# Virtual H&E Staining Using PLS Microscopy and Neural Networks

## An Investigation into the Benefits of Point Light Source Illumination Microscopy for Generating H&E Stained Images of Skin Tissue

Hanna Råhnängen, Sally Vizins

Master's thesis
2024:E1


Lund Institute of Technology
Lund University

Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

# Virtual H&E Staining Using PLS Microscopy and Neural Networks

An Investigation into the Benefits of Point Light Source Illumination Microscopy for Generating H&E Stained Images of Skin Tissue

**Written by**
**Hanna Råhnängen and Sally Vizins**

**Examiner**
Niels-Christian Overgaard
**Supervisors**
Håkan Wieslander (CellaVision)
Ida Arvidsson (Lund University)
Jonna Stålring Westerberg (CellaVision)


LUNDS UNIVERSITET

LTH
LUNDS TEKNISKA HÖGSKOLA

**2024 – 01 – 14**
Master's Thesis
Degree Project in Mathematics for Engineers
Faculty of Engineering

# Abstract

Histopathological examination, crucial in diagnosing diseases such as cancer, traditionally relies on time- and resource-consuming, poorly standardized chemical staining for tissue visualization. This thesis presents a novel digital alternative using generative neural networks and a point light source (PLS) microscope to transform unstained skin tissue images into their stained counterparts. This approach utilizes PLS microscopy's unique illumination angles, providing more structural information about a sample and thereby enhancing a neural network's ability to produce accurate, virtually stained images.

Two matched datasets, each containing paired unstained and chemically stained tissue images, were used for supervised training of several networks. One dataset comprised healthy tissue, while the other, in addition to healthy tissue, included basal and squamous cell carcinomas. Given the limited scope of this master's thesis, which constrained data acquisition, these datasets were relatively small, potentially impacting the generalizability of the model. The project explored the virtual staining capabilities of UNet and DenseUNet architectures, focusing on network depth and input channels. Variations in activation functions, upsampling blocks, and attention gates were tested, alongside the development of Relativistic Generative Adversarial Network (RGAN) models.

Quantitative evaluation using standard metrics and qualitative assessment by pathologists and other medical professionals demonstrated the potential of PLS microscopy in virtual staining. The final model, based on RGAN, achieved superior staining accuracy with a structural similarity (SSIM) score of 0.799, significantly outperforming traditional bright field imaging (SSIM 0.631). However, the limited diversity and size of the datasets may have inflated these scores and highlight the need for caution in interpreting the results. The pathologists and medical professionals found virtually stained images indistinguishable from their chemically stained counterparts, with average stain quality ratings of 6.40 out of 10 for virtual images, which did not differ significantly from the rating of 6.41 for chemically stained ones. The pathologists and medical professionals were also able to classify 95.83% of all images as healthy or containing cancerous tissue correctly.

In conclusion, virtual staining using PLS microscopy holds considerable promise, offering a more standardized and sustainable approach compared to chemical staining. This method has the potential to speed up diagnosis and facilitate further analysis using image analysis algorithms. Future research could expand this technique beyond skin tissues, enhancing its applicability across a broader range of histopathological examinations.

# Acknowledgements

We want to thank our supervisors Håkan Wieslander and Ida Arvidsson for their steady support and helpful suggestions throughout the project. Their expertise within machine learning, image analysis and project design have been paramount to our successes.

We are also grateful to Emily Källström and the team at Blekinge Hospital for their help with and troubleshooting of the staining process.

Fredrik Pontén and his research group have given us invaluable access to a much more diverse dataset, which has increased the relevancy and validity of our project's results manifold. We further want to thank him and all other experts who took the time out of their busy schedule to answer our questionnaire. Their input has given us valuable insights into the strengths and weaknesses of our generated images as it pertains to practical use.

Finally we would like to thank the Computational Imaging Team at CellaVision for being lovely colleagues during our time at the company. We especially want to thank Jonna Stålring Westerberg for the opportunity to work on such an engaging and explorative subject for our thesis.

# Contents

# 1 Introduction

In 2023, only 38% of Swedes diagnosed with cancer began treatment within the regulated timeframe, a delay significantly impacted by the lengthy process of histopathological diagnosis, as reported by the Regional Cancer Centers in Sweden (RCC) [1]. This challenge is compounded by a critical shortage of pathologists and laboratory personnel [2], leading to prolonged waiting times for pathology reports and hence, anxious periods for patients awaiting diagnosis [3]. Moreover, the necessity of sending samples to specialists for analysis often adds to these delays, further prolonging the diagnostic process.

Central to this delay in arriving at a diagnosis is the process of histopathological examination, the microscopic examination of tissue to study the manifestation of disease [4]. It combines histology, the study of microscopic structures of tissue, with pathology, the study of diseases. Traditionally, histopathologic examination consists of several steps: tissue removal via biopsy, surgery or autopsy, followed by fixation to prevent autolysis, embedding, sectioning, fixation to glass microscopy slides, and staining to prepare the tissue for microscopic analysis.

Staining, where chemicals bind to specific cellular or tissue structures, is a critical step in diagnosis of disease, permitting a detailed examination of cellular architecture or pathological state [5]. One prevalent method for staining is by the use of Hematoxylin and Eosin (H&E). The preservation of tissue structure and its permeability is paramount for reliable staining [6], and factors such as tissue fixation type and duration, tissue thickness, temperature, and target accessibility can influence the efficacy of staining [7]. Given this complexity, histopathology laboratories typically adhere to standardized fixation and staining protocols, ensuring minimal artifacts, reduced background interference, and fewer false positives. However, the entire process, especially the staining step, can be error-prone and time-consuming [8]. Staining is also poorly standardized between labs according to Fredrik Pontén, MD, professor at Uppsala University and pathologist with over 35 years of experience using H&E stained samples for diagnosis. Fredrik Pontén has no affiliation with CellaVision. It entails the use of hazardous chemicals, globally generating waste which needs >1 million liters of water annually to handle [8], and additionally, staining risks damaging the sample, a risk mitigated by removing larger samples from the patients [8, 9]. Staining is often irreversible, preventing the tissue from being used in other analyses, which further exacerbates the need for larger tissue removal and additional biopsies [10].

In response to the challenges faced in the traditional pathological workflow, the developing field of digital pathology has emerged as a promising solution. According to [11], digital pathology encompasses the acquisition and sharing of pathological data. Digital pathology devices normally include the salient parts of a classic microscope attached to a camera system, enabling images of physical glass microscopy slides to be captured to form a digital slide in the form of a set of images. Various methods of image analysis and machine learning methods can then be utilized to interpret or enhance the digital slide. According to Pontén, standardized images are crucial for the effective application and analysis of the aforementioned methods. A significant advancement within digital pathology is the concept of virtual staining, which aims to replace chemical staining with an image-to-image translation to create a digitally stained slide.

The possibility of generating virtual stained H&E images from unlabeled and unstained images using deep learning was first demonstrated by [12]. A comprehensive overview of recent advancements within the field is provided by [8], where unlabeled auto-florescent images and bright-field images are the most widely used inputs to the networks.

Virtual staining reduces the need for large samples, laboratory personnel, and chemicals [10]. It also eliminates the waiting time associated with traditional staining, making the process more efficient and sustainable [8]. The workflows of traditional and virtual staining are illustrated in Figure 1. With virtual staining, a digital image of stained tissues, which is easily distributed for consultation, is produced. Moreover, the digitized image facilitates analysis through algorithms [13] and supports computer-assisted diagnosis (CAD) and prognosis prediction [7].

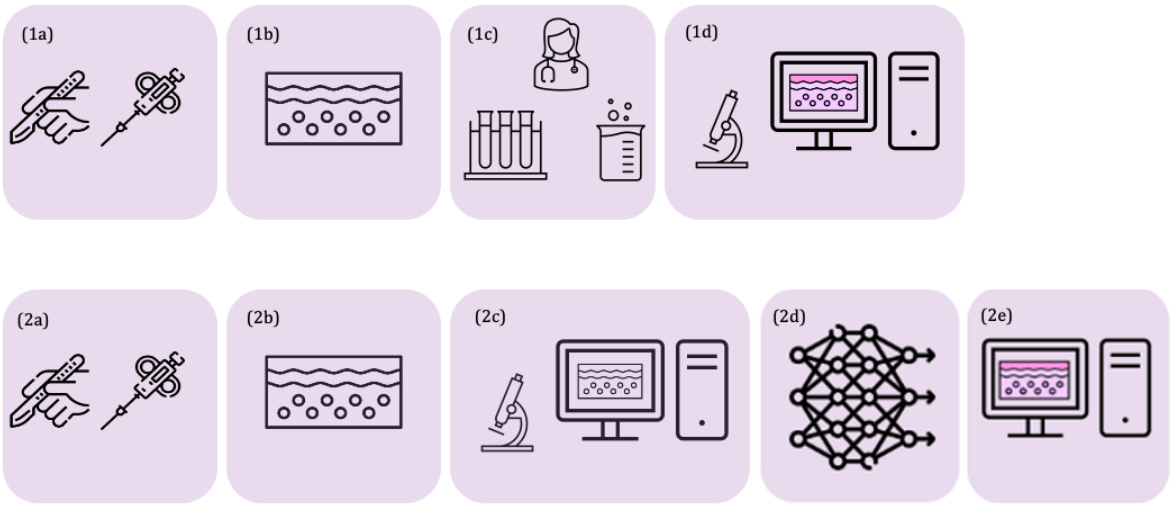


Figure 1: Illustration of the difference between traditional (1) and virtual (2) staining workflows inspired by [8]. Steps (a-b) are the same for both. (a) removal of tissue by biopsy or surgery. (b) fixation, embedding, sectioning and mounting tissue on a glass slide. (1c) traditional histological staining process. (1d) microscopic investigation of stained tissue. (2c) image collection using PLS microscope. (2d) artificial staining with a neural network. (2e) digital images of virtually stained tissue ready for histological analysis.

Despite the progress in virtual staining, there remains unexplored potential aimed to be addressed in this project. The aim of this project has been to explore the feasibility of virtually H&E staining skin tissue using generative neural networks provided with a previously untried input, Point-Light Source (PLS) images. In this project, multiple network architectures and training methods have been explored to accurately transform digital slides of unstained skin tissue into virtually H&E stained digital slides. The networks were trained using supervised learning with a paired stained-unstained dataset consisting of 132 890 and 367 372 unstained-H&E stained image pairs respectively. A recent study by [10] utilizing bright-field images as network input, compared the proficiency of virtual H&E staining of multiple tissues by an unsupervised versus a supervised network, and found that the supervised produced more accurate and realistic images, serving as inspiration for our choice.

The input to the network comprised unstained images captured using a PLS microscope. This type of microscope, which substitutes the traditional halogen lamp with a programmable light-emitting diode (LED) array, generates what is known as a PLS stack. A PLS stack consists of multiple images, each taken with a different LED, providing varied illumination angles. Compared to images from conventional microscopes, a PLS stack offers enhanced structural details about the sample, even allowing for a synthetic increase in magnification [14]. This additional detail and information is particularly beneficial for virtual staining, where a key challenge is the absence of certain information in unstained tissue images that should be present in stained ones. We hypothesize that the PLS microscope effectively can bridge this information gap, thereby enabling a neural network to accurately infer the required features from an unstained image to produce high-quality, virtually stained images. Utilization of PLS images has previously been investigated by Joel Wulff, who demonstrated promising results in virtual staining of white blood cells by training a Relativistic Generative Adversarial Network on 30,000 examples of such cells [15]. This project aims to extend insight from the aforementioned project into the diverse realm of pathology and explore the usefulness of PLS images for virtual staining of whole skin tissue slides.

Previous virtual staining projects, as investigated by [8], evaluate model proficiency using standard evaluation metrics, such as mean squared error, structural similarity index (SSIM) and peak-signal-to-noise ratio. Half of the virtual stain studies investigated also employ pathologist assessment of the images. In this project, we employ both of these approaches.

The results of this project were evaluated using standard quantitative metrics, SSIM and Laplacian focus scores, with the addition complex wavelet SSIM (CW-SSIM), which does not seem to have been explored in previous works, and the popular metric for evaluating generative networks, Frechet Inception Distance [16]. The quality of the generated virtually stained images and their usefulness were also evaluated in a questionnaire sent to a group of pathologists and other medical professionals. To evaluate the importance of the extra structural information provided by the PLS images, our final model was also trained using only traditional microscope images as input. With this evaluation process we aim to present an answer to the questions:

*Which of the attempted neural networks, trained with PLS images as input, demonstrate optimal performance in the task of virtually H&E staining skin tissue? Does training a neural network with PLS images as input, rather than traditional microscopy images only, enhance virtual staining performance? Additionally, to what extent do the generated images retain valuable pathological information?*

# 2 Background

This background section aims to provide the reader with the necessary knowledge to fully grasp the scope of the project. H&E staining will be introduced, accompanied by an overview of the basic histology of both healthy and cancerous skin tissue. An explanation of the optics in a PLS microscope will be provided, followed by an introduction to the image analysis methods employed in this project. The connection between artificial intelligence and neural networks will be presented, including a detailed discussion on how to train these networks, as well as an overview of various network architectures. Lastly, the metrics used to assess the performance of the networks trained will be introduced and explained.

## 2.1 Hematoxylin and Eosin Staining

For detailed visualization of cellular and tissue structures, samples are routinely stained using Hematoxylin and Eosin (H&E).

H&E staining is a cornerstone in tissue analysis [17]. Despite advancements in pathology, the chemical composition of this stain, along with its staining protocol, has endured for over a century. This can be attributed to its compatibility with a wide range of fixatives and its ability to distinctly highlight a vast spectrum of cytoplasmic, nuclear, and extracellular matrix features.

In the H&E staining process, Hematein — an oxidized form of hematoxylin — is complexed with a mordant, typically a metal cation such as aluminum alum [18]. This forms a cationic dye complex. Given its positive charge, it preferentially binds to negatively charged, basophilic cell components like nucleic acids in the nucleus and ribosomes. These structures subsequently manifest a blue or purple hue, the exact shade decided by the mordant. Conversely, Eosin, being anionic, serves as an acidic dye. Its negative charge facilitates binding to positively charged (acidophilic) tissue components, such as the amino groups in cytoplasmic proteins and collagen fibers within the extracellular matrix, staining these structures in various shades of pink. It is noteworthy to mention that, being an ionic stain, H&E does not stain neutrally charged components.

## 2.2 Skin Tissue

### 2.2.1 Normal Skin Histology

In Figure 2 some typical elements found in normal skin tissue are shown. Throughout these images the purple dark spots are cell nuclei, which play an important role in telling healthy tissue from sick tissue. In (a), the epithelium, the outermost layer of the skin which acts as a barrier to protect the body from the outside environment is depicted [19]. Image (b) shows connective tissue, the main component of the dermis, situated below the epidermis with the job of sustaining the epidermis. Some structures helping with this are (c) sweat glands, which help with temperature regulation, and (d) sebaceous glands, that keep the skin hydrated. (e) is an image of inflamed tissue. Inflammation is characterized by an increased number of nuclei in a concentrated area. Finally, (f) depicts fat tissue,

which also helps with temperature regulation of the body. In the skin, the layer of fat is called the subcutis and is found right beneath the dermis.

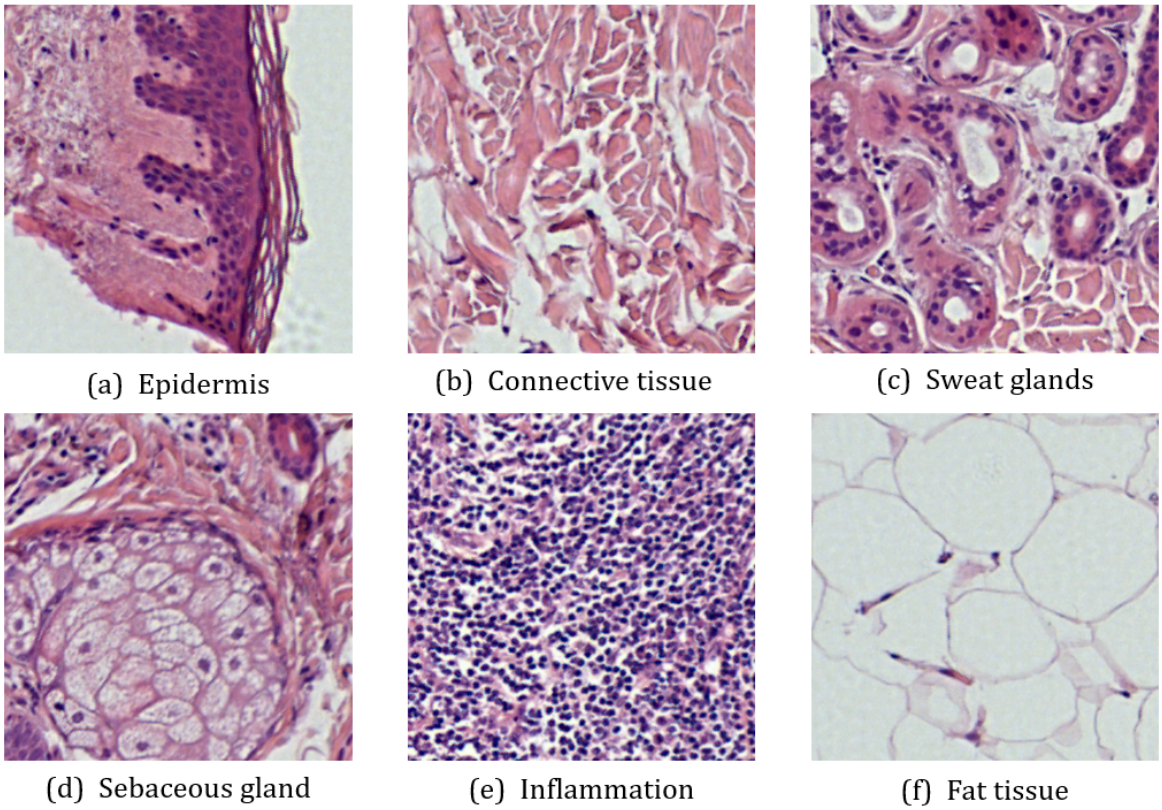


Figure 2: Some typical tissue types, stained by H&E, found in human skin.

### 2.2.2 Basal and Squamous Cell Carcinoma

Basal cell carcinoma is a type of cancer which originates in the basal cells [20]. Basal cells are located at the bottom of the epidermis and responsible for generating new skin cells as older cells die off. Over time, as skin cells age, they migrate towards the skin's surface and transform into flattened squamous cells. In Figure 3 a basal cell carcinoma can be seen to the left in the image, it is the darker purple area with high concentration of cell nuclei. Squamous cell carcinoma, another common form of skin cancer, arises from these matured skin cells. In Figure 3 the areas which are a heavier pink colour and with rather large nuclei along the edges are a squamous cell carcinoma. The squamous cell carcinoma is sometimes described as having a rose-like appearance. Both carcinomas generally begin close to the surface of the skin, in the epidermis.

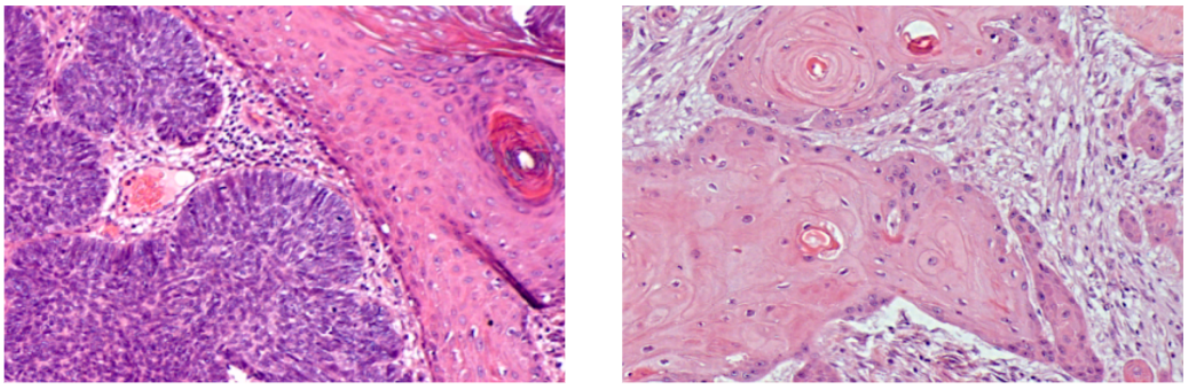

Figure 3: Basal cell carcinoma (left), squamous cell carcinoma (right).

## 2.3 Point-Light-Source Illumination Microscopy

A traditional microscope, as explained by [21], uses a halogen lamp to illuminate the sample from below, seen to the left in Figure 4. The residual of the incident light which is not absorbed by the sample and falls within the maximum normal illumination angle $\theta_{NA}$ is captured by the objective. The numerical aperture (NA), a critical factor decided by the objective lens, correlates directly to the microscope's resolution. NA can be understood as the ability of the lens to gather light and resolve fine specimen detail at a fixed object distance, calculated as

$$NA = n \sin \theta_{NA} \tag{1}$$

where $n$ denotes the refractive index of the lens and $\theta_{NA}$ the maximum half angle of the cone of light which may enter the objective from a point on the glass slide with the sample tissue on it. The area which can be seen through the objective is called the field of view (FOV). In digital microscopes, the objective is connected to a camera with a set number of pixels per image of one such FOV. For a larger $NA$, the area on the slide constituting an entire FOV is smaller, meaning the magnification is greater and therefore, the picture taken is of higher resolution.

In a PLS microscope, the traditional light source is replaced with a programmable LED array. By lighting one LED at a time, images can be obtained with a specified angle and direction of the incident light. This means that the array setup generates spatial information about the refracted light that reaches the objective. During a PLS scan of a tissue sample, a set of images are taken from a range of incident angles and directions and concatenated to form a so called a PLS stack.

As illustrated to the right in Figure 4, if the sample is illuminated outside of $\theta_{NA}$, no incident light reach the objective, and only light which has refracted and scattered in the sample to be within $\theta_{NA}$ can be observed. This results in a comparatively dark image referred to as a dark-field image (DF). Images where most of the light absorbed by the objective is the residual of the incident light can in contrast be referred to as bright-field (BF) images and BF images are those captured by traditional microscopy. Naturally, the way in which the light refracts in the sample has connections to its thickness and composition. A PLS microscope aims to increase retrieval of this type of information. The LED array is designed in such a way that some of the LED's illuminate the sample from

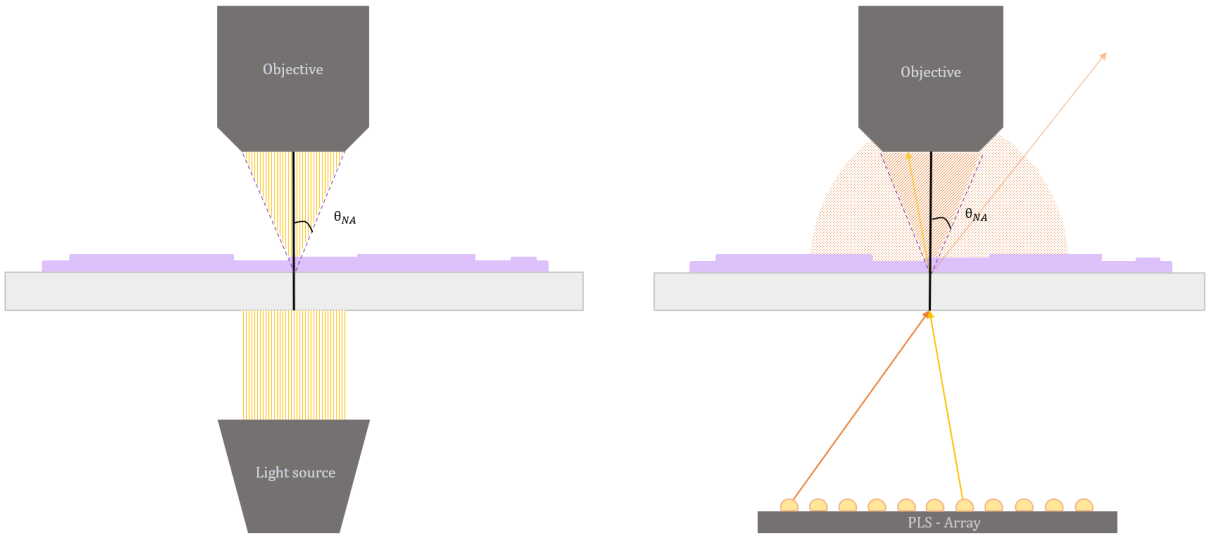


Figure 4: Traditional microscope with a light source designed so that most of the incident light is captured by the objective as it is scattered within $\theta_{NA}$ (left). PLS microscope designed so that most individual LED:s when illuminated do not yield any incident light within $\theta_{NA}$, instead producing dark-field (DF) images (visualized with orange arrows and orange shaded area) (right). The possibility to take a classic bright-field (BF) image is possible through the LED:s at the centre of the PLS array (visualized with the yellow arrows).

outside of $\theta_{NA}$, and therefore create DF-images. In Figure 5, three examples of images produced by a PLS microscope are shown, the middle image is particularly interesting as some incident light has reached the objective but also some scattered light, meaning it is taken at an angle at the border between BF and DF.

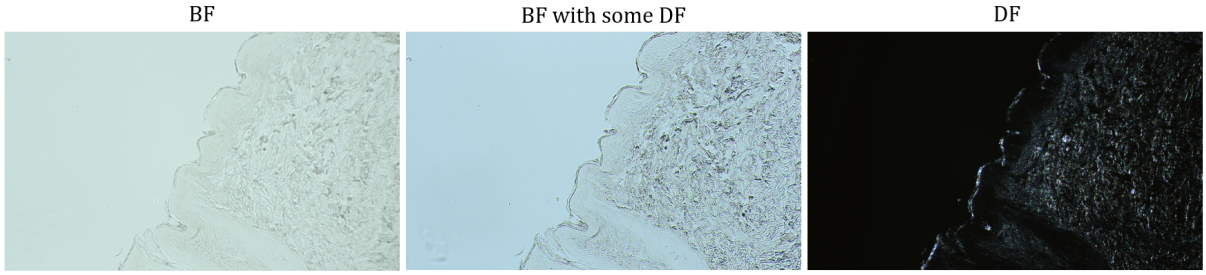


Figure 5: Image of sample with traditional BF illumination (left), sample illuminated by one LED, resulting in some incident light and some scattered captured reaching the objective (middle), sample illuminated by a different LED can also result in only scattered light being captured, producing a DF-image (right).

The information in a PLS stack compared to that in a BF image has been shown to provide more structural information to such an extent that the numerical aperture of an objective can be synthetically increased [14]. The aim of using PLS microscopy in this project is based on the hypothesis that this information can bridge the gap between the information present in an unstained tissue sample and a stained one as illustrated in

Figure 6.

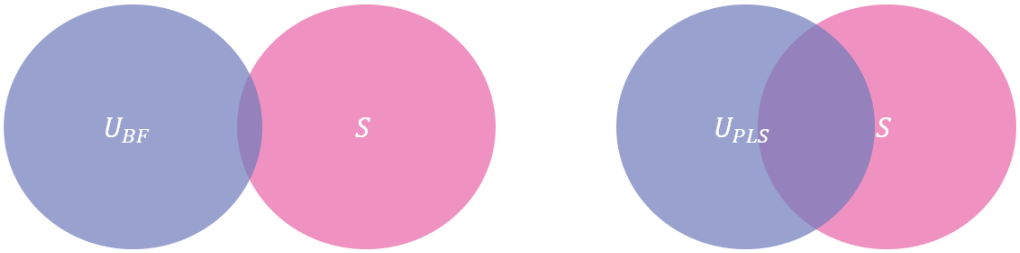


Figure 6: The overlap of information in $S$, and in $U_{BF}$ (left). A larger overlap of information between $U_{PLS}$, and $S$ (right). $U_{BF}$ indicates an unstained BF image, $U_{PLS}$ an unstained PLS stack and $S$, a stained image.

## 2.4 Image Analysis Methods

### 2.4.1 Template Matching

Template matching is a method of finding the location of the closest match of a smaller image, a template, in a larger image. This can be done by a variety of methods, but in its most basic form, the methods consists of moving the template over the larger image and by some metric measure their similarity. Once the similarity of the template has been measured for all possible positions in the larger image, the location with the best metric value is chosen as the matched location.

One method of template matching is to make use of the cross-correlation coefficient [22]. Denoting the template, $T(x, y)$ and the image, $I(x, y)$, placed at coordinate $(x, y)$, this method calculates the similarity coefficient, $R(x, y)$, as the element-wise product of the normalized template and image as

$$R(x, y) = \sum_{x'y'} (T'(x', y') \cdot I'(x + x', y + y')). \tag{2}$$

$T'(x', y')$ is the normalized template at coordinates $(x', y')$ and $I'(x + x', y + y')$ is the image normalized at $(x + x', y + y')$. The normalized values are calculated as

$$T'(x', y') = T(x', y') - \frac{1}{w \cdot h} \sum_{x''y''} T(x'', y'') \tag{3}$$

$$I'(x + x', y + y') = I(x + x', x + y') - \frac{1}{w \cdot h} \sum_{x''y''} I(x + x'', y + y''), \tag{4}$$

where $w$ and $h$ denote the width and height of the template. By calculating $R(x, y)$ for all possible placements of the template and finding location where the cross-correlation is maximized, the template can be matched to the larger image.

