleGAN+KL better fits the aim of the thesis work because there is no change in morphology in the synthetic images after style transfer. As the literature showed, though, visual judgment and the result of a classifier can sometimes lead to two different conclusions [6], that is why from a deep learning segmentation model point of view, the chance of obtaining good results from deformed images can not be excluded.

5.2. Results

ContextVision's segmentation tool and GAN methods share the same train and test WSI set. Are the results therefore biased? Is training style transfer on the same training set used for classification affecting somehow the performance? There is not a certain answer for this question but having two completely different objectives, as predicting cancer area and performing stain transfer respectively, this should be less of a problem.

The evaluation procedure (sec. 4.1) adopted in this thesis work was formulated according to the available resources. ContextVision's segmentation tool predicts on WSI in high resolution. To evaluate the performance of style transfer methods, instead, predictions on patches were performed making the original evaluation method used by the company impossible to use. Step 2 in sec. 4.1 causes information loss, especially when ground truth patches are compared with high quality predictions,

that is why focusing on the difference between models performance instead of looking at each result by its own is preferred.

Improvements are measured comparing predictions on Leica untransferred and transferred images. Information about the performance of the segmentation tool on Zeiss dataset are protected under non-disclosure agreement. The tumor classifier network shows an improvement of the performance on all images obtained from stain transfer methods compared to the untransferred ones, according to ROC curve and AUC. To be able to assess those results, all models were tested on other five sets consisting of 2,000 patches each (Table 4.3 and Table 4.9), showing that the worst model still increased the mean AUC by 8.5%. After obtaining such results, the question is: can the AUC be trusted as measure of performance? The ROC curve is built evaluating a classification model at each classification threshold, and therefore gives an overall evaluation of the performance. Beside that, though, classifying a cancer area from a not cancer area translates into a binary problem which requires the identification of a separator between classes. Do false negatives and false positives have a similar cost? Is detecting cancer in a no cancer area better or worse than the opposite in this context? From a medical point of view the costs are not similar, both situations have consequences but a false negative could lead to death; for a medical devices company point of view, instead, the tool requires to reach great performance regardless the error committed. The best threshold is then chosen according to the best F1 score so that the number of "false alarms" is minimized. Another indicator showing stain transfer success is represented by F1 curves (Figure 4.5 and Figure 4.10), not by the F1 score specifically, but by the best threshold values resulting from them. All stain transfer methods have as best threshold, according to F1 score, values going from 102 to 143, while the segmentation model on original image has 14, which means that predictions on patches are so dark that dark gray, corresponding to pixel value of 20, is classified as cancer.

Confusion matrices, accuracy, precision and recall, pixel accuracy, meanIU and MSE calculated at the best thresholds, on the overall pixels, showed the same behavior for all models except for CycleGAN trained on Dataset 1. Stain transfer helps to detect true positives and to decrease false negatives, but does not always succeed in detecting true negatives causing an increase in the number of false positives compared to the untransferred images. For this reason accuracy and precision values do not show improvements as recall does. It is also true that all those calculations are done between high quality predictions and low quality ground truth patches, so they are not so accurate. An improvement of 0.002 in accuracy has a larger impact than what it can seem because of the large number of samples it has been calculated from (when N is very large, even small improvements will be seen as significant in a t-test). Even if the values are close to the ones calculated for the original images, the question is if values obtained on transferred images are significantly different compared to the one obtained on untransferred images, and what the difference is among different models. Paired t-tests or permutation tests could be performed to compare all different models, the same methods trained on different amount of training data,

AUC calculated on different datasets for transferred and untransferred images, etc. After identifying the best model which showed to improve all measurements both on the overall pixels and on the patch-ground truth comparison as CycleGAN trained on Dataset 1 (increasing AUC by 16%), statistical tests confirmed this hypothesis showing the achievement of the thesis aim.

One limitation of this thesis work is not having target values to compare the results with: having unpaired images instead of paired, do not allow to evaluate prediction of real Zeiss images pointing out the best possible performance.