### 2.4.2 CIE-L*a*b Colorspace

Developed by the Commission of International Illumination, the CIE-L*a*b colorspace aims to create a colorspace where a step in any direction is percieved by the human eye as an equally large shift in color [23]. The CIE-L*a*b colorspace is divided into three channels:

- L* - for lightness which ranges from white to black.
- a* - describes the chromatic axis ranging from green to red.
- b* - describes the chromatic axis ranging from blue to yellow.

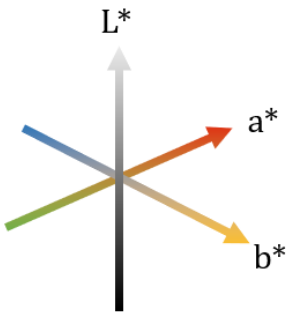


Figure 7: The CIE-L*a*b colorspace visualized by the color of the axes.

### 2.4.3 Post Processing

**Altering Saturation in CIE-L*a*b Colorspace**

One use of the CIE-L*a*b colorspace is simplifying saturation alteration. In the CIE-L*a*b colorspace, the color saturation can be described as the distance from the origin, radially from the L*-axis [23]. To alter the saturation one can therefore perform the translation shown in Equations 5 and 6,

$$x_c = x - n_{a*,b*}, \tag{5}$$

$$x_s = (x_c \cdot s) + n_{a*,b*}. \tag{6}$$

The saturation factor is denoted $s$, if it is larger than one it will increase saturation and if it is lower than one it will decrease it. To center the pixel values to form $x_c$, a neutral value of the a* and b* axes respectively $n_{a*,b*}$ is chosen. It is often chosen as 128 which is the central value if the possible pixel values are in the range $[0, 255]$. The subtraction of $n_{a*,b*}$ before applying the saturation factor, to form the saturated image $x_s$, ensures no bias is introduced in the saturation process by centering the image's pixels around a neutral value.

**Unsharp Mask Technique**

The unsharp mask technique is a way of sharpening an image by utilizing a blurred mask of the image, as visualized in Figure 8 [24]. Step one applies a blurring filter, for example a Gaussian blur, to an image to create an unsharp mask by subtracting the blurred image

from the original. To sharpen the original image, as shown in step two in Figure 8, one can then add the blurred mask to the original image. The method is effectively a high-pass filter which selectively increase contrast along the edges of an image. The results of this technique can be controlled by changing the kernel size of the blurring filter. A larger kernel will result in a more blurred mask and increasing the change to the image. However, if the kernel is too large, halo artifacts, where areas around sharp edges appear to be glowing, can appear.

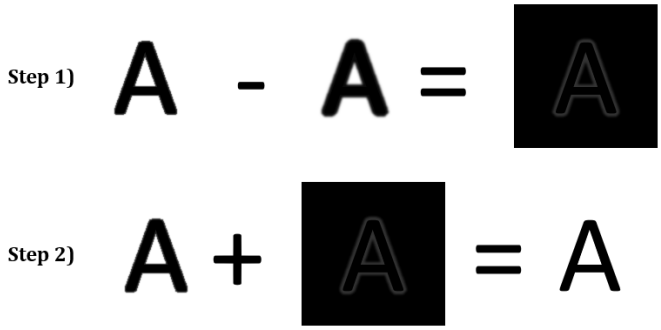


Figure 8: Visualization of the unsharp mask technique. Step 1 consists of subtracting a blurred version of the original to create a mask. This mask is added to the original image in step 2 to sharpen it.

## 2.5 Artificial Intelligence

Throughout history, the enigma of the mind has fascinated humans, and many attempts have been made to understand "how one thinks". Pioneers, from Greek philosophers to 20th century intellectuals, have tried to dissect how we, who in the purest physical sense are bundles of atoms, are capable of complex and analytic thinking [25]. What is it that enables us and other living organisms to perceive, analyze, and interact with our surroundings? From this curiosity, artificial intelligence (AI) has emerged, aiming to understand the essence of intelligence and to create intelligent beings.

In his influential 1950 paper *Computing Machinery and Intelligence*, Alan Turing posed the question, "Can machines think?" [26]. This question initiated a cascade of AI research, leading to multiple definitions of AI being explored [27], especially in the period following the Second World War, culminating with the term's formal introduction in 1956. John McCarthy defined AI as "the science and engineering of making intelligent machines, especially intelligent computer programs" [28], emphasizing the computational aspect of goal achievement observed in humans, animals, and certain machines. Stuart Russell and Peter Norvig further refined this concept in their textbook *Artificial Intelligence: A Modern Approach* [25]. They describe AI as multidimensional and classify systems as AI based on whether they think and act, either humanly or rationally. The human aspect focuses on the simulation of human behaviors and thought processes, while the rational aspect concentrates on optimal decision-making, judged against a measure of *ideal* performance [25]. According to Russell and Norvig, artificial systems can be considered intelligent if they exhibit rational thinking when performing logical and analytical tasks,

mirror human thought processes, act rational by performing goal-oriented actions, or act in a way that is indistinguishable from humans [25].

While McCarthy's definition of AI is concerned with the construction of intelligent machines, Russell and Norvig empathize the diversity of approaches one could take when developing AI. Together, these two definitions provide a holistic understanding of the AI landscape, which in its simplest form is a field which uses computer science and large datasets of information to solve intellectual problems.

### 2.5.1 Machine Learning

Machine learning (ML) is a branch of artificial intelligence which falls under the acting rational Russell and Norvig branch. In ML one tries to create systems that act to achieve a best outcome, or when the outcome is unknown, the best expected outcome. ML models adjust to and find patterns in data, learn from it, and make predictions and decisions without being explicitly told to do so. Rather than being pre-programmed with specific instructions, machine learning models adapt, evolve, and improve as they are exposed to more data. ML models consist of algorithms written to mimic the way humans learn and to gradually improve their accuracy in performing the task at hand with training.

An algorithm capable of learning is defined by Goodfellow et al. as one that is

> "able to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$" [29].

The term experience, $E$, in this context refers to the data or information the algorithm encounters and learns from over time [29].

To assess the capability of a machine learning algorithm, its performance, $P$, is measured quantitatively using metrics specific to the task $T$ at hand [30]. For instance, if the model's objective is to classify objects into categories like cats or dogs, its performance can be gauged by the number of objects it classifies correctly. On the other hand, if the model is designed to generate images, performance could be evaluated using the mean-squared-error between pixel values of the generated image and the true, ground truth, image.

ML models can learn in three different ways; supervised, unsupervised or by reinforcement learning, depending on the training data available and the order and method by which this data is received and in what data is used to evaluate the learning algorithm [31]. According to [30], supervised learning involves models learning from labeled data. It uses a dataset where input variables are paired with their corresponding correct outputs and trained to learn the mappings between inputs and outputs. Supervised learning is often used for classification or regression tasks. In unsupervised learning, models identify structures and patterns in unlabelled data. The model attempts to find hidden relationships and groupings in the inputs, which can be used for exploratory data analysis and clustering tasks.

A crucial aspect of training machine learning models is ensuring that they generalize well, meaning they perform well when presented with previously unseen data [29]. Consequently, it is common practice to evaluate a model's performance on a **test set** that's distinct from the training data. Available data is typically partitioned into three distinct segments: a **training set**, a **validation set** to monitor model performance during training and make changes accordingly, and a **test set** composed of previously unseen data [32]. It is customary to allocate approximately 10-20% of the data for validation and testing respectively, with the remaining data assigned to the training set [32].

### 2.5.2 Deep Learning and Artificial Neural Networks

One major challenge in AI is solving tasks that humans perform intuitively but find hard to formally articulate, such as distinguishing cats from dogs or understanding speech [29]. Deep learning, an extension of machine learning, has emerged as a powerful tool for these kinds of problems. Taking inspiration from biology and the way human brains are structured and function, using neural firing patterns for communication, artificial neural networks (ANNs) have been developed to model and understand complex patterns in data [33]. The networks typically consist of multiple layers through which data passes, a concept which together with their biological inspiration has given rise to their name *deep neural networks*.

These networks allow computers to learn from experience, build complex concepts from simpler ones, and to create a hierarchy of learned ideas [34]. This means that rather than requiring programmers to specify the knowledge needed explicitly, the network can learn and refine its understanding autonomously from data. In the following sections, the fundamental components of deep learning will be explored.

As explained by [35], artificial neurons (ANs) serve as the foundational elements of ANNs, mirroring the functional aspects of neurons in the human brain. In both ANNs and the human brain, neurons receive multiple inputs, process them using a transfer function, and produce an output value. This processing involves evaluating received inputs against certain weights and thresholds. If the cumulative output value surpasses a specified threshold after the application of the transfer function, the neuron activates or "fires." In ANNs, this means transmitting data to the subsequent layer, while in the human brain, it involves sending electrical signals to neighboring neurons. Conversely, if the output value is below the threshold, the neuron remains inactive, and no data is transmitted.

The components of a biological neuron and a basic AN are displayed in Figure 9. Inputs, $x_1, x_2, ..., x_n$, represent the values the AN processes. Weights, $w_1, w_2, w_3, ..., w_n$, are adjustable parameters of the neuron which determine the significance of their corresponding input. The bias, $b$, is another adjustable parameter, providing an offset to the neuron's output. The summation function calculates the combined value of the weighted inputs and the bias, expressed mathematically as $\Sigma_{i=1}^{n}(x_i w_i) + b$. The activation function, $f$, resembling the firing threshold of biological neurons, applies a mathematical operation to the summation output to produce the neuron's final output, $y_{output}$.

ANs can be combined into layers, to form neural networks. Typically a neural network consists of an input layer, multiple hidden layers, and an output layer [36]. The input

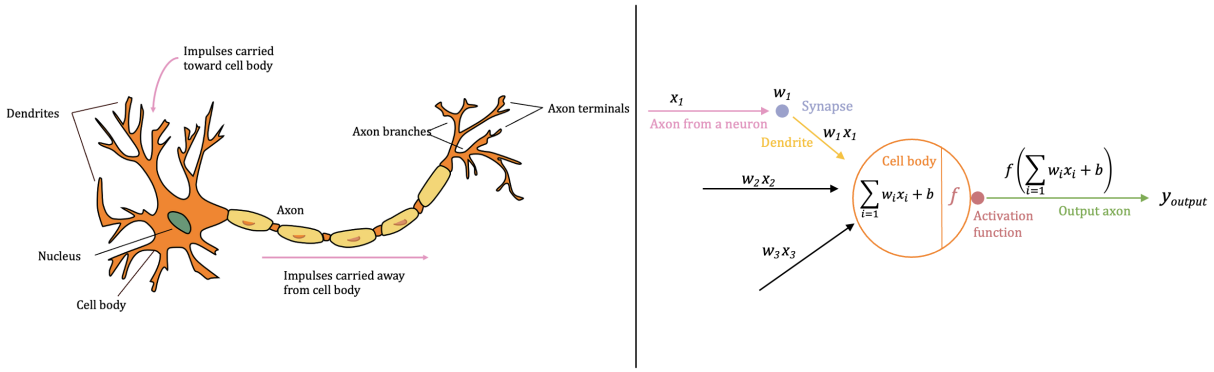


Figure 9: A biological neuron (left) and an artificial neuron (right) illustrated with inspiration from [35]. As described by [35], a biological neuron receive inputs through its dendrites and the signal is carried towards the cell body where it gets processed. If the signal is above the neurons signaling threshold, the signal is transmitted through the axon to other neurons connecting to it via a synapse at the axon terminals. An artificial neuron functions in a similar fashion by receiving and summing $n$ inputs, $x_1, x_2, ..., x_n$, to produce $y_{output}$ which is transmitted to other ANs.

layer receives an input and passes it to the subsequent layers, a process known as feed-forwarding of the data. Hidden layers, situated between the input and output layers, perform most of the computation and are responsible for feature extraction and transformation of the input into something the output layer can use. The output layer produces the final prediction based on the processed data from the preceding layers.

The architecture of a neural network, specifically the number of layers and the number of neurons within each layer can be adjusted to the task at hand. The complexity of the problem often dictates the depth of the network. For more intricate problems, deeper networks, meaning more layers and neurons, might be required to capture the subtle patterns and relationships in the data [37].

## 2.6 Training of a Neural Network

Training is the process by which neural networks learn from data to achieve their objectives. The weights and biases of neurons, which are iteratively adjusted to minimize the difference between the network's outputs and the expected outcomes, are central to this training. This difference is quantified by a *loss function* and the overarching objective of training a neural network is to minimize this loss. After calculating the loss, it is back-propagated through the network, allowing for computation of the gradient of the loss function concerning all neuronal weights and biases. The computed gradient holds both the direction and magnitude of changes needed to minimize the loss. Leveraging these gradients, *gradient descent* optimization algorithms, explained in section 2.6.5, adjust the weights and biases, and guide the network towards better performance by making refinements in the most beneficial direction. The above mentioned steps are then repeated for another set of inputs.

When training a neural network, selecting the appropriate training hyperparameters is crucial. These are settings or configurations established prior to training, which dictate

how the model learns. Unlike model weights and biases, hyperparameters are not learned from the data but are essential for tuning and optimizing the model's performance. These settings often depend on the task at hand, and there is rarely a one size fits all approach in choosing them, therefore the model developer must play around to find the best one [38]. Commonly tuned training hyperparameters are learning rate, batch size and network architecture. Learning rate is a parameter which get passed to the optimizing algorithm and determines the step size in each iteration towards minimizing the loss function. If the learning rate is too large, there is a risk for overshooting the loss function minima, and have the model not converging. Batch size indicates the number of separate inputs the model is allowed to process before the model-weights are updated.

During training it is important to ensure that the model neither underfits (fails to capture the underlying patterns) nor overfits (captures noise and is too biased to the training data) [29]. If not, the model will be unable to handle previously unseen data. To control the likelihood of a model underfitting or overfitting to data, its *capacity* can be altered. The capacity of a neural network can be thought of as the ability of the network to fit a variety of functions to the data. An illustration of this can be seen in Figure 10. If a model is of low capacity, it may struggle to find a function which fits the training data, and if it is of high capacity it may remember features of the training data which are not useful when trying to fit the test data. ML models generally perform best when their capacity is on par with the complexity of the task. The more complex the task, the higher capacity the model must have. To control the capacity, one can decide the model's *hypothesis space*, which is the set of functions the model is allowed to choose from to solve a given problem. The larger the hypothesis space, the more functions it entails and the higher capacity the model is said to have. Overfitting can be observed during training as a decrease in loss on training data simultaneous to an increase in loss on the validation data, as visualized in Figure 11, indicative of the models proficiency on familiar versus new data being different.

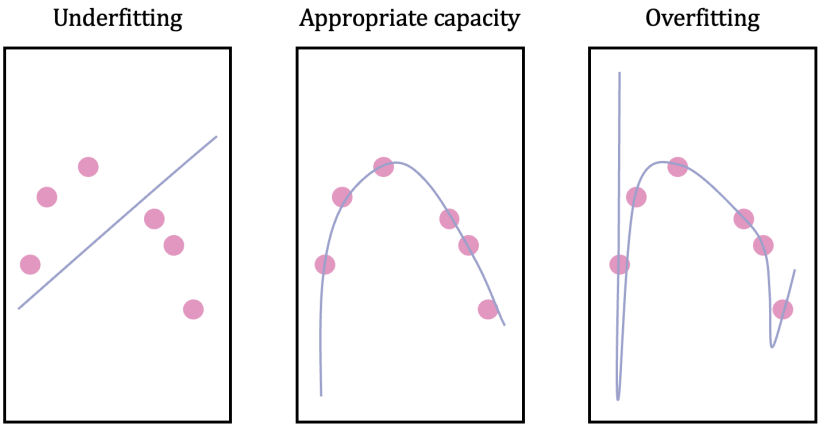


Figure 10: An illustration highlighting the how models of capacity fit to data inspired by [29]. Low capacity (left) tend to lead to underfitting, missing key data patterns, while high capacity (right) risks overfitting, being biased towards training data and losing generalizability. Appropriate capacity is seen in the image in the center.

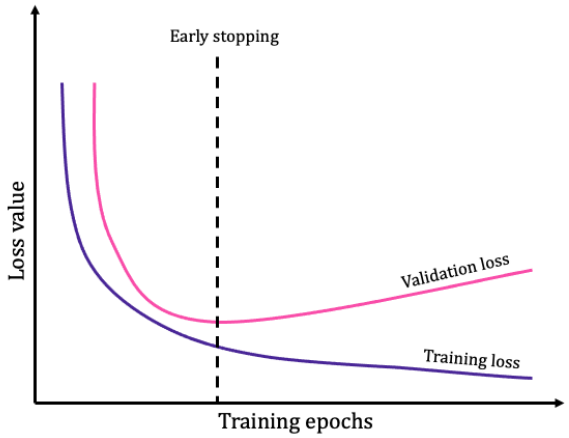


Figure 11: Illustration of training and validation loss behaviour as a network becomes overfitted to the training data inspired by [39]. After the early stopping mark, the training loss continues to decrease while validation loss increases, indicative of the model getting worse at adapting to unseen data.

Furthermore, the importance of a high-quality and large enough dataset must also be highlighted. A sufficiently large dataset ensures that the model is exposed to a wide range of scenarios, which is crucial for it to learn and generalize effectively [40]. The diversity within the dataset is as important as its size; it needs to be representative of the real-world scenarios the model will encounter. This diversity helps mitigating biases and improves the model's ability to perform well on new, unseen data [41]. The quality of the data also plays an important role. Data should be clean, well-labeled, and free from errors and biases as much as possible. Poor quality data can lead to a model learning incorrect patterns, which can severely worsen its performance on real-world tasks [40].

While training, it is also necessary to avoid the parameter gradient from becoming too large or too small. A large gradient parameter update could propel the parameters into a region of the loss landscape where the objective function is larger, thereby undoing much of the progress that had been made in reaching the current solution [30]. Moreover, if the gradient descent parameter becomes excessively large, it can exceed its numerical precision, a scenario that would cause the learning process to fail. When gradients become too large for the network to handle, this is known as *gradient explosion*. Gradient explosion can lead to instability in training and prevent the network from converging to a suitable solution. To avoid this, one may utilize gradient clipping, a technique involving a gradient threshold value. If the gradient exceeds this threshold, it is scaled down to a suitable range [42]. Gradient clipping not only prevents gradient explosion but also stabilizes training by controlling weight updates and aids in model convergence by avoiding extreme updates. However, the gradient clipping threshold is a hyperparameter that must be tuned.

Small gradients can also be problematic as they may lead to a situation known as *vanishing gradients*, where the weights are not sufficiently updated. At worst, the gradients can become so small that learning halts. Vanishing gradients can be combated by careful consideration of the activation functions used, a topic which will be discussed in Section 2.6.1.

### 2.6.1 Activation Functions

An activation function is in its simplest form a function which decides how a node in a neural network should be activated or if it should be activated at all [43]. In its implementation, an activation function often sums a number of input signals and based on these signals generates an output signal to be sent to the next layer of the neural network. Furthermore an activation function adds non-linearity to a neural network. This is vital because without non-linearity the layers of a neural network could be collapsed into a single layer. Meaning that no matter how intricate its structure it could mathematically be simplified to a linear regression model.

One commonly used activation function is the Rectified Linear Unit (ReLU) function which is defined as $f(x) = \max(0, x)$. The ReLU function is close to a linear function but allows for back-propagation that deactivates nodes with negative output from the linear transformations as

$$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

This characteristic combined with its simplicity makes the ReLU function both a valid activation function and a computationally efficient choice.

A potential issue with the ReLU function is the so called dying ReLU problem. This refers to the case where ReLU activated neurons only output the value 0 regardless of the input values, for example by learning a large negative bias term for its weights, meaning the node is dead. Once the network is in this position it cannot recover because the gradient of the ReLU function at zero is also zero.

Adapted to solve the dying ReLU problem, the Leaky ReLU function, defined as $f(x) = \max(\alpha x, x)$, typically using $\alpha = 0.01$, can effectively back-propagate negative output values as well. This will prevent nodes from becoming inactivated. The derivative of the Leaky ReLU becomes

$$f'(x) = \begin{cases} \alpha, & x < 0 \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

### 2.6.2 Upsampling

Upsampling is an operation which expands the size of an input. Its aim is to increase the spatial resolution of a low-resolution feature map, generating a higher-resolution output [44]. When processing images, upsampling means to increase the size of the image. Below, four upsampling blocks will be introduced.

A Deconv block applies a 2D transposed convolution of the input followed by a ReLU activation. In a transposed convolution, the effects of traditional convolution is reversed, and the dimensions of the output is increased instead of reduced. An example of a transposed convolution of a 2x2 input (purple), padded with a 2x2 border of zeros, with a 3x3 kernel (grey) yielding a 4x4 output (pink) when using a stride of one can be seen in Figure 12 [45]. The values in the kernel are learned during training.

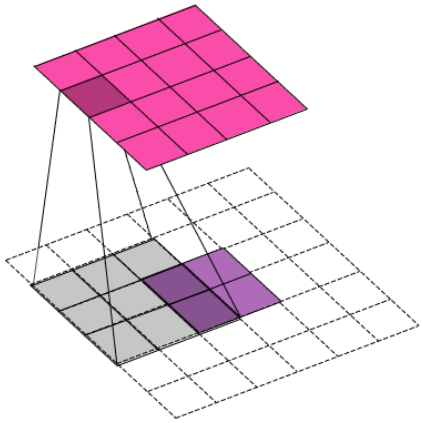

Figure 12: Transposed convolution of a 2x2 input (purple), padded with a 2x2 border of zeros, with a 3x3 kernel (grey) produces a 4x4 output (pink). Image inspired by [45].

An Upsample block applies a 2D nearest neighbor upsampling to an input followed by a 2D convolution and a ReLU activation [46].

According to [47], Pixelshuffle blocks apply a 2D convolution to the input, which increases its depth. It is followed by ReLU activation and a Pixelshuffle layer which rearrange the elements of the tensor from $(*, C \cdot r^2, H, W)$ to a tensor of size $(*, C, H \cdot r, W \times r)$. Here, $*$ denotes the number of elements in the tensor, $C \cdot r^2$ denotes the numbers of channels before upsampling, $H$ and $W$ denote the height and width of the tensor respectively. After upsampling, the number of channels is reduced by $r^2$, and height and width are both multiplied with a factor $r$. This increases spatial resolution while decreasing depth, illustrated in Figure 13. Pixelshuffle is an operation typically applied in super-resolution models to perform sub-pixel resolution convolutions with a stride of $1/r$ [47].

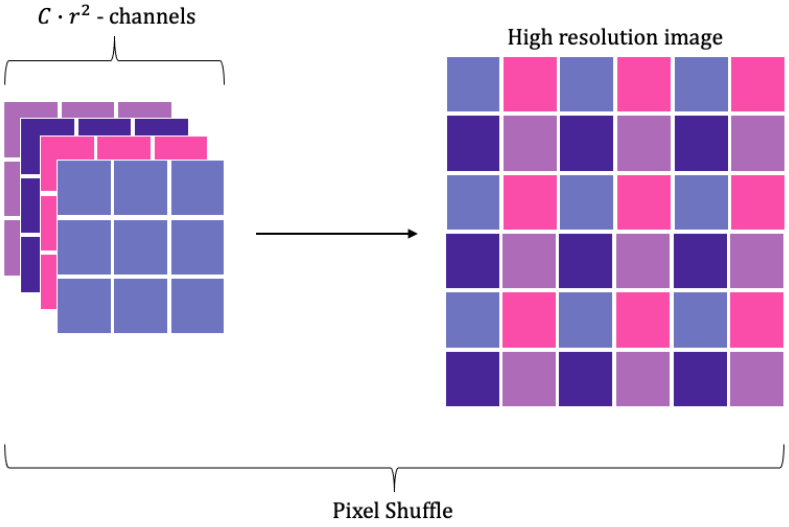


Figure 13: An example of input (left) upsampled with the Pixelshuffle method. $C \cdot r^2$ denotes the numbers of channels before upsampling, and after upsampling the number of channels is reduced to $C$. Image inspired by [48].

As described by [49], the Pixelshuffle Blur block is identical to the Pixelshuffle block with the addition of padding and blurring of the features using an average pooling layer. Blurring the output is suggested to help avoiding checker board patterns in generated images.

### 2.6.3 Batch Normalization

As networks become deeper and consist of multiple interconnected layers, training becomes challenging. One problem is that gradients, which dictate parameter adjustments, operate under the assumption that other layers remain static. Yet, in practice, all layers undergo simultaneous updates. This results in each layer having to navigate a constantly shifting loss function landscape, which lengthens and complicates the training process [30]. Batch normalization is a technique to overcome these challenges. By normalizing the outputs of each layer, they will maintain steady means and standard deviations, making the training process more efficient and stable [50].

Batch normalisation is typically applied before a layers activation function [50]. Given a batch $B$ of size $m$, and the activation from a neuron for the *i-th* data-point in the batch $x_i$, the mean, $\mu_B$, and variance, $\sigma_B$, of the batch's activations can be calculated as

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m}(x_i) \tag{9}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2. \tag{10}$$

These values are then used to normalize the activations

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{11}$$

using $\epsilon$, a small number (typically $10^{-5}$) to avoid division by zero. Lastly, $\hat{x}_i$ is scaled by $\gamma$ and shifted by $\beta$ to yield the normalized output

$$y_i = \gamma\hat{x}_i + \beta \tag{12}$$

which gets passed to the activation function. The hyperparameters $\gamma$ and $\beta$, which adjust the standard deviation and bias of the normalization respectively, are learned via gradient descent during training.

### 2.6.4 Augmentation

Augmenting data is a way to artificially increase the amount of data by altering the original data slightly to enable a machine learning model to extract more information from it. Examples of augmentations include but are not limited to rotations, random cropping and applying filters to the data. When considering what augmentations to use for a specific application it can be beneficial to consider in what way the input data could have been altered but still valid. When, for example, trying to generate images of cells a rotational augmentation could make sense, since the model should be able to recognize a cell in all orientations.

### 2.6.5 Optimization Algorithms

Deep learning optimizing algorithms adjust the model parameters (weights and biases) during training with the aim of improving model performance, $P$. Improvements in $P$ are measured using an external test set, which remains hidden during training, and training is therefore focused on the minimization of a loss function, $L$. The loss, $L$, is calculated on training data with the expectation that the minimization of $L$ correlates with a refinement in $P$.

**Optimization via Gradient Descent**
Machine learning optimization algorithms mainly operate on the principles of gradient descent. It is an iterative methodology which minimizes the loss function by continuously adjusting model parameters in the direction of the steepest descent of the gradient [51]. Consider a loss function represented as $L(x) = y$ and its derivative with respect to $y$ denoted as $\frac{dL}{dy}$. The derivative, indicating the slope of $L(x)$ at $x$, holds information about how a change in $x$ will result in a change in $y$. A consequence of this is that by taking small steps in $x$, opposite to the direction of the derivative, the value of $L(x)$ can be reduced.

For functions with multiple inputs, one instead studies partial derivatives [51]. In such instances, each partial derivative holds information about the loss function's sensitivity to variations in individual parameters. The ensemble of these partial derivatives constitutes the gradient, a vector that points in the direction of the greatest increase of a function and has a magnitude equal to the steepest slope at a particular point. To optimize such a loss function, strategic steps are taken in the direction opposite to the gradient. This process steers the optimization towards function minima.

The problem of optimizing a neural network is generally non-convex and may have multiple local minima, which may result in problems if the optimization algorithms find a local minima with large loss and struggles to move away from it [52]. However, for sufficiently large neural networks, most local minima possess a relatively low loss function value. Researchers also argue that identifying a point within the parameter space that harbors a low (albeit not minimal) loss function value could suffice for effective neural network optimization.

**Adaptive Moment Estimation (Adam)**
Adaptive Moment Estimation (Adam) [53] is an optimization algorithm primarily used for training deep learning models published by Kingma and Ba in 2015. It is a stochastic gradient optimization algorithm, which means that a single training batch is randomly chosen to calculate the gradients and update the model parameters from, introducing randomness into the process, hence the name "stochastic" [54]. Adam combines insights from the AdaGrad and RMSProp optimization algortihms, and is efficient in problems with large dataset [53]. Unlike traditional stochastic gradient descent which utilizes a single learning rate for all weight updates, Adam dynamically adjusts learning rates for each parameter. This property proves beneficial when dealing with sparse gradients, common in fields like computer vision and natural language processing.

Adam synthesizes ideas from both algorithms and estimates the first and second moments

of the gradients, yielding the a rule for updating $x$ as

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\ x_{t+1} &= x_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \end{aligned} \tag{13}$$

The variables $m_t$ and $v_t$ denotes the first- and second-moment vectors at time step $t$. The exponential decay rate for the moments, $\beta_1$ and $\beta_2$, are typically set to 0.9 and 0.999 respectively and $g_t$ denotes the gradient at time step $t$. The $\hat{m}_t$ and $\hat{v}_t$ terms are bias-corrected estimates of the moments to account for their initialization at zero. This bias correction ensures that these estimates are unbiased during the initial time steps. The $\epsilon$ term, often set to $10^{-7}$ ensures numerical stability. [55]

### 2.6.6 Loss Functions

The loss function is a vital part of a neural network. It is a function which measures the difference between the output of the network and the ground truth. It is directly responsible for fitting the output of the neural network to the training data [56]. The goal of the neural network is to minimize the loss function. For optimal performance it is therefore important that an increase in the networks loss function reflects a decrease in the quality of the output and vice versa. It is possible to combine several loss functions into one by means of summation. This also enables weighting of the different components of the loss function. Below are descriptions of the loss functions implemented during this project.

**Mean Square Error**
The mean square error (MSE), also referred to as $\ell_2$, is defined as

$$MSE(\hat{Y}, Y) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{14}$$

where $\hat{Y}$ is the generated image and $Y$ is the ground truth, $N$ is the number of pixels and $\hat{y}_i$ and $y_i$ are the pixel values in the generated image and ground truth image respectively [57].

**Perceptual Loss**
A perceptual loss function is designed to capture semantic information which is not based on pixel-wise differences in an image but rather high level features extracted from pre-trained convolutional neural network, as explained by [58]. An illustration of extraction of these high level features can be found in Figure 14. By using different data for pre-training, different network structures or different intermediate layers to extract information from leads to a large number of perceptual losses to choose between.

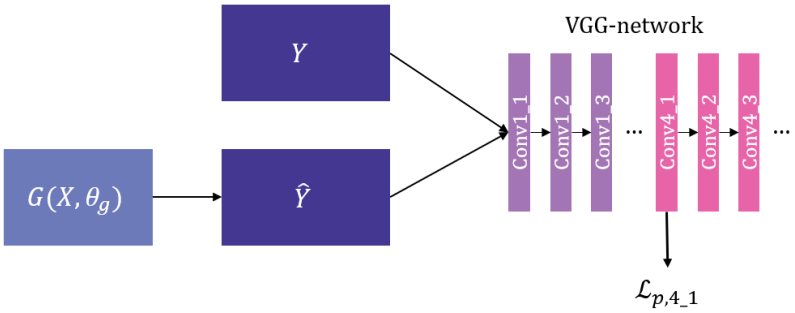


Figure 14: Diagram of how a perceptual loss is retrieved. $Y$ represents the ground truth image and $\hat{Y}$ is the image generated by the generator $G(X, \theta_g)$ with input image $X$ and weights $\theta_g$. Shown to the left are a summary of the convolutional layers of the VGG-network from which various perceptual losses $\mathcal{L}_{p,j}$ can be extracted.

Impressive results were shown by Johnson, Alahi and Fei-Fei [58] of both style-transfer and super-resolution using VGG-networks pretrained on the ImageNet dataset, which containes images of eveyday objects. Their perceptual loss function was designed as follows

$$\mathcal{L}_{p,j} = ||\phi_j(\hat{Y}) - \phi_j(Y)|| \tag{15}$$

where $\hat{Y}$ and $Y$ denote the generated image and ground truth image respectively, $\phi_j$ is the feature map from the $j$-th convolutional layer in the pre-trained network and the norm used was a $\ell_1$-norm.

**Fourier Loss**

The Fourier transform is commonly used to analyze frequency content of signals. The frequency content of an image can be useful to identify textures and patterns in an image. In [59], it was proposed that the Fourier transform can be used to detect high frequency differences in an image which can not be found in the spatial domain. Utilizing the two-dimensional discrete Fourier transform, the complex components of an image can be found as

$$\mathcal{F}(x)_{u,v} = X_{u,v} = \frac{1}{HW} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} x_{h,w} e^{i2\pi(u\frac{h}{H} + v\frac{w}{W})} [59]. \tag{16}$$

Using this, the amplitude of the transform can be calculated as

$$|\mathcal{F}(x)_{u,v}| = |X_{u,v}| = \sqrt{\mathcal{R}(X_{u,v})^2 + \mathcal{I}(X_{u,v})^2}, \tag{17}$$

from which a simplified version of the Fourier loss function $\mathcal{L}_{\mathcal{F}}$ can be constructed as

$$\mathcal{L}_{\mathcal{F}} = \frac{2}{UV} \sum_{u=0}^{U/2-1} \sum_{v=0}^{V-1} \Big| |\hat{Y}|_{u,v} - |Y|_{u,v} \Big|. \tag{18}$$

## 2.7 Generative Neural Networks

As mentioned above, the architecture of a neural network is dependent upon the task it should perform. Networks can be designed to perform image-to-image translation tasks, such as virtual staining. For these kinds of tasks it is common to use convolutional neural networks.

### 2.7.1 Convolutional Neural Network

Convolutional neural networks (CNNs) are specialized neural networks adept at processing data with a known grid-like topology, such as 2D images composed of pixel grids or audio signals [60]. The defining characteristic of CNNs is their use of convolution, a specialized type of linear operation, instead of general matrix multiplication in at least one of their layers [61].

Convolution of two functions $f$ and $g$ is a mathematical operation defined as

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{19}$$

in one dimension. The discrete version, often used in computer vision is defined as

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n-m], \tag{20}$$

with $f$ often being refereed to as the **input**, $g$ the **kernel** and the output as a **feature map** [60]. For multidimensional convolution, such as with images, the equation expands to a multiple summation over the dimensions.

The kernel can be thought of as a feature extractor which get flipped and slid over the signal, computing the sum of a point-wise multiplication at each position [61]. Sliding is done in discrete steps the size of which are denoted *stride*. Figure 15 illustrates performing this operation on a matrix of pixels. The kernel is a two-dimensional array of weights representing parts of the image. Kernels vary in size depending on the application and the desired size of the receptive field. Commonly, 3x3 matrices are used. If the kernel does not fit the input image, one resorts to padding. Padding sets all elements outside of the input matrix to zero to producing an output of correct size. When the kernel is swept across the entire image a feature map highlighting certain features of the image is produced. The kernel-weights are fixed as it moves over the image but adjusted in the backpropagation during training.

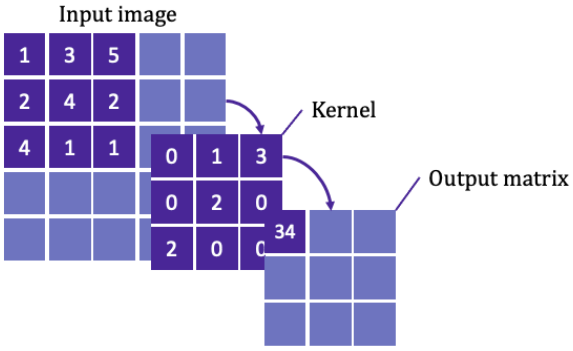


Figure 15: Illustration adapted from [61] showing a convolutional operation where each of the kernel weights are multiplied with their respective pixel-value in the input image and summed to form the first element of the output matrix.

When CNNs are used for image-to-image translation tasks, such as virtual staining, convolutional and pooling layers are connected to extract hierarchical features from input images [61]. The deeper the layer, the more complex and high-level are the features extracted [61]. For instance, early layers might detect edges and colors, while deeper layers might detect shapes or more complex patterns [30]. An illustration of this concept can be seen in Figure 16.

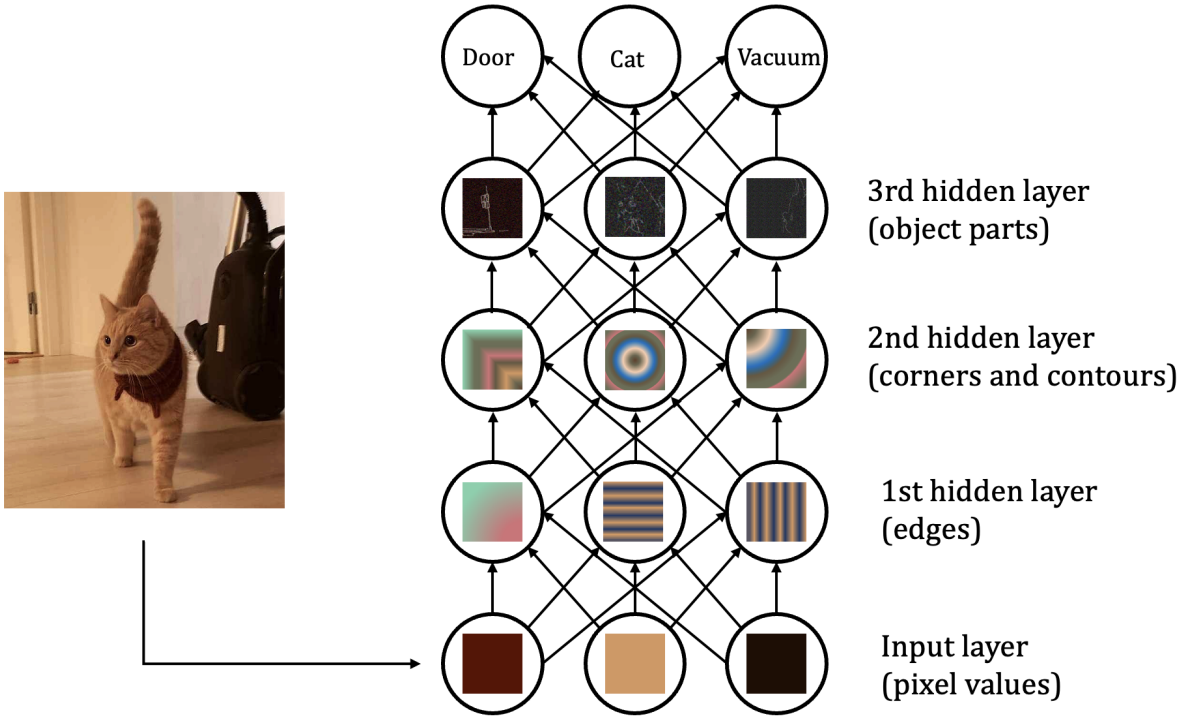


Figure 16: Hierarchical feature extraction from an image of an orange cat with a red scarf in front of a vacuum cleaner. In early layer, low level features such as edges and corners are extracted, while the deeper layer extract high level features, such as object parts. The information is then used for classification. Image inspired by [34].

Pooling, or downsampling, layers perform dimensionality reduction, which reduces the number of parameters in the input [61]. The pooling operation, similar to convolution, is swept across an input, but it does not contain any weights. Instead, it applies an aggregation function to the values within the receptive field to populate the output array. There are two primary types of pooling: max pooling, which selects the pixel with the maximum value, and Average pooling, which calculates the average value within the receptive field. Although pooling layers result in the loss of some information, they may offer benefits like reduced computational complexity, improved efficiency, and limited risk of overfitting.

### 2.7.2 UNet

A UNet is a CNN developed for biomedical image segmentation tasks, introduced by [62] in 2015. The network is known for its effectiveness in segmenting complex images where the location of the target object within the image is not known in advance. It was

developed in response to the need for more efficient and accurate segmentation of cellular structures in medical images. The architecture of UNet is characterized by its U-shaped structure shown in Figure 17, which consists of two main parts, a contractive and an expansive path.

The contracting path of the network, as explained by [62], captures the context in the input image, and is therefore often referred to as the *encoder*. It consists of a series of blocks, each comprising convolutional and max pooling layers that reduce the spatial dimensions of the image while increasing the depth, i.e. the number of feature channels. Each convolution layer is followed by an activation function.

According to [62], the expansive path enables precise localization by the use of transposed convolutions. Referred to as the *decoder*, it reconstructs the segmented image from the encoded features obtained after the contracting path. The spatial dimensions are progressively increased while the depth is reduced, leading to a segmented output. Each decoder-block is composed of an upsampling layer, a convolutional layer (known as up-convolution), a concatenation of the cropped feature map from the contracting path, followed by two convolutional layers with activation functions. The upsampling layer increases the spatial dimensions (height and width) of the feature map, bringing it closer to the original input size [44]. The up-convolution decreases the number of feature channels. The concatenation, known as skip connections, helps the network with precise segmentation by combining low-level, high-resolution features from the contracting path with high-level, abstract features from the expansive path, thus avoiding the loss of spatial information during downsampling. The final layer of the decoder path is a convolutional layer, mapping the feature vector to the desired number of classes.

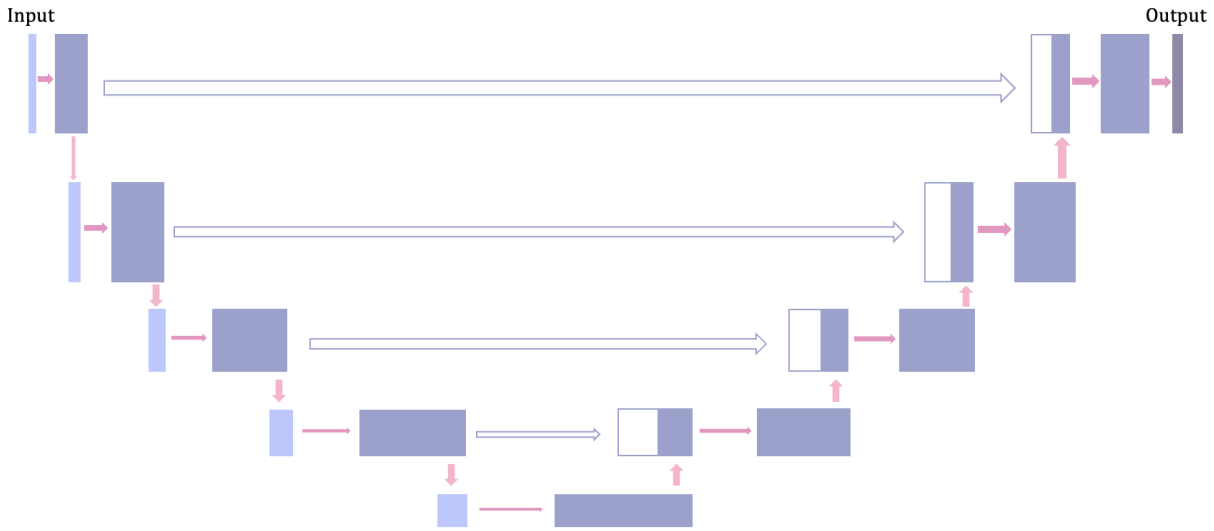


Figure 17: The architecture of a UNet is U-shaped and consists of a contractive (left) and an expansive path (right). The contractive path consists of four blocks of two 3x3 convolution layers each followed by ReLU activation. The output from these blocks are maxpooled (pink vertical downwards arrows). The expansive path consists of four blocks composed of an upsampling operation (pink vertical upwards arrows), a concatenation of the features from the contractive path, known as a skip connection (white horizontal arrow), and two 3x3 convolutions followed by ReLU activations. Image adapted from [62].

### 2.7.3 Dense UNet

The Dense UNet combines the ideas and structural elements of the UNet and a Dense Convolutional Network (DenseNet). It was introduced by [63] for *in vivo* cellular image segmentation. The architecture is similar to a UNet with a contracting and a expansive path, but the blocks are replaced with Dense Blocks, following a DenseNet strategy. The strategy includes all layers within a Dense Block receiving the concatenated feature maps from all previous layers within the same block. This is visually explained in Figure 18. A Dense Block is composed of four densely connected layers (DCLs). Each DCL in the block follows a specific seven-layer structure: it begins with batch normalization, followed by a 1x1 convolution, and then a ReLU activation function. This sequence is repeated with another batch normalization, a 3x3 convolution, and another ReLU activation. The final layer in each DCL is a dropout layer, incorporated to prevent overfitting. The design of a Dense Block is believed to aid in segmentation tasks by letting deep layers have access to the more basic feature information captured by earlier layers. This setup gives the deeper layers access to a mixture of simple and complex features, enabling the network to process a broad spectrum of information simultaneously. As a result, the network can make decisions based on a detailed understanding of both low-level and high-level data. It is also believed that the Dense Block architecture minimize the occurrence of vanishing gradients.

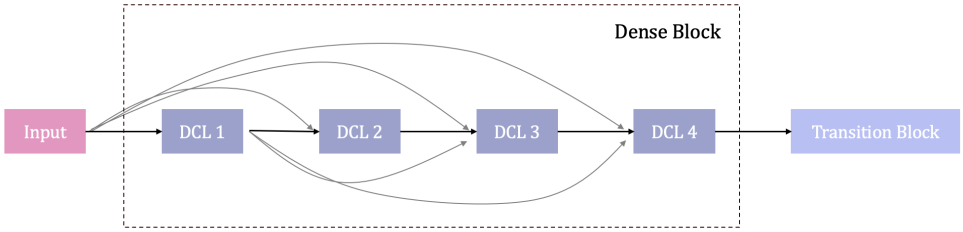


Figure 18: Visualisation of the basic structure of a Dense Block within a Dense UNet. DCL denotes a densely connected block consisting of batch normalizations, activations, convolutions and drop out layers. Image adapted from [63].

### 2.7.4 Attention UNet

An Attention UNet is an enhanced version of the standard UNet architecture, incorporating attention gate mechanisms to improve its performance in image segmentation [64].

The architecture presented by [64] integrates attention gates into the UNet framework, just before the concatenation of the skip connections. These gates are designed to enable the network to selectively concentrate on the most relevant features of the input images during training. Essentially, the model learns to identify and highlight features that are crucial for its specific segmentation task, while actively suppressing less relevant ones. This approach, as explained by [65], helps in reducing the computational resources expended on processing irrelevant activations, thereby enhancing the efficiency of the model. Additionally, this targeted processing provides the network with better generalization power, as it becomes proficient in distinguishing and prioritizing the most informative aspects of the input data, a skill beneficial for biomedical image segmentation, where the ability to accurately focus on relevant features within complex images is crucial.

The mathematical representation of an attention gate is shown below, in Equation 21. The input feature map is denoted $X$, $G$ is a gating function containing contextual information about what features to focus on. The transforms $\phi_x$ and $\phi_g$ are applied to $X$ and $G$. The feature map and the gating signal are combined, passed through a non-linear activation function, $\delta$, and linearly transformed by $\phi$. Lastly a sigmoid function, $\sigma$, is applied to output an attention coefficient map, $A$ as

$$A = \sigma(\phi(\delta(\phi_x(X) + \phi_g(G)))). \tag{21}$$

The calculated attention coefficient map is then used to gate the feature map,

$$Y = AX,$$

to yield the gated output $Y$. In Figure 19, an attention gate is visualized.

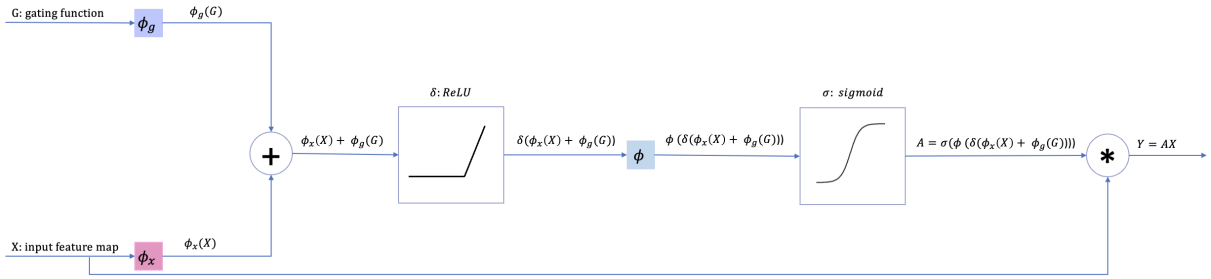


Figure 19: Visual illustration of an attention gate which can be added to the skip connection in a UNet to create an Attention UNet. The variable $G$ denotes a gating function, $X$ an input feature map and $\phi_x$. The gating function and the input feature map are transformed by $\phi_g$ and summed, ReLU activated and transformed by $\phi$. A sigmoid function is then applied to form $A$, an attention coefficient map. The coefficient map is used to gate the feature map to produce a gated output $Y$. Image adapted from [64].

## 2.8 Generative Adversarial Network

Generative Adversarial Networks (GANs) are ANNs developed by [66] in 2014 as a means to improve generative models. A GAN consists of two neural networks trained simultaneously, a generator, $G$, and a discriminator, $D$. The generator is designed to produce data which mimics real data, thus increasing the likelihood that generated data is perceived as real. In contrast, the discriminator is trained to effectively classify data as real or generated by $G$. The two networks are trained simultaneously but with different aims. $G$ tries to minimize $\log(1 - D(G(z, \theta_g)))$. The function $D(G(z, \theta_g))$ represents the discriminator's assessment of whether the generated data is real, with $z$ denoting the generator input and $\theta_g$ the generator weights. A discriminator output of 1 suggests that the data is real, while 0 indicate it being generated. Therefore, a minimization of the expression means that the discriminator classifies the generated images as real. $D$, on the other hand, aims to maximize its classifying accuracy by maximizing $\log(D(x, \theta_d))$ and $\log(1 - D(G(z, \theta_g)))$, with $x$ denoting real data sampled from the distribution $p_{\text{data}(x)}$. This minmax game has the following value function

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x, \theta_d)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z, \theta_g)))]. \tag{22}$$

The competition between the networks and their opposite aims is what makes GANs adversarial, which was nicely explained by [66]:

> "The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency."

Training a generator alongside a discriminator encourages the generator to produce more realistic outputs. This aspect has made the GAN architecture particularly useful for tasks like virtual staining, where GANs excel in generating images that are perceptually more realistic and higher in resolution [8, 10]. In supervised virtual staining tasks, the generator is trained to minimize its adversarial loss in addition to eventual other loss functions. This approach helps to prevent micro-scale hallucinations and ensures that the generated images closely correspond to their ground truths [8]. A basic illustration of a GAN is shown in Figure 20. The generator $G(z, \theta_g)$, receiving input $z$ and having weights $\theta_g$, attempts to produce data mimicking real data. The discriminator $D_{x=Y,\hat{Y}}(x, \theta_d)$, where $x$ denotes either real data or generated data and $\theta_d$ the discriminator weight outputs the probability of the images passed to it being real, i.e. from $p_{data}$, using a loss function. The results of this evaluation is then backpropagated to the generator and discriminator respectively, and the two networks are updating their weights according to their different aims.

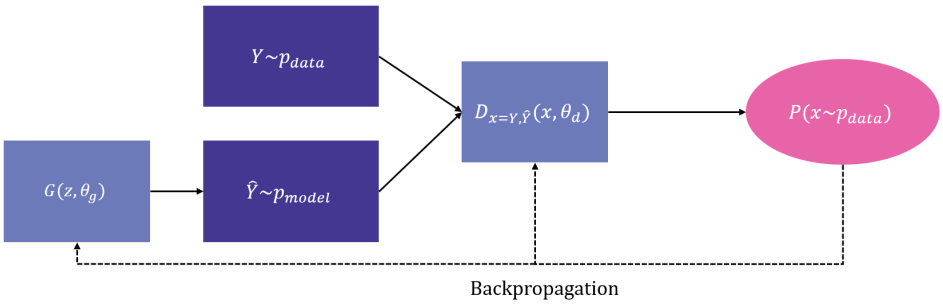


Figure 20: Basic structure of a GAN. The generator $G(z, \theta_g)$, with input $z$ and weights $\theta_g$, produces a generated image $\hat{Y}$. The ground truth image $Y$, drawn from the distribution $p_{data}$, and the generated image $\hat{Y}$, from the distribution $p_{model}$, are evaluated by the discriminator $D_{x=Y,\hat{Y}}(x, \theta_d)$, where $\theta_d$ denote the discriminator weights. The discriminator outputs a probability that the image sent to it was a real image from $p_{data}$, through a loss function which is then backpropagated to the generator and discriminator respectively. Inspired by [15].

### 2.8.1 Training of a Generative Adversarial Network

When training GANs, Equation 22 is split into two loss functions, one for the generator, $L_G$, and one for the discriminator, $L_D$, which they try to minimize during training.

$$L_G = -\mathbb{E}_{z \sim p_z(z)}\Big[\log D(G(z, \theta_g))\Big], \tag{23}$$

$$L_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)}\Big[\log D(x, \theta_d)\Big] - \mathbb{E}_{z \sim p_z(z)}\Big[\log(1 - D(G(z, \theta_g)))\Big]. \tag{24}$$

In code, these loss functions are easily implemented as binary cross entrop (BCE) losses, which, as explained by [67], is a commonly used loss function for binary classifications tasks. The BCE loss quantifies the difference between a true label and the predicted probability, and is defined as

$$BCE(y, p(y)) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i)) \tag{25}$$

where $y$ is the true labels and $p(y)$ the predicted labels.

GANs implemented using the above loss function are commonly referred to as standard-GANs or SGANs. Training a GAN in the "standard" way has been found to be unstable and not converging [68], as well as prone to mode collapse and problematic gradients [69]. Mode collapse is a phenomena where the generator produces the same output over and over, often due to it having found a way to fool the discriminator and then over-optimizing towards these outputs. If the discriminator is not able to learn that these generated outputs are fake, the GAN may stay stuck in a cycle where only one kind of outputs are generated and the modes collapsed [69]. When training GANs, vanishing or saturated gradients can occur if the discriminator has become too effective and easily distinguishes between real and generated data. This will make the gradients very small and the network will struggle to improve [68].

To combat these challenges, the relativistic GAN loss was introduced by [68] in 2018. In a relativistic GAN (RGAN), the discriminator instead estimates the probability that the real data is more realistic than the generated data. The loss functions are instead written as

$$L_G^{RSGAN} = -\mathbb{E}_{z\sim p_z(z), x\sim p_{\text{data}(x)}}\Big[\log(\sigma(D(G(z,\theta_g)) - D(x,\theta_d)))\Big], \tag{26}$$

$$\begin{aligned} L_D^{RSGAN} = &- \mathbb{E}_{x\sim p_{\text{data}(x)}}\left[\log\left(\sigma(D(G(z,\theta_g)) - D(x,\theta_d))\right)\right] \\ &- \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - \sigma(D(G(z,\theta_g)) - D(x,\theta_d))\right)\right], \end{aligned} \tag{27}$$

where $\sigma$ denotes a sigmoid function.

There are many ways to introduce a discriminator to the process of training a generator. The discriminator can be allowed to learn and influence training from the beginning but one can also allow the generator to reach its peak potential and then introduce the discriminator. If the discriminator is introduced at the beginning of training, the difference between the generated images and the ground truth will be large, making the job easier for the discriminator [70]. This may lead to the discriminator quickly becoming proficient in recognizing all generated images as fakes, which in turn will lead to vanishing gradients. In summary it can be a delicate process to ensure that the generator and discriminator are balanced, and to ensure that the generator benefits from the discriminator's feedback without getting lost in its loss landscape.

### 2.8.2 Convolutional Discriminator

The Convolutional Discriminator used in this project is constructed from a series of batch-normalized convolutional layers activated with ReLU. The discriminator processes an image and produces a single scalar classification output, indicative of whether the input image is generated or real. By applying a sigmoid function to the output number, one can determine the network's probability of the image being real or fake.

### 2.8.3 Patch-GAN Discriminator

A Patch-GAN Discriminator is a version of a Convolutional Discriminator [71]. Unlike a standard Convolutional Discriminator, the Patch-GAN discriminator studies NxN patches of the input image individually and outputs a two-dimensional matrix with a classification score for each patch, and therefore only penalizes structural differences on the scale of the image patches. Implementation of this type of discriminator in competition with a UNet generator has been shown by [71] to be effective at, among other things, colorizing images.

### 2.8.4 UNet Discriminator

The UNet Discriminator is based on a UNet structure, see 2.7.2, and the specific architecture was introduced in [72]. They concluded that this discriminator had increased performance on small details in real-life images and also reduced the occurrence of artifacts.

## 2.9 Evaluation Metrics for Generative Machine Learning Models

In machine learning, the training process is iterative, involving continuous assessment and refinement of the model's performance until satisfactory results are achieved. Quantitative measurements, commonly referred to as *evaluation metrics*, are essential for evaluating a model's effectiveness and performance. This section addresses the mathematical expressions for the various metrics employed in this project.

### 2.9.1 Laplacian Sharpness

Laplacian Sharpness is a metric used to evaluate the sharpness or level of focus in an image. It makes use of the Laplacian operator, $\Delta$, a second order differential operator providing information about the rate at which a function is changing. In the context of image processing, it is useful for edge detection and to identify objects and certain patterns. By studying Laplacian operator's variance over an image one can deduce the sharpness of an image, since a high variance typically corresponds to an image with sharp edges, indicating a sharper image.

The Laplacian operator is defined as the divergence ($\nabla\cdot$) of the gradient ($\nabla f$) of a function $f(x, y)$. In a two-dimensional space, the Laplacian of a function is defined as the sum of the second partial derivatives with respect to $x$ and $y$ as

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{28}$$

By analyzing and comparing the variance, $\sigma^2$, of the Laplacian between generated and ground truth images, one can determine the relative sharpness, identifying whether the generated image, $\hat{Y}$, is sharper or more blurred compared to the ground truth image, $Y$. This computation of the relative Laplacian Sharpness (LS) between a generated image $\hat{Y}$ and a ground truth image $Y$ is shown in Equation 29. A negative value indicates that the generated image is less sharp than the ground truth image.

$$LS = \sigma^2_{\Delta\hat{Y}} - \sigma^2_{\Delta Y}. \tag{29}$$

### 2.9.2 Structural Similarity Index

The structural similarity index (SSIM) is a metric to measure the similarity between two images where the value 1 indicates perfect correlation, 0 indicates no correlation, and -1 indicates perfect anti-correlation [73]. It is a perception-based metric developed to provide a more accurate and meaningful assessment of image quality aligned with human visual perception. By considering changes in structural information, luminance, and contrast, which are crucial elements in how humans perceive visual quality [73], SSIM is a widely used metric [74].

Structural information, central to SSIM, refers to the idea that pixels within an image have strong inter-dependencies, especially when spatially close [73]. These dependencies form patterns that carry significant information about the structure of objects in the visual scene, such as edges, textures, and shapes. By assessing this structural information, SSIM effectively captures the essence of an image as perceived by the human eye.