5.3. Future Work

As explained in sec. 1.1, there are two different approaches that could be used to improve performance in this context: increasing variability in the training data or decreasing it adapting the testing set to its style. In this thesis work GANs helped to face the second approach, but they could also be used in the first one, performing data augmentation. In case, for example, few images coming from one institute using a different scanner are available for the segmentation model to learn, GANs could be used to perform style transfer and generate new training data [3].

What if the segmentation tool was trained on Leica instead? Would style transfer work better in the other direction?

What if many other scanners were used to produce WSI? This thesis work only focused on two different scanners, another idea in case WSI from many different scanners are available, is to perform Image to Image translation for multiple domains using only a single model as proposed by StarGAN [7].

6. Conclusions

Are Generative Adversarial Networks an effective approach, as preprocessing step, to reduce the impact that 'non-biological' variations on histopathology data has on the performance of a computer driven segmentation tool?

Generative Adversarial Networks shows to be an effective approach for the stain transfer problem resulting in high quality transferred images and improved classification results. On an image quality level, Zeiss image domain and Fake Zeiss image domain (obtained applying GANs on Leica image domain conditioning on Zeiss domain) were compared on Y, Cb and Cr color channels, showing a very strong similarity in distribution. According to Kullback-Leibler divergence the loss of information encountered when the transferred images are used to approximate Zeiss domain is very low especially on Cb and Cr channels where Leica and Zeiss domains differ the most. Quantitatively, GANs showed superior performance, always increasing the AUC of the segmentation tool (the best model reaches 16% of improvement).

Are all the Unsupervised Image to Image translation methods able to significantly improve predictions of the segmentation tool in the same way?

Although the segmentation tool always outperformed, in terms of AUC, when used to predict on transferred images compared to untransferred images, an overall improvement was obtained only by CycleGAN trained on 40,000 patches. UNIT shows potential especially in learning the color and the staining style, but fails in preserving the tissue morphology and therefore produces predictions which are sometimes worse than original Leica's predictions. Adding KL divergence to CycleGANs loss function, instead, showed some improvements in detecting cancer areas but performed worse than original Leica images in detecting no cancer tissue, however these patches predictions have an higher similarity structure when compared with the ground truth patches then the original Leica images.

A. Software

A Linux computer Ubuntu 16.04 with two GeForce GTX TITAN X 12GB GPUs was used for this thesis project. Tensorflow was used for both CycleGAN and UNIT frameworks. Two virtual environments were created according to requirements each implementation had.

- CycleGAN:
  - CUDA 9.0
  - cuDNN 7
  - Python 3.5
  - tensorflow 1.12.0
- UNIT:
  - CUDA 8.0
  - cuDNN 6
  - Python 3.6
  - tensorflow 1.4

In the data preprocessing part, working on WSI in python was done using the OpenSlide library. It allows to read image data at the resolution closest to a desired zoom level. The documentation is available at `https://openslide.org/api/python/`.

The CycleGAN implementation used in this work is available at `https://github.com/xhujoy/CycleGAN-tensorflow` while the UNIT implementation at `https://github.com/taki0112/UNIT-Tensorflow`. The default hyperparameters were used for both the implementations beside the **batch size** raised to 4 and the **number of epochs** as reported in chapter 4. In CycleGAN the initial **learning rate** for Adam optimizer is set equal to 0.0002, as default, while in UNIT equal to 0.0001. In CycleGAN the cycle consistency loss is regularized by a factor of $\lambda_{cyc} = 10$. The CycleGAN+KL model was obtained adding to the CycleGAN implementation the KL divergence coded in tensorflow. For λ_{KL} a value of 0.1 was chosen. In UNIT, instead, the regularization terms in the loss function were set to $\lambda_0 = 10$, $\lambda_1 = 0.1$, $\lambda_2 = 100$, $\lambda_3 = 0.1$ and $\lambda_4 = 100$.

Training CycleGAN and UNIT requires much time, especially if the number of training data is very large (e.g. over 10 000). CycleGAN and CycleGAN+KL

trained on Dataset 1 required around 12 and 10 days each for 90 and 70 epochs respectively. On Dataset 2, instead, 2 or 3 days each were enough for CycleGAN, CycleGAN+KL and UNIT for 90, 70 and 70 epochs respectively.

For the evaluation of results, some of the evaluation metrics were inspired by `https://github.com/martinkersner/py_img_seg_eval` while the others were manually implemented. Parametric tests were performed using Scipy library while nonparametric tests using Permute (documentation at `http://statlab.github.io/permute/permute.pdf`).

Bibliography

[1] Hatim Aboalsamh. *Applied Computing and Informatics*. Academic Press, 2019.

[2] Tinku Acharya, Ajoy K. Ray, and Andrew C. Gallagher. Image Processing: Principles and Applications. *J. Electronic Imaging*, 15(3):039901, 2006.

[3] Antreas Antoniou, Amos J. Storkey, and Harrison Edwards. Data Augmentation Generative Adversarial Networks. *CoRR*, abs/1711.04340, 2017.

[4] Aïcha BenTaieb and Ghassan Hamarneh. Adversarial Stain Transfer for Histopathology Image Analysis. *IEEE Trans. Med. Imaging*, 37(3):792–802, 2018.

[5] Nikolay Burlutskiy, Nicolas Pinchaud, Feng Gu, Daniel Hägg, Mats Andersson, Lars Björk, Kristian Eurén, Cristina Svensson, Lena Kajland Wilén, and Martin Hedlund. Segmenting Potentially Cancerous Areas in Prostate Biopsies using Semi-Automatically Annotated Data. *CoRR*, abs/1904.06969, 2019.

[6] Hyungjoo Cho, Sungbin Lim, Gunho Choi, and Hyunseok Min. Neural Stain-Style Transfer Learning using GAN for Histopathological Images. *CoRR*, abs/1710.08543, 2017.

[7] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8789–8797, 2018.

[8] Srilakshminarayana Gali. On Importance of Normality Assumption in Using a T-Test: One Sample and Two Sample Cases. 2015.

[9] Tiago Marques Godinho, Rui Lebre, Luís A. Bastião Silva, and Carlos Costa. An efficient architecture to support digital pathology in standard medical imaging repositories. *Journal of Biomedical Informatics*, 71:190–197, 2017.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[12] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[13] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann, 2011.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5967–5976, 2017.

[15] Salome Kazeminia, Christoph Baur, Arjan Kuijper, Bram van Ginneken, Nassir Navab, Shadi Albarqouni, and Anirban Mukhopadhyay. GANs for Medical Image Analysis. *CoRR*, abs/1809.06222, 2018.

[16] Anat Levin, Dani Lischinski, and Yair Weiss. A Closed-Form Solution to Natural Image Matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):228–242, 2008.

[17] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 700–708, 2017.

[18] Martin Posch Michael Proschan, Ekkehard Glimm. Connections between Permutation and t-Tests: Relevance to Adaptive Methods. *Stat Med.*, 2014.

[19] Andrew Y. Ng and Michael I. Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 841–848, 2001.

[20] Sheldon M. Ross. *Introduction to probability and statistics for engineers and scientists (2. ed.)*. Academic Press, 2000.

[21] M. Tarek Shaban, Christoph Baur, Nassir Navab, and Shadi Albarqouni. StainGAN: Stain Style Transfer for Digital Histological Images. *CoRR*, abs/1804.01601, 2018.

[22] C.V.kulkarni Swati A.Gandhi. MSE Vs SSIM. *International Journal of Scientific Engineering Research*, 4, 2013.

[23] Jesper Kers Jeroen van der Laak Geert Litjens Thomas de Bel, Meyke Hermsen. Stain-Transforming Cycle-Consistent Generative Adversarial Networks for Improved Segmentation of Renal Histopathology. 2019.

[24] Zhaoyang Xu, Carlos Fernández Moro, Béla Bozóky, and Qianni Zhang. GAN-based Virtual Re-Staining: A Promising Solution for Whole Slide Image Analysis. *CoRR*, abs/1901.04059, 2019.

[25] Xin Yi, Ekta Walia, and Paul Babyn. Generative Adversarial Network in Medical Imaging: A Review. *CoRR*, abs/1809.07294, 2018.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017.