In addition to structure, SSIM evaluates the luminance, which measures the average brightness, and the contrast, which assesses the difference in brightness between objects and their surroundings. These elements are crucial as our visual system is highly sensitive to variations in brightness and contrast, using them to extract detailed information from our environment. The formula for SSIM is

$$\text{SSIM}(\hat{Y}, Y) = \frac{(2\mu_{\hat{Y}}\mu_Y + c_1)(2\sigma_{\hat{Y}Y} + c_2)}{(\mu^2_{\hat{Y}} + \mu^2_Y + c_1)(\sigma^2_{\hat{Y}} + \sigma^2_Y + c_2)}, \tag{30}$$

where $\hat{Y}$ and $Y$ denote the images being compared, $\mu_Y$ and $\mu_{\hat{Y}}$ denote the luminance, $\sigma^2_{\hat{Y}}$ and $\sigma^2_Y$ the variance of $\hat{Y}$ and $Y$ respectively, and $\sigma_{\hat{Y}Y}$ the covariance of $\hat{Y}$ and $Y$. The constants $c_1$ and $c_2$ are included to prevent division by zero and provide numerical stability.

SSIM performs comparisons on a local basis, therefore, spatial shifts between the images significantly affect the outcome [75]. Misaligned images can have vastly different local pixel values even if their contents are the same, resulting in a lower SSIM score. Another aspect to consider when using SSIM is that it focuses on contrast and luminance, it may have trouble catching color distortions.

### 2.9.3 Complex Wavelet Structural Similarity Index

The complex wavelet structural similarity index (CW-SSIM) is, according to [75], an extension of the traditional SSIM into the complex wavelet domain. This adaptation is designed to handle scenarios involving image scaling, translation, and rotation. Unlike SSIM, which requires precise pixel-level alignment, CW-SSIM takes advantage of the Complex Wavelet Transform (CWT) to provide more accurate similarity scores even when images are not perfectly aligned. Similar to SSIM, the maximum value, 1, of CW-SSIM indicates a perfect structural match.

Wavelet transforms can, according to [76] be used to decompose signals into sets of wavelets, which are wave-like oscillations located in time. A wavelet $\psi(t)$ has zero mean, $\int_{-\infty}^{\infty} \psi(t)\, dt = 0$, and finite energy, $\int_{-\infty}^{\infty} |\psi(t)|^2\, dt < \infty$. By applying wavelet transforms to signals one can deduce information about its time, location, and frequency [76]. Neurons in the primary visual cortex apply wavelet transform to extract information from our surroundings, and similarly to how ANNs are based on mimicking biological phenomena, so does CW-SSIM [75].

As explained by [75], CWT enhances the standard wavelet transform by using complex-valued wavelets, enabling the capture of both amplitude and phase information from an image. This capability makes CWT effective in preserving the structural and textural integrity of images under various geometric transformations. CW-SSIM operates on the principle that small geometric distortions in images result in consistent phase shifts in the local wavelet coefficients, without significantly altering the structural content of the image. This focus on phase rather than amplitude information enables CW-SSIM to be more tolerant of misalignment and distortions.

According to [75], the mathematical formulation of CW-SSIM is

$$\text{CW-SSIM}(\mathbf{c}_x, \mathbf{c}_y) = \frac{2|\sum_{i=1}^{N} c_{x,i} c_{y,i}^*| + K}{\sum_{i=1}^{N} |c_{x,i}|^2 + \sum_{i=1}^{N} |c_{y,i}|^2 + K}, \tag{31}$$

where $\mathbf{c}_x = \{c_{x,i} | i = 1, ..., N\}$ and $\mathbf{c}_y = \{c_{y,i} | i = 1, ..., N\}$ are the local complex wavelet coefficients of images $x$ and $y$ respectively, $c^*$ denotes the complex conjugate of $c$, $N$ is the total number of coefficients considered, and $K$ is a small positive constant for numerical stability.

### 2.9.4 Fréchet Inception Distance

The Fréchet inception distance (FID) is a metric which evaluates the distance between feature distributions of real and generated images to assess how "real" the generated images are [77]. The technique can be used to evaluate the quality of images produced by generative models, such as GANs, where traditional metrics often fall short. This metric is able to both detect if generated images are of lower quality, for example containing noise and blur, and if the generated images are not realistic, such as portraying a person with too many fingers or a dog and cat hybrid.

The generated and the real images are both passed through a pre-trained image classification network to extract their features. The distribution of features are multidimensional Gaussian, and the squared Fréchet distance between the two distributions, $\mathcal{N}(\mu_{\hat{Y}}, \Sigma_{\hat{Y}})$ and $\mathcal{N}(\mu_Y, \Sigma_Y)$, from the generated and the real image respectively, can be calculated as

$$d_F((\mathcal{N}(\mu_{\hat{Y}}, \Sigma_{\hat{Y}}), \mathcal{N}(\mu_Y, \Sigma_Y)) = \|\mu_{\hat{Y}} - \mu_Y\|_2^2 + \operatorname{tr}\left(\Sigma_{\hat{Y}} + \Sigma_Y - 2(\Sigma_{\hat{Y}}\Sigma_Y)^{\frac{1}{2}}\right). \tag{32}$$

A FID distance of 0 indicates that the generated images are indistinguishable from real images. The larger the score, the less realistic the image is.

# 3 Methods

## 3.1 Data Collection

The data collected for this project consisted of a PLS stack of images of unstained skin tissue and bright-field images of the same tissue stained with H&E. All images were taken using a version of CellaVision's DC-1 machine with 10x magnification and field-of-view (FOV) images of size 1920x1200. A 3-dimensional coordinate system was used to maneuver with the DC-1 . The z-coordinate denotes the depth along the optical axis, while the x- and y-coordinates facilitate movement within the plane parallel to the slide, allowing navigation across its area.

The data is divided into two datasets, summarised in Tables 1 and 4. Unless otherwise specified, the method of data collection remained the same for both datasets. The second dataset was gathered to enhance variability in terms of patients, pathology, and biopsies. All tissue used was approximately 4 $\mu$m thick cuts from a paraffin embedded section of human skin and was fixated on glass microscopy slides, hereafter referred to simply as slides.

The slides of the first dataset had consecutive cuts of tissue from the same biopsy for each patient. There were three patients in the dataset and 10-19 slides from each patient was used. Each cut had an approximate cross-section area of 0.3 $\mathrm{cm}^2$. The consecutive cuts and small area of the cuts meant that the variation in the data collected from each patient was limited. Furthermore, the small number of patients also limited variation. All skin used in the first dataset was healthy skin tissue containing melanin. No information about the patients' ages was available.

Table 1: Summary of first dataset.

| | **Number of Slides** | **Number of FOV Images** |
|---|---|---|
| **Patient 1** | 10 | 170 |
| **Patient 2** | 14 | 296 |
| **Patient 3** | 19 | 319 |

The second dataset consisted of slides from 11 patients that were not part of the first dataset. Consecutive cuts from the same biopsy were also used for the second dataset. However, only two slides from each patient was used. The second dataset contained tissue from 11 different patients with an average tissue cross-section area of 3 $\mathrm{cm}^2$. Out of the patients, six were deemed healthy, three were diagnosed with basal cell carcinoma and two with squamous cell carcinoma.

Table 2: Summary of second dataset.

| | Age | Pathology | Number of Slides | Number of FOV Images |
|---|---|---|---|---|
| **Patient 1** | 73 | Normal | 2 | 117 |
| **Patient 2** | 55 | Normal | 2 | 108 |
| **Patient 3** | 66 | Normal | 2 | 88 |
| **Patient 4** | 73 | Basal Cell Carcinoma | 3 | 93 |
| **Patient 5** | - | Normal | 1 | 13 |
| **Patient 6** | 55 | Normal | 2 | 89 |
| **Patient 7** | 69 | Basal Cell Carcinoma | 2 | 287 |
| **Patient 8** | 44 | Squamous Cell Carcinoma | 2 | 248 |
| **Patient 9** | 67 | Normal | 2 | 67 |
| **Patient 10** | 75 | Basal Cell Carcinoma | 2 | 130 |
| **Patient 11** | 62 | Squamous Cell Carcinoma | 2 | 224 |

### 3.1.1 Unstained PLS scan

An attempt was made to take images of the paraffin embedded tissue directly but it was soon discovered that the paraffin both created dark outlines along the edges of the tissue and was not reliably even to take focused images. It was therefore decided that the tissue should be deparaffinized and protected by a cover slip. This both reduced the appearance of shadows along the edges and improved the ability to focus well on the tissue in an entire FOV with one z-coordinate setting.

Due to the very pale appearance of the unstained tissue compared to the background of the slide it was challenging to find a quantitative focus metric that worked well to automatically find the best focus for a FOV. Among other things a Laplace focus score was evaluated but unsuccessfully. Instead the z-coordinate for each slide was qualitatively selected by examining the produced images. The area of the slide that the sample occupied was then identified, and images were captured in a rectangular region enclosing this defined $xy$-area.

Once the 3D-coordinates for the scan were determined, a PLS scan was done. The LED's chosen were from a typical configuration to artificially increase NA in the field of pathology. This included one image with normal BF illumination, three with both BF and DF illumination and 20 fully BF. All but the BF only image were taken with singular LED's from the PLS array. In choosing the light calibration for each of the LEDs, the objective was to minimize over-exposed areas while maximizing the illumination of the most transparent areas which do not scatter as much light.

### 3.1.2 H&E Staining

For the first dataset, the H&E staining was done in-house at CellaVision, using the staining protocol found in Appendix A.1.1. Several protocols for staining were attempted but the intensity of the stain's colors remained a problem. This meant that the stain used for the first dataset was pale compared to the ideal H&E stain. The reason for this is

believed to be an imbalance between the strength of the hematoxylin and eosin's staining efficiency. In discussion with a biomedical scientist and pathologist we got the feedback that this stain was non-ideal for a pathologist due to the low contrast but that all essential structures for an analysis in 10x magnification were visible although paler than the norm. With this information in mind we moved ahead with the first dataset to be able to begin training the ANNs, with the hopes of improving stain quality with a second round of data collection.

The second dataset was instead H&E stained at the Department of Clinical Pathology and Cytology at Blekinge hospital. Their lab has extensive experience with H&E staining and were able to efficiently and consistently stain the samples with better color vibrancy and contrast. The staining protocol used can be found in Appendix A.1.2.

### 3.1.3 Stained Scan

Starting at the z-coordinate used for the slide when unstained, some fine tuning was done to produce optimally focused images once the slide was stained. It was found that the precision of CellaVision's DC-1 was such that no calibration was necessary to take images of the near exact same area of the slide. Only BF images were collected of the slides once H&E stained to be used as ground truth.

## 3.2 Data Processing

Template matching was selected as the method of matching the images of unstained and stained tissue to each other. Although the stained and unstained images were rather different, many of the contours were the same between the two. This allowed for pixel-wise precision of the matching of many image pairs. No visible rotation was observed and the magnification remained constant between the scans, so no scale transformation was deemed necessary. The method of template matching used was cross-correlation. Both images were converted to gray scale for the template match. The stained image was cropped by a set number of pixels on each side and then matched to its unstained counterpart. For most images it was found to be sufficient with a crop of 10 pixels on each side, however some images needed a larger crop (maximum 50 pixels) to match the images. Visual confirmation was used to check that a satisfactory template matching had been achieved.

Using a threshold for the images' Laplace variance, empty FOV images were eliminated if their variance was below the threshold, which was found through visual inspection. Then, the images in the first dataset were divided into 64x64 sections. This was done to make the training of the neural network more efficient. During the division process, each 64x64 image was also eliminated from the dataset if their Laplace variance was too low. A low Laplace variance was used to identify images which contained no tissue, these images were deemed irrelevant for the network to learn from at this stage.

For the second dataset, the images were instead cropped into 128x128 and 256x256 sections and a random crop of the images to 64x64 was instead applied during the training of the model. This will be explained in greater detail in a later section. Furthermore, images of slide-background were collected using the same method used in section 3.1 to remove

empty areas but instead flipping the threshold to only keep images with low Laplace variance.

### 3.2.1 Data Partitioning

For the first dataset, images used for training, validation and test data were chosen with equal distribution from each of the three patients. For each of the patients 70% was allocated to the training data and 15% of the data was allocated to validation and test data respectively. In total 156 341 64x64 images were processed and used to train, validate and test the networks.

Table 3: Summary of first dataset after data processing.

| | **Number of Slides** | **Number of 64x64 images** | **Partition** |
|---|---|---|---|
| **Patient 1** | 10 | 36 656 | 70% Train, 15% Val, 15% Test |
| **Patient 2** | 14 | 58 137 | 70% Train, 15% Val, 15% Test |
| **Patient 3** | 19 | 61 548 | 70% Train, 15% Val, 15% Test |

For the second data set the partitioning of data was stratified based on patient. To ensure that the test data represented a fully external evaluation, three patients were specifically allocated to this partition, including one healthy patient and one patient for each of the respective carcinoma types available in the dataset.

To maintain representation of each type of carcinoma in the validation set, the decision was made to distribute the data from three patients evenly between the training and validation datasets. Given that the full dataset only included two patients with squamous cell carcinoma, this approach was considered the most suitable. The data from these patients were divided in such a way that there was little to no overlap in the area of tissue included in each partition, this is similar to what was done in [10]. The remaining five patients were entirely allocated to the training dataset. A total of 134 818 128x128 images were collected which corresponds to 539 272 unique 64x64 images. For easier comparison with the amount of data from the first dataset, the number included in Table 4 is the unique 64x64 images. As can be calculated from the values in Table 4, 32% of data went to the test set and 68% remained for training and validation.

Table 4: Summary of second dataset after data processing.

| | **Age** | **Pathology** | **Number of Slides** | **Number of 64x64 images** | **Partition** |
|---|---|---|---|---|---|
| **Patient 1** | 73 | Normal | 2 | 37 524 | Train |
| **Patient 2** | 55 | Normal | 2 | 26 864 | Train |
| **Patient 3** | 66 | Normal | 2 | 26 836 | Train |
| **Patient 4** | 73 | Basal Cell Carcinoma | 3 | 40 592 | Train |
| **Patient 5** | - | Normal | 1 | 2344 | Train |
| **Patient 6** | 55 | Normal | 2 | 38 868 | 50% Train, 50% Val |
| **Patient 7** | 69 | Basal Cell Carcinoma | 2 | 98 272 | 50% Train, 50% Val |
| **Patient 8** | 44 | Squamous Cell Carcinoma | 2 | 96 036 | 50% Train, 50% Val |
| **Patient 9** | 67 | Normal | 2 | 28 224 | Test |
| **Patient 10** | 75 | Basal Cell Carcinoma | 2 | 52 224 | Test |
| **Patient 11** | 62 | Squamous Cell Carcinoma | 2 | 91 488 | Test |

### 3.2.2 Augmentations

Different augmentations were experimented with throughout the process in an attempt to find a version that would increase the information extraction from the input data and benefit the generalization of the model.

Random rotational augmentation was implemented by randomly rotating each input image-stack by $n \cdot 90°$ with $n \in \{0, 1, 2, 3\}$.

A random crop augmentation was implemented by randomly generating the $(x, y)$-coordinate for the upper left corner to crop a smaller image which was used as input to the network. With the first dataset only one random crop augmentation was attempted 48x48 from 64x64. The augmentation was however utilized throughout the experiments with the second dataset where all versions of trained networks used at least a 64x64 random crop from 128x128 images. Other random crop augmentations were also explored and in the network specifications for the more complex models it will be specified if a different crop was used.

In the second dataset, images with little or no tissue present, i.e. images of background, were added to increase the models performance in generating images with empty corners and sharp edges. Doing this separately from the initial data processing enabled control over how many background images were included in the training dataset. No background images were added to the validation or test dataset.

## 3.3 Training of the Generative Network

The following sections are dedicated to the experimental task of designing and training an optimal neural network for virtual H&E staining of skin tissue. These sections aim to provide a comprehensive understanding of the methodologies employed.

The experiments were characterized by continuous adaptations and enhancements building upon the outcomes of each successive step. During this process, many structural changes and choices of parameters were evaluated and the solution found to work best in one step of the methodology was built upon in the next. Decision-making regarding network modifications was guided by maximizing CW-SSIM, minimizing the validation loss and training time. CW-SSIM was chosen over SSIM as the main evaluation metric due to its robustness to pixelwise disalignment. Laplace focus was an evaluation metric used at the end of experiments but it was not monitored during the training of the models, this was due to the fact that the Fourier loss was tarcking the same type of performance. The validation loss was also monitored during training for signs of overfitting. Visual inspection of the produced images also became an important tool for evaluating performance changes in specific regions of larger images.

The timeline of the project significantly influenced the training duration for the models. Prioritizing the exploration of various architectures and settings was deemed more beneficial for the project than dedicating extended periods to training a more complex network. Given a longer project timeline, this compromise of opting for shorter training times across multiple models could have been avoided. In each subsequent section, the specific network employed in the respective steps will be clearly identified.

All networks were evaluated and developed with PLS images as input. However, the networks were flexible in that the number of input channels could be adapted while keeping the remaining network and filters the same. In this way, all networks architectures in this thesis could also easily be used with only a BF image as input.

### 3.3.1 Architecture of the Generative Network

To limit the number of parameters needing tuning in the experiments, some parameters were fixed throughout the project. The number of blocks of the networks was fixed to four, and an Adam optimizer was used for all experiments. It was observed early on that all initial networks struggled with instability, and to counteract this, batch normalization and gradient clipping was added.

In the first experiment, six models, **UNet-X** and **Dense UNet-X** with $\mathbf{X} \in \{64, 96, 128\}$ denoting the number of input channels in the first block, were trained and evaluated. The number of input channel experiments were performed to explore if an increase in input channels could help the network retain relevant information from the PLS stack. For all six experiments, an MSE loss function, Deconv blocks for upsampling, and ReLU activation functions were used. The learning rate was fixed at 0.001.

The **UNet** blocks consisted of two convolutional layers, each followed by a ReLU activation function. The convolutional layers used a kernel of size three, a stride of one and had a border of zero padding of width one. The max pooling layers at the end of each block used kernel size two and stride two and had no padding.

In the **Dense UNet** experiments, each block consisted of four dense layers. For each layer in the block, the output from all previous layers within the same block were concatenated to form the input. This input was then convolved with a kernel of size three, stride one and zero padding of width one. The convolved input was then ReLU activated and passed through a max pooling layer, with kernel size two and stride two, before being passed to the subsequent block.

Training time, evaluation scores, particularly CW-SSIM, and overfitting tendencies, were considered for these models. Based on this assessment, the **UNet-96** model was deemed the most effective and was subsequently selected for further experiments.

### 3.3.2 Loss Functions

Experiments involving different loss functions, their weightings, and linear combinations of these elements were performed on the first dataset. In the initial experiments described above, the loss function only consisted of an MSE-function. To enhance the perceptual similarity of the generated images to the ground truth, particularly in a way that resonates with human visual perception, a perceptual loss was added. The features used to calculate the perceptual loss were extracted from the deep $'conv4-1'$ layer of the VGG-19 network. To further improve the quality and structural similarity of the generated images, a Fourier loss was integrated into the training regiment.

To enhance the contrast of the cell nuclei in our images, we developed a weighted MSE loss specifically focused on the nuclei. This was done by thresholding the images to identify areas believed to contain cell nuclei and then applying the MSE loss to these identified regions. Our goal was to more heavily penalize errors in the cell nuclei areas. However, upon visual inspection, it became apparent that this technique led to a decrease in overall image quality, and the nuclei appearing blurred and smudged. Consequently, we decided to discontinue this approach and instead limit our loss functions to the standard MSE, Fourier, and Perceptual 4-1 losses.

Experiments with different weightings of the loss functions were performed. It was found that a 1, 5, 5 weighting of MSE, Perceptual and Fourier respectively, yielded the best results and this weighting was kept for the remainder of the experiments.

### 3.3.3 Activation Functions, Upsampling Blocks and Attention Gates

All experiments until this point, used ReLU activation. No apparent issues arose from this choice but to ensure that dying ReLU was not impairing the results in any undiscovered way, a leaky ReLU activation function, with $\alpha = 0.1$ was applied in future experiments instead.

Four distinct types of upsampling blocks: Deconv, Upsample, PixelShuffle, and PixelShuffle Blur were assessed by training separate models.

At this stage, the generated images lacked some small details. To try to alleviate the problem, attention gates were added to the skip-connections of the networks moving forward.

It is worth mentioning that at this point in the project the second dataset was ready for use. The best network from this section was therefore trained on both the first dataset and second dataset. The models past this point were only trained on the second dataset. It was found that the model performing the best on the first dataset also performed satisfactorily on the second, and therefore little backtracking in the methods so far was done with the second dataset. The one noteworthy difference is the augmentation used, which has been previously covered in Section 3.2.2.

This part of the project culminated in a generator named **UNet-96-MSEpf-Pixelshuffle-Attention**.

### 3.3.4 Addition of a Discriminator

Initially a multitude of experiments were conducted with the typical SGAN loss function. It was found difficult with this type of generator and discriminator interaction to find a discriminator architecture and training method that was both stable and did not lead to either mode collapse or vanishing gradients. The choice was therefore made to implement an RGAN instead, and to introduce the discriminator to the network from the point where our best generator had trained to reach its highest CW-SSIM score before overfitting to the training data.

Three types of discriminator architectures were attempted: a **Conv**olutional discriminator, a **Patch-GAN** discriminator, and a **UNet** discriminator. Among these, the **UNet** discriminator performed the best. The **Conv** discriminator comprised an initial convolutional layer with a 4x4 kernel (stride two, padding one), followed by Leaky ReLU activation. This was followed by four blocks, each including a convolutional layer with a 4x4 kernel (stride two, padding one), a batch normalization layer, and Leaky ReLU activation. The **Conv** discriminator concluded with a final convolutional layer with a 2x2 kernel (stride of one, zero padding). The **Patch-GAN** discriminator began with a convolutional layer with a 4x4 kernel (stride two, padding one) followed by Leaky ReLU activation. This was followed by two blocks, each consisting of a convolutional layer with 4x4 kernels (stride two, padding one) and Leaky ReLU activations, ending with a 4x4 convolution (stride one, padding one) and Leaky ReLU activation. The **UNet** discriminator started with a convolutional layer with a 3x3 kernel (stride one, padding one) and Leaky ReLU activation. Its contracting path included three blocks of convolutional layers with 4x4 kernels (stride two, padding one) activated by Leaky ReLU, and its expansive path comprised three blocks of convolutional layers with 3x3 kernels (stride one, padding one). The upsampled features were then processed by two additional convolutional layers with 3x3 kernels (stride one, padding one), and Leaky ReLU activations, before reaching the final output layer, a convolutional layer with a 3x3 kernel (stride one, padding one). All Leaky ReLU activations used in the discriminators had a slope of 0.2.

Fine-tuning experiments were then performed on the **UNet-Disc-RGAN** where the learning rate of both the generator and discriminator was lowered, for the generator to $lr_G = 10^{-4}$ and $lr_D = 10^{-5}$ for the discriminator. A larger random crop of the input images, from 512x512 to 128x128 was tried, and lastly, a model was trained where the images were converted to the CIE-L*a*b color space. The discriminator was allowed access only to the L*-channel and the generator's MSE loss was calculated on the a*- and b*-channels.

Table 5: Settings for the GAN experiments. Learning rates of the discriminator and the generator are denoted $lr_D$ and $lr_G$ respectively.

| | **Discriminator** | **Random crop** | **Color space** | **Learning rate** |
|---|---|---|---|---|
| **Conv-Disc-RGAN** | Convolutional | 64x64 from 128x128 | RGB | $lr_D = lr_G = 10^{-3}$ |
| **Patch-GAN-Disc-RGAN** | PatchGAN | 64x64 from 128x128 | RGB | $lr_D = lr_G = 10^{-3}$ |
| **UNet-Disc-RGAN** | UNet | 64x64 from 128x128 | RGB | $lr_D = lr_G = 10^{-3}$ |
| **VS-RGAN** | UNet | 128x128 from 512x512 | CIE-L*a*b | $lr_D = 10^{-5}$ $lr_G = 10^{-4}$ |

## 3.4 Evaluation of Generated Images

The evaluation of images generated by the different networks involved both quantitative metrics and a qualitative assessment through a questionnaire administered to pathologists and other trained professionals. The quantitative evaluation was performed on the validation and test data. For the qualitative assessment, large scale images 1816x1096 were generated from the test data patients only.

### 3.4.1 Quantitative Evaluation

CW-SSIM, SSIM and Laplace Sharpness were used to assess the images generated by all models, and evaluation of the FID between the set of generated and ground truth images was calculated for images generated by the best Generator-only network, named **UNet-96-MSEpf-Pixelshuffle-Attention**, and the four GANs in Table 5. For **UNet-96-MSEpf-Pixelshuffle-Attention**, the complete evaluation was performed for networks trained on PLS images and BF images respectively, for all other networks the evaluation was only performed on networks trained using PLS images. Detailed mathematical explanations of these metrics can be found in Section 2.9.

To ensure the reliability of the evaluation metrics, we tested what score they provided when passed various image pairs: two identical images, two vastly different images, an unaligned unstained image and its stained counterpart, and a sharp image alongside its blurred version. These tests confirmed the metrics' proficiency.

For examination of the model's ability to generate specific tissue structures, single-image comparisons were performed. Additionally, color histograms were created to analyze the model's proficiency in accurately replicating colors throughout the entire test dataset compared to their corresponding ground truth.

## 3.5 Post-processing

All metric scores where calculated on unprocessed generated images compared to unprocessed ground truth images. To enhance the appearance of the images and make them more vibrant in the report a saturation enhancement with saturation factor $s = 1.5$ was performed. Further the images shown in the report were sharpened using the unsharp

mask technique. The technique was applied on the images converted to the CIE-L*a*b colorspace and only applied to the L*-channel. The unsharp mask was created with a Gaussian blur filter with kernel size 15.

### 3.5.1 Qualitative Evaluation

For the questionnaire, the images in the test dataset were annotated and categorized as:

- **HH** - Images of healthy tissue from a healthy patient
- **HS** - Images of healthy tissue from patients with carcinoma
- **Sq** - Images containing Squamous Cell Carcinoma
- **Ba** - Images containing Basal Cell Carcinoma

Annotation was done by us with the help of Veronika Jenei, Scientific and Medical Affairs Manager at CellaVision, for full FOV images. All images in the second dataset's external test dataset that were deemed possible to clearly categorize were fed to the **VS-RGAN** network. From each class, the three best and three worst images, according to their CW-SSIM score, were chosen for the questionnaire.

The questionnaire was made to include 50% virtually stained images and 50% chemically stained images. 50% of the images in each of these categories were of healthy tissue, 25% depicted squamous cell carcinoma 25% depicted basal cell carcinoma. An illustration of the image distribution in the questionnaire is shown in Figure 21. All images in the questionnaire were post-processed, by the method explained in Section 2.4.3.

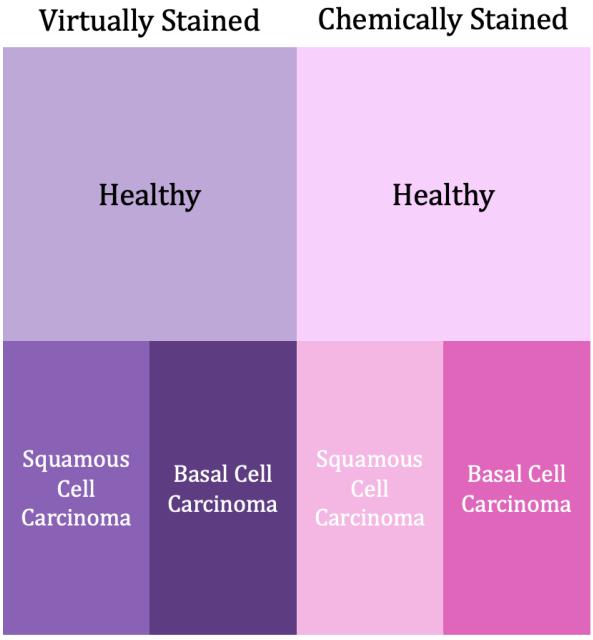


Figure 21: Illustration of the distribution of images in the questionnaire.

Two versions of the questionnaire were made with identical questions. However, the images that were generated in version one were replaced with the ground truth images in

version two, and vice versa. An equal number of each questionnaire was then sent out to the participants.

The questions asked for each of the images were the following:

1. How would you rate the quality of the stain in this image? (scale 1-10)
2. Do you see any cancerous tissue in this image?
3. Would you feel comfortable using images of similar quality as the image above to help you make a diagnosis?

An example of all unique questions asked is available in Appendix A.2.

# 4 Results

## 4.1 Data Collection

Figure 22 shows an example of damaged tissue which has been moved in the staining process (see the upper left corner). From the input image it is clear that the generated image recreates the shape of the tissue well but that the tissue has shifted in the ground truth image. It was noted that both the CW-SSIM and SSIM score were negatively impacted by such disalignment between the images.

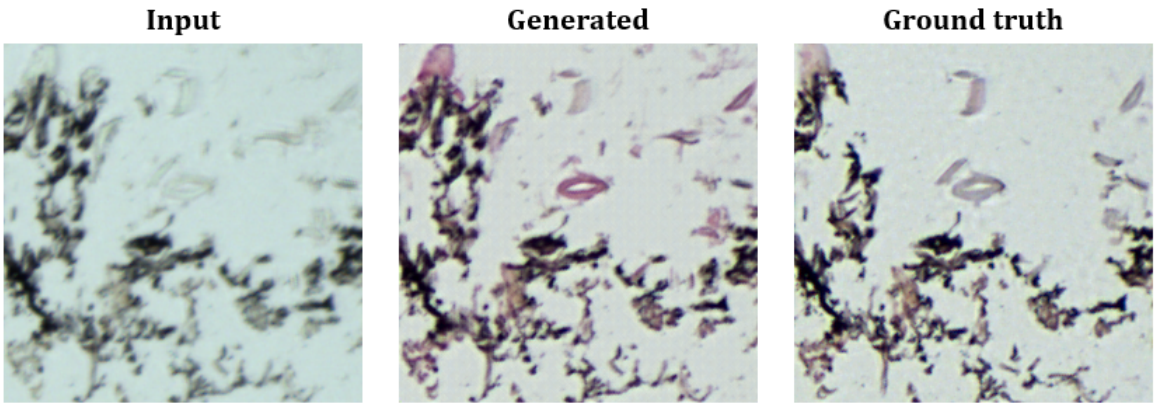


Figure 22: Example of tissue moving as a result of the staining process. The generative network used was **VS-RGAN** and the image got CW-SSIM score 0.5634 and SSIM score 0.348. Image from the second dataset's external test data.

Figure 23 shows an example of a ground truth and generated image before and after post-processing. The colors in the post-processed images are more saturated and the sharpness of edges in the image are somewhat increased but no halo artifacts are visible.

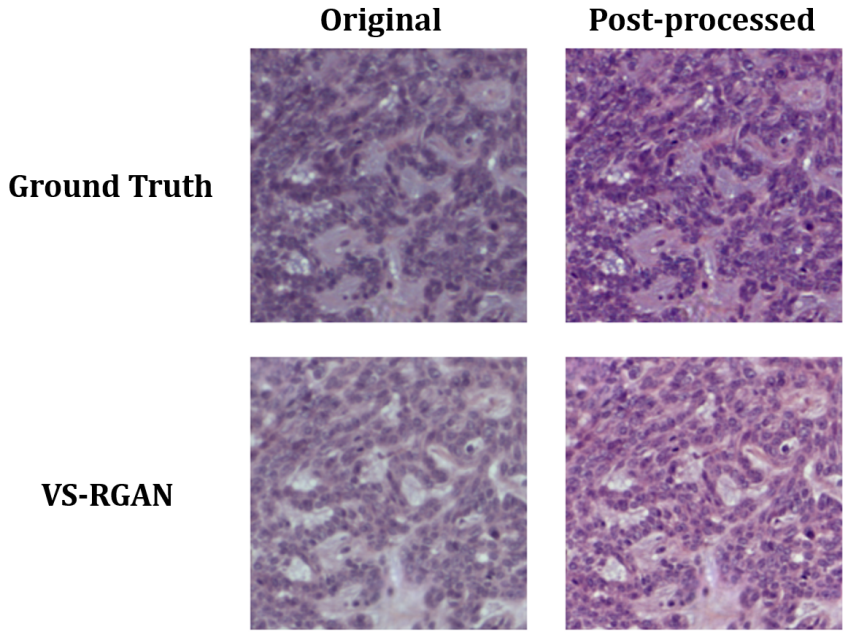


Figure 23: Example of the difference in image appearance before and after post-processing.

For the interested reader, a full PLS stack can be found in Appendix A.3

## 4.2 Training of the Generative Network

### 4.2.1 Architecture of the Generative Network

To find the optimal network architecture for the task of virtually staining skin tissue, six different architectures were trained and their performance evaluated. Data used came from the first dataset. Table 6 and 7 hold the mean evaluation scores for each model on the validation and test datasets respectively.

Table 6: Quantitative evaluation scores. Models trained on the first dataset's training data and evaluated on the first dataset's validation data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-64** | 24 | 0.88 (±0.03) | 0.865 (±0.04) | -0.00157 (±0.0004) |
| **UNet-96** | 17 | 0.891 (±0.03) | 0.865 (±0.04) | -0.00154 (±0.0004) |
| **UNet-128** | 20 | 0.894 (±0.03) | 0.865 (±0.04) | -0.00115 (±0.0004) |
| **Dense UNet-64** | 80 | 0.888 (±0.03) | 0.864 (±0.04) | -0.00155 (±0.0004) |
| **Dense UNet-96** | 70 | 0.891 (±0.03) | 0.865 ±0.04) | -0.00159 (±0.0005) |
| **Dense UNet-128** | 65 | 0.893 (±0.03) | 0.863 (±0.04) | -0.00156 (±0.0005) |

Table 7: Quantitative evaluation scores. Models trained on the first dataset's training data and evaluated on the first dataset's external test data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-64** | 24 | 0.862 (±0.04) | 0.852 (±0.07) | -0.00153 (±0.0005) |
| **UNet-96** | 17 | 0.868 (±0.04) | 0.847 (±0.08) | -0.00146 (±0.0005) |
| **UNet-128** | 20 | 0.875 (±0.04) | 0.854 (±0.07) | -0.00121 (±0.0004) |
| **Dense UNet-64** | 80 | 0.863 (±0.04) | 0.847 (±0.07) | -0.00151 (±0.0005) |
| **Dense UNet-96** | 70 | 0.866 (±0.04) | 0.847 (±0.08) | -0.00153 (±0.0005) |
| **Dense UNet-128** | 65 | 0.869 (±0.04) | 0.849 (±0.07) | -0.00150 (±0.0005) |

The architecture study reveals that the models perform worse on the test data than on the validation data. It is apparent that the **UNet** architecture reaches its best performance after about 20 training epochs, while the **Dense UNet** does so after about 70 training epochs. All models produce images which are less sharp than their ground truth ones, as seen from the Laplace scores all being negative. The results also show that a larger number of input channels in the first convolutional block yields higher CW-SSIM and SSIM scores. In addition, it was observed during training that the more features in the input layer, the longer time a training epoch takes. Training the **Dense UNet** models took about twice as long per epoch as training of their UNet counterpart.

Based on the evaluation scores provided in Tables 6 and 7 and training times, it was decided to move forward with the **UNet-96** model.

### 4.2.2 Evaluation of Loss Functions

The evaluation scores of the best performing models with the addition of perceptual (**p**) and later a Fourier (**f**) loss function are found in Table 8 and 9.

Table 8: Quantitative evaluation scores. Models trained on the first dataset's training data and evaluated on the first dataset's validation data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-96** | 17 | 0.891 (±0.03) | 0.865 (±0.04) | -0.00154 (±0.0005) |
| **UNet-96-MSEp** | 20 | 0.899 (±0.04) | 0.866 (±0.04) | -0.00100 (±0.0004) |
| **UNet-96-MSEpf** | 25 | 0.904 (±0.03) | 0.868 (±0.04) | -0.00065 (±0.0003) |

Table 9: Quantitative evaluation scores. Models trained on the first dataset's training data and evaluated on the first dataset's external test data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-96** | 17 | 0.868 (±0.04) | 0.847 (±0.08) | -0.00146 (±0.0005) |
| **UNet-96-MSEp** | 20 | 0.880 (±0.04) | 0.850 (±0.08) | -0.00107 (±0.0004) |
| **UNet-96-MSEpf** | 25 | 0.882 (±0.04) | 0.850 (±0.08) | -0.00075 (±0.0003) |

The addition of both the perceptual and the fourier loss improved the metric scores. **UNet-96-MSEpf**, trained with an MSE, a perceptual and a Fourier loss, performed the best overall and was chosen as the model to move forward with.

### 4.2.3 Upsampling Blocks and Attention Gates

The **UNet-96-MSEpf** was fine-tuned by testing a variety of upsampling blocks. Little improvement according to the quantitative metrics was found by this change, and furthermore, the PixleShuffle Blur block was not found to reduce checkerboard patterns in empty areas. The choice was made to continue with PixelShuffle upsampling blocks. Attention gates were found to aid performance, making the best model trained on the first dataset the **UNet-96-MSEpf-Pixelshuffle-Attention**.

The evaluation scores of **UNet-96-MSEpf-Pixelshuffle-Attention** trained on the first and second dataset respectively are shown in Table 10 and 11.

Table 10: Quantitative evaluation scores. **UNet-96-MSEpf-Pixelshuffle-Attention** trained on either the first or second dataset's training data and evaluated on the first or second dataset's validation data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-96-MSEpf-Pixelshuffle-Attention First Dataset** | 35 | 0.907 (±0.03) | 0.873 (±0.06) | -0.00054 (±0.0003) |
| **UNet-96-MSEpf-Pixelshuffle-Attention Second Dataset** | 450 | 0.851 (±0.04) | 0.780 (±0.07) | -0.00111 (±0.0004) |

Table 11: Quantitative evaluation scores. **UNet-96-MSEpf-Pixelshuffle-Attention** trained on either the first or second dataset's training data and evaluated on the first or second dataset's external test data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace |
|---|---|---|---|---|
| **UNet-96-MSEpf-Pixelshuffle-Attention First Dataset** | 35 | 0.888 (±0.04) | 0.869 (±0.05) | -0.00051 (±0.0005) |
| **UNet-96-MSEpf-Pixelshuffle-Attention Second Dataset** | 450 | 0.831 (±0.05) | 0.799 (±0.05) | -0.00095 (±0.0007) |

**UNet-96-MSEpf-Pixelshuffle-Attention**'s ability to generate images of different tissue types found in the first dataset is shown in Figure 24. One problem with these generated images, especially seen in 24(a) and (c) is that the contrast between the nuclei and the tissue surrounding it in the darker areas of the epithelium and along the sweat gland, to the lower right in 24(c), is poor. In 24(b) there is also some cell nuclei, the darker spots seen in the ground truth image, missing in the generated image. Finally, the focus in 24(d) is also significantly poorer than in the ground truth.

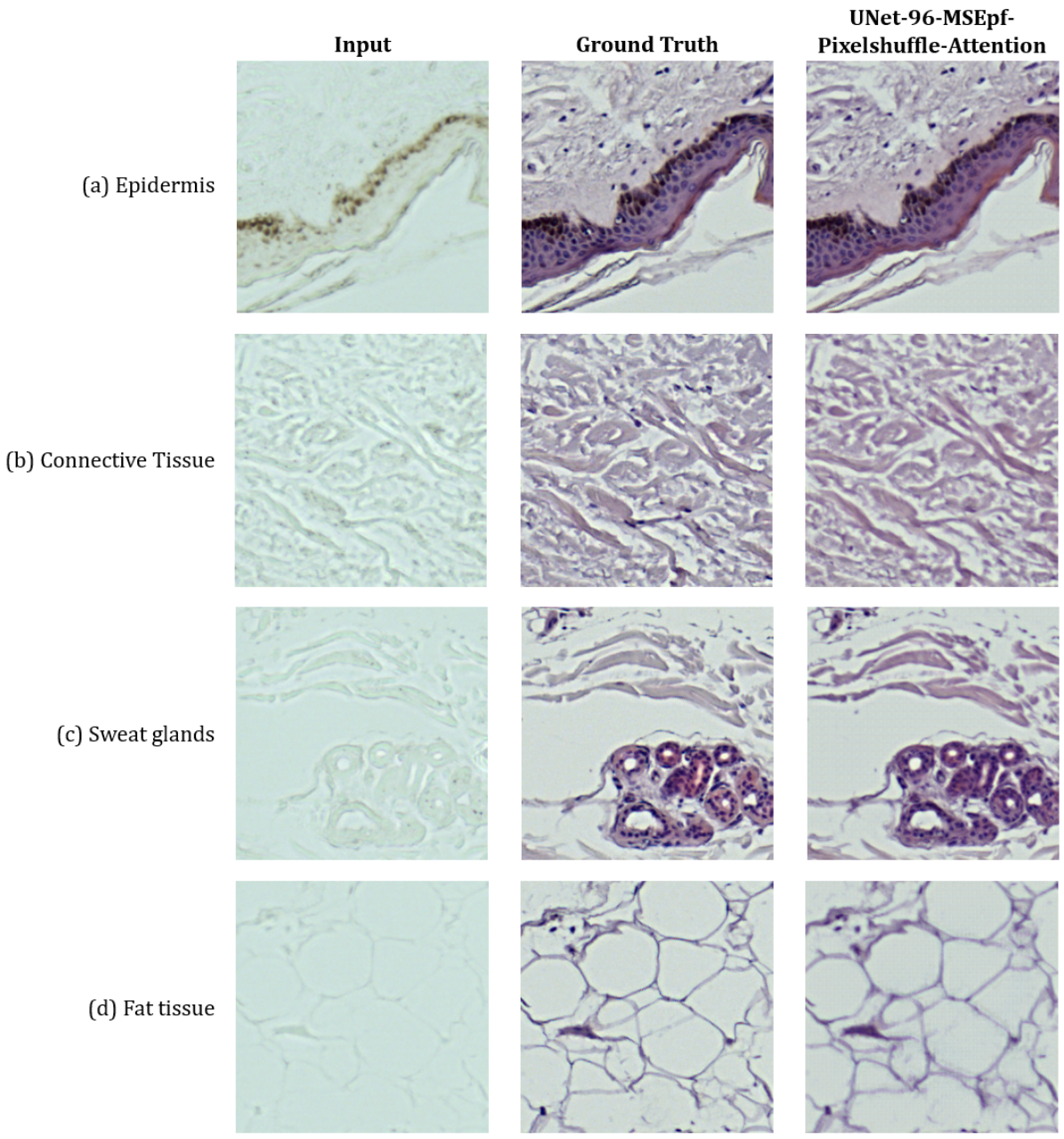


Figure 24: **UNet-96-MSEpf-Pixelshuffle-Attention** trained on the first dataset's ability to generating different tissue types. (a) depict epidermis, (b) connective tissue, (c) sweat glands and (d) fat tissue. Images from the first dataset's external test data.

A problem seen occasionally in the first dataset but more prominently with the introduction of the second dataset was the appearance of background artifacts. The most

prominent such was in the form of a purple cast artifact on empty areas of the images. An example of this can be seen in Figure 25.


Figure 25: Purple background artifacts generated by **UNet-96-MSEpf-Pixelshuffle-Attention** trained on the first dataset. The same phenomena was seen for images generated by the same model trained on the second dataset as well. Image from the first dataset's validation data.

The purple cast artifact was resolved by including images of empty areas of the slides in the training data of the second dataset. As can be seen in the empty areas of Figure 26 and 29(f), no purple cast is visible.

Another artifact observed throughout the project was a patchwork pattern, most prominent in empty areas of the images, as shown in Figure 26. A resolution to this artifact was attempted with changing upsampling methods, especially PixelShuffle Blur upsampling was believed to have the ability to resolve this, but no change was observed. The inclusion of a discriminator did not alleviate the problem either.

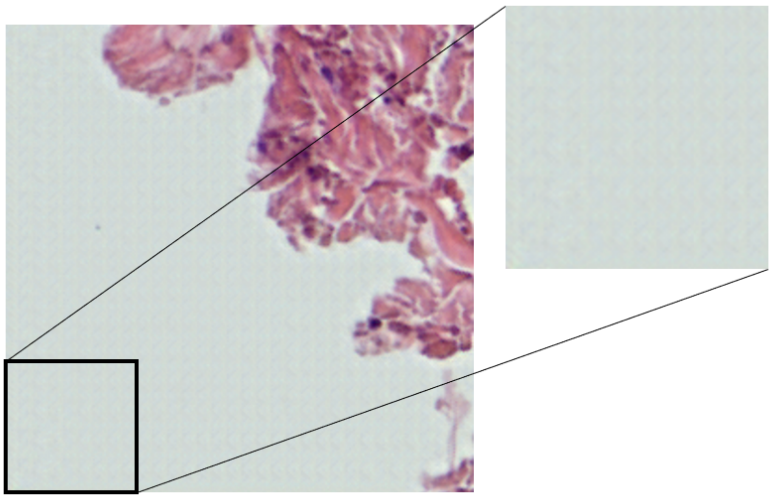

Figure 26: Patchwork artifact in empty areas of an image generated by **VS-RGAN** trained on the second dataset. Images from the second dataset's external test data.

### 4.2.4 Bright-Field versus PLS

By training the **UNet-96-MSEpf-Pixelshuffle-Attention** model exclusively on Bright-Field (BF) images and comparing the evaluation scores and quality of the generated images to those produced by the model trained on PLS stacks, the additional information contained in the PLS stacks was confirmed. Tables 12 and 13 highlight these differences, and it should be noted that there is a statistically significant improvement when using PLS image stacks as inputs.

Table 12: Quantitative evaluation scores. **BF** trained on BF images only from the second dataset's training data and evaluated on the second dataset's validation data. **PLS** trained with the normal PLS stack from the second dataset's training data and evaluated on the second dataset's validation data. For both **BF** and **PLS**, the model architecture used was **UNet-96-MSEpf-Pixelshuffle-Attention**.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace | FID |
|---|---|---|---|---|---|
| **BF** | 110 | 0.721 (±0.06) | 0.692 (±0.03) | -0.00157 (±0.0002) | 1.343 |
| **PLS** | 450 | 0.851 (±0.04) | 0.780 (±0.05) | -0.00111 (±0.0004) | 0.650 |

Table 13: Quantitative evaluation scores. **BF** trained on BF images only from the second dataset's training data and evaluated on the second dataset's external test data. **PLS** trained with the normal PLS stack from the second dataset's training data and evaluated on the second dataset's external test data. For both **BF** and **PLS**, the model architecture used was **UNet-96-MSEpf-Pixelshuffle-Attention**.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace | FID |
|---|---|---|---|---|---|
| **BF** | 110 | 0.704 (±0.06) | 0.631 (±0.05) | -0.00198 (±0.003) | 1.626 |
| **PLS** | 450 | 0.851 (±0.04) | 0.780 (±0.05) | -0.00095 (±0.0007) | 0.858 |

Figure 27 shows a comparison between images generated by **UNet-96-MSEpf-Pixelshuffle-Attention** trained with PLS stacks versus those generated using only BF images as inputs. ΔCW-SSIM denotes the difference in CW-SSIM score between the PLS and the BF image. A small |ΔCW-SSIM| difference between the two sets of generated images indicate high similarity between the two sets, and the smallest difference is found in images of connective tissue. The difference increases when the models are tasked with generating more complex structures, such as connective tissue with nuclei, which the BF model cannot capture, resulting in a larger CW-SSIM difference. The two sets of images on the bottom half of the figure are from samples containing squamous cell carcinomas, and these structures are not accurately replicated by the BF model.

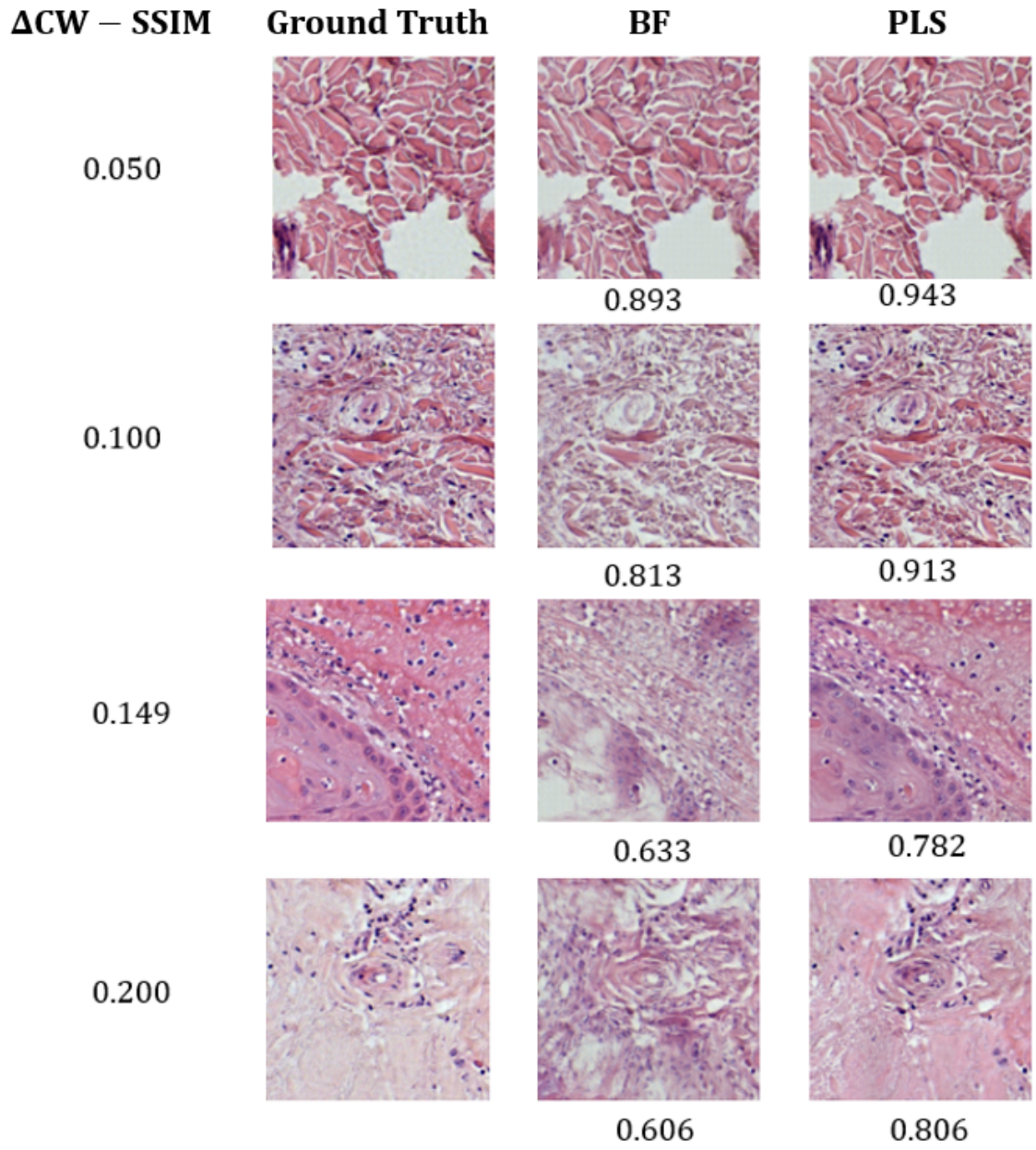


Figure 27: Comparison of test images generated by **UNet-96-MSEpf-Pixelshuffle-Attention** when trained on PLS image stacks versus solely on BF images. The expression ΔCW-SSIM represents the difference in CW-SSIM score between the PLS and BF images. The range of ΔCW-SSIM shown is intended to reflect the difference within one standard deviation of the CW-SSIM scores. The two top images depict connective tissue, the bottom two middle and the bottom depict different areas of a squamous carcinoma. The images chosen were the ones closest to the ΔCW-SSIM 0.050, 0.100, 0.150 and 0.200 respectively. Images from the second dataset's external test data.

Figure 28 illustrates an additional flaw of the **UNet-96-MSEpf-Pixelshuffle-Attention** trained on BF images only, it does not have the capacity to generate large images to the same extent as the PLS version of the model. The BF model fails to capture nuclei, glands and the epithelium in the bottom left corner.

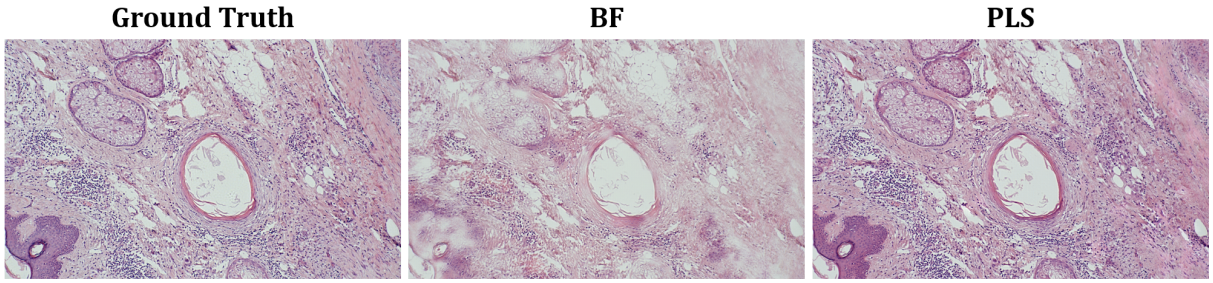


Figure 28: Comparison of full FOV test images generated by **UNet-96-MSEpf-Pixelshuffle-Attention** when trained on PLS image stacks versus solely on BF images. Image from the second dataset's external test data. Larger versions of these images can be found in Appendix A.4.

### 4.2.5 Addition of a Discriminator

The scores from evaluating the three RGAN models along with the **UNet-96-MSEpf-Pixelshuffle-Attention** network's performance on the second dataset is shown in Table 14 and Table 15 respectively.

From the results, primarily by studying the FID score, it can be deduced that the addition of a discriminator improves the quality of the generated images. The **UNet** discriminator architecture generates the best focused and highest quality images, and was therefore further fine tuned by changing the learning rates used during training. This yielded the projects best model, which was a **RGAN** with a **UNet-96-MSEpf-Pixelshuffle-Attention** generator and a **UNet** discriminator. This model is coined **VS-RGAN** to highlight its effectiveness in virtual staining tasks.

Table 14: Quantitative evaluation scores. Models trained on the second dataset's training data and evaluated on the second dataset's validation data.

| **Model** | **Best Epoch** | **CW-SSIM** | **SSIM** | **Laplace** | **FID** |
|---|---|---|---|---|---|
| **UNet-96-MSEpf-Pixelshuffle-Attention** | 450 | 0.851 (±0.04) | 0.780 (±0.07) | -0.00111 (±0.0004) | 0.650 |
| **Conv-Disc-RGAN** | 58 | 0.847 (±0.03) | 0.780 (±0.07) | -0.00034 (±0.0004) | 0.310 |
| **PatchGAN-Disc-RGAN** | 12 | 0.855 (±0.03) | 0.792 (±0.07) | -0.00069 (±0.0004) | 0.499 |
| **UNet-Disc-RGAN** | 6 | 0.856 (±0.03) | 0.793 (±0.07) | -0.00042 (±0.0004) | 0.423 |
| **VS-RGAN** | 54 | 0.860 (±0.04) | 0.782 (±0.07) | -0.00091 (±0.0004) | 0.403 |

Table 15: Quantitative evaluation scores. Models trained on the second dataset's training data and evaluated on the second dataset's external test data.

| Model | Best Epoch | CW-SSIM | SSIM | Laplace | FID |
|---|---|---|---|---|---|
| **UNet-96-MSEpf-Pixelshuffle-Attention** | 450 | 0.831 (±0.04) | 0.799 (±0.05) | -0.00095 (±0.0007) | 0.858 |
| **Conv-Disc-RGAN** | 58 | 0.823 (±0.05) | 0.798 (±0.06) | -0.00041 (±0.0006) | 0.767 |
| **PatchGAN-Disc-RGAN** | 12 | 0.834 (±0.03) | 0.812 (±0.07) | -0.00073 (±0.0005) | 0.703 |
| **UNet-Disc-RGAN** | 6 | 0.834 (±0.03) | 0.807 (±0.07) | -0.00040 (±0.0004) | 0.656 |
| **VS-RGAN** | 54 | 0.836 (±0.05) | 0.799 (±0.05) | -0.00076 (±0.0004) | 0.683 |

### 4.2.6 Image Comparison

Throughout the project, the generated images were inspected visually to control that the metrics seemed to correlate with human perception of the images. In Figure 29, a selection of images showing different structural elements in skin tissue are shown for comparison of the final models of this project. Six different tissue types are depicted. The model only trained on BF images struggles to accurately generate virtually stained images. It does not capture as many nuclei and fails in capturing fat tissue. It also produces less sharp images. From visual inspection it is challenging to tell the **UNet-96-MSEpf-Pixelshuffle-Attention PLS** and the **VS-RGAN** apart. Close inspection may show the **VS-RGAN** improving focus and nuclei definition somewhat, but the difference is very small.

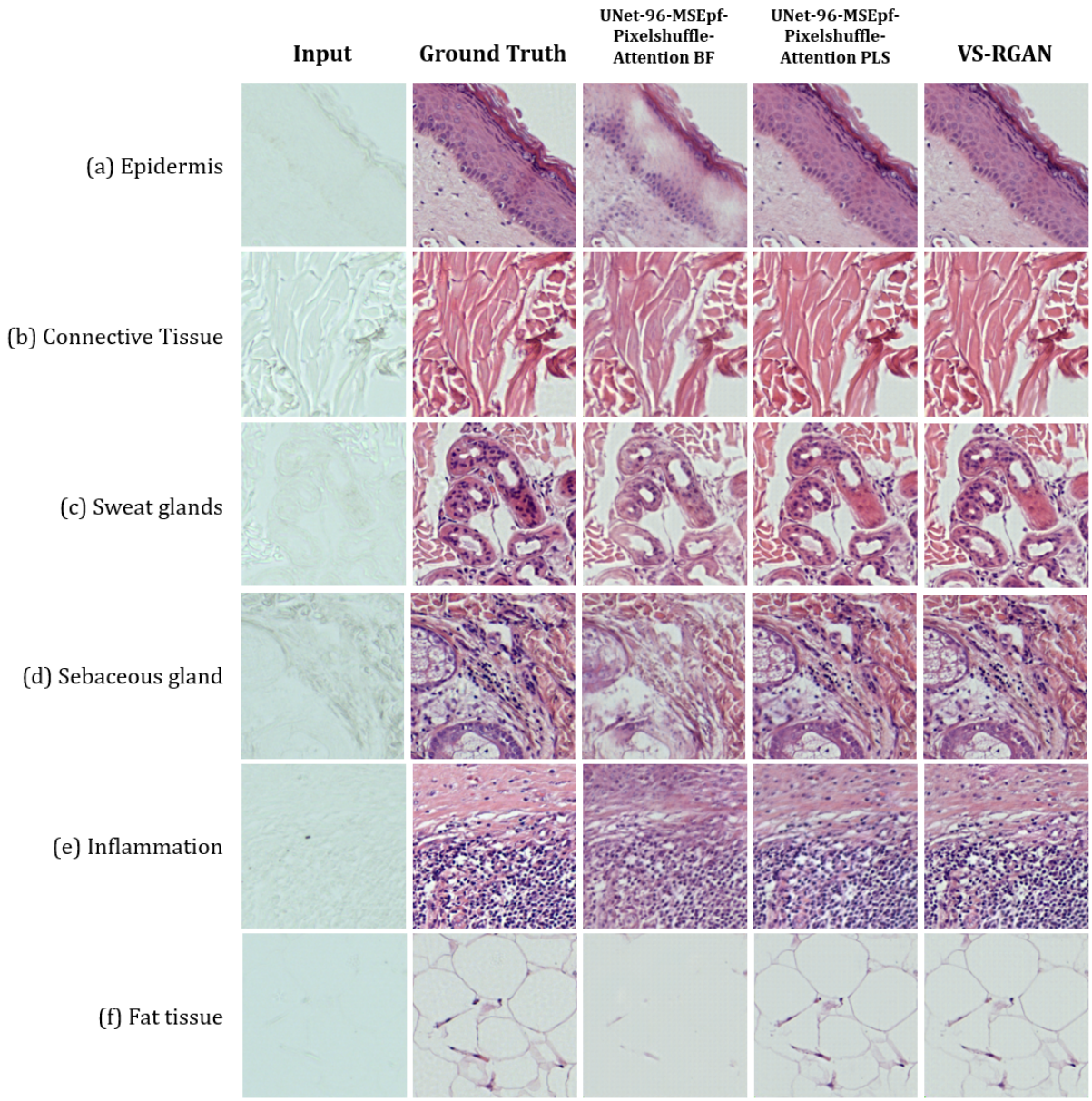


Figure 29: Images of different tissue types generated by **UNet-96-MSEpf-Pixelshuffle-Attention BF**, **UNet-96-MSEpf-Pixelshuffle-Attention PLS** and the **VS-RGAN** from the same input image. The BF model is fed only a BF image, and the two other models were fed a 72 channel PLS stack. (a) depict epidermis, (b) connective tissue, (c) sweat glands, (d) sebaceous glands, (e) inflammation and (f) fat tissue. Images from the second dataset's external test data.

By instead studying the model's abilities to replicate cancerous tissue, differences are more obvious. Figure 30 shows how the BF model struggles with both carcinoma types, especially to generate squamous cell carcinomas. **VS-RGAN** is better than **UNet-96-MSEpf-Pixelshuffle-Attention** in capturing nuclei inside of the tumour. Regardless of model, generated images of basal cell carcinoma lack some contrast found in the ground truth image, and the squamous cell images have a more purple tone than the ground truth image.

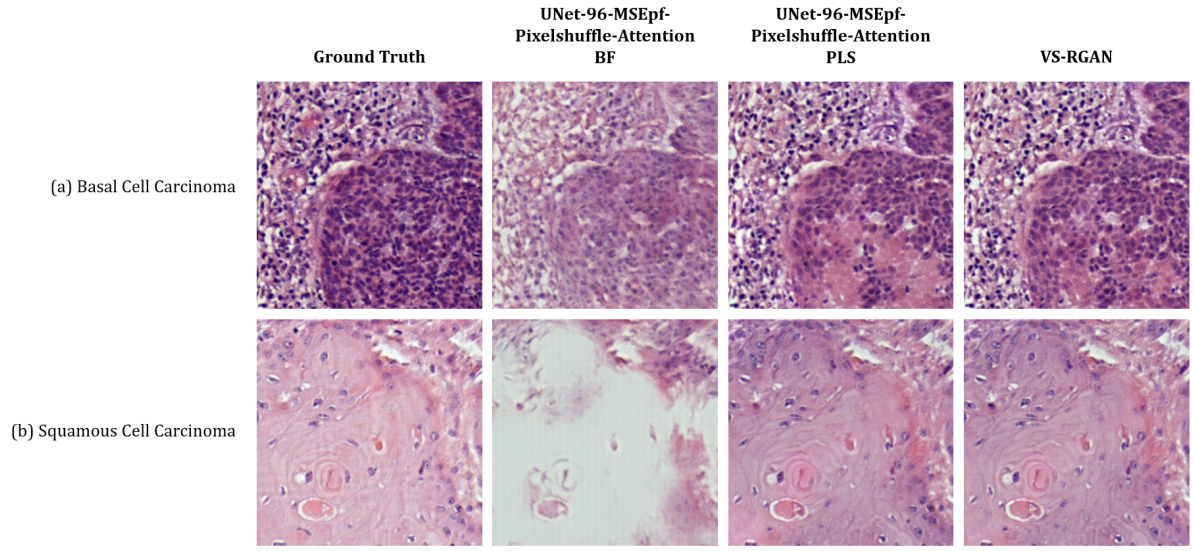


Figure 30: Images of different carcinoma types generated by **UNet-96-MSEpf-Pixelshuffle-Attention BF**, **UNet-96-MSEpf-Pixelshuffle-Attention PLS** and the **VS-RGAN** from the same input image. The BF model is fed only a BF image, and the two other models were fed a 72 channel PLS stack. (a) depict basal cell carcinoma, (b) squamous cell carcinoma. Images from the second dataset's external test data.

In Figure 31, an example of a full FOV generated with the **VS-RGAN** network on test data from the second dataset is seen. The generated image lacks somewhat in contrast within the connective tissue but in general preserves all salient details. With closer inspection it can be seen that the folds along the epithelium are less sharp in the generated image.

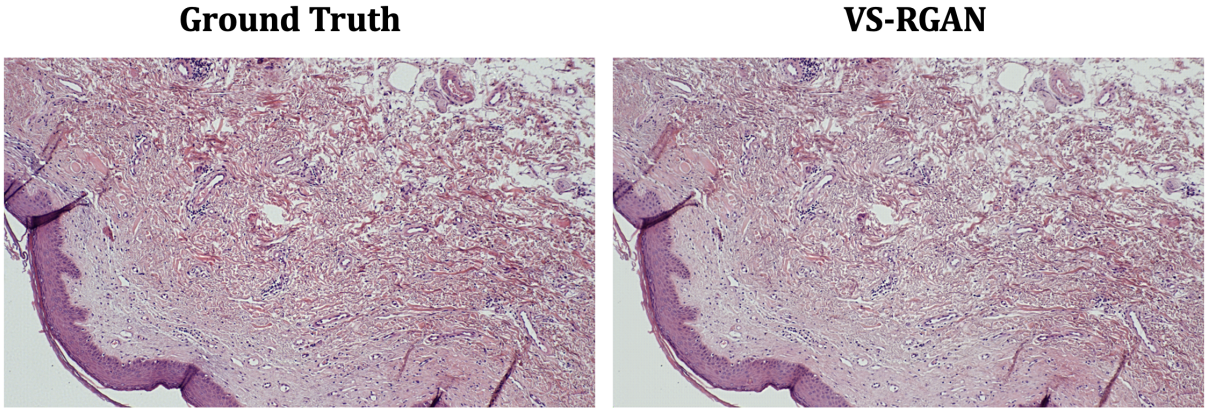


Figure 31: Full FOV image of healthy tissue, ground truth (left) and generated by the **VS-RGAN** (right). Image from the second dataset's external test data. Larger versions of these images can be found in Appendix A.5.

## 4.3 Results from Qualitative Questionnaire

The qualitative questionnaire was answered by seven participants with different professional backgrounds, including medical doctor, consultant, pathologist and personnel from

Table 16: Confusion matrix for the cancer identification in virtually stained images.

| | | True diagnosis | | |
|---|---|---|---|---|
| | | Positive | Negative | Total |
| Expert response | Positive | 30 | 0 | 30 |
| | Negative | 3 | 39 | 42 |
| | Total | 33 | 39 | 72 |

Table 17: Confusion matrix for the cancer identification in chemically stained images.

| | | True diagnosis | | |
|---|---|---|---|---|
| | | Positive | Negative | Total |
| Expert response | Positive | 30 | 0 | 30 |
| | Negative | 3 | 39 | 42 |
| | Total | 33 | 39 | 72 |

Lund University Bio Imaging Centre (LUBIC). Four of the participants answered version 1 of the questionnaire and three answered version 2. The participant from LUBIC declined to answer the questions about pathological states as they had no experience in the field. This meant that for this question we received three answers for each version of the questionnaire.

Table 16 and 17 represent confusion matrices for classification between cancerous and non-cancerous tissue. The matrices are identical for the virtually stained images and the chemically stained images, which is also presented in Table 18. For both the real and generated images, the three incorrectly classified images were of squamous cell carcinoma. From these confusion matrices the observant reader may notice that there were 11 images with cancerous tissue and 13 without. This was a flaw in our questionnaire design. In private discussion with Fredrik Pontén it was discovered that one image included was incorrectly annotated by us as a basal cell carcinoma, in fact was entirely healthy. What we had believed was a carcinoma was in fact a hair follicle. We therefor retroactively changed this annotation to healthy tissue, and the results for the corrected questionnaire is what is shown throughout the results of this section.

Table 18 presents the results from the qualitative evaluation questionnaire. The quality of stain is rated 6.40 and 6.41 out of 10 for the virtually and chemically stained images respectively. The recognized usefulness of the images were 63.2% for virtually stained images and 62.8% for chemically stained images.

Table 18: Qualitative evaluation questionnaire results. The classification accuracy includes true negative and true positive diagnoses.

| | **Stain quality (out of 10)** | **Usefulness (%)** | **Classification accuracy (%)** |
|---|---|---|---|
| **Virtually Stained** | 6.40 ($\pm$1.04) | 63.2($\pm$21.5) | 95.83 ($\pm$11.7) |
| **Chemically Stained** | 6.41 ($\pm$1.18) | 62.8 ($\pm$28.1) | 95.83 ($\pm$16.2) |

From comments provided by the participants it was made clear that the magnification of the images was seen as a negative for the overall quality and usefulness of the images.

Furthermore, some images were criticized for tissue being folded or containing damaged tissue.

### 4.3.1 Discussion with Pathologist Fredrik Pontén

Upon completion of the evaluation questionnaire, a discussion was had with Fredrik Pontén about his thoughts on the results of the project and the potential benefits of virtual staining in the future. The main points from this discussion were as follows.

Pontén deemed all large scale images we showed to be useful to a pathologist in that all relevant structures were distinguishable and tumor tissue could be identifiable as such in the images containing it. Some shifts in staining compared to the ground truth were shown to him, similar to what is shown in Figure 30(b). In his opinion, this difference looked like a slight stain variation but it did not make the diagnosis unclear or the image obviously appear generated.

Pontén emphasized the benefit of standardization of staining through virtual stain for image analysis methods. Pathologists are trained to adapt to variations in staining, but international studies using images from different labs struggle to apply image analysis methods due to poor standardization across labs and countries. When asked, he said that cost and efficiency would likely be improved with virtual staining but that he had no knowledge of the specifics of the size of those reductions. Further exploration of this would be needed.

It was also pointed out by Pontén during this discussion that this virtual staining method would likely not be applicable to some other histological stains which rely more heavily on chemical reactions within tissue to generate color contrast. Pontén does not think it is possible from illumination of unstained tissue alone to differentiate for example fluids with varying pH levels.

# 5 Discussion

## 5.1 General Reflections

We find the achieved results quite impressive. One strength, which we did not initially realize the importance of, was the ability of our models to generate significantly larger images, such as the one shown in Figure 31. These are much larger than those the **VS-RGAN** was trained on, and yet no visible artifacts or detriment in metric scores were observed. Pathology professionals typically work with full FOV images, allowing them to observe various structures and the context surrounding potential abnormalities. The ability to train the networks on smaller images while still being able to generate complete FOV images without complications enabled us to retain these advantages while enhancing the efficiency of our neural network training process. This is because training on smaller images yields less trainable parameters which decreases the time-consumption of each training epoch.

From the first functional UNet trained in this project the generated images yielded impressive metric scores, although by visual inspection, some issues were revealed. Because of this, the main part of this project focused on fine-tuning the model by various means of structural changes to the network or changes to training parameters. By visual inspection we believe the quality of the generated images have increased, however only a few statistically significant improvements in quantitative evaluation scores were noted. Despite the challenges in achieving statistically significant improvements between different models trained using PLS, a key result of this project was that both **UNet-96-MSEpf-Pixelshuffle-Attention** and **VS-RGAN** trained using PLS images outperformed the model trained on BF images in all evaluation scores, in a statistically significant way.

Based on the results we are satisfied that our method of choosing the best epoch worked well as there was no significant difference in metrics between the validation and test data. This is a strong indication of our model being well generalized and not overfitted. During the project we did not observe any visual indications of overfitting or decrease in overall quality of the images when switching to the external test set. Because of the improved choices made in partitioning the data, these results are especially convincing for the second dataset. This will be covered in the next section.

## 5.2 Data Collection and Data Processing

Images with damaged tissue or dirt were manually removed, as were images not possible to align with template matching. Although checks were performed to eliminate these types of problems, some inevitably remained in the data used, which may have lowered the metrics scores and increased the standard deviation. Especially, poorly template matched images having slipped through the checks proved detrimental to metric scores such as SSIM, due to it being sensitive to translations, but even metrics more robust to such changes, as CW-SSIM, were negatively impacted. When working with large datasets, it is very hard to ensure that all examples are of high quality, and if there had been more time we would have liked to better ensure the quality of the data, which is a science in itself. An especially difficult task was working with images where relatively small changes were observed such as that shown in Figure 22. We chose to let these types of images

remain during data processing to preserve data but found that even small changes of this kind were detrimental to evaluation scores, especially SSIM but also the more robust CW-SSIM.

A limitation of our experiments that was not realized until we received answers from the questionnaire was that the quality of the tissue sections was varying. Areas of broken and folded tissue was pointed out to us, and to some extent we believe this affected the model's performance. In Figure 29 (c) and 31 (along the epithelium) these types of areas are shown. It can be seen that these areas lack some cell nuclei and sharpness.

The first dataset exhibited reduced contrast with respect to crucial structural elements, and both datasets encountered challenges with uniform staining. Notably, the generated stain appeared to represent an even average of the stain variations seen in each of the datasets. This observation underscores the difficulty in achieving consistent staining, even within a singular laboratory setting, and speaks to the benefits of virtual staining in ensuring color consistency. The second dataset provided essential improvements to the relevance and credibility of the networks trained. Beside the pale stain, the first dataset was significantly lacking in both variation of patients and in unique areas of tissue. The first dataset also entirely lacked certain structures commonly found in skin tissue (see Figure 24 and Figure 29, showing the identified tissue types in each dataset) which further limited how general a model trained on this dataset could become.

The biopsies in the first dataset were significantly smaller than those in the second dataset, and although a larger number of slides were available from each biopsy, the slides were all from the same cross section of consecutive cuts of tissue. This meant the content of each slide was very similar to all other slides from the same patient. One benefit of this was that there was a natural introduction of rotational augmentation as most slides were fixated to the glass slide in different orientation. This could have increased the amount of information the models were able to extract from each biopsy.

In the second dataset, each biopsy was significantly larger and each biopsy was also more significantly different from the others. This meant that not only were structures unseen in the first dataset introduced with this dataset, but more often than not, several examples of each structure was present within one biopsy and between biopsies. The addition of cancerous tissue also opened up for the possibility to investigate the model's ability to generate images useful for pathologists in making important, even life-saving diagnoses. With this, it also became highly relevant that the models did not generate images that would lead to false positive or negative diagnosis. The qualitative questionnaire, which will be discussed in more detail later, was introduced to assess this.

The method of partitioning was another aspect which was improved upon between the first and second data collections. In the first dataset, images from all patients was present in all partitions, which with the limited variation described above, likely skewed results to be more positive than they would have been for a truly external validation and test set. The choice to divide the data in this way was made because it was believed to be too limiting to train the network on only one patient and loose almost 60% of the data to validation and testing. In hindsight, a better alternative would probably have been to at least use the tissue from one patient as an external test set.

In the second dataset the increased number of patients available meant that the test set could be made external from the training and validation data. Partitioning the patient biopsies included in both training and validation by tissue meant that these partitions were differentiated, although not by patient. A limitation with this partitioning was that because the cancer patient biopsies were significantly larger than those from healthy patients 32% of the second dataset became test data, while only 68% remained for training and validation. We deemed this loss of training data necessary for the validity of the results but it was a frustrating decision which further highlights the difficulty in acquiring enough diverse data. Partitioning by tissue would have been a good choice for the first dataset as well but would have been difficult due to the rotation and limited area of each slide in that dataset.

The first dataset, although it had significant limitations, did provide a stable basis for the continued experiments with the second dataset. Visual examination of the images produced when training the **UNet-96-MSEpf-Pixelshuffle-Attention** model using the second dataset showed high fidelity to the ground truth images and few noticeable artifacts.

The drop in metric scores, when switching between the datasets, especially in SSIM and CW-SSIM scores was, at first, concerning. It is however relevant to consider that all metric scores are calculated based on the ground truth images. A dataset with significantly lower color contrasts, variation of tissue and separation between the training, validation and test data scoring higher quantitative metrics is not surprising. The validity of these metric scores, and thereby model generalizability, is only as good as the ground truth, and in the case of the first dataset the ground truth was lacking.

It was decided that both the ground truth and generated images shown in the report should be post-processed to enhance color saturation and sharpen the appearance of edges. This choice was made because the raw ground truth images were somewhat lacking in these aspects. We did not calculate the metric scores based on the post-processed images because we wanted the scores to solely focus on the performance of the neural networks. In Figure 23 an example of both a ground truth and generated image, before and after post-processing, is shown. In our opinion, and support from a brief discussion of the post-processing with Fredrik Pontén we concluded that the changes make the images appear more pleasant but does not alter the ability to diagnose or identify different structures in the tissue.

### 5.2.1 Augmentations

The introduction of a rotational augmentation was found to be a detriment to the quality and metric scores of the images. One plausible explanation for this lies in the nature of PLS microscopy. Each image taken with individual LED:s contain information about both angle and direction of the light which may be learned by the network. Random alterations in direction may therefore result in information loss, potentially impacting the effectiveness of this augmentation technique.

Random crop augmentation on the other hand did yield positive results. This success can be attributed to the variation offered by different crop areas, contributing valuable contextual information to the network, which may have been especially significant as

the training data was comprised of small images. Visual inspection indicated that this fine-tuning aided the network in capturing smoother transitions between tissue types, as observed in the full-sized images.

An alteration of the data, although not an augmentation in the pure sense, was the inclusion of some images containing only empty areas in the second dataset. This was, as mentioned in section 3.2, done to aid the network in generating areas with a lot of background which before had a purple cast, seen in Figure 25. We hypothesized that the model struggled due to not having been exposed to these areas, which were removed by the Laplace variance threshold (see Section 3.2), in earlier training of the network. As the artifacts disappeared with this inclusion this hypothesis seemed to be accurate. The network was then able to generate thin tissue, background, and fat tissue, an example of which can be seen in Figure 29 (f).

## 5.3 Training of the Generative Network

Improvements, albeit modest, in average CW-SSIM and Laplace focus scores were seen when increasing the number of input channels of the **UNet**, while SSIM remained constant. We opted to continue with 96 input channels, as training a network with 128 input channels was significantly more time consuming and computationally costly. The choice of 96 channels represented a compromise, serving as a middle ground between computational efficiency and performance. Notably, 96 channels meant that the PLS stack, consisting of 72 channels, was not immediately compressed by the network, which we hypothesized would be helpful to the model's performance.

The **Dense UNets** tested did not show any discernible improvement except for the version with 128 input channels. These models did not only need a higher number of epochs to reach their peak performance, but due to the complexity of the network architecture, each epoch took significantly longer than the normal **UNets** as well. This was what ultimately made us decide to continue with the simpler **UNet** architecture. By visual inspection and metric scores, the **UNets'** capacity seemed on par with the virtual staining task which made the less efficient, more complex **Dense UNet** an unnecessary evolution. We do not want to rule out that for a more complex virtual staining task, for example working with a multitude of different tissues, a **Dense UNet** would prove superior.

### 5.3.1 Loss Functions

The choice of the MSE loss function for the initial experiments was motivated by its simplicity and effectiveness in quantifying differences between two images at the pixel level. While MSE is effective in reducing the overall error, it tends to smooth out image details. This was found to result in a loss of sharpness and textural features in the generated images. Recognizing these problems as inherent problems with the MSE loss function lead to the continued search for a combination of loss functions to better preserve sharpness and high-frequency details in the images.

To address these shortcomings, we incorporated additional loss functions, each chosen for their specific abilities and we tried to combine them in a way to complement and counterbalance the weaknesses of the others.

A Fourier loss was added to match the frequency content of generated images with that present in the real ones. Its inclusion aimed to enhance image sharpness and focus, particularly emphasizing the appearance of high-frequency components like cell nuclei. The loss proved effective, and was therefore kept.

The lack of cell nuclei and poor color contrast in the images also motivated the inclusion of a perceptual loss function. Perceptual loss, known for its effectiveness in enhancing perceptual similarity in image translation tasks, was anticipated to address these issues. While a known pitfall of perceptual loss is the potential for generating hallucinations or artifacts, our models did not exhibit these behaviours, and the average evaluation scores improved with this addition. Worth noting is that our perceptual loss was based on the VGG19 network, trained on the ImageNet dataset. The ImageNet dataset is not focused on medical images, but rather everyday objects such as animals and landscapes. We did not explore it in this project, but do believe that a similar network trained on pathology-specific images, particularly H&E stained images, would have improved the perceptual loss function when used for virtual staining tasks.

As can be seen in the images in Figure 24 generated by the **UNet-96-MSEpf-Pixelshuffle-Attention**, they still lack some contrast, especially between the tissue and the nuclei. To make the model focus on this lack of contrast we experimented with a weighted MSE approach where a mask of the cell nuclei was created and the loss on the nuclei pixels was weighted five times larger than the rest of the pixels. This experiment highlighted that intuitive loss functions, believed to promote a wanted behavior, not always translate effectively to neural networks, as this attempt led to more issues than progress. It underscores that although neural networks often generate results that align with human intuition, they do not possess the ability to interpret underlying intentions with the loss functions chosen. There is not a straight-forward way to tell a network "focus more on preserving cell nuclei", and the network can often find another approach to minimize a loss function than what was intended by the user.

The selection of loss functions was fully in the hands of us as model designers. While our chosen functions (MSE, Fourier, and Perceptual loss) and their weighting seemed to aid the model, it is possible that other, perhaps more effective, loss functions exist for this particular task.

### 5.3.2 Activation Functions, Upsampling and Attention Gates

The proficiency of the model at this stage of the project was already promising, providing a solid foundation for further refinements.

To enhance the robustness of the model, the activation function was changed from ReLU to Leaky ReLU to safeguarding against potential vanishing gradients. However, the lack of noticeable improvement in evaluation scores indicated that dying ReLU was not a significant issue for our network. Given that the training time remained unaffected, we opted to continue using Leaky ReLU. Its presence, though not dramatically transformative, offered protection against possible mode collapse — a valuable feature, particularly in light of our plans to shift to a more complex dataset later.

In our exploration of upsampling methods, Pixelshuffle emerged as the best choice. It did not slow down training, and theoretically, Pixelshuffle is believed to enhance resolution, aligning well with our objectives. Despite not showing significant improvements in evaluation scores, it was retained for potential benefits. The advanced Pixelshuffle Blur variant was discarded due to its impact on training speed and failure to reduce the patchwork artifact in empty areas, which was one of the key reasons for its potential use.

The integration of attention gates was motivated by the need to capture finer details, such as cell nuclei, and to enhance the sharpness of common and peripheral structures such as connective and fat tissue. Attention gates theoretically help the model focus on important features while suppressing less relevant ones, and our results, showing some improvement with this inclusion, aligned with theory. This addition did not significantly alter the training time, thus being considered a valuable addition to the model.

In conclusion, while none of the alterations discussed above led to statistically significant improvements in performance, they contributed to the overall robustness and functionality of the model. In terms of evaluation scores, the generator seemed to have plateaued in terms of performance. In an effort to break through this plateau, we introduced a discriminator as a final enhancement.

### 5.3.3 Addition of a Discriminator

In our exploration of GANs with SGAN loss, we anticipated challenges in keeping the loss stable and enhancing generator performance, a complexity presented in prior research. This instability stems from issues like volatile and vanishing gradients, which we also encountered. In our experiments, the discriminator dominated the training process. It rapidly differentiated between generated and real images, resulting in its loss approaching zero after just a few epochs. This imbalance led to the GAN generator loss dominating the generator's loss function, meaning the input from the remaining loss functions was essentially ignored. This degraded the image quality rather than improving it, which is logical since the GAN generator loss does not focus on preserving image content, as long as generated images appear real.

Due to these challenges, we decided to employ a RGAN loss instead. Given its recognition as a "state-of-the-art" approach in current GAN research, this transition was deemed a logical step. We implemented three distinct discriminator models within the RGAN framework. As outlined in Section 4.2.5, the **UNet** discriminator emerged as the most effective. In comparing the quantitative metrics of our GANs with the **UNet-96-MSEpf-Pixelshuffle-Attention** model, we observed no statistically significant enhancements in CW-SSIM, SSIM, or Laplace scores. However, the improvement in the Frechet Inception Distance (FID) was notable. An improvement in the FID score is expected when integrating a discriminator into training of a generator. This stems from the discriminator's role in analyzing and discerning features to differentiate between real and generated images, and the generator in response leveraging this analysis to enhance its capability in producing more authentic-looking images. The FID, which evaluates the similarity in feature distributions between sets of real and generated images, consequently reflects these advancements in image authenticity.

The process of training a GAN and balancing the capabilities of the generator and discriminator proved to be a challenging task. Identifying the respective strengths and weaknesses of each network and determining how to modify them for optimal results was challenging. We experimented with various configurations, altering the size of the discriminator and the learning rates of the networks. A particularly effective adjustment was moderating the learning rate of the discriminator, preventing its loss from quickly becoming close to zero and thereby averting a stop in the GAN's learning process. Additionally, providing the GAN with larger images as inputs appeared to enhance the realism of the generated images. This improvement might be attributed to the GAN's increased capacity to detect subtle discrepancies, such as inadequate color transitions between different tissue structures only visible when studying larger images, which the generator previously struggled to discern.

### 5.3.4 Evaluation of the Generated Images

**Comparison by Visual Inspection**

As mentioned, the variation of the first dataset was very limited, with the sweat gland shown in Figure 24(c) being one of circa five glands present in the entire dataset. Upon comparing the staining of the ground truth images in Figure 24 and 29, it is clear that the color contrasts are smaller, and that the stain overall is darker and more blueish in the first dataset. This may have impacted the models' performance since the contrasts between background structures and cell nuclei for example are vitally important for analysis. Since the ground truth had lower contrast in the first dataset, it is possible that the loss functions and quantitative metrics did not react as strongly to even poorer contrasts in the generated images, as this change could be relatively small.

From Figure 29, a number of observations can be made. The first of which being that the images generated by the network trained on only BF images are lacking in color contrast and have patchy areas, easily noted in the images of epidermis and sebaceous glands. The BF model also incorrectly generates the fat tissue as an empty area. The BF model does however, to some extent, manage to capture the structure of the tissue. It performs quite well on the connective tissue, which makes up a relatively large component all skin tissue on a slide. Although it lacks in focus and nuclei definition, the generated images of sweat glands and inflammation show some promise, although, they are significantly worse than the same image generated by the two models trained with PLS images as input. When attempting to generate full FOV images using the BF model, as seen in Figure 28, its struggles become more apparent. The BF model is not able to generate any cell nuclei not already visible in the unstained image, and it is also unable to accurately generate epithelium. By also taking into account the BF models ability to generate carcinoma structures, as seen in Figure 30, the images generated by this model would likely be useless for a pathologist, since none of the vital structures are appropriately generated.

It is difficult from the images in Figure 29 to make a compelling case for the benefits of including a discriminator at the end of the training process. If one inspects the images depicting epidermis and inflammation closely, some improvement of nuclei definition and focus can be seen. However, it is unlikely that such a small difference would be imperative to a pathologist examining the images. In our discussions with Fredrik Pontén it was made

clear that the presence or non-presence of a few nuclei, as seen when comparing the images in Figure 29, would likely not raise a concern or change a diagnosis made by a pathologist.

In Figure 30, a more compelling case can be made for the **VS-RGAN**. It is clear that the BF model struggles with color contrast in the inflamed connective tissue surrounding the basal cell carcinoma. It also entirely miss-calculates a large section of the squamous cell carcinoma as empty. For the squamous cell carcinoma, both **UNet-96-MSEpf-Pixelshuffle-Attention PLS** and **VS-RGAN** perform similarly. The generated stain is in both instances darker and more purple than the ground truth, however the nuclei definition and overall appearance of the tumor is well captured. For the basal cell carcinoma, the **UNet-96-MSEpf-Pixelshuffle-Attention PLS** seems to misinterpret the middle of the tumor as connective tissue, while the **VS-RGAN** captures almost all of the tumor correctly. This example may be indicative of the benefits of the inclusion of a discriminator network, although further exploration would be needed to confirm that this improvement is due to the discriminator and not a random fluctuation in model performance on a specific area.

**Quantitative Results**
To evaluate the quality of the generated images, we chose a set of quantitative metrics with the aspiration of covering a wide array of image quality aspects. SSIM was chosen due to its widespread use as a metric for assessing the quality of generated images, including virtually stained ones as have been shown in previous studies. Being aware of potential problems with alignment between the unstained-stained image pair, we also included CW-SSIM. This choice was made based on its reduced sensitivity to pixel-wise differences and was a recommendation from the Computational Imaging team at CellaVision. Since the Fourier transform, yielding frequency content of an image, was used as a loss function, we opted for Laplace scores to asses image sharpness and focus. Lastly, FID was chosen to provide perceptual insight not captured by the other, more perceptually low-level metrics.

There is a possibility of evaluation biases, and of the scores not fully reflecting the models abilities. Since the first functional network, we have observed a consistent trend where all networks excel at generating connective tissue and other common structures. This has lead us to believe that the representation of smaller or less common structures, such as nuclei, glands, and tumors, might be underrepresented in our metric scores. If a continuation of this work was to be done, we would recommend the input data to be annotated by tissue type and pathological state. Such annotations would hopefully lead to easier identification of areas the model struggles with, by making this easier to measure quantitatively, which could lead to more significant changes between model alterations. This approach would in addition allow for tailoring of the training data. By identifying areas where the model's performance is weaker, we could enrich the training dataset with more examples of a specific annotation.

The project highlights the limitations of SSIM, and its sensitivity to local differences. In Figure 22 the SSIM score is 0.348 and the CW-SSIM score is 0.563, which indicates that SSIM is more sensitive to small differences than CW-SSIM. We believe that this might be one of the reasons for the larger standard deviation seen in SSIM scores throughout, and makes a good case for the importance of data cleansing. We would have liked to filter out the 20% of the unstained-stained image pairs with lowest SSIM score, and then

gone through them to see if there are structures which have moved or disappeared in the staining step. Because the model only gets to look at the unstained image, it will naturally match the structures location in that, leading to lower evaluation scored when compared to the chemically stained image if structures have shifted.

Studying the generated images in Figure 27, it is apparent that although the metric scores are similar, with a small, although statistically significant improvement, when going from BF to PLS images as input, the perceptual difference is large. None of the metrics, except for perhaps FID, capture the big perceptual differences noted by the human eye in these images, providing some critique to our method. Our chosen metric to guide training, CW-SSIM, capture structural differences very well, but it would have been beneficial to also study color generation capabilities. A convincing color metric was not found during the extent of this project. We would have liked to have digged deeper into the investigation of literature on this subject specifically, because in the articles on virtual staining read, these metrics were often completely absent. Ideally a metric which could identify color differences in a localized manner with some consideration for perceptual shift would have been utilized.

**Qualitative Questionnaire**
For all questions of the questionnaire the results did not differ significantly between the generated and ground truth images. This lends itself to the idea that the participants could not tell the difference between generated and ground truth images. Two participants raised that there were other aspects such as section context and quality which was lacking that impacted their ability to properly evaluate the images. Furthermore two other participants raised the concern that the resolution of the images was too low to discern the internal structure of the cells. Both of these issues may have skewed the results to be lower. To us this highlighted the difficulty in constructing a questionnaire where the intended meaning of each question is understood by the participants. In being careful to not unfairly skew the participants responses in our favor, and also not deter them with a too verbose introduction, some known limitations of the project were left out. Although the extent to which this impacted results can not be known we did observe a trend that damaged tissue sections were rated lower on stain quality, and that overall the participants that requested higher resolution scored the staining quality lower throughout.

The most promising results from the questionnaire was the cancer classification accuracy which was very high. This indicates that the virtual staining was sufficient for accurate diagnosis to the same extent as the traditional staining. Furthermore there were no false positives which alleviates the fear that the generative network may hallucinate carcinomas where none are present in reality. All false negatives, concerning cancerous tissue, from the questionnaire were on images of squamous cell carcinoma potentially indicates a limitation of the model. The identification process of the squamous cell carcinomas may also have been more difficult due to the fact that they often covered an entire FOV image, which made it impossible for the participants to study the edges of the tumor.

## 5.4 Comparison to Previous Virtual Staining Works

The inspiration for utilizing PLS microscopy for virtual staining comes from the work by [15]. We designed our networks based on the insights from [8], who highlight GANs as a prevalent framework in virtual staining due to their robust representation capabilities. This approach aligns with the successful applications seen in [10]. The method of partitioning tissue slides in halves for the training and test datasets, as adopted in [10], inspired our data partitioning strategy. Additionally, we followed a similar methodology for evaluation of the generated images, utilizing both quantitative metrics and pathologist assessments, as done in some of the studies discussed in [8].

Recent virtual staining studies, as mentioned in Section 1, typically use BF or autofluorescent images. The study by [78], using BF images as input for a GAN, achieved an SSIM score of 0.58. When compared to our SSIM scores – 0.799 for **VS-RGAN** and 0.631 for **UNet-96-MSEpf-Pixelshuffle-Attention BF** – our networks seem more proficient. However, [78]'s network is a multi-stain network capable of producing H&E, Picrosirius Red (PSR), and Elastin van Gieson (EVG) stains, and therefore trained with a somewhat different objective than ours. Their model was trained on 33 351 H&E, 35 964 PSR, and 36 228 EVG 512x512 unstained-stained image pairs and tested against 12 048, 12 488, and 10 469 such image pairs. Our model is trained on 132 890 and 367 372 unique 64x64 image pairs and tested on 23 451 and 171 936 unique image pairs contained in the first and second datasets respectively, we note that our dataset is much smaller than theirs, because their images are of larger size than ours. Therefore, our higher evaluation scores might not accurately represent our model's true capabilities, and direct comparison is made difficult by the fact that the similarities in variation and image content between their and our dataset is not clear. As previously discussed, expanding our dataset would help in generating more accurate metric scores.

The generator used in [78] was trained with an adversarial loss combined with an MSE loss, and presents results consistent with our findings. We noted that the use of only an MSE loss smooths image content, generating images lacking in nuclei contrast and sharpness. By visually inspecting [78]'s generated H&E images, this seems to be the case in those images as well. We believe this reduction in detail to likely have contributed to the the three pathologists in their study deeming the quality of virtually stained images lower than that of chemically stained ones. Their pathologist evaluation approach is similar to ours, and our results, which show no discernible difference in stain quality between virtually and chemically stained images, might stem from our integration of other loss functions as well.

The study by [10] compares the proficiency of an unsupervised approach, using a CycleGAN, with a supervised pix2pix model trained using pixel-wise ground truth, in H&E staining of prostate, liver, testis, and kidney tissue. They show that the supervised approach yields better results, supporting our use of supervised learning. No quantitative evaluation is done on the images generated by the supervised method, but visual inspection reveals some of the same issues noted in our images. Some nuclei are lost, and the images lack somewhat in sharpness. The images provided, being in 20x and 40x magnification and of different tissues than those investigated in this project, make a comparison challenging. The fact that they managed to build a model capable of staining different tissues, with a network architecture similar to ours, piques our interest in testing our model

on different tissues. This study utilized 85 000 256x256 images to train the networks, a larger dataset than ours. In machine learning, the size of the dataset plays a significant role, and we believe that with even more data, our results would improve.

We are cautiously optimistic that we have created a well generalized and impressively proficient network for virtual H&E staining of skin tissue. Upon comparison to previous research we note the need for a larger and more diverse dataset. Inspired by these articles, future research should aim to expand dataset scope and explore the model's applicability to a broader range of tissues and staining types.

# 6 Conclusions

The project showed promising results for virtual staining of skin tissue using PLS images as input to a neural network. The highest performing model is coined VS-RGAN, but this network was not statistically superior to the best generator-only network, UNet-96-MSEpf-Pixelshuffle-Attention, or the other RGANs explored according to quantitative metrics. However, visual inspection did show some local improvements. We are therefore unable to certainly determine the optimal neural network trained for this task; rather, we have found several variations of similar caliber. We would therefore suggest to use the model of smallest computational complexity.

The similarity in metric scores between the validation and test data evaluations throughout the project indicates that the models are well-generalized and not overfitted to the training data. In the first dataset, the variation of patient and tissue was limited both within and between data partitions, limiting the value of these results. However, the same trend was seen with the second dataset, which was significantly more diverse and the partitions were more well-differentiated, giving more convincing evidence that the final models of the project were indeed well-generalized.

A **UNet-96-MSEpf-Pixelshuffle-Attention** trained on only traditional BF images struggled with color definition, structure identification, and focus. When trained with PLS images, this model performed significantly better based on all assessed quantitative metrics, and by visual inspection, the improvement was evident. **VS-RGAN** scored 0.836, 0.799, -0.00076, and 0.683 on CW-SSIM, SSIM, Laplace, and FID, respectively, while the best BF model scored 0.704, 0.631, -0.00198, and 1.626. It is therefore concluded that utilizing PLS stacks as input to neural networks, be that a generator-only network or an RGAN, is beneficial for the task of virtually staining skin tissue.

The seven professionals presented with the qualitative questionnaire deemed the stain quality and usefulness of the generated images as high as their chemically stained counterparts. The questionnaire also showed that the generated images could be correctly diagnosed as healthy or containing cancerous tissue with 95.83% accuracy. It can therefore also be concluded that the tissue virtually stained using a **VS-RGAN** with a PLS stack as input was not significantly different in stain quality or usefulness compared to their chemically stained counterparts. Furthermore, the ability to correctly diagnose cancerous tissue was not negatively impacted by the virtual staining process.

Architecture experiments, changing the number of input channels, loss functions, upsampling methods, etc., did not generally show statistically significant improvements, although visual inspection confirmed improvements in specific and relevant areas. This led us to conclude that it would be beneficial to the method to annotate relevant tissue sections and measure metrics in these areas specifically. Challenges with data limitation and variation were identified in the first dataset and alleviated to some extent by the introduction of the second. Partitioning of the second dataset was also improved to ensure the metric scores reflected the generalizability of the model more accurately. The small difference between the metric scores of the validation and test data indicates that the model is well generalized.

Although our study would benefit from further development and an extension of the dataset, it provides evidence of an impressively capable virtual staining network. It therefore adds to the feasibility of virtual staining using PLS microscopy, and expands the concept to the field of pathology and generation of large scale images. The results are promising for virtual staining's possible application as a standardized, fast, cheap, and more sustainable alternative to chemical staining.

# 7 Future work

One weakness which was identified by several of the experts in the qualitative questionnaire was the low resolution of the images. To pinpoint a number of pathological states, it is necessary to inspect the internal structure of cells, which requires working in 20-40x magnification. We believe that increasing the magnification of the images used in this project, either by using a different objective or by artificially increasing the NA, would be a fascinating continuation of this project.

Expanding the project to include other types of tissue and, in general, more data would be a relevant next step to assess the applicability and generalizability of a single network for virtually staining a range of tissues. Furthermore we believe that working with tissue that has been annotated with for example origin of the tissue, age of the patient and structural elements would be beneficial. With annotations, we believe that evaluating the strengths and weaknesses of the networks could be measured more accurately with the quantitative metrics which showed poor differentiation in our work despite observed improvements. These metrics could then hopefully guide training parameters and structural changes to the network, or even data collection to try and include more of structures that the network performs poorly on.

This project does not include an ablation study of the LEDs used. Although improvement was seen with the addition of the PLS stack it is therefore unclear what components of the stack were most vital. We suggest that this is studied further to make the data collection process more efficient and to further understand which illumination angles are of importance and possibly the physics behind why that is. The choice of exposure levels was also not experimented with in this project. Although avoiding overexposure is a sound choice some observations were made where thinner areas of tissue may have been underexposed in the PLS stack which may have been a detriment to the results. We would therefore encourage exploration of a range of exposure levels and experimentation with data collection in general to enable more even exposure levels throughout a tissue sample.

A further use of the virtual staining is the standardization of staining which could increase the applicability of other machine learning and image analysis methods. An interesting aspect of this is to integrate the virtual staining process in a network which has the end goal to pre-classify different pathological states, such as carcinomas.

# References


[1] Regionala CancerCentrum. Väntetider i standardiserade vårdförlopp (svf). https://cancercentrum.se/samverkan/vara-uppdrag/statistik/svf-statistik/vantetider-i-svf/, May 2023. Accessed: 2023-12-12.

[2] Centrum för Medicinsk Bildvetenskap och Visualisering, Linköpings Universitet. Digital pathology. https://liu.se/artikel/digital-pathology, Sep 2019. Accessed: 2023-12-12.

[3] C. Jansson and I. Schmidt. Nationellt och regionalt arbete med standardiserade vårdförlopp 2022. Audit Report Diarienummer: 3.1.1-2021-0803r, Socialstyrelsen, Stockholm, Sweden, 2022.

[4] S. C. Lester. *Manual of Surgical Pathology*. Elsevier, 3 edition, 2010. ISBN 9780323546324. Language: English.

[5] H. A. Alturkistani et al. Histological stains: A literature review and case study. *Global Journal of Health Science*, 8(3):72–79, 2015. doi: 10.5539/gjhs.v8n3p72.

[6] Sigma-Aldrich. Histopathology. https://www.sigmaaldrich.com/SE/en/applications/clinical-testing-and-diagnostics-manufacturing/histology, n.d. Accessed: 2023-12-12.

[7] M. N. Gurcan et al. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009. doi: 10.1109/RBME.2009.2034865.

[8] B. Bai, X. Yang, and Y. Li et al. Deep learning-enabled virtual histological staining of biological samples. *Light Science Applications*, 12:57, 2023. doi: 10.1038/s41377-023-01104-7.

[9] Y. Zhang, K. de Haan, and Y. Rivenson et al. Digital synthesis of histological stains using micro-structured and multiplexed virtual staining of label-free tissue. *Light: Science Applications*, 9(1):78, 2020. doi: 10.1038/s41377-020-0315-y. URL https://doi.org/10.1038/s41377-020-0315-y.

[10] S. Koivukoski et al. Unstained tissue imaging and virtual hematoxylin and eosin staining of histologic whole slide images. *Lab Invest*, 103(5):100070, May 2023. doi: 10.1016/j.labinv.2023.100070. Epub 2023 Jan 25. PMID: 36801642.

[11] Royal College of Pathologists. Digital pathology. https://www.rcpath.org/profession/digital-pathology.html. Accessed: 2023-12-11.

[12] Y. Rivenson, H. Wang, and Z. Wei et al. Virtual histological staining of unlabelled tissue-autofluorescence images via deep learning. *Nature Biomedical Engineering*, 3: 466–477, June 2019. doi: 10.1038/s41551-019-0362-y.

[13] S. Heffner, O. Colgan, and C. Doolan. Digital pathology. https://www.leicabiosystems.com/en-se/knowledge-pathway/digital-pathology/#What_are_the_Benefits_of_Digital_Pathology. Accessed: 2023-10-18.

[14] G. Zheng, R. Horstmeyer, and C. Yang. Wide-field, high-resolution fourier ptychographic microscopy. *Nature Photonics*, 2013. https://www.nature.com/articles/nphoton.2013.187.

[15] J. Wulff. Virtual staining of blood cells using point light source illumination and deep learning. *Lunds Tekniska Högskola*, 2022. https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9074363&fileOId=9074364.

[16] Y. Benny, T. Galanti, and S. Benaim et al. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731, March 2021. ISSN 1573-1405. doi: 10.1007/s11263-020-01424-w. URL http://dx.doi.org/10.1007/s11263-020-01424-w.

[17] A. H. Fischer et al. Hematoxylin and eosin staining of tissue and cell sections. *CSH Protoc*, page pdb.prot4986, May 2008. doi: 10.1101/pdb.prot4986.

[18] Leica Biosystems. He staining overview: A guide to best practices. https://www.leicabiosystems.com/en-se/knowledge-pathway/he-staining-overview-a-guide-to-best-practices/, 2023. 2023-10-09.

[19] *Normal Skin Histology - Explained by a Dermatopathologist*. J. Gardner, Aug 2016.

[20] Basal cell carcinoma. https://www.mayoclinic.org/diseases-conditions/basal-cell-carcinoma/symptoms-causes/syc-20354187, Oct 2021. Accessed: 05-01-2024.

[21] R. Rottenfusser. *Chapter 3 - Proper Alignment of the Microscope*, volume 114 of *Methods in Cell Biology*. Academic Press, 2013. doi: https://doi.org/10.1016/B978-0-12-407761-4.00003-8.

[22] OpenCV. Object detection: Image processing. https://docs.opencv.org/4.x/df/dfb/group__imgproc__object.html#gga3a7850640f1fe1f58fe91a2d7583695dac5babb7dfda59544e3e31ea928f8cb16, 2023. Acessed: 2023-10-09.

[23] Wolfram Research. Labcolor. https://reference.wolfram.com/language/ref/LABColor.html, 2021. Accessed: 2023-12-16.

[24] S. McHugh. Sharpening: Unsharp mask. https://www.cambridgeincolour.com/tutorials/unsharp-mask.htm. Accessed: 2024-01-03.

[25] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, Upper Saddle River, NJ, 3rd edition, 2018. ISBN 978-0136042594.

[26] A. M. Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[27] IBM. What is artificial intelligence (ai)? https://www.ibm.com/topics/artificial-intelligence, 2023. Accessed 2023-12-15.

[28] J. McCarthy. What is artificial intelligence? *Computer Science Department, Stanford University*, November 2007.

[29] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 5, pages 96–161. MIT Press, 2016. Chapter Title: Machine Learning Basics.

[30] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[31] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

[32] P. Baheti. Train test validation split: How to best practices. https://www.v7labs.com/blog/train-validation-test-set, 2023. Accessed: 2023-12-04.

[33] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 6, pages 164–223. MIT Press, 2016. Chapter Title: Deep Feedforward Networks.

[34] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 1, pages 1–26. MIT Press, 2016. Chapter Title: Introduction.

[35] Stanford University. Neural networks part 1: Setting up the architecture - cs231n: Convolutional neural networks for visual recognition. https://cs231n.github.io/neural-networks-1/, 2016. Accessed 2023-12-14.

[36] IBM. Neural networks - ibm. https://www.ibm.com/topics/neural-networks, 2023. Accessed 2023-12-14.

[37] DeepAI. Hidden layer - deepai. https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning, 2023. Accessed 2023-12-14.

[38] K. Shen. Effect of batch size on training dynamics. https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e. Accessed: 2024-01-05.

[39] Z. Graves. Loss functions and their use in neural networks. https://medium.com/neural-network-nodes/overtraining-neural-networks-trends-vs-noise-e6e50aa5ef52. Accessed: 2024-01-02.

[40] Google Developers. Data size and data quality. https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality, 2023. Accessed: 2023-12-14.

[41] Harvard John A. Paulson School of Engineering and Applied Sciences. Neural network trained using diverse dataset outperforms conventionally trained. https://seas.harvard.edu/news/2023/02/neural-network-trained-using-diverse-dataset-outperforms-conventionally-trained, February 2023. Accessed: 2023-12-14.

[42] DeepAI. Exploding gradient problem. https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem#:~:text=The%20exploding%20gradient%20problem%20is,(weights)%20become%20excessively%20large. Accessed: 2023-12-04.

[43] P. Baheti. Activation functions in neural networks [12 types use cases]. https://www.v7labs.com/blog/neural-networks-activation-functions, 2021. Acessed:2023-10-10.

[44] M. Xiang. Convolutions: Transposed and deconvolution. *Medium*, Mar 2021.

[45] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning, 2018. Accessed: 17-11-2023.

[46] PyTorch. torch.nn.upsamplingnearest2d — pytorch 1.x documentation. https://pytorch.org/docs/stable/generated/torch.nn.UpsamplingNearest2d.html, 2023. Accessed: 2023-12-14.

[47] W. Shi et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *arXiv*, Sep 2016.

[48] M. Qin et al. Remote sensing single-image resolution improvement using a deep gradient-aware network with image-specific enhancement. *Remote Sensing*, 12:758, 02 2020. doi: 10.3390/rs12050758.

[49] M. Galar et al. Super-resolution of sentinel-2 images using convolutional neural networks and real ground truth data. *Remote Sensing*, 12:2941, 09 2020. doi: 10.3390/rs12182941.

[50] J. Huber. Batch normalization in 3 levels of understanding. *Medium*, Nov 2020.

[51] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 4, pages 78–95. MIT Press, 2016. Chapter Title: Numerical Computation.

[52] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 8, pages 271–325. MIT Press, 2016. Chapter Title: Optimization for Training DeepModels.

[53] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[54] GeeksforGeeks. Ml — stochastic gradient descent (sgd). https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/, 2023. Acessed: 2023-10-30.

[55] A. Ajagekar. Adam - cornell university computational optimization open textbook. https://optimization.cbe.cornell.edu/index.php?title=Adam, Dec 2021. Accessed: 2023-11-20.

[56] V. Yathish. Loss functions and their use in neural networks. https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9. Accessed: 2023-12-23.

[57] Mseloss pytorch 2.1 documentation. https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html. Accessed: 10-12-2023.

[58] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *Computer Vision – ECCV*, 2016. https://arxiv.org/pdf/1603.08155.pdf.

[59] D. Fuoli, L. V. Gool, and R. Timofte. Fourier space losses for efficient perceptual image super-resolution. *IEEE Xplore*, 2021. https://openaccess.thecvf.com/content/ICCV2021/papers/Fuoli_Fourier_Space_Losses_for_Efficient_Perceptual_Image_Super-Resolution_ICCV_2021_paper.pdf.

[60] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 9, pages 326–366. MIT Press, 2016. Chapter Title: Convolutional Neural Networks.

[61] IBM. What are convolutional neural networks? https://ibm.com/topics/convolutional-neural-networks, 2023. Acessed: 2023-10-30.

[62] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015. Accessed: 16-11-2023.

[63] S. Cai et al. Dense-unet: A novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network. *Quantitative Imaging in Medicine and Surgery*, 10(6):1275–1285, Jun 2020. doi: 10.21037/qims-19-1090.

[64] O. Oktay et al. Attention u-net: Learning where to look for the pancreas, 2018. Accessed: 05-01-2023.

[65] R. Vinod. A detailed explanation of the Attention U-Net — towardsdatascience.com. https://towardsdatascience.com/a-detailed-explanation-of-the-attention-u-net-b371a5590831. Accessed: 16-11-2023.

[66] I. Goodfellow et al. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[67] Towards Data Science. Understanding binary cross-entropy / log loss: A visual explanation. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a, 2023. Accessed: 2023-12-10.

[68] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan, 2018. Accessed: 16-11-2023.

[69] Google Developers. Common problems with gans. https://developers.google.com/machine-learning/gan/problems, 2023. Accessed: 2023-12-14.

[70] J. Hui. Gan - why it is so hard to train generative adversarial networks! https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b, Apr 2023. Accessed: 20-12-2023.

[71] P. Isola. Image-to-image translation with conditional adversarial networks, 2018. Accessed: 05-12-2023.

[72] X. Wang et al. Real-esrgan: Training real-world blind super-resolution with pure synthetic data, 2021. Accessed: 14-12-2023.

[73] Z. Wang. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[74] J. Nilsson and T. Akenine-Möller. Understanding ssim. *arXiv*, abs(2006.13846), 2020.

[75] M. P. Sampat et al. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385, Nov 2009.

[76] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005. doi: 10.1109/MSP.2005.1550194.

[77] M. Heusel, H. Ramsauer, and T. Unterthiner et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. Accessed: 16-12-2023.

[78] G. Zhang, B. Ning, and H. Hui et al. Image-to-images translation for multiple virtual histological staining of unlabeled human carotid atherosclerotic tissue. *Molecular Imaging and Biology*, 24:31–41, 2022. doi: 10.1007/s11307-021-01641-w. URL `https://doi.org/10.1007/s11307-021-01641-w`.

# A Appendix

## A.1 H&E staining protocols

### A.1.1 In-house staining at CellaVision

| Reagent | Time [mm:ss] | Comment |
|---|---|---|
| Xylenes, 99%, Thermo Scientific | 02:00 | |
| Xylenes, 99%, Thermo Scientific | 02:00 | |
| Ethanol, absolute 99.8%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 99.8%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 95%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 90%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol 70% in water solution, Chemlab | 01:00 | Agitate when inserting and removing slide from bath |
| Water | 01:00 | Rinse in running tap water |
| Mayer hemalum, RAL | 03:00 | |
| Water | 03:00 | Rinse in running tap water |
| Eosin, 1% in aqueous solution, RAL | 07:00 | |
| Water | - | Rinse briefly in running tap water |
| Ethanol 70% in water solution, Chemlab | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 90%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 95%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 99.8%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Ethanol, absolute 99.8%, Fisher Scientific | 01:00 | Agitate when inserting and removing slide from bath |
| Xylenes, 99%, Thermo Scientific | 02:00 | |
| Xylenes, 99%, Thermo Scientific | 02:00 | |
| Mounting Medium, Pertex | - | Mount coverslip onto slide |

Figure 32: Staining protocol used at CellaVision.

## A.1.2 At Blekinge Hospital



### Tissue-Tek Prisma Plus

## 2. HTX-Erytrosin

### Syfte
Översiktsfärgning. På en bra översiktsfärgning ställs kravet att det ger en klar avgränsning mellan kärnor och cytoplasma samt tydligt visualisera olika kärn- och vävnadsstrukturer. Storsnittsvarianten står längre tid i ugn och går inte till montering.

### Miljöaspekter
**Arbetsmiljö**
Säkerhetsdatablad för respektive kemikalie, *se kemikaliehanteringssystemet KLARA*.
Allt arbete med xylen ska ske i skyddsventilerat utrymme.

**Avfall**
Isopropanol hälls i utslagsvask i färgrummet. Färglösningar och xylen får ej hällas ut i avlopp. Uppmärkta uppsamlingsdunkar till kemikalier finns i spritrummet *se anvisning Kemikalier för destruktion 18-334*. Resterande lösningar får hällas ut i avloppet.

### Förberedande behandling
Formaldehydfixerat och paraffininbäddat material, snittat med standardmikrotominställning.

### Kontroller
Kontrolglas finns försnittade och körs vid byte av färglösningar 1 gång/vecka. Glas märks med datum och sparas i 2 månader vid mikroskopet vid utlämningen. Anteckna på *Loggblad kontrollglas 18-640*.

### Apparatur/Tillbehör
Sakura Tissue-Tek Prisma Plus

### Utförande - HTX-Erytrosin och Stor HTX

| Utförande | | Tider |
|---|---|---|
| 1 | Ugn 60º C | 15 min (Stor 20 min) |
| 2 | Xylen | 2 min |
| 3 | Xylen | 2 min |
| 4 | Isopropanol | 2 min |
| 5 | Isopropanol | 2 min |
| 6 | Wash Station rinnande kranvatten | 1 min |
| 7 | Harris Htx | 6 min |
| 8 | Wash Station rinnande kranvatten | 2 min |
| 9 | HCl-dest 0,1 % | 2 min |
| 10 | Wash Station rinnande kranvatten | 4 min (Stor 2 min) |
| 11 | Erytrosin 0,4 % | 6 min |
| 12 | Wash Station rinnande kranvatten | 1 min |
| 13 | Isopropanol | 1 min |
| 14 | Isopropanol | 1 min |
| 15 | Isopropanol | 1 min |
| 16 | Xylen | 1:30 min |
| 17 | Xylen | 1:30 min |
| 18 | Montering i maskin (Stor slutstation Iso) | |

# Tissue-Tek Prisma Plus

**Lösningar**

| | |
|---|---|
| HCl-dest 0,1 % | 2 ml HCl + 2000 ml dest. vatten |
| Erytrosin 0,4 % | 8 g erytrosin + 2000 ml dest. vatten |

## Bakgrund/Princip

Hematoxylin är ett av de viktigaste kärnfärgämnena inom histologisk diagnostik. Lösningen måste ”mogna” och därför tillsätts oxidationsmedel natriumjodat för att påskynda processen. Oxidationsprodukten kallas hematein som är ett surt färgämne, som ger svaga, ospecifika, brunaktiga färgningsresultat. Genom betmedelstillsats får man en komplex förening - ett färglack. Det positivt laddade hemateinkomplexet förenar sig med nukleinsyrans fosforgrupper i kromatinet och färgar cellkärnorna i olika nyanser. Hematoxylin enligt Harris innehåller kalialun (kalium-aluminium-sulfat) som betmedel som ger olika blå nyanser. För cytoplasma - eller motfärgning används surt erytrosin B som är en anjonfärg. Den färgar basiska eller acidofila vävnadskomponenter som cytoplasma, acidofila granula, muskel, kollagen bindväv och erytrocyter och ger en rosaröd ärg.

## Resultat

Kärnor - blå
Cytoplasma - röd-rosa

## Referenser

Modifierad Theory and Practice of Histological Techniques, Bancroft, J, Gamble, M, Fifth edition 2002
Kemikaliehanteringssystemet KLARA

## A.2 Evaluation Questionnaire

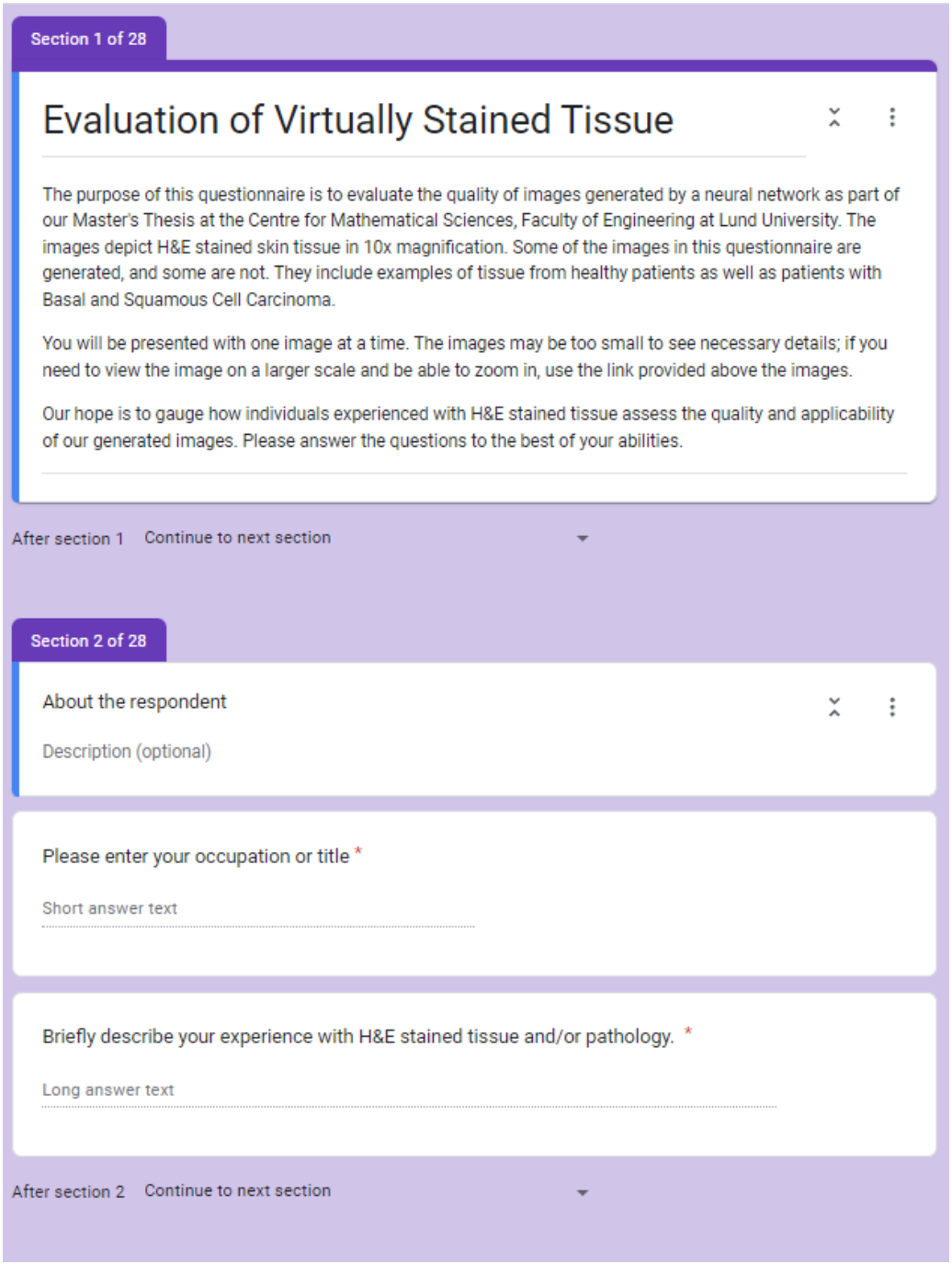


Figure 33: Evaluation questionnaire, introductory page.

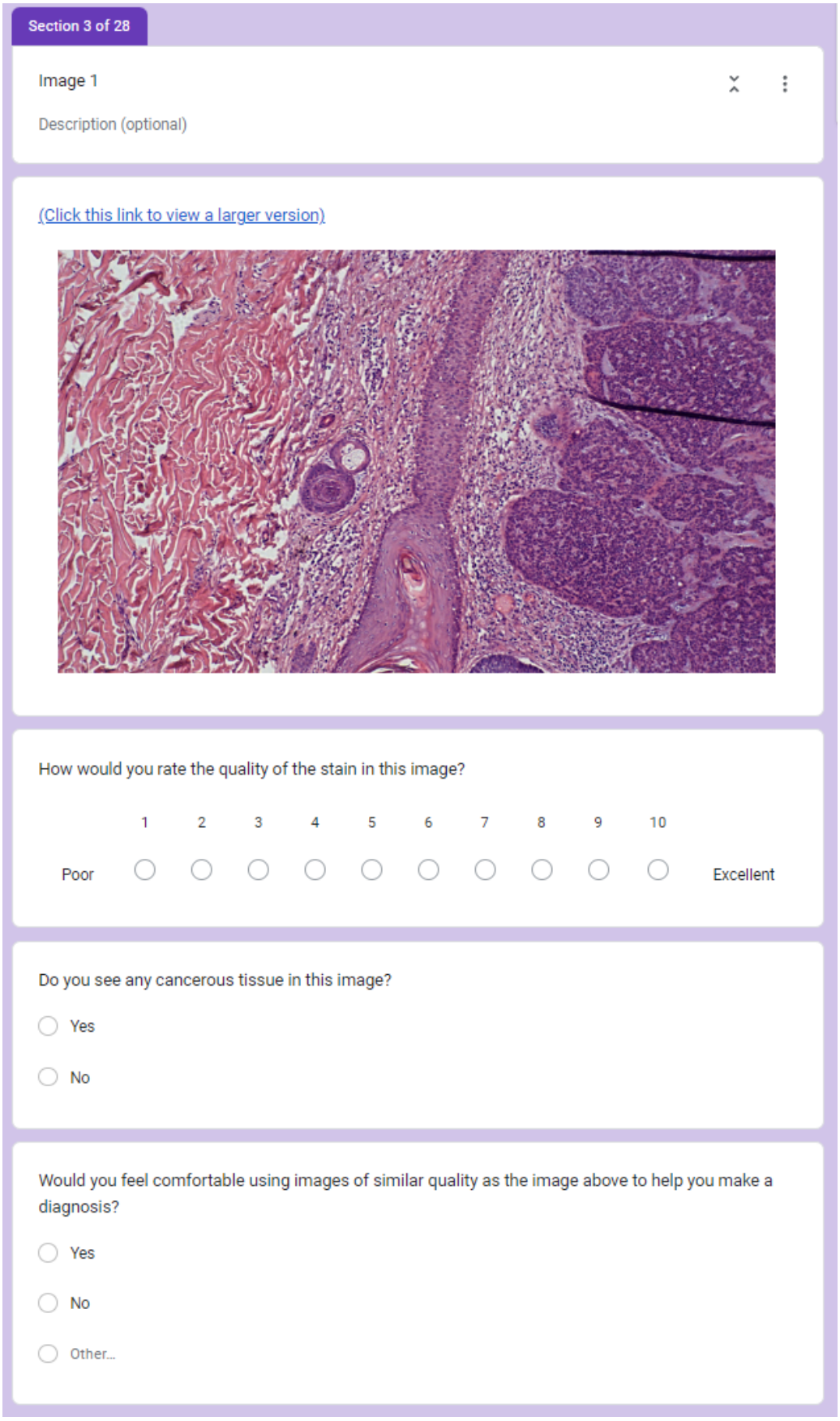


Figure 34: Evaluation questionnaire, questions asked for all included images.

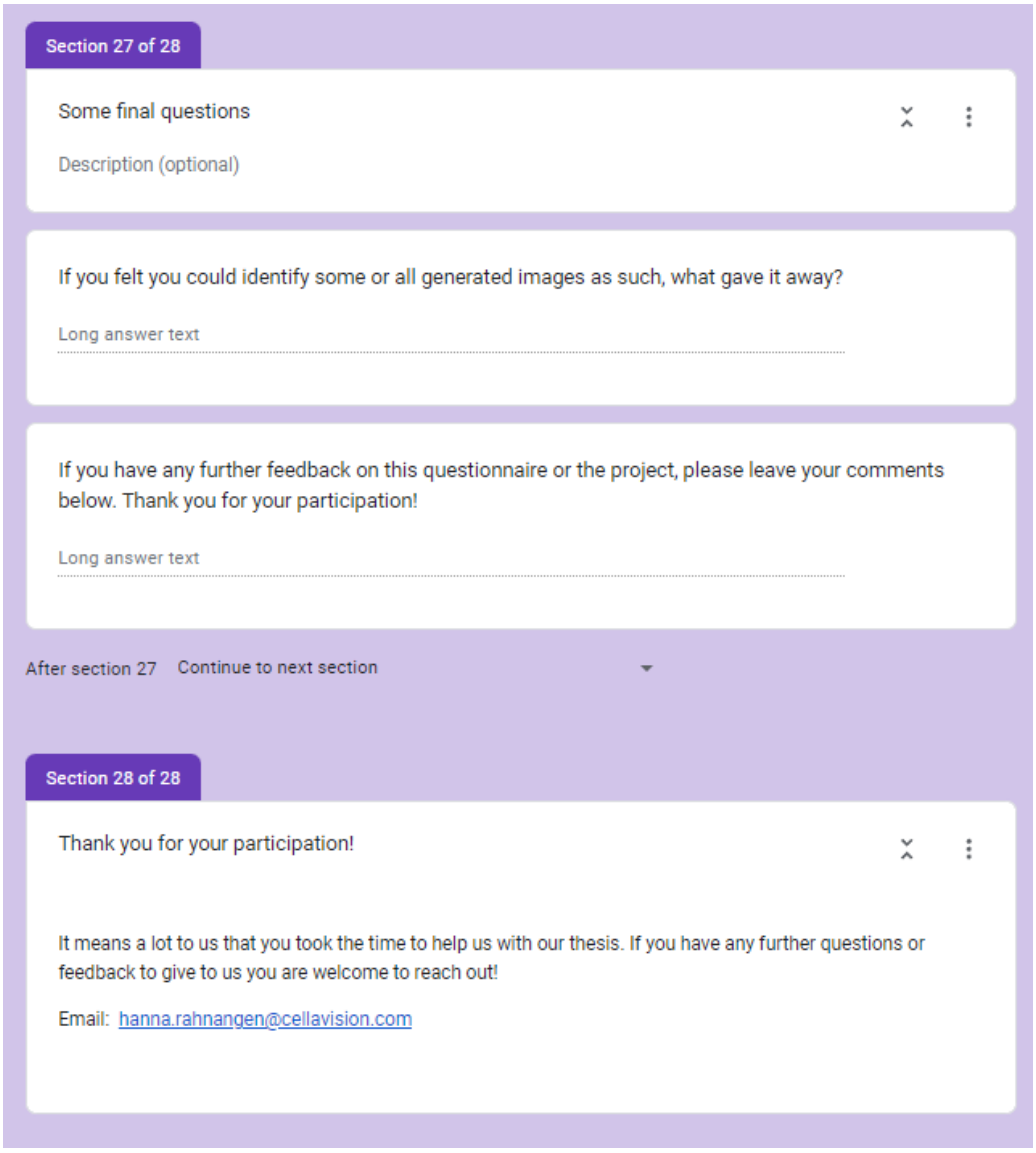


Figure 35: Evaluation questionnaire, final free form questions.

## A.3 Full PLS stack

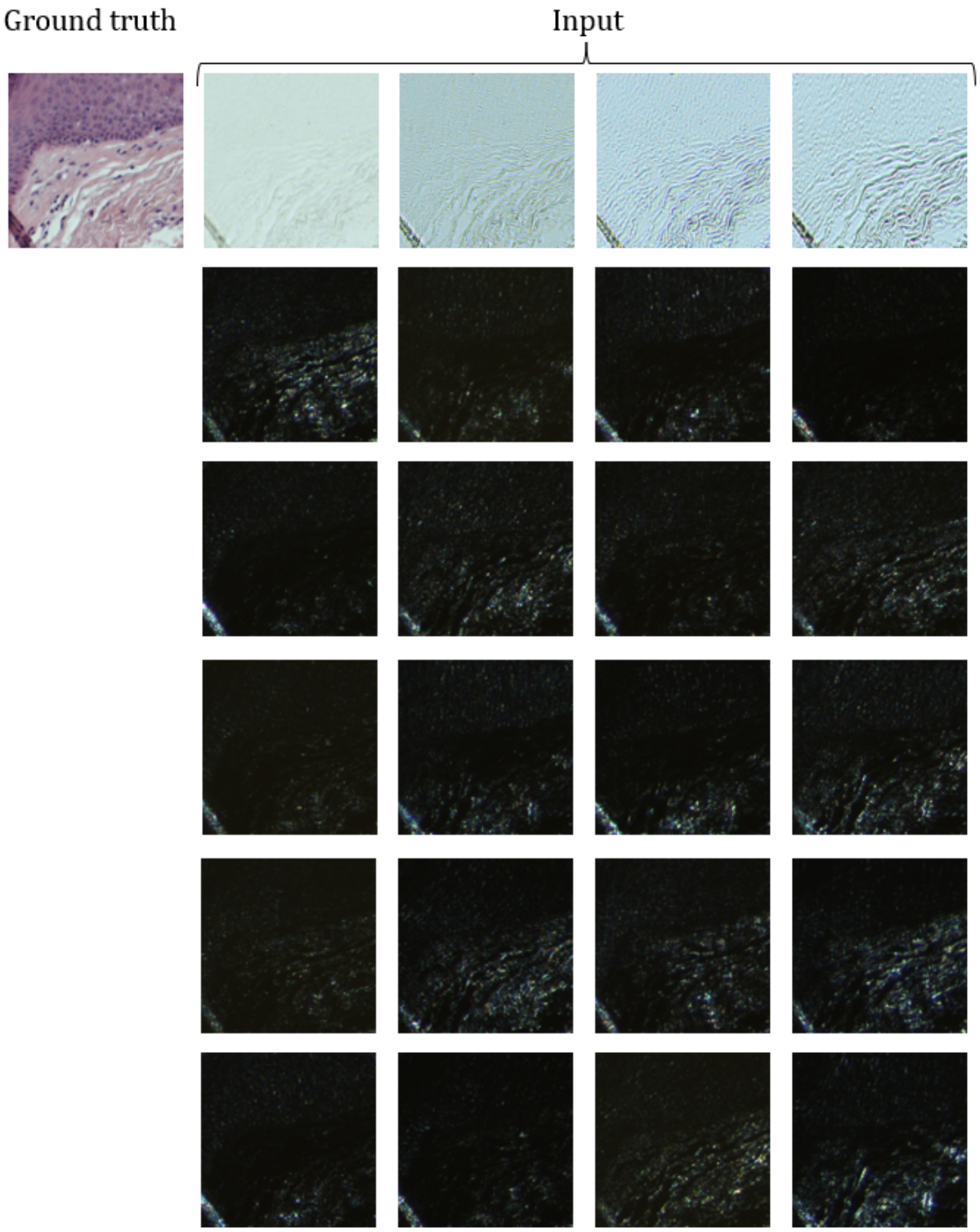


Figure 36: Example of the full PLS stack used throughout the project. Each image shown is an RGB-image, meaning it has three color channels, making the total number of input channels 72 for a full PLS stack.

## A.4 Bright-Field versus PLS

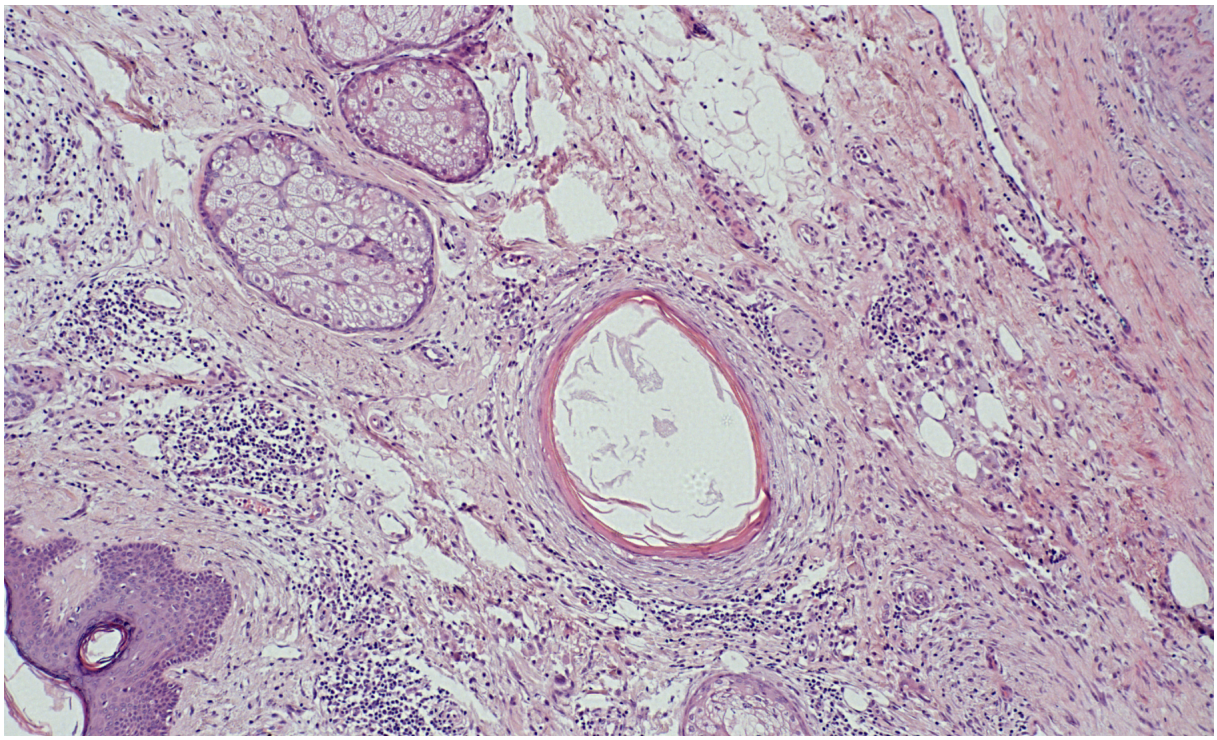

Figure 37: Full sized version of the ground truth image in Figure 28.

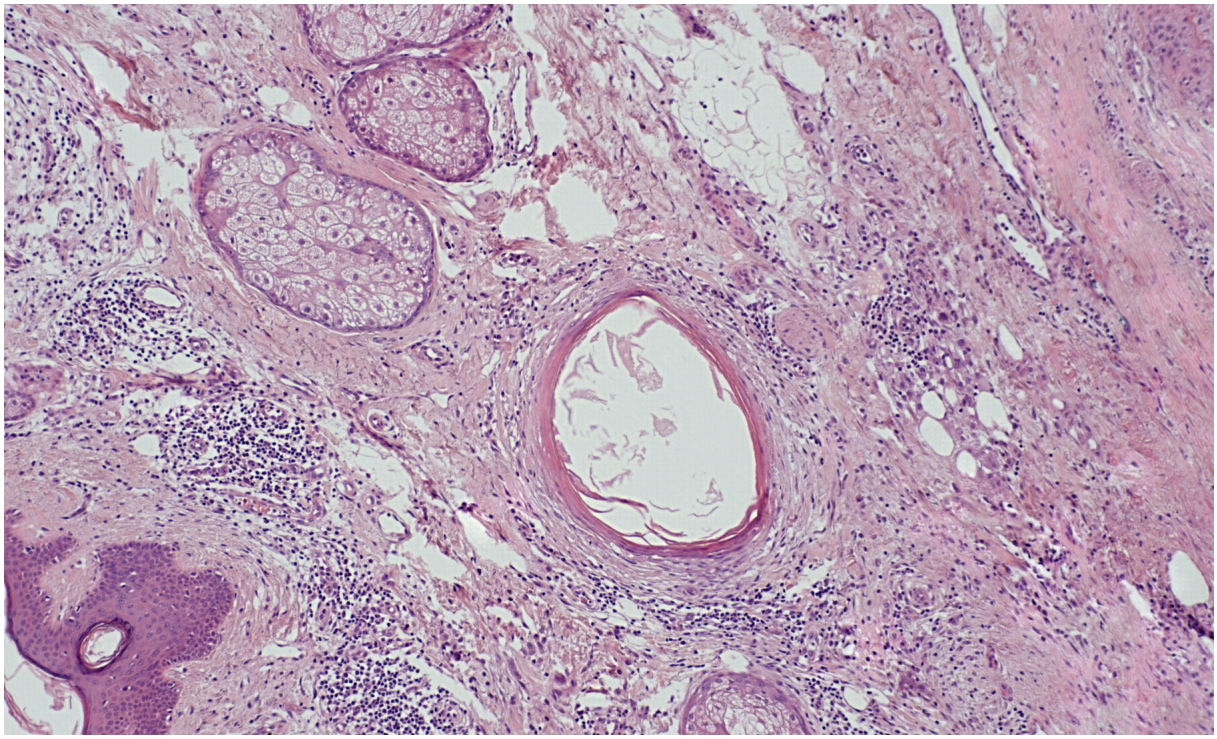

Figure 38: Full sized version of the image in Figure 28 generated by **UNet-96-MSEpf-Pixelshuffle-Attention** trained on PLS image stacks.

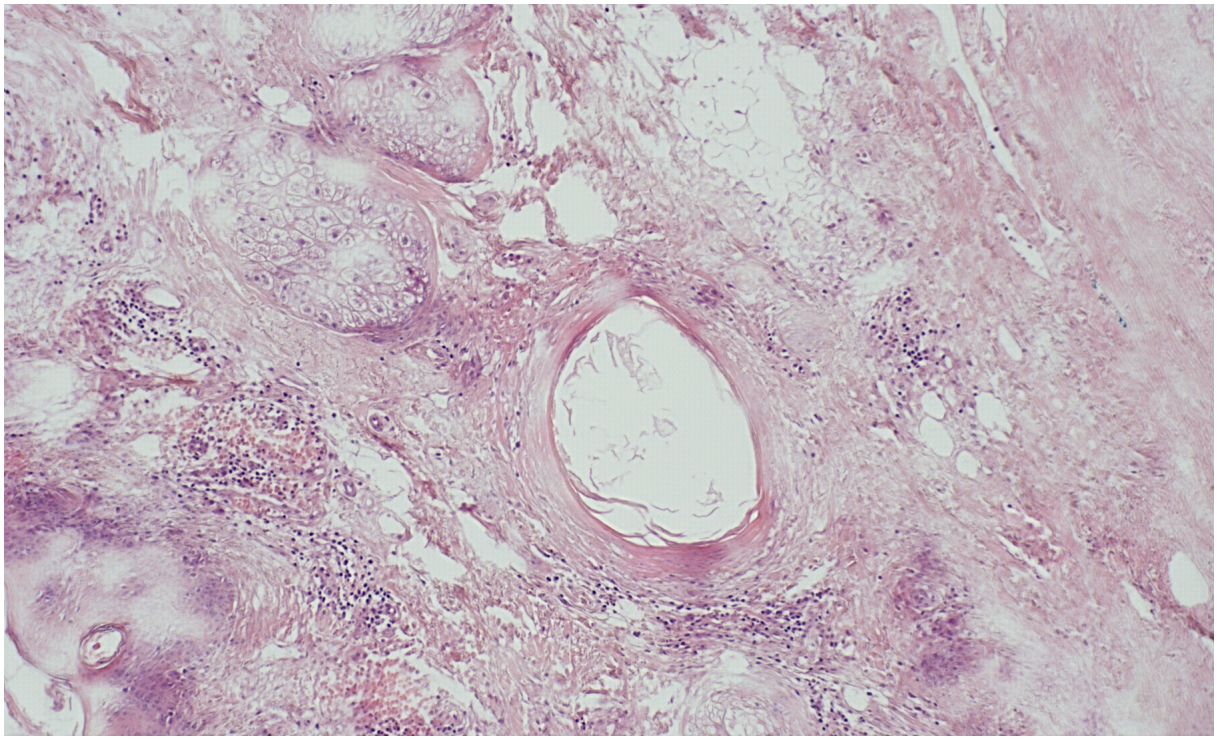

Figure 39: Full sized version of the image in Figure 28 generated by **UNet-96-MSEpf-Pixelshuffle-Attention** trained on BF images.

## A.5 Image Comparison

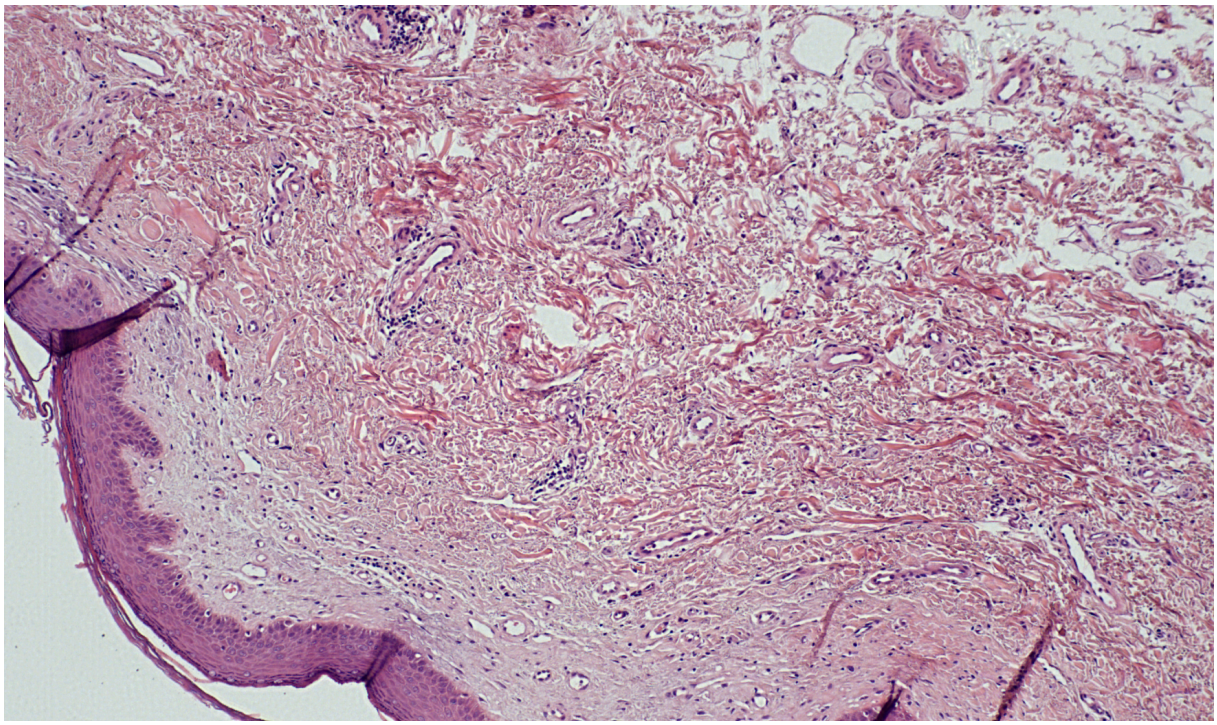

Figure 40: Full sized version of the ground truth image in Figure 31.

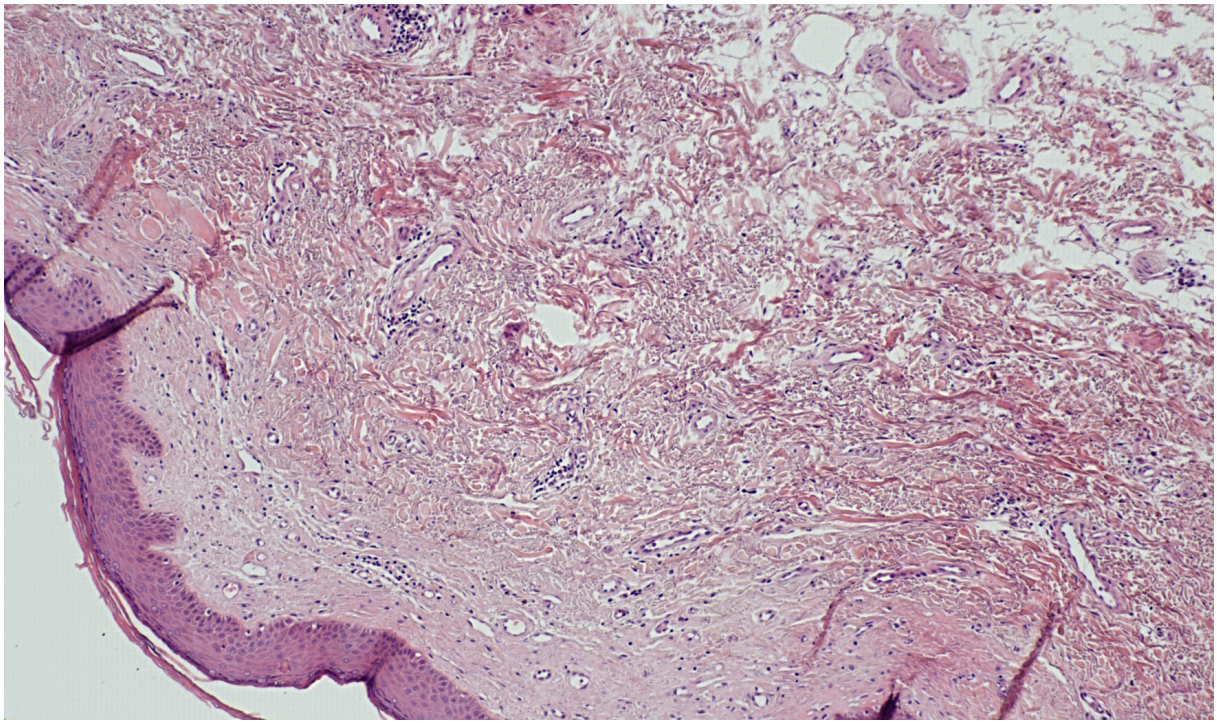

Figure 41: Full sized version of the image in Figure 31 generated by **VS-RGAN**.

Master's Theses in Mathematical Sciences 2024:E1
ISSN 1404-6342

LUTFMA-3522-2024

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/